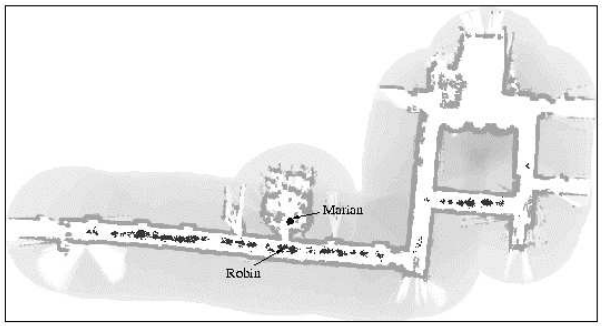
590DF
Introduction to AI-based Mobile Robotics

Some more particle filters

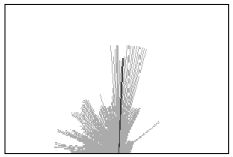
Mapping

Multi-robot Localization: Idea



Robot Detection

Camera image



Laser scan

Multi-robot Localization

- Combined belief state $X = X_1 \otimes X_2 \otimes \ldots \otimes X_N$ has dimension $3N$ ⟶ complexity grows **exponentially** in number of robots
- **Factorial** representation of the belief

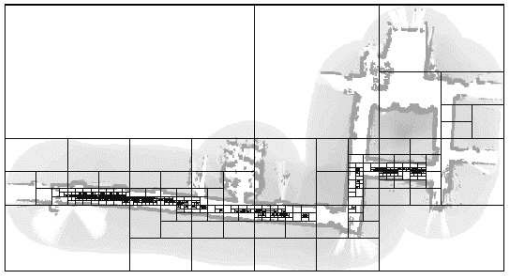
$$P(X_1^{(t)},\ldots,X_N^{(t)} \mid d) = P(X_1^{(t)} \mid d) \cdot \ldots \cdot P(X_N^{(t)} \mid d)$$

- Perform localization for each robot and use detections to **constrain the beliefs**

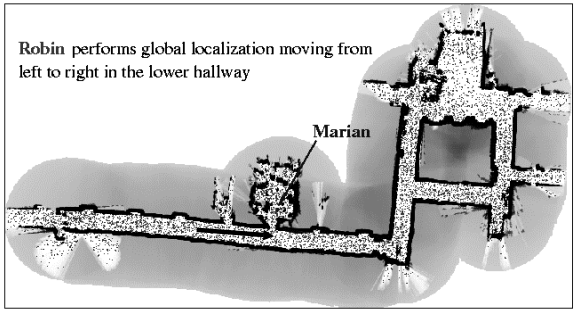
Density Trees

Integration of robot detection requires a **density**

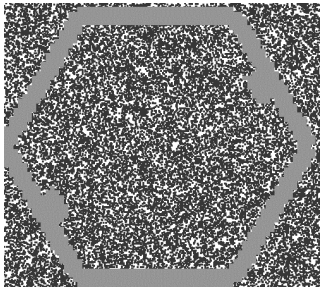
$$Bel(x_m) \leftarrow \int P(x_m \mid x_n, r_n)\, Bel(x_n)\, dx_n \cdot Bel(x_m)$$



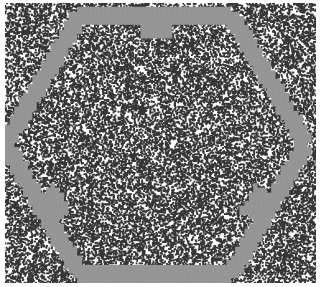
Example Run



Heterogeneous Robots

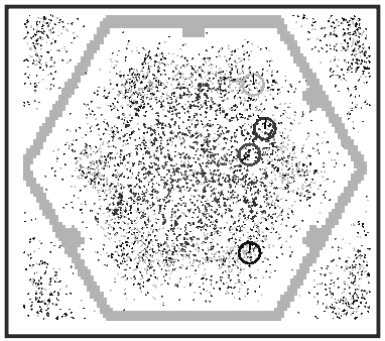


Laser



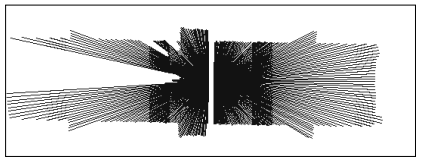
Sonar

Example Run



People Tracking

- **Key questions**
  - How many people are there?
  - Where do they go?
- **Requirements**
  - Real time
  - No model of the environment
  - Robot in motion



[Schulz et al. 2001]

Tracking Multiple Objects

- **Kalman filters**
  - MHT (Multiple hypothesis tracking)

    JPDAF (Joint probabilistic data association filters)
    - Efficient, especially JPDAF
    - Track individual objects using Kalman filters
- **Particle filters**
  - One sample set for multiple objects
    - Efficient
    - Brittle, introduces dependencies
  - Joint state space
    - Exact representation
    - Exponential state space

Multi Object Tracking

- **Idea**: Track each person with an individual particle filter.
- Data association: Which observation corresponds to which object/sample set?
  - Indistinguishable features
  - Missing features
  - False alarms
- Track management: When to add or delete tracks/sample sets?

Joint Probabilistic Data Association

- Solve data association problem
  - Compute assignment probabilities of features to objects
  - Consider missing features by introducing special feature z_0
  - Handle false alarms by Poisson distribution
- Track management outside JPDAF

Assignment Probabilities

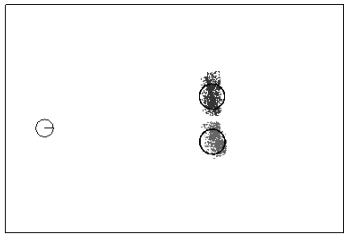
- Consider all possible assignments Θ of features to sample sets.
- Compute marginal probability β_{ji} of feature z_j originating from sample set x_i.

$$\beta_{ji} = \sum_{\theta \in \Theta_{ji}} P(\theta \mid Z^t)$$

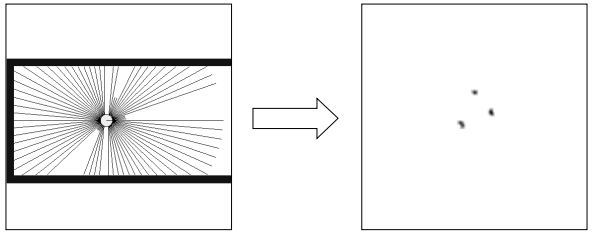
$$P(\theta \mid Z^t) = \eta \, P(\theta \mid X^t) \prod_{(j,i) \in \theta} p(z_j^t \mid x_i^t)$$

Particle Filter + JPDAF

$$\omega_{i,n}^t = \eta \sum_{j=0}^{m_t} \beta_{ji} \, p(z_j^t \mid x_{i,n}^t)$$

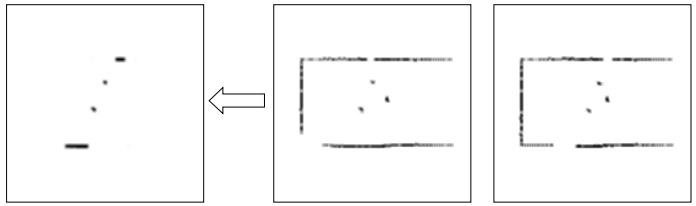


Laser-based People Tracking



- Legs appear as local minima in range scans.
- Transform such features into a position probability grid indicating P(legs$_{x,y}$).

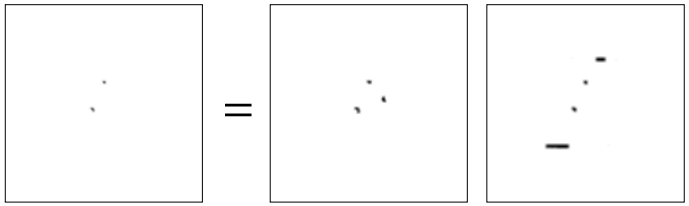
Observing Movements



Difference between consecutive range scans:

$$P(new_{x,y}^t) = P(occ_{x,y}^t) \cdot (1 - P(occ_{x,y}^{t-1}))$$

Integrating Observations



$$P(z_j^t \mid x_{i,n}^t) = P(\text{legs}_{x_{i,n}}^t) \cdot P(\text{new}_{x_{i,n}}^t)$$

Handling Occlusions

Reasons for a missing feature z^t_0:

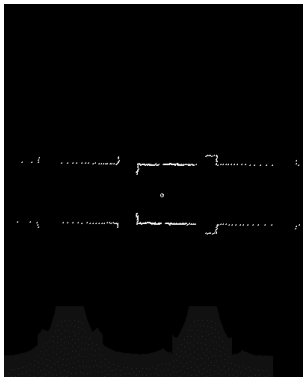
- Feature **not detected** $P(\neg\text{Detect})$
- Feature **occluded** by other object $P(\text{occluded}(x_i^t))$

Idea

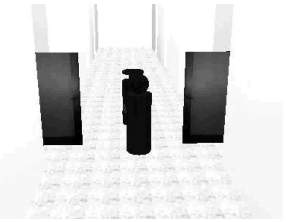
- Compute **occlusion grid**

$$P(z_0^t \mid x_i^t) = P(\text{occluded}(x_i^t) \vee \neg\text{Detect})$$

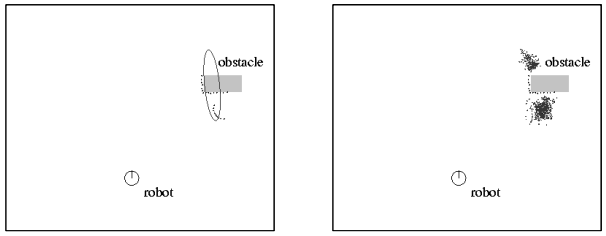
Example Run



Tracking with a Moving Robot

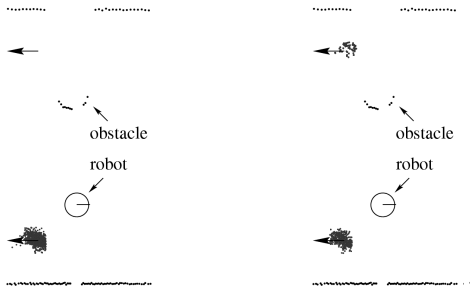


Comparison to Kalman Filters



Kalman Filter Particle Filter

One Particle Filter vs. SJPDAF

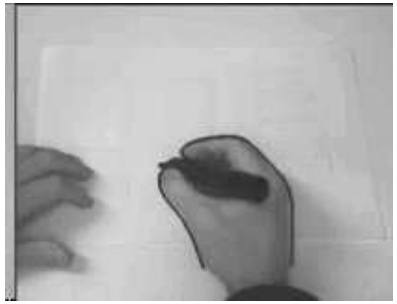


One filter SJPDAF

Condensation

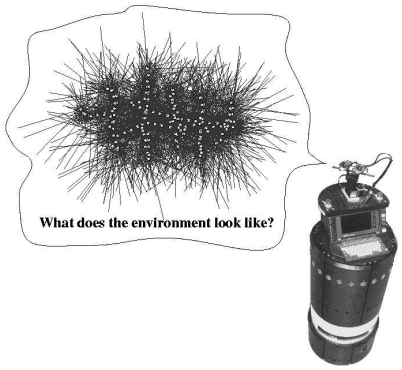
[Isard et al. 96]

Mixed-State Condensation

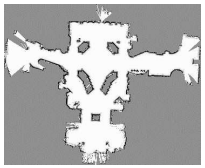
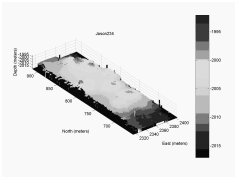
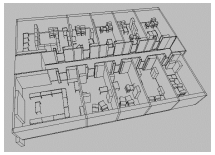
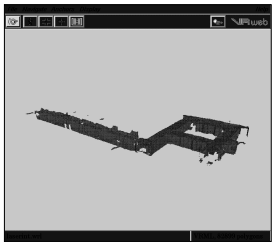
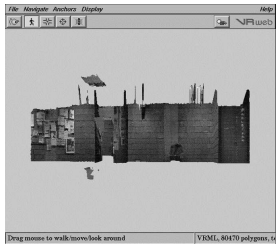


[Isard et al. 96]

Mobile Robot Mapping



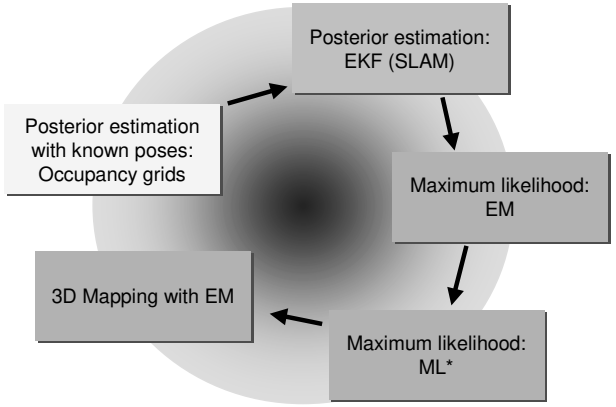
Some Maps

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 known
  - How can we estimate them during mapping?

Mapping: Outline



Occupancy Grid Maps

- Introduced by Moravec and Elfes in 1985
- Represent environment by a grid.
- Estimate the probability that a location is occupied by an obstacle.
- Key assumptions
  - Occupancy of individual cells is independent

$$Bel(m_t) = P(m_t \mid u_1, z_2 \ldots, u_{t-1}, z_t) = \prod_{x,y} Bel(m_t^{[xy]})$$

  - 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

- Typically updated using inverse sensor model and odds ratio:

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

- Or log-odds ratio $\overline{B}$:

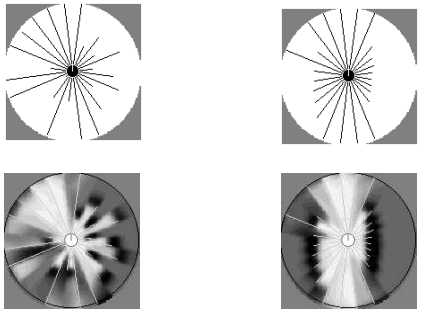
$$\overline{B}(m_t^{[xy]}) = \overline{B}(m_{t-1}^{[xy]}) + \ln P(m_t^{[xy]} \mid z_t, x_t) - \ln\left(1 - P(m_t^{[xy]} \mid z_t, x_t)\right) + \ln\left(1 - P(m_t^{[xy]})\right) - \ln P(m_t^{[xy]})$$

Occupancy Grid Algorithm

1. Algorithm **Occupancy_grid**(z_t, x_t):
2. If t = 0 then
3. For all <x,y> do
4. $Bel(m^{[xy]}) = P(m^{[xy]})$
5. For all <x,y> do
6. If robot is at <x,y> then
7. $Bel(m^{[xy]}) = 0$
8. else if <x,y> is in range(z_t, x_t) then
9. $Bel(m_t^{[xy]}) = 1 - \left(1 + \frac{P(m_t^{[xy]} \mid z_t, x_t)}{1 - P(m_t^{[xy]} \mid z_t, x_t)} \frac{1 - P(m_t^{[xy]})}{P(m_t^{[xy]})} \frac{Bel(m_{t-1}^{[xy]})}{1 - Bel(m_{t-1}^{[xy]})}\right)^{-1}$
10. Return $Bel(m)$

Learning Inverse Sensor Model

- Learn probabilities with neural network.
- Consider four beams simultaneously.



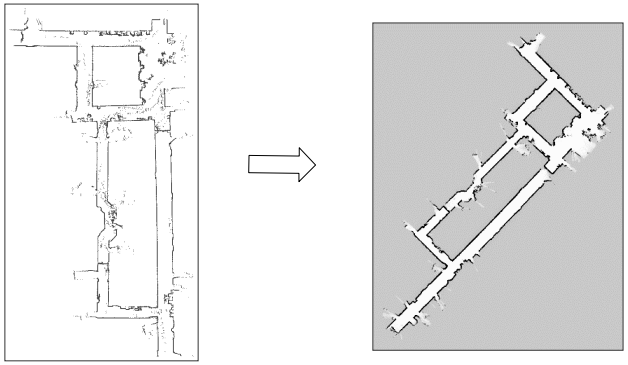
Alternative: Simple Counting

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

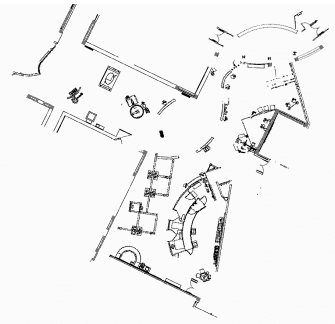
$$Bel(m^{[xy]}) = \frac{\text{hits}(x, y)}{\text{hits}(x, y) + \text{misses}(x, y)}$$

- Assumption: $P(occupied(x,y)) = P(reflects(x,y))$

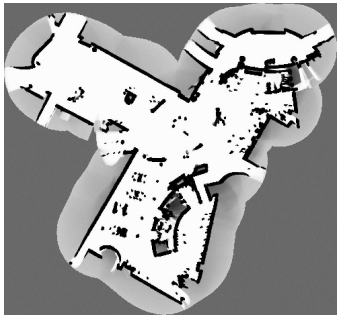
Occupancy Grids: From scans to maps



Tech Museum, San Jose



CAD map



occupancy grid map