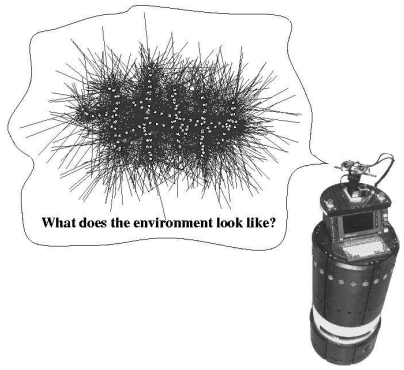
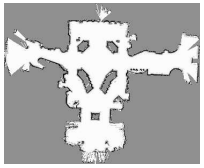
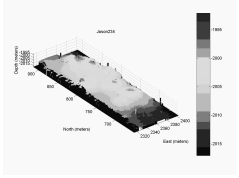
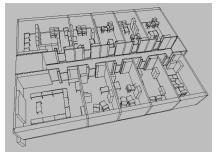
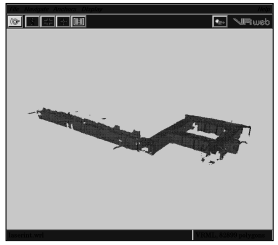
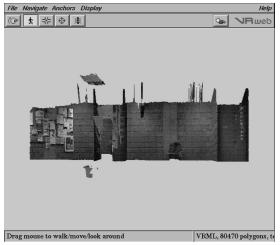
CSE-571
AI-based Mobile Robotics

Mapping

Mobile Robot Mapping



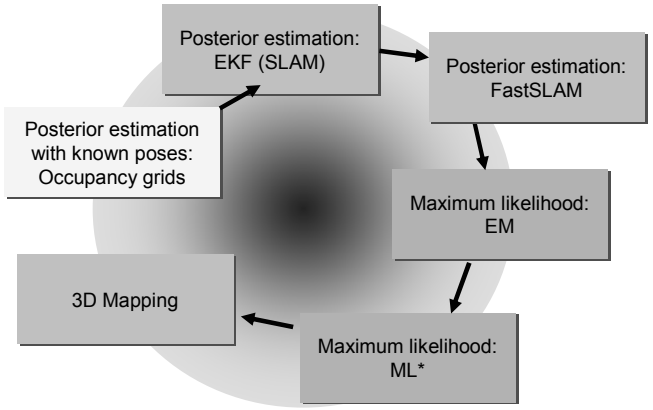
Some Maps

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 log odds ratio:

$$\hat{b}(m_t^{[xy]}) = \ln \frac{Bel(m_t^{[xy]})}{1 - Bel(m_t^{[xy]})}$$

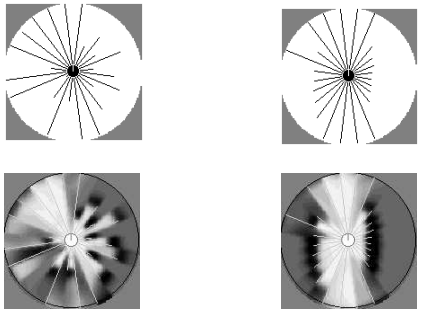
$$\hat{b}(m_t^{[xy]}) = \ln P(m_t^{[xy]} \mid z_t, x_t) - \ln\left(1 - P(m_t^{[xy]} \mid z_t, x_t)\right) + \ln\left(1 - P(m^{[xy]})\right) - \ln P(m^{[xy]}) + \hat{b}(m_{t-1}^{[xy]})$$

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

Learning Inverse Sensor Model

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



[Thrun '98]

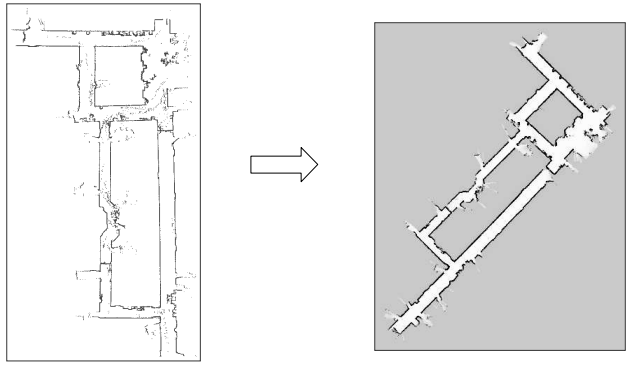
Alternative: Simple Counting

- 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>

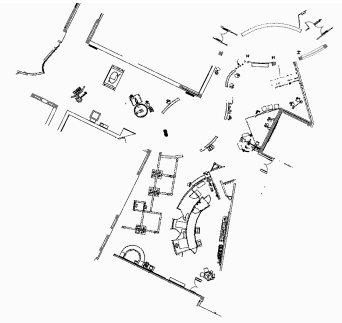
$$Bel(m^{[xy]}) = \frac{\text{hits}(x,y)}{\text{hits}(x,y) + \text{misses}(x,y)}$$

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

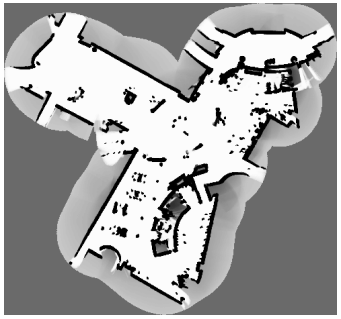
Occupancy Grids: From scans to maps



Tech Museum, San Jose

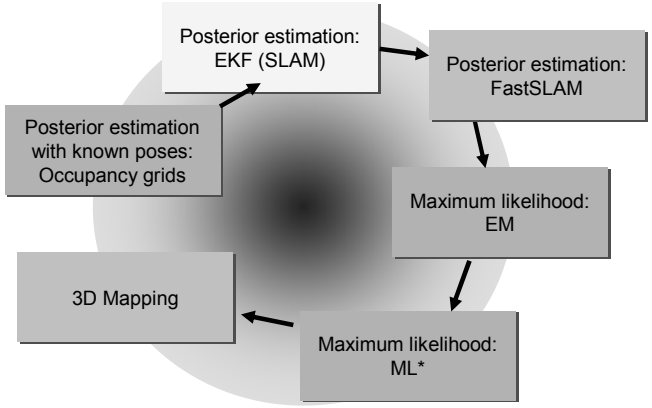


CAD map



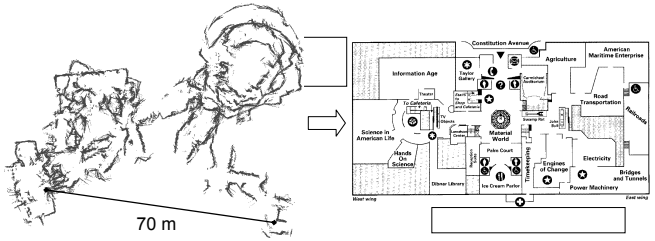
occupancy grid map

Mapping: Outline



Concurrent Mapping and Localization

- Chicken-and-egg problem
  - Mapping with known poses is "simple"
  - Localization with known map is "simple"
  - But in combination the problem is hard!



Mapping as Posterior Estimation

- Problem: Positions of robot and landmarks are not independent.
- Idea: Bayes filter over both robot positions and map.

$$Bel(m_t, x_t) = \eta\ p(z_t \mid m_t, x_t) \iint p(m_t, x_t \mid m_{t-1}, x_{t-1}, u_{t-1}) Bel(m_{t-1}, x_{t-1}) dm_{t-1} dx_{t-1}$$

- Static world assumption:

$$Bel(m_t, x_t) = \eta\ p(z_t \mid m_t, x_t) \iint p(x_t \mid x_{t-1}, u_{t-1}) Bel(m_{t-1}, x_{t-1}) dm_{t-1} dx_{t-1}$$

- Any problems?

SLAM: Mapping with Kalman Filters

- Map with N landmarks:(2N+3)-dimensional Gaussian

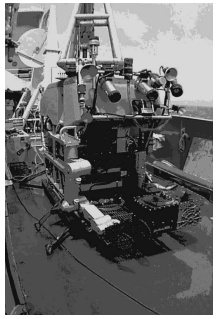
$$Bel(x_t, m_t) = \left\langle \begin{pmatrix} l_1 \\ l_2 \\ \vdots \\ l_N \end{pmatrix}, \begin{pmatrix} \sigma_{l_1}^2 & \sigma_{l_1 l_2} & \cdots & \sigma_{l_1 l_N} \\ \sigma_{l_1 l_2} & \sigma_{l_2}^2 & \cdots & \sigma_{l_2 l_N} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{l_1 l_N} & \sigma_{l_2 l_N} & \cdots & \sigma_{l_N}^2 \end{pmatrix} \begin{pmatrix} \sigma_{l_1 x} & \sigma_{l_1 y} & \sigma_{l_1 \theta} \\ \sigma_{l_2 x} & \sigma_{l_2 y} & \sigma_{l_2 \theta} \\ \vdots & \vdots & \vdots \\ \sigma_{l_N x} & \sigma_{l_N y} & \sigma_{l_N \theta} \end{pmatrix}, \begin{pmatrix} x \\ y \\ \theta \end{pmatrix}, \begin{pmatrix} \sigma_{l_1 x} & \sigma_{l_2 x} & \cdots & \sigma_{l_N x} \\ \sigma_{l_1 y} & \sigma_{l_2 y} & \cdots & \sigma_{l_N y} \\ \sigma_{l_1 \theta} & \sigma_{l_2 \theta} & \cdots & \sigma_{l_N \theta} \end{pmatrix}, \begin{pmatrix} \sigma_x^2 & \sigma_{xy} & \sigma_{x\theta} \\ \sigma_{xy} & \sigma_y^2 & \sigma_{y\theta} \\ \sigma_{x\theta} & \sigma_{y\theta} & \sigma_\theta^2 \end{pmatrix} \right\rangle$$

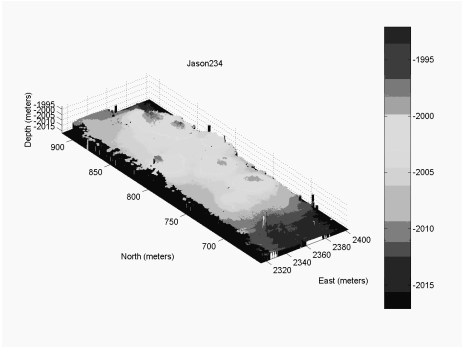
- Can handle hundreds of dimensions

E

1. Algorithm **Kalman_filter_mapping**(<μ,Σ >, *d*):
2. If *d* is a perceptual data item *z*=<*r*,*φ*>then
3. $q = (\mu_x - \mu_{x_i})^2 + (\mu_y - \mu_{y_i})^2$
4. $C = \begin{pmatrix} \frac{\mu_x - \mu_{x_i}}{\sqrt{q}} & \frac{\mu_y - \mu_{y_i}}{\sqrt{q}} & 0 & 0...0 & \frac{\mu_{x_i} - \mu_x}{\sqrt{q}} & \frac{\mu_{y_i} - \mu_y}{\sqrt{q}} & 0...0 \\ \frac{\mu_{y_i} - \mu_y}{q} & \frac{\mu_x - \mu_{x_i}}{q} & -1 & 0...0 & \frac{\mu_y - \mu_{y_i}}{q} & \frac{\mu_{x_i} - \mu_x}{q} & 0...0 \end{pmatrix}$
5. $K = \Sigma C^T \left(C \Sigma C^T + V \Sigma_{obs} V^T \right)^{-1}$
6. $\mu = \mu + K(z - C\mu)$
7. $\Sigma = (I - KC)\Sigma$
8. Else if *d* is an action data item *u*=<*v*,*ω*> then
9. $B = \begin{pmatrix} \Delta t \cos \mu_\theta & \Delta t \sin \mu_\theta & 0 & 0...0 \\ 0 & 0 & \Delta t & 0...0 \end{pmatrix}^T$
10. $\mu = \mu + Bu$
11. $\Sigma = \Sigma + W \Sigma_{act} W^T$
12. Return <μ,Σ>

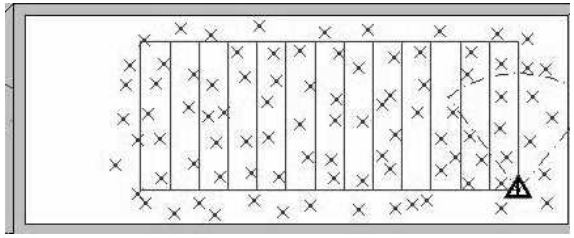
Underwater Mapping





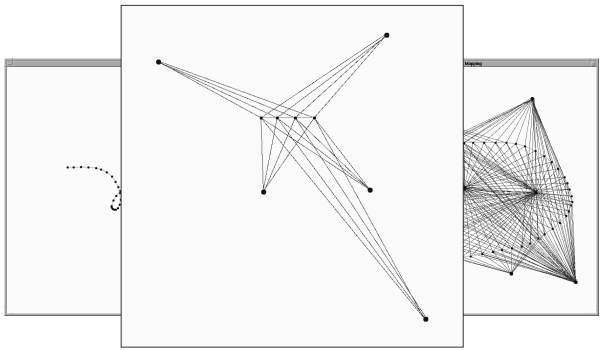
Courtesy of Louis L. Whitcomb, Johns Hopkins University

Mapping with Extended Kalman Filters



[Leonard et al 1998]

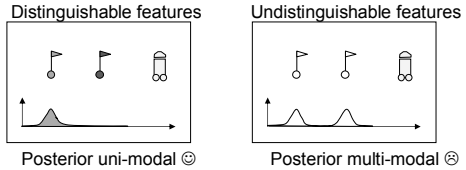
Bearings-only Mapping



[M. Deans 2002]

The Key Assumption

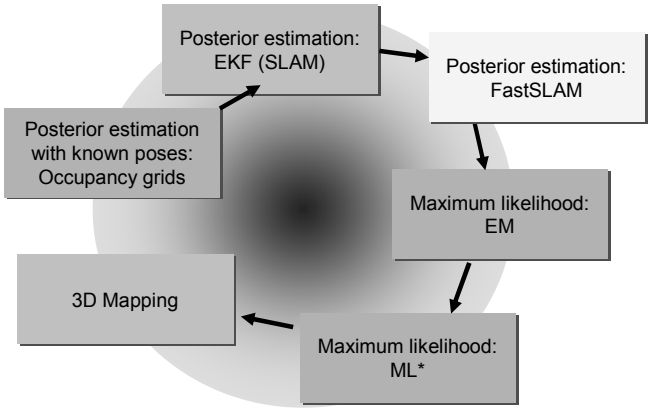
- Sensor model must be Gaussian.
- Main problem: Data association



In practice:

- Extract small set of highly distinguishable features from sensor data.
- Discard all other data.
- If ambiguous, take best guess for landmark identity.

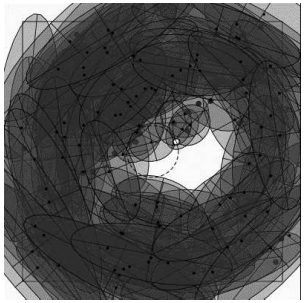
Mapping: Outline

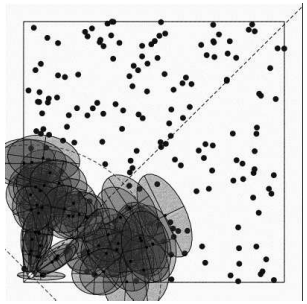


FastSLAM

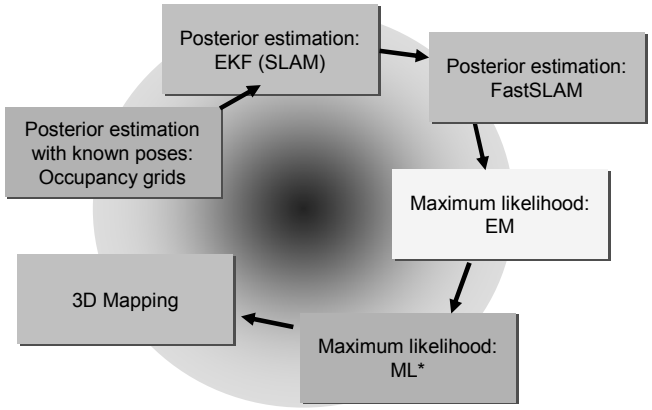
- **Idea**: Given the true trajectory of the robot, all landmark detections are independent.
- We can use Rao-Blackwellised particle filters to estimate robot locations and landmark locations.
- Which part should we sample?
- Update can be done efficiently (O(m log n)).

FastSLAM Example





Mapping: Outline



Mapping with Expectation Maximization

- **Idea**: Maximum likelihood with unknown data association.

$$Bel(m,x_t) = \eta\ p(z_t \mid m,x_t) \int p(x_t \mid x_{t-1},u_{t-1}) Bel(m,x_{t-1})\ dx_{t-1}$$

$$Bel(m) = \eta \iint \cdots \int \prod_{\tau=0}^{t} p(z_\tau \mid m,x_\tau) \prod_{\tau=1}^{t} p(x_\tau \mid x_{\tau-1},u_{\tau-1})\ dx_1 dx_2 \ldots dx_t$$

- **EM**: Maximize log-likelihood by iterating

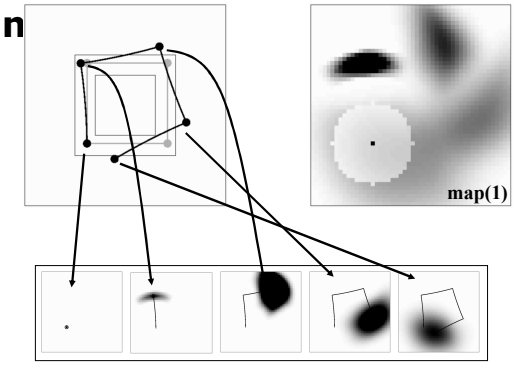
E-step: $Q[m \mid m^{[k]}] = E_{m^{[k]}}[\log p(x_0,..,x_t,d_{0..t} \mid m^{[k]}) \mid d_{0..t})]$

→ Localization (bi-directional)

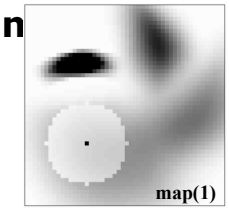
M-step: $m^{[k+1]} = \underset{m}{\operatorname{argmax}}\ Q[m \mid m^{[k]}]$

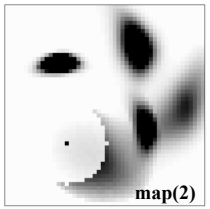
→ Mapping with known poses

Exan



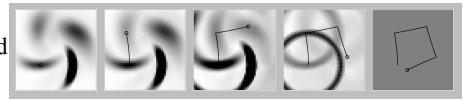
Exan



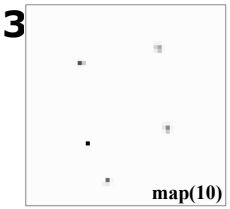


forward

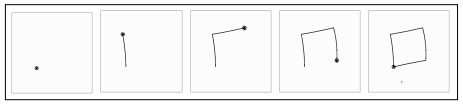
backward



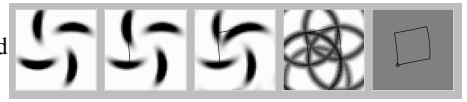
Example (3



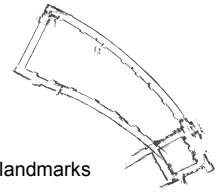
forward



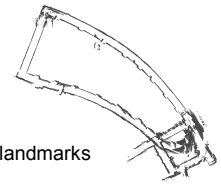
backward



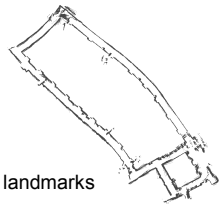
CMU's Wean Hall (80 x 25 meters)



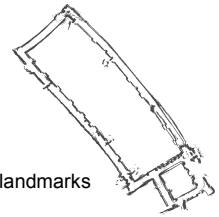
15 landmarks



16 landmarks

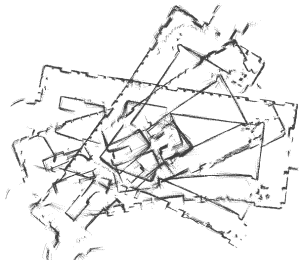


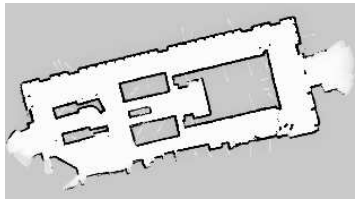
17 landmarks



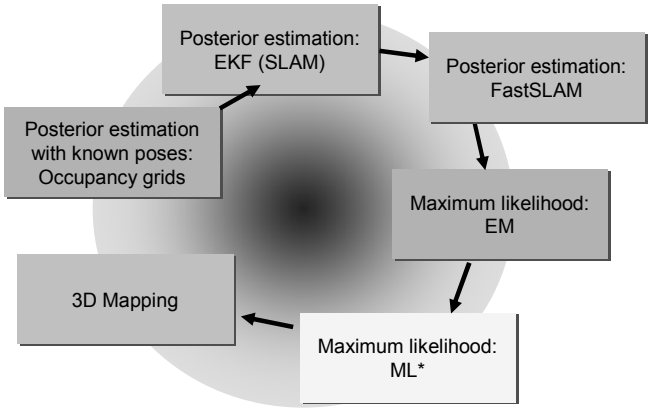
27 landmarks

EM Mapping, Example (width 45 m)





Mapping: Outline



ML* Mapping, Online

- **Idea**: step-wise maximum likelihood

$$Bel(m_t, x_t) = \eta \; p(z_t \mid m_t, x_t) \iint p(x_t, m_t \mid x_{t-1}, u_{t-1}, m_{t-1}) Bel(m_{t-1}, x_{t-1}) \; dx_{t-1} dm_{t-1}$$

1. Incremental ML estimate:

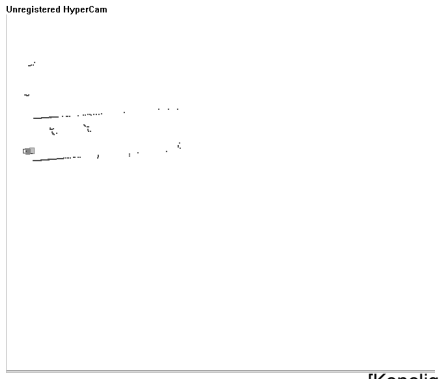
$$< m_t, x_t > = \arg\max_{<m,x>} p(z_t \mid m, x) \, p(x, m \mid x_{t-1}, u_{t-1}, m_{t-1})$$

2. Posterior:

$$Bel(x_t) = p(z_t \mid m_{t-1}, x_{t-1}) \int p(x_t \mid x_{t-1}, u_{t-1}, m_{t-1}) Bel(x_{t-1}) \; dx_{t-1}$$

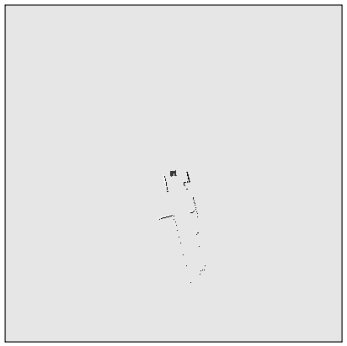
ML* Mapping, Online

Courtesy of Steffen Gutmann, Kurt Konolige



[Konolige, Gutmann 00]

ML* Mapping, Online

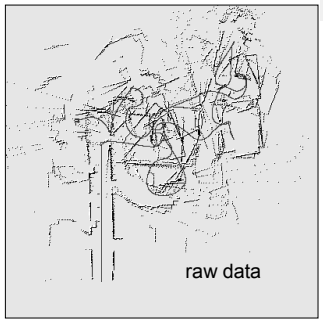


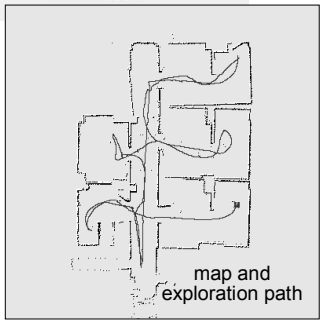
Yellow flashes:

artificially distorted map (30 deg, 50 cm)

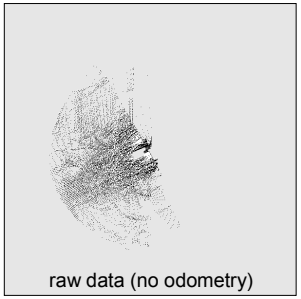
[Thrun et al. 00]

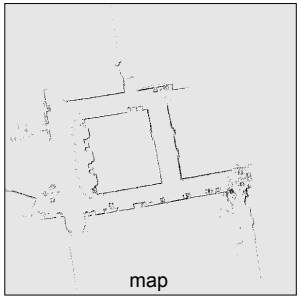
Mapping with Poor Odometry





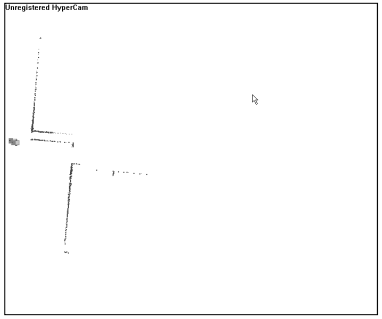
Mapping Without(!) Odometry





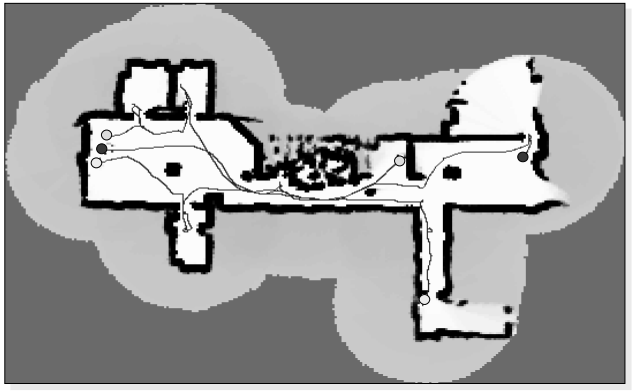
Localization in Multi-Robot Mapping

Courtesy of S. Gutmann and K. Konolige

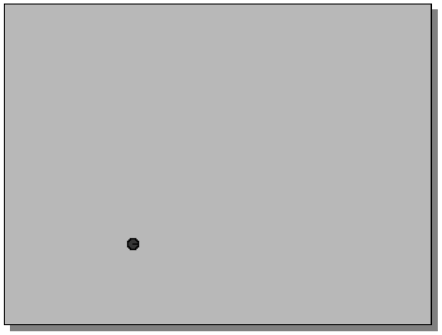


[Gutmann & Konolige, 00]

DARPA Demo 9/99, San Antonio, Texas

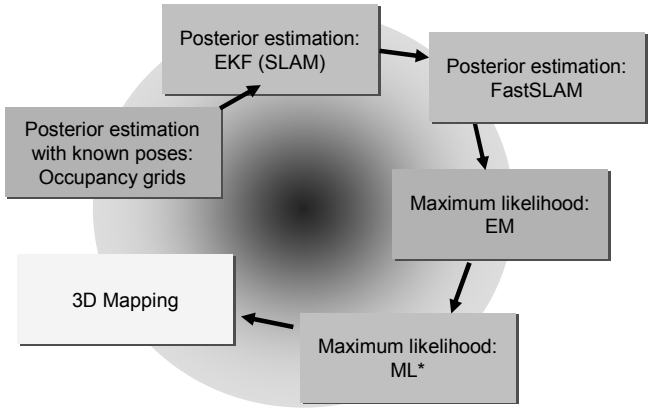


Multi-Robot Mapping

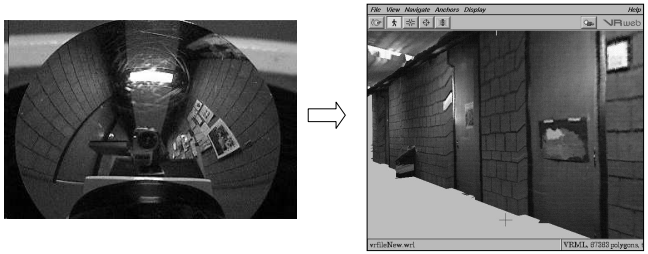


DARPA TMR Demo,
Maryland 9/00

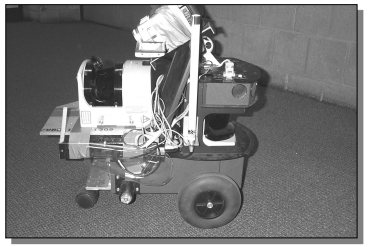
Mapping: Outline



The Problem

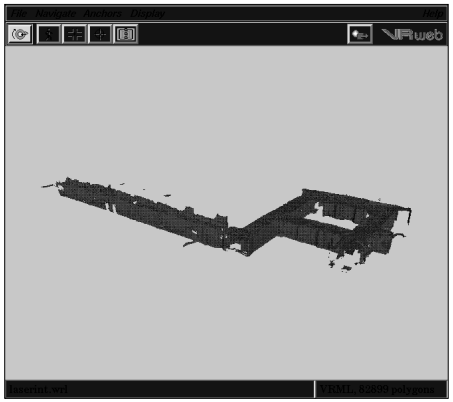


The Robot



- Forward-pointed laser (2D information)
- Upward-pointed laser (3D information)
- Panoramic camera

3D Structure Mapping (Raw Data)



3D Texture Mapping (Raw Data)

Quality of 3D model

Without EM

With EM

Mapping Summary

- Concurrent mapping and localization: hard estimation problem.
- Best known algorithms are probabilistic
  1. Occupancy grids: Binary Bayes filter, assumes known poses (= much easier).
  2. EKF/SLAM: Full posterior estimation, but restrictive assumptions (data association).
  3. FastSLAM uses sampling to estimate trajectories, thereby rendering observations independent. O(m log n) instead of $O(n^2)$.
  4. EM: Maximum likelihood, solves data association.
  5. ML*: less robust but online.
  6. 3D Maps: Away from occupancy, towards multi-object models.

Mapping Algorithms - Comparison

	SLAM (Kalman)	EM	ML*	FastSLAM
Output	Posterior	ML/MAP	ML/MAP	Posterior
Convergence	Strong	Weak?	No	Stong
Local minima	No	Yes	Yes	No
Real time	Yes	No	Yes	Yes
Odom. Error	Unbounded	Unbounded	Unbounded	Unbounded
Sensor Noise	Gaussian	Any	Any	Any
# Features	10^3	∞	∞	$>10^3$
Feature uniq	Yes	No	~No	Yes
Raw data	No	Yes	Yes	No