Mapping with Known Pose*

Prof. Mohan Sridharan
Chair in Robot Systems
University of Edinburgh, UK
https://homepages.inf.ed.ac.uk/msridhar/
m.sridharan@ed.ac.uk

*Revised original slides that accompany the book: PR by Thrun, Burgard and Fox.

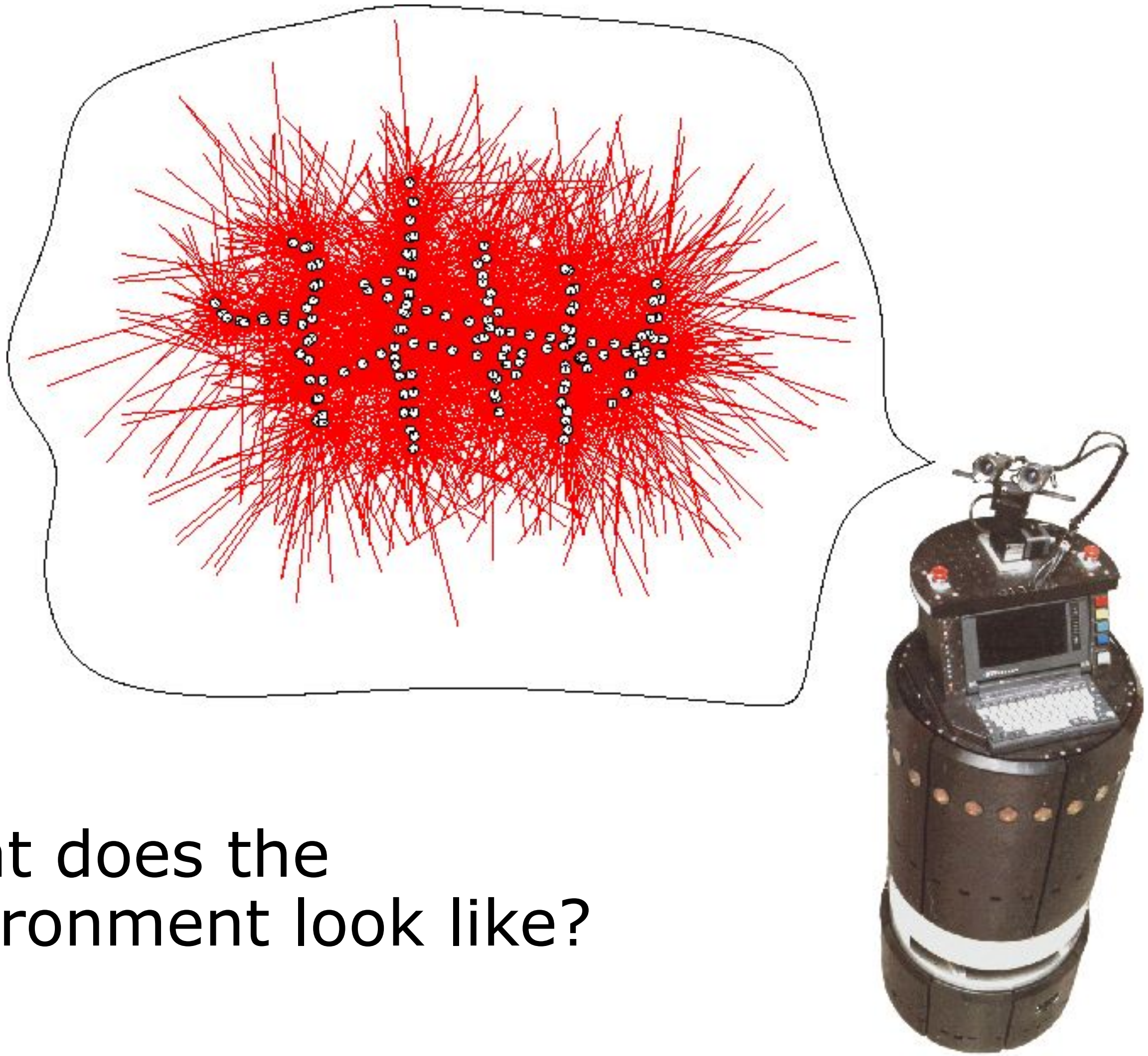
Agenda...

- Introduction to mapping:
  - Motivate SLAM.
  - Mapping with known poses.
  - Some examples.
  - Simple counting-based mapping.
  - MAP estimation.

Why Mapping?

- Maps are a fundamental requirement:
  - Provides a frame of reference to humans and robots!
- Maps used for localization, path-planning, activity-planning, active-sensing…
- Autonomous behavior requires *Simultaneous Localization And Mapping* (SLAM).

The General Problem of Mapping



What does the
environment look like?

Mathematical Formulation

- Formally, given the sensor data:

$$d = \{u_{1:t}, z_{1:t}\}$$

  mapping involves finding the most likely map (mode):

$$p(m \mid z_{1:t}, u_{1:t})$$

- Finding full posterior is easier with independence assumption:

$$Bel(m) = p(m \mid z_{1:t}, u_{1:t})$$

Mapping is a Chicken and Egg Problem

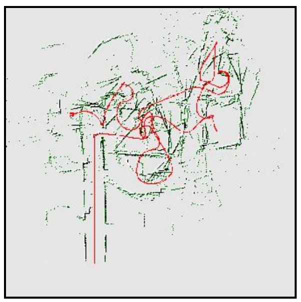
- Localization: estimate robot pose given sensor data and map.
- Goal: mobile robots that require little/no human supervision.
- Challenge: simultaneously estimate robot pose and the map (SLAM).
  - Bootstrap off localization and map-building with known pose.

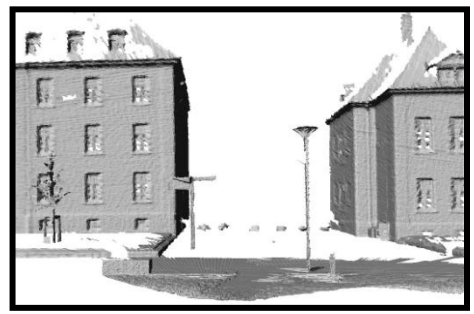
Problems in Mapping

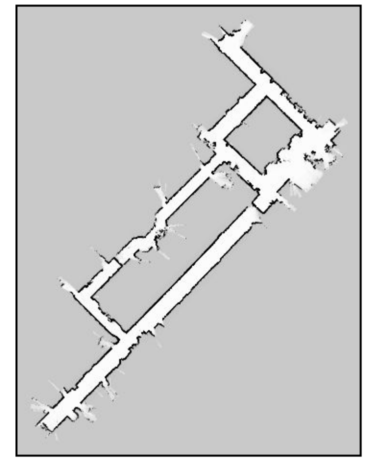
- Noise in sensing and actuation:
  - Extract information from noisy sensory data?
  - Model and account for motion error accumulation?
- Ambiguity in perception:
  - Establish correspondence between sensor readings?
- Data association:
  - Identify that robot is at a previously visited place?
  - Close the loop?
- Large continuous search space:
  - Binary map with N grid-cells represents 2^N maps!!

Types of SLAM-Problems

- Grid maps or scans:

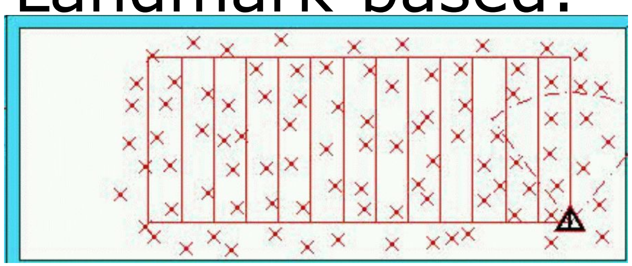


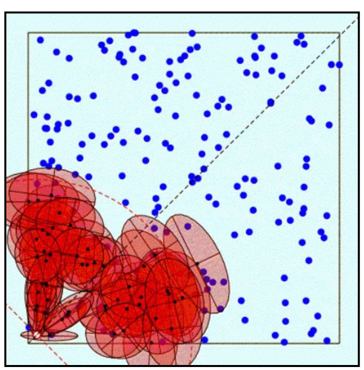


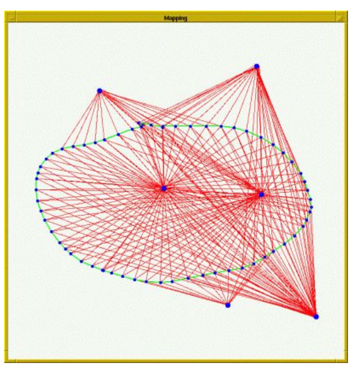


[Lu & Milios, 97; Gutmann, 98: Thrun 98; Burgard, 99; Konolige & Gutmann, 00; Thrun, 00; Arras, 99; Haehnel, 01;…]

- Landmark-based:







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

Occupancy Grid Maps (Moravec and Elfes, 1985)

- Environment a collection of grid cells.
- Estimate the probability that a cell is occupied.

- **Key assumptions:**
  - Occupancy of individual cells ($m^{[ij]}$) is independent!

$$Bel(m_t) = p(m_t \mid z_{1:t}, u_{1:t}) = \prod_{i,j} Bel(m_t^{[ij]})$$

$$Bel(m_t^{[ij]}) = p(m_t^{[ij]} \mid z_{1:t}, x_{1:t})$$

  - Robot poses are known!
  - Post-processing tool…

Updating Occupancy Grid Maps

- **Idea:** Update each individual cell within field-of-view using a binary Bayes filter.

- **Additional assumption:** Map is static i.e. control commands can be neglected!!

$$Bel(m_t^{[ij]}) = p(m_t^{[ij]} \mid z_{1:t}, x_{1:t})$$

Binary Bayes Filter (Chapter 4, PR)…

- Elegant, avoids numerical errors.
- Chapter 9 (Table 9.1); also see Table 4.2. PR.

$$l_t = l_{t-1} + \log\frac{p(x \mid z_t)}{1 - p(x \mid z_t)} - \log\frac{p(x)}{1 - p(x)}$$

$$l_0(x) = \log\frac{p(x)}{1 - p(x)} = \log \text{odds}(x), \quad Bel_t(x) = 1 - \frac{1}{1 + \exp(l_t)}$$

$$l_{t,i} = \log\frac{p(m_i \mid z_{1:t}, x_{1:t})}{1 - p(m_i \mid z_{1:t}, x_{1:t})}, \quad l_0 = \log\frac{p(m_i)}{1 - p(m_i)}$$

$$p(m_i \mid z_{1:t}, x_{1:t}) = 1 - \frac{1}{1 + \exp(l_{t,i})}$$

Updating Occupancy Grid Maps

- Update the map cells using the *inverse sensor model*:

$$\log \frac{P\left(m_t^{[ij]} \mid z_t, x_t\right)}{1 - P\left(m_t^{[ij]} \mid z_t, x_t\right)}$$

- Log-odds of occupancy of grid cell, given the current measurement and known pose.
- Information about the world conditioned on measurements caused by the world.
- Reasons from effects to causes; hence ”inverse”. Adhoc approach: see **Table 9.2**, **PR**.

Inverse Sensor Model Operation

- Determine beam index k and range, consider any given grid cell m_i.
- Consider robot pose (x_t) to compute whether cell is in range of beam.
- If cell outside range (+threshold), return prior log likelihood (l_0).
- If distance to cell less than measured range, consider cell to the free (l_{free}).
- Otherwise return log-likelihood value of being occupied (l_{occ}).

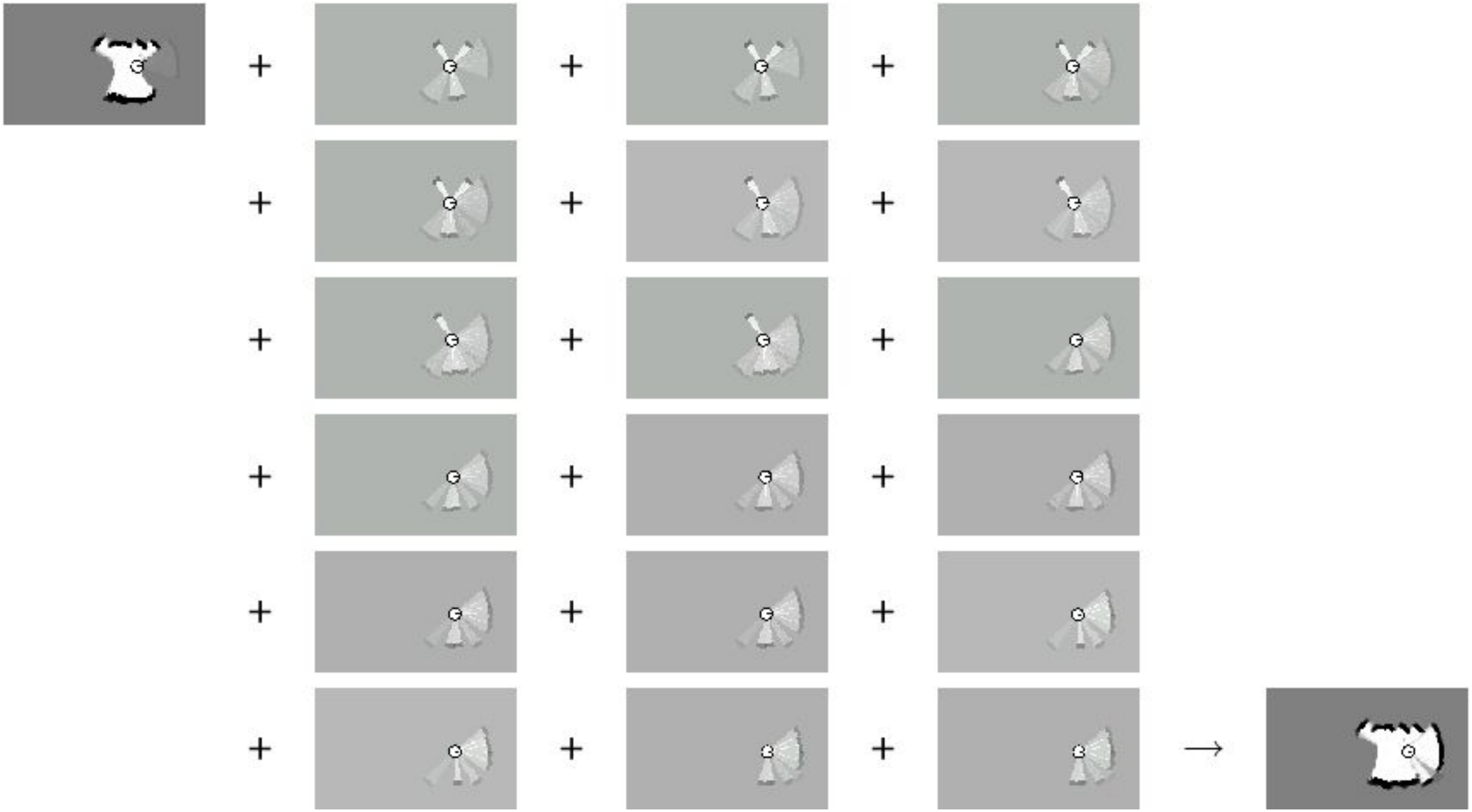
Function Approx. Inv. Sensor Model

- Sampling-based approach.
- Approach:
  - Sample a map from feasible maps: $m^{[k]} \sim p(m)$
  - Sample a robot pose in map: $x^{[k]}$
  - Sample a measurement given map and pose: $z^{[k]}$
  - Get ground-truth occupancy value from map: $occ(m^{[ij][k]})$
  - Learn predictor that minimizes error over data samples.

$$< x^{[k]}, z^{[k]} > \rightarrow occ(m^{[ij][k]}), \; k = [1, N]$$

- Use relative pose, model sensor characteristics.

Incremental Updating of Occupancy Grids (Example)



Recap...

- Goal: simultaneously localize and map (SLAM); start with mapping

$$p(m \mid z_{1:t}, u_{1:t})$$

- Occupancy Grid mapping: assume known pose and independence:

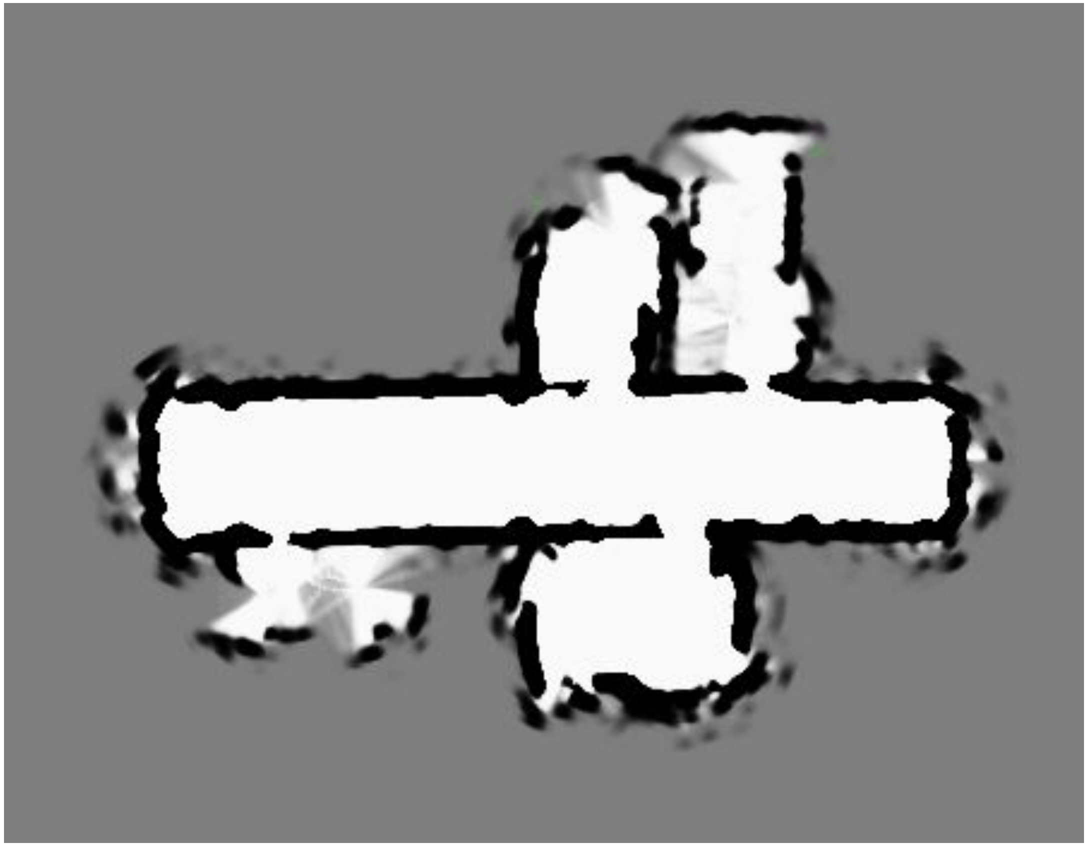
$$Bel(m_t) = p(m_t \mid z_{1:t}, u_{1:t})$$

$$Bel(m_t) = \prod_{i,j} Bel(m_t^{[ij]}) = \prod_{i,j} p(m_t^{[ij]} \mid z_{1:t}, x_{1:t})$$

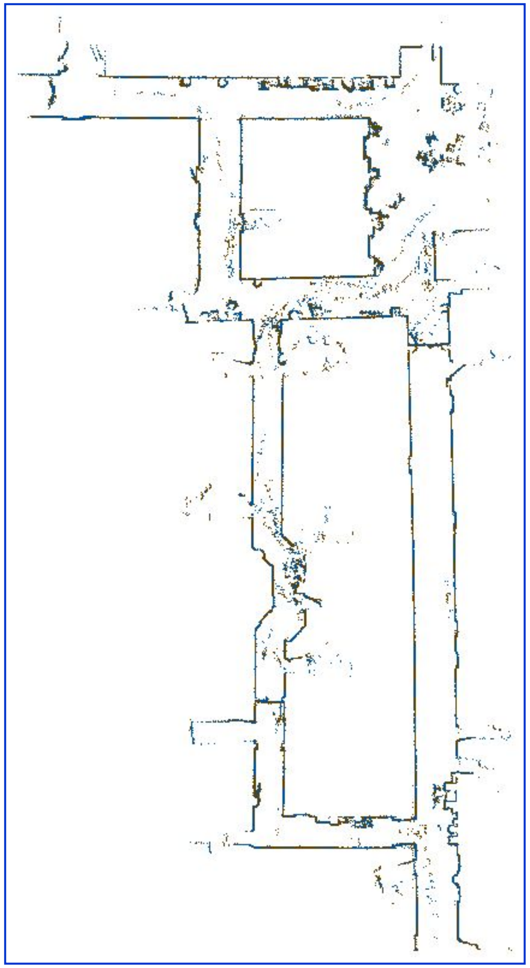
- Binary Bayes filter, inverse sensor models to update cells.
- Ad hoc and sampling-based approaches for inverse sensor model.

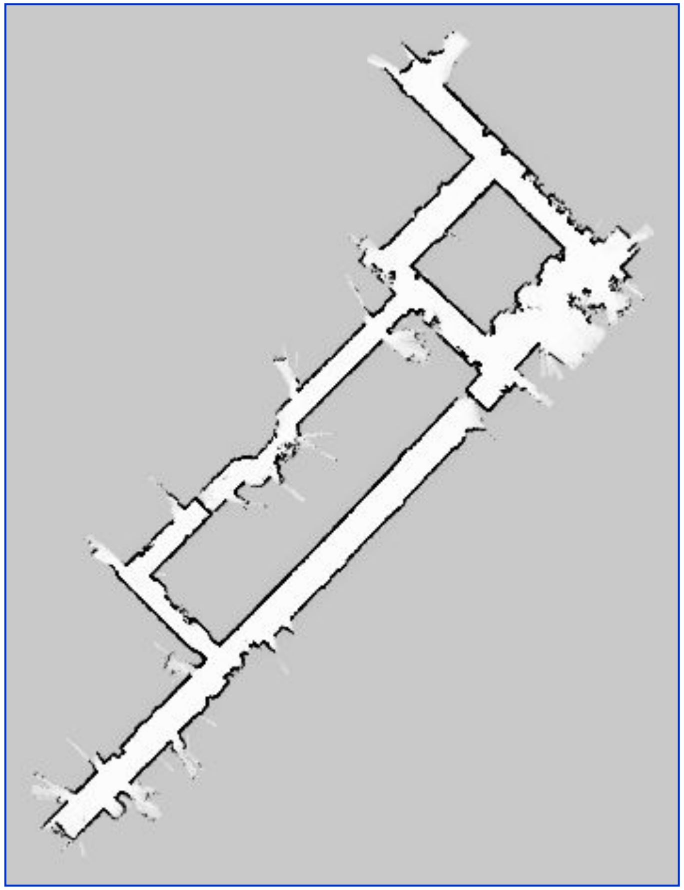
$$\log \frac{P(m_t^{[ij]} \mid z_t, x_t)}{1 - P(m_t^{[ij]} \mid z_t, x_t)}$$

Resulting Map Obtained with Ultrasound Sensors

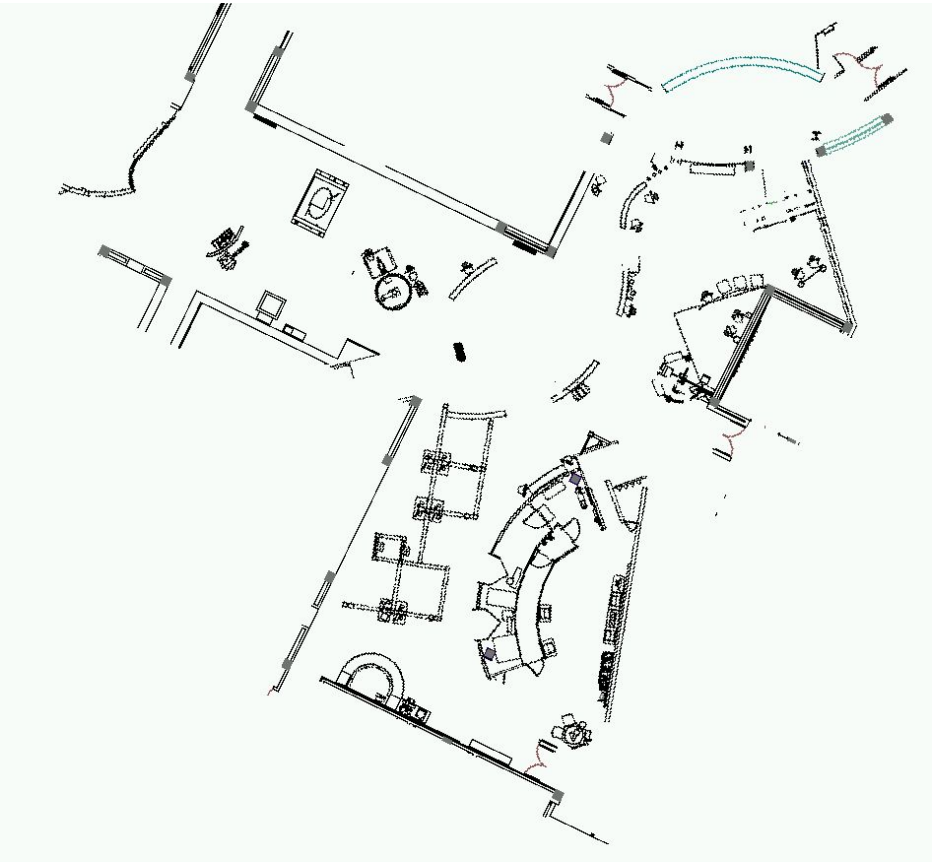


Occupancy Grids: From scans to maps

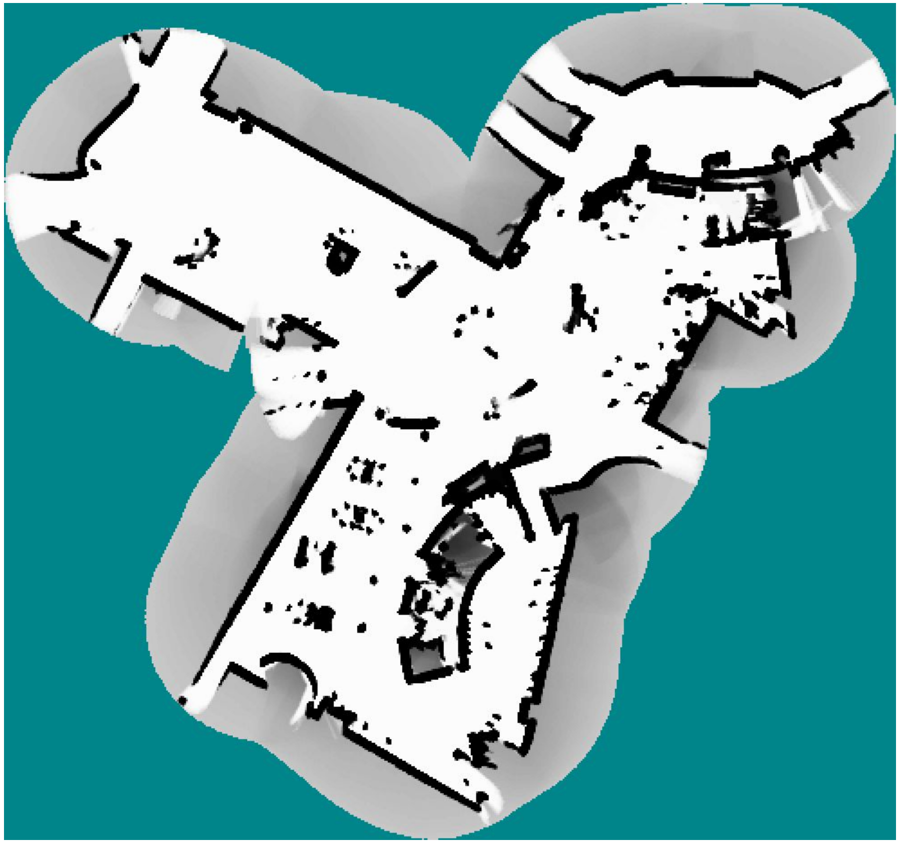




Tech Museum, San Jose



CAD map



occupancy grid map

Agenda...

- Introduction to mapping:
  - Motivate SLAM.
  - Mapping with known poses.
  - Some examples.
  - Simple counting-based mapping.
  - MAP estimation.

Alternative: Simple Counting

- For every cell count:
  - **hits(i, j):** no. of times a beam ended at <i, j>.
  - **misses(i, j):** no. of times a beam passed through <i, j>

$$Bel(m^{[ij]}) = \frac{\mathrm{hits}(i,j)}{\mathrm{hits}(i,j) + \mathrm{misses}(i,j)}$$

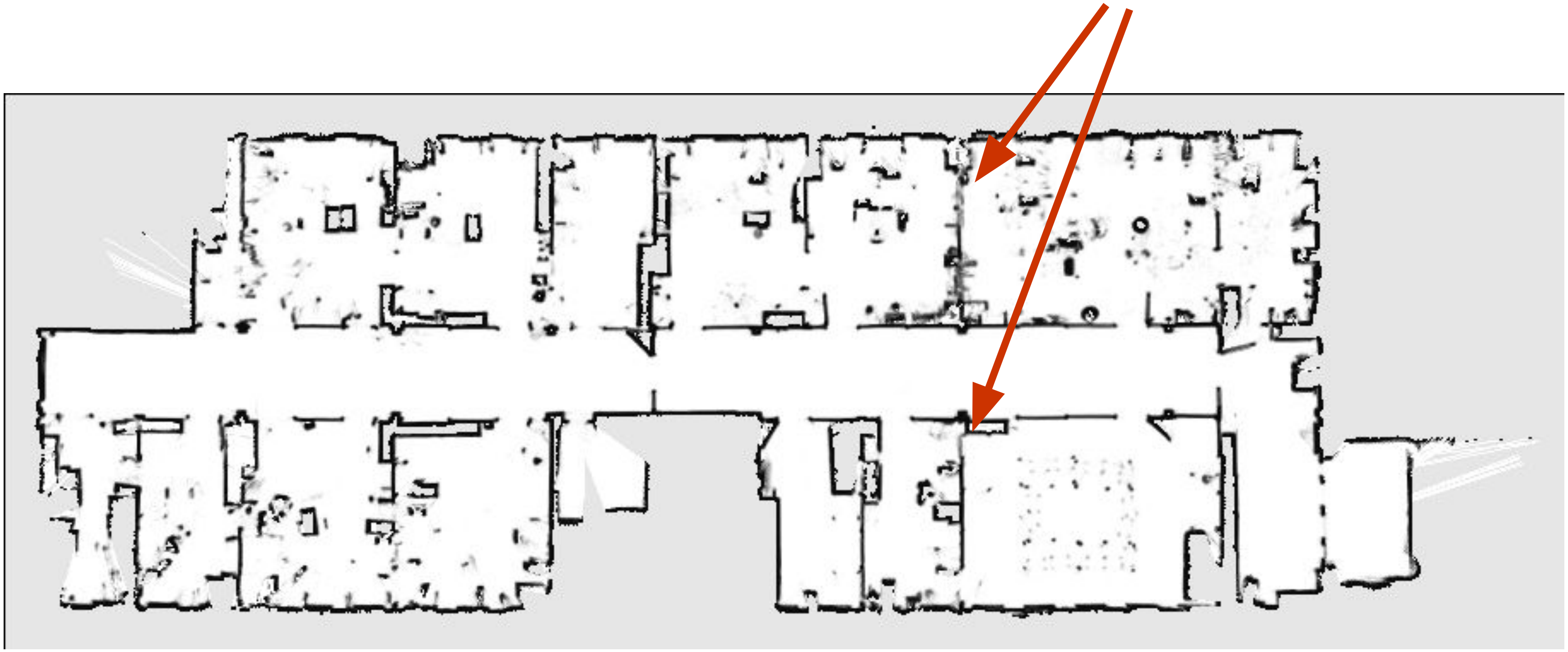
- Value of interest: $P(reflects(i, j))$
- Count how often a cell has reflected a beam.

Difference between Occupancy Grid Maps and Counting

- The counting model (with MAP) determines how often a cell reflects a beam:
  - No inverse sensor model ☺
  - Store all data i.e. incremental updates not possible ☹
- The occupancy model represents whether or not a cell is occupied by an object.
  - Incremental updates possible ☺
  - Inverse sensor models, independence assumption ☹
- Although a cell might be occupied by an object, it says nothing about the reflection probability.

Example Occupancy Map

Example Reflection Map



Summary

- Occupancy grid maps a popular approach to represent the environment of a mobile robot given known poses.
- Stores the posterior probability that the corresponding area in the environment is occupied.
- Occupancy grid maps can be learned efficiently considering each cell independently from all others.

- Reflection maps are an alternative representation; each cell stores probability that a beam is reflected by cell.
- Reflection maps are more optimal.
- MAP approach relaxes independence constraint but requires batch processing.