Occupancy grids

Sudharshan Suresh

Feb 19th, 2020

Slides adapted and sourced from:
Michael Kaess, Ryan Eustice, C. Stachniss, and Joshua Mangelson

In this lecture

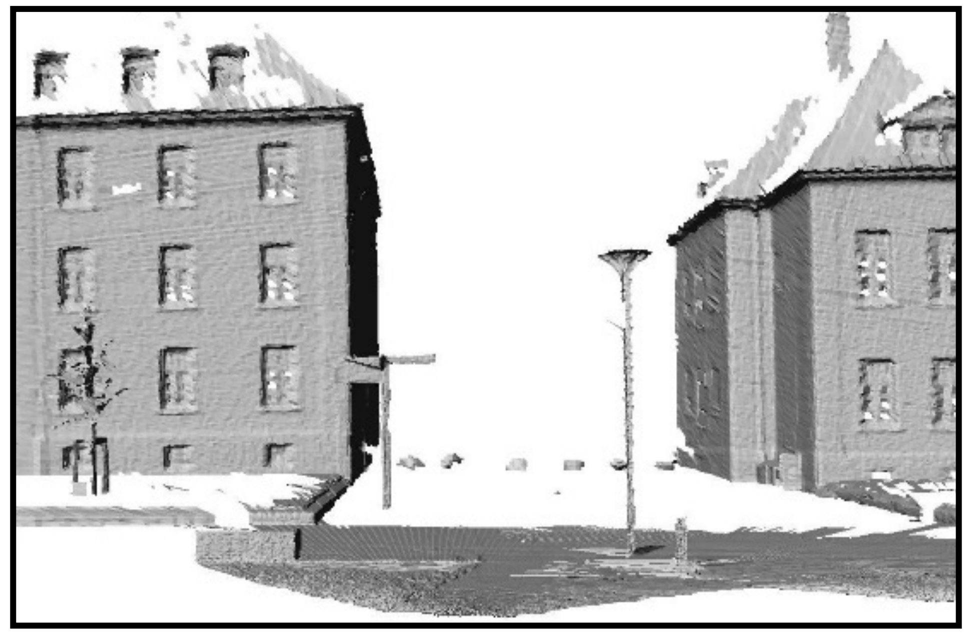
- **Occupancy grid mapping**
  - Representation
  - Assumptions
  - OGM → Bayes filter
  - Log-odds updates
  - Examples and drawbacks

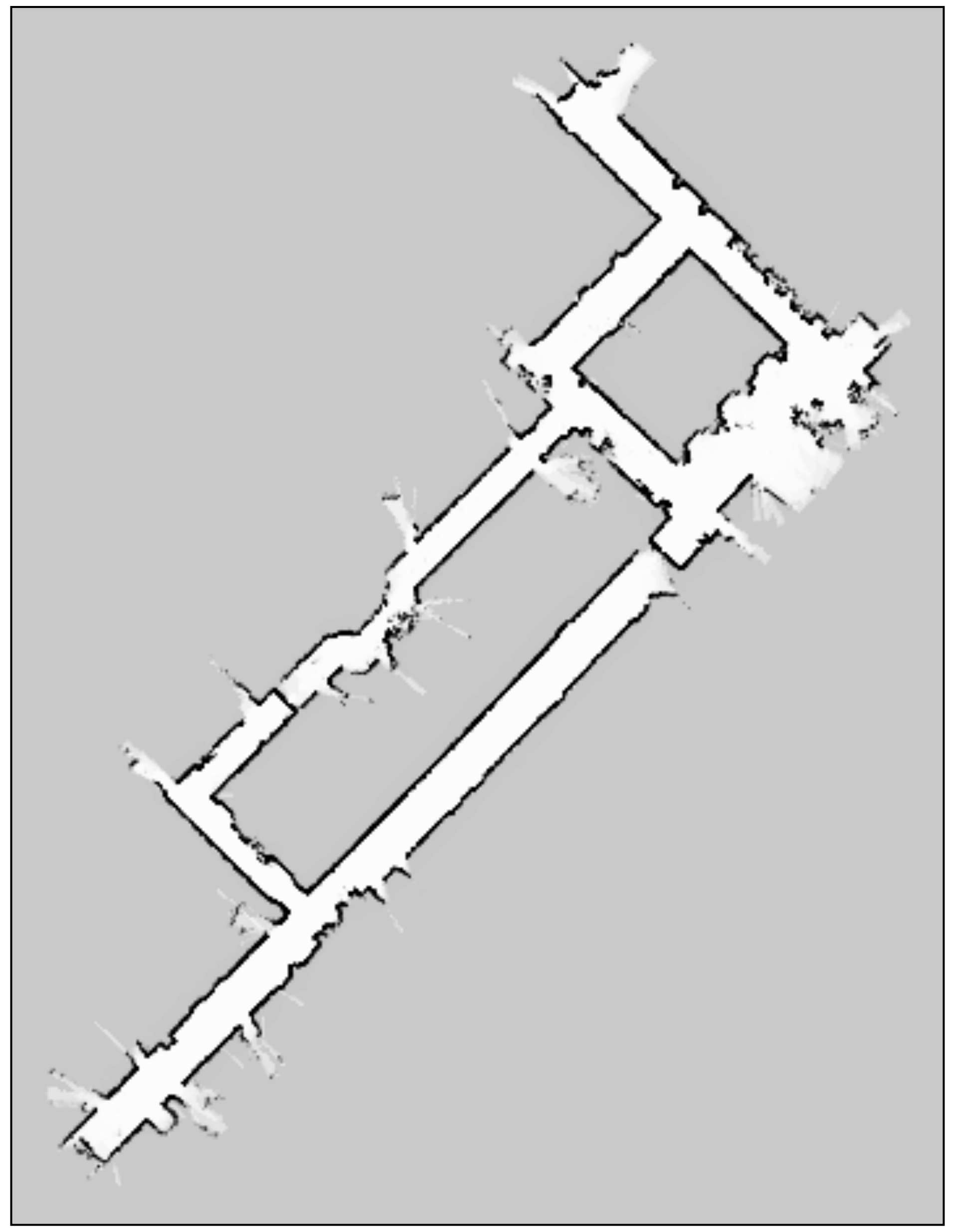
- **Scan-matching**
- **Rao-Blackwellization**
- **Beyond OGMs**

- **Occupancy grid mapping**
    - Representation
    - Assumptions
    - OGM → Bayes filter
    - Log-odds updates
    - Examples and drawbacks

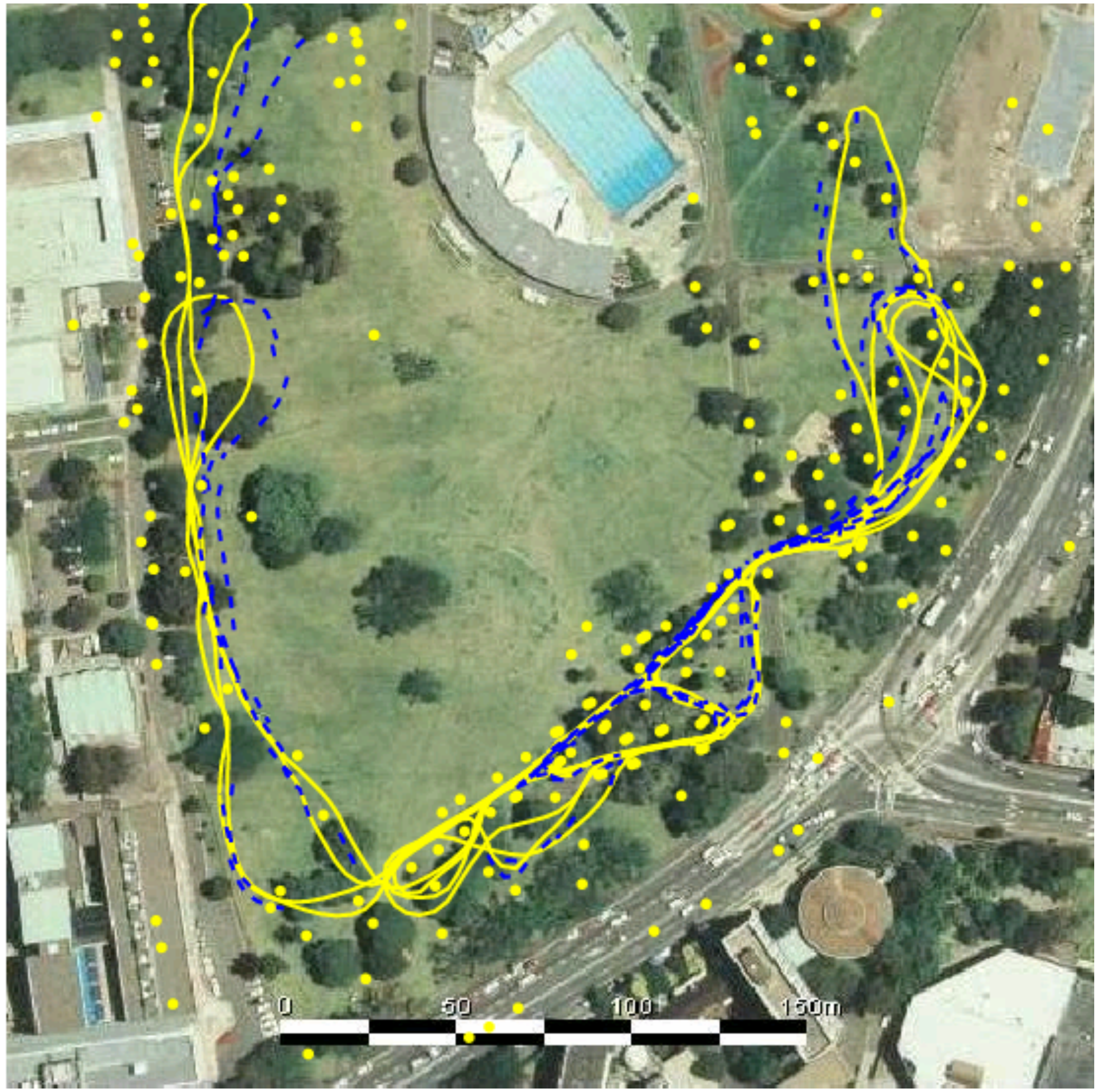
- **Scan-matching**
- **Rao-Blackwellization**
- **Beyond OGMs**

Volumetric vs. feature maps





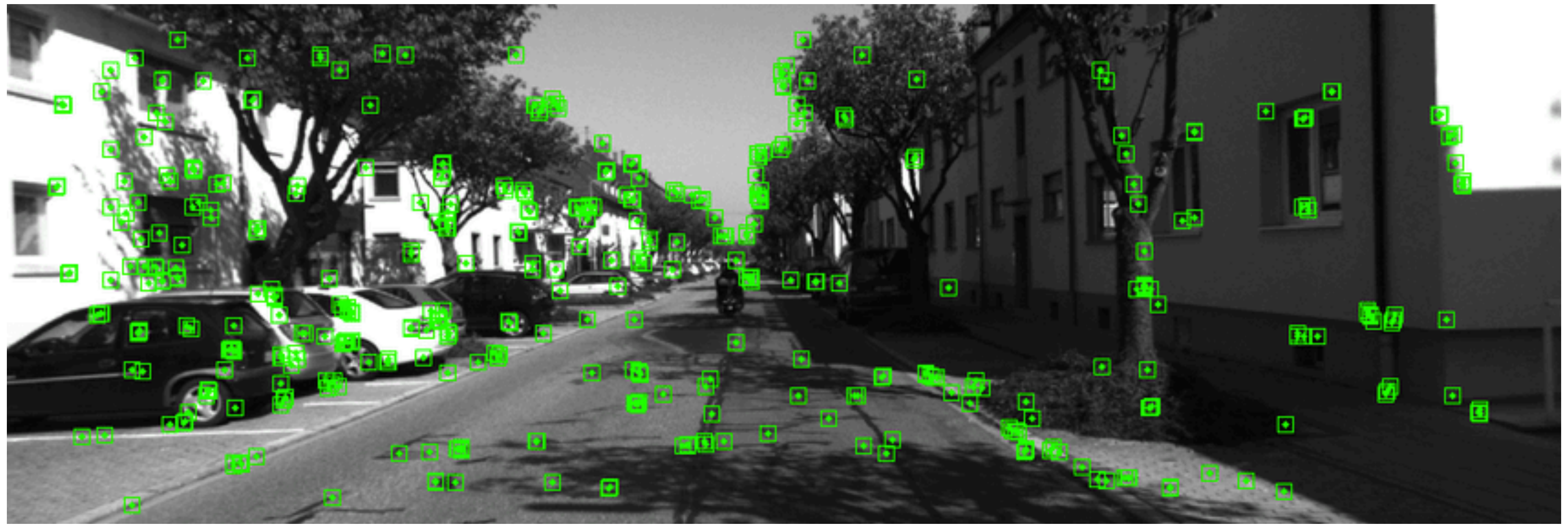
Courtesy: D. Hähnel



Courtesy: E. Nebot

Feature maps

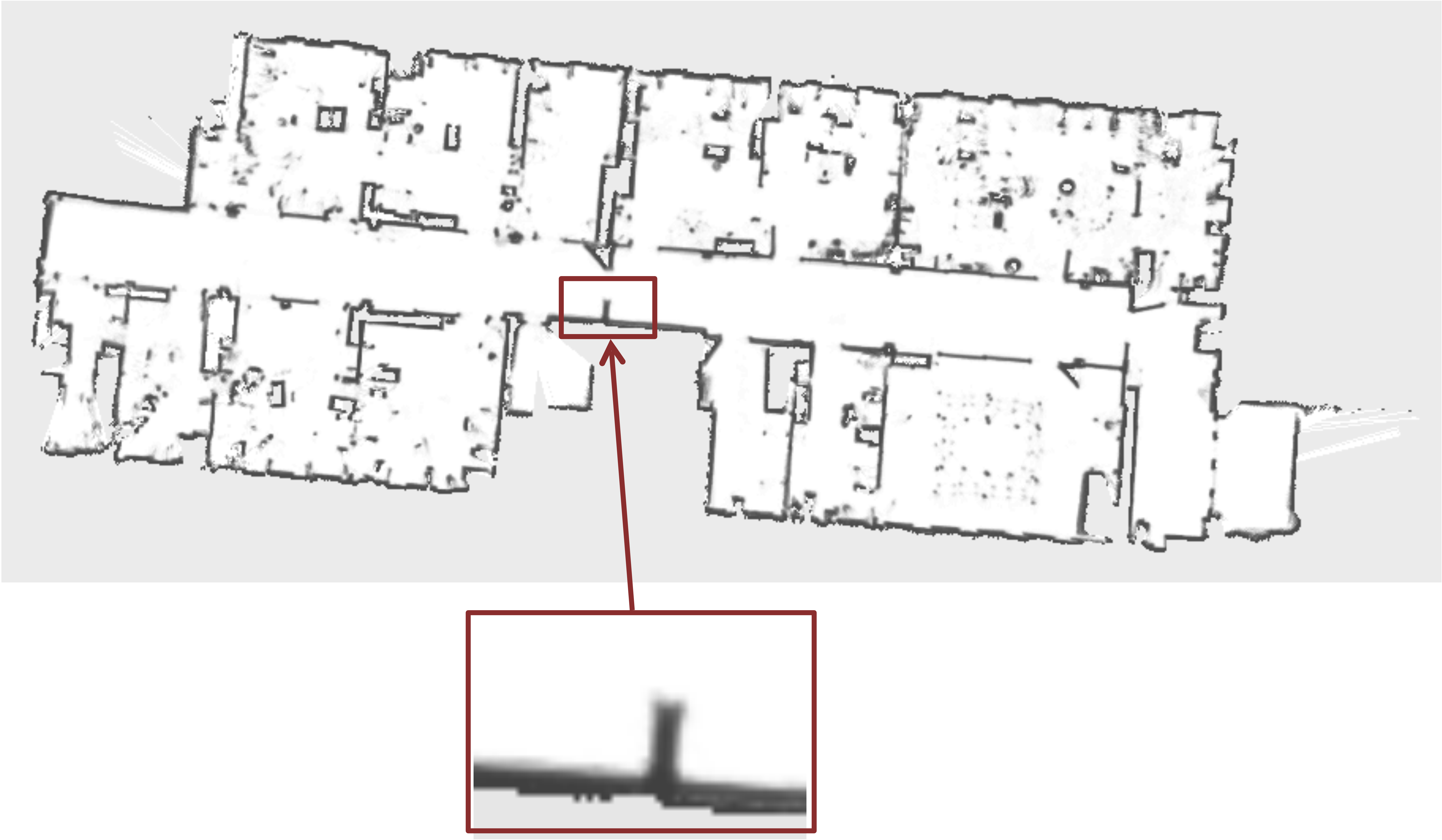
- Most successful SLAM/odometry/VIO implementations use feature maps
- Ideal for SLAM systems
- Compact representation
- Multiple feature observations improve landmark position estimates



Grid maps

- Discretizes environment into cells
- Structure is (usually) rigid
- Cells are assumed to be free or occupied
- Memory requirements scale with map size
- Doesn't depend on feature detection

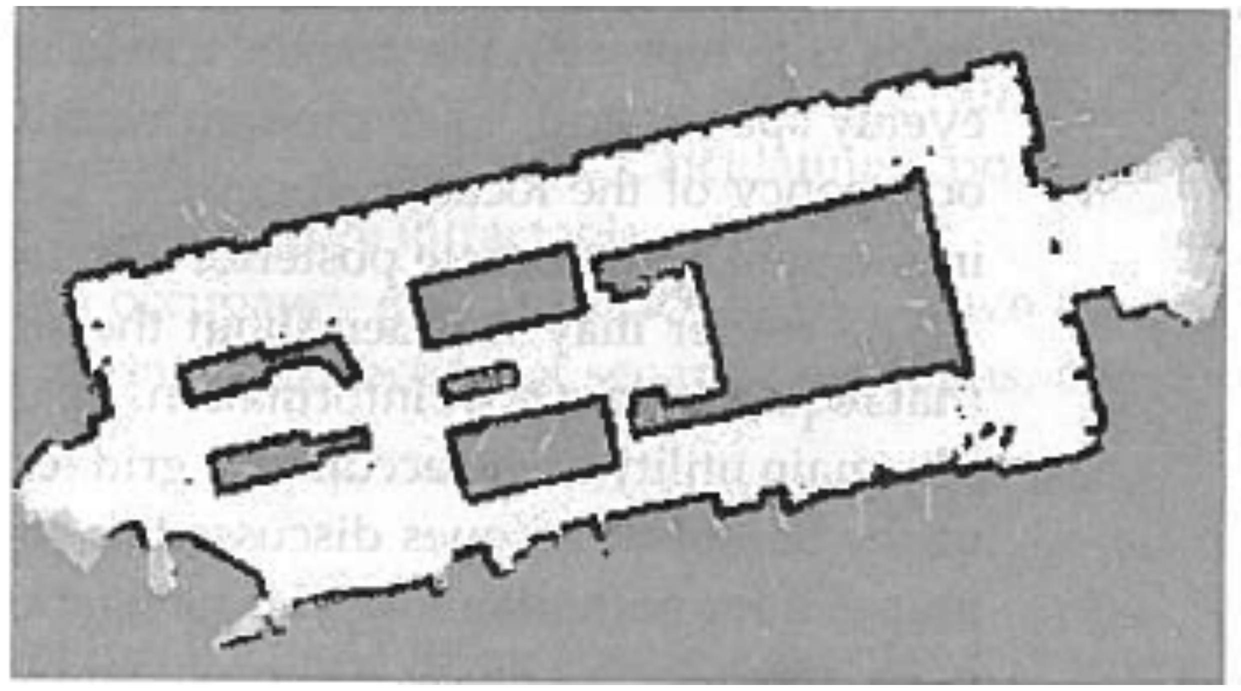
Grid maps

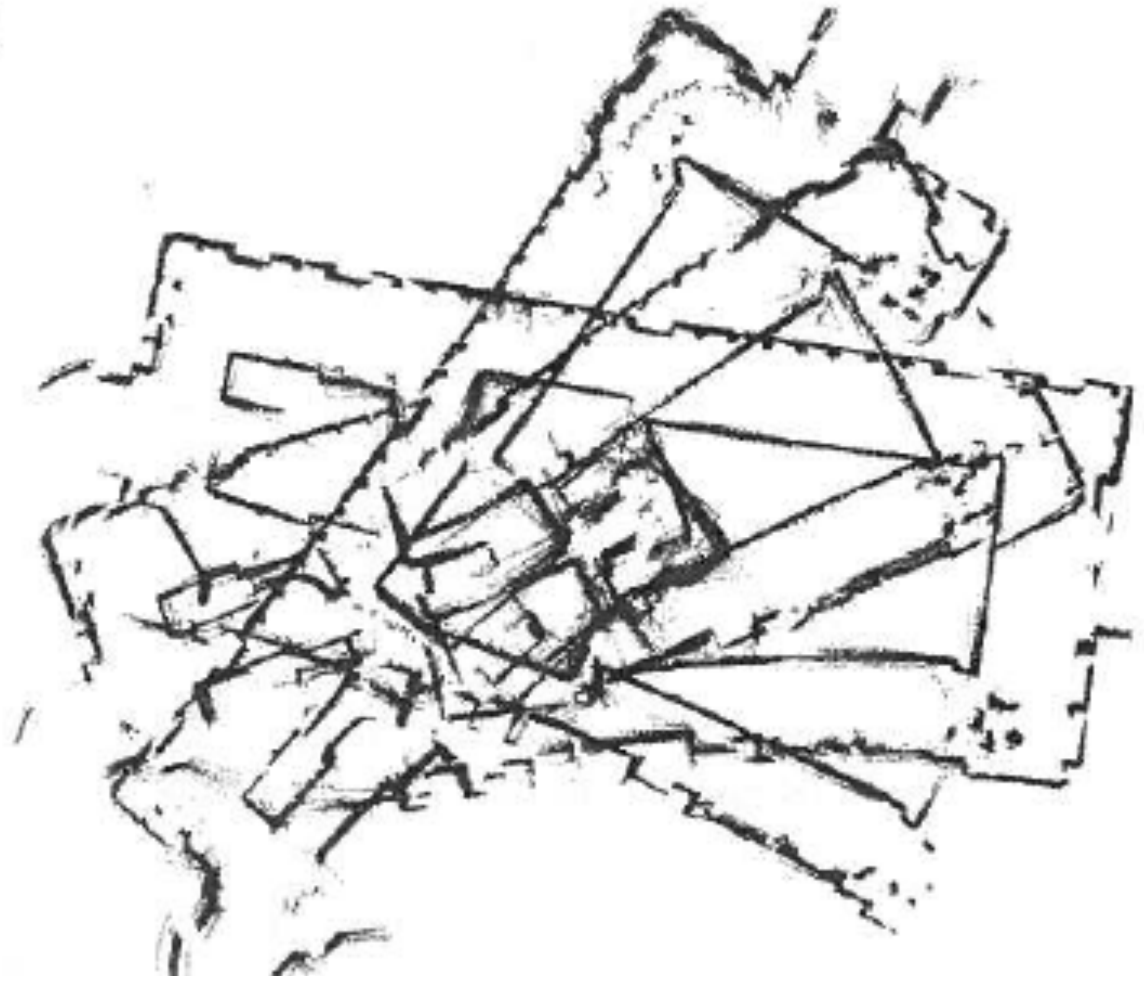


Courtesy: C. Stachniss

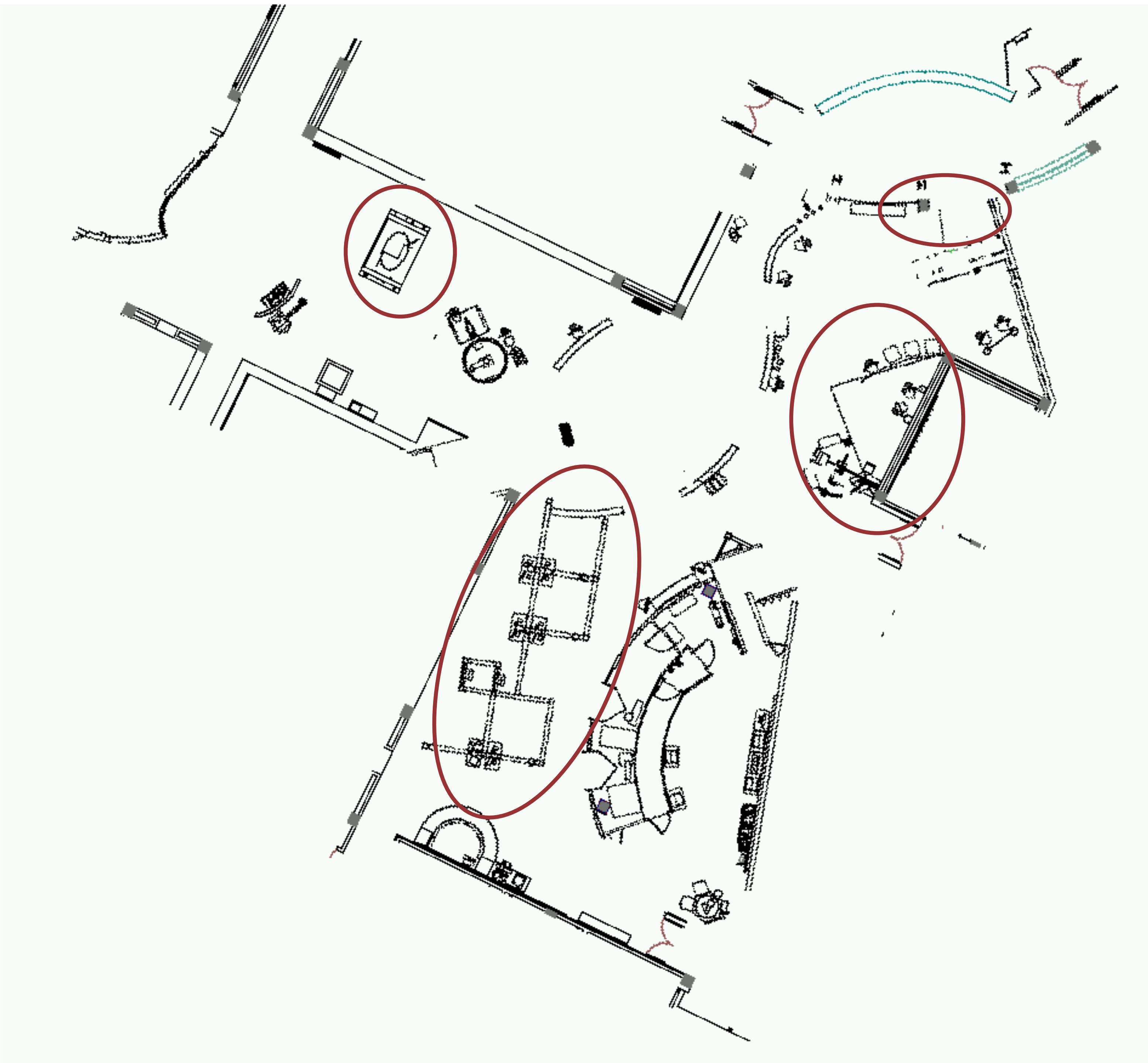
Grid maps encode free space

- **Using robot odometry to place sensor measurements**
  - Inconsistent
  - Robot state drifts with time
  - No notion of free-space
- **Solve for robot state using sensor measurements**
  - Consistent map
  - No notion of free-space
- **Pose-optimization with occupancy map**
  - Satisfies objectives of SLAM and planning

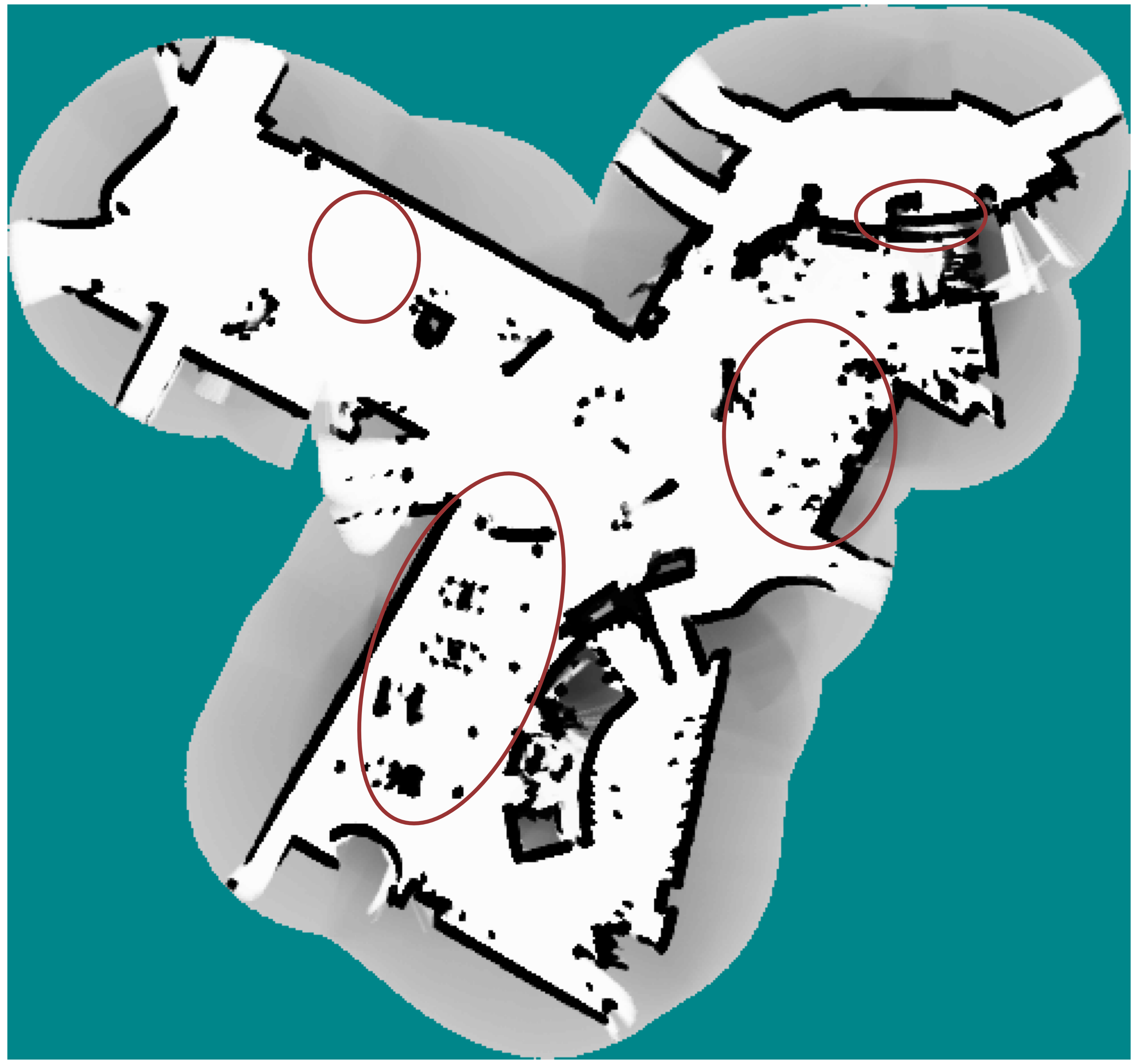




Even prior maps can be inaccurate



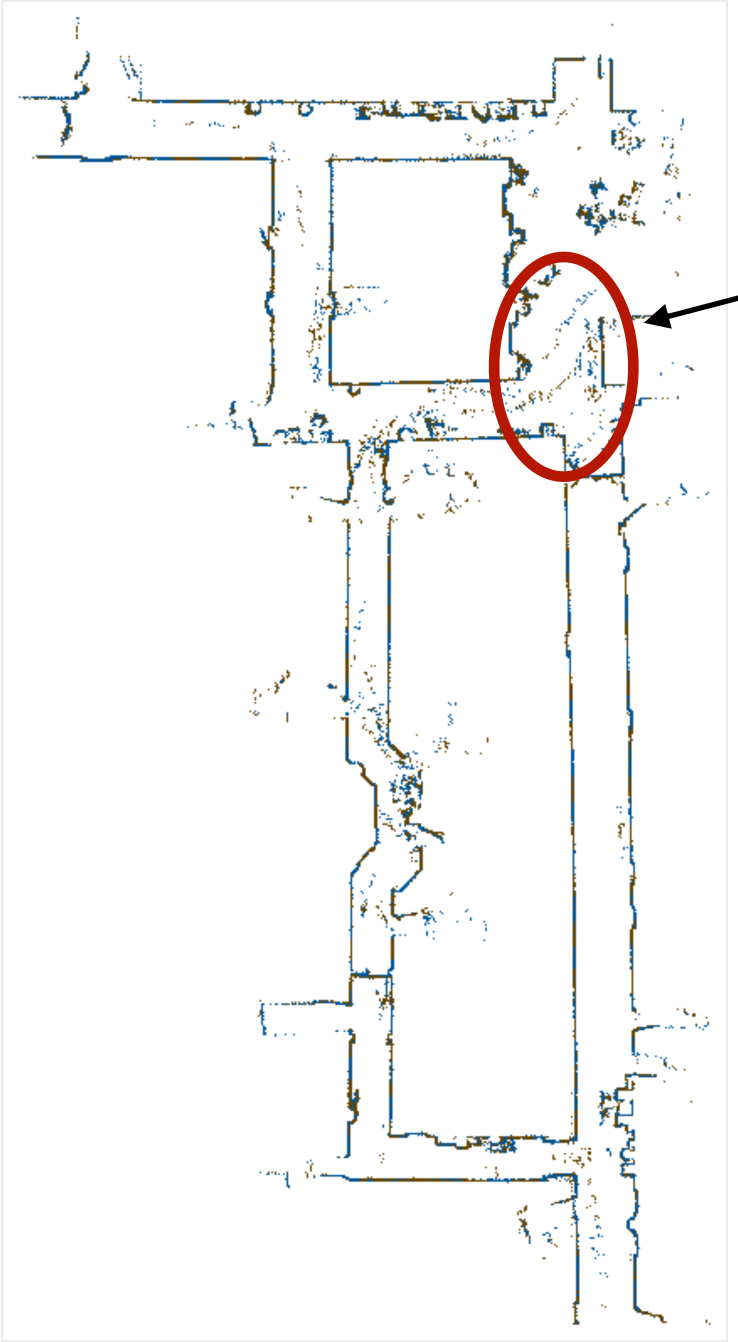
CAD map



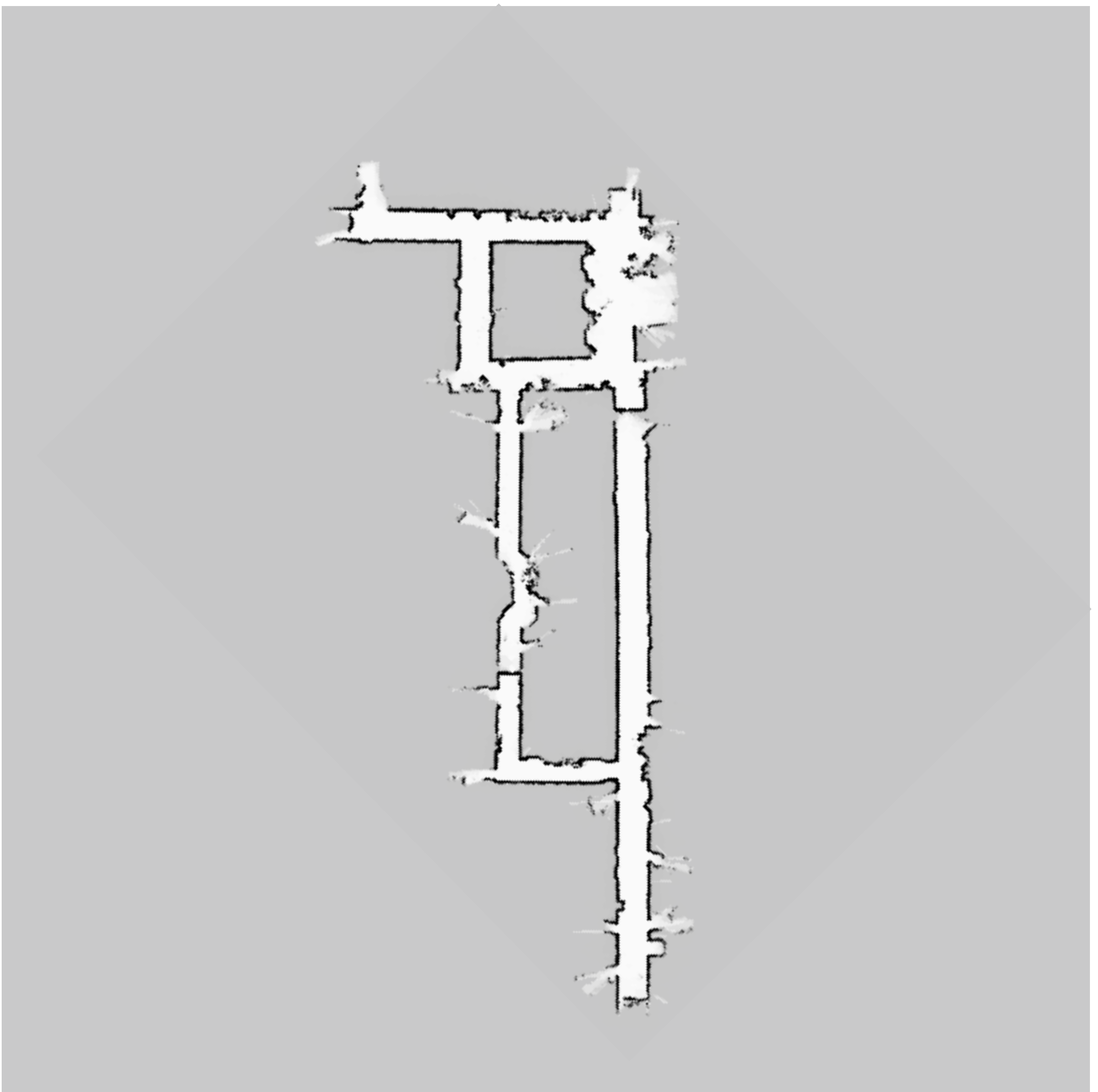
OGM

Tech Museum, San Jose

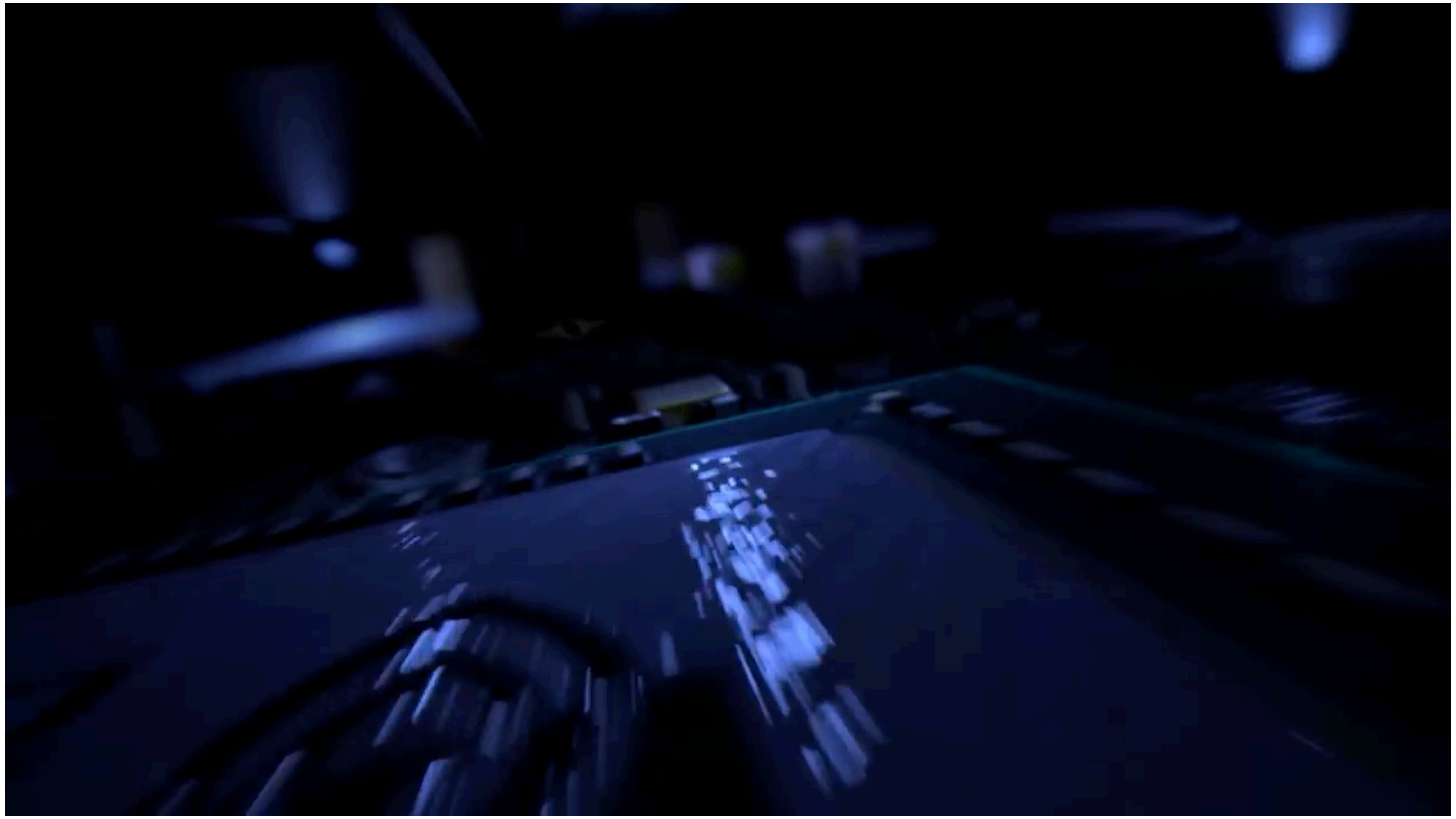
De-noises sensor measurements



False returns from people in environment



Still the gold standard?



Occupancy grid mapping

- Proposed in the mid 1980’s by Moravec and Elfes
- Initially developed (and still works well!) for noisy sonar sensors
- Alternatively known as “mapping with known poses”

High Resolution Maps from Wide Angle Sonar

Hans P. Moravec Alberto Elfes
The Robotics Institute
Carnegie-Mellon University

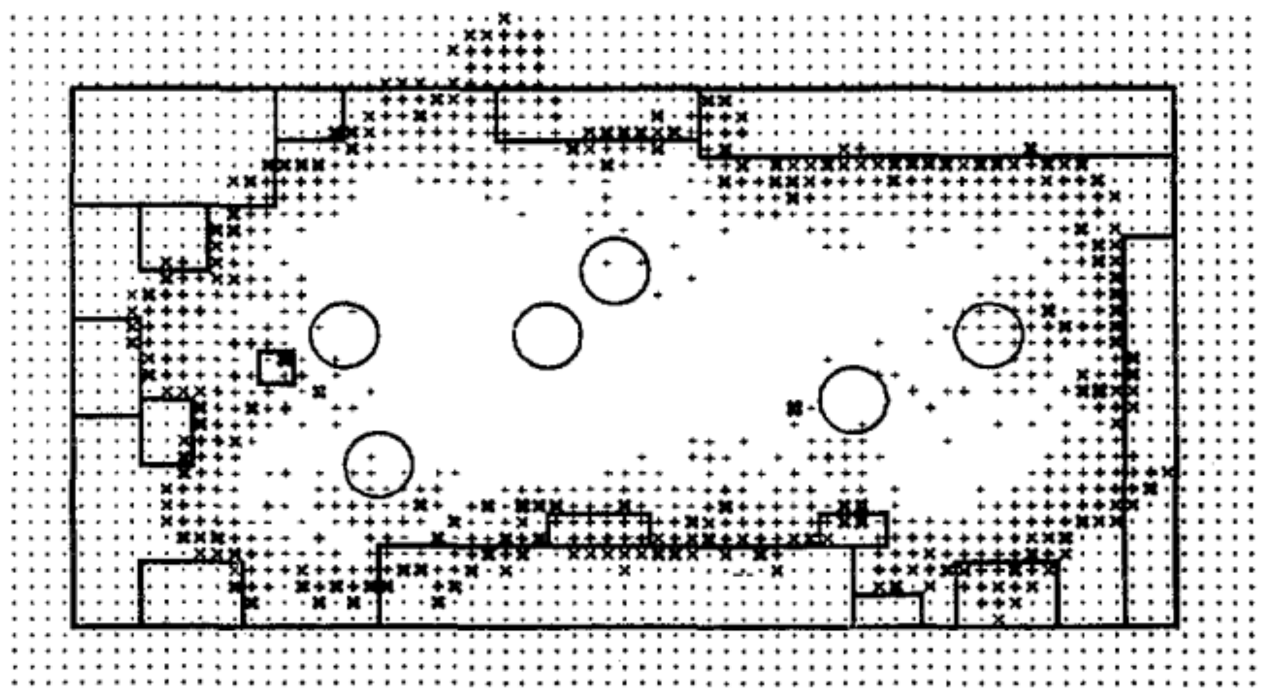


Figure 6: The Two-Dimensional Sonar Map After Thresholding.

Moravec, Hans, and Alberto Elfes. "High resolution maps from wide angle sonar." *Proceedings. 1985 IEEE international conference on robotics and automation*. Vol. 2. IEEE, 1985.

Occupancy grid mapping

- Represents the environment as a grid
  - eg: 25m $\times$ 25m area with 25cm resolution yields a 100×100 grid $= 10000$ grid cells

$\approx 10^{3010}$ maps!

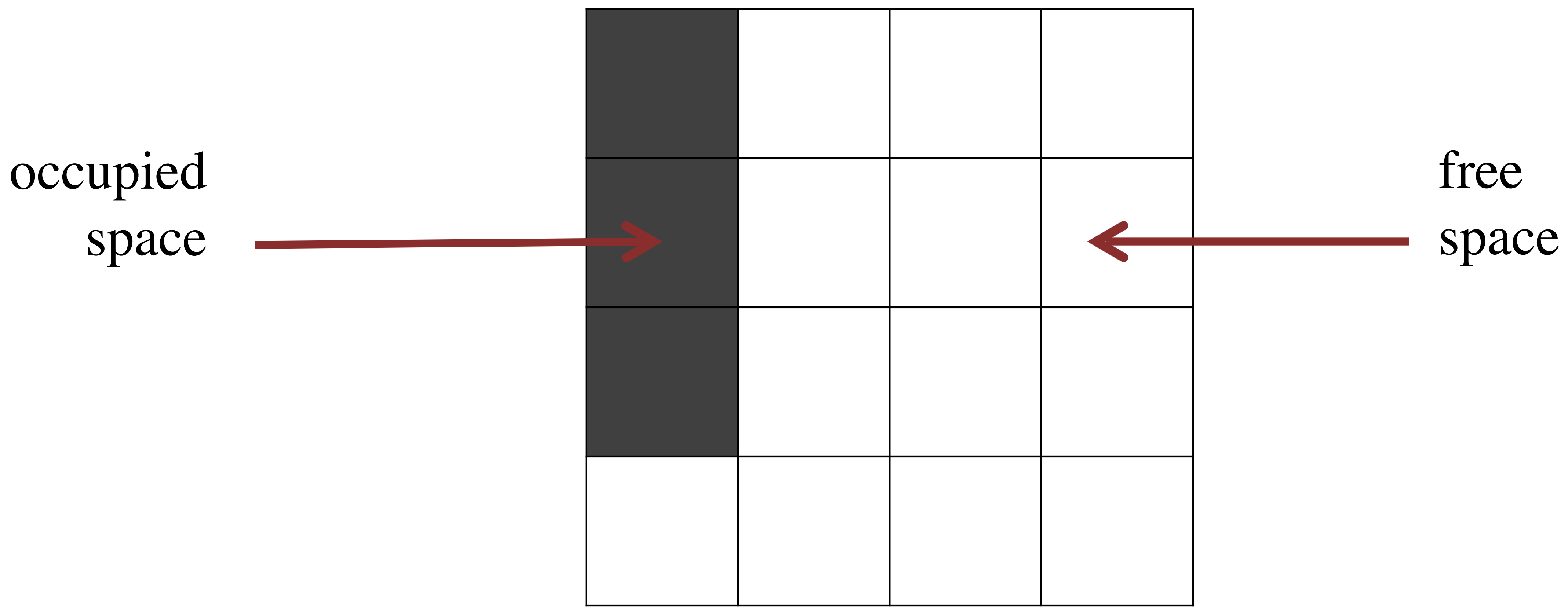
- Estimate the probability that a cell is occupied by an obstacle (binary)

$$m_i = \begin{cases} 1, & \text{occupied} \\ 0, & \text{free} \end{cases}$$

- **Map:** $m = \{m_i\}$ **Belief:** $\textbf{bel}_t(m) = p(m \,|\, \mathbf{z}_{1:t}, \mathbf{x}_{1:t})$
- A discrete Bayes estimation problem. How many possible maps exist for the example above?

Assumption I

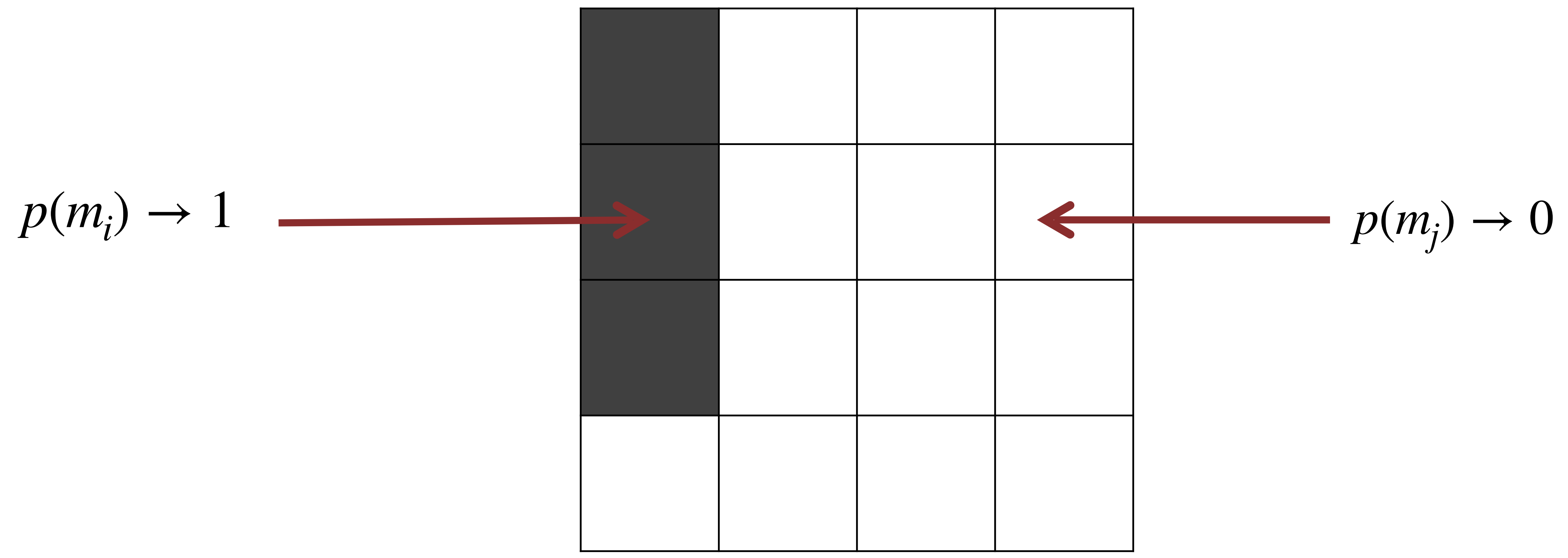
- The area that corresponds to a cell is either completely free or occupied



Courtesy: C. Stachniss

Representation

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



Courtesy: C. Stachniss

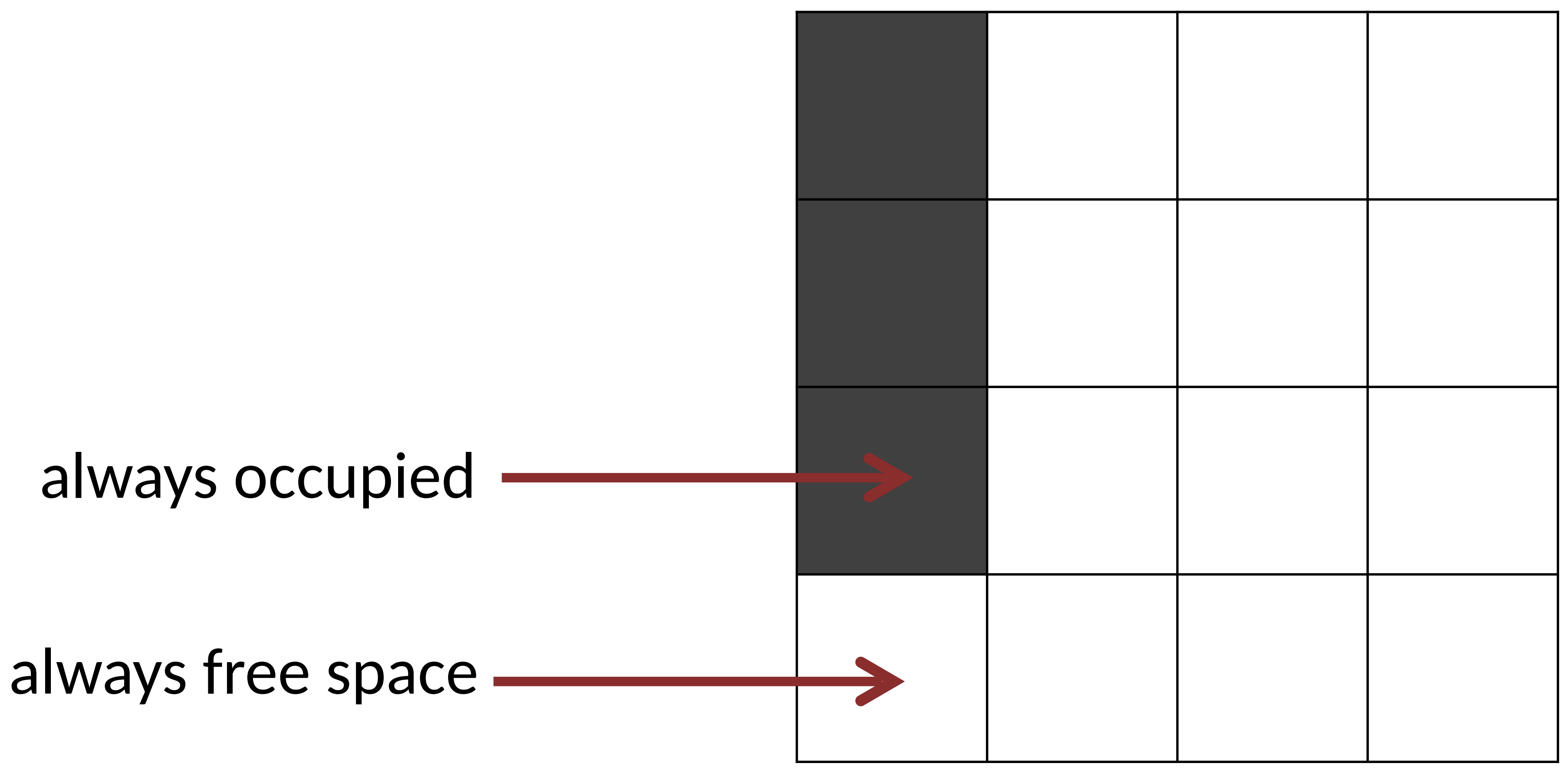
Occupancy probability

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

Cell is occupied:	$p(m_i) = 1$
Cell is unoccupied:	$p(m_i) = 0$
No knowledge:	$p(m_i) = 0.5$

Assumption II

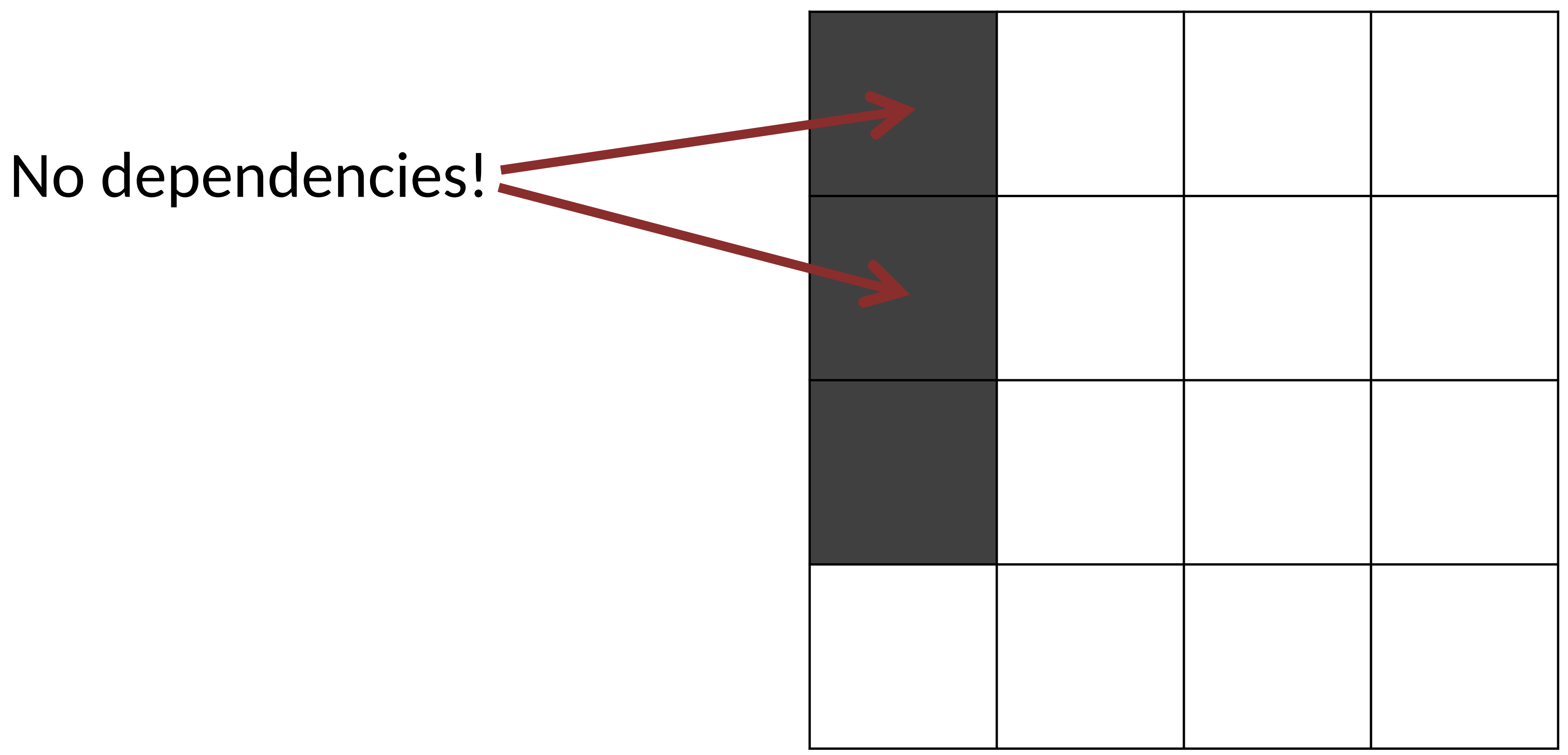
- The world is **static**!
    - Common assumption in many mapping systems.



Courtesy: C. Stachniss

Assumption III

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



Courtesy: C. Stachniss

Representation

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

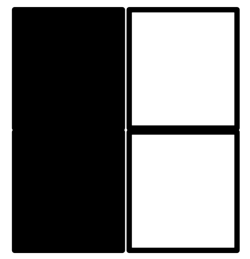
$$\underbrace{p(m)}_{\text{map}} = \prod_i \underbrace{p(m_i)}_{\text{cell}}$$

Courtesy: C. Stachniss

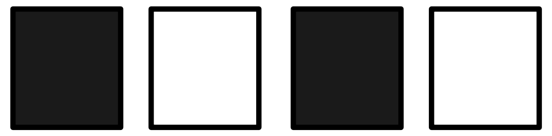
Representation

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

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



example map
(4-dim state)



4 individual cells

Courtesy: C. Stachniss

Estimating the map from data

- Given sensor data $\mathbf{z}_{1:t}$ and poses $\mathbf{x}_{1:t}$ of the sensor, estimate the map!

$$p(m \mid \mathbf{z}_{1:t}, \mathbf{x}_{1:t}) \quad = \quad \prod_i p(m_i \mid \mathbf{z}_{1:t}, \mathbf{x}_{1:t})$$

binary random variable

…this is a binary Bayes filter! (for a static state)

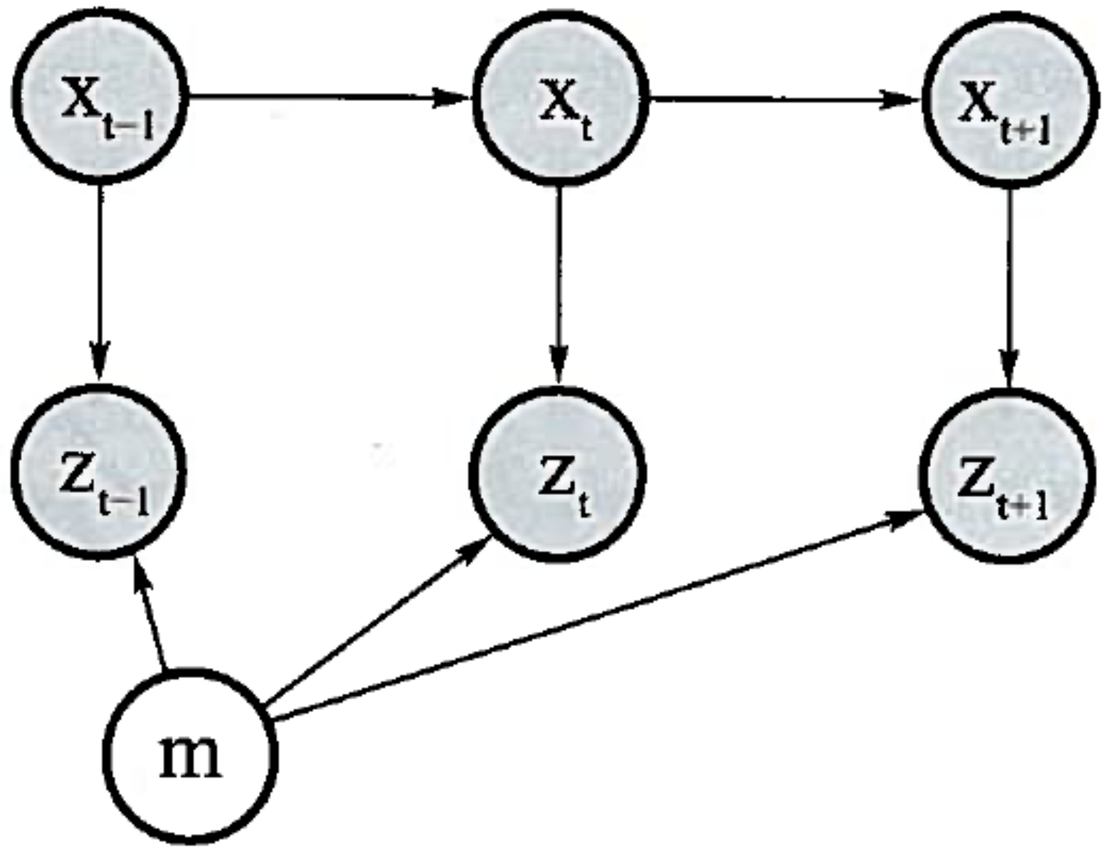
Courtesy: C. Stachniss

Occupancy grid maps

- Key assumptions (for tractability)
  - Robot poses are **known**!
  - Occupancy of individual cells (m_i) are **independent**

$$bel_t(m) = p(m \,|\, \mathbf{z}_{1:t}, \mathbf{x}_{1:t}) = \prod_i p(m_i \,|\, \mathbf{z}_{1:t}, \mathbf{x}_{1:t})$$

- OGM graphical model
  - $\mathbf{z}$ and $\mathbf{x}$ are known (shaded)
  - Goal is to infer map m
  - Controls $\mathbf{u}$ play no role in the belief (since we know $\mathbf{x}$!)



Static state binary Bayes filter

$$p(m_i \mid \mathbf{z}_{1:t}, \mathbf{x}_{1:t}) \overset{\text{Bayes rule}}{=} \frac{p(\mathbf{z}_t \mid m_i, \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})\, p(m_i \mid \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})}{p(\mathbf{z}_t \mid \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})}$$

Static state binary Bayes filter

$$p(m_i \mid \mathbf{z}_{1:t}, \mathbf{x}_{1:t}) \overset{\text{Bayes rule}}{=} \frac{p(\mathbf{z}_t \mid m_i, \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})\, p(m_i \mid \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})}{p(\mathbf{z}_t \mid \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})}$$

$$\overset{\text{Markov}}{=} \frac{p(\mathbf{z}_t \mid m_i, \mathbf{x}_t)\, p(m_i \mid \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-1})}{p(\mathbf{z}_t \mid \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})}$$

Static state binary Bayes filter

$$p(m_i \mid \mathbf{z}_{1:t}, \mathbf{x}_{1:t}) \overset{\text{Bayes rule}}{=} \frac{p(\mathbf{z}_t \mid m_i, \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})\, p(m_i \mid \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})}{p(\mathbf{z}_t \mid \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})}$$

$$\overset{\text{Markov}}{=} \underbrace{\frac{p(\mathbf{z}_t \mid m_i, \mathbf{x}_t)\, p(m_i \mid \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-1})}{p(\mathbf{z}_t \mid \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})}}_{\textbf{Forward model}}$$

- When measurement space is more complex than state space, an inverse sensor model may be preferred!
  - e.g., determining if a door is open or closed from a camera image

$$p(\mathbf{z}_t \mid m_i, \mathbf{x}_t) \overset{\text{Bayes rule}}{=} \underbrace{\frac{p(m_i \mid \mathbf{z}_t, \mathbf{x}_t)\, p(\mathbf{z}_t \mid \mathbf{x}_t)}{p(m_i \mid \mathbf{x}_t)}}_{\textbf{Inverse model}}$$

Static state binary Bayes filter

$$p(m_i \mid \mathbf{z}_{1:t}, \mathbf{x}_{1:t}) \overset{\text{Bayes rule}}{=} \frac{p(\mathbf{z}_t \mid m_i, \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})\, p(m_i \mid \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})}{p(\mathbf{z}_t \mid \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})}$$

$$\overset{\text{Markov}}{=} \frac{p(\mathbf{z}_t \mid m_i, \mathbf{x}_t)\, p(m_i \mid \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-1})}{p(\mathbf{z}_t \mid \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})}$$

$$p(\mathbf{z}_t \mid m_i, \mathbf{x}_t) \overset{\text{Bayes rule}}{=} \frac{p(m_i \mid \mathbf{z}_t, \mathbf{x}_t)\, p(\mathbf{z}_t \mid \mathbf{x}_t)}{p(m_i \mid \mathbf{x}_t)}$$

Static state binary Bayes filter

$$p(m_i \mid \mathbf{z}_{1:t}, \mathbf{x}_{1:t}) \overset{\text{Bayes rule}}{=} \frac{p(\mathbf{z}_t \mid m_i, \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})\, p(m_i \mid \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})}{p(\mathbf{z}_t \mid \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})}$$

$$\overset{\text{Markov}}{=} \frac{p(\mathbf{z}_t \mid m_i, \mathbf{x}_t)\, p(m_i \mid \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-1})}{p(\mathbf{z}_t \mid \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})}$$

$$p(\mathbf{z}_t \mid m_i, \mathbf{x}_t) \overset{\text{Bayes rule}}{=} \frac{p(m_i \mid \mathbf{z}_t, \mathbf{x}_t)\, p(\mathbf{z}_t \mid \mathbf{x}_t)}{p(m_i \mid \mathbf{x}_t)}$$

$$p(m_i \mid \mathbf{z}_{1:t}, \mathbf{x}_{1:t}) \overset{\text{indep.}}{=} \frac{p(m_i \mid \mathbf{z}_t, \mathbf{x}_t)\, p(\mathbf{z}_t \mid \mathbf{x}_t)\, p(m_i \mid \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-1})}{p(m_i)\, p(\mathbf{z}_t \mid \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})}$$

- Made for sheer convenience (the pose of the robot tells us that the cell must be free!)

Static state binary Bayes filter

$$p(m_i \mid \mathbf{z}_{1:t}, \mathbf{x}_{1:t}) \overset{\text{indep.}}{=} \frac{p(m_i \mid \mathbf{z}_t, \mathbf{x}_t)\ p(\mathbf{z}_t \mid \mathbf{x}_t)\ p(m_i \mid \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-1})}{p(m_i)\ p(\mathbf{z}_t \mid \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})}$$

(Do exactly the same for the opposite event)

$$p(\neg m_i \mid \mathbf{z}_{1:t}, \mathbf{x}_{1:t}) \overset{\text{the same}}{=} \frac{p(\neg m_i \mid \mathbf{z}_t, \mathbf{x}_t)\ p(\mathbf{z}_t \mid \mathbf{x}_t)\ p(\neg m_i \mid \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-1})}{p(\neg m_i)\ p(\mathbf{z}_t \mid \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})}$$

Static state binary Bayes filter

- Taking the ratio we get:

$$\frac{p(m_i \mid \mathbf{z}_{1:t}, \mathbf{x}_{1:t})}{p(\neg m_i \mid \mathbf{z}_{1:t}, \mathbf{x}_{1:t})} = \frac{\dfrac{p(m_i|\mathbf{z}_t,\mathbf{x}_t)\ \cancel{p(\mathbf{z}_t|\mathbf{x}_t)}\ p(m_i|\mathbf{z}_{1:t-1},\mathbf{x}_{1:t-1})}{p(m_i)\ \cancel{p(\mathbf{z}_t|\mathbf{z}_{1:t-1},\mathbf{x}_{1:t})}}}{\dfrac{p(\neg m_i|\mathbf{z}_t,\mathbf{x}_t)\ \cancel{p(\mathbf{z}_t|\mathbf{x}_t)}\ p(\neg m_i|\mathbf{z}_{1:t-1},\mathbf{x}_{1:t-1})}{p(\neg m_i)\ \cancel{p(\mathbf{z}_t|\mathbf{z}_{1:t-1},\mathbf{x}_{1:t})}}}$$

- This conveniently eliminates difficult to come by quantities!

Static state binary Bayes filter

- Taking the ratio we get:

$$\frac{p(m_i \mid \mathbf{z}_{1:t}, \mathbf{x}_{1:t})}{p(\neg m_i \mid \mathbf{z}_{1:t}, \mathbf{x}_{1:t})} = \frac{\dfrac{p(m_i|\mathbf{z}_t,\mathbf{x}_t)\ \cancel{p(\mathbf{z}_t|\mathbf{x}_t)}\ p(m_i|\mathbf{z}_{1:t-1},\mathbf{x}_{1:t-1})}{p(m_i)\ \cancel{p(\mathbf{z}_t|\mathbf{z}_{1:t-1},\mathbf{x}_{1:t})}}}{\dfrac{p(\neg m_i|\mathbf{z}_t,\mathbf{x}_t)\ \cancel{p(\mathbf{z}_t|\mathbf{x}_t)}\ p(\neg m_i|\mathbf{z}_{1:t-1},\mathbf{x}_{1:t-1})}{p(\neg m_i)\ \cancel{p(\mathbf{z}_t|\mathbf{z}_{1:t-1},\mathbf{x}_{1:t})}}}$$

$$= \frac{p(m_i \mid \mathbf{z}_t, \mathbf{x}_t)\ p(m_i \mid \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-1})\ p(\neg m_i)}{p(\neg m_i \mid \mathbf{z}_t, \mathbf{x}_t)\ p(\neg m_i \mid \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-1})\ p(m_i)}$$

$$= \frac{p(m_i \mid \mathbf{z}_t, \mathbf{x}_t)}{1 - p(m_i \mid \mathbf{z}_t, \mathbf{x}_t)}\ \frac{p(m_i \mid \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-1})}{1 - p(m_i \mid \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-1})}\ \frac{1 - p(m_i)}{p(m_i)}$$

Static state binary Bayes filter

- Taking the ratio we get:

$$\frac{p(m_i \mid \mathbf{z}_{1:t}, \mathbf{x}_{1:t})}{p(\neg m_i \mid \mathbf{z}_{1:t}, \mathbf{x}_{1:t})} = \frac{\dfrac{p(m_i|\mathbf{z}_t,\mathbf{x}_t)\ \cancel{p(\mathbf{z}_t|\mathbf{x}_t)}\ p(m_i|\mathbf{z}_{1:t-1},\mathbf{x}_{1:t-1})}{p(m_i)\ \cancel{p(\mathbf{z}_t|\mathbf{z}_{1:t-1},\mathbf{x}_{1:t})}}}{\dfrac{p(\neg m_i|\mathbf{z}_t,\mathbf{x}_t)\ \cancel{p(\mathbf{z}_t|\mathbf{x}_t)}\ p(\neg m_i|\mathbf{z}_{1:t-1},\mathbf{x}_{1:t-1})}{p(\neg m_i)\ \cancel{p(\mathbf{z}_t|\mathbf{z}_{1:t-1},\mathbf{x}_{1:t})}}}$$

$$= \frac{p(m_i \mid \mathbf{z}_t, \mathbf{x}_t)\ p(m_i \mid \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-1})\ p(\neg m_i)}{p(\neg m_i \mid \mathbf{z}_t, \mathbf{x}_t)\ p(\neg m_i \mid \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-1})\ p(m_i)}$$

$$= \underbrace{\frac{p(m_i \mid \mathbf{z}_t, \mathbf{x}_t)}{1 - p(m_i \mid \mathbf{z}_t, \mathbf{x}_t)}}_{\text{uses } \mathbf{z}_t}\ \underbrace{\frac{p(m_i \mid \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-1})}{1 - p(m_i \mid \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-1})}}_{\text{recursive term}}\ \underbrace{\frac{1 - p(m_i)}{p(m_i)}}_{\text{prior}}$$

Static state binary Bayes filter

- We can convert the ratio into a probability:

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

Static state binary Bayes filter

- Using $p(x) = [1 + Y^{-1}]^{-1}$ directly gives us:

$$p(m_i \mid \mathbf{z}_{1:t}, \mathbf{x}_{1:t}) \quad = \quad \left[1 + \frac{1 - p(m_i \mid \mathbf{z}_t, \mathbf{x}_t)}{p(m_i \mid \mathbf{z}_t, \mathbf{x}_t)} \frac{1 - p(m_i \mid \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-1})}{p(m_i \mid \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-1})} \frac{p(m_i)}{1 - p(m_i)}\right]^{-1}$$

- For efficiency, we perform the calculations in **log-odds** notation

Carnegie Mellon
THE ROBOTICS INSTITUTE

Log-odds notation

- Computes the logarithm of the ratio of probabilities

$$\frac{p(m_i \mid \mathbf{z}_{1:t}, \mathbf{x}_{1:t})}{1 - p(m_i \mid \mathbf{z}_{1:t}, \mathbf{x}_{1:t})} = \underbrace{\frac{p(m_i \mid \mathbf{z}_t, \mathbf{x}_t)}{1 - p(m_i \mid \mathbf{z}_t, \mathbf{x}_t)}}_{\text{uses } \mathbf{z}_t} \underbrace{\frac{p(m_i \mid \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-1})}{1 - p(m_i \mid \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-1})}}_{\text{recursive term}} \underbrace{\frac{1 - p(m_i)}{p(m_i)}}_{\text{prior}}$$

$$l(m_i \mid \mathbf{z}_{1:t}, \mathbf{x}_{1:t}) = \log\left(\frac{p(m_i \mid \mathbf{z}_{1:t}, \mathbf{x}_{1:t})}{1 - p(m_i \mid \mathbf{z}_{1:t}, \mathbf{x}_{1:t})}\right)$$

Log-odds notation

- Log-odds ratio:

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

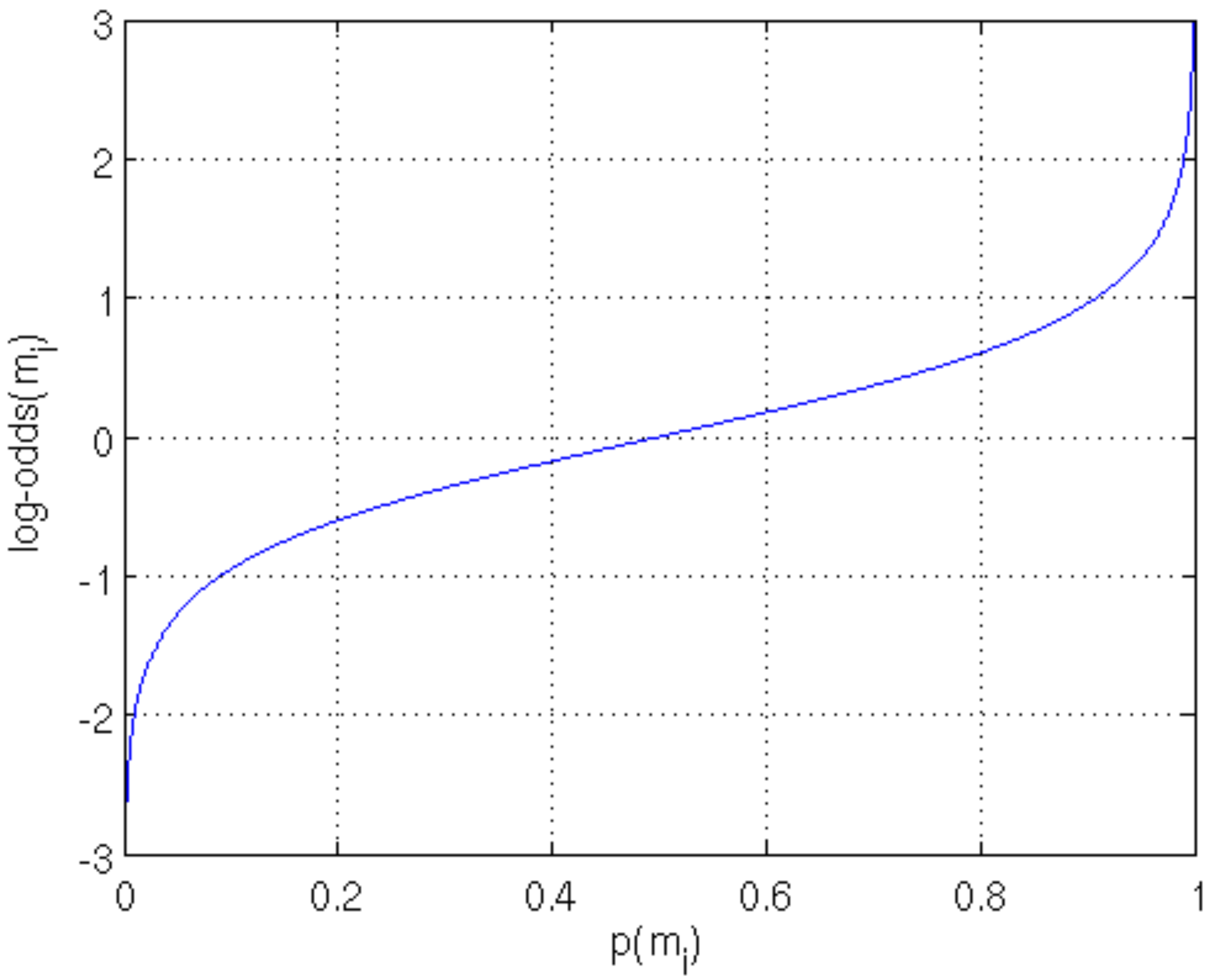
- So as to retrieve:

$$p(x) = 1 - \frac{1}{1 + e^{l(x)}}$$

Why Log-odds?

- Computationally elegant to updating beliefs!
- Updates are additive
- No truncation problems that arise for probabilities close to 0 or 1

$$l(x) \in [-\infty, \infty]$$



Occupancy mapping in log-odds

- Recall that:

$$\frac{p(m_i \mid \mathbf{z}_{1:t}, \mathbf{x}_{1:t})}{1 - p(m_i \mid \mathbf{z}_{1:t}, \mathbf{x}_{1:t})} = \underbrace{\frac{p(m_i \mid \mathbf{z}_t, \mathbf{x}_t)}{1 - p(m_i \mid \mathbf{z}_t, \mathbf{x}_t)}}_{\text{uses } \mathbf{z}_t} \underbrace{\frac{p(m_i \mid \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-1})}{1 - p(m_i \mid \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-1})}}_{\text{recursive term}} \underbrace{\frac{1 - p(m_i)}{p(m_i)}}_{\text{prior}}$$

- With log-odds, this turns into a sum:

$$l(m_i \mid \mathbf{z}_{1:t}, \mathbf{x}_{1:t}) = \underbrace{l(m_i \mid \mathbf{z}_t, \mathbf{x}_t)}_{\text{inverse sensor model}} + \underbrace{l(m_i \mid \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-1})}_{\text{recursive term}} - \underbrace{l(m_i)}_{\text{prior}}$$

- Or succinctly:

$$l_{t,\,i} = \textbf{InverseSensorModel}(m_i, \mathbf{x}_t, \mathbf{z}_t) + l_{t-1,\,i} - l_0$$

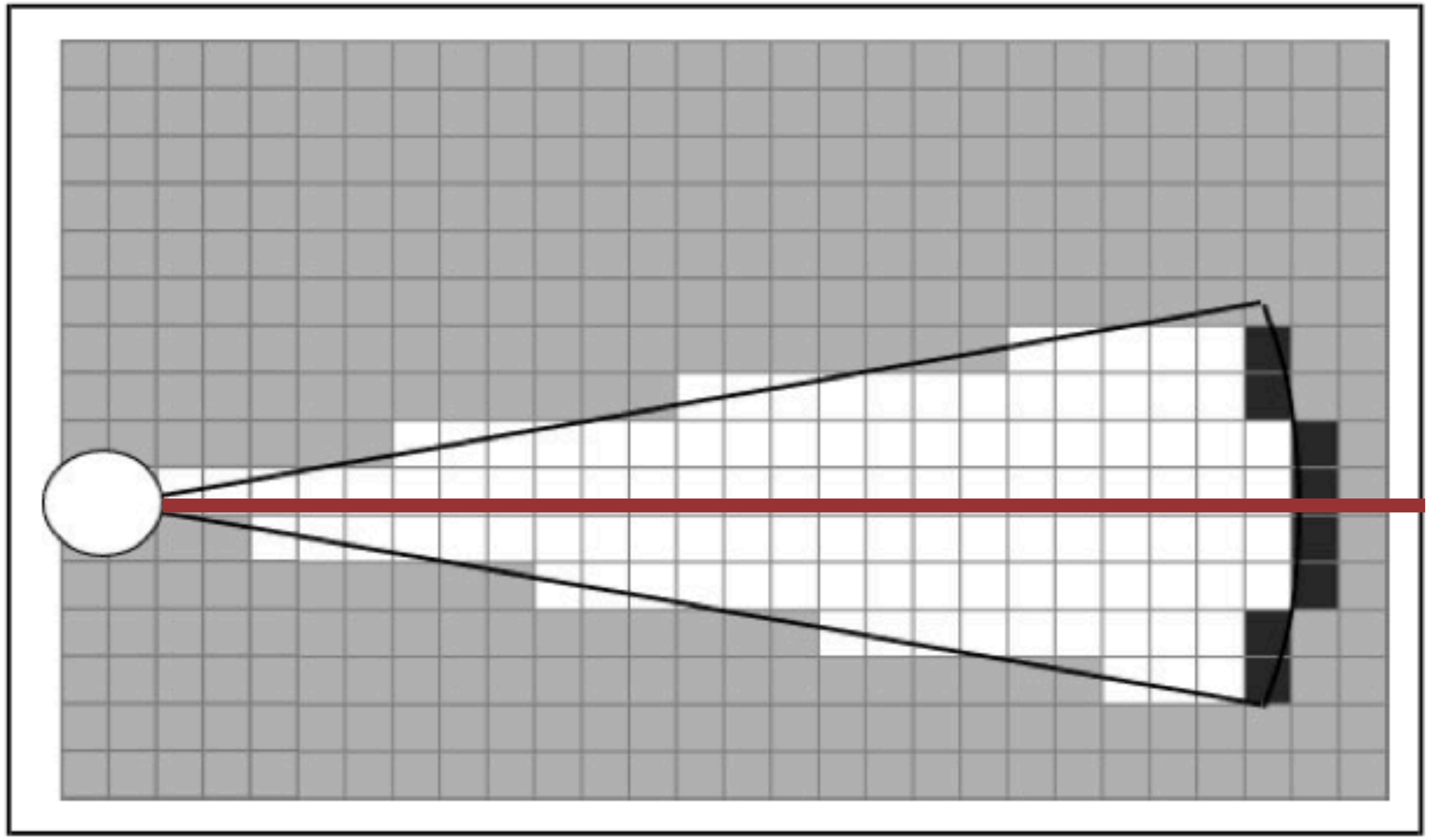
Occupancy mapping algorithm

occupancy_grid_mapping($\{l_{t-1,i}\}, \mathbf{x}_t, \mathbf{z}_t$)**:**

1: *for all cells* m_i *do*
2: *if* m_i *in perceptual field of* $\mathbf{z}_t$ *then*
3: $l_{t,\,i} \;=\;$ **InverseSensorModel**$(m_i, \mathbf{x}_t, \mathbf{z}_t) \;+\; l_{t-1,\,i} \;-\; l_0$
4: *else*
5: $l_{t,i} = l_{t-1,i}$
6: *endif*
7: *endfor*
8: *return* $\{l_{t,i}\}$

We only compute sums: **highly efficient!**

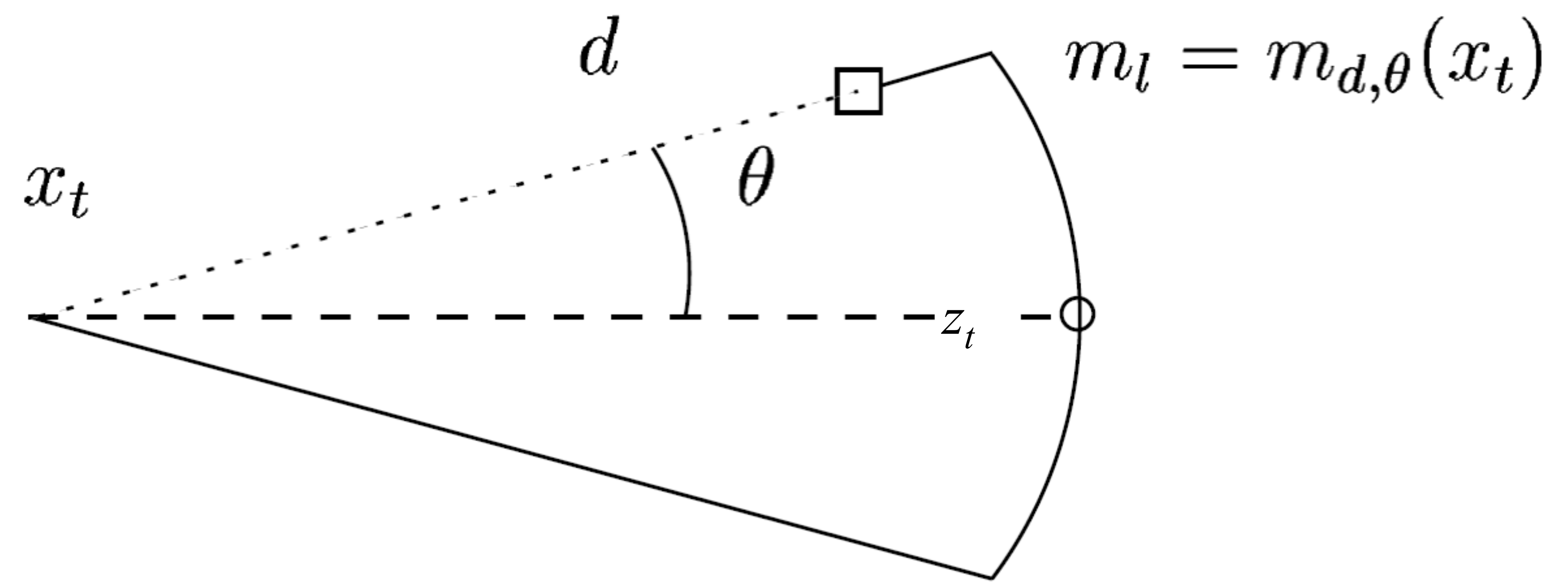
Inverse sensor model for SONAR

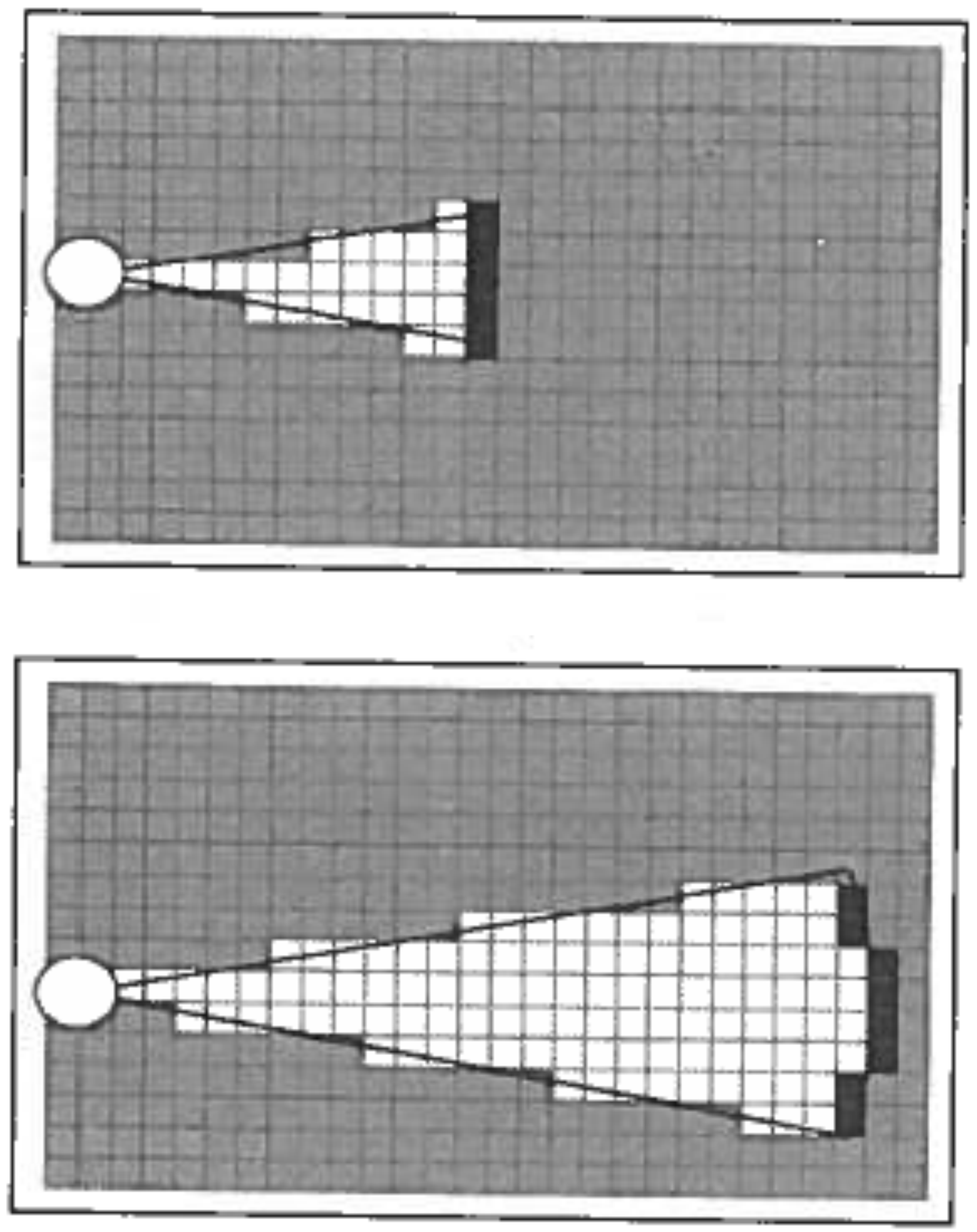


- Consider the cells **along the optical axis**

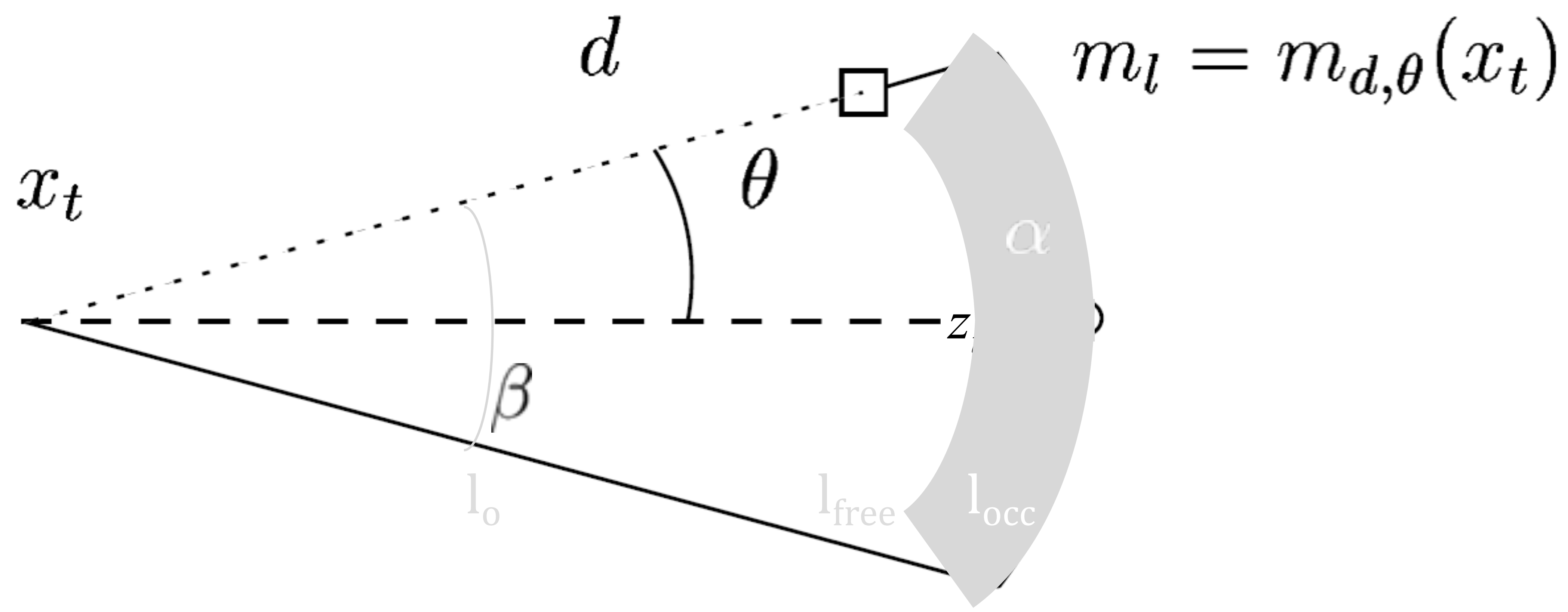
Courtesy: Thrun, Burgard, Fox

A (crude) inverse sensor model





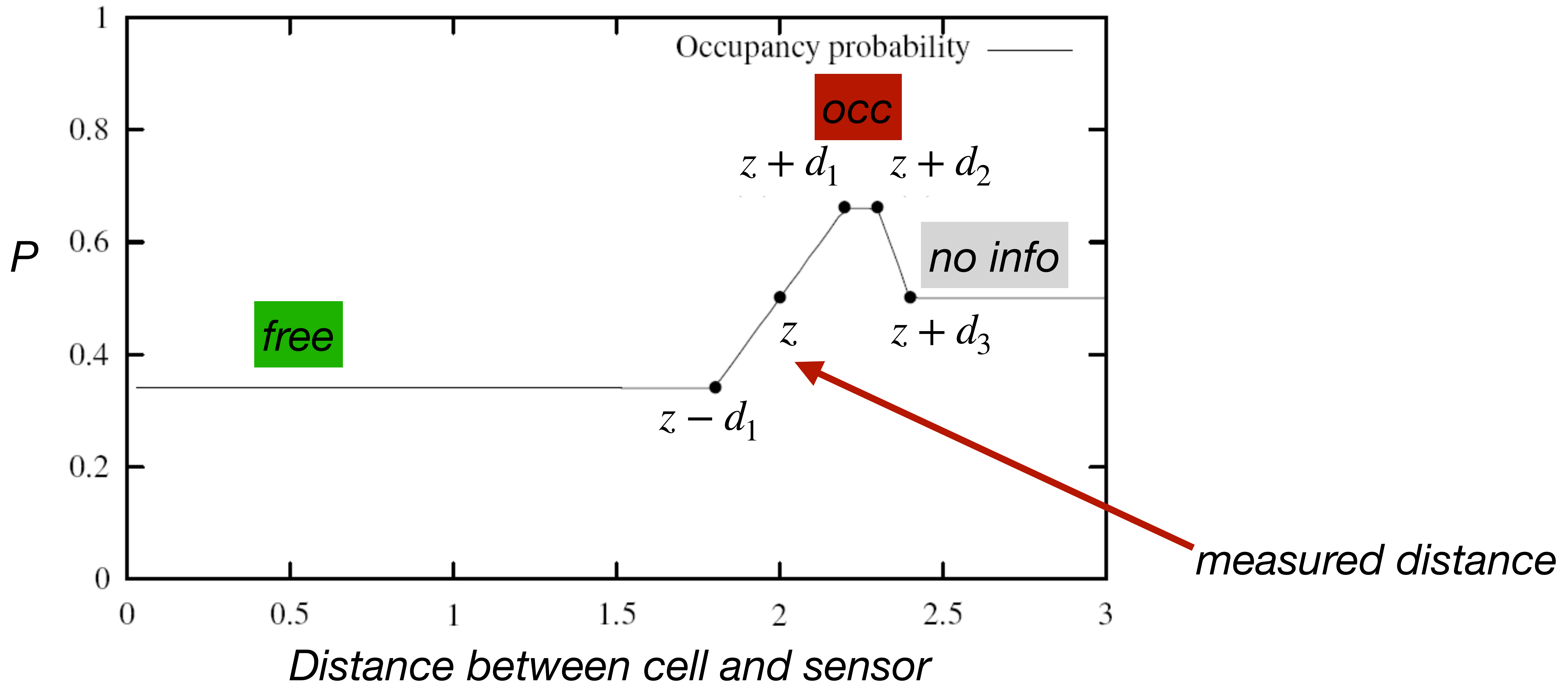
A (crude) inverse sensor model



Algorithm inverse_range_sensor_model($\mathbf{m}_i, x_t, z_t$)**:**

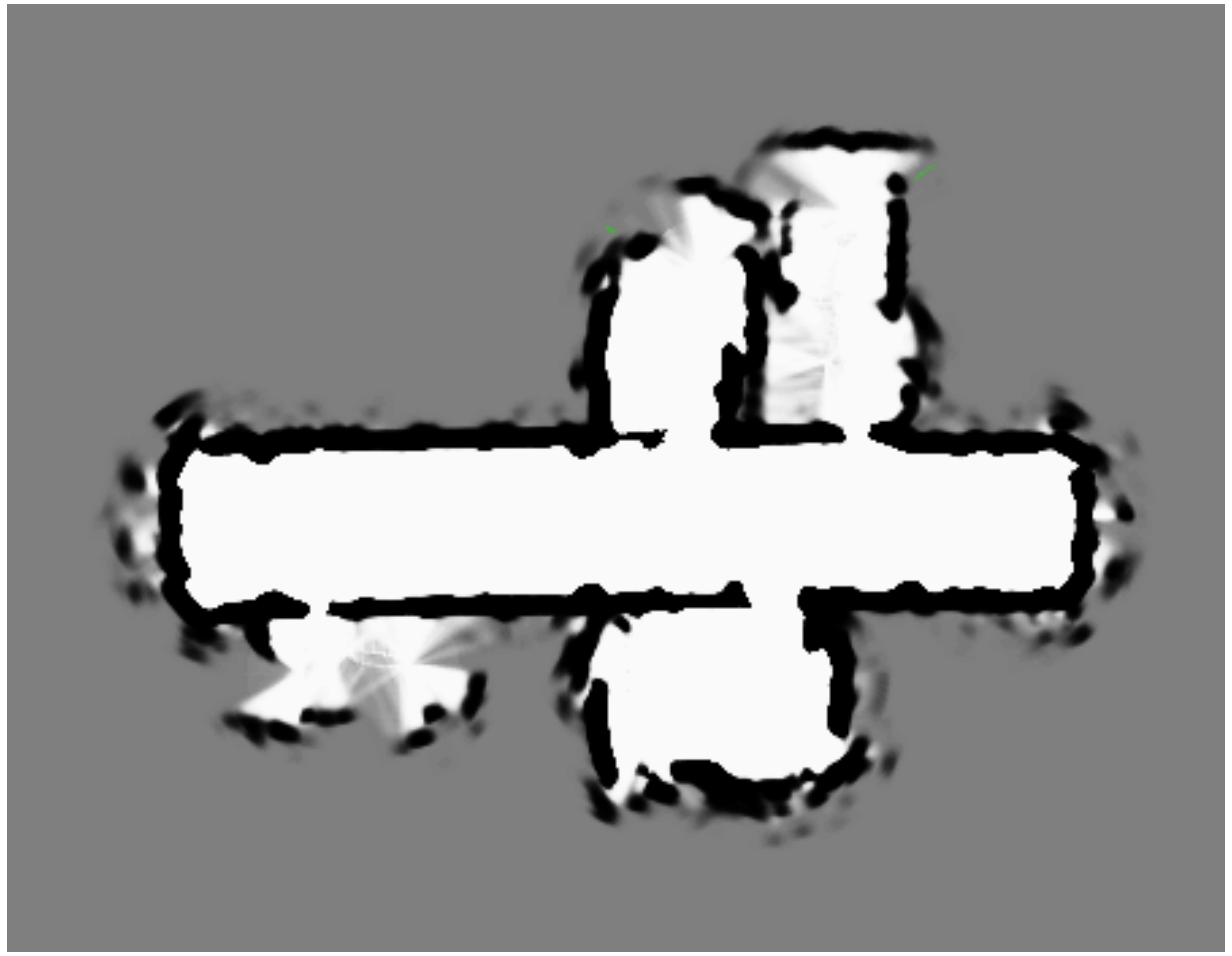
- *Let* x_i, y_i *be the center-of-mass of* $\mathbf{m}_i$
- $r = \sqrt{(x_i - x)^2 + (y_i - y)^2}$
- $\phi = \operatorname{atan2}(y_i - y, x_i - x) - \theta$
- $k = \operatorname{argmin}_j |\phi - \theta_{j,\text{sens}}|$
- *if* $r > \min(z_{\max}, z_t^k + \alpha/2)$ or $|\phi - \theta_{k,\text{sens}}| > \beta/2$ *then*
  - *return* l_0
- *if* $z_t^k < z_{\max}$ and $|r - z_t^k| < \alpha/2$
  - *return* l_{occ}
- *if* $r \leq z_t^k$
  - *return* l_{free}
- *endif*

Occupancy vs. distance



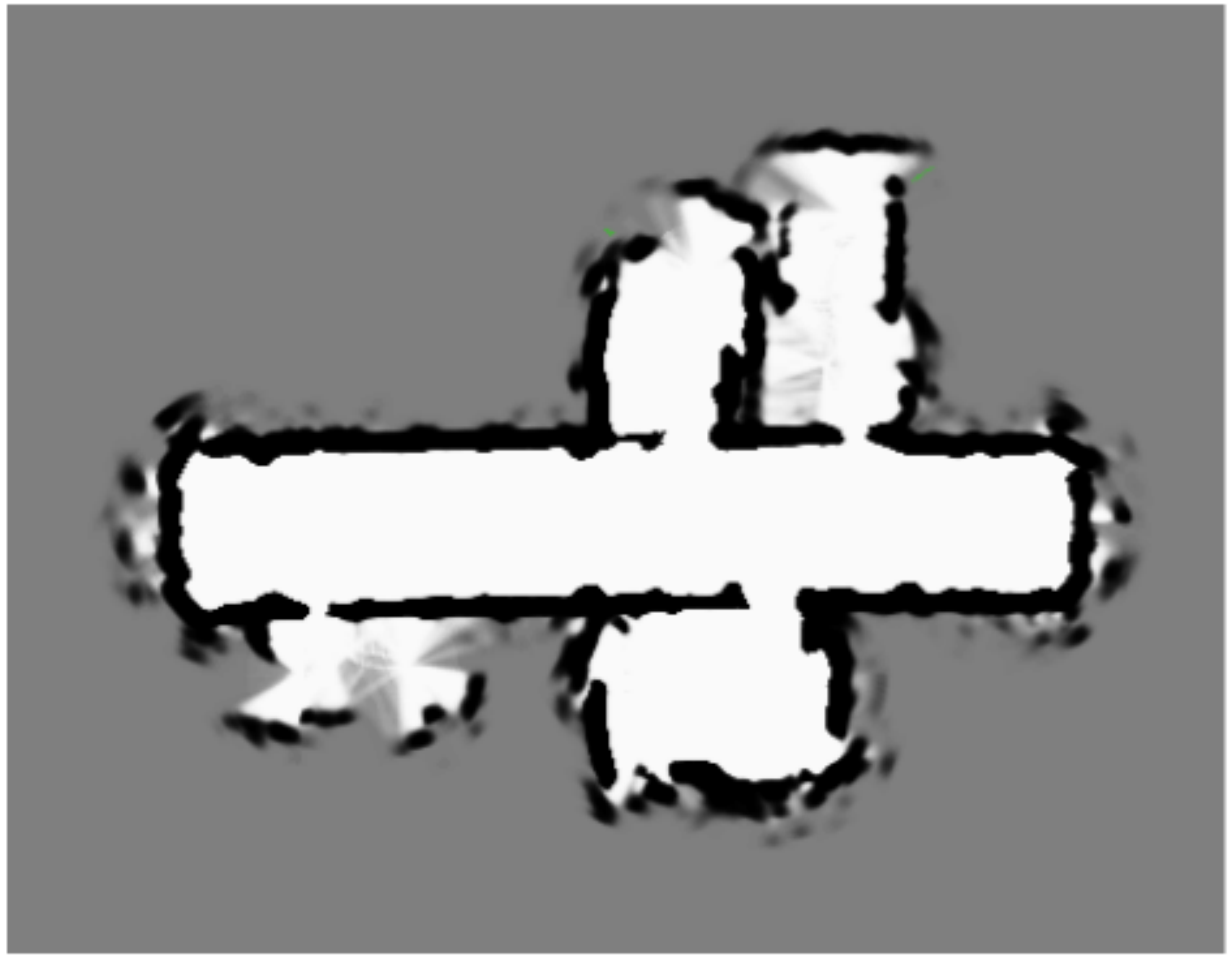
Courtesy: C. Stachniss

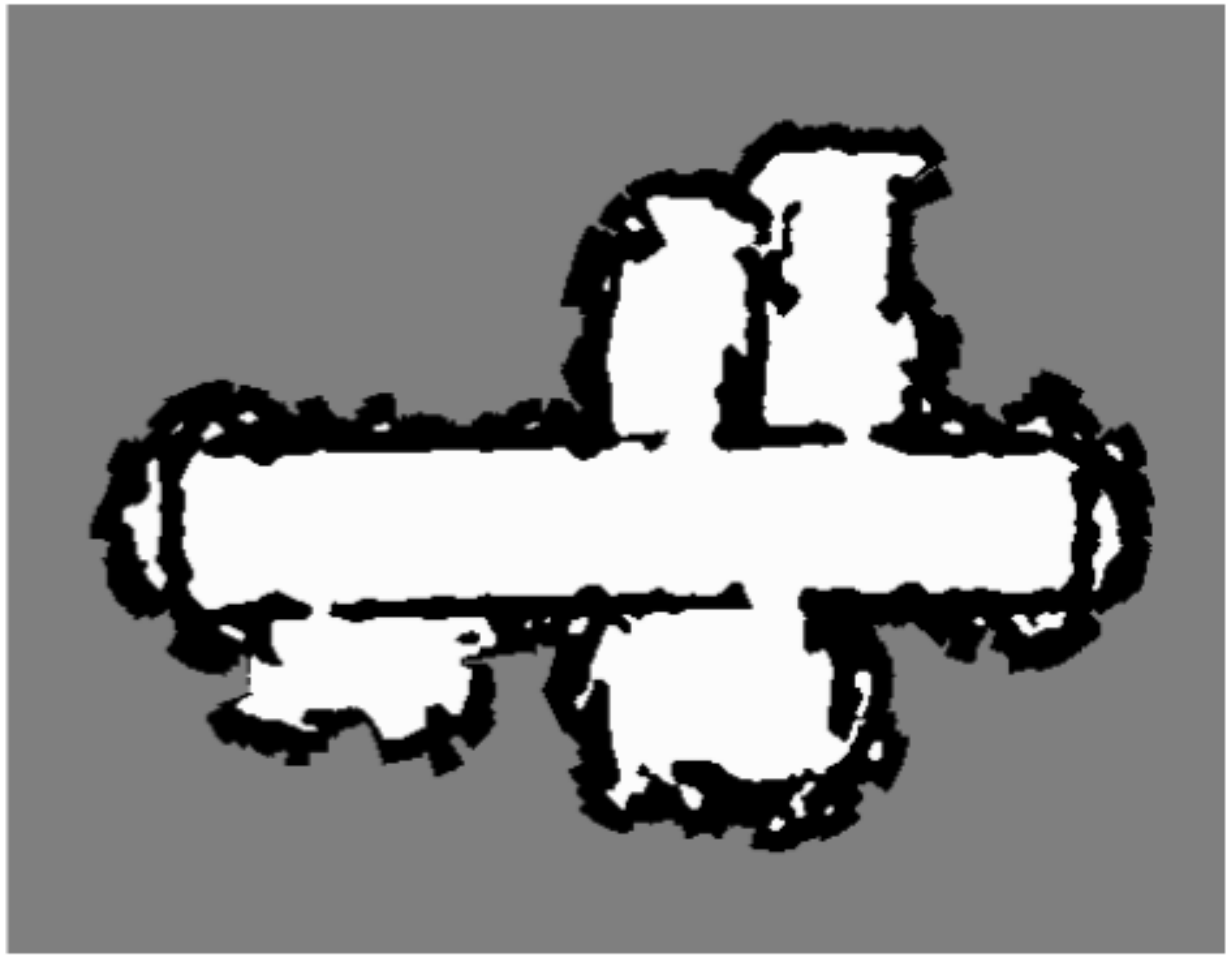
Results: Map from 24 SONAR range sensors



Courtesy: C. Stachniss

Occupancy vs. max. likelihood map

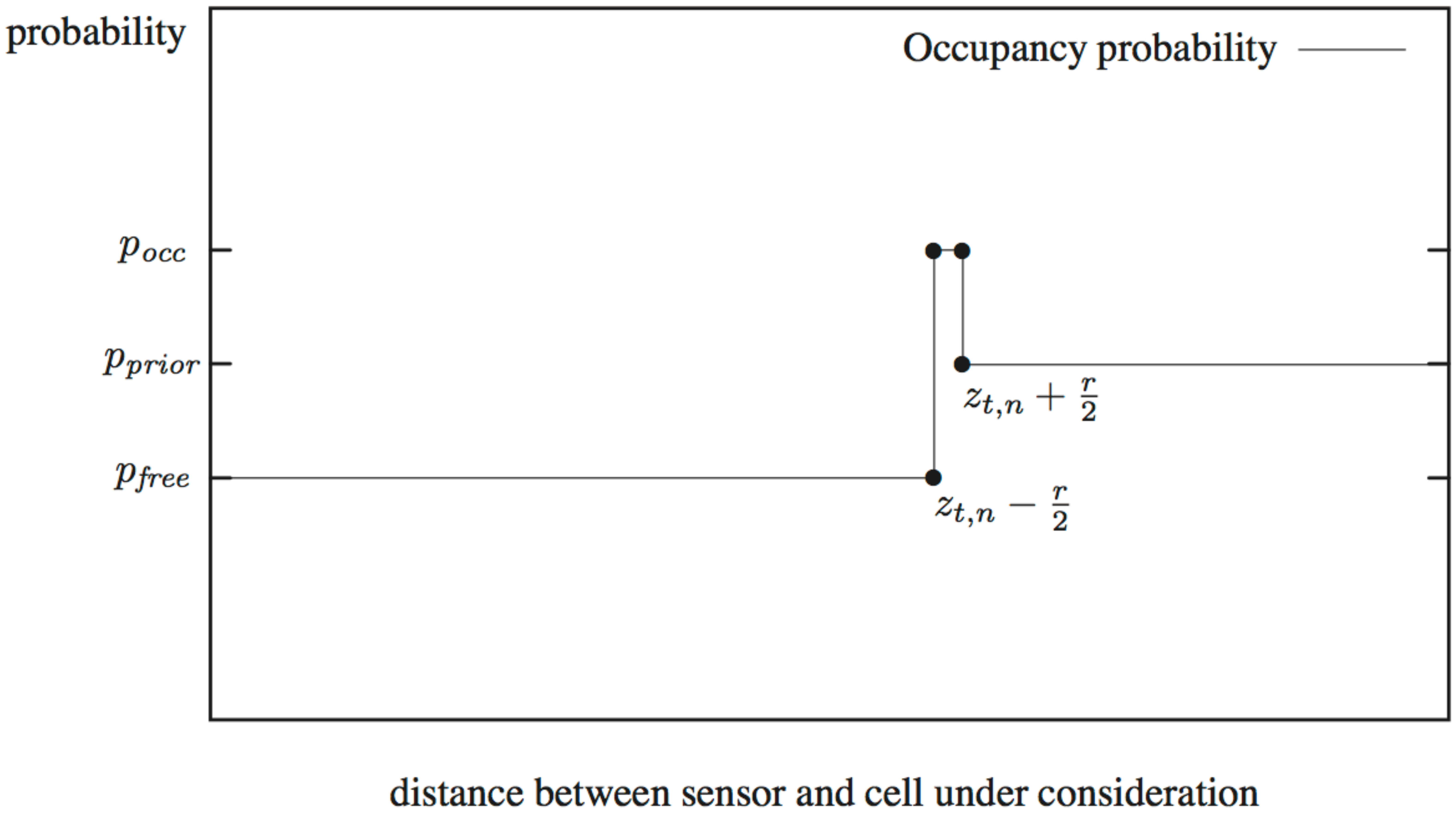




- The max. likelihood map is obtained by rounding the probability for each cell $\rightarrow 0$ or 1

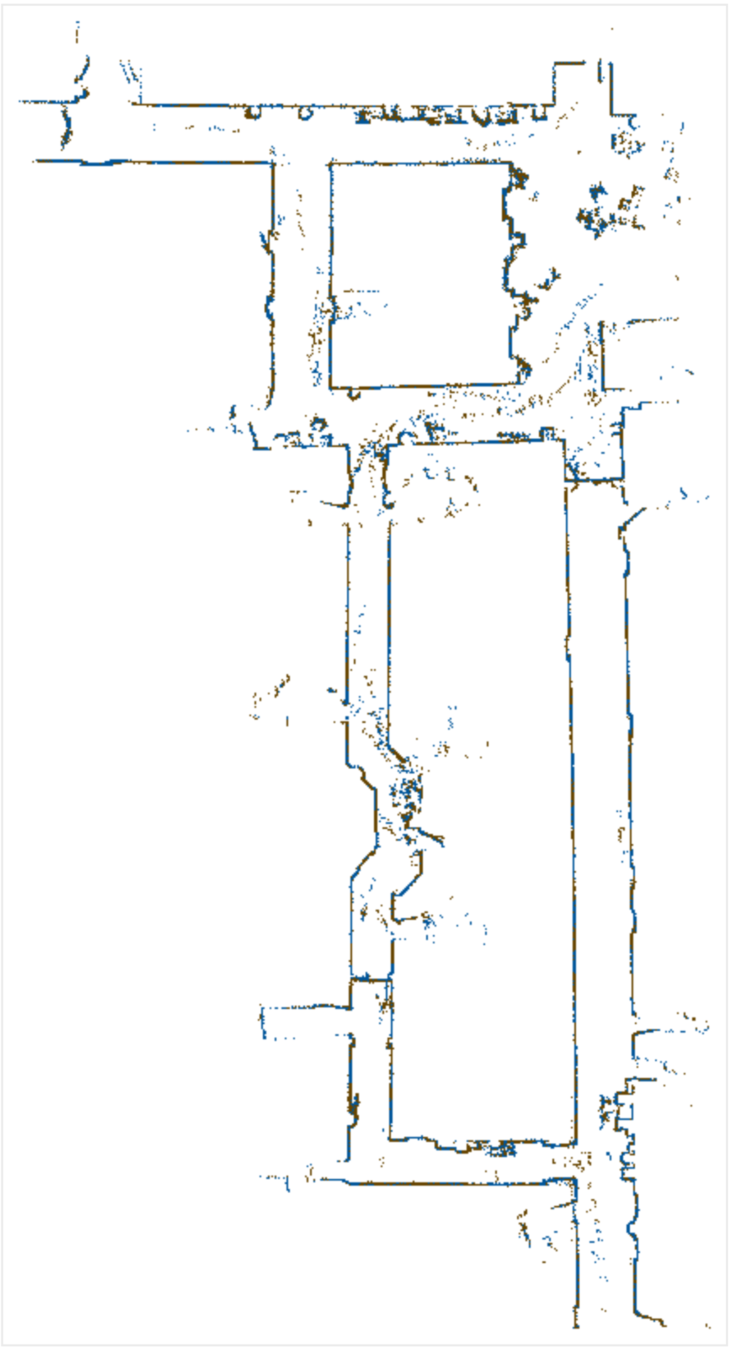
Courtesy: C. Stachniss

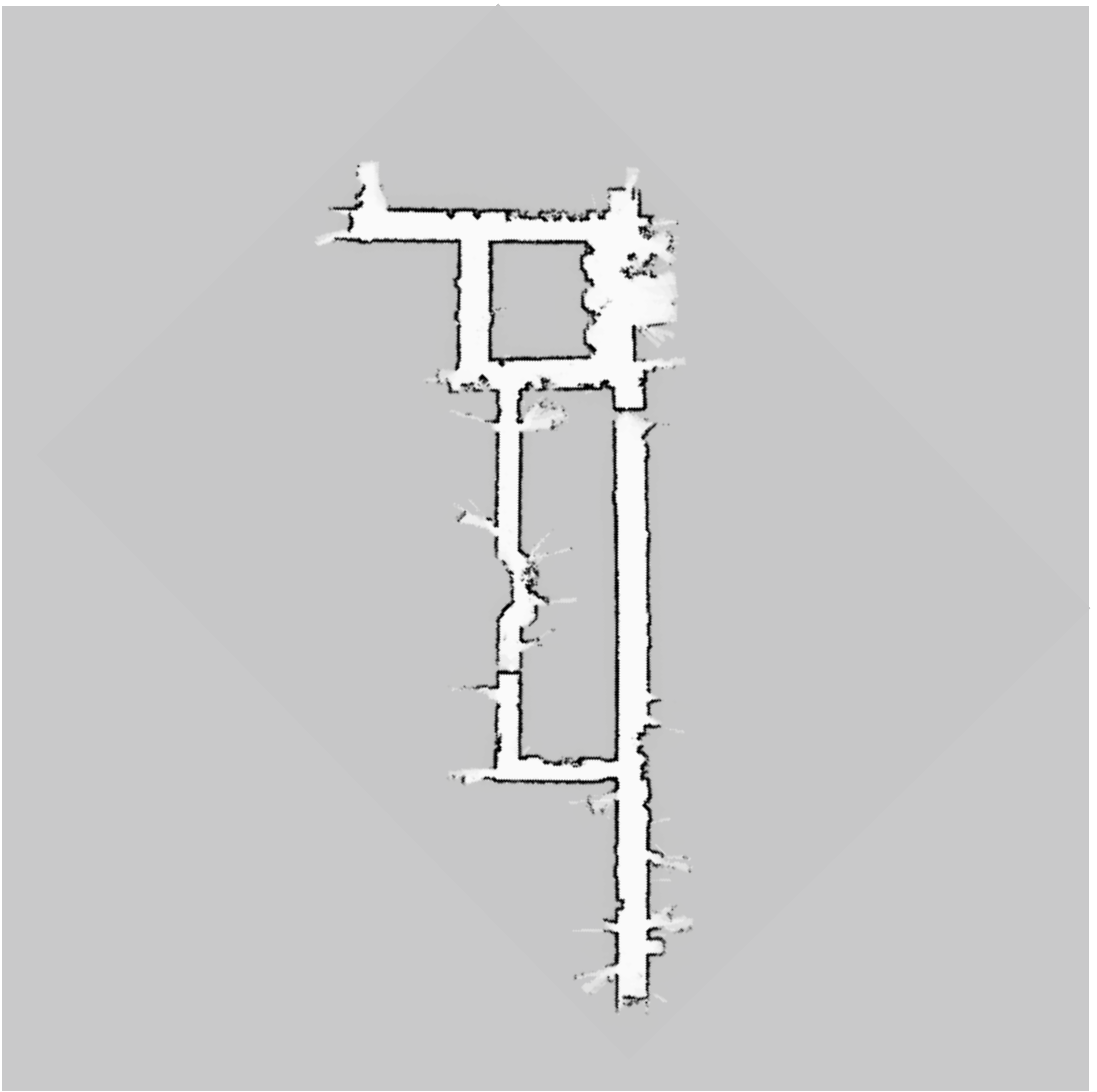
Inverse sensor model for laser range finder



Courtesy: C. Stachniss

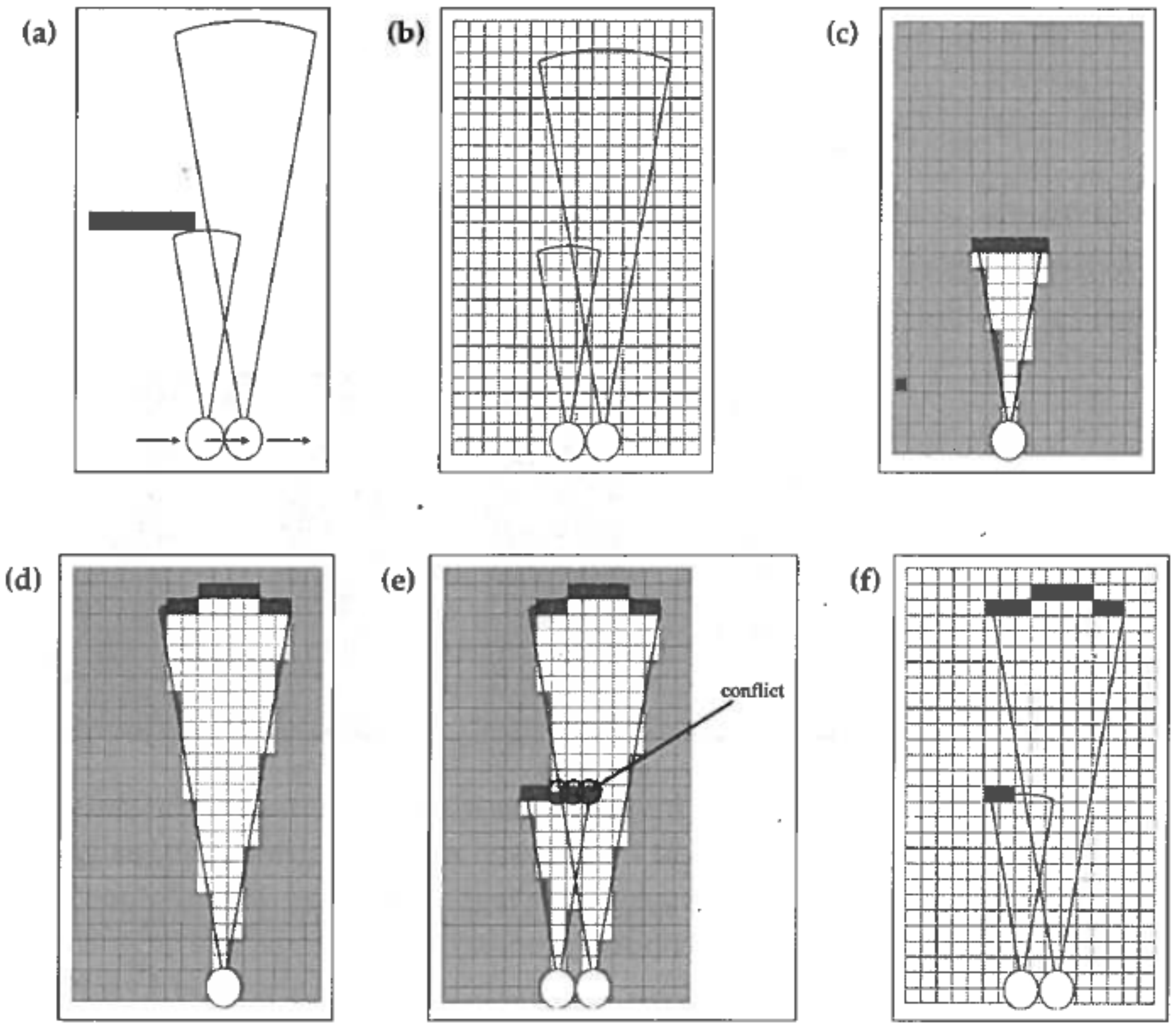
Occupancy grids: scans → maps





Courtesy: D. Hähnel

Drawback of independence assumption



Courtesy: Thrun

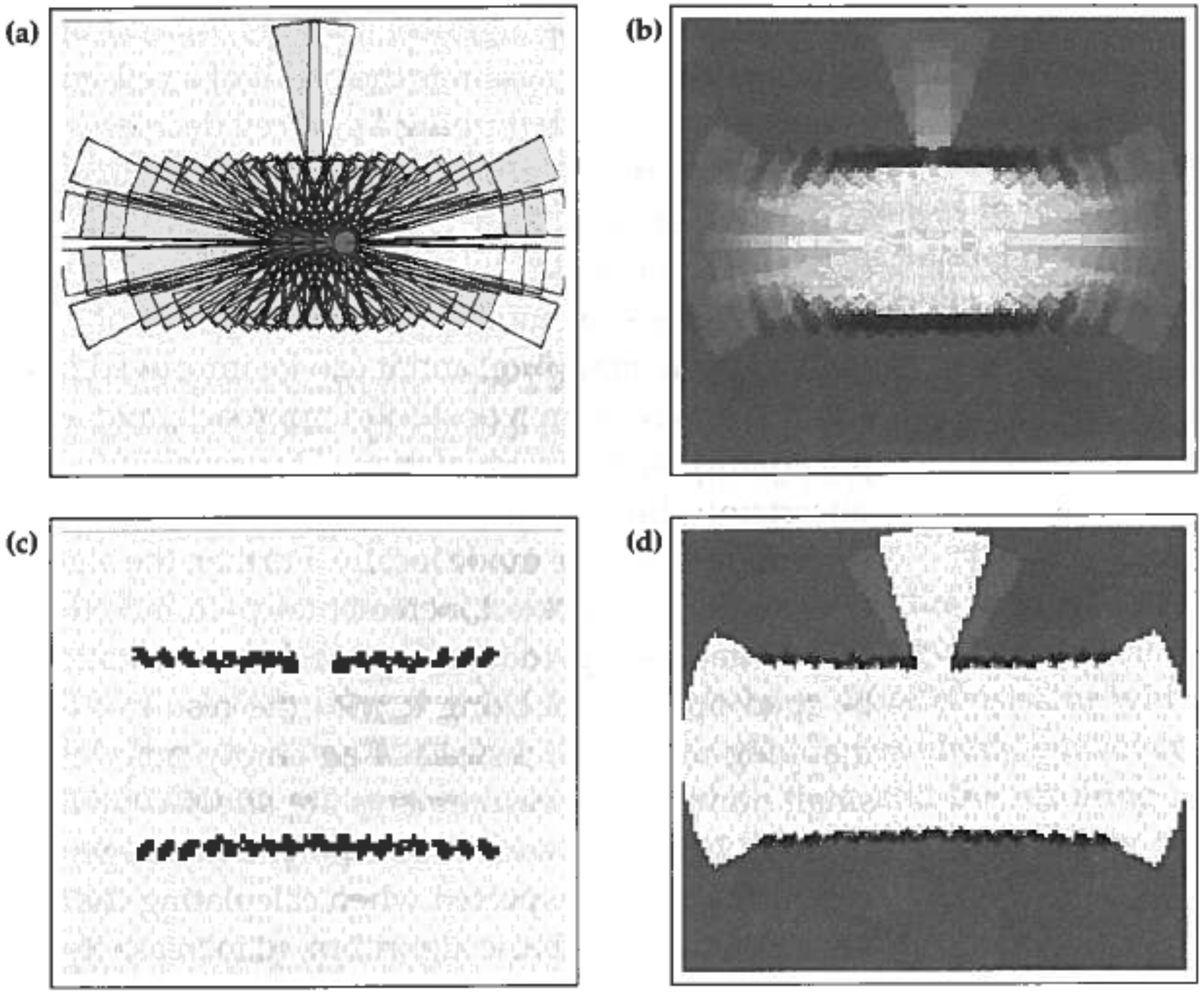
The solution

- Recover the mode of the posterior (but lose measure of probability)

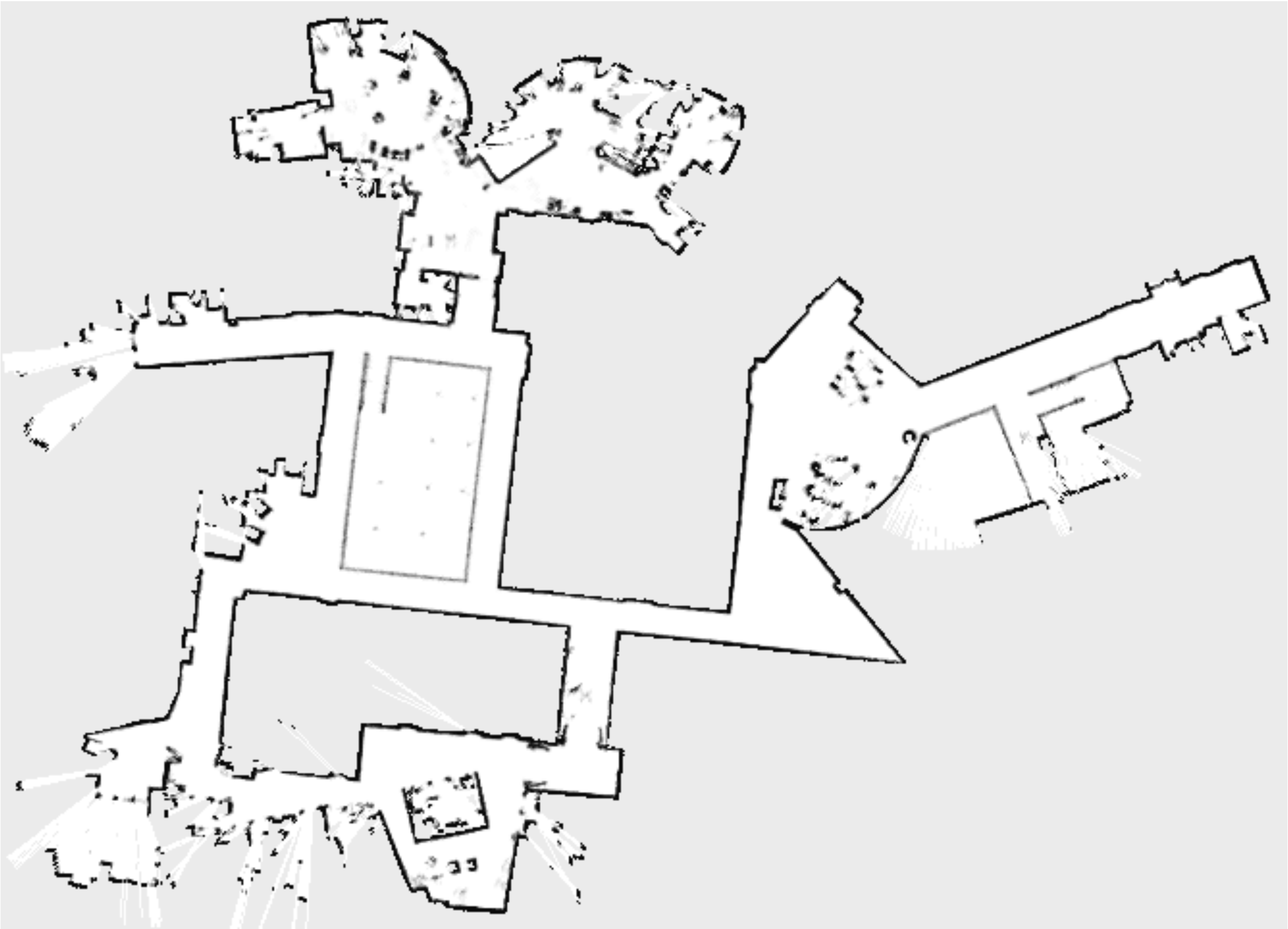
```
1:  Algorithm MAP_occupancy_grid_mapping(x_{1:t}, z_{1:t}):
2:      set m = {0}
3:      repeat until convergence
4:          for all cells m_i do
5:              m_i = argmax_{k=0,1} k l_0 + Σ_t log
                        measurement_model(z_t, x_t, m with m_i = k)
6:          endfor
7:      endrepeat
8:      return m
```

Table 9.3 The maximum a posteriori occupancy grid algorithm, which uses conventional measurement models instead of inverse models.

Example: Maximum a Posteriori OGM

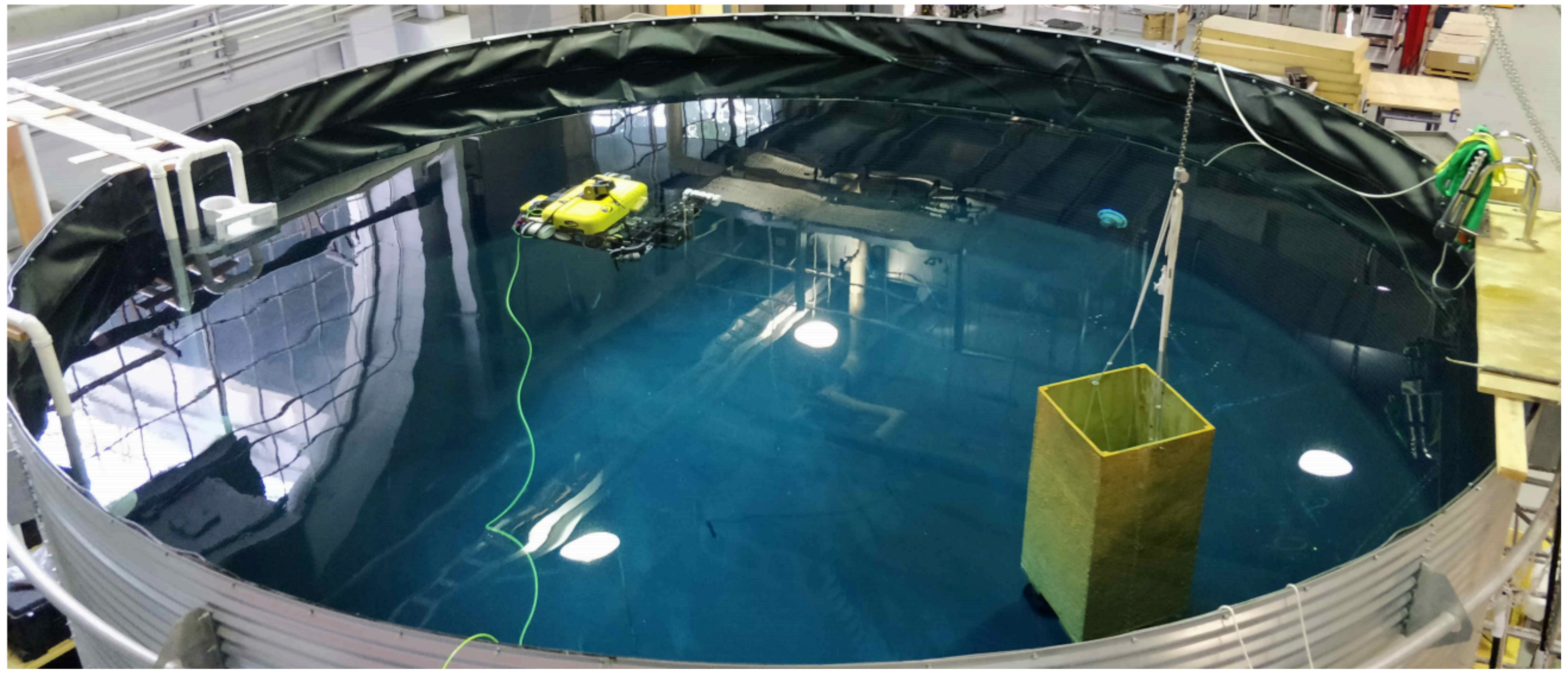


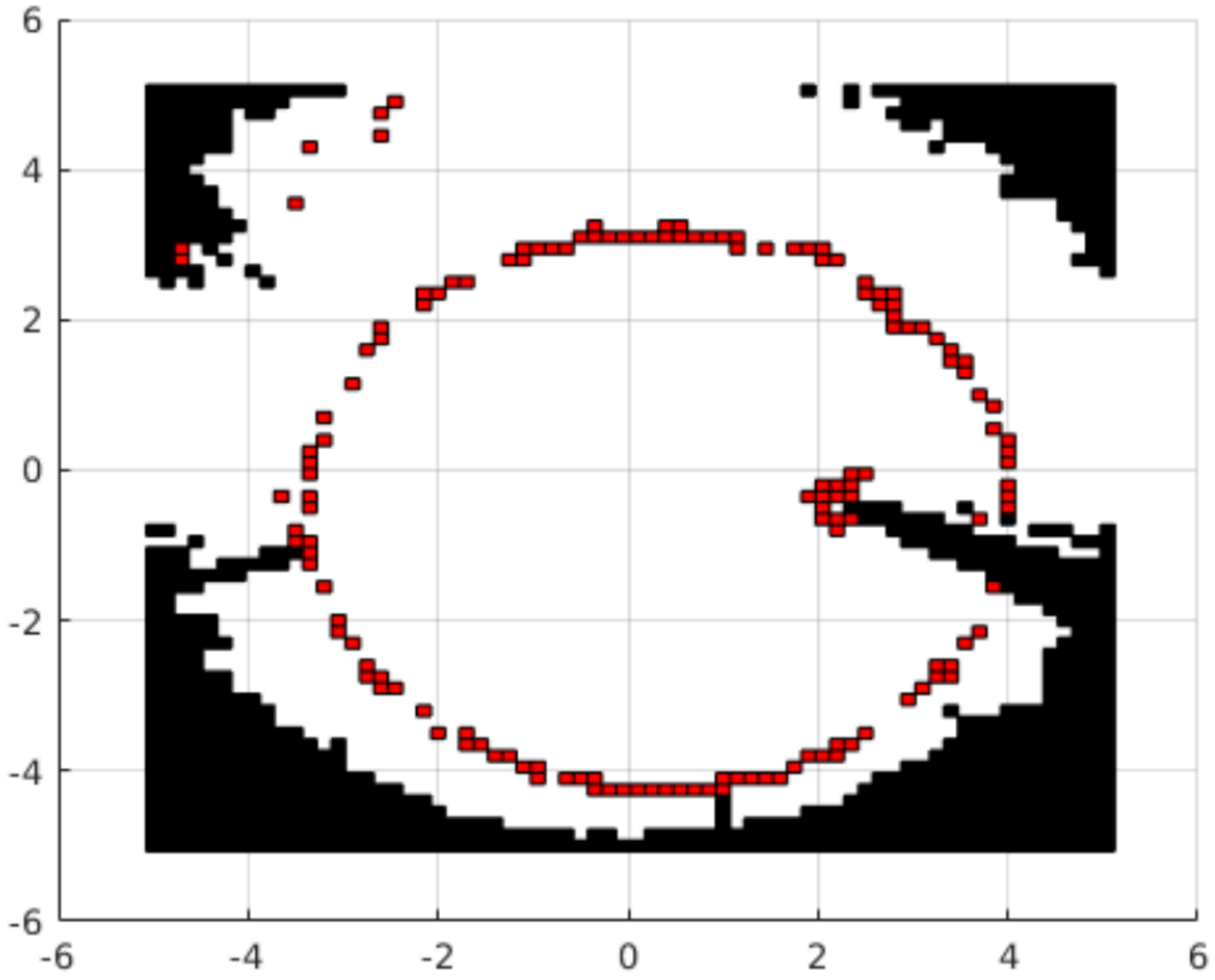
Example: MIT CSAIL 3rd floor



Courtesy: C. Stachniss

Example: underwater mapping

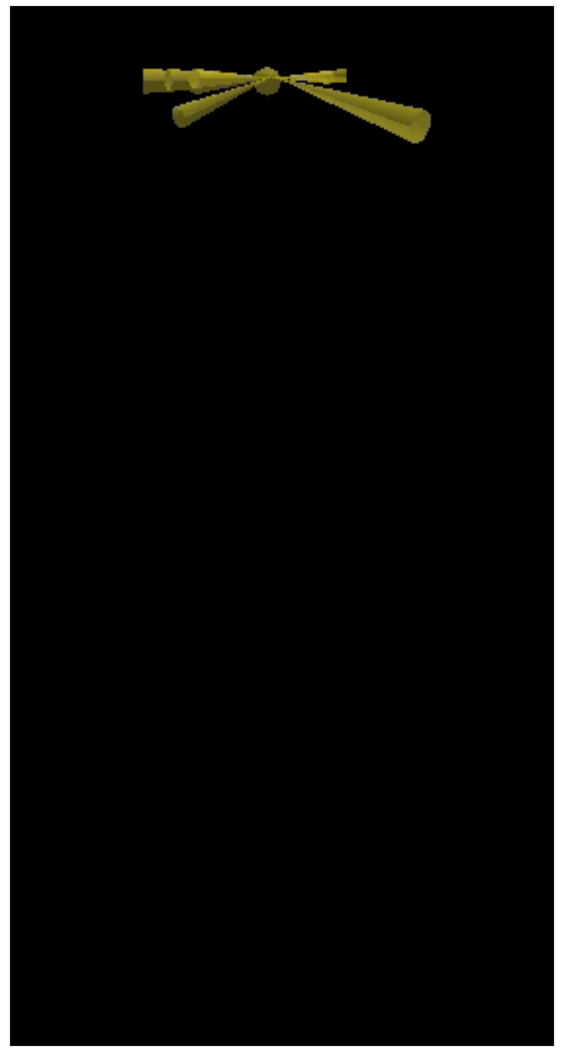




3D OGM: CMU DepthX

- Nathaniel Fairfield, George A. Kantor, and David Wettergreen, "Real-Time SLAM with Octree Evidence Grids for Exploration in Underwater Tunnels," Journal of Field Robotics, 2007

Camera
Science probe
SONAR
IMU + RLG



OGM summary

- OGMs discretize the environment into independent cells
- Each cell is a binary random variable estimating its occupancy
- It is a static state binary Bayes filter per cell
- Mapping with known poses simplifies the problem
- Log-odds model is efficient to compute

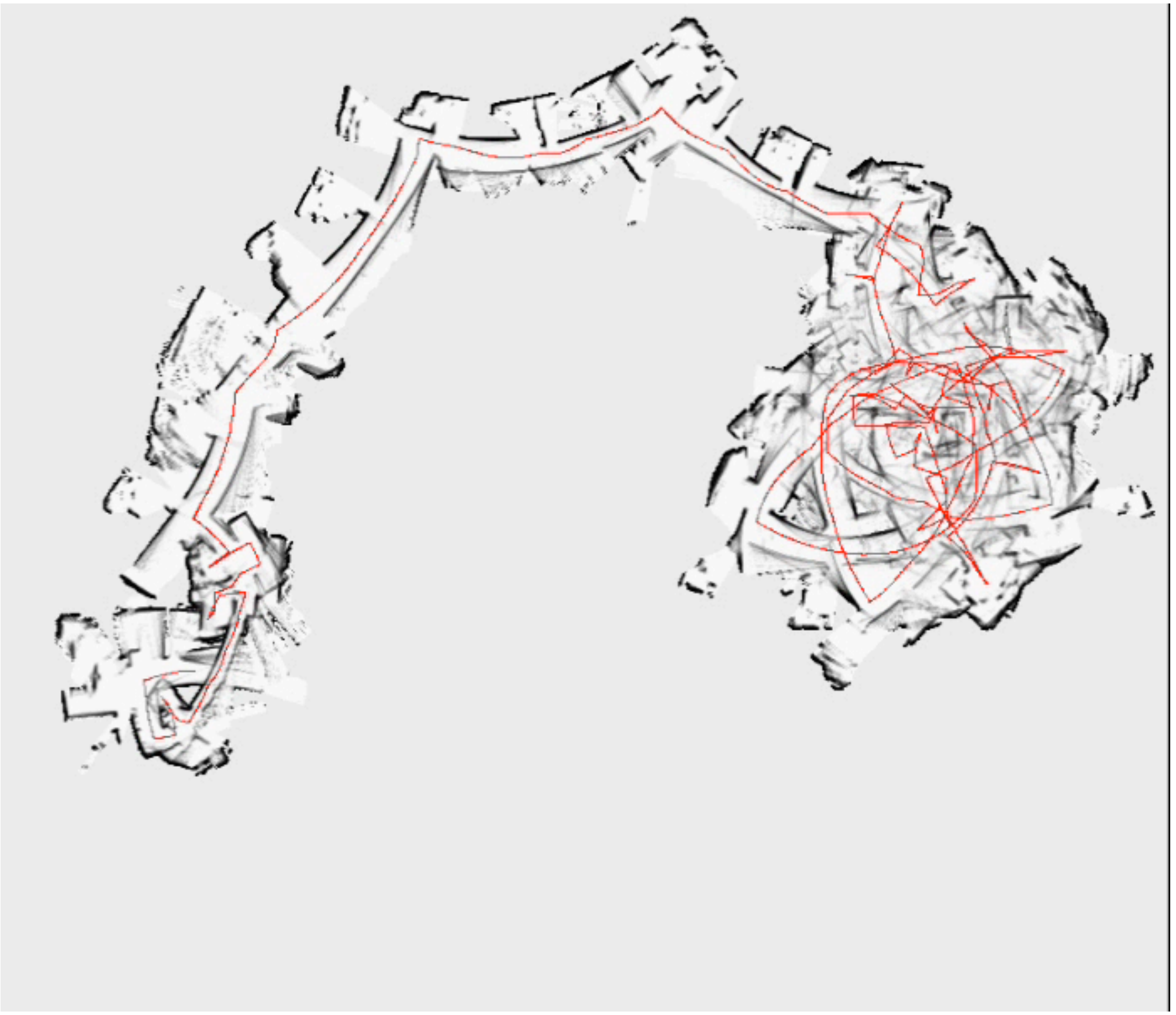
- **Occupancy grid mapping**
  - Representation
  - Assumptions
  - OGM $\rightarrow$ Bayes filter
  - Log-odds updates
  - Examples and drawbacks

- **Scan-matching**
- **Rao-Blackwellization**
- **Beyond OGMs**

Incremental scan alignment

- Pure odometry is noisy
- Known poses assumption falls short
- We can trust sensor precision to correct drift
- Scan-matching (ICP) tries to incrementally align two scans, or a scan to a map

Applications: Maps with raw odometry



Pose correction via scan-matching

- Maximize the likelihood of the **current** pose relative to the **previous** pose and map

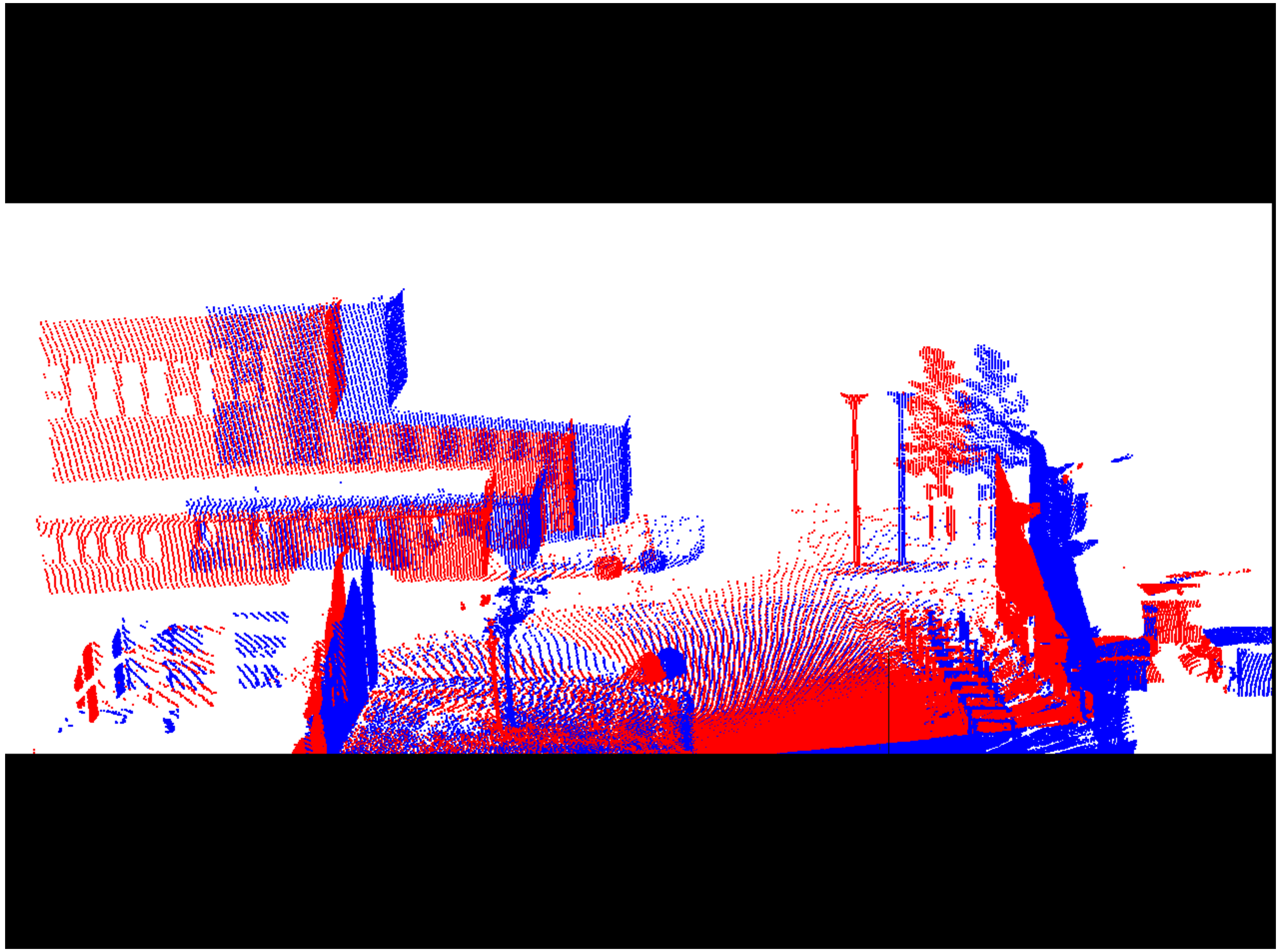
$$\mathbf{x}_t^* = \underset{\mathbf{x}_t}{\operatorname{argmax}} \left\{ p(\mathbf{z}_t \mid \mathbf{x}_t, m_{t-1})\, p(\mathbf{x}_t \mid \mathbf{u}_{t-1}, \mathbf{x}_{t-1}^*) \right\}$$

current measurement

robot motion

map constructed so far

Incremental alignment

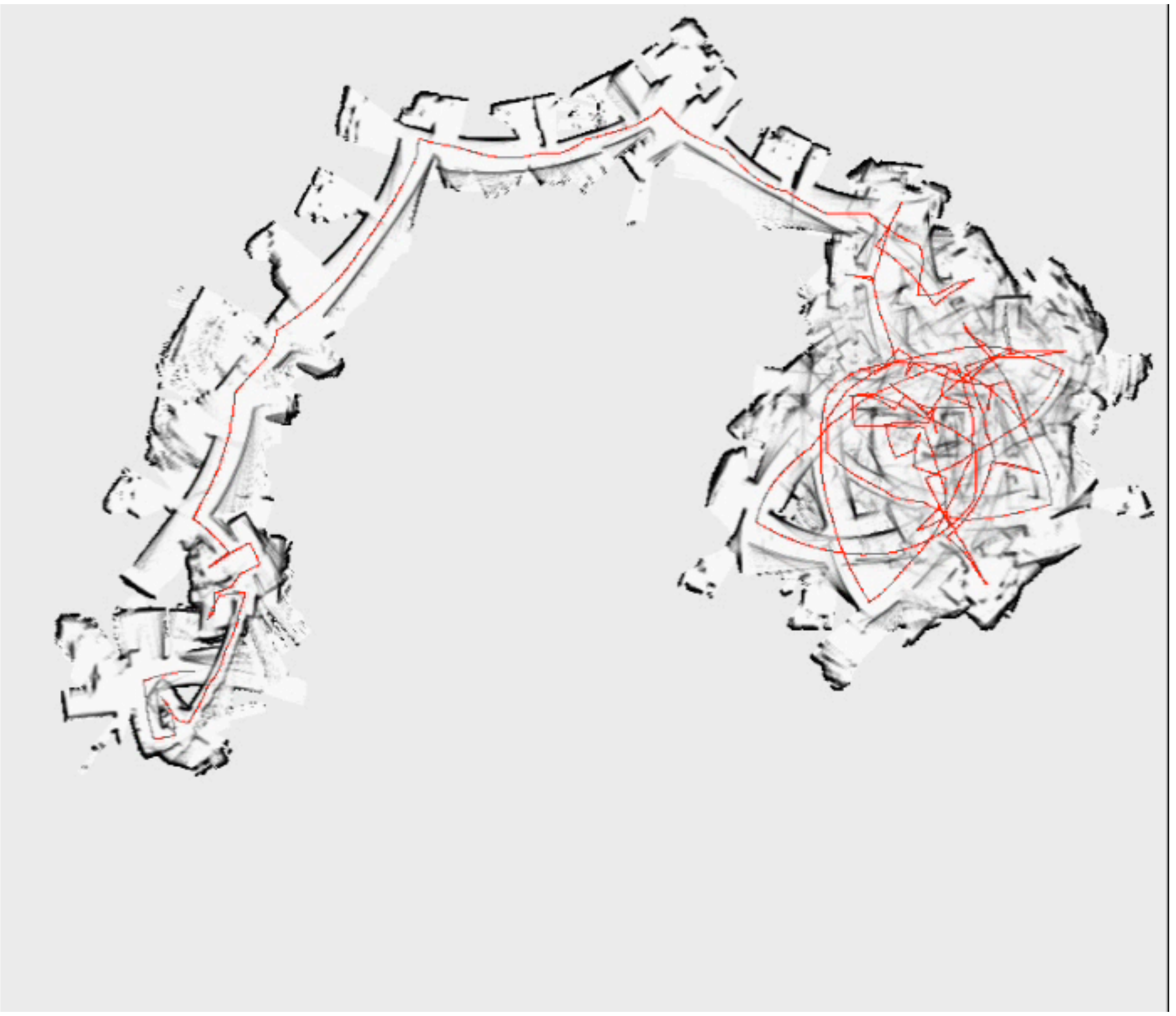


Scan matching techniques

- Iterative closest point (ICP)
- Scan-to-scan
- Scan-to-map
- Map-to-map
- Feature-based
- With RANSAC for outlier rejection
- Correlative matching
- …

Courtesy: E. Olson

With and without scan matching



Scan-matching summary

- Improves pose-estimates, and consequently, the underlying map
- Maintains local consistency
- But probably not sufficient to build a **large** consistent map (psst… large-scale loop closures)

- **Occupancy grid mapping**
  - Representation
  - Assumptions
  - OGM $\rightarrow$ Bayes filter
  - Log-odds updates
  - Examples and drawbacks

- **Scan-matching**
- **Rao-Blackwellization**
- **Beyond OGMs**

Exploiting dependencies

$$x_{1:t}\, , \, m$$

If we know the poses of the robot, mapping is easy!

- **If we use the particle set to model robot path, each sample is a path hypothesis.**
- **For each particle, we can compute an individual map using it's path.**

Rao-Blackwellization

- First introduced by Murphy in 1999

$$p(a, b) = p(a)\ p(b \mid a)$$

- If $p(b \mid a)$ can be efficiently computed, we can represent $p(a)$ with samples and compute $p(b \mid a)$ for each sample!

$$p(\mathbf{x}_{1:t}, m \mid \mathbf{z}_{1:t}, \mathbf{u}_{0:t-1}) = p(\mathbf{x}_{1:t} \mid \mathbf{z}_{1:t}, \mathbf{u}_{0:t-1}) \cdot p(m \mid \mathbf{x}_{1:t}, \mathbf{z}_{1:t})$$

poses map observations & movements

Rao-Blackwellization

$$\underbrace{p(\mathbf{x}_{1:t}, m \mid \mathbf{z}_{1:t}, \mathbf{u}_{0:t-1})}_{\textbf{SLAM posterior}} = \underbrace{p(\mathbf{x}_{1:t} \mid \mathbf{z}_{1:t}, \mathbf{u}_{0:t-1})}_{\textbf{Robot path posterior}} \cdot \underbrace{p(m \mid \mathbf{x}_{1:t}, \mathbf{z}_{1:t})}_{\textbf{Mapping with known poses}}$$

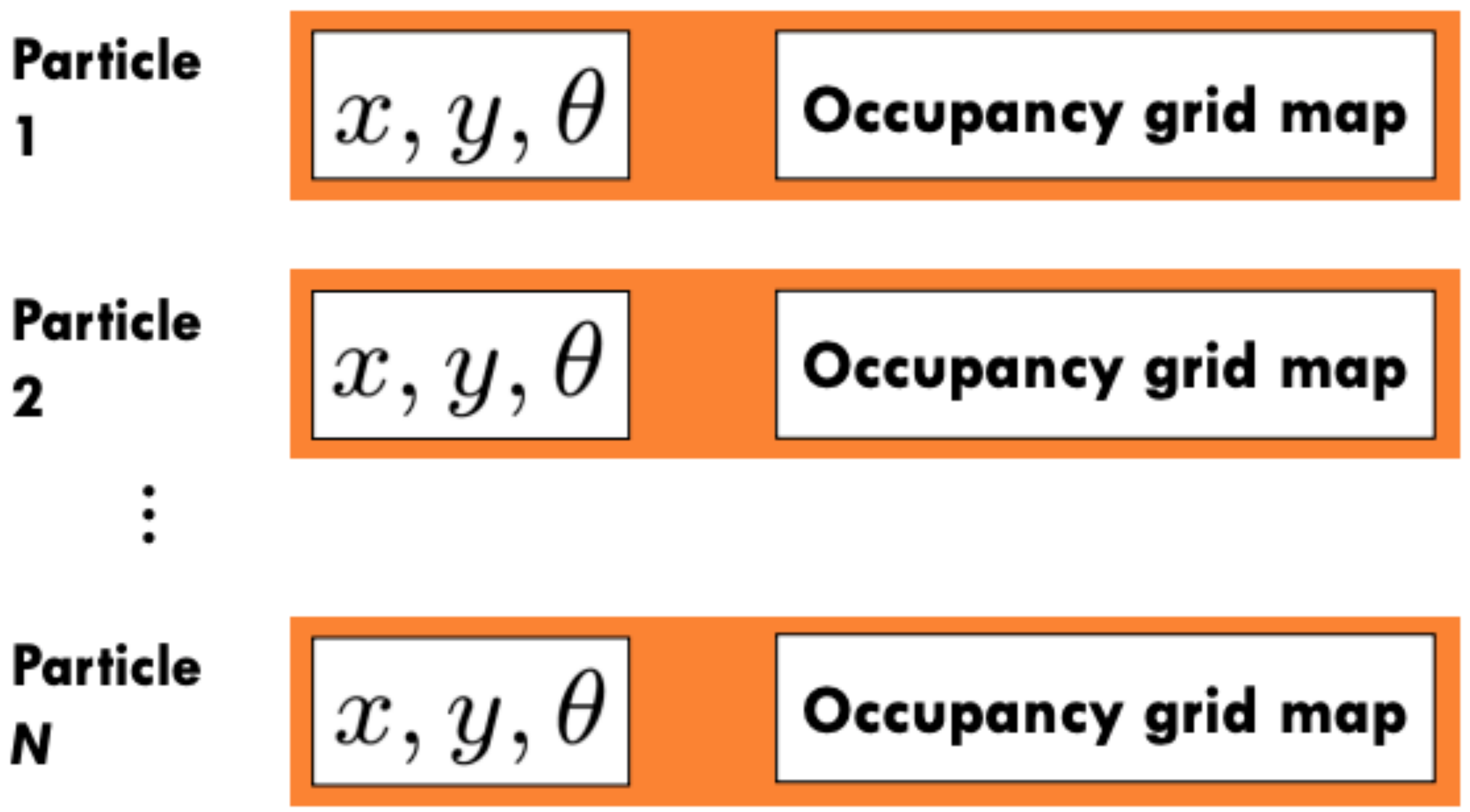
poses map observations & movements

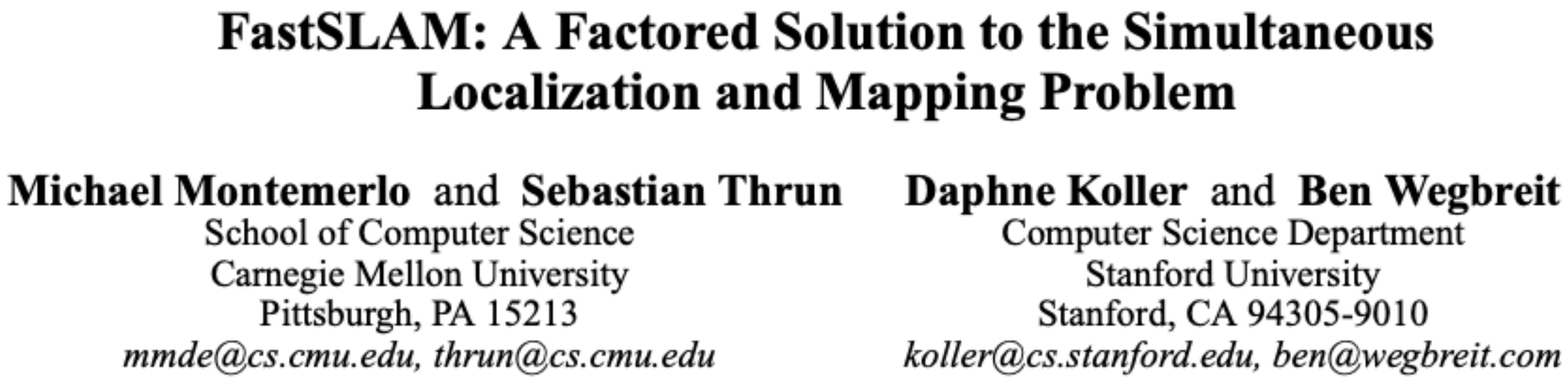
Rao-Blackwellization

$$\underbrace{p(\mathbf{x}_{1:t}, m \mid \mathbf{z}_{1:t}, \mathbf{u}_{0:t-1})}_{\text{SLAM posterior}} = \underbrace{p(\mathbf{x}_{1:t} \mid \mathbf{z}_{1:t}, \mathbf{u}_{0:t-1})}_{\text{Localization: use MCL!}} \cdot \underbrace{p(m \mid \mathbf{x}_{1:t}, \mathbf{z}_{1:t})}_{\text{Mapping: use pose estimate from MCL and apply mapping with known poses}}$$

FastSLAM (2002)

- Uses Rao-Blackwellization, each particle has a pose and a map



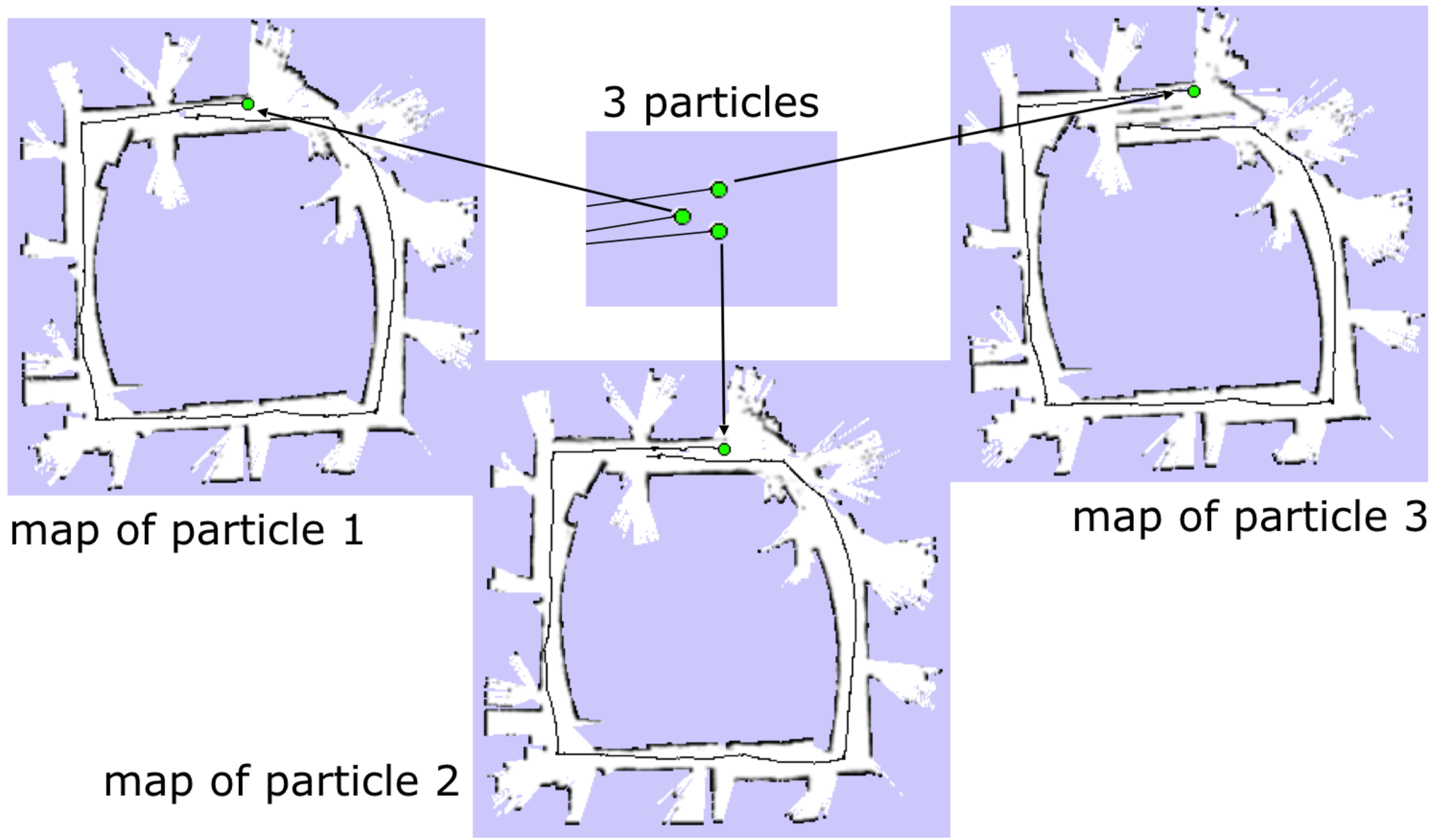


FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem

Michael Montemerlo and **Sebastian Thrun**
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
mmde@cs.cmu.edu, thrun@cs.cmu.edu

Daphne Koller and **Ben Wegbreit**
Computer Science Department
Stanford University
Stanford, CA 94305-9010
koller@cs.stanford.edu, ben@wegbreit.com

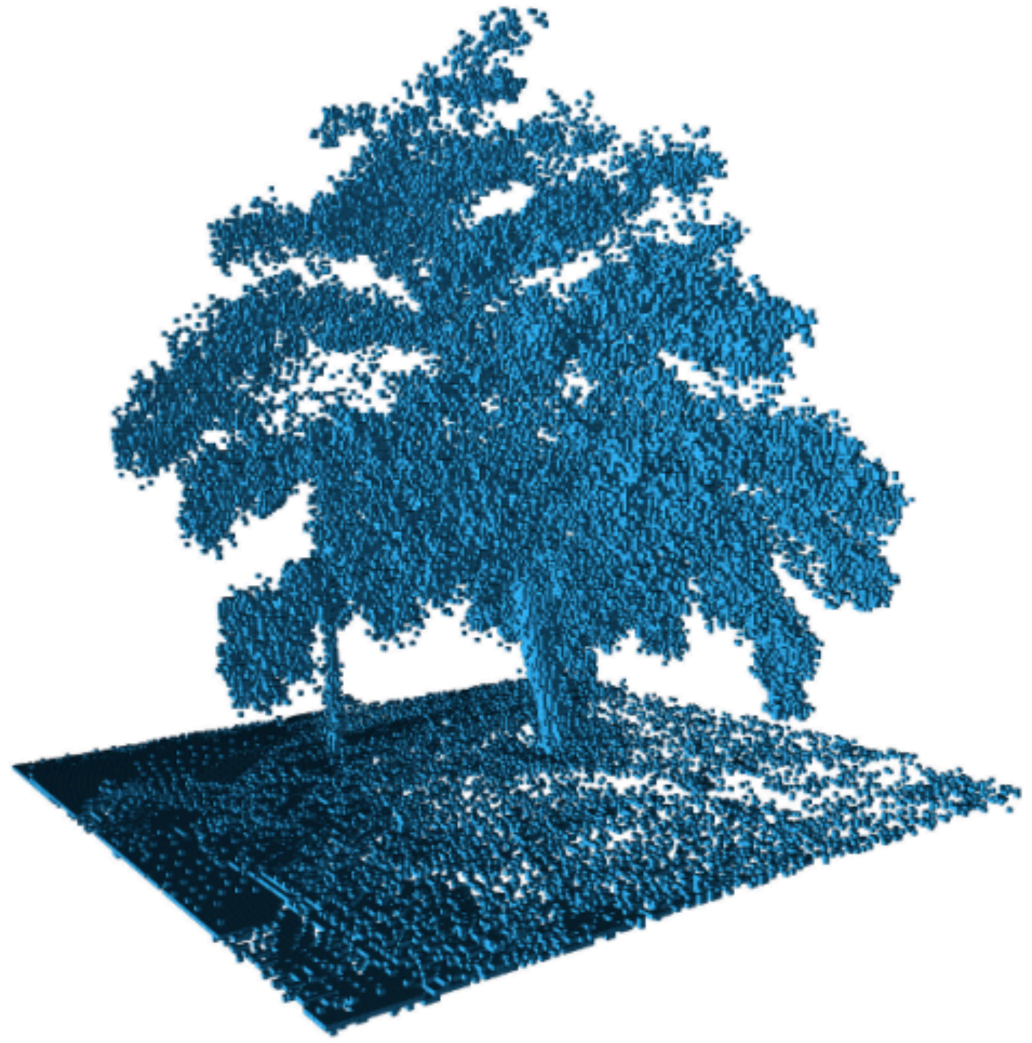
Example: with particle filter



Carnegie Mellon
THE ROBOTICS INSTITUTE

- **Occupancy grid mapping**
  - Representation
  - Assumptions
  - OGM → Bayes filter
  - Log-odds updates
  - Examples and drawbacks
- **Scan-matching**
- **Rao-Blackwellization**
- **Beyond OGMs**

Beyond OGM



3D occupancy grid mapping:

Pros:

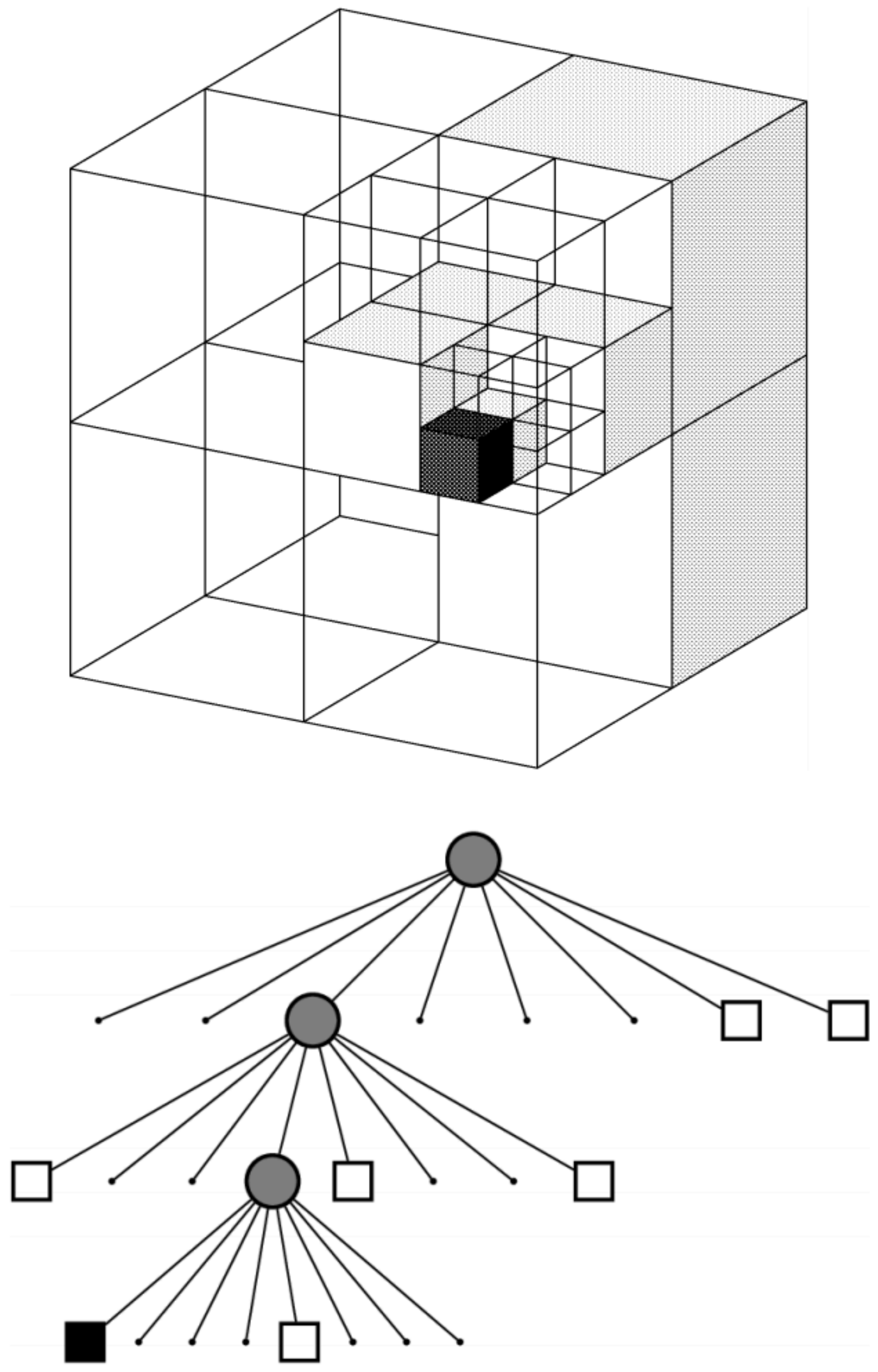
- Probabilistic update
- Constant time access

Cons:

- Memory requirement
    - Extent of map must be known
    - Complete map is allocated in memory

Octrees

- Tree-based data structure
- Recursive subdivision of space into octants
- Volumes allocated as needed
- Multi-resolution



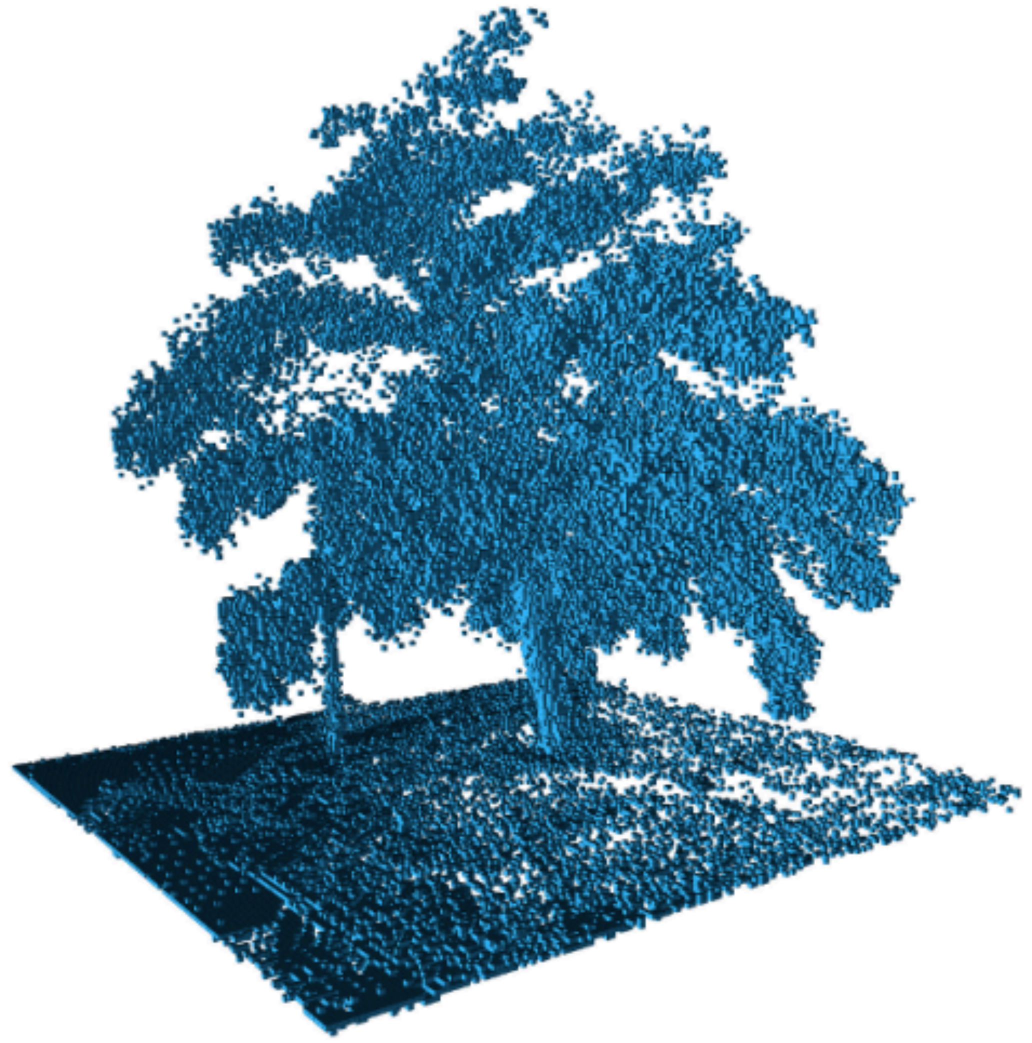
Octrees

Pros:

- Probabilistic update
- Flexible, multi-resolution
- Memory efficient

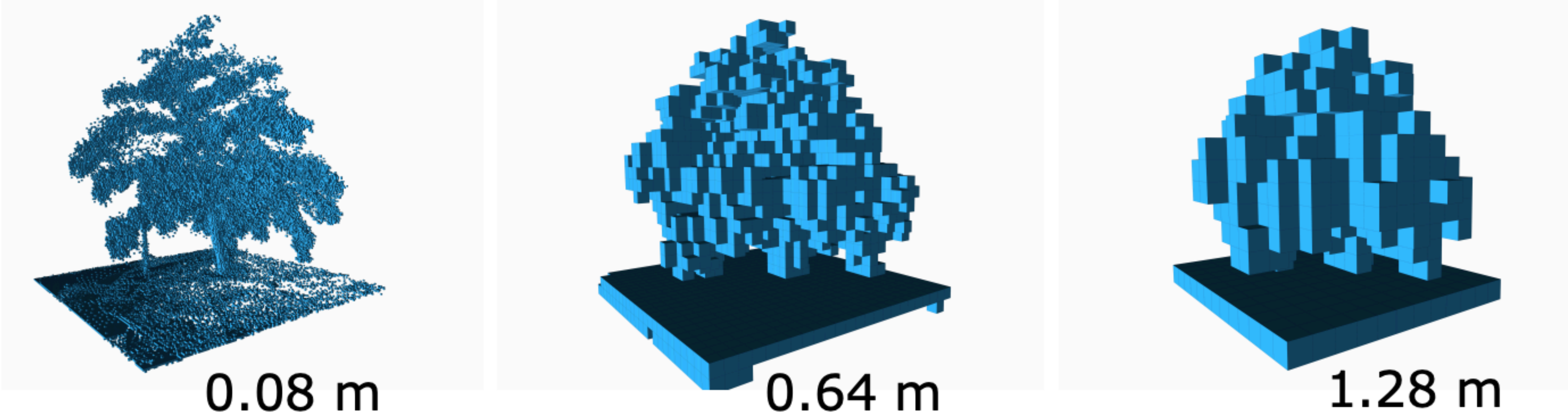
Cons:

- Tricky implementation (memory, update, map files…)



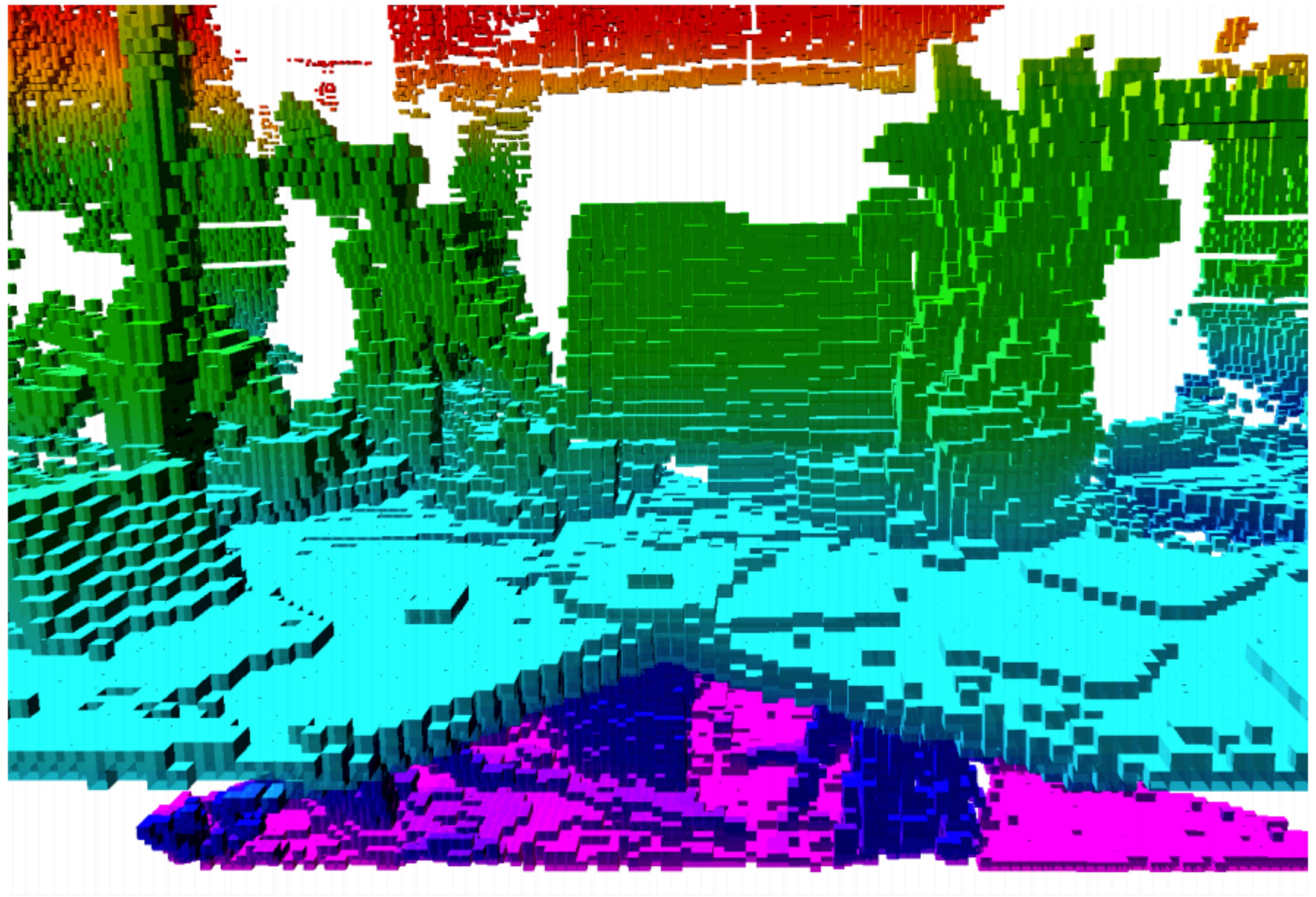
Open-source implementation

- **Octamap:** http://octomap.github.io/
- Supports multi-resolution map queries, lossless compression, compact map files



Open-source implementation

- **Octamap:** http://octomap.github.io/
- Supports multi-resolution map queries, lossless compression, compact map files

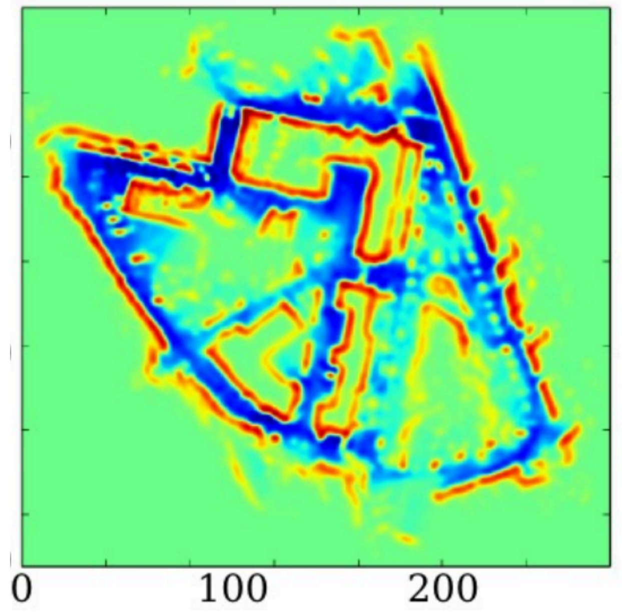


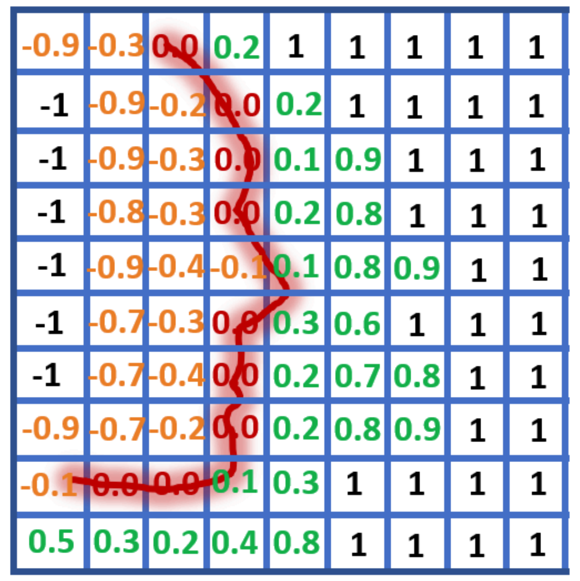
Map resolution: 2 cm

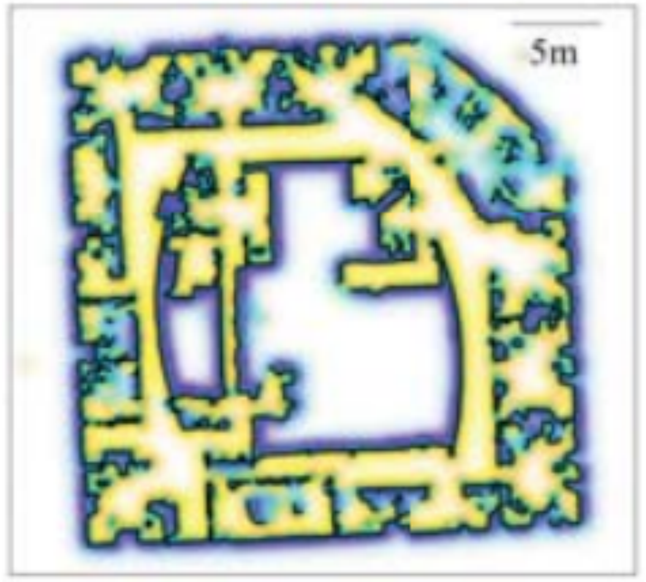
Beyond OGM

Alternate representations:

- SDFs/TSDFs
- Hilbert maps
- Gaussian processes







Interesting research directions:

- Maps at arbitrary resolutions, without grid-cell discretization
- Encode spatial relationships naturally
- Outlier robustness

References

- **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 (Grid mapping)
    - Chapter 6 (Beam-based sensor models)
- **Scan-Matching**
  - Besl and McKay. A method for Registration of 3-D Shapes, 1992
  - Olson. Real-Time Correlative Scan Matching, 2009