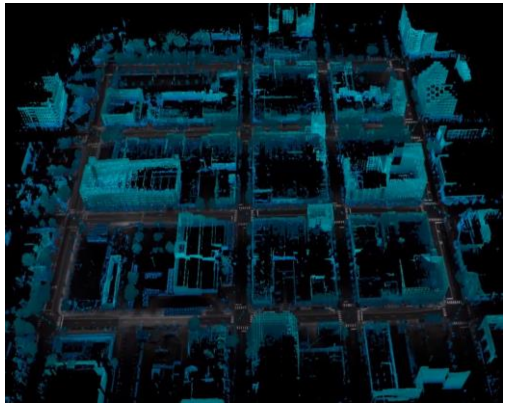
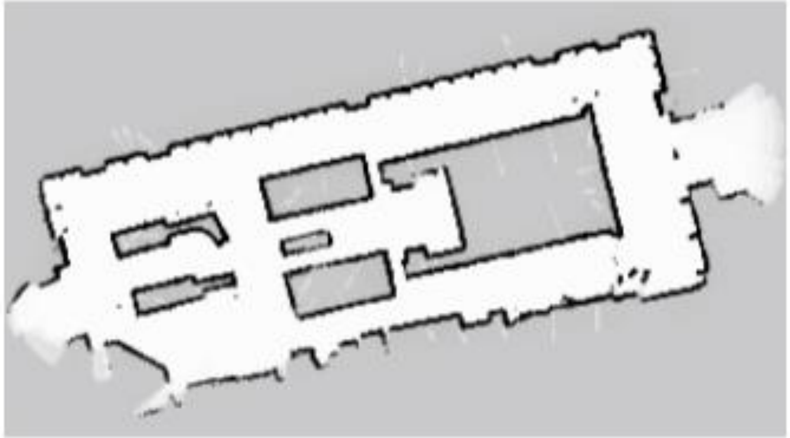
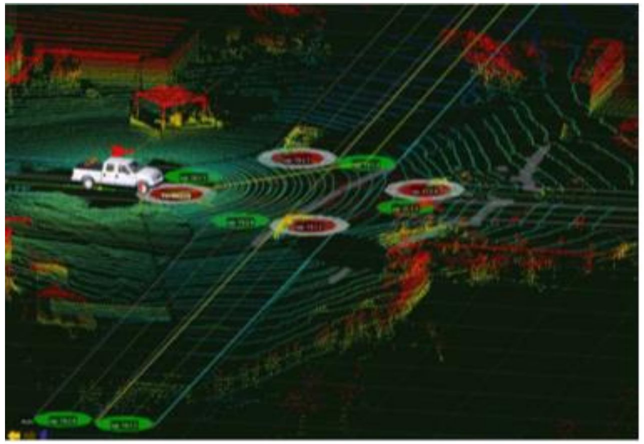
BYU Electrical & Computer Engineering





OCCUPANCY GRID MAPPING

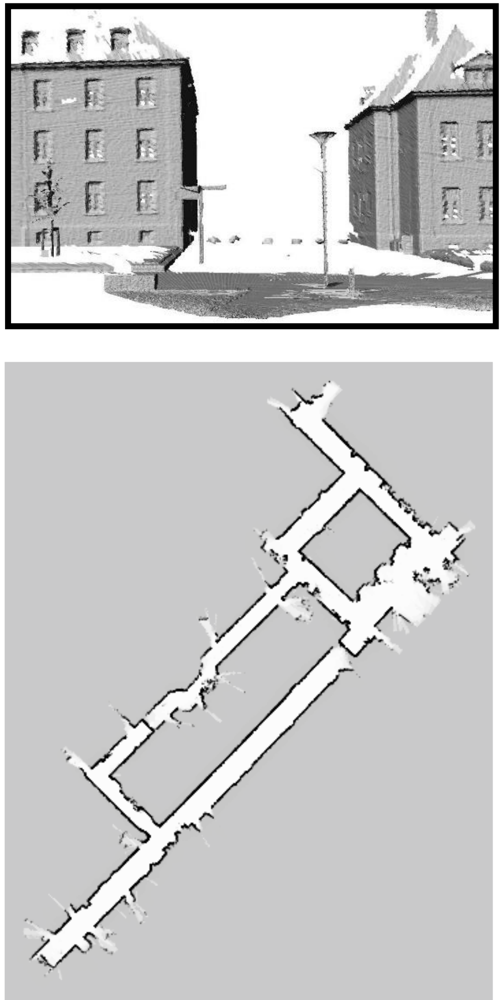
ECEN 633: Robotic Localization and Mapping

Slides Based on probabilistic-robotics.org and a slide deck by Ryan Eustice.

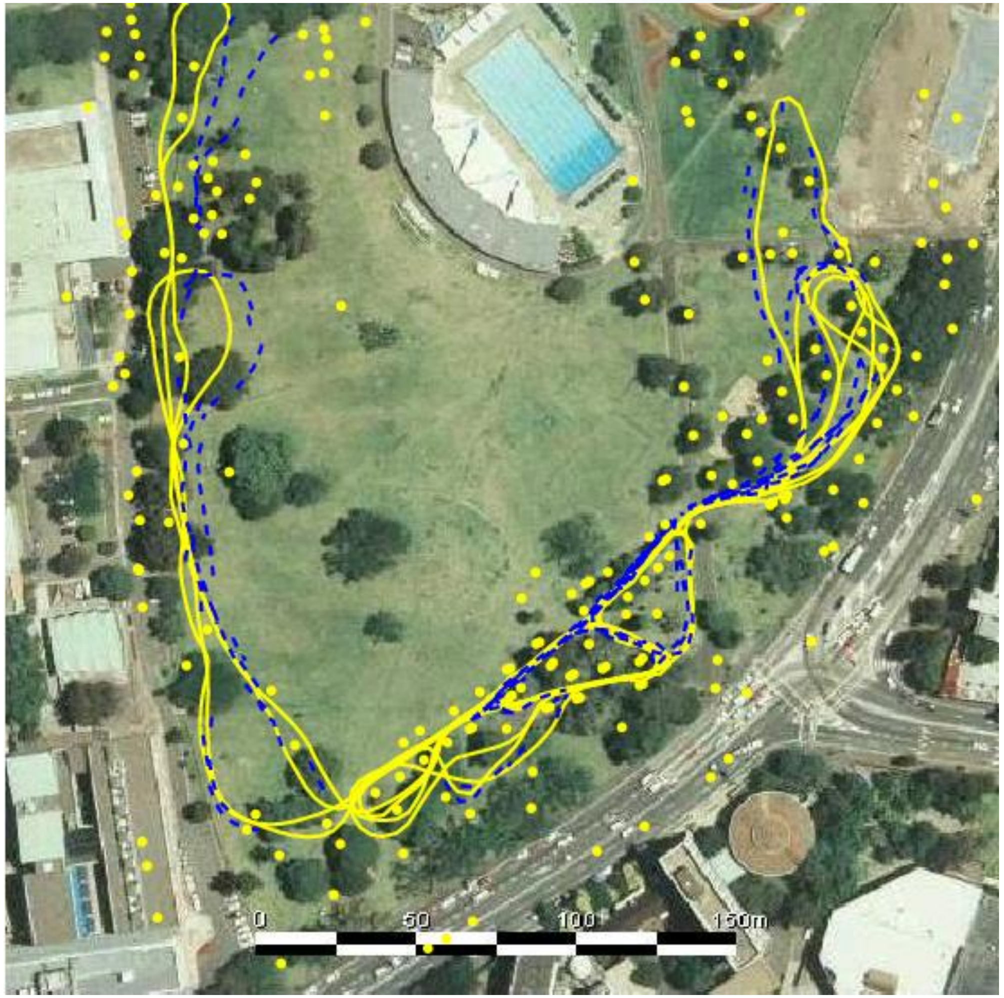
Agenda

- Map Representations
- Occupancy Grids
- Occupancy Grid Mapping

Map Representation: Volumetric Maps vs Features



Courtesy: D. Hähnel



Courtesy: E. Nebot

Feature Based Maps

- Natural choice for SLAM systems
- Compact representation
- Multiple feature observations improve the landmark position estimate

- Do NOT model what space is free of obstacles.

Grid Maps

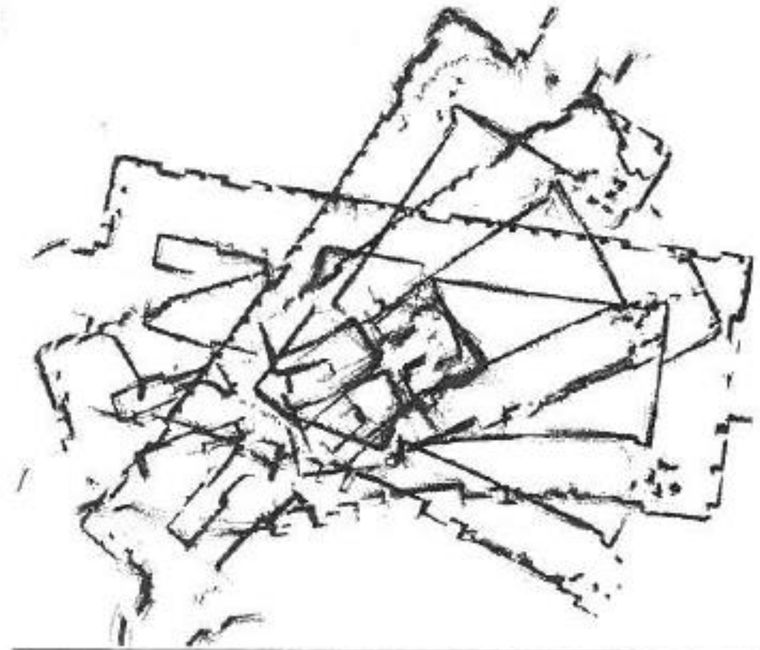
- Discretize the world into cells
- Grid structure is rigid
- Each cell is assumed to be occupied or free space
- Non-parametric model
- Large maps require substantial memory resources
- Do not rely on a feature detector

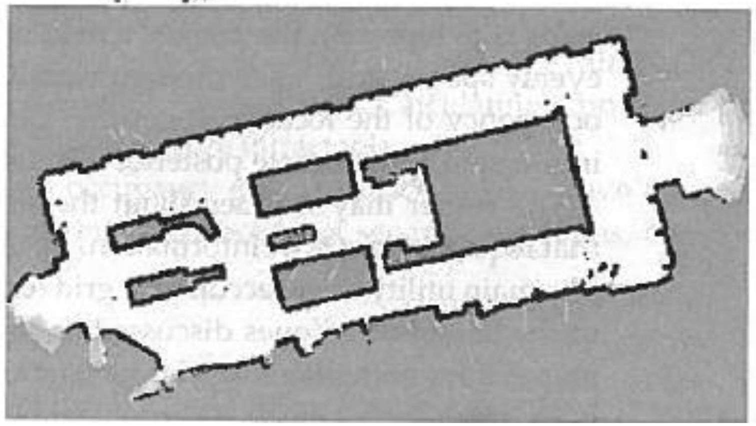
Example

Courtesy: C. Stachniss

Occupancy maps provide a notion of "free space" - Important for planning

- Raw laser scans placed using odometry
  - Inconsistent
- Solve for poses using SLAM
  - Consistent, but no notion of free space
- Use SLAM-derived poses as input to generate an occupancy map
  - Planning, localization

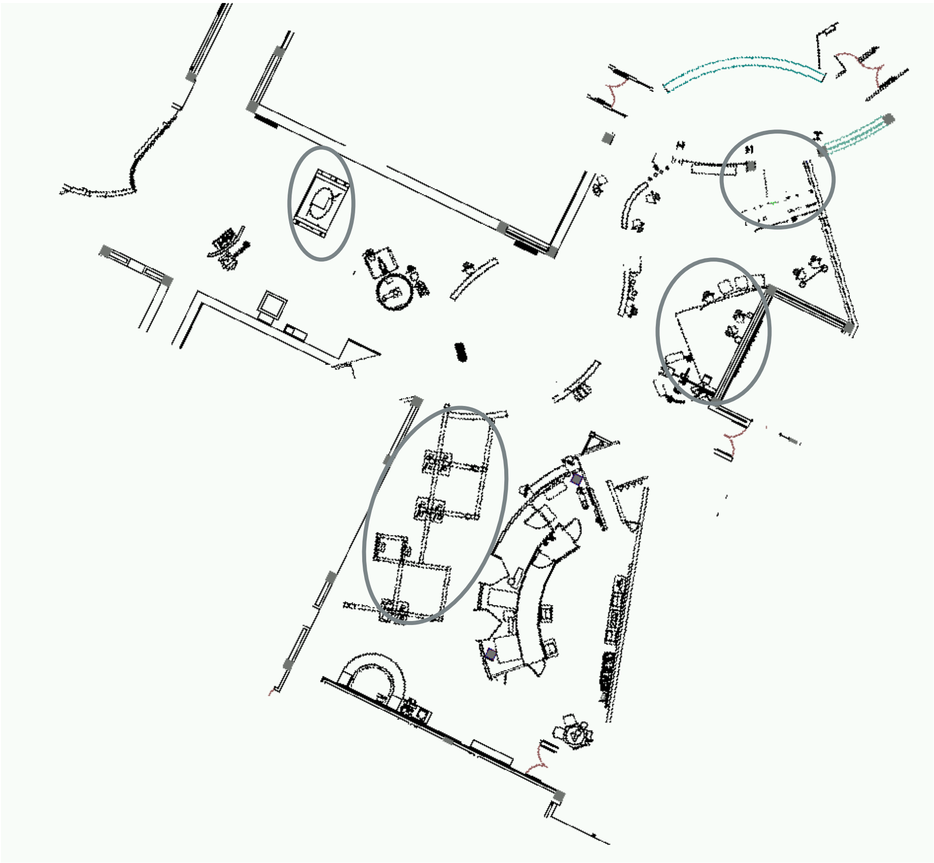




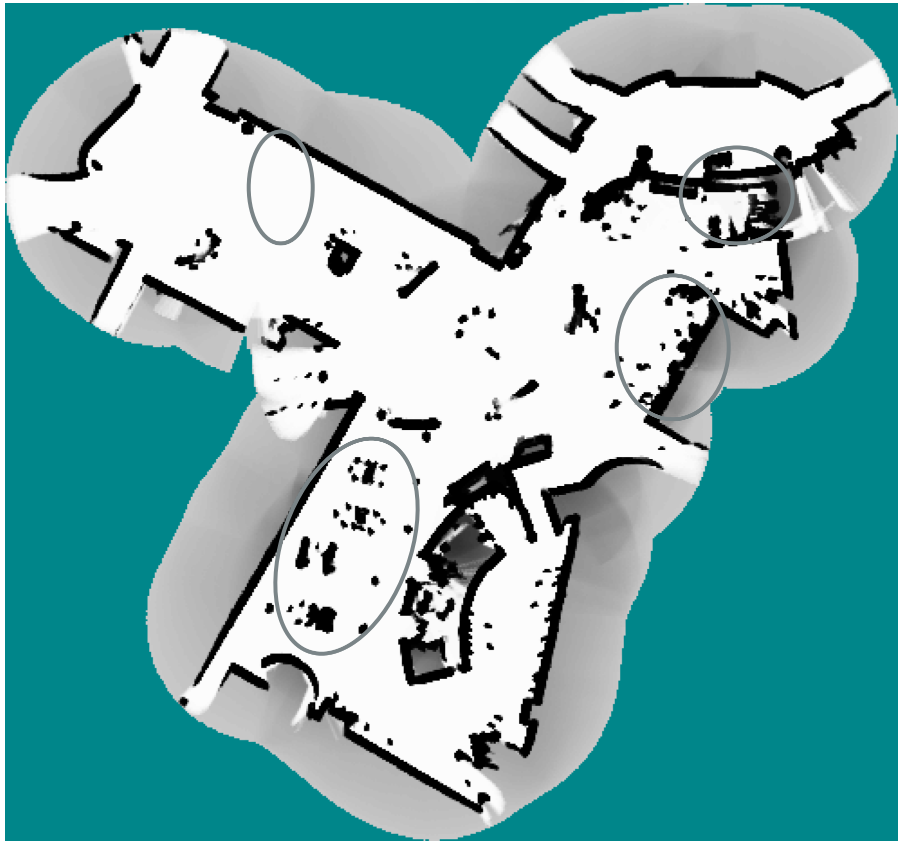
Old Algorithms – New Technology

- https://youtu.be/Gh5pAT1o2V8?t=1m21s

Even when we have prior maps, they may be inaccurate…



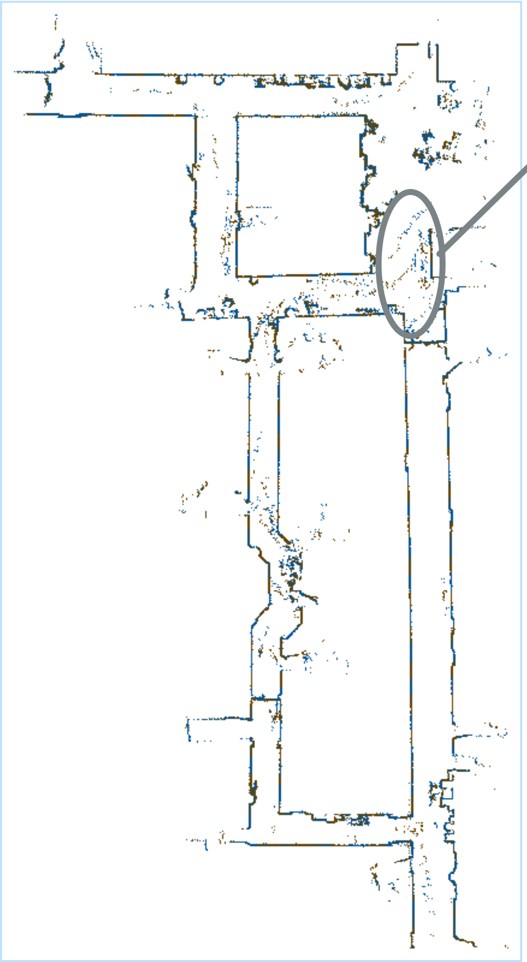
CAD map



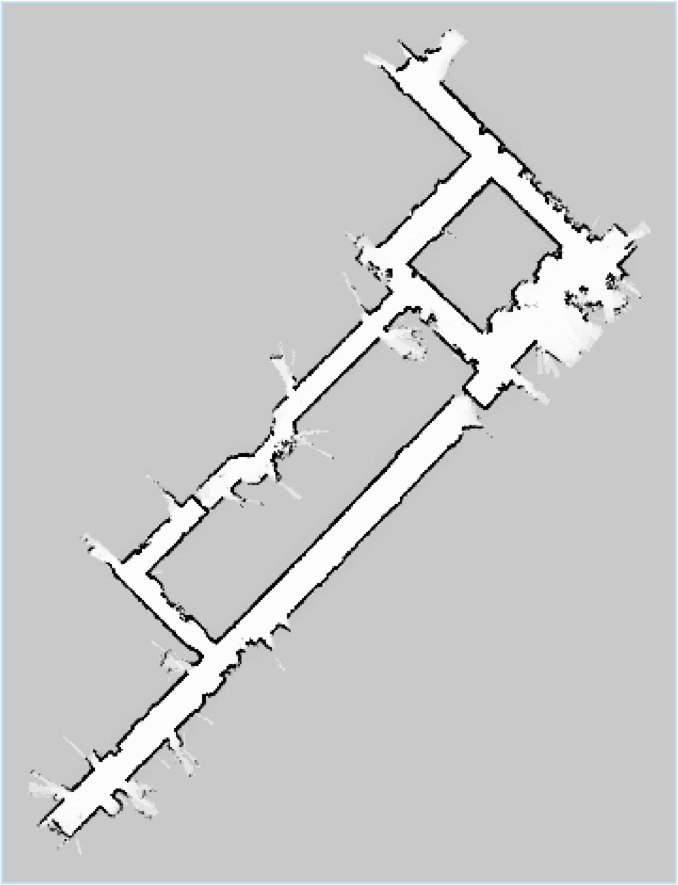
occupancy grid map

Tech Museum, San Jose

Occupancy Grids: From scans to maps



False "hits" from people in the environment



Occupancy Grid Maps

- Moravec and Elfes proposed occupancy grid mapping in the mid 1980's
- Developed for noisy sonar sensors
- Also called "mapping with known poses"

Occupancy Grid Maps

- Introduced by Moravec and Elfes in 1987
- Represent environment by a grid
  - e.g. 25 m x 25 m area at 25 cm resolution yields a 100 x 100 grid = 10,000 cells
- Estimate the probability that a cell is occupied by an obstacle.

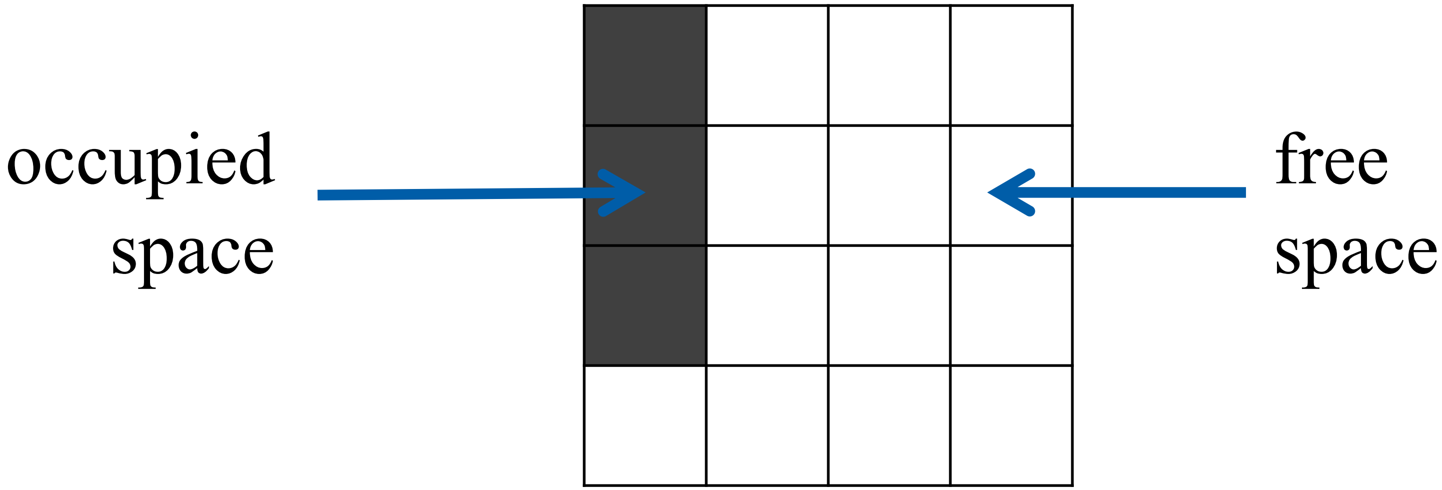
  Binary state: $\mathbf{m}_i = \begin{cases} 1 & \text{occupied} \\ 0 & \text{free} \end{cases}$

- Map: $m = \{m_i\}$ Belief: $bel_t(m) = p(m \mid \mathbf{z}_{1:t}, \mathbf{x}_{1:t})$
  - Discrete Bayes estimation problem.
  - In our example above, how many possible maps?

That's 10^{3010} maps!!!

Assumption 1

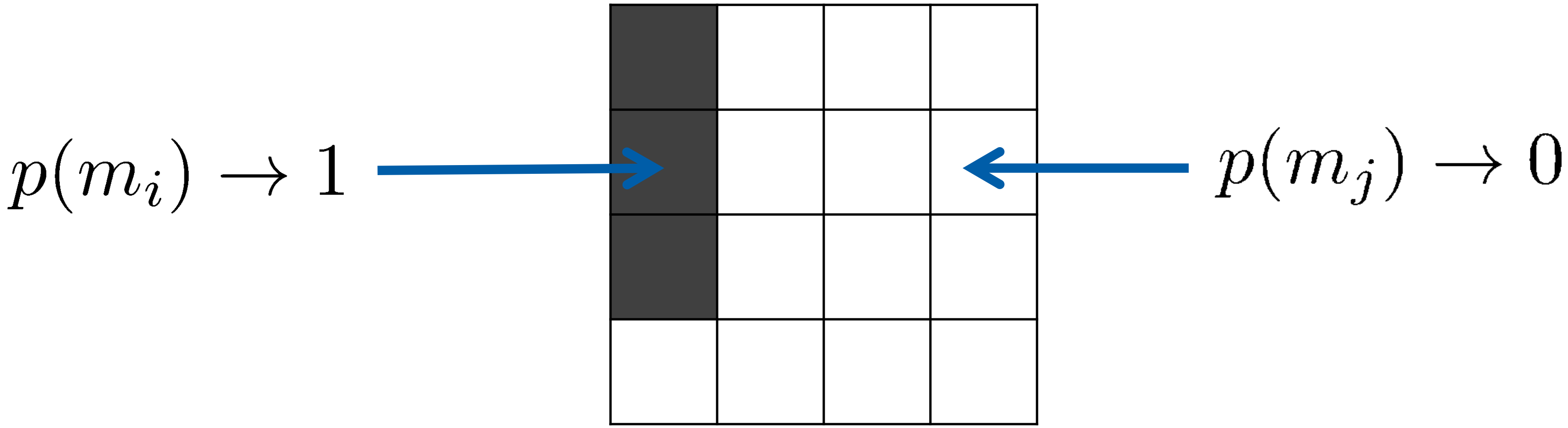
- The area that corresponds to a cell is either completely free or occupied



Courtesy: C. Stachniss

Representation

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



Courtesy: C. Stachniss

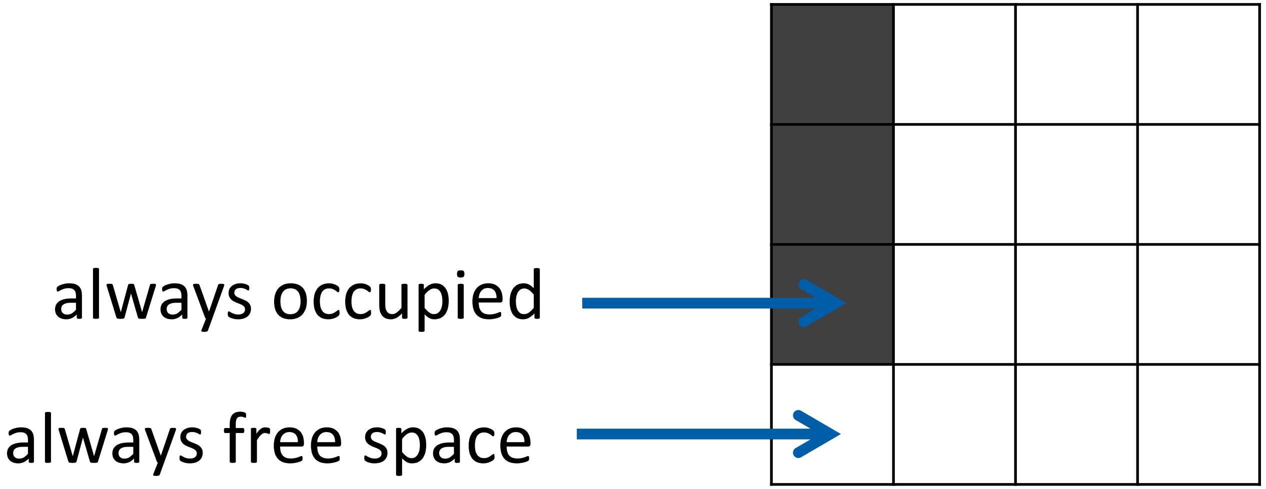
Occupancy Probability

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

Courtesy: C. Stachniss

Assumption 2

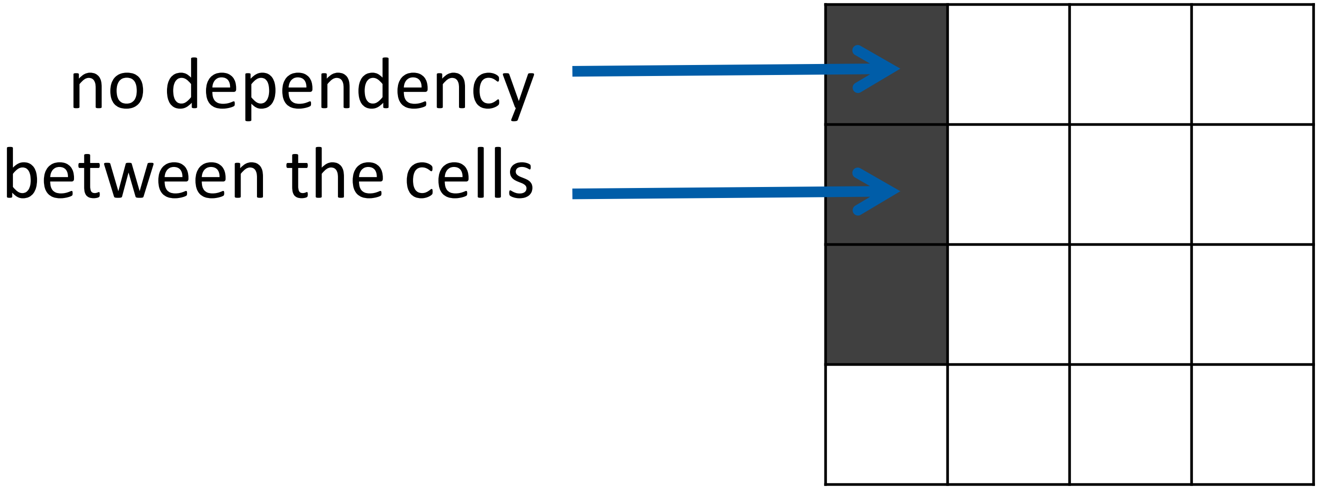
- The world is **static** (most mapping systems make this assumption)



Courtesy: C. Stachniss

Assumption 3

- 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

Assumption –
Cells are Independent

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

map

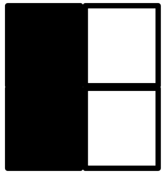
cell

Courtesy: C. Stachniss

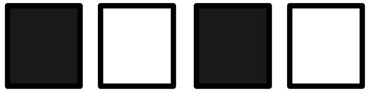
Representation

- 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 a Map From Data

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

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

binary random variable

$\Rightarrow$ Binary Bayes filter (for a static state)

Courtesy: C. Stachniss

Occupancy Grid Maps (OGM)

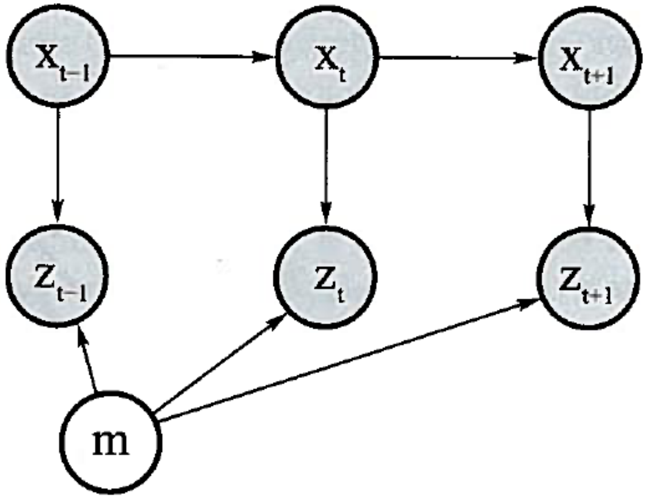
- Key **assumptions** (for tractability)
  - Each cell is completely free or occupied
  - Each cell is static
  - Robot positions are <u>known!</u>
  - Occupancy of individual cells (m_i) are <u>independent</u>

Bayes Net:

- Nodes denote RVs
- Edges denote cond. indep.
  - RVs are cond. Indep (given the variables with edges pointing to them) from all other variables.

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

- OGM Bayes Net (Graphical Model)
  - **z** and **x** are known (shaded)
  - Goal is to infer map m
  - Controls **u** play no roll in the belief since **x** is <u>given</u>



Static State Binary Bayes Filter

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

Courtesy: C. Stachniss

Static State Binary Bayes Filter

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

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

Courtesy: C. Stachniss

Static State Binary Bayes Filter

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

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

Forward model

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

- Rewriting in terms of inverse sensor model we have:

Inverse model

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

Courtesy: C. Stachniss

Static State Binary Bayes Filter

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

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

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

Courtesy: C. Stachniss

Static State Binary Bayes Filter

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

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

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

Courtesy: C. Stachniss

Static State Binary Bayes Filter

$$
\begin{aligned}
p(m_i \mid \mathbf{z}_{1:t}, \mathbf{x}_{1:t}) &\overset{\text{Bayes rule}}{=} \frac{p(\mathbf{z}_t \mid m_i, \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})\, p(m_i \mid \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})}{p(\mathbf{z}_t \mid \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})} \\
&\overset{\text{Markov}}{=} \frac{p(\mathbf{z}_t \mid m_i, \mathbf{x}_t)\, p(m_i \mid \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-1})}{p(\mathbf{z}_t \mid \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})} \\
&\overset{\text{Bayes rule}}{=} \frac{p(m_i \mid \mathbf{z}_t, \mathbf{x}_t)\, p(\mathbf{z}_t \mid \mathbf{x}_t)\, p(m_i \mid \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-1})}{p(m_i \mid \mathbf{x}_t)\, p(\mathbf{z}_t \mid \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})} \\
&\overset{\text{indep.}}{=} \frac{p(m_i \mid \mathbf{z}_t, \mathbf{x}_t)\, p(\mathbf{z}_t \mid \mathbf{x}_t)\, p(m_i \mid \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-1})}{p(m_i)\, p(\mathbf{z}_t \mid \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})}
\end{aligned}
$$

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

Courtesy: C. Stachniss

Static State Binary Bayes Filter

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

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

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

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

Do exactly the same for the opposite event:

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

Courtesy: C. Stachniss

Static State Binary Bayes Filter

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

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

  - Note how this eliminates difficult to come by quantities

Courtesy: C. Stachniss

Static State Binary Bayes Filter

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

$$
\begin{aligned}
&\frac{p(m_i \mid \mathbf{z}_{1:t}, \mathbf{x}_{1:t})}{p(\neg m_i \mid \mathbf{z}_{1:t}, \mathbf{x}_{1:t})} \\
&\quad = \frac{p(m_i \mid \mathbf{z}_t, \mathbf{x}_t)\; p(m_i \mid \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-1})\; p(\neg m_i)}{p(\neg m_i \mid \mathbf{z}_t, \mathbf{x}_t)\; p(\neg m_i \mid \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-1})\; p(m_i)} \\
&\quad = \frac{p(m_i \mid \mathbf{z}_t, \mathbf{x}_t)}{1 - p(m_i \mid \mathbf{z}_t, \mathbf{x}_t)} \; \frac{p(m_i \mid \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-1})}{1 - p(m_i \mid \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-1})} \; \frac{1 - p(m_i)}{p(m_i)}
\end{aligned}
$$

Courtesy: C. Stachniss

Static State Binary Bayes Filter

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

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

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

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

Courtesy: C. Stachniss

From Ratio to Probability

- We can easily turn the ratio into the probability

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

Courtesy: C. Stachniss

From Ratio to Probability

- We can easily turn the ratio into the probability

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

Courtesy: C. Stachniss

From Ratio to Probability

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

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

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

Courtesy: C. Stachniss

Log Odds Notation

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

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

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

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

Courtesy: C. Stachniss

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 $p(x)$

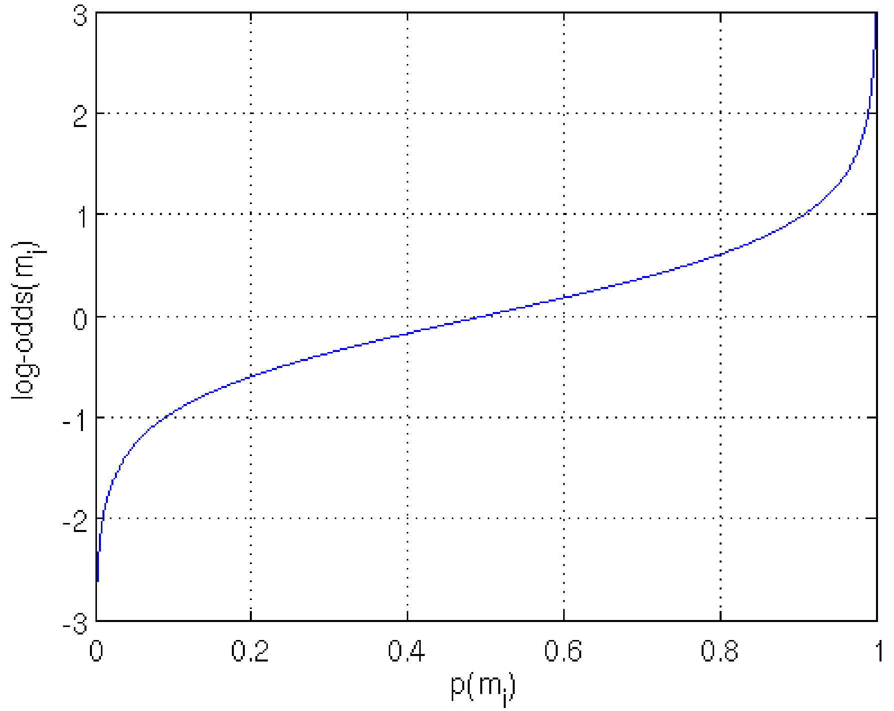
$$p(x) \quad = \quad 1 - \frac{1}{1 + \exp l(x)}$$

Courtesy: C. Stachniss

Why Log-Odds form?

- Computationally elegant for updating beliefs in log-odds form because updates are additive and avoids truncation problems that arise for probabilities close to 0 or 1

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



Courtesy: C. Stachniss

Occupancy Mapping in Log Odds Form

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

- The product 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 in short

$$l_{t,i} = \text{inv\_sensor\_model}(m_i, \mathbf{x}_t, \mathbf{z}_t) + l_{t-1,i} - l_0$$

Courtesy: C. Stachniss

Occupancy Mapping Algorithm

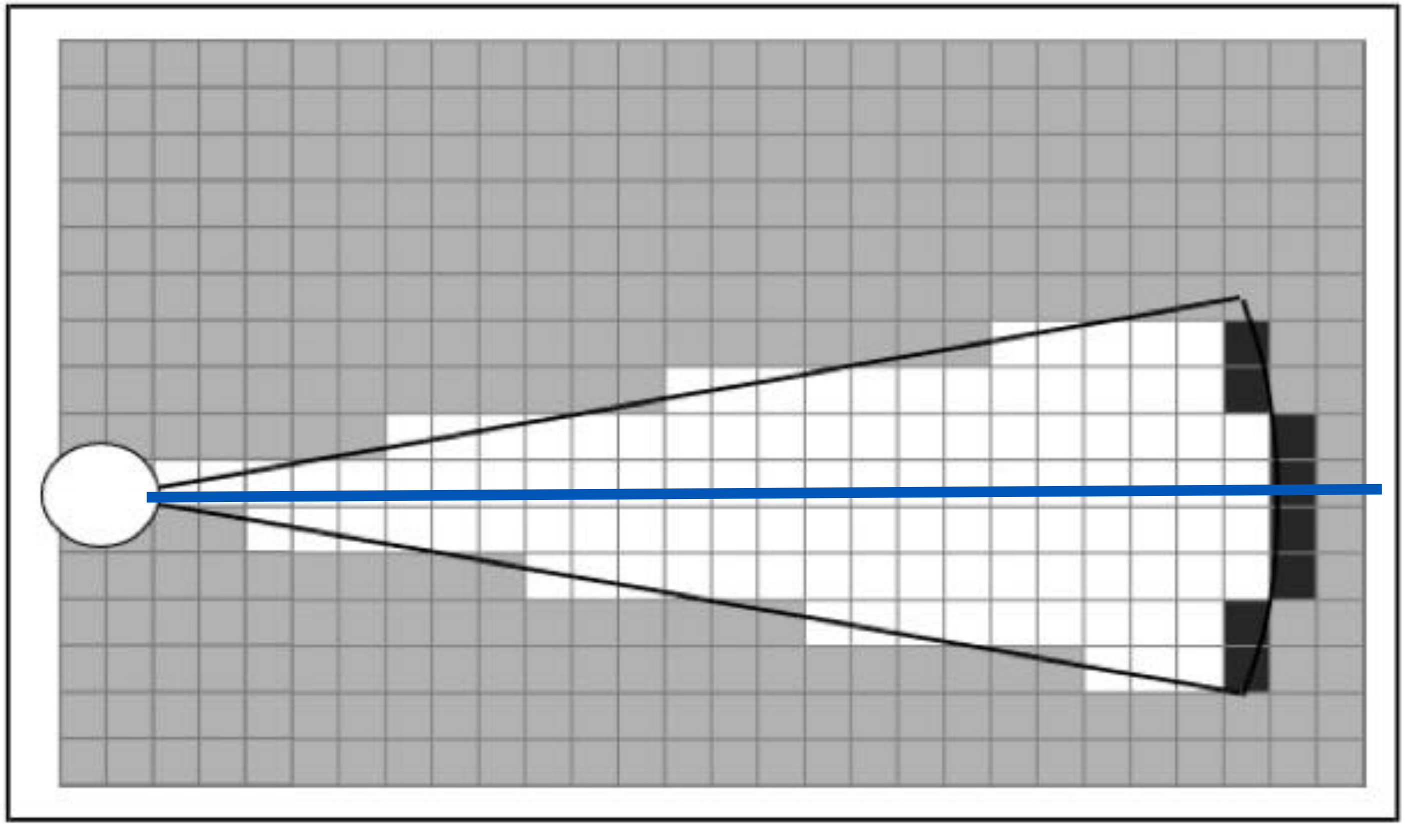
occupancy_grid_mapping($\{l_{t-1,i}\}, \mathbf{x}_t, \mathbf{z}_t$):

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

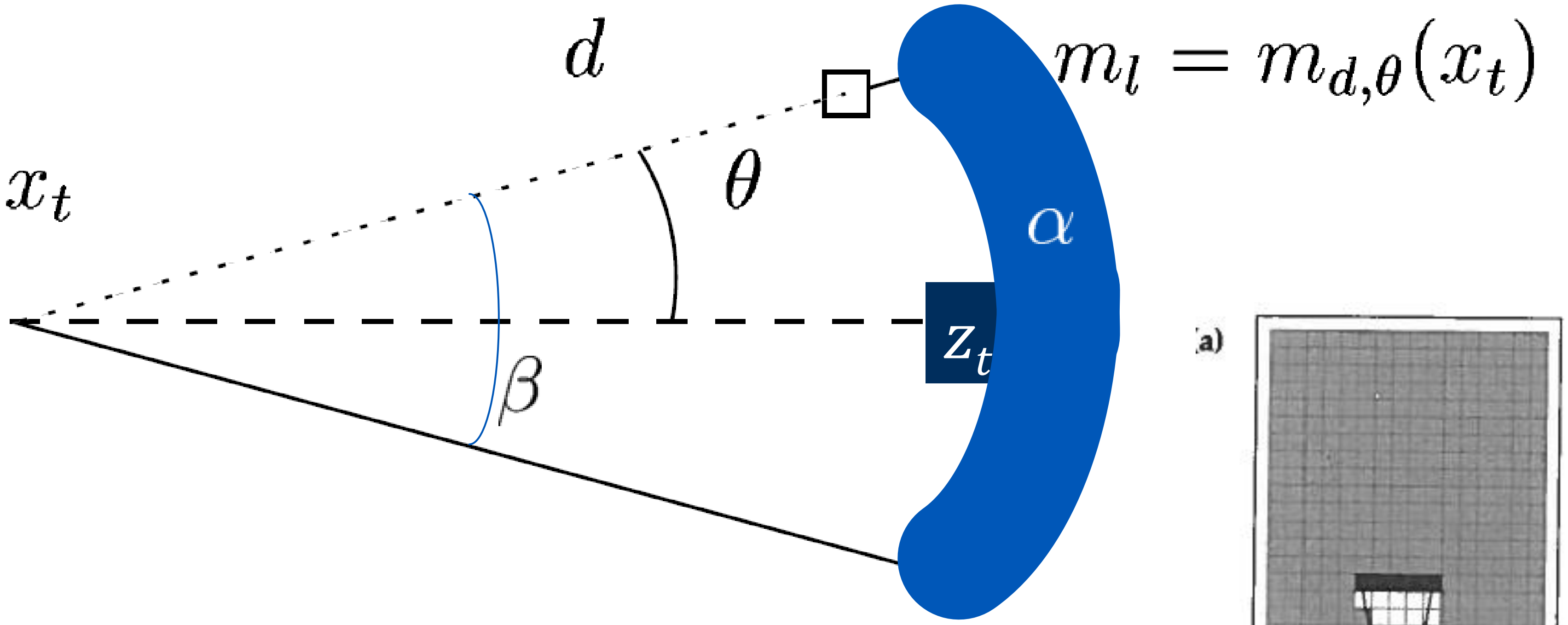
Courtesy: C. Stachniss

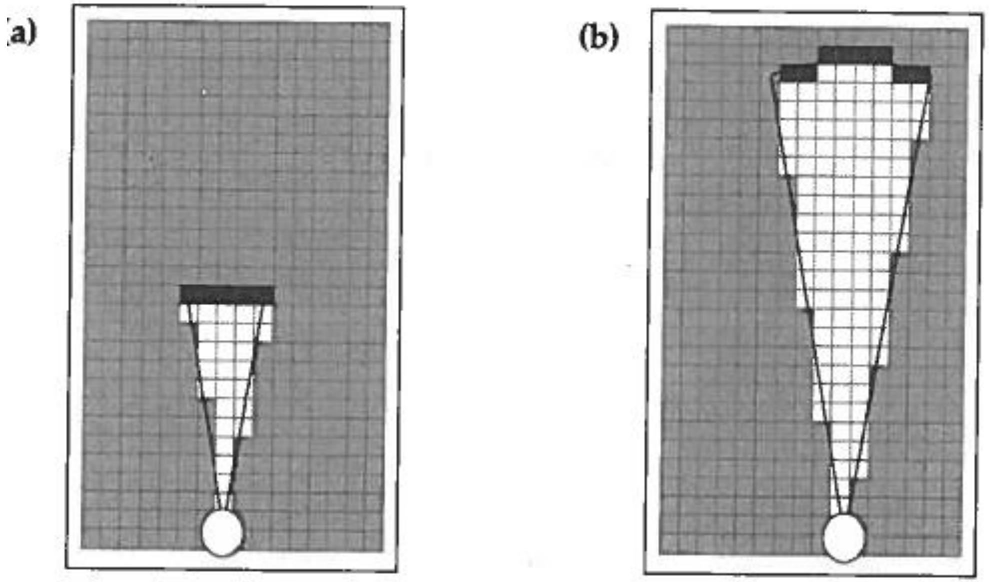
Inverse Sensor Model for Sonar Range Sensors



Courtesy: C. Stachniss

Example of a (Crude) Inverse Sensor Model





Courtesy: C. Stachniss

(Crude) Inverse Sensor Model

```
1:  Algorithm inverse_range_sensor_model(m_i, x_t, z_t):
2:      Let x_i, y_i be the center-of-mass of m_i
3:      r = sqrt((x_i - x)^2 + (y_i - y)^2)
4:      φ = atan2(y_i - y, x_i - x) - θ
5:      k = argmin_j |φ - θ_{j,sens}|
6:      if r > min(z_max, z_t^k + α/2) or |φ - θ_{k,sens}| > β/2 then
7:          return l_0
8:      if z_t^k < z_max and |r - z_t^k| < α/2
9:          return l_occ
10:     if r ≤ z_t^k
11:         return l_free
12:     endif
```

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

2: *Let* x_i, y_i *be the center-of-mass of* $\mathbf{m}_i$

3: $r = \sqrt{(x_i - x)^2 + (y_i - y)^2}$

4: $\phi = \text{atan2}(y_i - y, x_i - x) - \theta$

5: $k = \text{argmin}_j \, |\phi - \theta_{j,\text{sens}}|$

6: *if* $r > \min(z_{\max}, z_t^k + \alpha/2)$ *or* $|\phi - \theta_{k,\text{sens}}| > \beta/2$ *then*

7: *return* l_0

8: *if* $z_t^k < z_{\max}$ *and* $|r - z_t^k| < \alpha/2$

9: *return* l_{occ}

10: *if* $r \leq z_t^k$

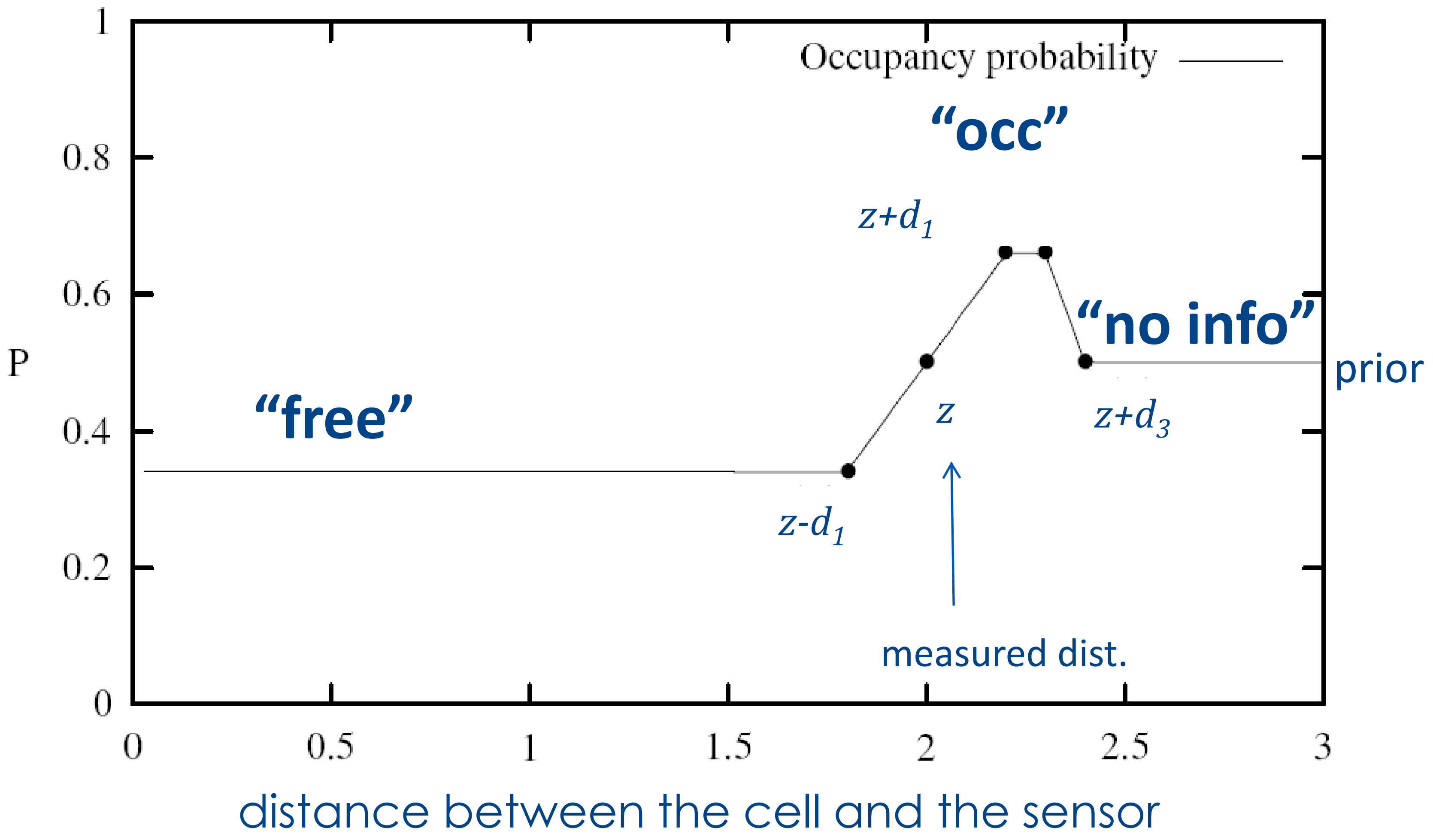
11: *return* l_{free}

12: *endif*

Table 9.2 A simple inverse measurement model for robots equipped with range finders. Here α is the thickness of obstacles, and β the width of a sensor beam. The values l_{occ} and l_{free} in lines 9 and 11 denote the amount of evidence a reading carries for the two different cases.

Courtesy: C. Stachniss

Occupancy Value Depending on the Measured Distance

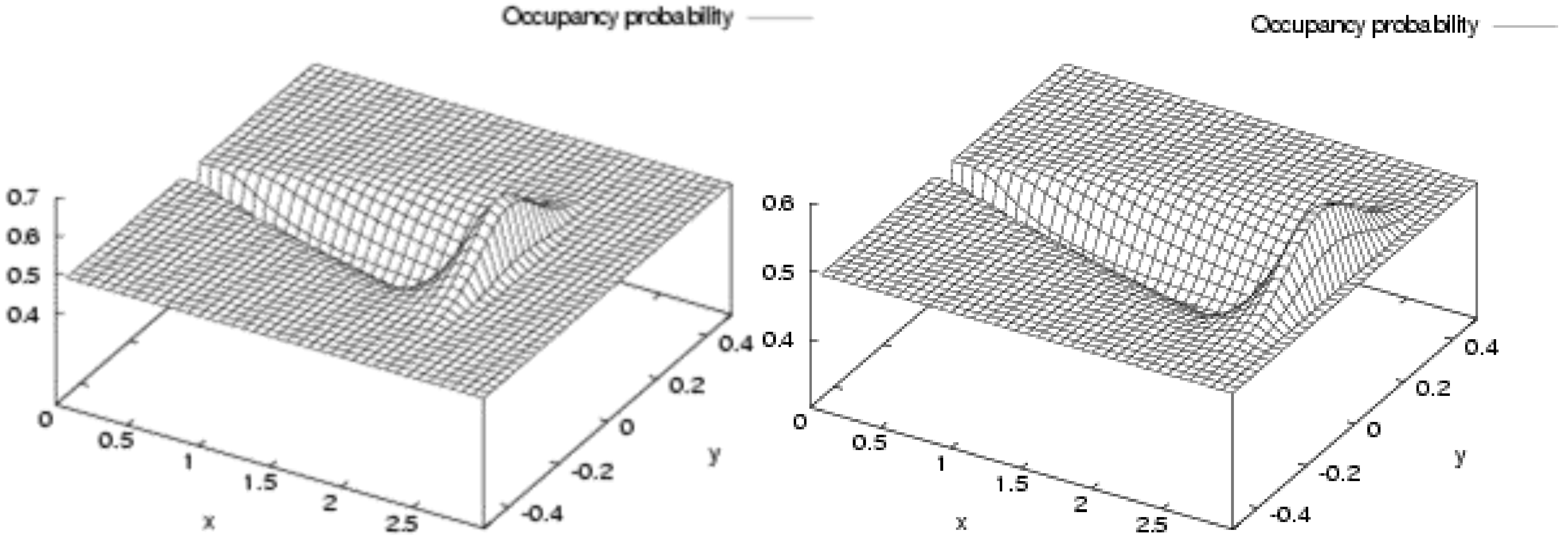


Courtesy: C. Stachniss

Typical Sensor Model for Occupancy Grid Maps

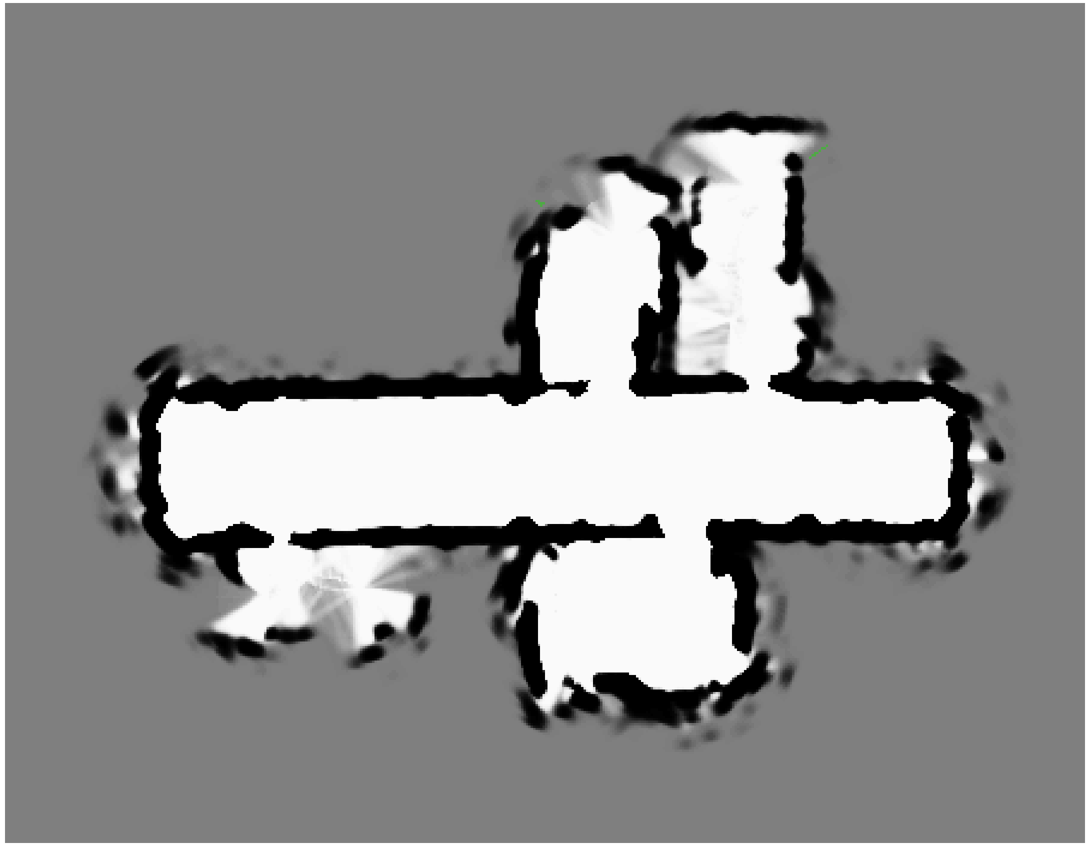
Chap 6 ProbRob

Combination of a linear function and a Gaussian:



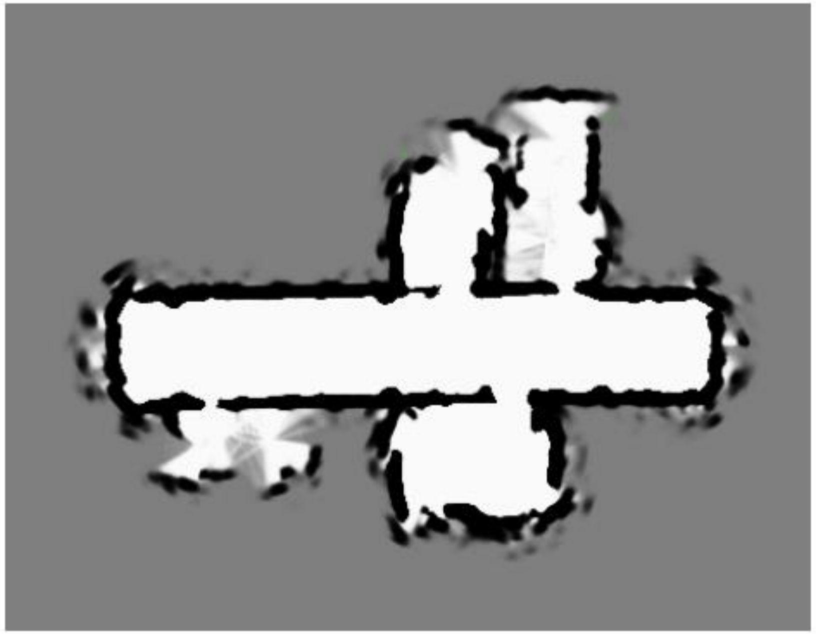
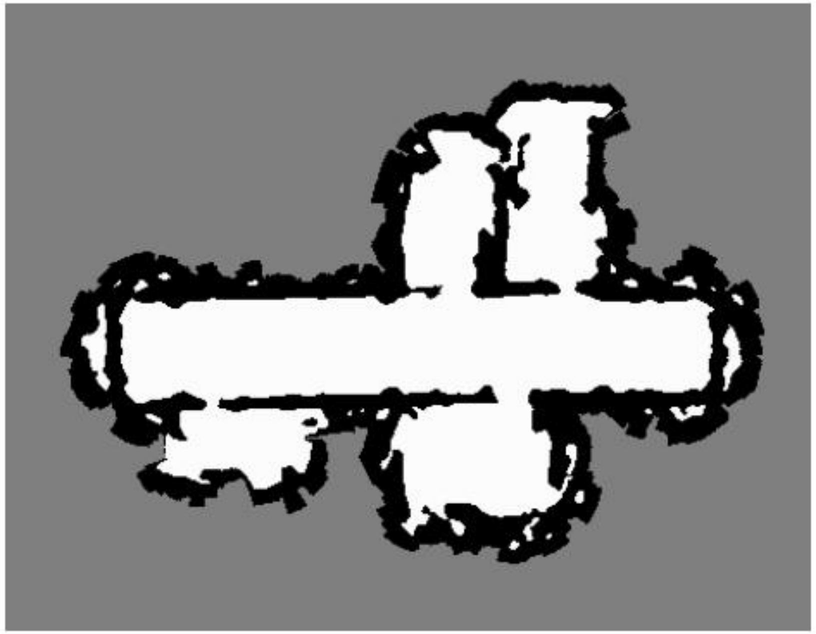
Courtesy: C. Stachniss

Resulting Map Obtained with 24 Sonar Range Sensors



Courtesy: C. Stachniss

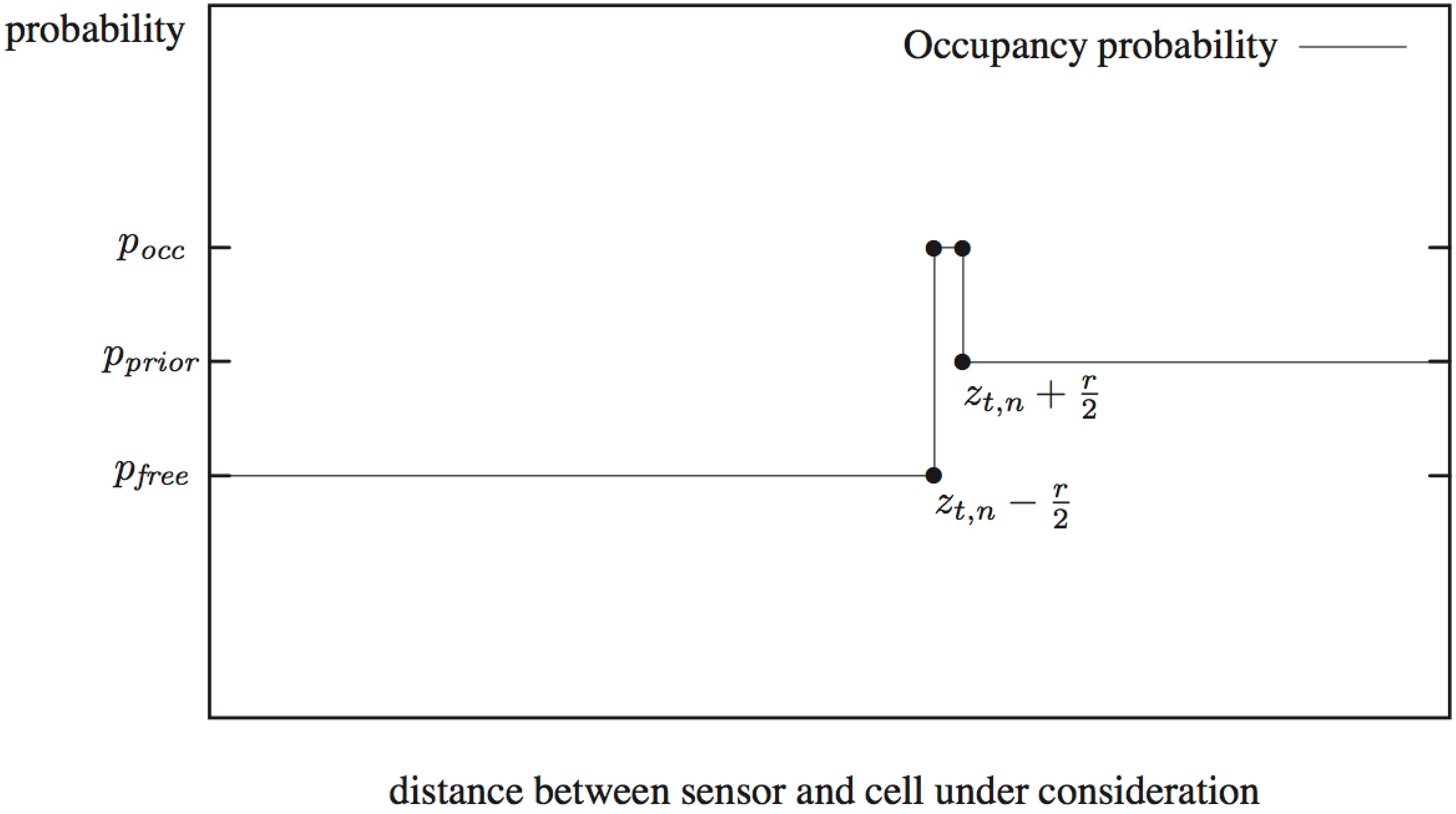
Resulting Occupancy and Maximum Likelihood Map

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

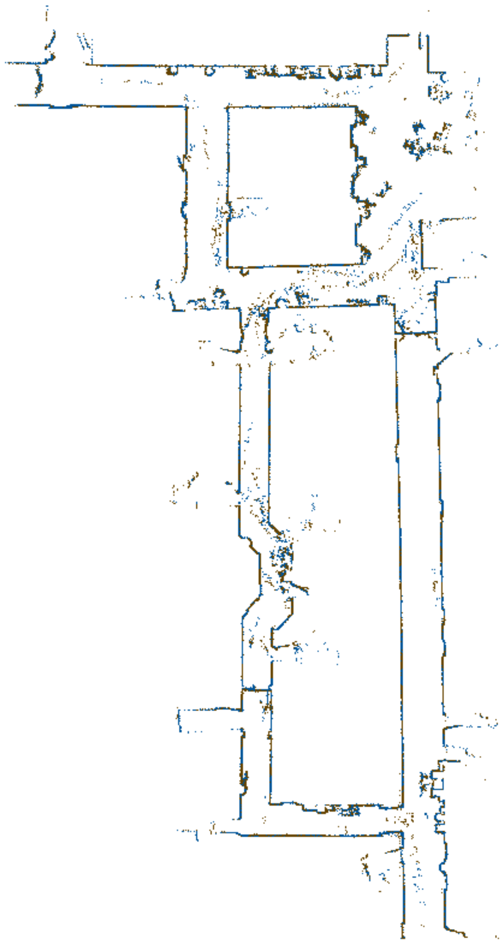
Courtesy: C. Stachniss

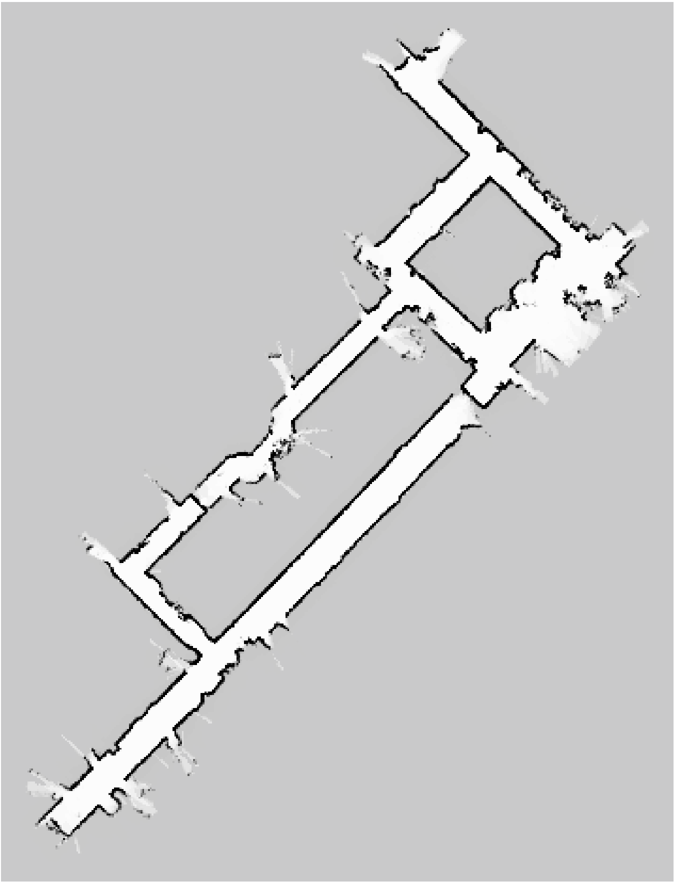
Inverse Sensor Model for Laser Range Finders



Courtesy: C. Stachniss

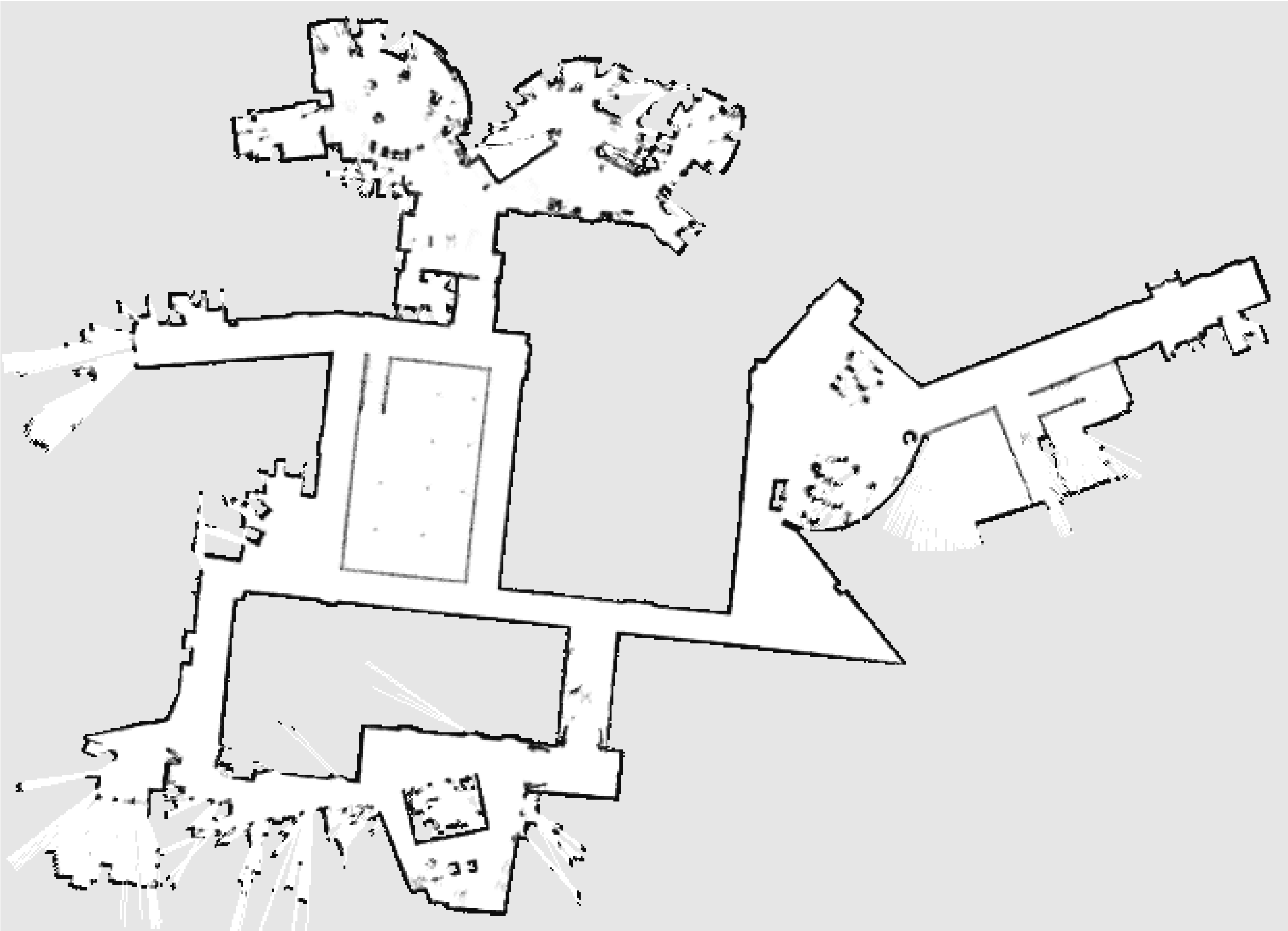
Occupancy Grids: From Laser Scans to Maps





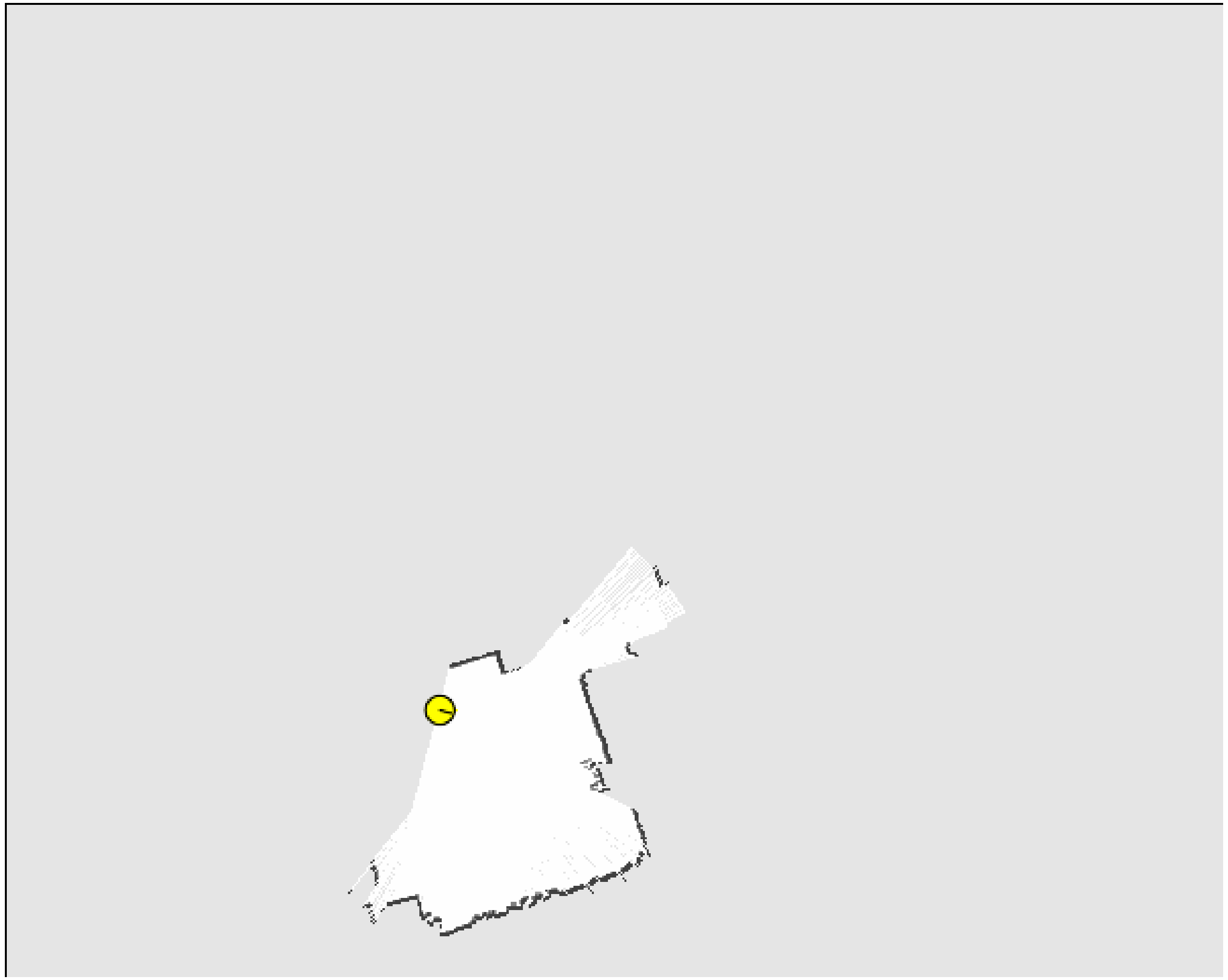
Courtesy: C. Stachniss

Example: MIT CSAIL 3rd Floor



Courtesy: C. Stachniss

Uni Freiburg Building 106



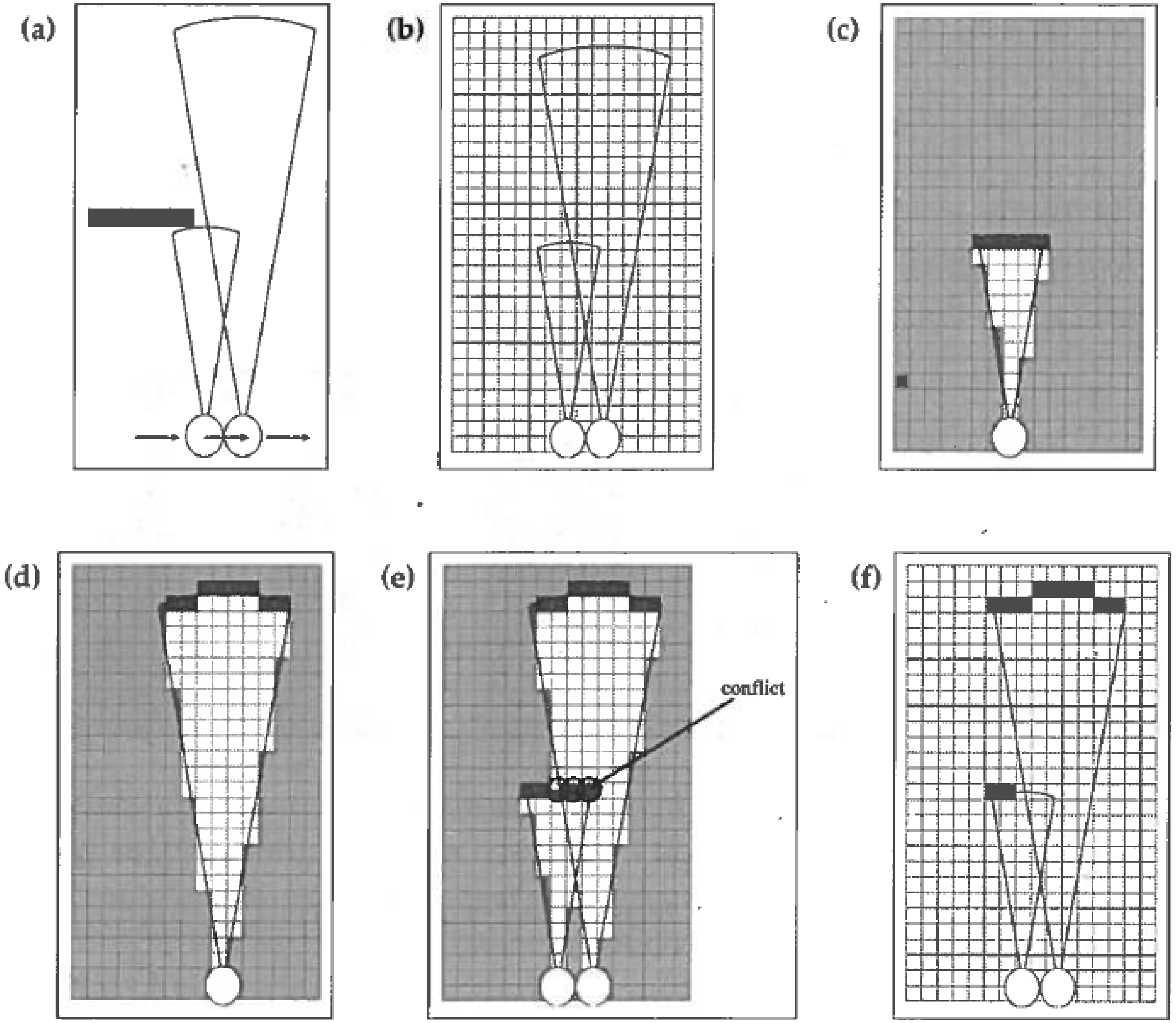
Courtesy: C. Stachniss

Weaknesses?

- Weakness of Independence Assumption

- Assumes Known Poses

Weakness of the Independence Assumption



Remedy by Recovering the Mode of the Posterior - but lose measure of probability

Algorithm MAP_occupancy_grid_mapping($x_{1:t}, z_{1:t}$):
  set $m = \{0\}$
  repeat until convergence
    for all cells $\mathbf{m}_i$ *do*
      $$m_i = \underset{k=0,1}{\operatorname{argmax}} \; k\, l_0 + \sum_t \log \textbf{measurement\_model}(z_t, x_t, m \textit{ with } \mathbf{m}_i = k)$$
    endfor
  endrepeat
  return m

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

Grid Mapping meets reality…

Mapping with Raw Odometry

Courtesy: D. Hähnel

Occupancy Grid Map 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

Often run SLAM or some type of localization first and then run occupancy grid mapping on top.

Courtesy: C. Stachniss