World modeling for mobile robots

Václav Hlaváč

Czech Technical University in Prague

Faculty of Electrical Engineering

Department of Cybernetics

Czech Republic

http://cmp.felk.cvut.cz/~hlavac

Courtesy to several authors of presentations on the web.

Lecture outline

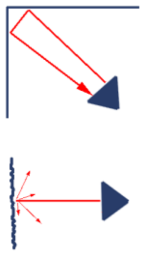
- Problem formulation – localization, mapping, simultaneous localization and mapping.
- Localization methods taxonomy.
- Representation used in mobile robot environment modeling: occupancy grid, elevation maps, full 3D map.
- Occupancy grid update using Bayesian probabilistic reasoning.
- Occupancy grid update by hit/misses counting.
- Lines and planes as the world models.

Major issues with autonomy

- Movement inaccuracy

- Enviromental uncertainty

- Sensor inaccuracy



Problem one - localization

Given:

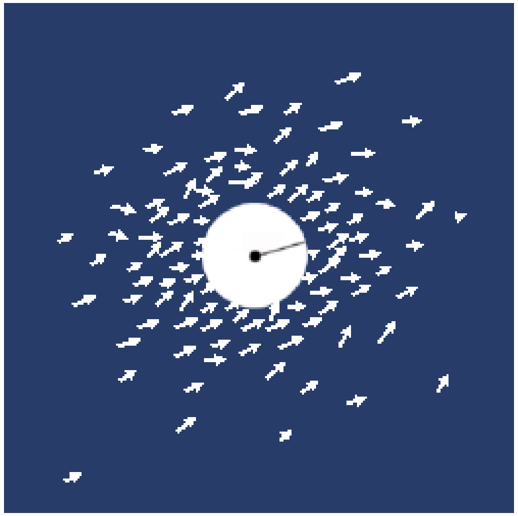
- World map.
- Robot’s initial pose.
- Sensor updates.

Find:

- Robot’s pose as it moves.

How do we solve localization?

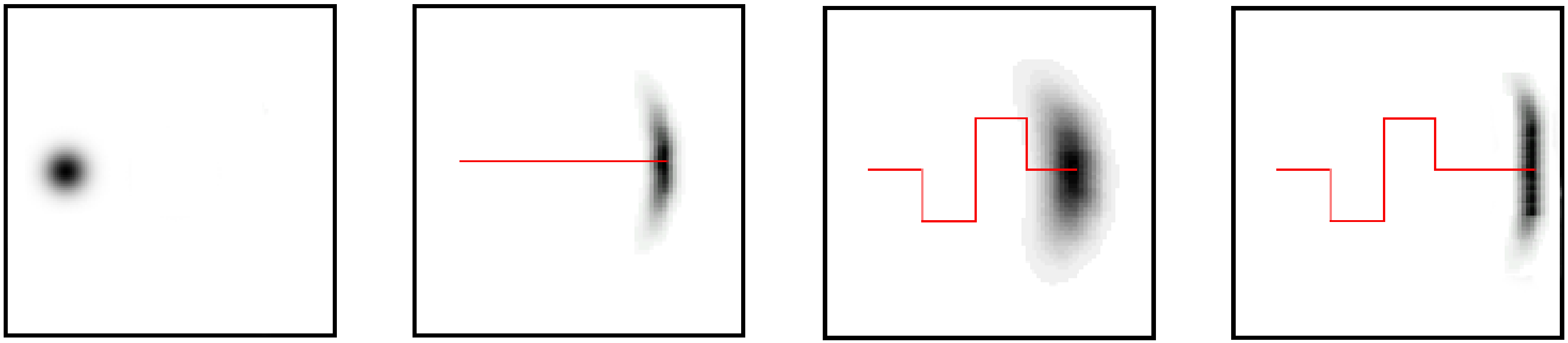
- Represent beliefs as a probability density.
- Markovian assumption - pose distribution at time t. conditioned on:
  - Pose distribution at time t-1.
  - Movement at time t-1.
  - Sensor readings at time t.
- Discretize the density by sampling.



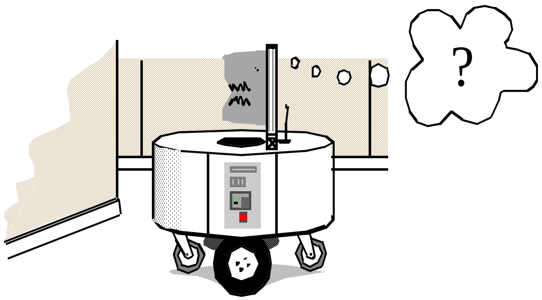
Localization loop

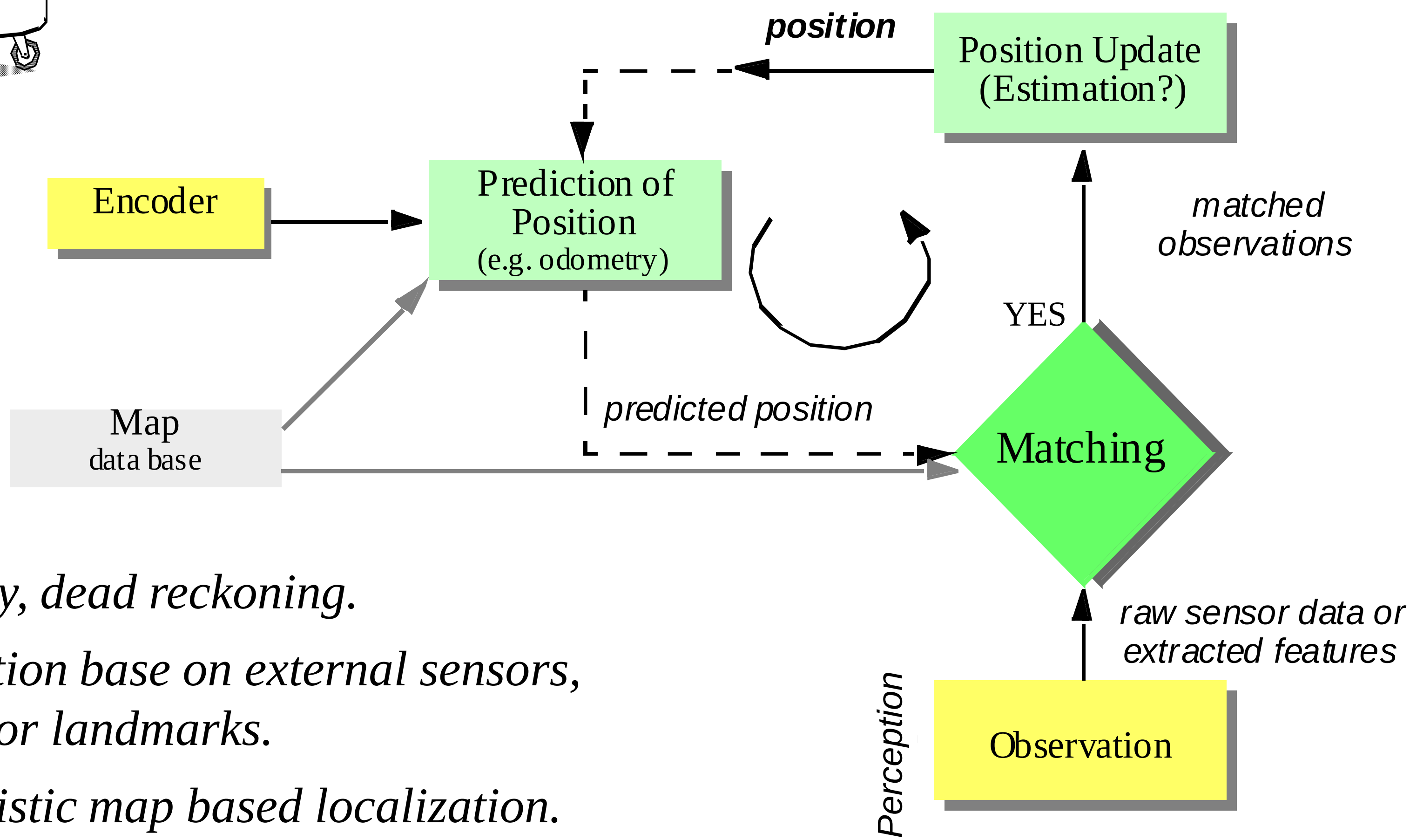
At every time step t:

- **Update** each sample's new location based on movement.
- **Resample** the pose distribution based on sensor readings.



Localization, where am I?





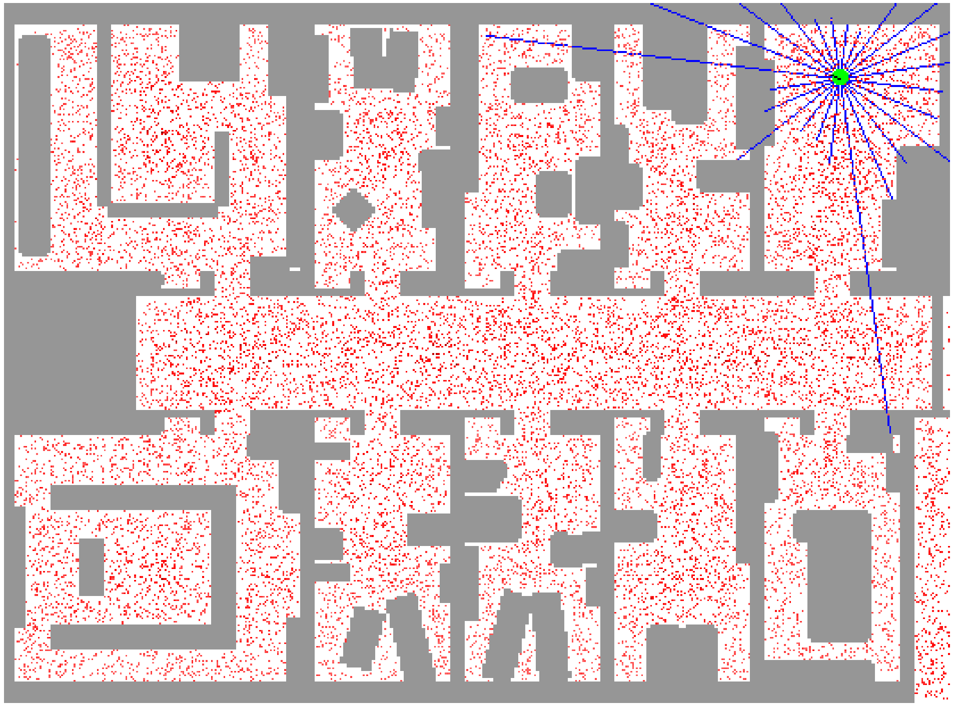
- *Odometry, dead reckoning.*
- *Localization base on external sensors, beacons or landmarks.*
- *Probabilistic map based localization.*

Localization methods

- Mathematic Background, Bayes Filter
- Markov Localization:
  - Central idea: robot position as the probability distribution, Bayes' rule and convolution to update the belief.
  - Markov Assumption: past and future data are independent if one knows the current state.
- Kalman Filtering
  - Central idea: posing localization problem as a sensor fusion problem
  - Assumption: gaussian distribution function
- Particle Filtering
  - Central idea: Sample-based, nonparametric Filter
  - Monte-Carlo method
- SLAM (simultaneous localization and mapping)
- Multi-robot localization.

Globalization sidekick

Localization without knowledge of the start location.



Credit to Dieter Fox for this demo.

Problem two - mapping

Given:

- Robot.
- Sensor readings.

Find:

- Map of the environment,
- and implicitly, the robot’s location as it moves in the environment.

SLAM – Simultaneous localization and mapping

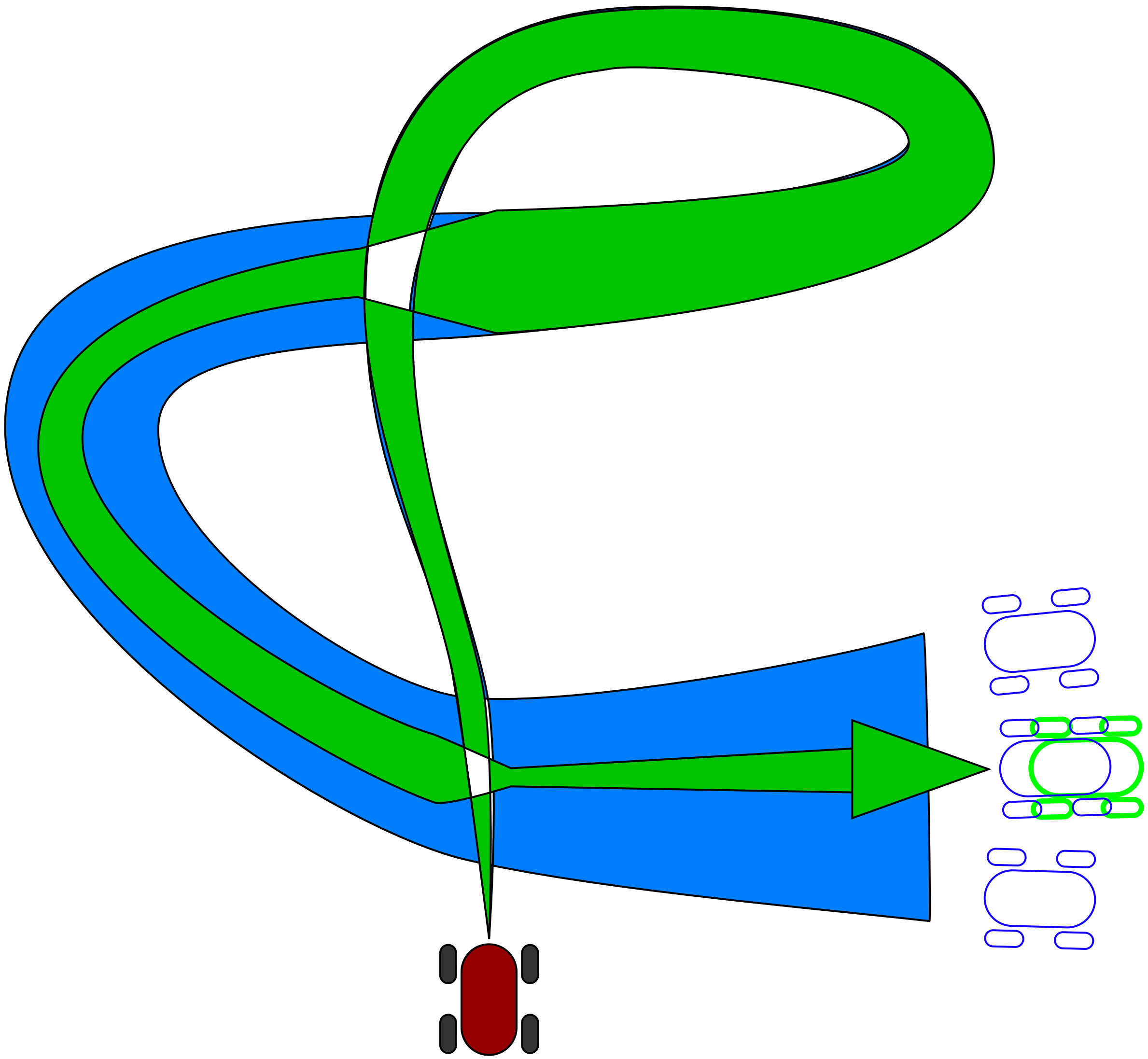
If we have a map:

We can localize!

If we can localize:

We can make a map!

Odometry versus SLAM



- **Odometry**
  - Incremental growth of the position uncertainty.
  - Optimization methods used.
- **Visual SLAM**
  - Carthographic memory.
  - Closing the loop ⇒ decrease of uncertainty.

This lecture – world modeling

The aim of the world modeling:

- The aim is to construct/update the model of the world (environment) of a mobile robot.
- The world model of the robot allows the robot to adapt its decisions to the current state of the world.
- The world model is constructed/updated from sensor data as the robot explores its environment.

- Throughout this lecture we will describe how to calculate a map given we know the pose of the vehicle. This is not the SLAM problem.

Challenges in world modeling

1. **Compact models** are needed to be used efficiently by other components (as path planners).

   ⇒ *Universal world representation does not exist ⇒ choice from several approaches.*

2. **The model** must be **adapted to the task and environment**. E.g., model based as set of planes is not suited for natural terrains.

3. **Uncertainty**: the model must accommodate to uncertainty in both sensor data and to robot's state estimation.

A historical perspective

2D occupancy grid

- Indoor environment
- Uncertainty expressed as a probability of occupancy of a cell in a grid.
- Highly structured environment $\Rightarrow$ lines, planes.

Elevation maps

- Came with longer range sensing (laser, stereo vision)
- 2½D grid, each grid contains elevation (possibly other features).

Full 3D maps

- Needed to represent vertical or overhanging structures, e.g., in the urban environment.
- 3D grid.
- Point clouds, meshes.

The general problem of mapping

Formally, mapping involves, given the sensor data z_i (observations), $i=1,\ldots,n$

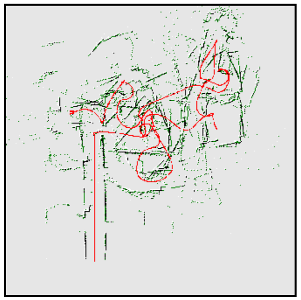
$$d = \{z_1, z_2, \ldots, z_n\}$$

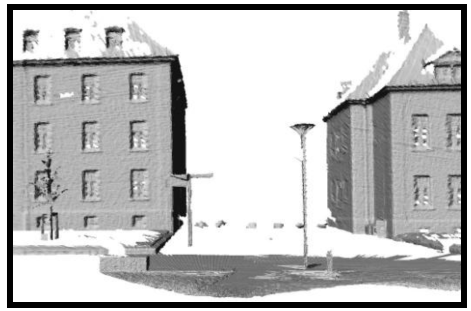
The goal is to calculate the most likely map

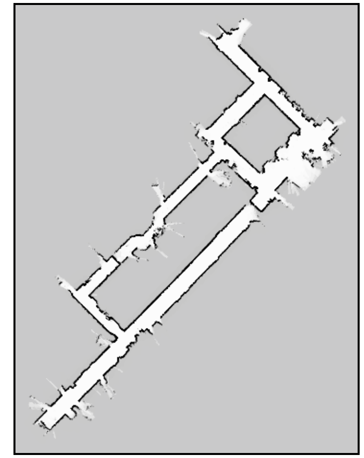
$$m^* = \arg\max_m P(m \mid d)$$

Types of localization tasks

- Grid maps or scans

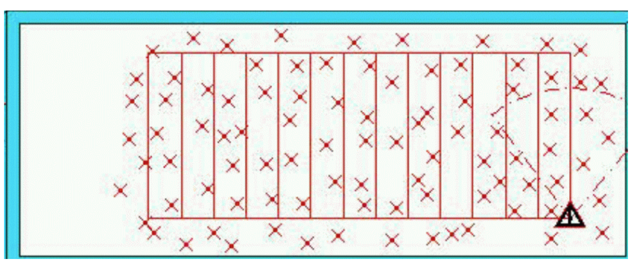


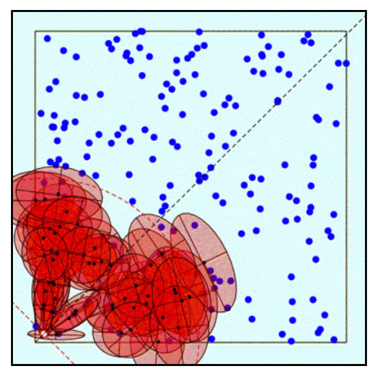


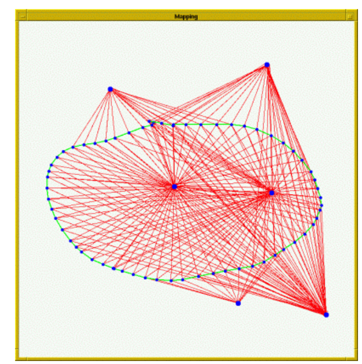


[Lu & Milios, 97; Gutmann, 98: Thrun 98; Burgard, 99; Konolige & Gutmann, 00; Thrun, 00; Arras, 99; Haehnel, 01;…]

- Landmark-based







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

Problems in mapping

- **Sensor interpretation**
  - How do we extract relevant information from raw sensor data?
  - How do we represent and integrate this information over time?

- **Robot locations have to be estimated**
  - How can we identify that we are at a previously visited place?
  - This problem is the so-called data association problem.

Occupancy grid maps

- Introduced by Moravec and Elfes in 1985.
- Represent environment by a grid.
- Estimate the probability $m_t^{[x,y]}$ that a location x,y is occupied by an obstacle in the time instant t.
- Key assumptions
  - Occupancy of individual cells ($m[xy]$) is independent

$$Bel(m_t) = P(m_t \mid z_2, \ldots, z_t)$$
$$= \prod_{x,y} Bel(m_t^{[xy]})$$

  - Robot positions are known!

Updating occupancy grid maps

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

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

- Additional assumption: Map is static.

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

Updating occupancy grid maps

- Update the map cells using the inverse sensor model

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

- Or use the log-odds representation

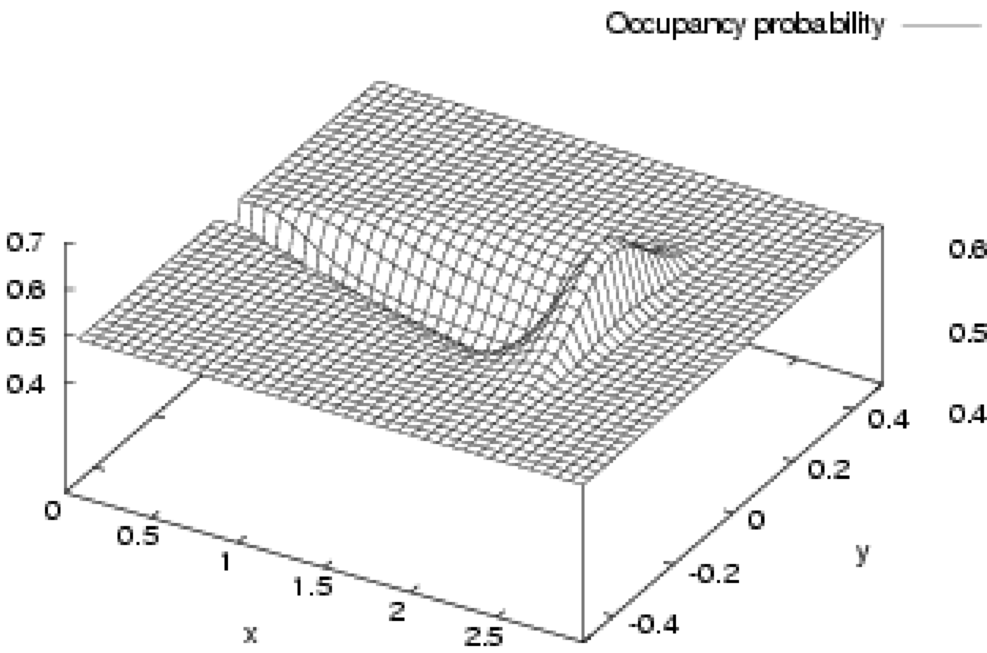
$$\begin{aligned}\overline{B}\left(m_t^{[xy]}\right)=&\log odds\left(m_t^{[xy]}\mid z_t,u_{t-1}\right)\\&-\log odds\left(m_t^{[xy]}\right)\\&+\overline{B}\left(m_{t-1}^{[xy]}\right)\end{aligned}$$

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

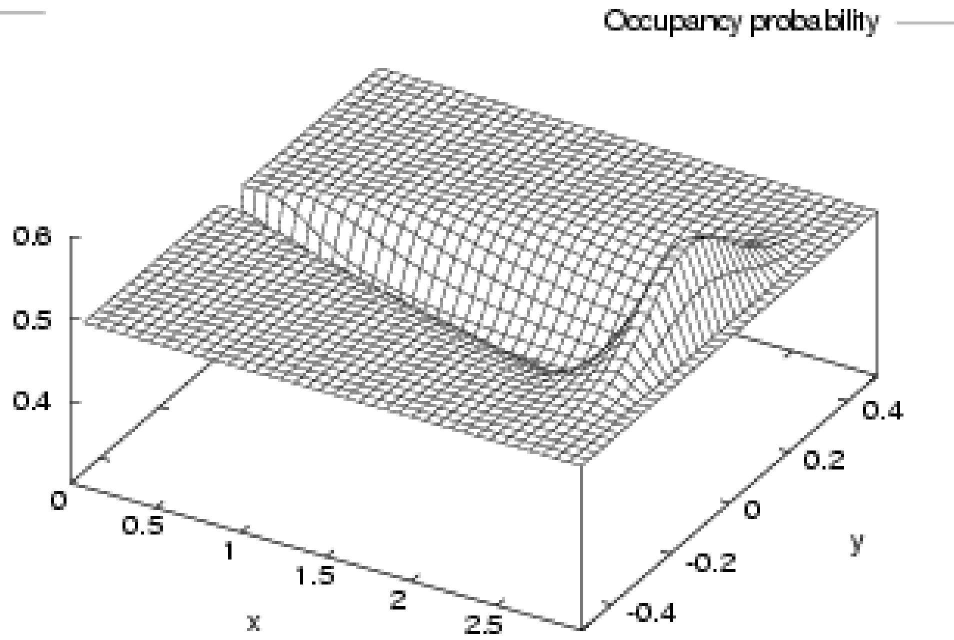
$$odds(x):=\left(\frac{P(x)}{1-P(x)}\right)$$

Typical sonar model for occupancy grid maps

Combination of a linear function and a Gaussian:

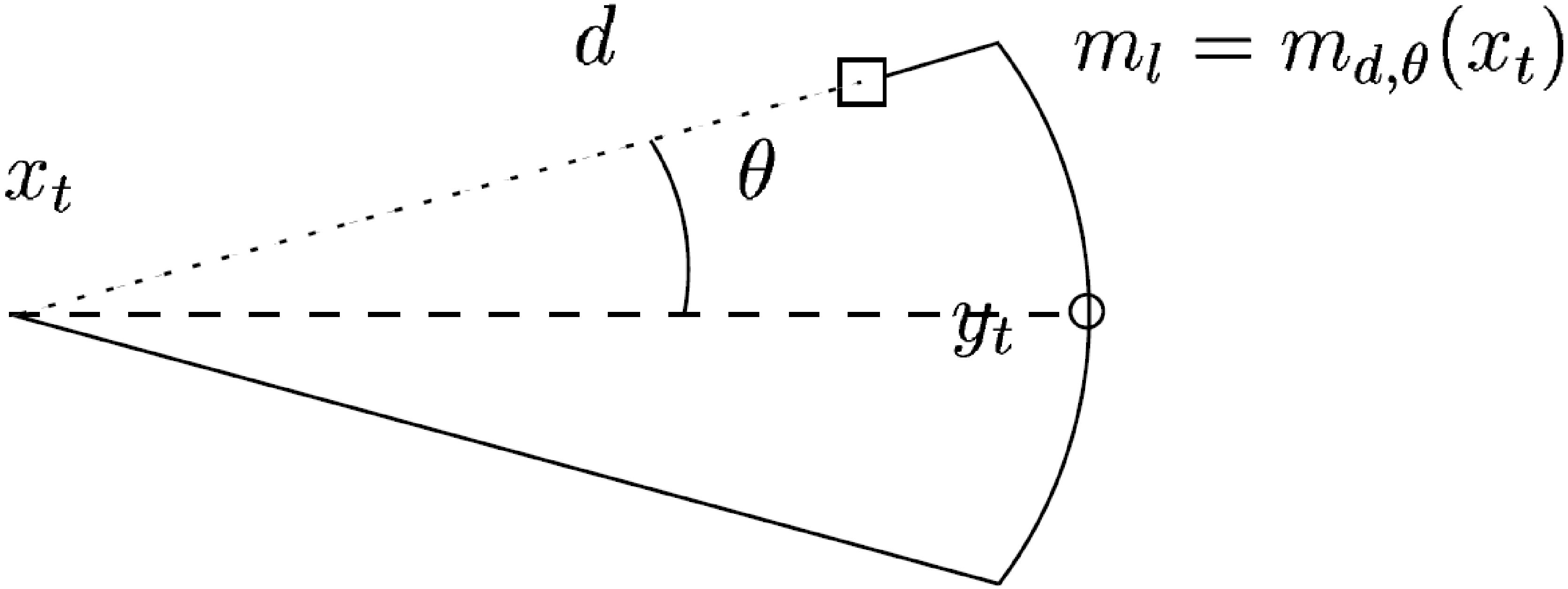


For z = 2.0 m.

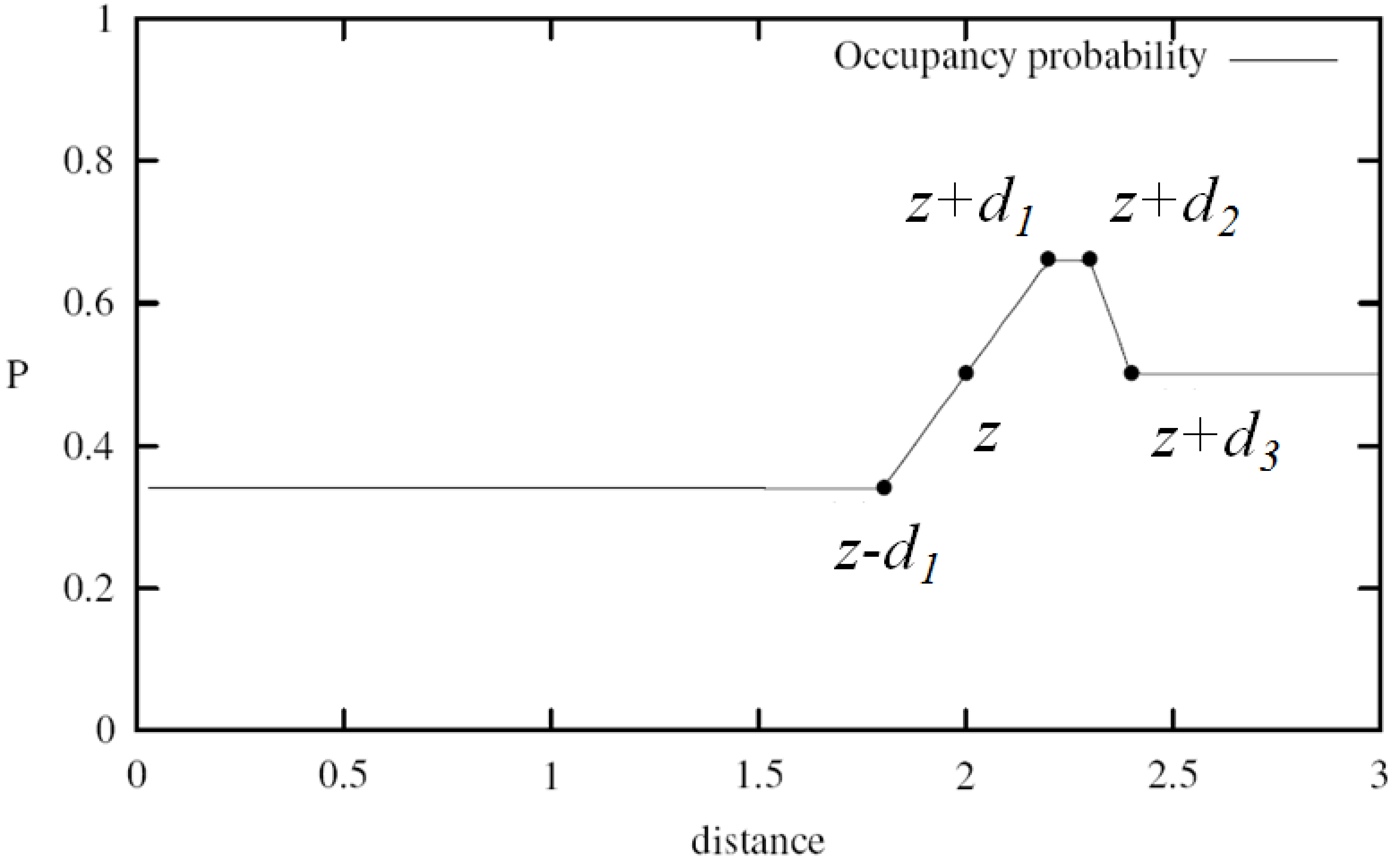


For z = 2.5 m.

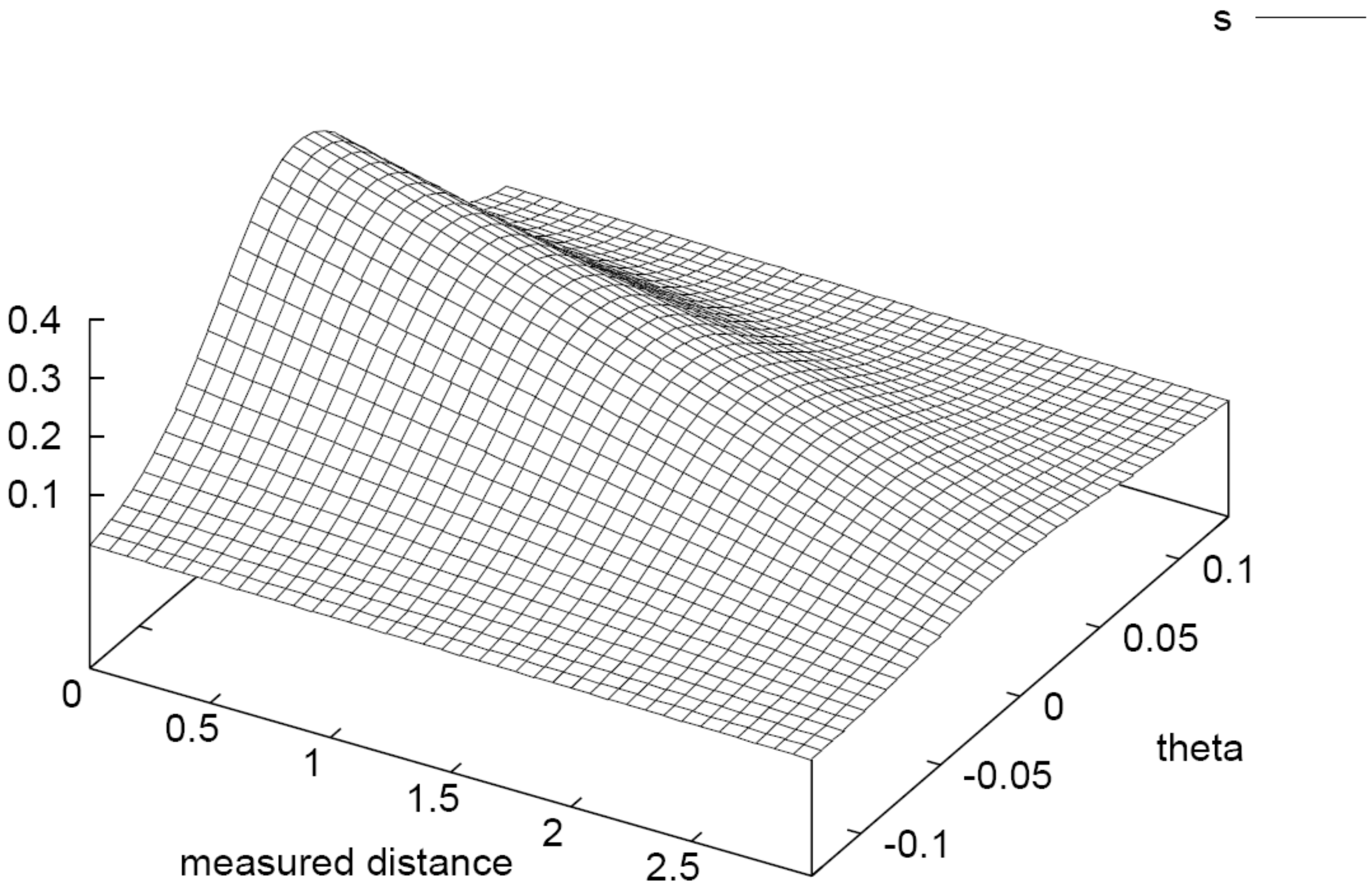
Key parameters of the model



Occupancy value depending on the measured distance



Deviation from the prior belief

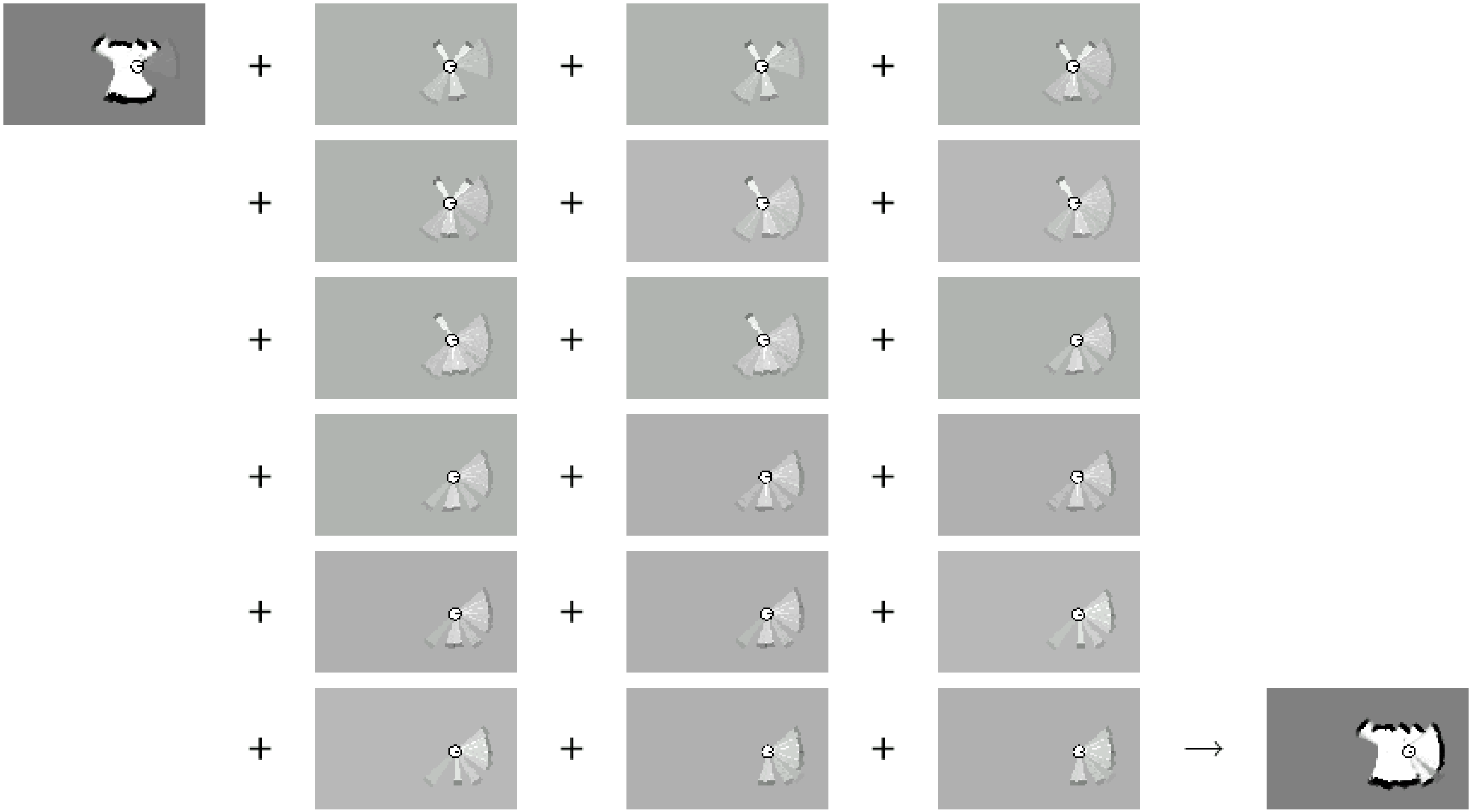


Calculating the occupancy probability based on a single observation

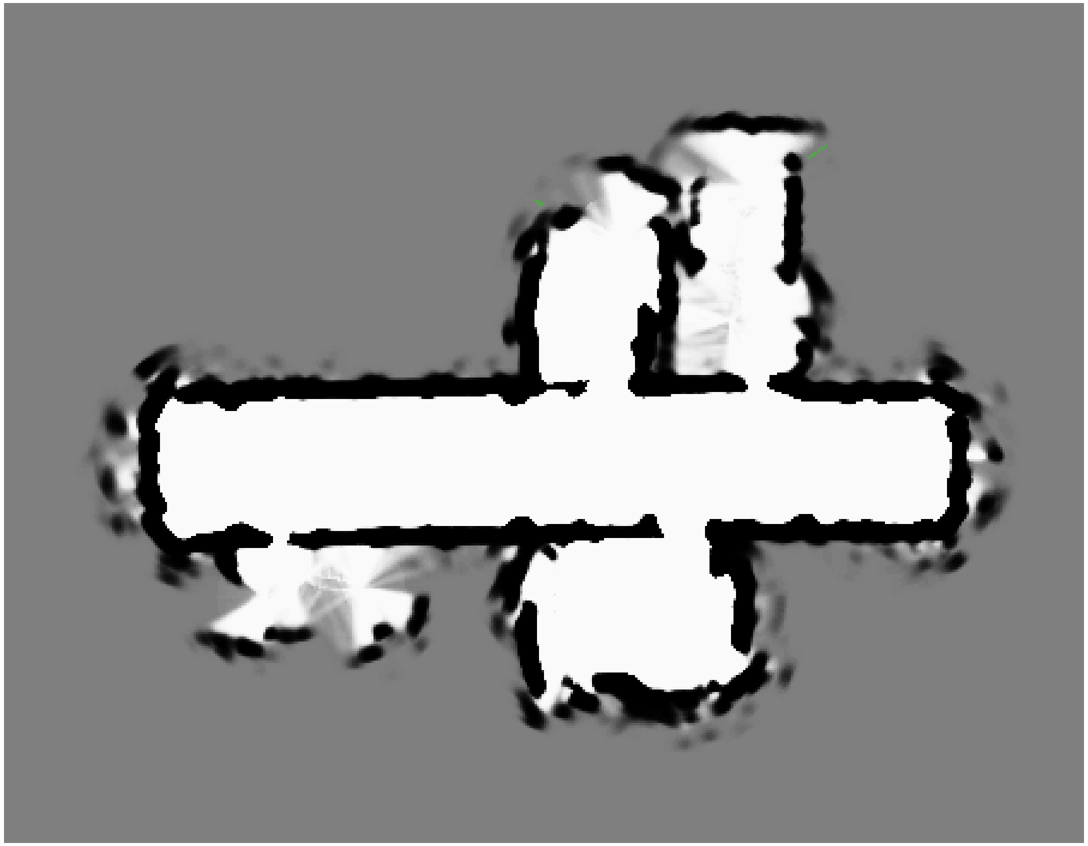
$$P(m_{d,\theta}(x(k)) \mid y(k), x(k)) = P(m_{d,\theta}(x(k)))$$

$$+ \begin{cases} -s(y(k),\theta) & d < y(k) - d_1 \\ -s(y(k),\theta) + \frac{s(y(k),\theta)}{d_1}\,(d - y(k) + d_1) & d < y(k) + d_1 \\ s(y(k),\theta) & d < y(k) + d_2 \\ s(y(k),\theta) - \frac{s(y(k),\theta)}{d_3 - d_2}\,(d - y(k) - d_2) & d < y(k) + d_3 \\ 0 & \text{otherwise.} \end{cases}$$

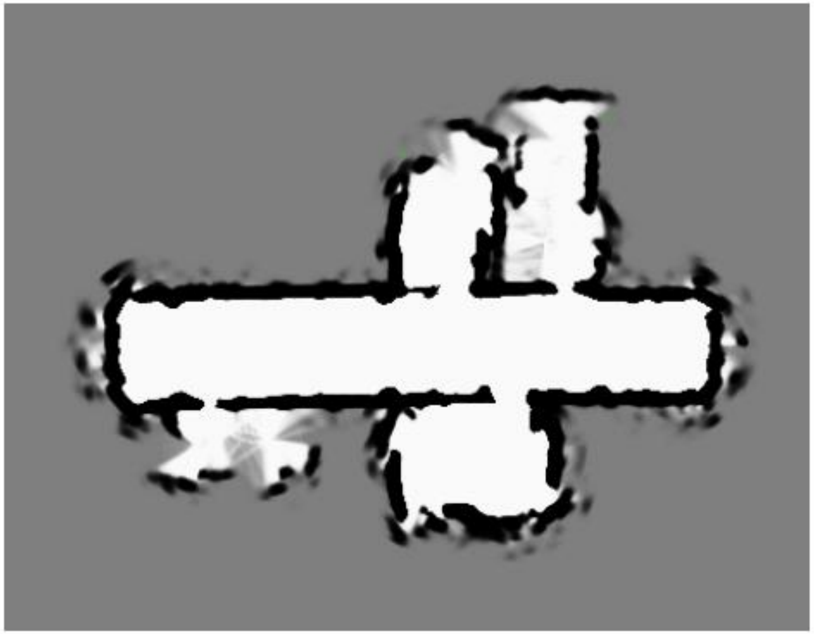
Incremental updating of occupancy grids, example

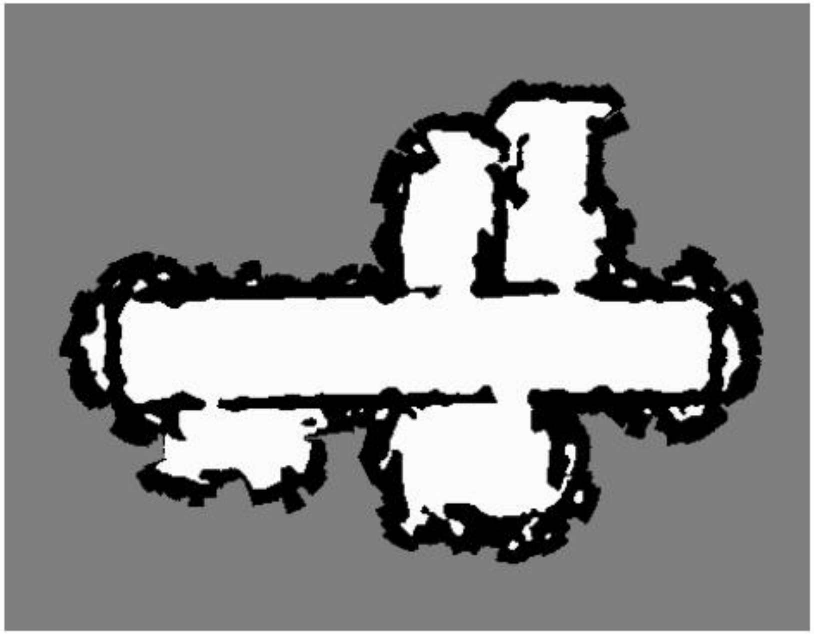


Resulting map obtained with ultrasound sensors



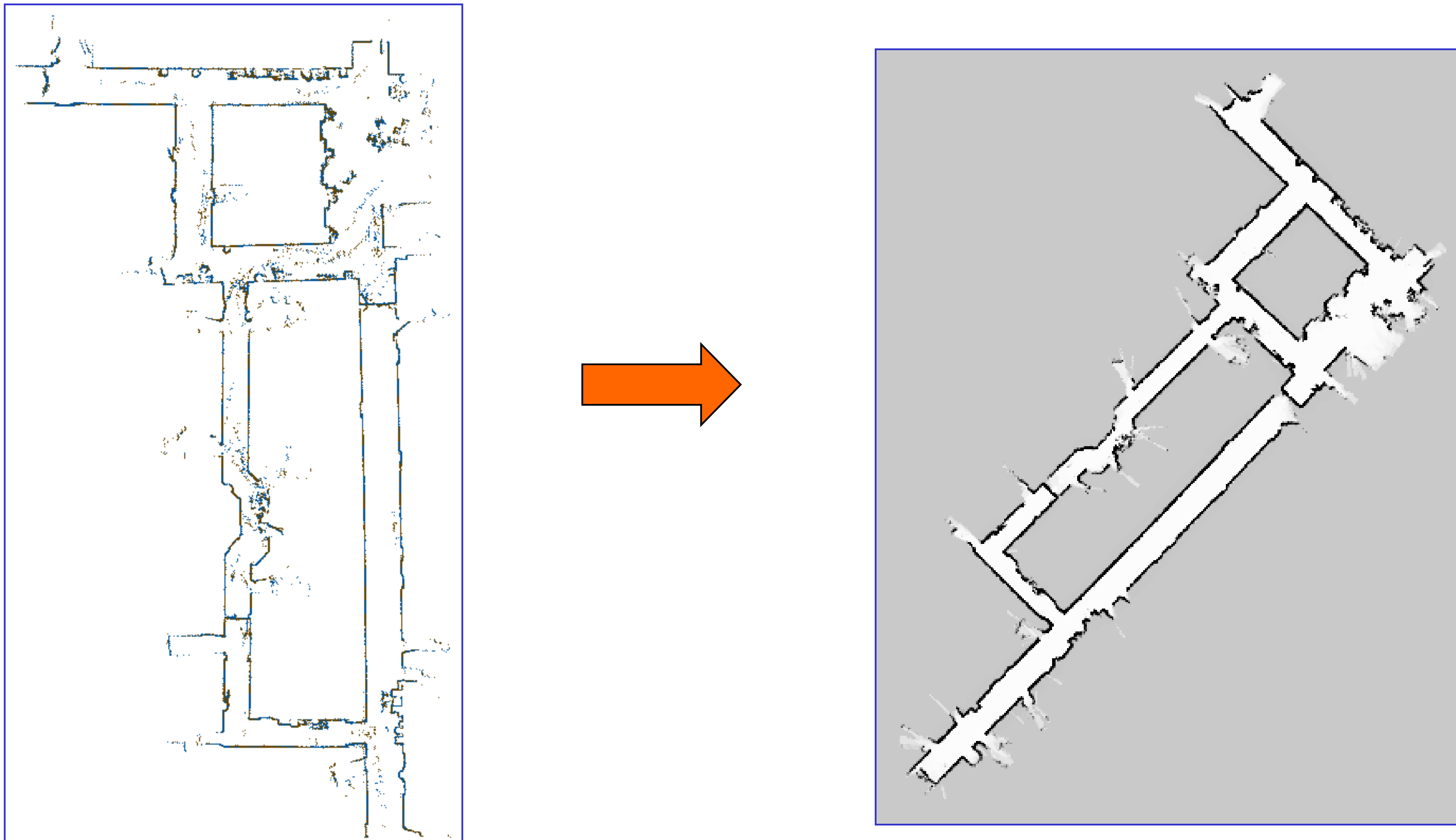
Resulting occupancy and maximum likelihood map



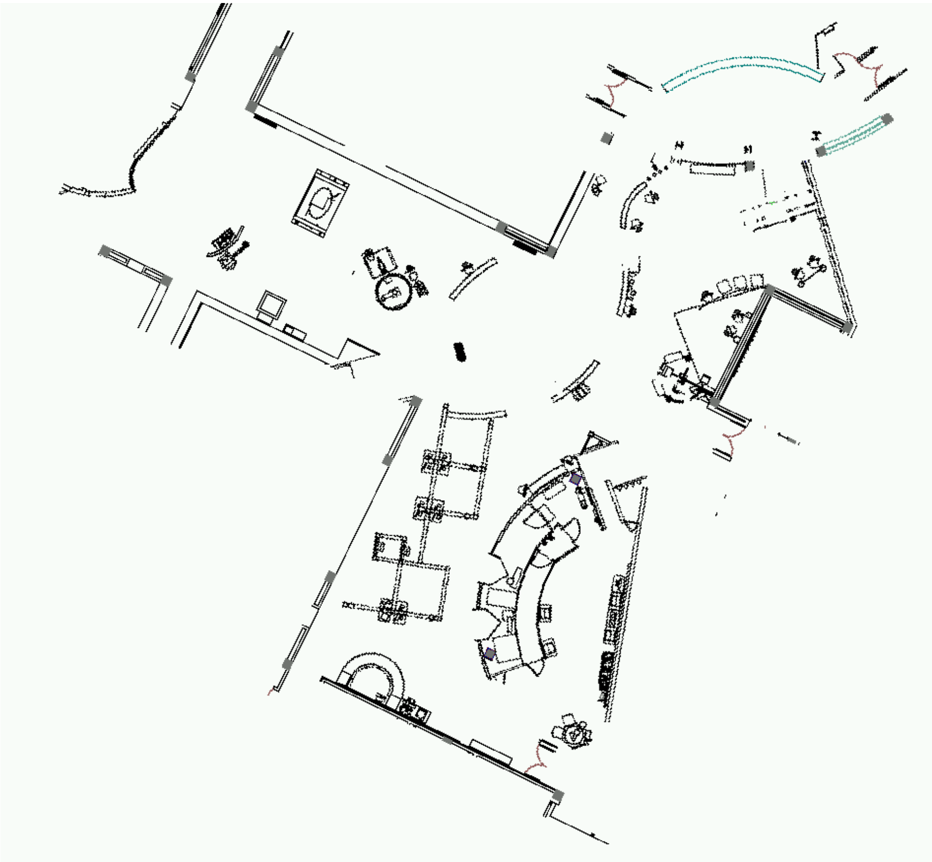


The maximum likelihood map is obtained by clipping the occupancy grid map at a threshold of 0.5.

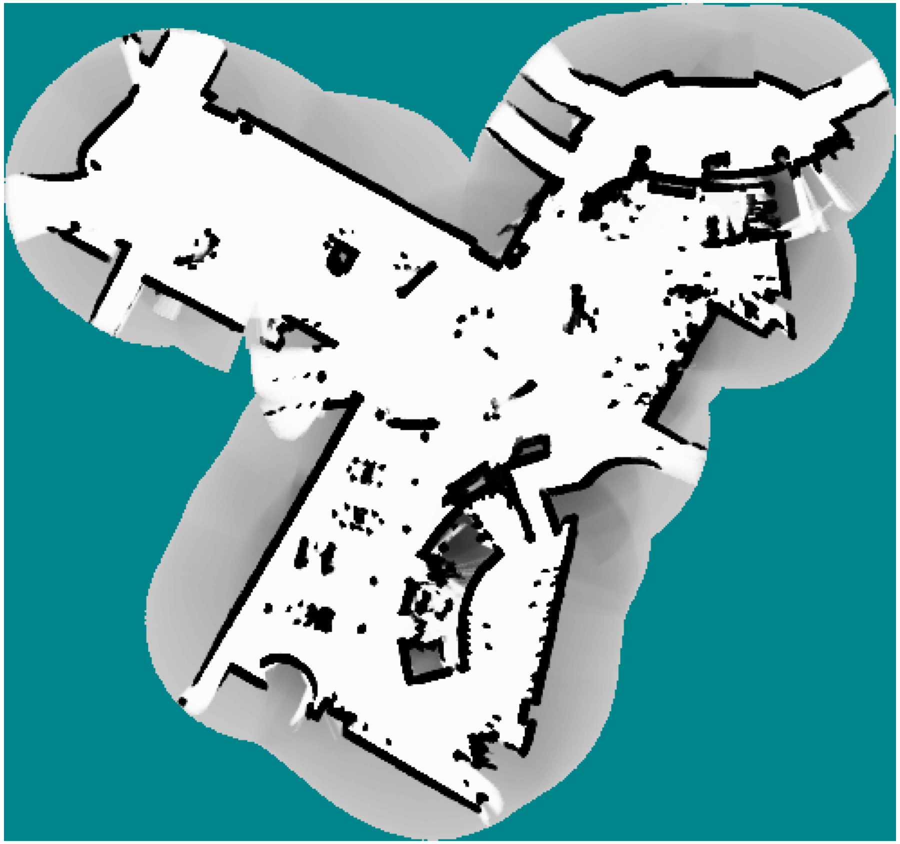
Occupancy grids from scans to maps



Tech museum, San Jose



CAD map



occupancy grid map

Alternative: Simple counting

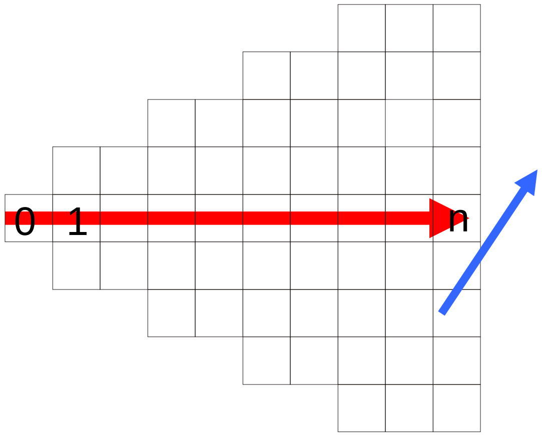
- For every cell count
  - hits(x,y): number of cases where a beam ended at (x,y).
  - misses(x,y): number of cases where a beam passed through (x,y).

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

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

The measurement model

1. pose at time t: x_t
2. beam n of scan t: $z_{t,n}$
3. maximum range reading: $\varsigma_{t,n} = 1$
4. beam reflected by an object: $\varsigma_{t,n} = 0$



$$m_{f(x_t, n, z_{t,n})}$$

$$p(z_{t,n} \mid x_t, m) = \begin{cases} \prod_{k=0}^{z_{t,n}-1} (1 - m_{f(x_t,n,k)}) & \text{if } \varsigma_{t,n} = 1 \\ m_{f(x_t,n,z_{t,n})} \prod_{k=0}^{z_{t,n}-1} (1 - m_{f(x_t,n,k)}) & \text{if } \varsigma_{t,n} = 0 \end{cases}$$

Computing the most likely map

- Compute values for m that maximize

$$m^* = \underset{m}{\arg\max}\, P(m \mid z_1, \ldots, z_t, x_1, \ldots, x_t)$$

- Assuming a uniform prior probability for $P(m)$, this is equivalent to maximizing (application of Bayes rule)

$$\begin{aligned} m^* &= \underset{m}{\arg\max}\, P(z_1, \ldots, z_t \mid m, x_1, \ldots, x_t) \\ &= \underset{m}{\arg\max} \prod_{t=1}^{T} P(z_t \mid m, x_t) \\ &= \underset{m}{\arg\max} \sum_{t=1}^{T} \ln P(z_t \mid m, x_t) \end{aligned}$$

Computing the most likely map

$$m^* = \arg\max_m \left[\sum_{j=1}^{J} \sum_{t=1}^{T} \sum_{n=1}^{N} \Big(I(f(x_t, n, z_{t,n}) = j) \cdot (1 - \varsigma_{t,n}) \cdot \ln m_j \right.$$

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

Suppose

$$\alpha_j = \sum_{t=1}^{T} \sum_{n=1}^{N} I(f(x_t, n, z_{t,n}) = j) \cdot (1 - \varsigma_{t,n})$$

$$\beta_j = \sum_{t=1}^{T} \sum_{n=1}^{N} \left[\sum_{k=0}^{z_{t,n}-1} I(f(x_t, n, k) = j) \right]$$

Meaning of α_j and β_j

$$\alpha_j = \sum_{t=1}^{T} \sum_{n=1}^{N} I(f(x_t, n, z_{t,n}) = j) \cdot (1 - \varsigma_{t,n})$$

corresponds to the number of times a beam that is not a maximum range beam ended in cell j (*hits(j)*)

$$\beta_j = \sum_{t=1}^{T} \sum_{n=1}^{N} \left[\sum_{k=0}^{z_{t,n}-1} I(f(x_t, n, k) = j) \right]$$

corresponds to the umber of times a beam intercepted cell j without ending in it (*misses(j)*).

Computing the most likely map

We assume that all cells m_j are independent:

$$m^* = \underset{m}{\arg\max} \left(\sum_{j=1}^{J} \alpha_j \ln m_j + \beta_j \ln(1 - m_j) \right)$$

If we set

$$\frac{\partial m}{\partial m_j} = \frac{\alpha_j}{m_j} - \frac{\beta_j}{1 - m_j} = 0$$

we obtain

$$m_j = \frac{\alpha_j}{\alpha_j + \beta_j}$$

Computing the most likely map amounts to counting how often a cell has reflected a measurement and how often it was intercepted.

Difference between occupancy grid maps and counting

- The counting model determines how often a cell reflects a beam.
- The occupancy model represents whether or not a cell is occupied by an object.
- Although a cell might be occupied by an object, the reflection probability of this object might be very small.

Example of the occupancy map

Example the reflection map

Example

- Out of 1000 beams only 60% are reflected from a cell and 40% intercept it without ending in it.
- Accordingly, the reflection probability will be 0.6.
- Suppose *p(occ | z)* = 0.55 when a beam ends in a cell and *p(occ | z)* = 0.45 when a cell is intercepted by a beam that does not end in it.
- Accordingly, after n measurements we will have

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

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

Summary, occupancy grid

- 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.
- The cell the posterior probability that the corresponding area in the environment is occupied.
- Occupancy grid maps can be learned efficiently using a probabilistic approach.
- Reflection maps are an alternative representation.
- The reflection map stores in each cell the probability that a beam is reflected by this cell.
- We provided a sensor model for computing the likelihood of measurements and showed that the counting procedure underlying reflection maps yield the optimal map.

Line maps

- Suitable for man-made structures as common indoor scenes.
- Parametric representation (unlike non-parametric occupancy grid).
- Advantages:
  - Substantially less memory than grids.
  - Higher accuracy because they do not suffer from discretization problem.
- Disadvantages:
  - No closed-form solution for situation when data points correspond to multiple linear structures.
  - How many lines there are? $\Rightarrow$ Data association problem.

Line fitting, least squares

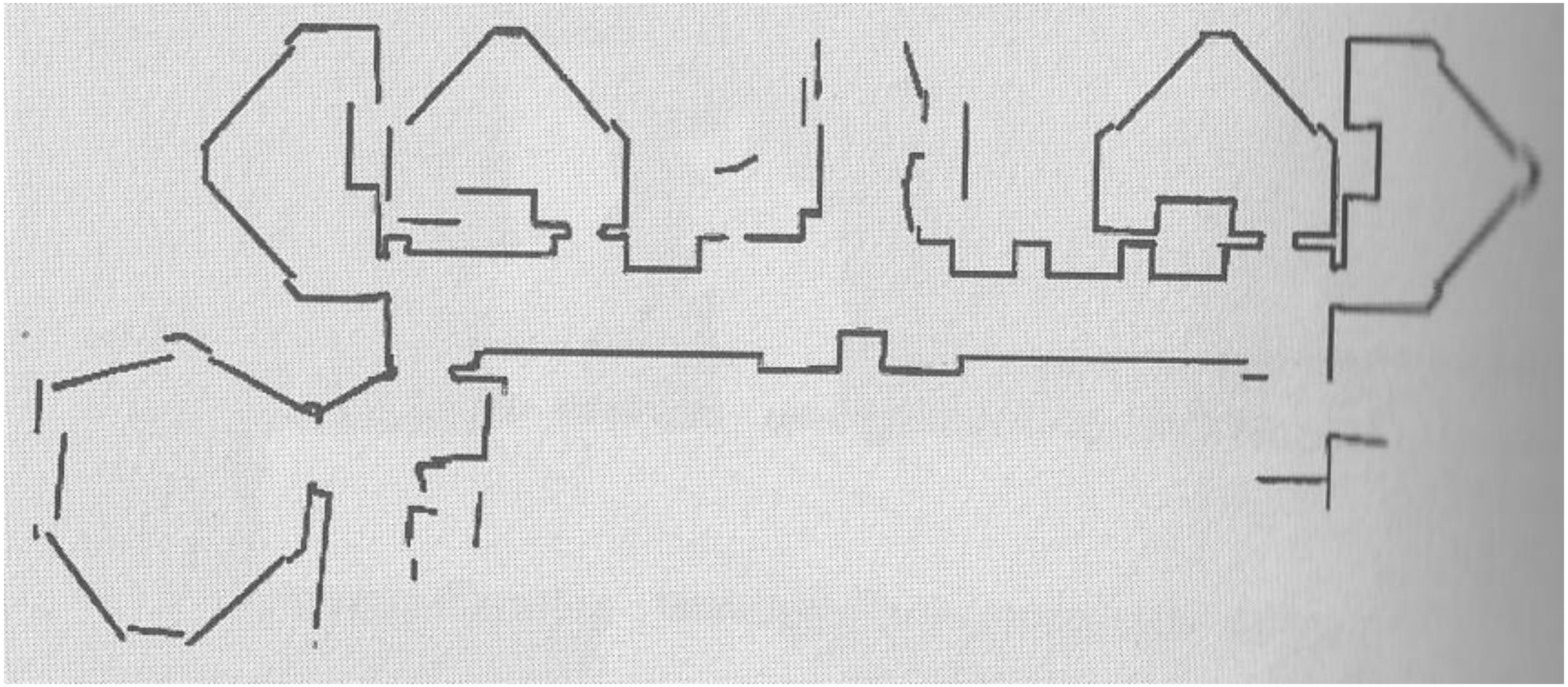
- Data points x_i, y_i.
- The closed-form line approximation

$$\bar{x} = \sum_i x_i, \quad \bar{y} = \sum_i y_i$$

$$\tan 2\phi = \frac{-2\sum_i (\bar{x}-x_i)(\bar{y}-y_i)}{\sum_i ((\bar{y}-y_i)^2-(\bar{x}-x_i)^2)}$$

$$r = \bar{x}\cos\phi + \bar{y}\sin\phi$$

Line map, an example



94 lines, example from Handbook of Robotics, Springer 2008