Carnegie Mellon
THE ROBOTICS INSTITUTE

# Occupancy Grids

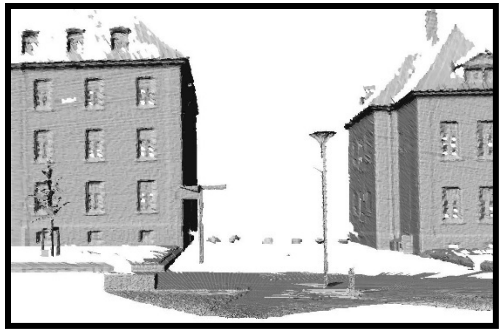
## Robot Localization and Mapping
## 16-833

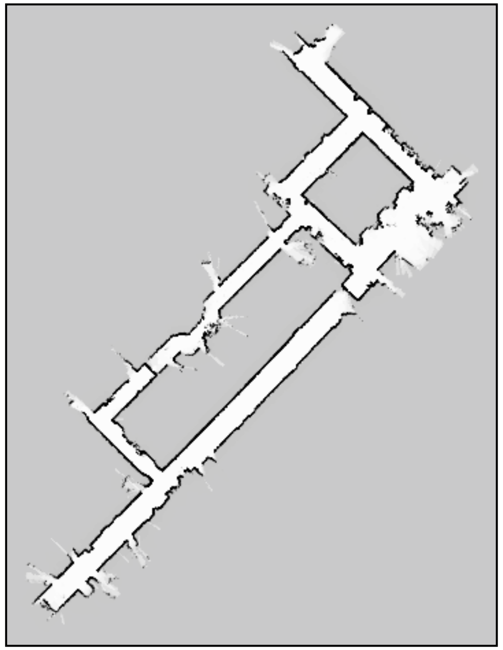
Michael Kaess

November 25, 2024

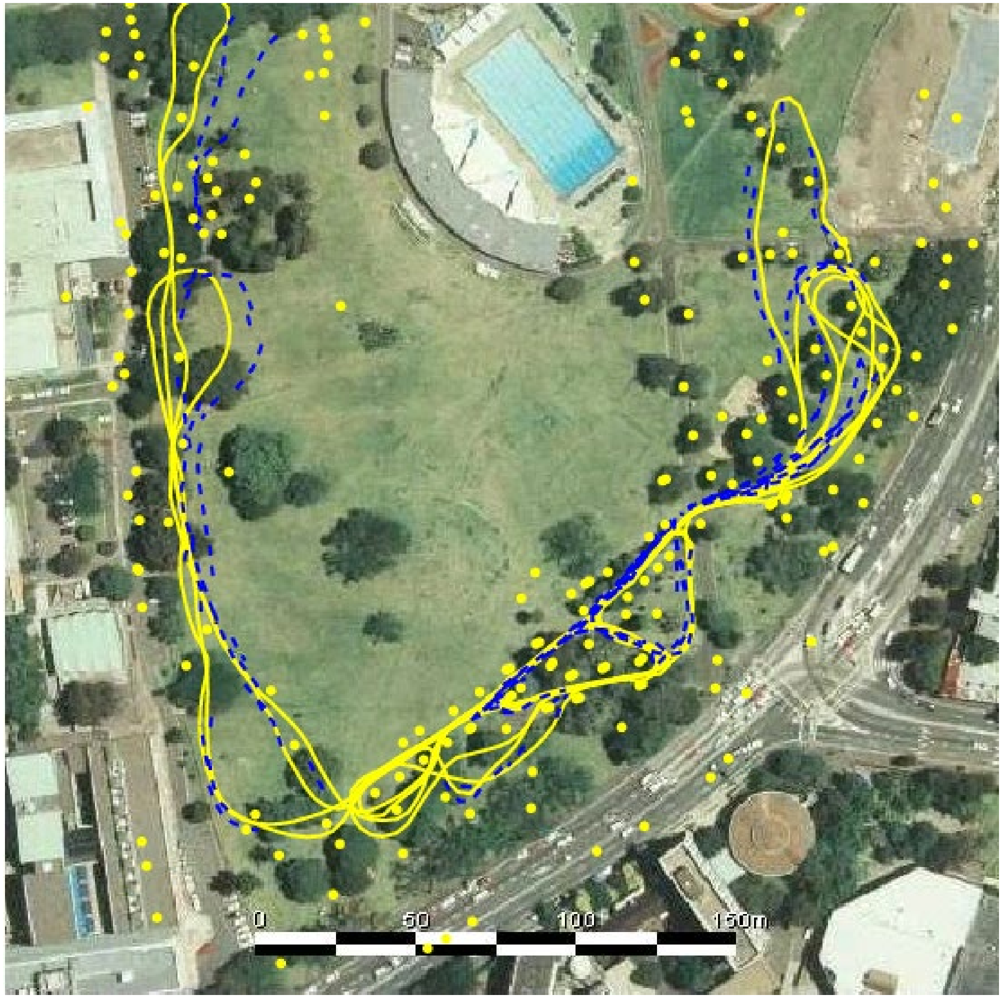
Slides courtesy of Ryan Eustice

# Volumetric Maps vs Features





Courtesy: D. Hähnel




Courtesy: E. Nebot

# Features

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

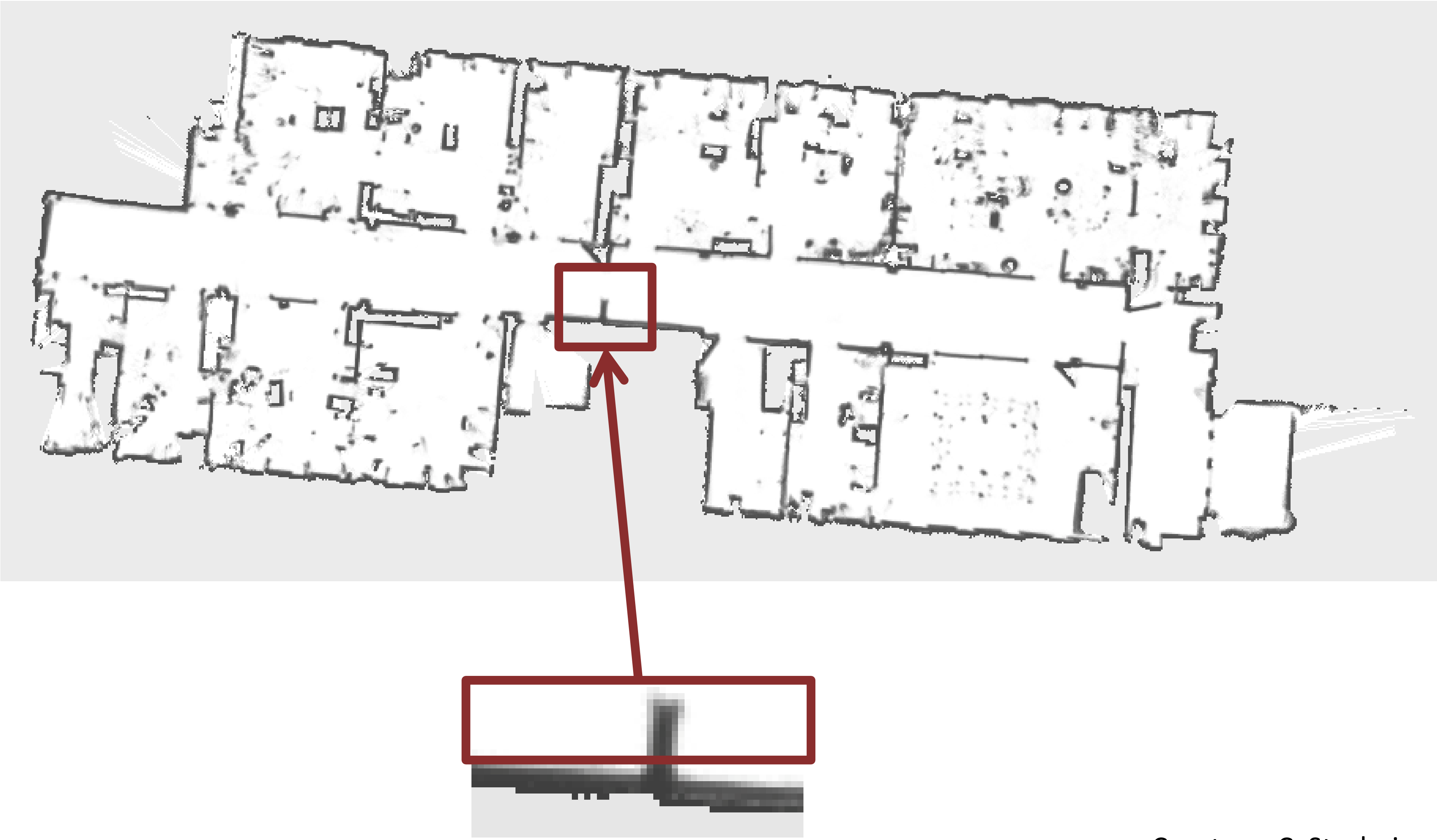
Courtesy: C. Stachniss

# 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

Courtesy: C. Stachniss

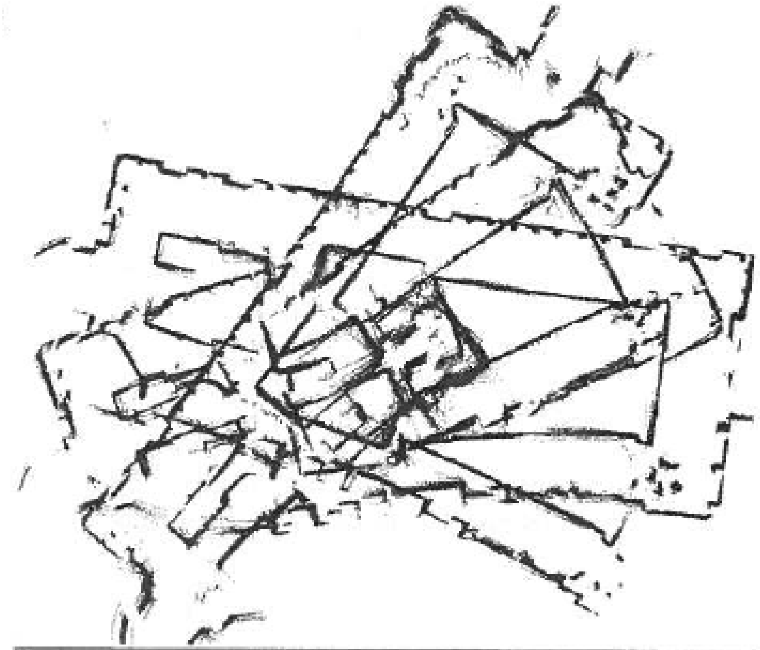
# Example

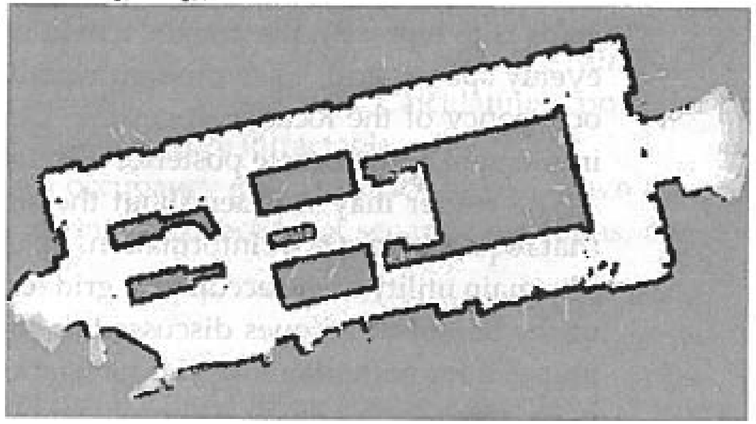


Courtesy: C. Stachniss

# 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

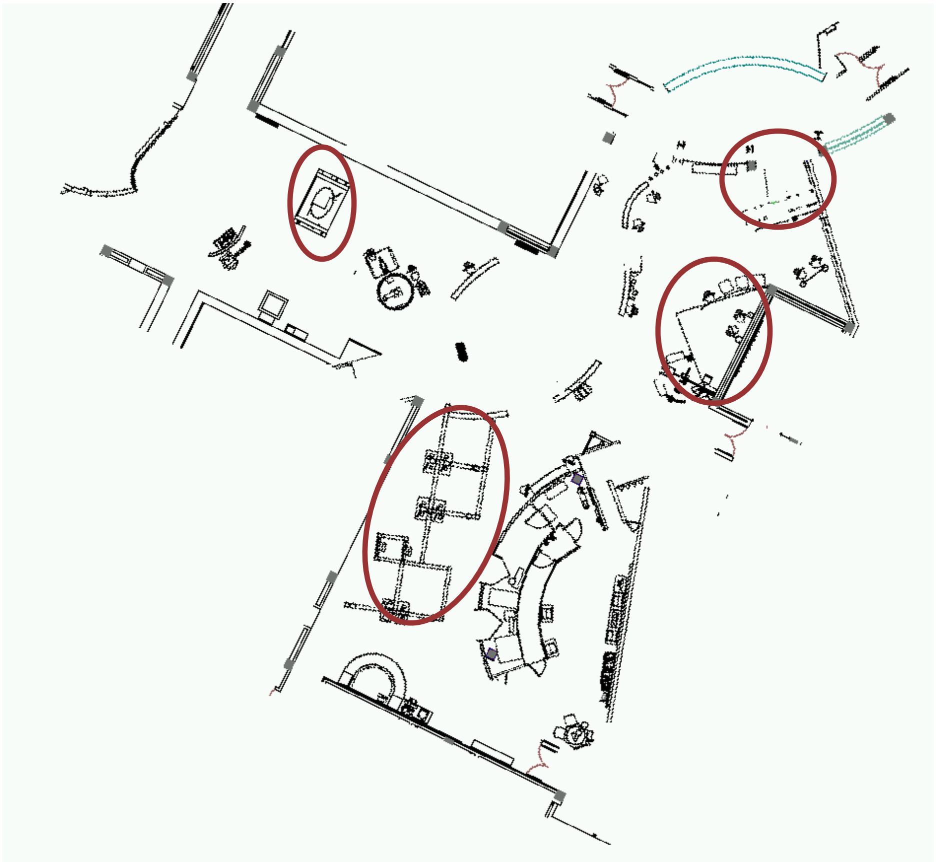




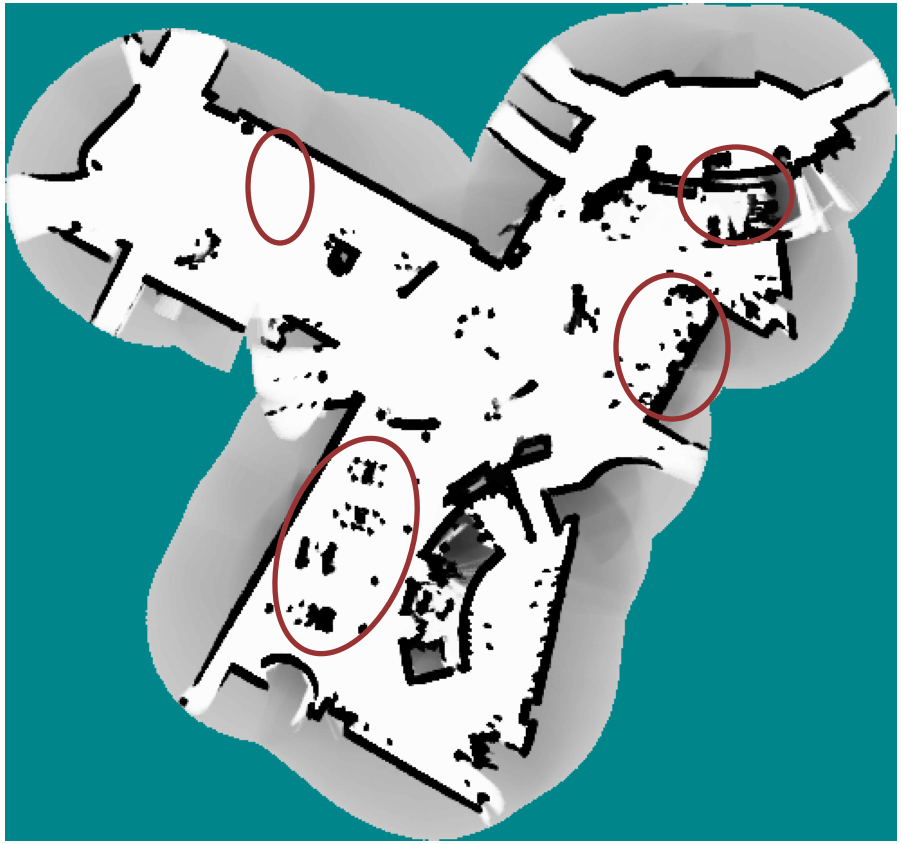
# Old Algorithms – New Technology

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

# Even when we have prior maps, they can be inaccurate…



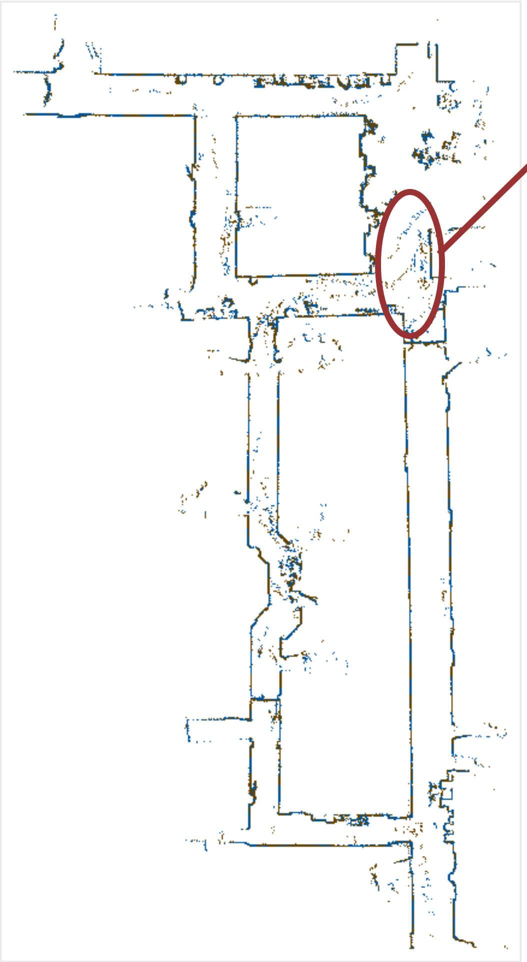
**CAD map**



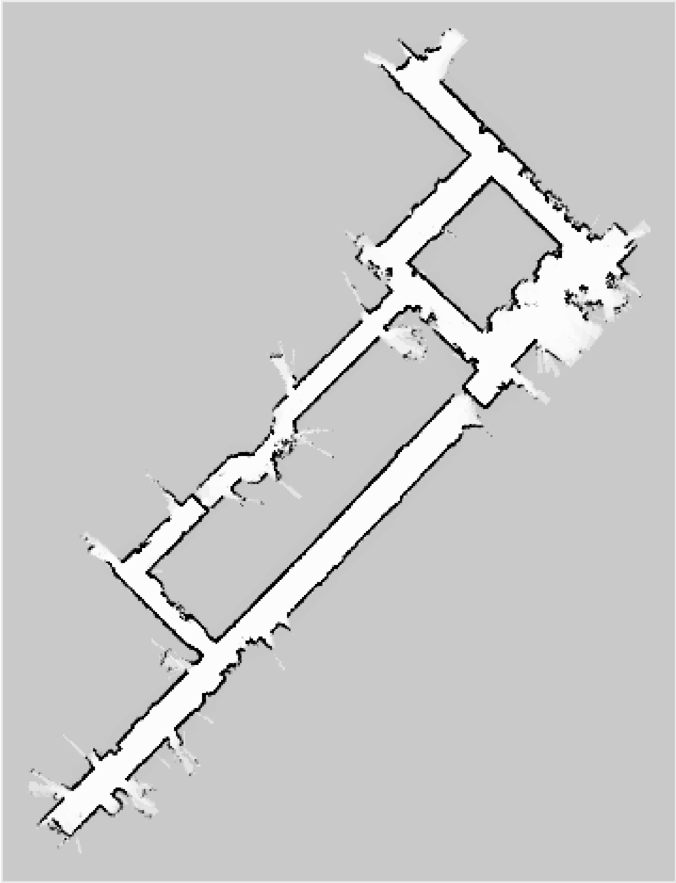
**occupancy grid map**

Tech Museum, San Jose

# Occupancy Grids: From scans to maps



False “hits” from people in the environment



# Occupancy Grid Mapping

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

Courtesy: C. Stachniss

# Occupancy Grid Maps (OGM)

- 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.

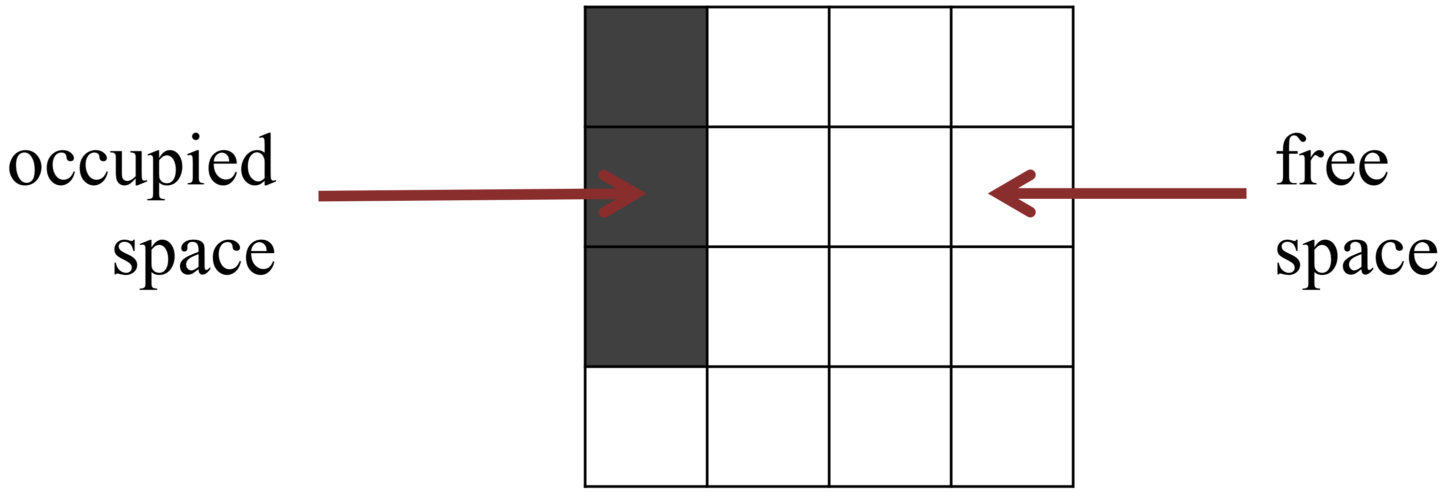
  Binary state: $\mathbf{m}_i = \begin{cases} 1 & \text{occupied} \\ 0 & \text{free} \end{cases}$

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

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

# Assumption 1

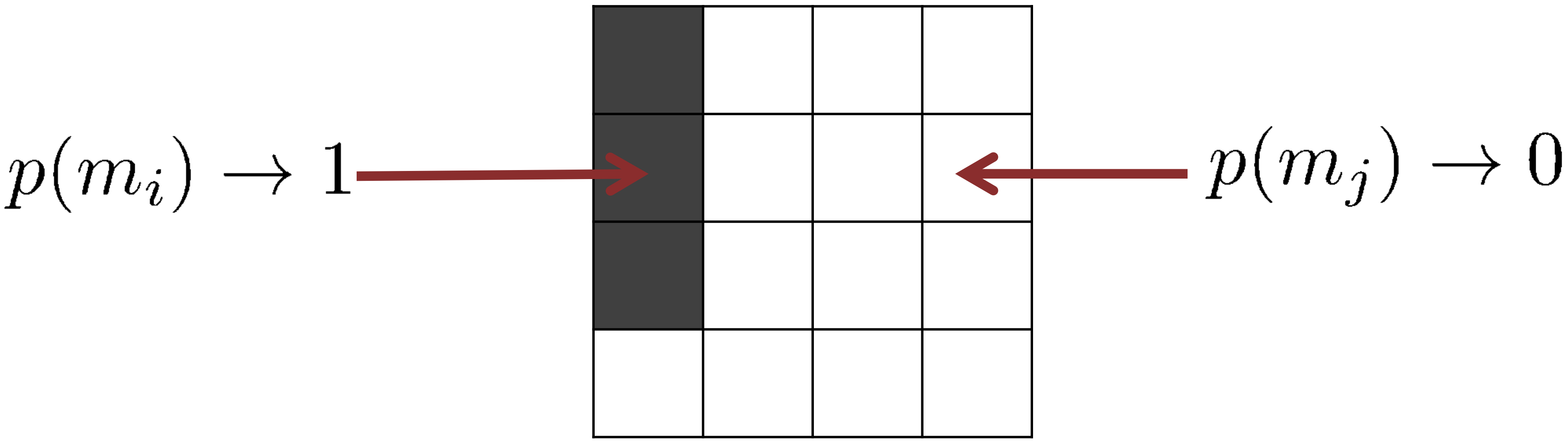
- The area that corresponds to a cell is either completely free or occupied




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:
  - Cell is not occupied:
  - No knowledge:

$$p(m_i) = 1$$

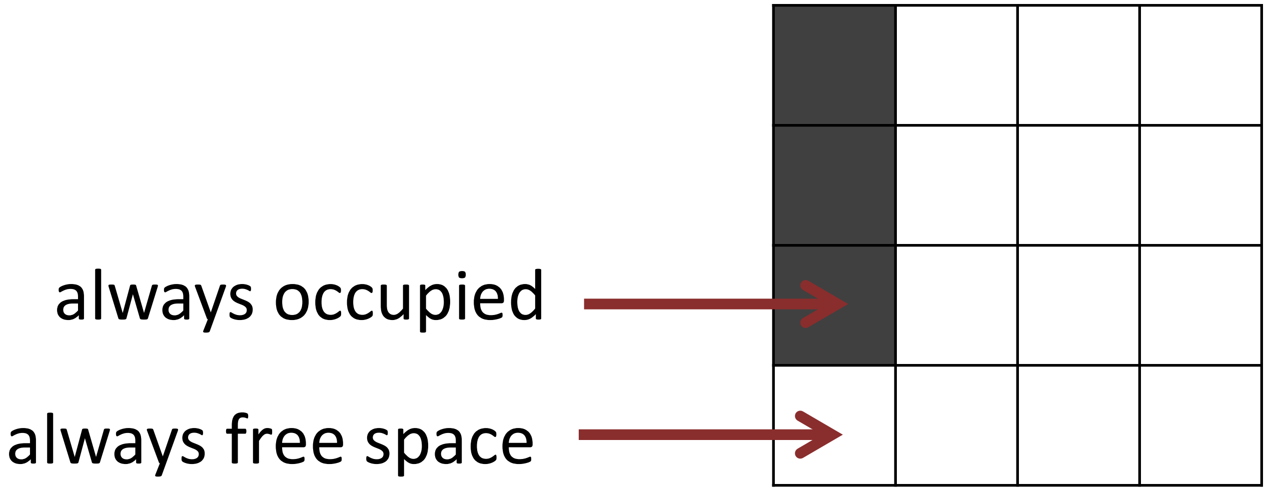
$$p(m_i) = 0$$

$$p(m_i) = 0.5$$

Courtesy: C. Stachniss

# Assumption 2

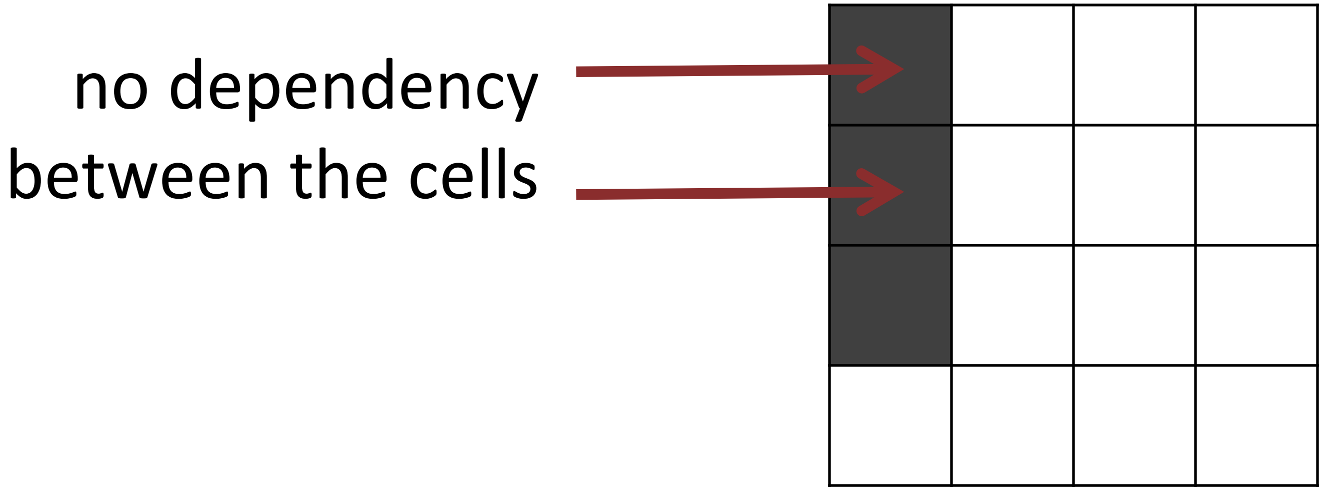
- The world is **static** (most mapping systems make this assumption)




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

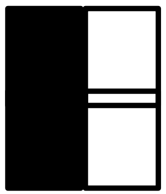
$$\underbrace{p(m)}_{\text{map}} = \prod_i \underbrace{p(m_i)}_{\text{cell}}$$

Courtesy: C. Stachniss

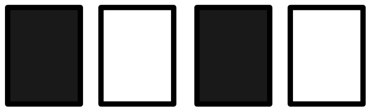
# Representation

- 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 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

⇨ Binary Bayes filter
(for a static state)

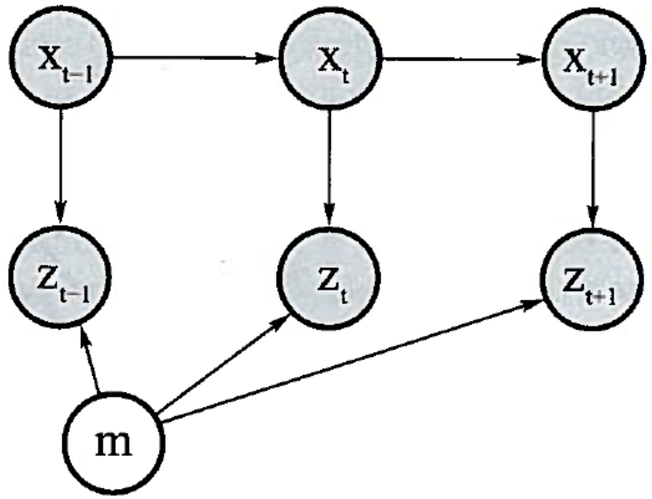
Courtesy: C. Stachniss

# Occupancy Grid Maps (OGM)

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

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

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



# Static State Binary Bayes Filter

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

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:

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

Inverse model

# Static State Binary Bayes Filter

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

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

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

Courtesy: C. Stachniss

# Static State Binary Bayes Filter

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

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!)*

# Static State Binary Bayes Filter

Courtesy: C. Stachniss

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

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

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

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

  - 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:

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

Courtesy: C. Stachniss

# Static State Binary Bayes Filter

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

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

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

Courtesy: C. Stachniss

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

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

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

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

# Log Odds Notation

- Log odds ratio is defined as

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

- and with the ability to retrieve $p(x)$

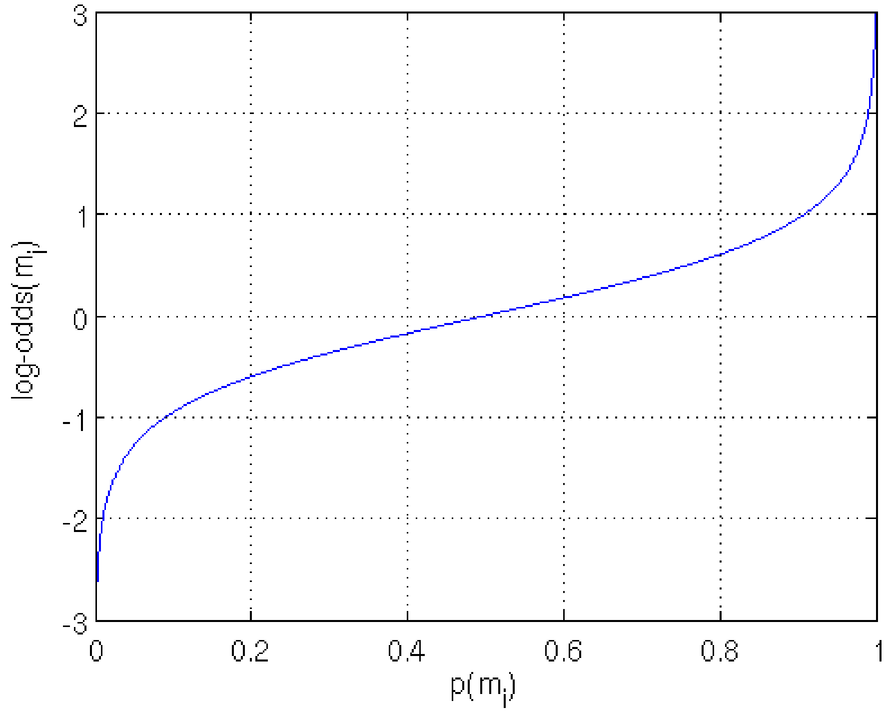
$$p(x) = 1 - \frac{1}{1 + \exp l(x)}$$

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

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



# 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

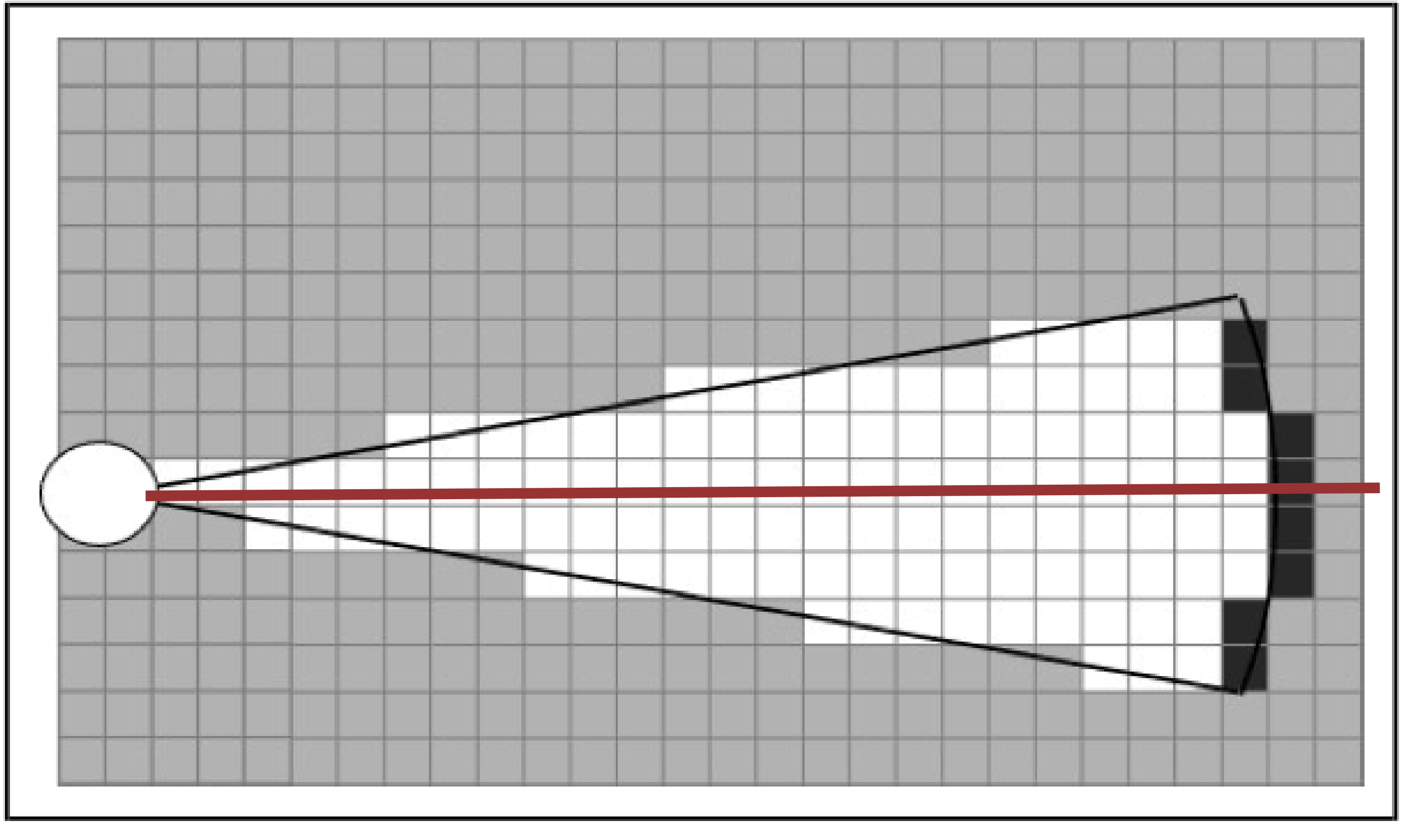
**occupancy_grid_mapping**$(\{l_{t-1,i}\}, \mathbf{x}_t, \mathbf{z}_t)$**:**

1: *for all cells* $m_i$ *do*
2: *if* $m_i$ *in perceptual field of* $\mathbf{z}_t$ *then*
3: $l_{t,i} = l_{t-1,i} + \text{inv\_sensor\_model}(m_i, \mathbf{x}_t, \mathbf{z}_t) - l_0$
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

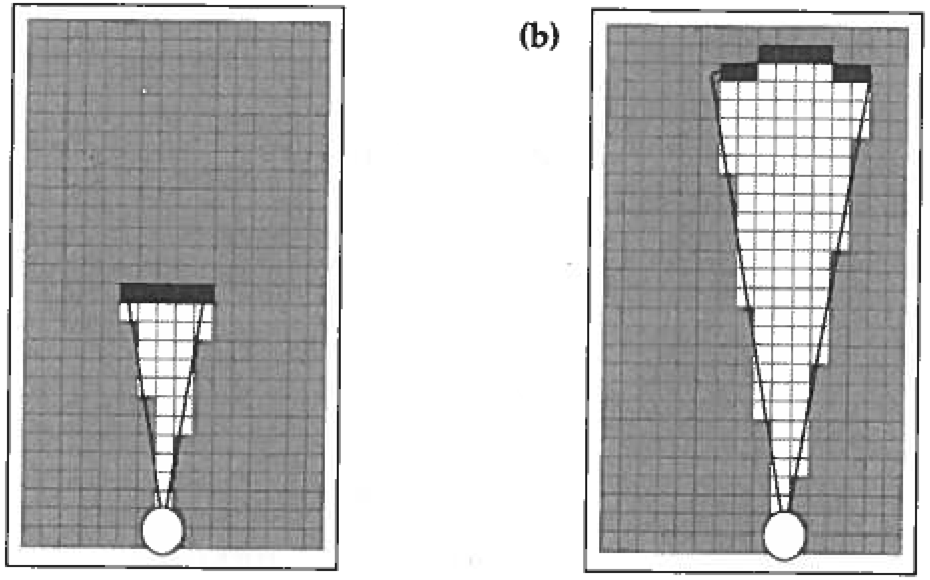


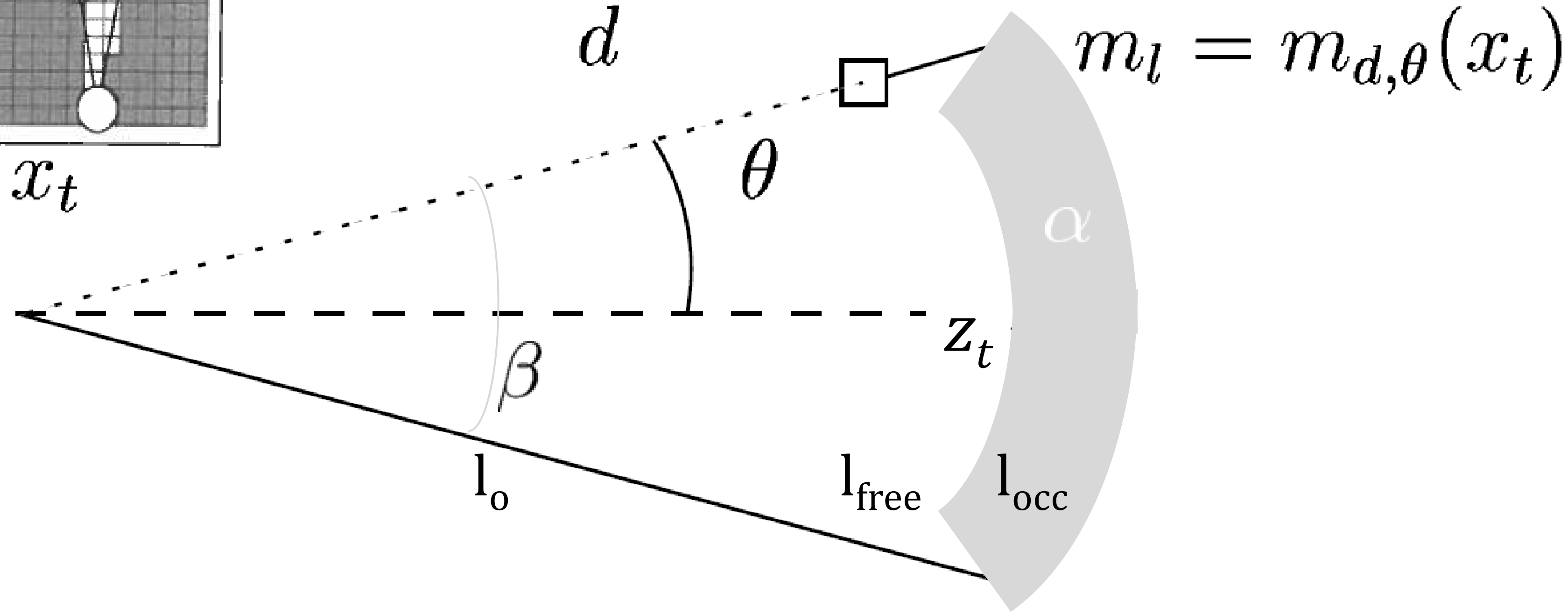
In the following, consider the cells along the optical axis (orange line)

Courtesy: Thrun, Burgard, Fox

# Example of a (Crude) Inverse Sensor Model

(a) (b)





# (Crude) Inverse Sensor Model

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* $l_{\text{occ}}$

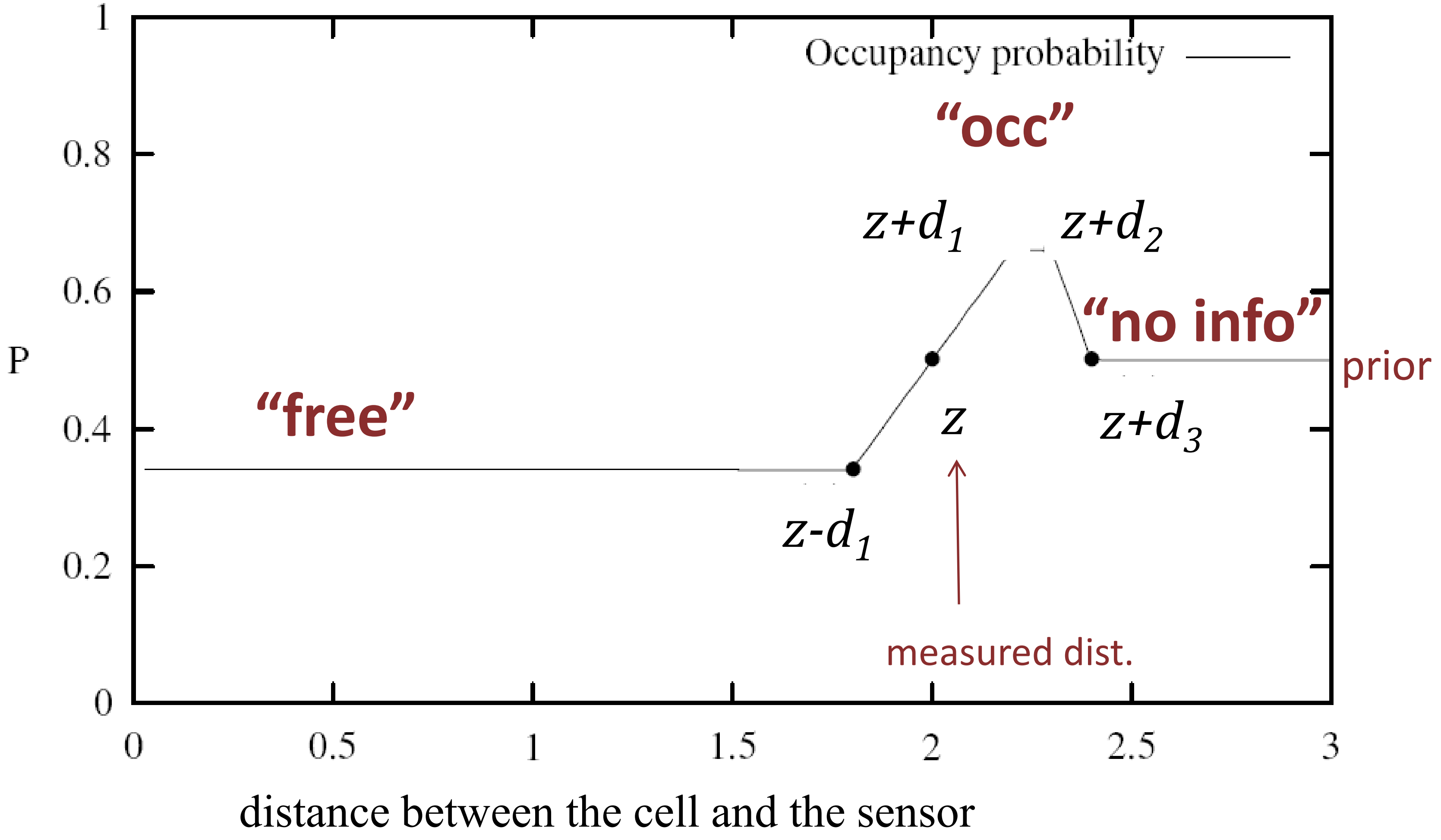
10: *if* $r \leq z_t^k$

11: *return* $l_{\text{free}}$

12: *endif*

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

# Occupancy Value Depending on the Measured Distance

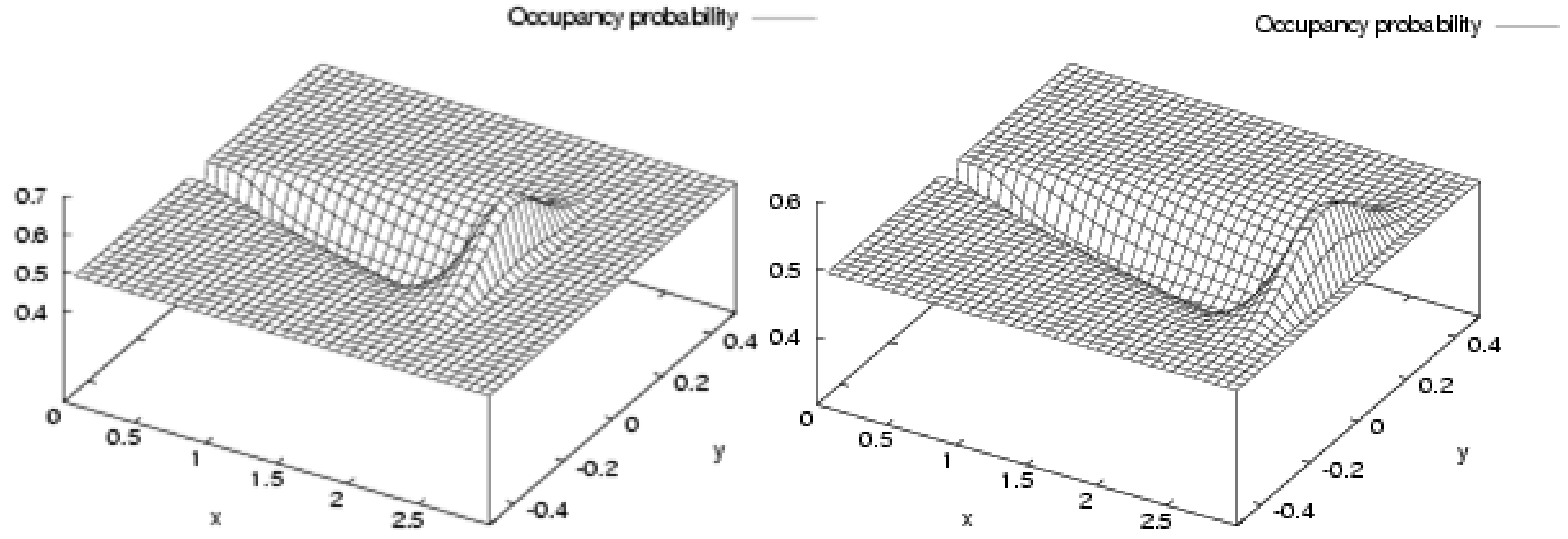



Courtesy: C. Stachniss

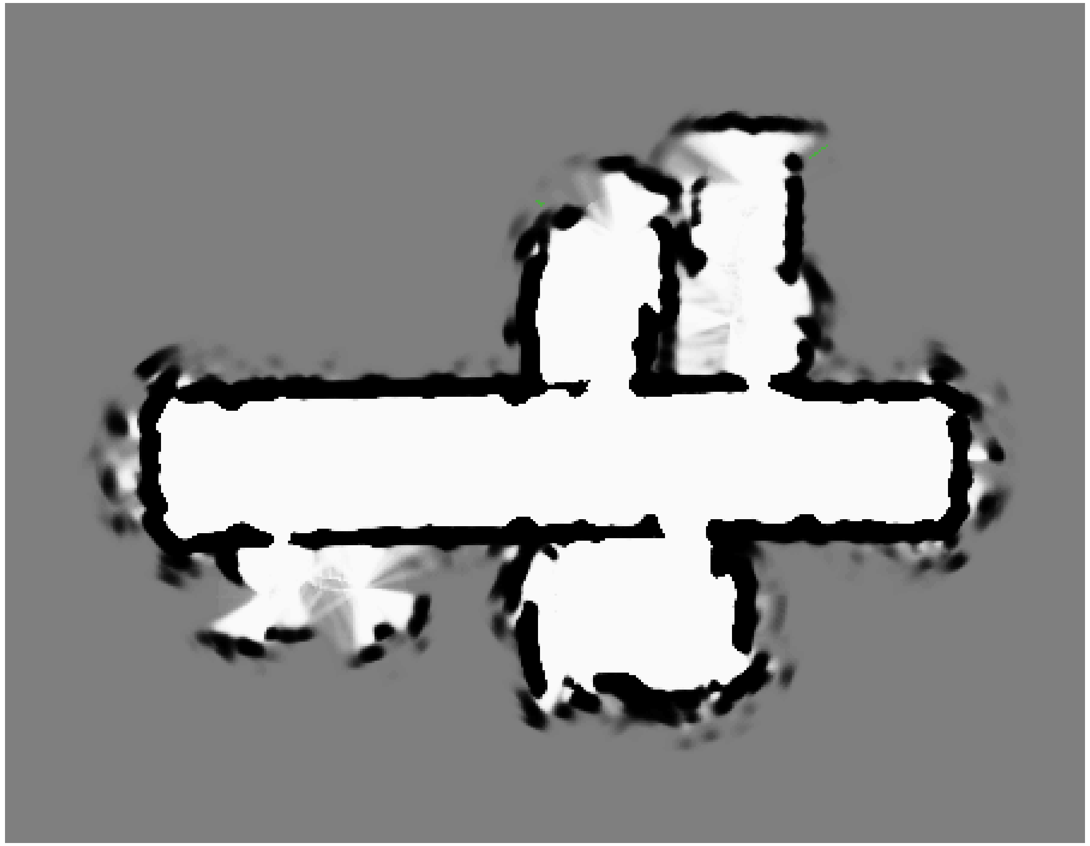
# Typical Sensor Model for Occupancy Grid Maps

Chap 6 ProbRob

Combination of a linear function and a Gaussian:

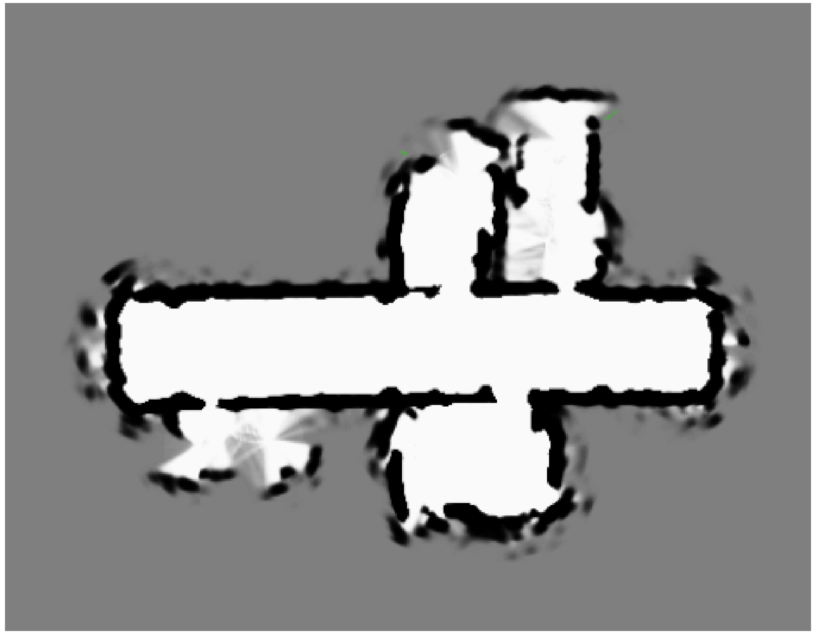


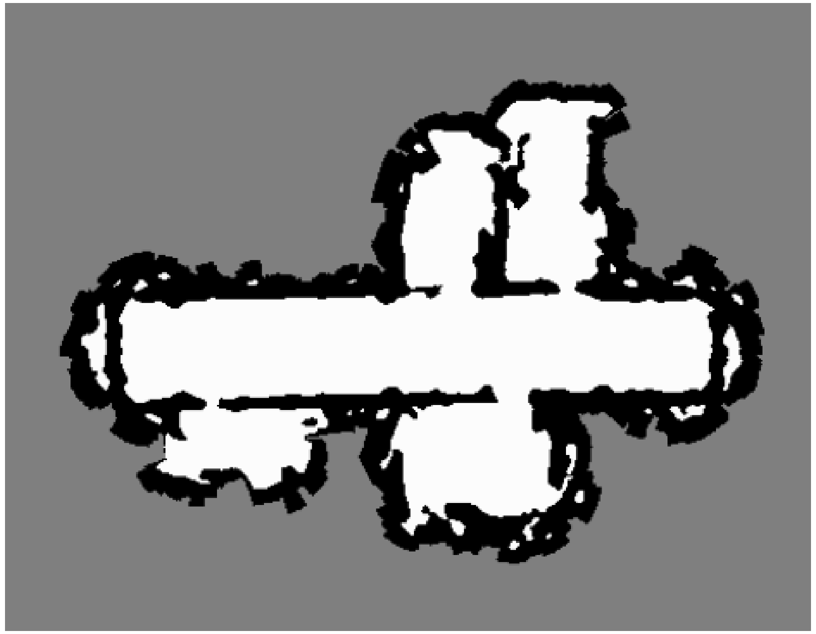
# Resulting Map Obtained with 24 Sonar Range Sensors



Courtesy: C. Stachniss

# Resulting Occupancy and Maximum Likelihood Map

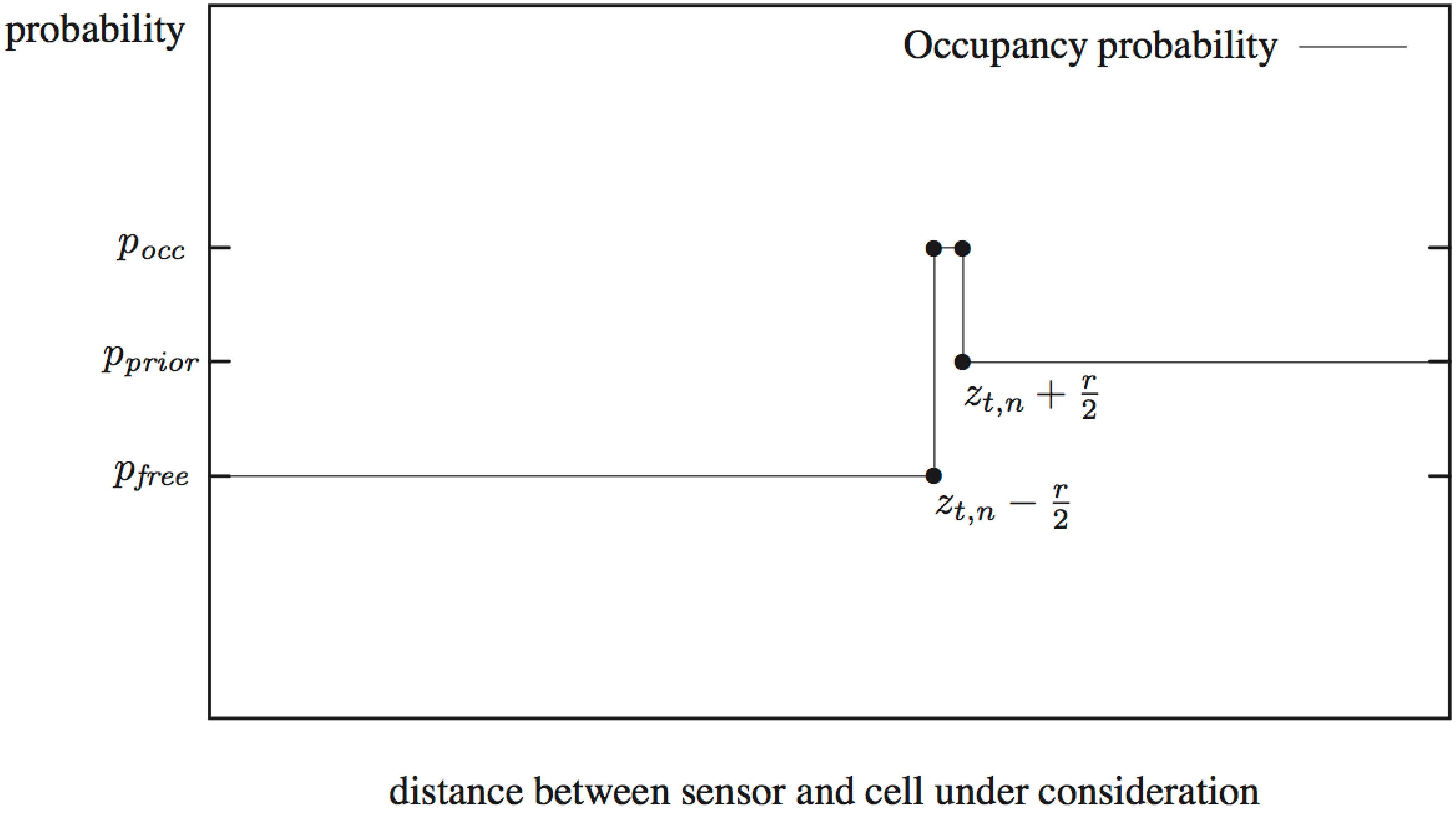




The maximum likelihood map is obtained by rounding the probability for each cell to 0 or 1.

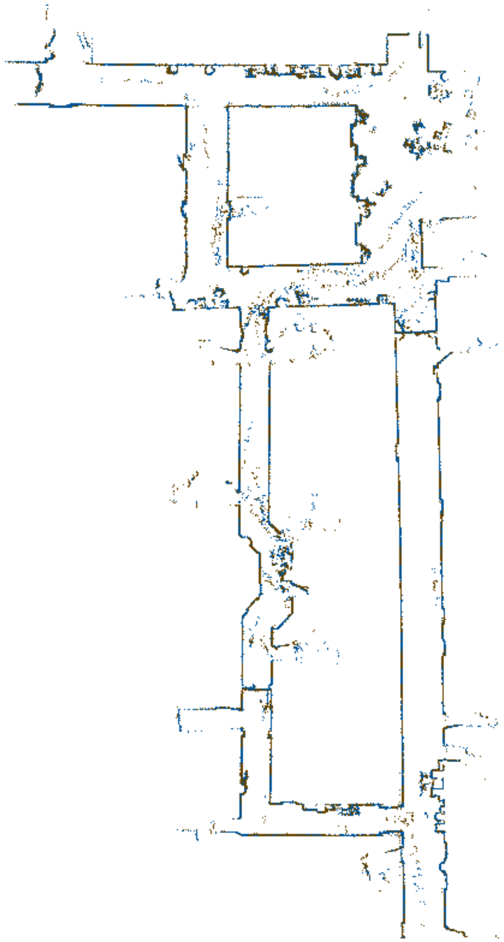
Courtesy: C. Stachniss

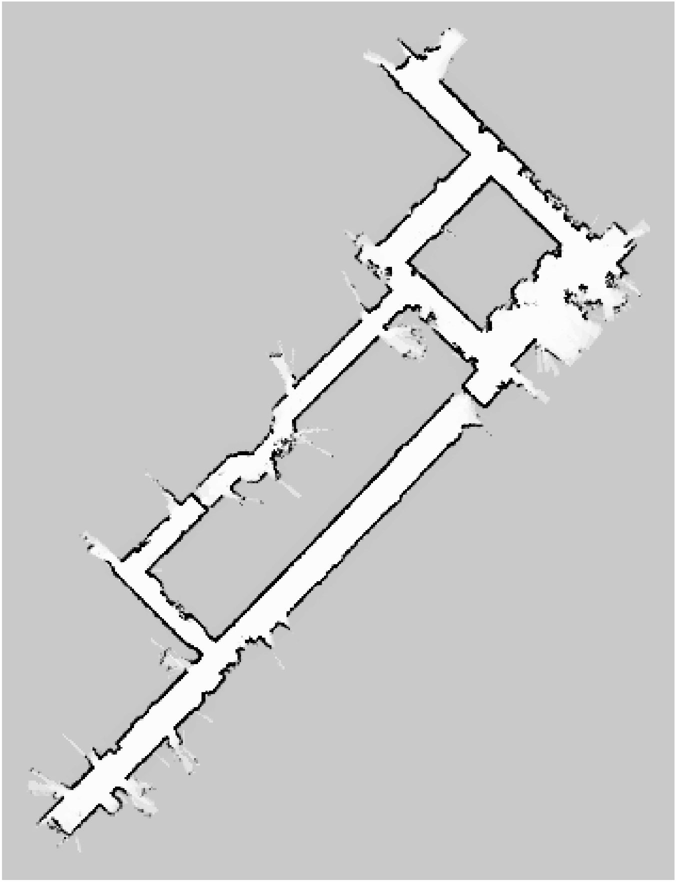
# Inverse Sensor Model for Laser Range Finders




Courtesy: C. Stachniss

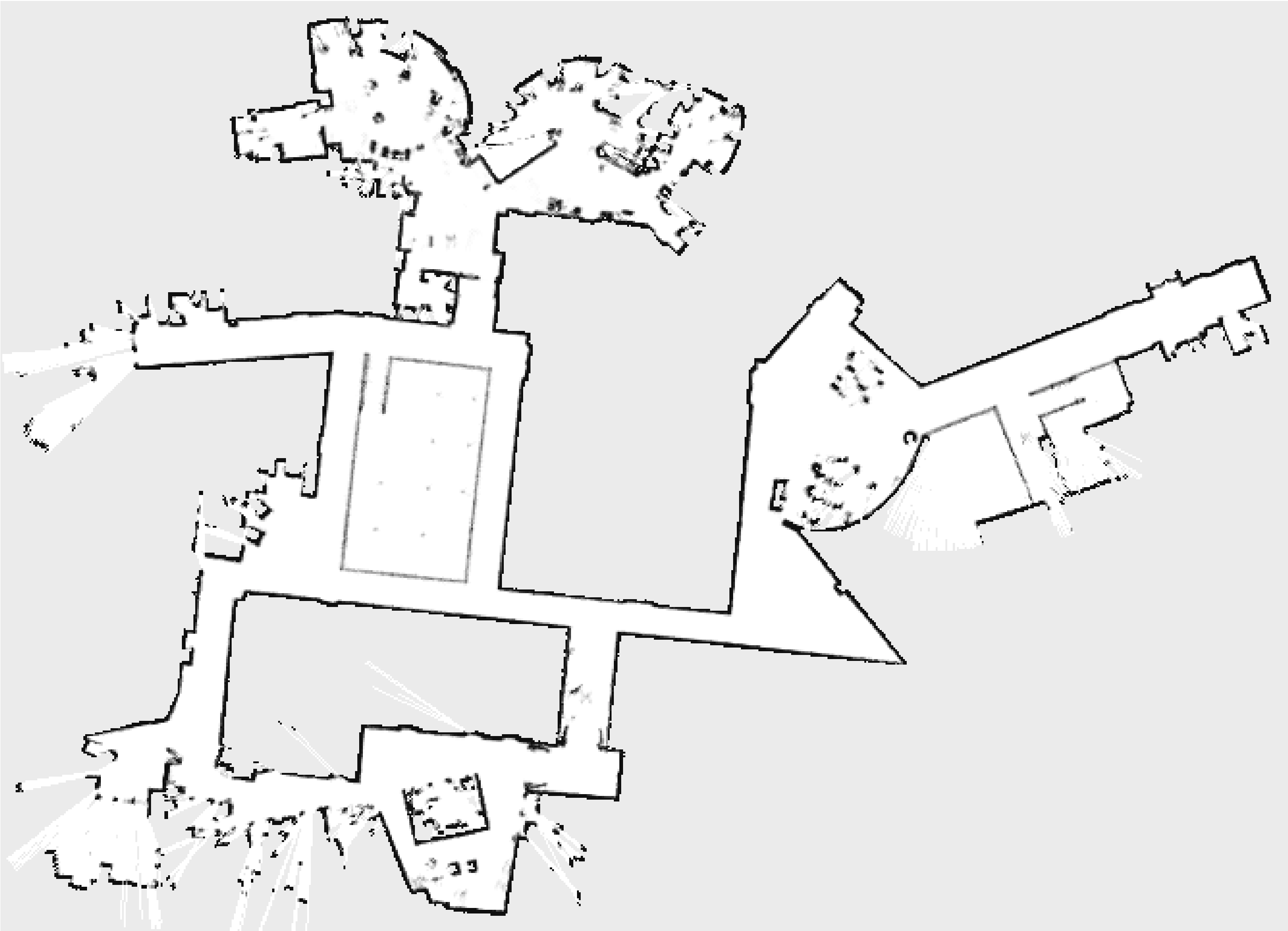
# Occupancy Grids: From Laser Scans to Maps





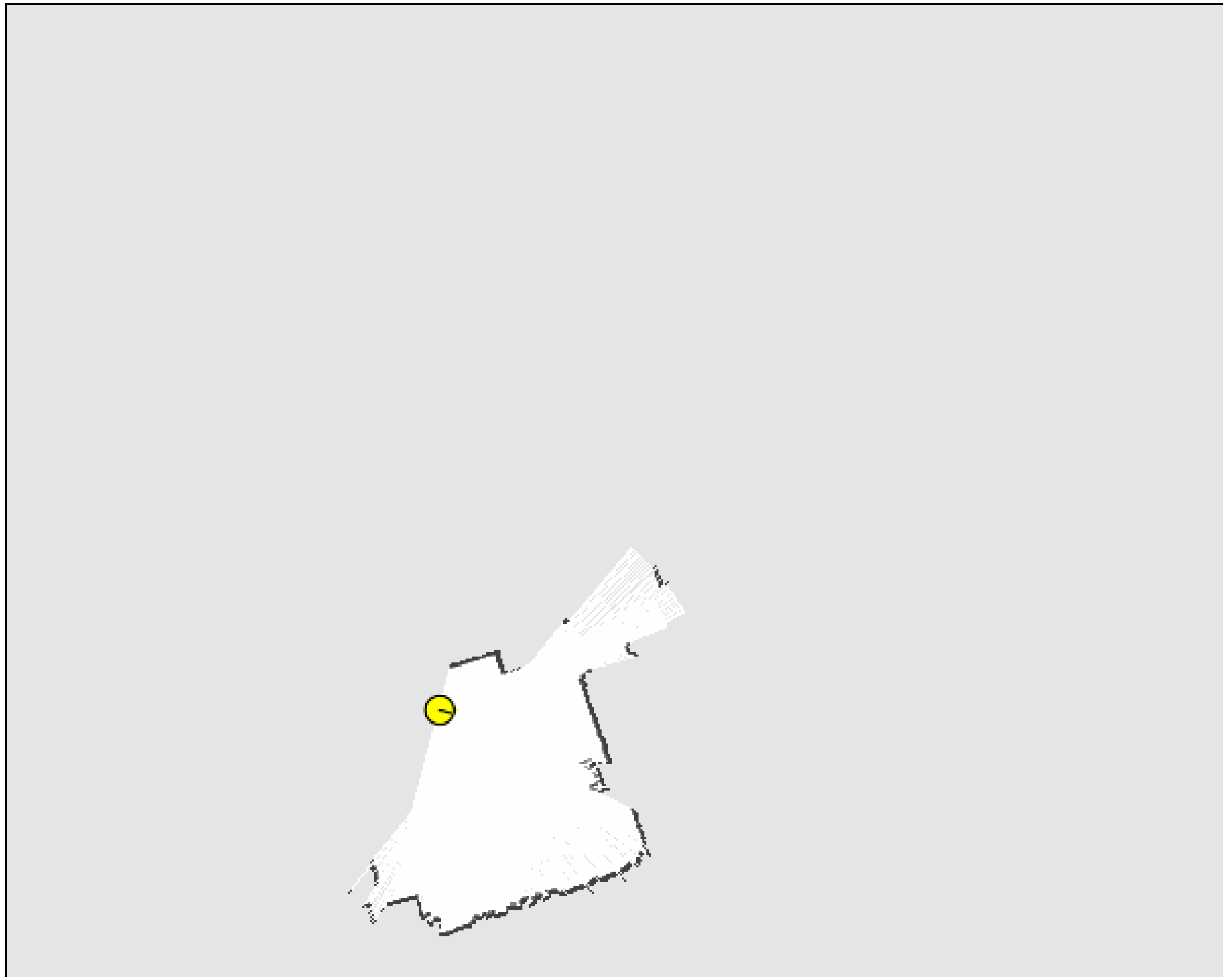
Courtesy: D. Hähnel

# Example: MIT CSAIL 3$^{rd}$ Floor



Courtesy: C. Stachniss

# Uni Freiburg Building 106



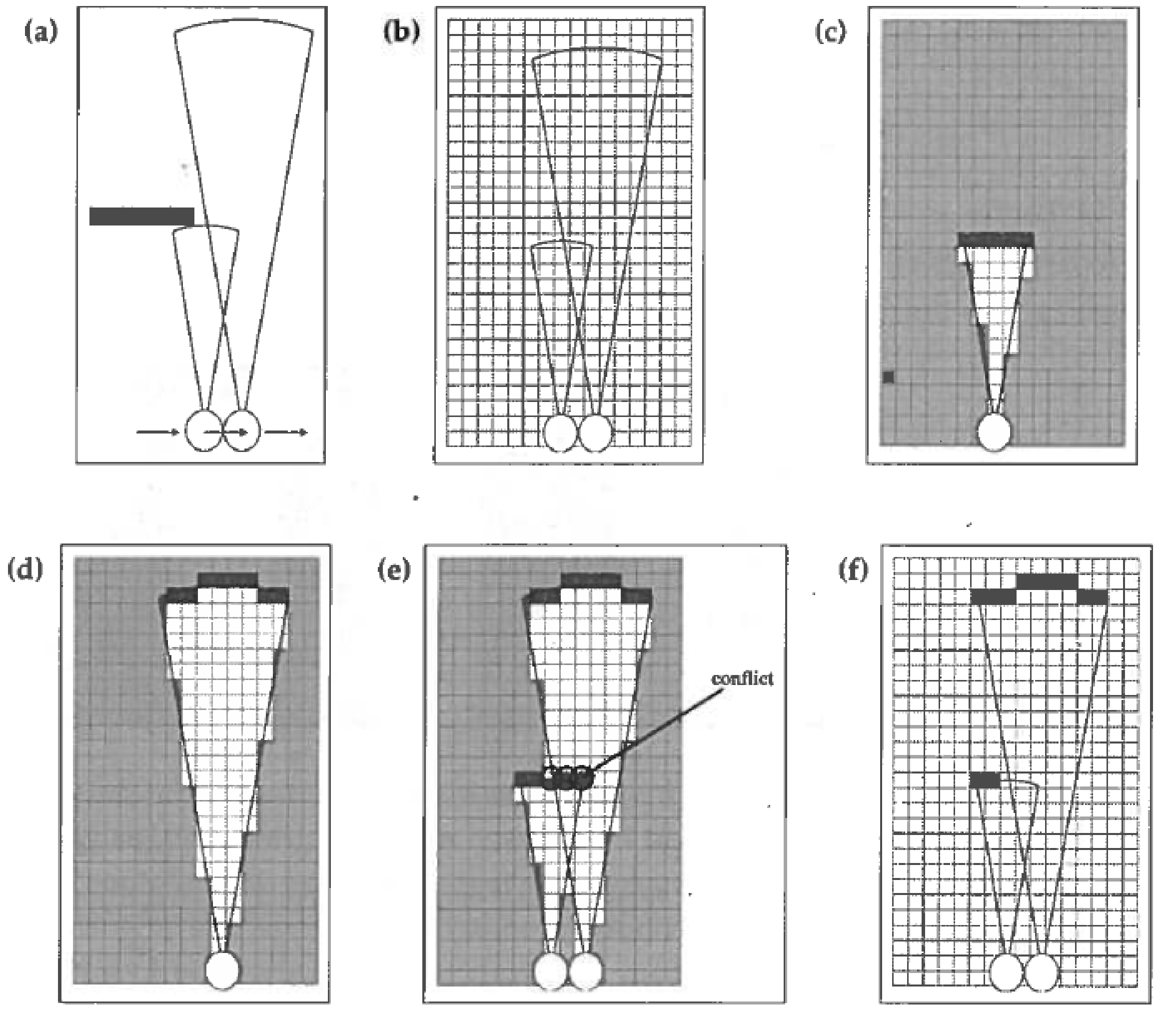
Courtesy: C. Stachniss

# 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

Courtesy: C. Stachniss

# 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*

$$m_i = \operatorname*{argmax}_{k=0,1} \; k\, l_0 + \sum_t \log \; \textbf{measurement\_model}(z_t, x_t, m \textit{ with } \mathbf{m}_i = k)$$

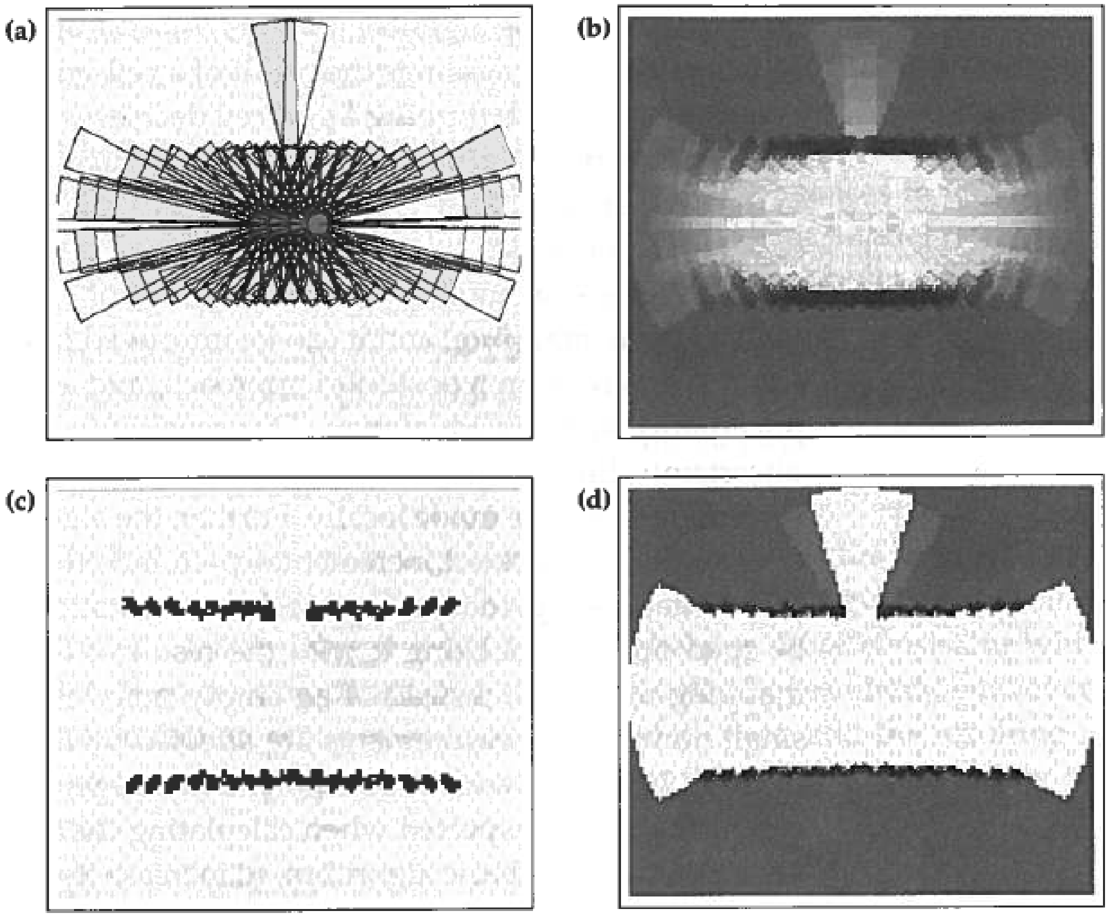
        *endfor*

    *endrepeat*

    *return* $m$

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

# Example Maximum A Posteriori OGM



# Grid Mapping Meets Reality…

Courtesy: C. Stachniss

# Mapping With Raw Odometry

Courtesy: D. Hähnel

# Incremental Scan Alignment

- Motion is noisy, we cannot ignore it
- Assuming known poses fails!
- Often, the sensor is rather precise

- Scan-matching tries to incrementally align two scans or a map to a scan, without revising the past/map

Courtesy: C. Stachniss

# Pose Correction Using Scan-Matching

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

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

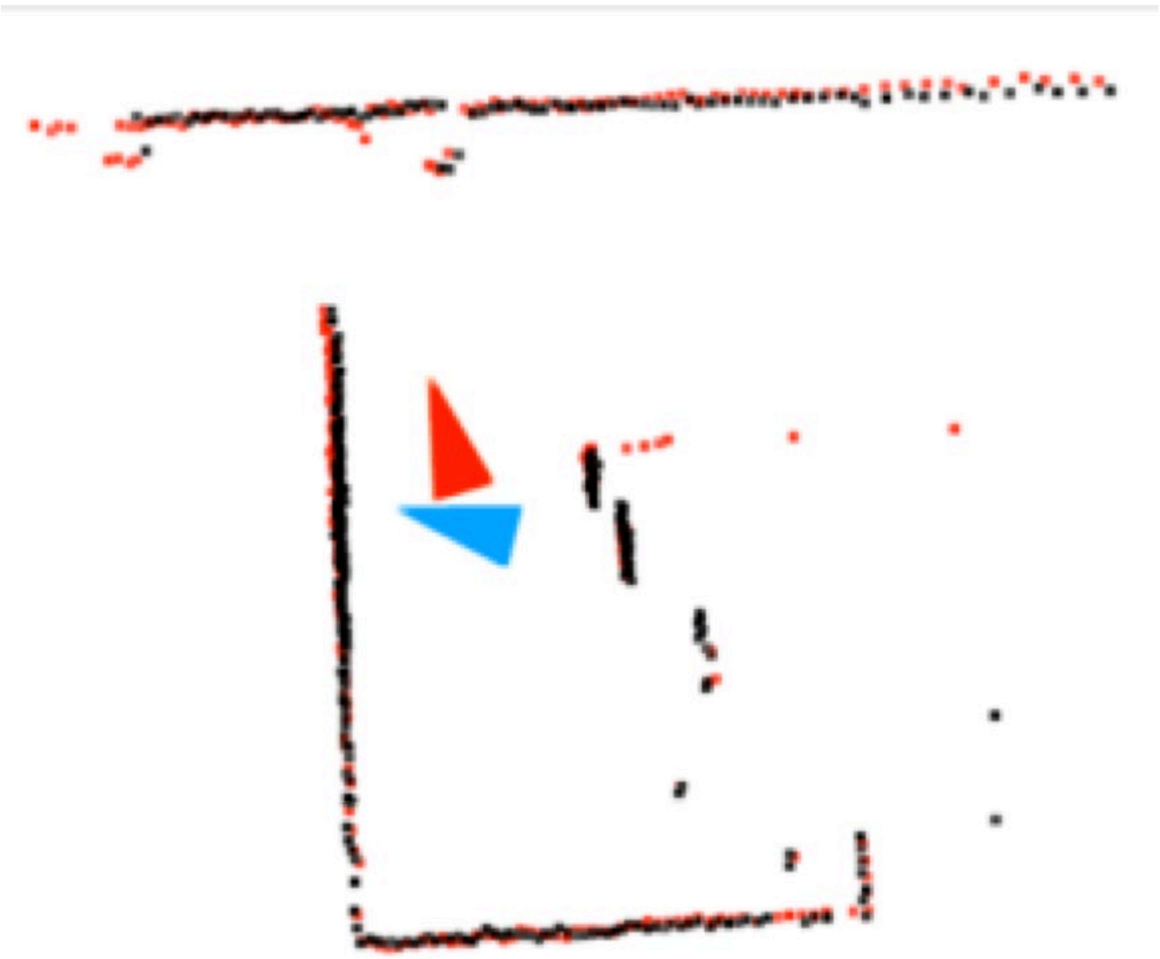
current measurement

robot motion

map constructed so far

Courtesy: C. Stachniss

# Incremental Alignment



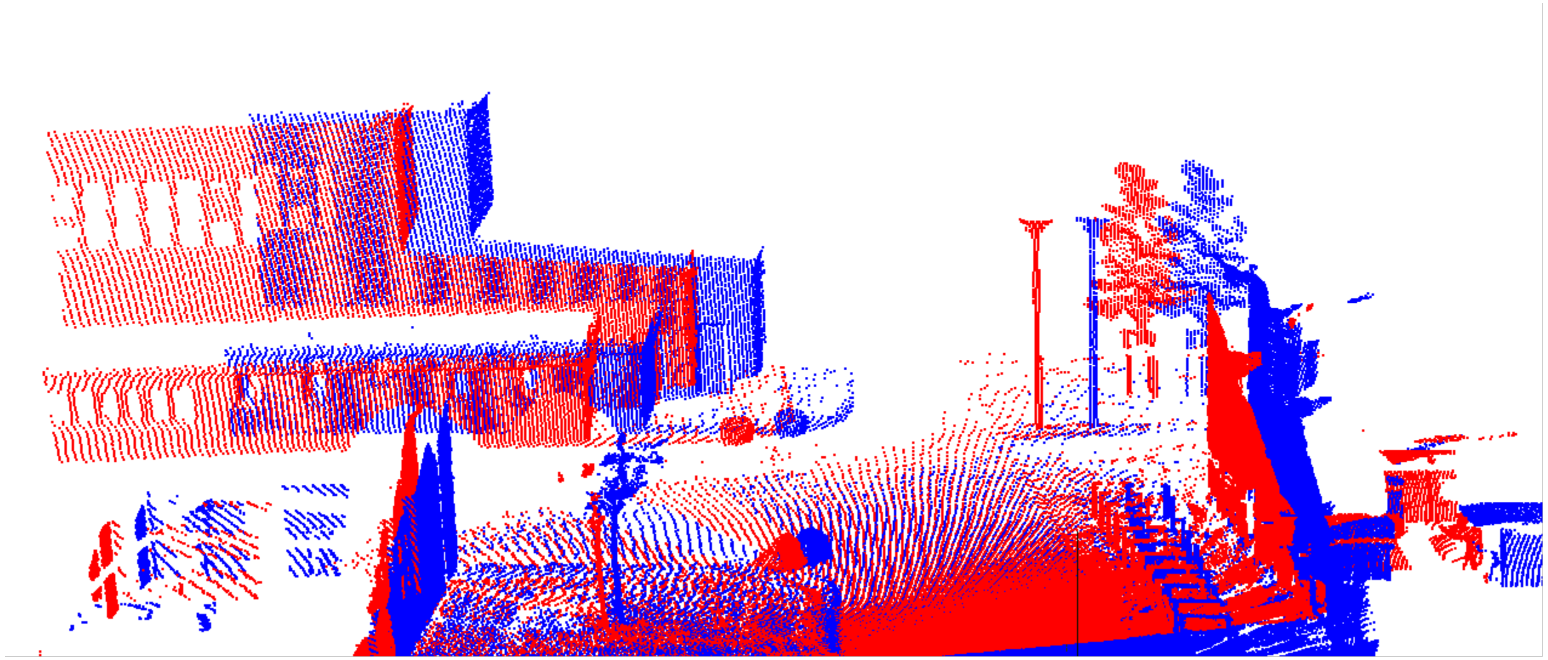
Courtesy: E. Olson

# Various Different Ways to Realize Scan-Matching

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

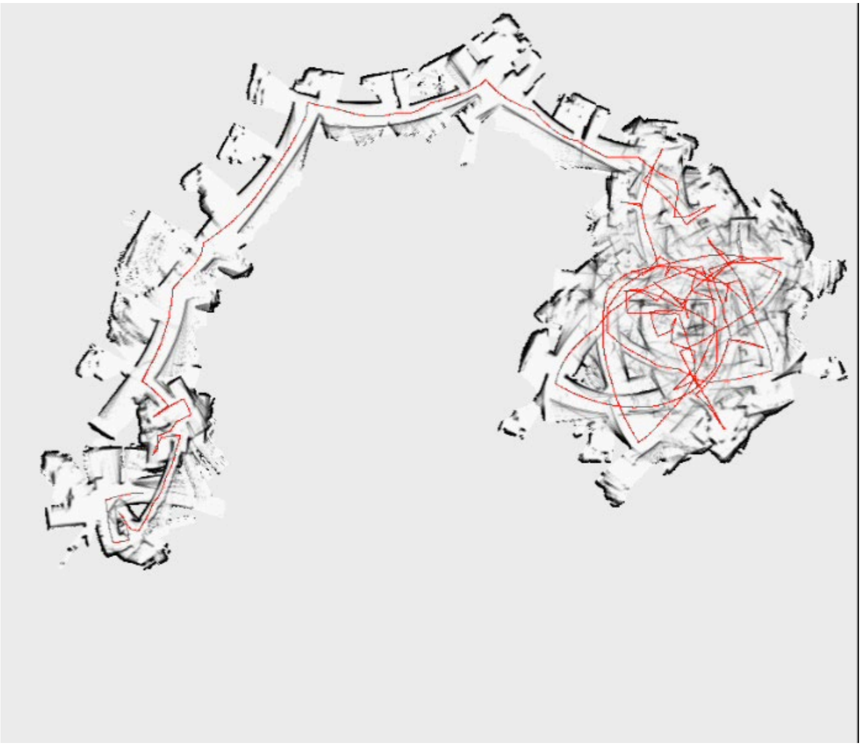
Courtesy: C. Stachniss

# Example: Aligning Two 3D Maps



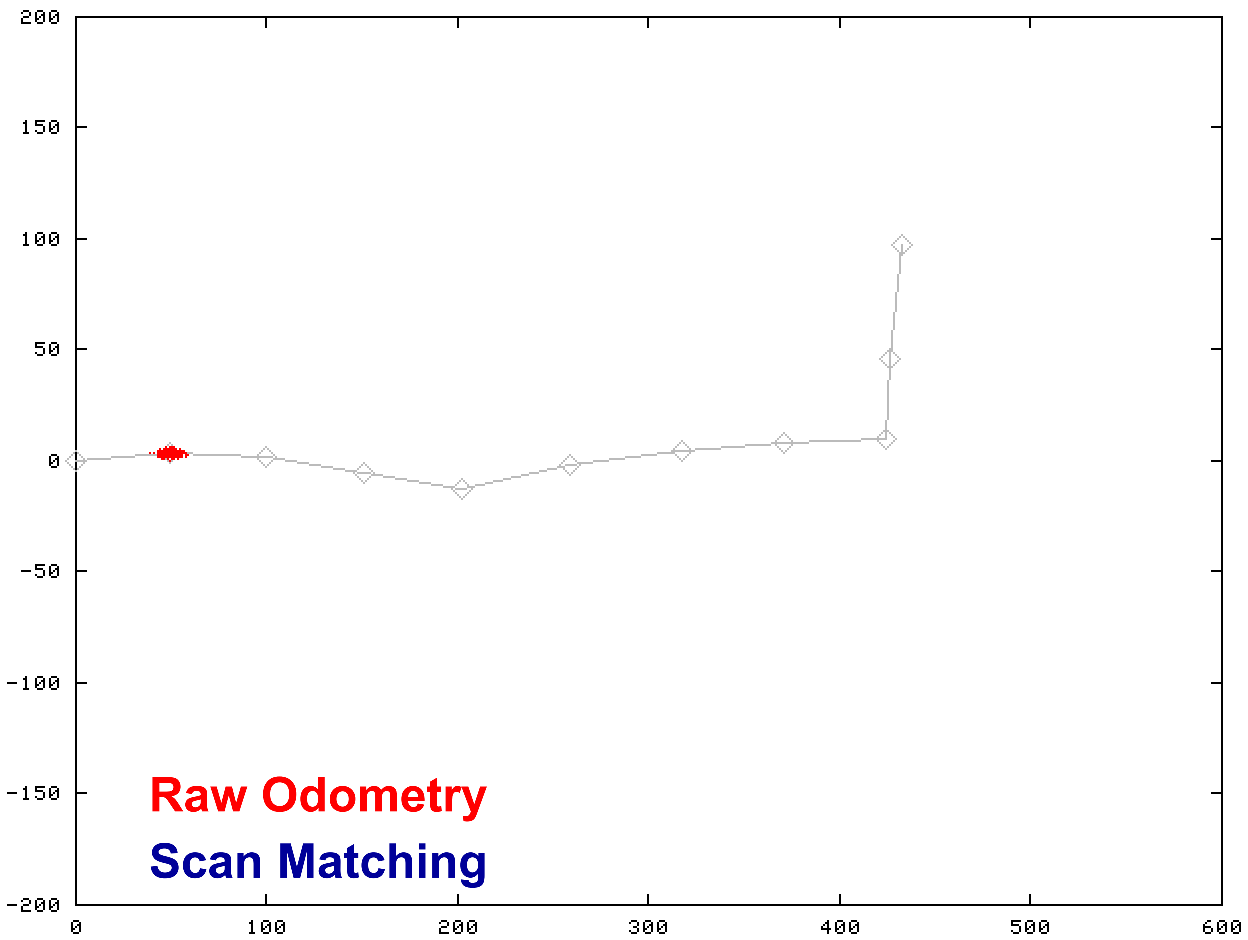
Courtesy: P. Pfaff

# With and Without Scan-Matching



Courtesy: D. Hähnel

# Motion Model for Scan Matching




Courtesy: C. Stachniss

# Scan Matching Summary

- Scan-matching improves the pose estimate (and thus mapping) substantially
- Locally consistent estimates
- Often scan-matching is not sufficient to build a (large) consistent map

Courtesy: C. Stachniss

# Grid-based SLAM

- Can we solve the SLAM problem if no pre-defined landmarks are available?
- Can we use the ideas of FastSLAM to build grid maps?
- As with landmarks, the map depends on the poses of the robot during data acquisition
- If the poses are known, grid-based mapping is easy ("mapping with known poses")

# Rao-Blackwellization

poses map observations & movements

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

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

Factorization first introduced by Murphy in 1999

# Rao-Blackwellization

poses

map

observations & movements

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

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

SLAM posterior

Robot path posterior

Mapping with known poses

Factorization first introduced by Murphy in 1999

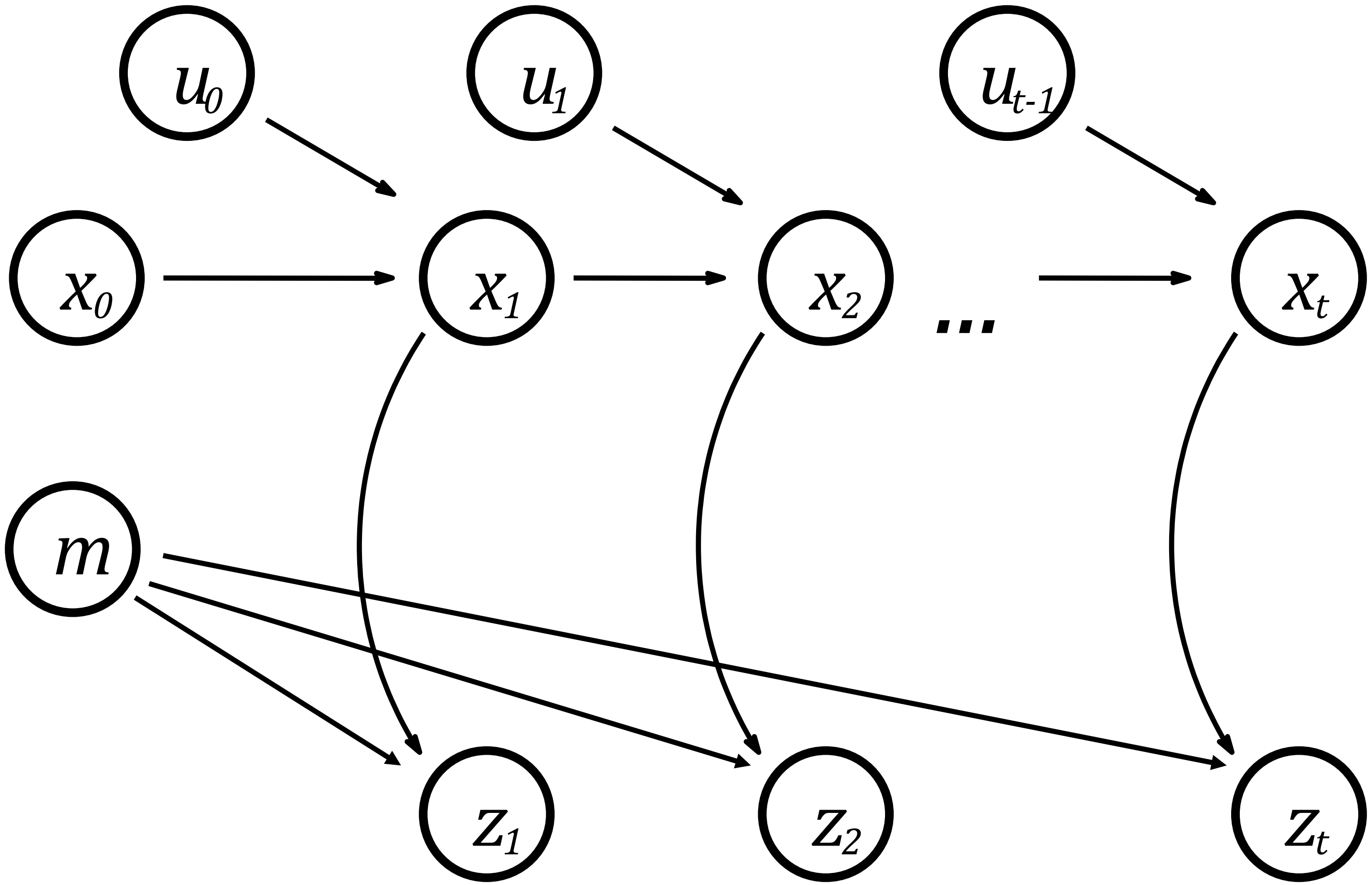
# Rao-Blackwellization

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

This is localization, use MCL

Use the pose estimate from the MCL part and apply mapping with known poses

# A Graphical Model of Rao-Blackwellized Mapping



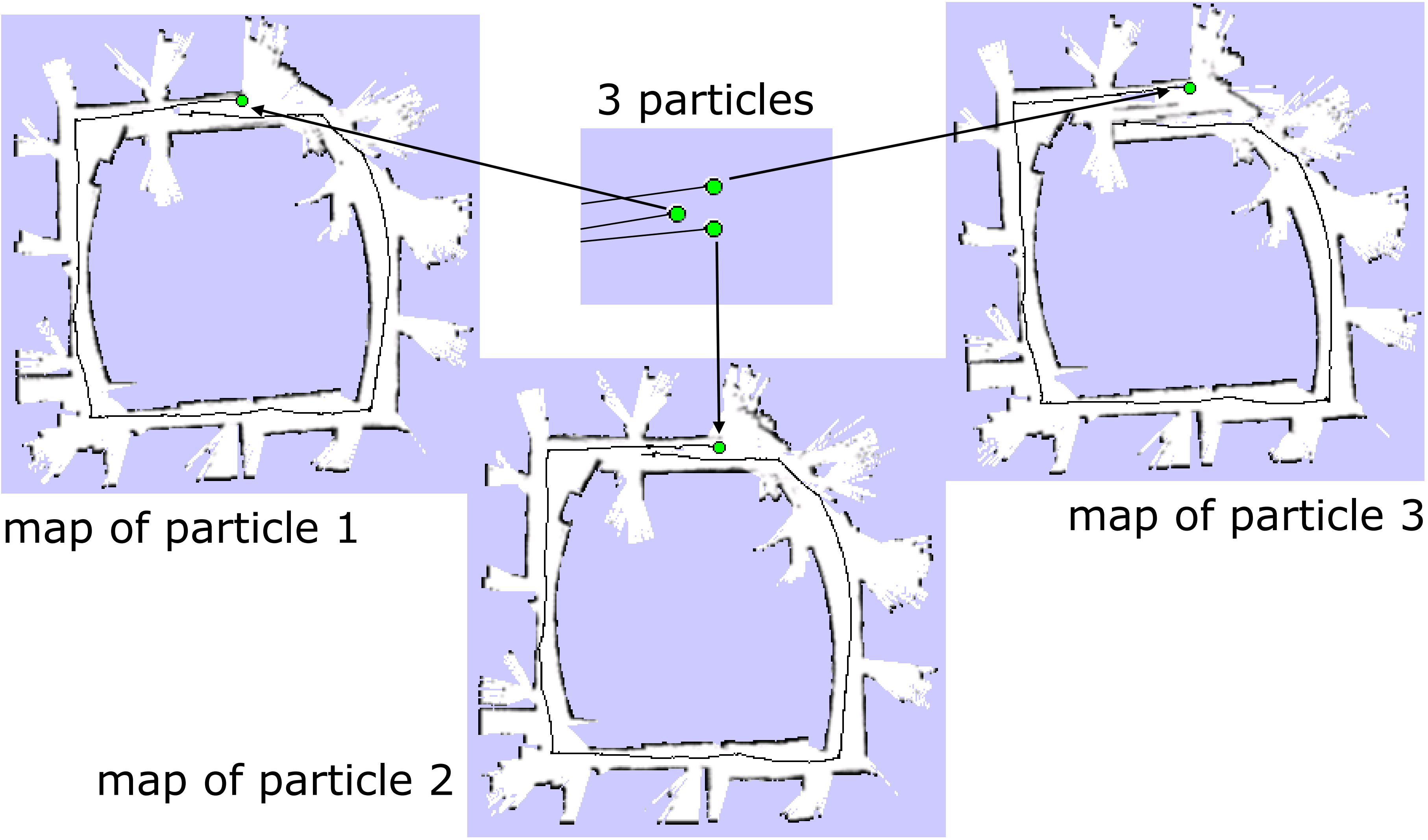
# Rao-Blackwellized Mapping

- Each particle represents a possible trajectory of the robot

- Each particle
  - maintains its own map and
  - updates it upon "mapping with known poses"

- Each particle survives with a probability proportional to the likelihood of the observations relative to its own map

# Particle Filter Example

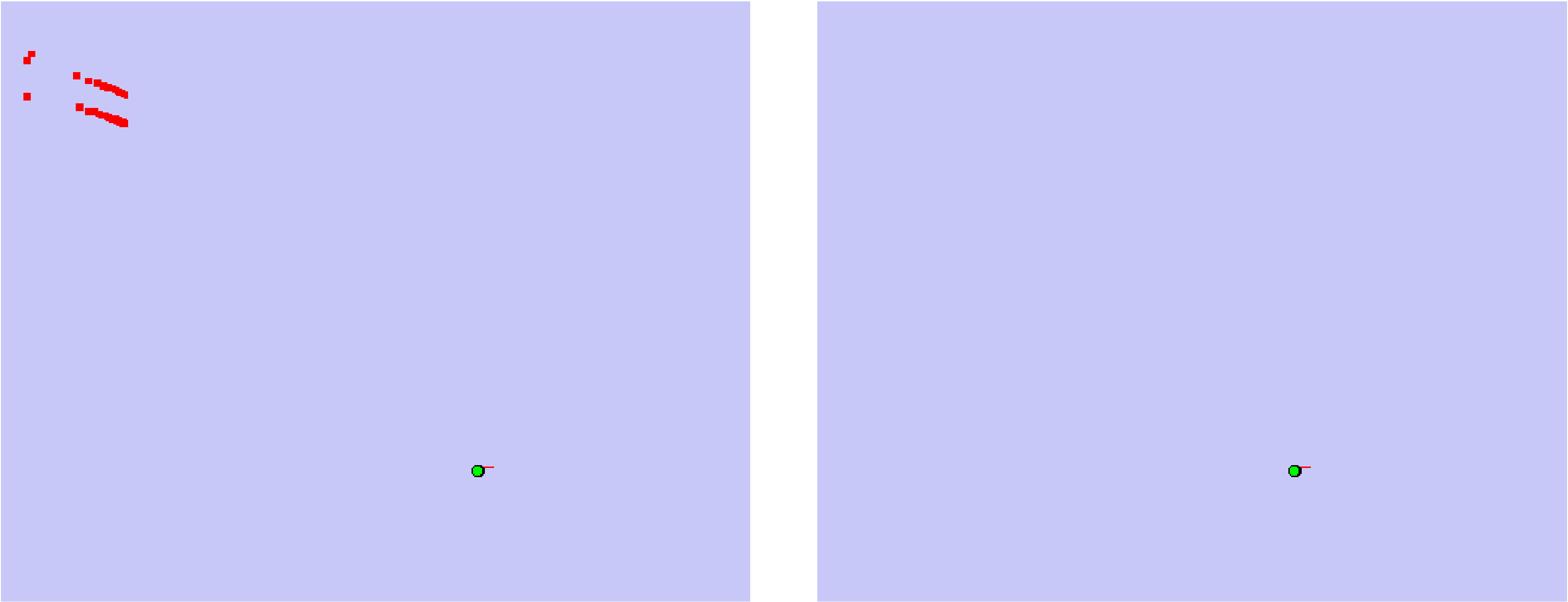


# Problem

- Each map is quite big in the case of grid maps since each particle maintains its own map
- Therefore, one needs to keep the number of particles small

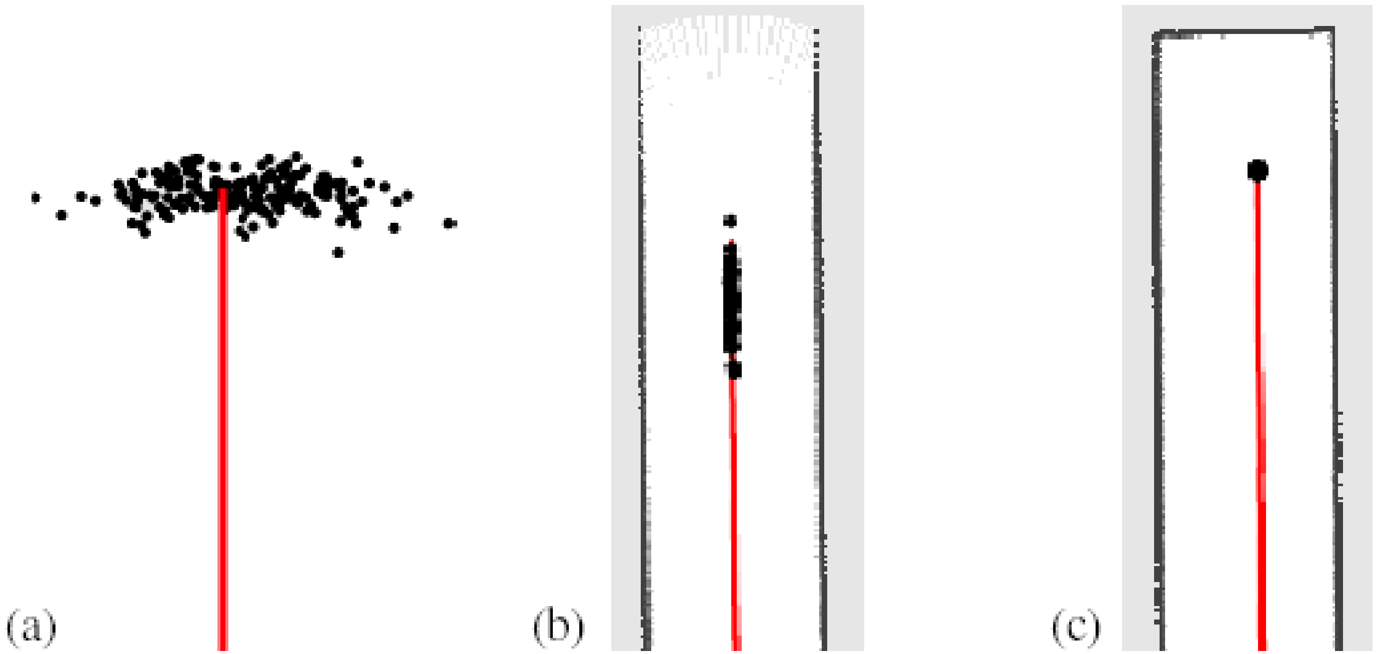
- Solution:
  Compute better proposal distributions!
- Idea:
  Improve the pose estimate before applying the particle filter

# Scan-Matching Proposal Distribution



# Improved Proposal

- The proposal adapts to the structure of the environment

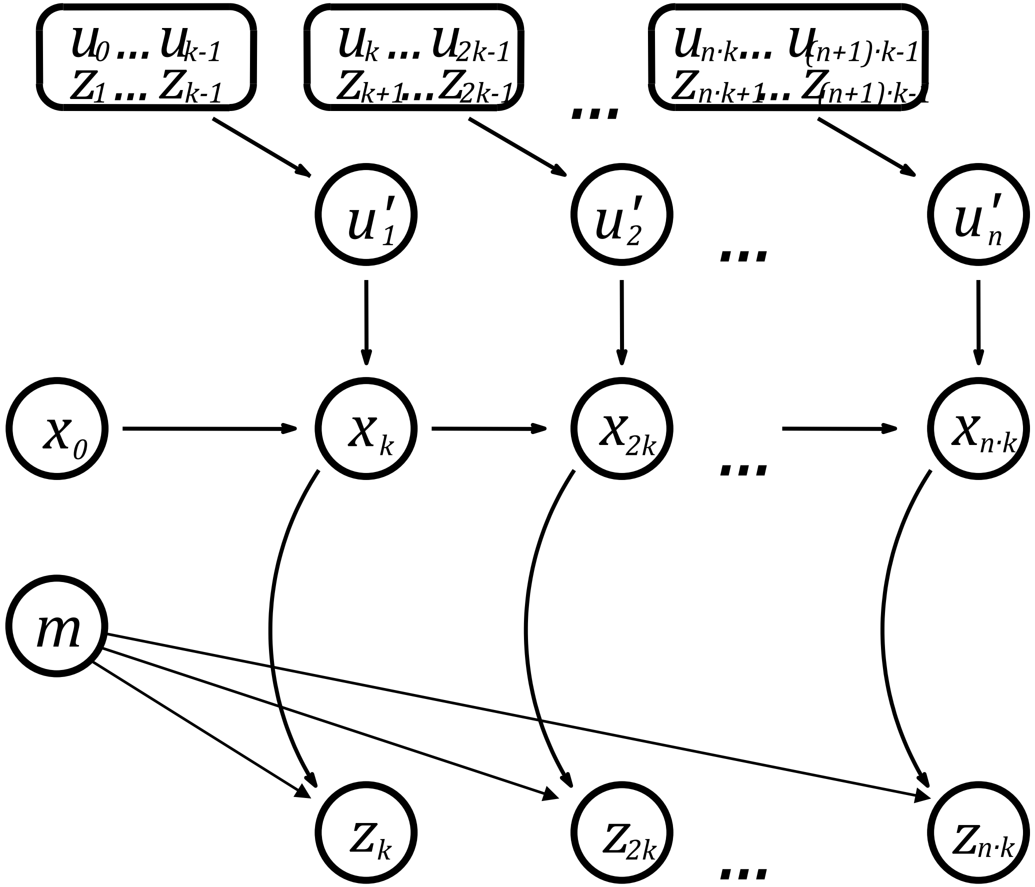


# FastSLAM with Improved Odometry

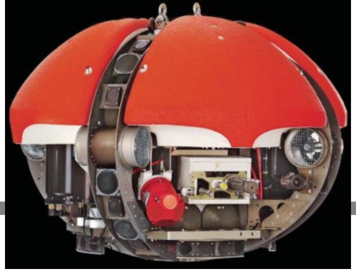
- Scan-matching provides a locally consistent pose correction
- Pre-correct short odometry sequences using scan-matching and use them as input to FastSLAM
- Fewer particles are needed, since the error in the input in smaller

[Haehnel et al., 2003]

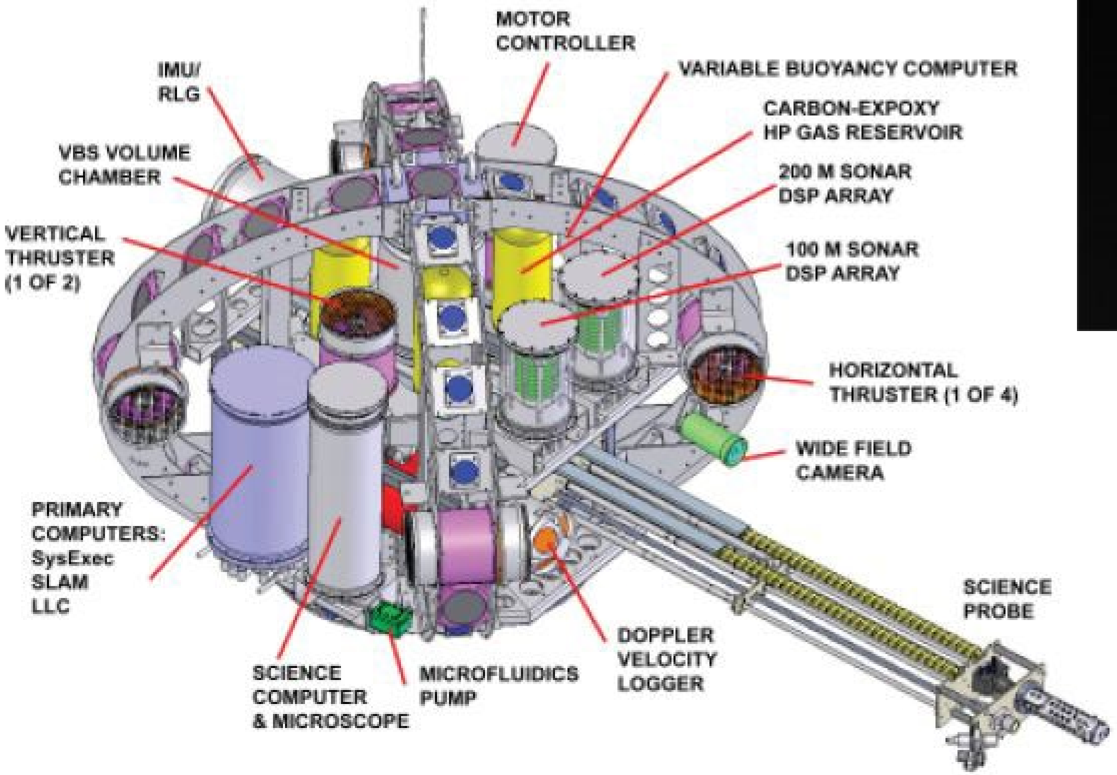
# Graphical Model for Mapping with Improved Odometry



# 3D OGM: CMU - DEPTHX



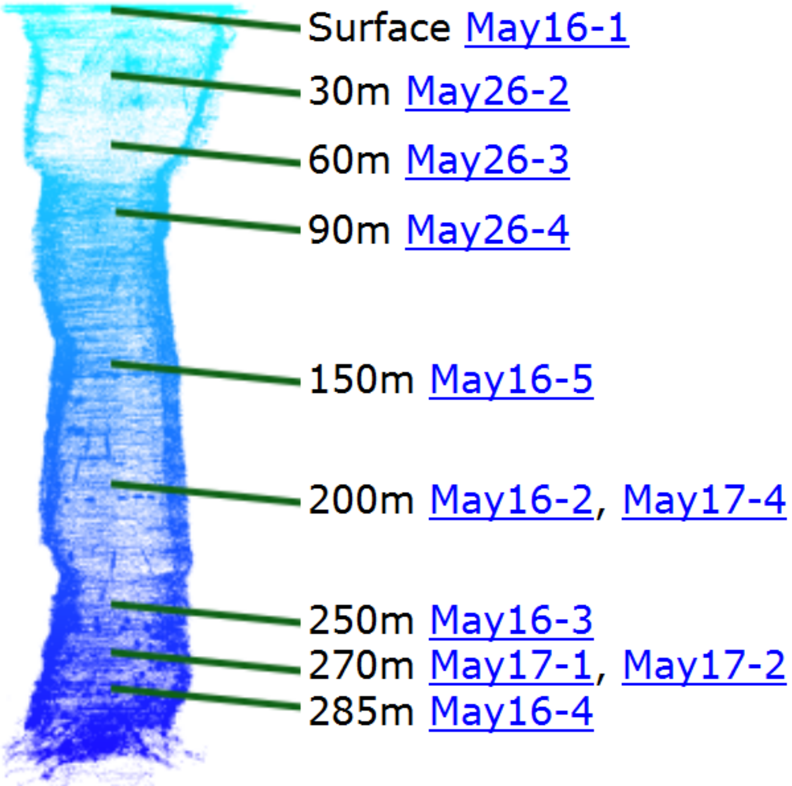
- http://www.frc.ri.cmu.edu/depthx/index.html
- Nathaniel Fairfield, George A. Kantor, and David Wettergreen, "Real-Time SLAM with Octree Evidence Grids for Exploration in Underwater Tunnels," Journal of Field Robotics, 2007

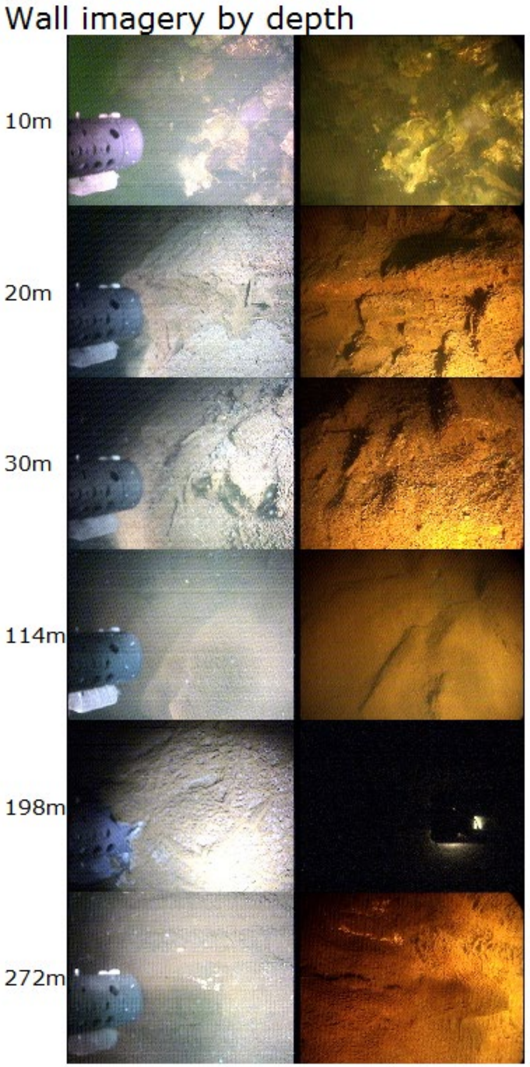



Camera
Science probe
Sonar
IMU / RLG

# Underwater Cave Exploration

- Cave Zacatón in Mexico
- The deepest sinkhole in the world
- Complete 3D map using real time SLAM
- (Particle filter)



# Occupancy Map Summary

- Occupancy grid maps are a popular approach to represent the environment of a mobile robot given known poses.
- In this approach each cell is considered independently from all others.
- It stores the posterior probability that the corresponding area in the environment is occupied.
- Occupancy grid maps can be learned efficiently using a probabilistic approach.

# FastSLAM w/ GridMaps Conclusion

- The ideas of FastSLAM can also be applied in the context of grid maps
- Utilizing accurate sensor observation leads to good proposals and highly efficient filters
- It is similar to scan-matching on a per-particle base
- The number of necessary particles and re-sampling steps can be dramatically reduced
- Improved versions of grid-based FastSLAM can handle larger environments than naïve implementations in “real time” since they need one order of magnitude fewer samples

# 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 (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

Courtesy: C. Stachniss