Simultaneous Localization and Mapping

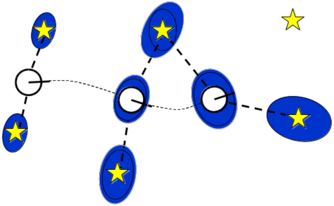
Miroslav Kulich

Intelligent and Mobile Robotics Group
Czech Institute of Informatics, Robotics and Cybernetics
Czech Technical University in Prague

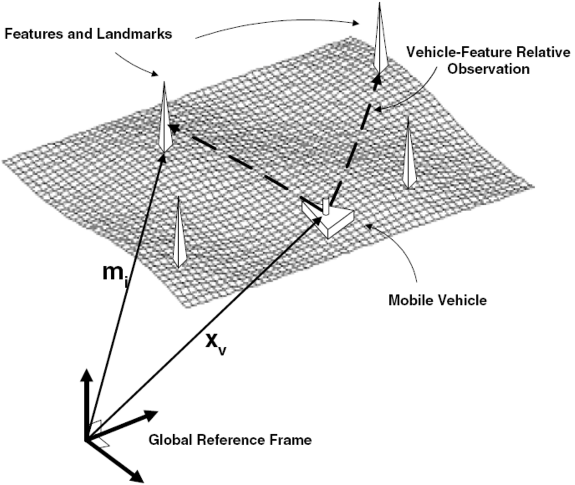
Winter semester 2017/2018

SLAM - Simultaneous Localization and Mapping

- The task of building a map while estimating the pose of the robot relative to this map.
- More difficult than separate localization or mapping.
- Chicken and egg problem: a map is needed to localize the robot and a pose estimate is needed to build a map.
- Given: robot control, sensing.
- Estimate: map (feature-based, grid), path of the robot
- State: $\langle x, y, \theta, map \rangle$
- Map:
  - Feature-based: $\langle l_1, l_2, \ldots, l_n \rangle$
  - Grid: $\langle c_{11}, c_{12}, \ldots c_{1n}, \ldots, c_{mn} \rangle$

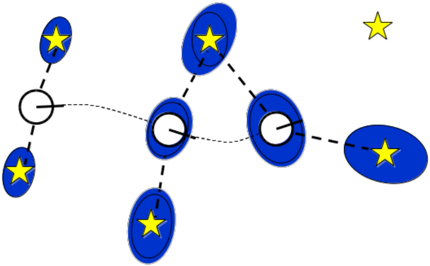


Structure of the Landmark-based SLAM-Problem



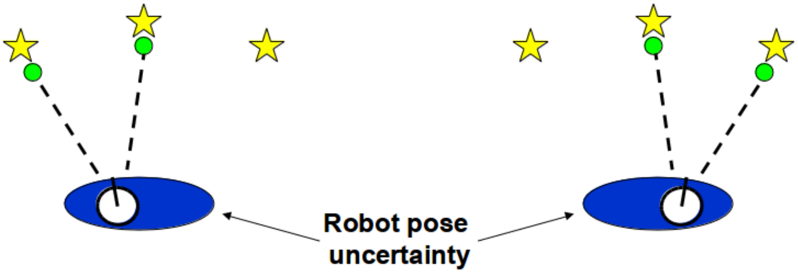
Why is SLAM a hard problem?

- **SLAM:** robot path and map are both unknown!



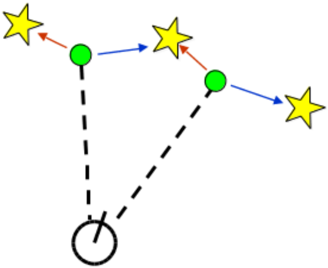
- Robot path error correlates errors in the map.

Why is SLAM a hard problem?



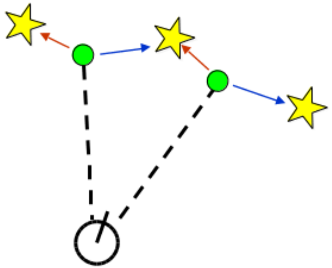
- In the real world, the mapping between observations and landmarks is unknown.
- Picking wrong data associations can have catastrophic consequences.
- Pose error correlates data associations.

Data Association Problem



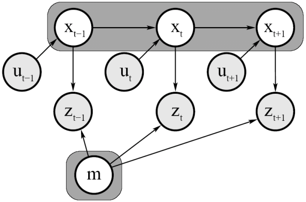
- A data association is an assignment of observations to landmarks.
- In general for n measurements and m landmarks there are ??? possible associations.
- Also called "assignment problem"

Data Association Problem



- A data association is an assignment of observations to landmarks.
- In general for n measurements and m landmarks there are $\binom{m}{n}$ possible associations.
- Also called "assignment problem"

SLAM



Full SLAM

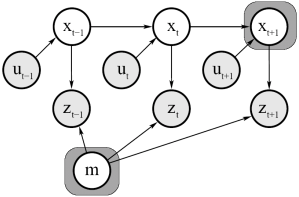
- Estimates entire path and map
- $p(x_{1:t}, m | z_{1:t}, u_{1:t})$

Online SLAM

- Estimates most recent pose and map

$$p(x_t, m | z_{1:t}, u_{1:t}) = \int \int \cdots \int p(x_{1:t}, m | z_{1:t}, u_{1:t}) dx_1 dx_2 \ldots dx_{t-1}$$

- Integrations typically done one at a time



Techniques for Generating Consistent Maps

- Scan matching
- EKF-SLAM
- Fast-SLAM
- Probabilistic mapping with a single map and a posterior about poses.
- Graph-SLAM
- Sparse Extended Information Filters (SEIFs)
- RAT-SLAM, ...

Scan matching

- Maximization of the likehood of the i-th pose and a map relative to the $(i-1)$-th position and a map.

$$\hat{x}_t = \arg\max_{x_t} \left\{ p(z_t | x_t, \hat{m}^{[t-1]}) p(x_t | u_{t-1}, \hat{x}_{t-1}) \right\}$$

- Calculate the map $\hat{m}^{[t]}$ according to "mapping with known poses" and observations.

EKF-SLAM

- Feature map (number of landmarks < 1000)
- State:
  $s_t = (x_t, m)^T = (x, y, \theta, l_{1,x}, l_{1,y}, l_{2,x}, l_{2,y}, \ldots, l_{n,x}, l_{n,y},)^T$
- Assuming knowledge of associations
- Only positive information is processed
- PDF represented with a high-dimensional Gaussian (3+2n)

$$Bel(x_t, m_t) = \begin{pmatrix} x \\ y \\ \theta \\ l_1 \\ l_2 \\ \vdots \\ l_n \end{pmatrix}, \begin{pmatrix} \sigma_x^2 & \sigma_{xy} & \sigma_{x\theta} & \sigma_{xl_1} & \sigma_{xl_2} & \cdots & \sigma_{xl_n} \\ \sigma_{xy} & \sigma_y^2 & \sigma_{y\theta} & \sigma_{yl_1} & \sigma_{yl_2} & \cdots & \sigma_{yl_n} \\ \sigma_{x\theta} & \sigma_{y\theta} & \sigma_\theta^2 & \sigma_{\theta l_1} & \sigma_{\theta l_2} & \cdots & \sigma_{\theta l_n} \\ \sigma_{xl_1} & \sigma_{yl_1} & \sigma_{\theta l_1} & \sigma_{l_1}^2 & \sigma_{l_1 l_2} & \cdots & \sigma_{l_1 l_n} \\ \sigma_{xl_2} & \sigma_{yl_2} & \sigma_{\theta l_2} & \sigma_{l_1 l_2} & \sigma_{l_2}^2 & \cdots & \sigma_{l_2 l_n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ldots & \vdots \\ \sigma_{xl_n} & \sigma_{yl_n} & \sigma_{\theta l_n} & \sigma_{l_n l_1} & \sigma_{l_n l_2} & \cdots & \sigma_{l_n}^2 \end{pmatrix}$$

- Is matrix K (gain) sparse?

EKF-SLAM

- Feature map (number of landmarks < 1000)
- State:
  $s_t = (x_t, m)^T = (x, y, \theta, l_{1,x}, l_{1,y}, l_{2,x}, l_{2,y}, \ldots, l_{n,x}, l_{n,y},)^T$
- Assuming knowledge of associations
- Only positive information is processed
- PDF represented with a high-dimensional Gaussian (3+2n)

$$Bel(x_t, m_t) = \begin{pmatrix} x \\ y \\ \theta \\ l_1 \\ l_2 \\ \vdots \\ l_n \end{pmatrix}, \begin{pmatrix} \sigma_x^2 & \sigma_{xy} & \sigma_{x\theta} & \sigma_{xl_1} & \sigma_{xl_2} & \cdots & \sigma_{xl_n} \\ \sigma_{xy} & \sigma_y^2 & \sigma_{y\theta} & \sigma_{yl_1} & \sigma_{yl_2} & \cdots & \sigma_{yl_n} \\ \sigma_{x\theta} & \sigma_{y\theta} & \sigma_\theta^2 & \sigma_{\theta l_1} & \sigma_{\theta l_2} & \cdots & \sigma_{\theta l_n} \\ \sigma_{xl_1} & \sigma_{yl_1} & \sigma_{\theta l_1} & \sigma_{l_1}^2 & \sigma_{l_1 l_2} & \cdots & \sigma_{l_1 l_n} \\ \sigma_{xl_2} & \sigma_{yl_2} & \sigma_{\theta l_2} & \sigma_{l_1 l_2} & \sigma_{l_2}^2 & \cdots & \sigma_{l_2 l_n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ldots & \vdots \\ \sigma_{xl_n} & \sigma_{yl_n} & \sigma_{\theta l_n} & \sigma_{l_n l_1} & \sigma_{l_n l_2} & \cdots & \sigma_{l_n}^2 \end{pmatrix}$$

- Matrix K (gain) is not sparse: measuring a landmark improves precision of both landmark position and robot position.

EKF-SLAM

Prediction

EKF_SLAM_Prediction($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t, c_t, R_t$):

2: $$F_x = \begin{pmatrix} 1 & 0 & 0 & 0 \cdots 0 \\ 0 & 1 & 0 & 0 \cdots 0 \\ 0 & 0 & 1 & 0 \cdots 0 \end{pmatrix}$$

3: $$\bar{\mu}_t = \mu_{t-1} + F_x^T \begin{pmatrix} -\frac{v_t}{\omega_t} \sin \mu_{t-1,\theta} + \frac{v_t}{\omega_t} \sin(\mu_{t-1,\theta} + \omega_t \Delta t) \\ \frac{v_t}{\omega_t} \cos \mu_{t-1,\theta} - \frac{v_t}{\omega_t} \cos(\mu_{t-1,\theta} + \omega_t \Delta t) \\ \omega_t \Delta t \end{pmatrix}$$

4: $$G_t = I + F_x^T \begin{pmatrix} 0 & 0 & -\frac{v_t}{\omega_t} \cos \mu_{t-1,\theta} + \frac{v_t}{\omega_t} \cos(\mu_{t-1,\theta} + \omega_t \Delta t) \\ 0 & 0 & -\frac{v_t}{\omega_t} \sin \mu_{t-1,\theta} + \frac{v_t}{\omega_t} \sin(\mu_{t-1,\theta} + \omega_t \Delta t) \\ 0 & 0 & 0 \end{pmatrix} F_x$$

5: $$\bar{\Sigma}_t = G_t \, \Sigma_{t-1} \, G_t^T + \underbrace{F_x^T \, R_t^x \, F_x}_{R_t}$$

EKF-SLAM

Correction 1

EKF_SLAM_Correction

6: $Q_t = \begin{pmatrix} {\sigma_r}^2 & 0 \\ 0 & {\sigma_\phi}^2 \end{pmatrix}$

7: *for all observed features* $z_t^i = (r_t^i, \phi_t^i)^T$ *do*

8: $j = c_t^i$

9: *if landmark* j *never seen before*

10: $\begin{pmatrix} \bar{\mu}_{j,x} \\ \bar{\mu}_{j,y} \end{pmatrix} = \begin{pmatrix} \bar{\mu}_{t,x} \\ \bar{\mu}_{t,y} \end{pmatrix} + \begin{pmatrix} r_t^i \cos(\phi_t^i + \bar{\mu}_{t,\theta}) \\ r_t^i \sin(\phi_t^i + \bar{\mu}_{t,\theta}) \end{pmatrix}$

11: *endif*

12: $\delta = \begin{pmatrix} \delta_x \\ \delta_y \end{pmatrix} = \begin{pmatrix} \bar{\mu}_{j,x} - \bar{\mu}_{t,x} \\ \bar{\mu}_{j,y} - \bar{\mu}_{t,y} \end{pmatrix}$

13: $q = \delta^T \delta$

14: $\hat{z}_t^i = \begin{pmatrix} \sqrt{q} \\ \operatorname{atan2}(\delta_y, \delta_x) - \bar{\mu}_{t,\theta} \end{pmatrix}$

EKF-SLAM

Correction 2

$$F_{x,j} = \begin{pmatrix} 1 & 0 & 0 & 0\cdots0 & 0 & 0 & 0\cdots0 \\ 0 & 1 & 0 & 0\cdots0 & 0 & 0 & 0\cdots0 \\ 0 & 0 & 1 & 0\cdots0 & 0 & 0 & 0\cdots0 \\ 0 & 0 & 0 & 0\cdots0 & 1 & 0 & 0\cdots0 \\ 0 & 0 & 0 & \underbrace{0\cdots0}_{2j-2} & 0 & 1 & \underbrace{0\cdots0}_{2N-2j} \end{pmatrix}$$

$$H_t^i = \frac{1}{q} \begin{pmatrix} -\sqrt{q}\delta_x & -\sqrt{q}\delta_y & 0 & +\sqrt{q}\delta_x & \sqrt{q}\delta_y \\ \delta_y & -\delta_x & -q & -\delta_y & +\delta_x \end{pmatrix} F_{x,j}$$

$$K_t^i = \bar{\Sigma}_t \; H_t^{iT} (H_t^i \; \bar{\Sigma}_t \; H_t^{iT} + Q_t)^{-1}$$

$$\bar{\mu}_t = \bar{\mu}_t + K_t^i (z_t^i - \hat{z}_t^i)$$

$$\bar{\Sigma}_t = (I - K_t^i \; H_t^i) \; \bar{\Sigma}_t$$

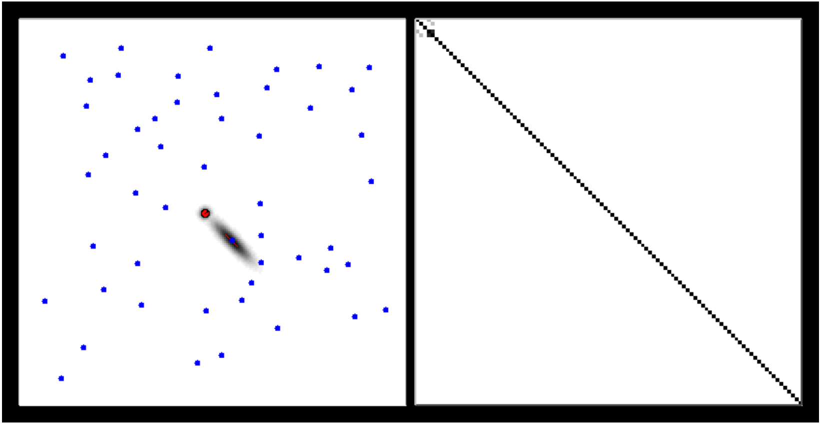
endfor

$$\mu_t = \bar{\mu}_t$$

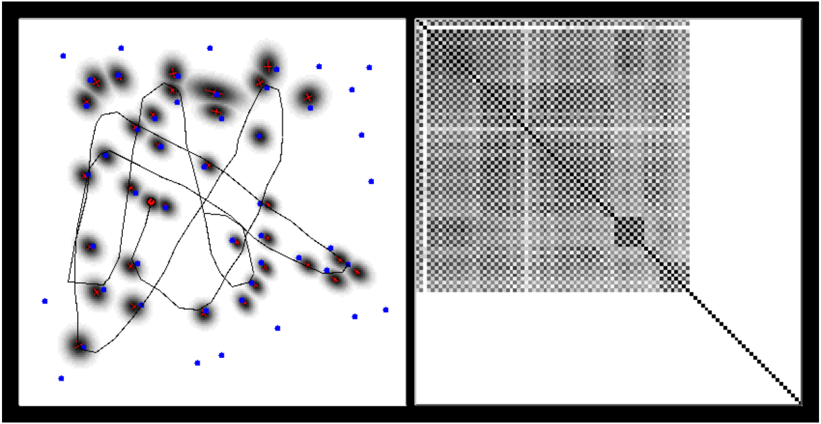
$$\Sigma_t = \bar{\Sigma}_t$$

return μ_t, Σ_t

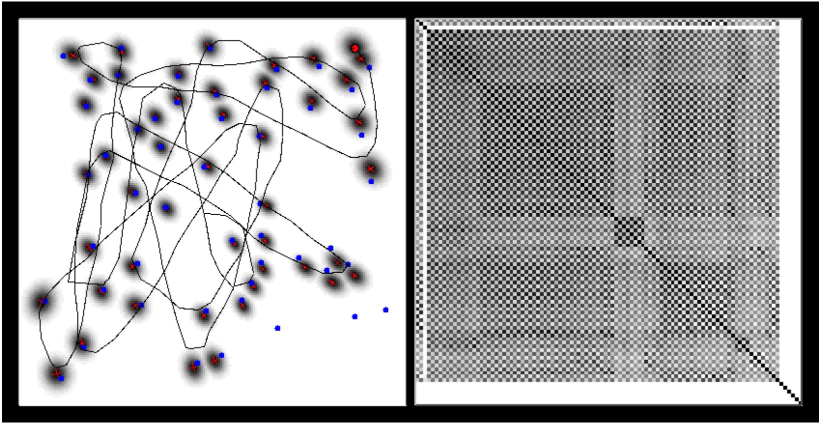
EKF-SLAM



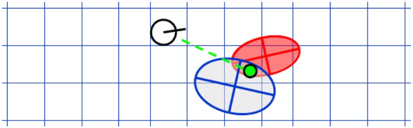
EKF-SLAM



EKF-SLAM

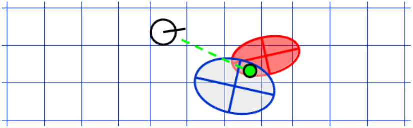


When correspondences are unknown: Data Association



Was the observation generated by the red or the blue landmark?

When correspondences are unknown: Data Association



Was the observation generated by the red or the blue landmark?

$$p(observation|{\color{red}red} = 0.3) \quad p(observation|{\color{blue}blue} = 0.7)$$

- Two options for data association:
  - Pick the most probable match.
  - Pick an random association weighted by the observation likelihoods.
- If the probability is too low, generate a new landmark.
- Each measurement is processed separately: one landmark can correspond to several measurements.

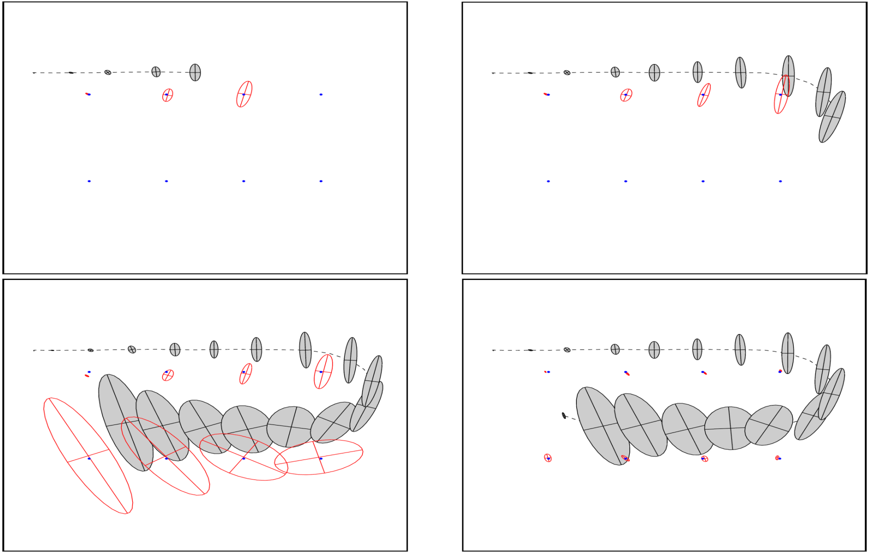
EKF-SLAM

- EKF-SLAM is online: previous positions are „hidden“ in the covariance matrix.
- Initialization of landmarks:
  - To zero: $(0, 0, 0)$
  - To the first measurements of the landmark:

$$\begin{pmatrix} \mu_{j,x} \\ \mu_{j,y} \end{pmatrix} = \begin{pmatrix} \mu_{t,x} \\ \mu_{t,y} \end{pmatrix} + \begin{pmatrix} r_t^i \cos(\phi_t^i + \mu_{j,\phi}) \\ r_t^i \sin(\phi_t^i + \mu_{j,\phi}) \end{pmatrix}$$

  - Bearing only sensors (cameras): integration of several measurements.
- Filtration of outliers: provisional landmark list.
- Landmark existence probability – log odds ratio.
- Numerical instability – initialization of new landmark estimate.

EKF-SLAM: example



Localization vs. SLAM

- A particle filter can be used to solve both problems.
- Localization: state space $\langle x, y, \theta \rangle$.
- SLAM: state space $\langle x, y, \theta, \text{map} \rangle$.
  - for landmark maps: $\langle l_1, l_2, \ldots, l_n \rangle$
  - for grid maps: $\langle c_{1,1}, c_{1,2}, \ldots, c_{1,n}, c_{2,1}, \ldots, c_{n,m} \rangle$

Localization vs. SLAM

- A particle filter can be used to solve both problems.
- Localization: state space $\langle x, y, \theta \rangle$.
- SLAM: state space $\langle x, y, \theta, \text{map} \rangle$.
  - for landmark maps: $\langle l_1, l_2, \ldots, l_n \rangle$
  - for grid maps: $\langle c_{1,1}, c_{1,2}, \ldots, c_{1,n}, c_{2,1}, \ldots, c_{n,m} \rangle$
- **Problem:** The number of particles needed to represent a posterior grows exponentially with the dimension of the state space!

Dependencies

How to reduce dimension of the state space

- Is there a dependency between the dimensions of the state space?
- If so, can we use the dependency to solve the problem more efficiently?

Dependencies

How to reduce dimension of the state space

- Is there a dependency between the dimensions of the state space?
- If so, can we use the dependency to solve the problem more efficiently?
- In the SLAM context
  - The map depends on the poses of the robot.
  - We know how to build a map given the position of the sensor is known.

Factored Posterior Probability (landmarks)

poses landmarks observations movements

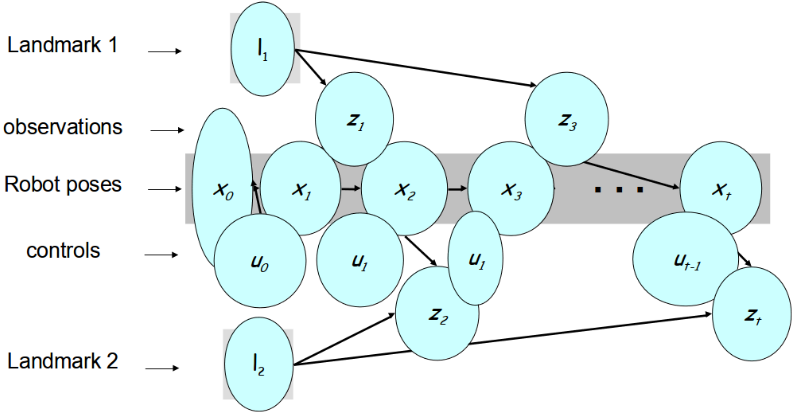
$$p(x_{1:t}, l_{1:m} | z_{1:t}, u_{0,t-1}) = p(x_{1:t} | z_{1:t}, u_{0,t-1}) p(l_{1:m} | x_{1:t} z_{1:t})$$

SLAM posterior — Robot path posterior — Landmark positions (using known poses)

Does this help to solve the problem?

Factorization first introduced by Murphy in 1999.

Mapping using landmarks



Knowledge of the robot's true path renders landmark positions conditionally independent!

Factored Posterior

$$
\begin{aligned}
p(x_{1:t}, l_{1:m} | z_{1:t}, u_{0,t-1}) \\
&= p(x_{1:t} | z_{1:t}, u_{0,t-1}) p(l_{1:m} | x_{1:t}, z_{1:t}) \\
&= p(x_{1:t} | z_{1:t}, u_{0,t-1}) \prod_{i=1}^{m} p(l_i | x_{1:t}, z_{1:t})
\end{aligned}
$$

Robot path posterior (localization problem) x

Conditionally independent landmark positions

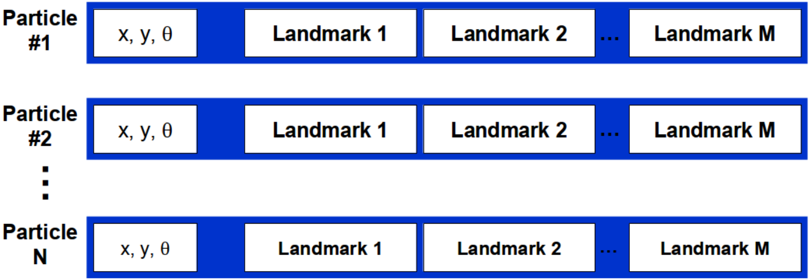
Rao-Blackwellization

$$p(x_{1:t}, l_{1:m} | z_{1:t}, u_{0,t-1}) = p(x_{1:t} | z_{1:t}, u_{0,t-1}) \prod_{i=1}^{m} p(l_i | x_{1:t}, z_{1:t})$$

- This factorization is also called Rao-Blackwellization
- Given that the second term can be computed efficiently, particle filtering becomes possible!

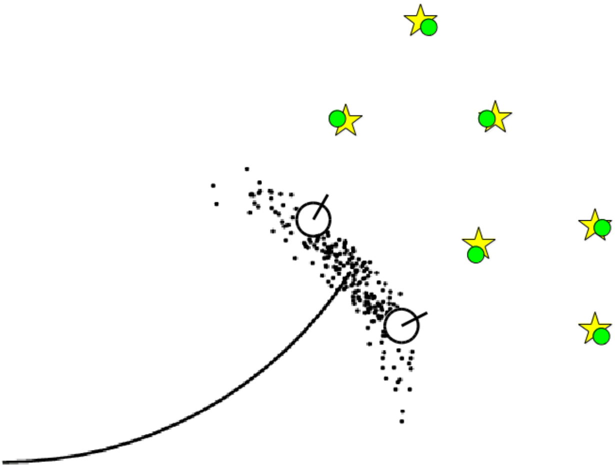
FastSLAM

- Rao-Blackwellized particle filtering based on landmarks (Montemerlo et al., 2002)
- Each landmark is represented by a 2x2 Extended Kalman Filter (EKF).
- Each particle therefore has to maintain M EKFs.



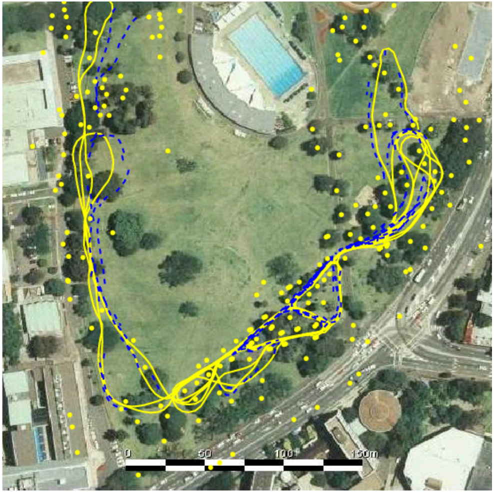
Multi-Hypothesis Data Association

- Data association is done on a per-particle basis.
- Robot pose error is factored out of data association decisions



Results - Victoria Park

- Trajectory length: 4 km
- Position error < 5 m RMS
- 100 particles

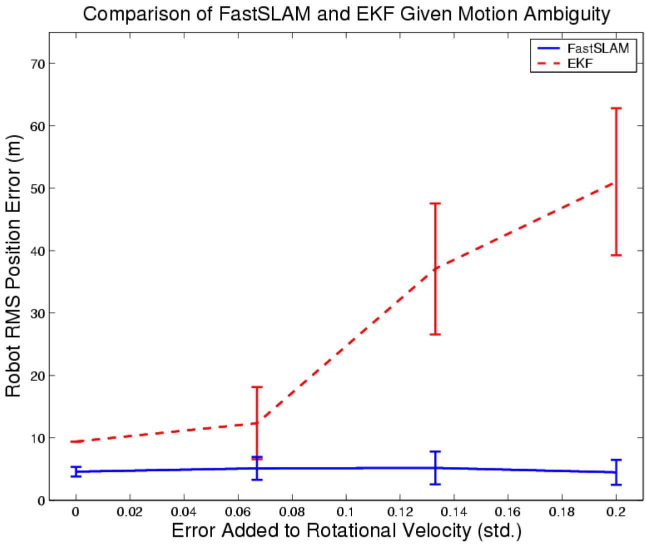


Dataset courtesy of University of Sydney

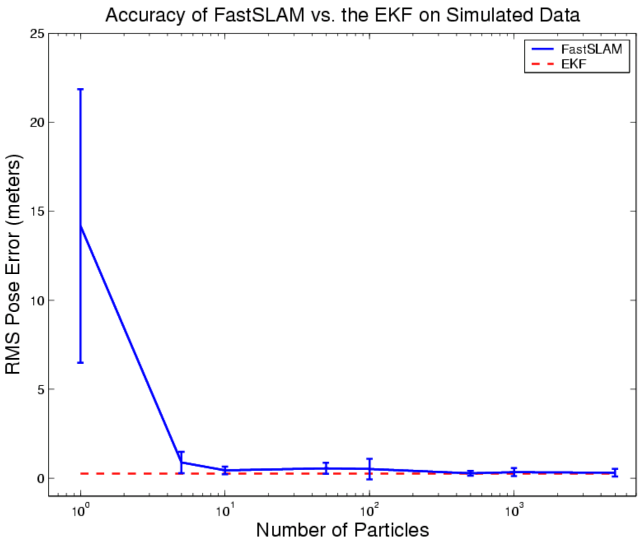
Blue=GPS

Yellow=FastSLAM

Results - Data Association

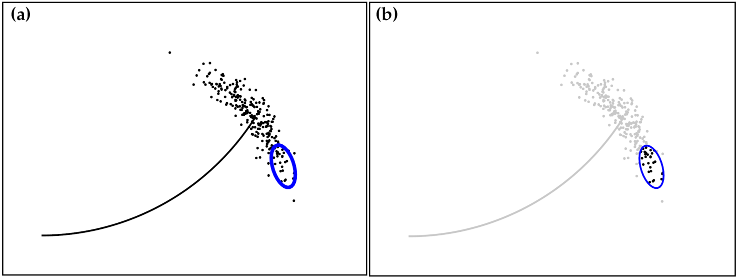


Results - Accuracy



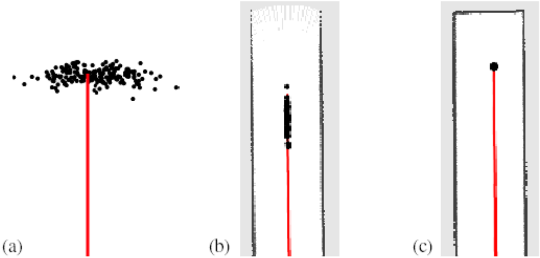
FastSLAM - conclusion

- FastSLAM is both full SLAM and online SLAM.
- Complexity of $O(nm)$ can be improved to $O(n \log m)$.
- Unknown correspondences: similarly to EKF-SLAM (for each particle separately!).
- Motion model does not correspond with the desired distribution $\Rightarrow$ sampling with respoect to sensor model.
- Can process also negative informantion.



FastSLAM - improved proposal

The proposal adapts to the structure of the environment.



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

observations movements

poses map

$$p(x_{1:t}, m | z_{1:t}, u_{0,t-1}) = p(x_{1:t} | z_{1:t}, u_{0,t-1}) p(m | x_{1:t}, z_{1:t})$$

SLAM posterior Robot path posterior Mapping with known poses

Rao-Blackwellization

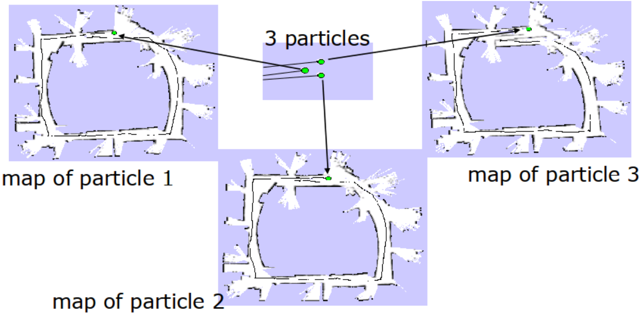
$$p(x_{1:t}, m | z_{1:t}, u_{0,t-1}) = p(x_{1:t} | z_{1:t}, u_{0,t-1}) p(m | x_{1:t}, z_{1:t})$$

- Use particle filter for localization.
- Use the pose estimate from the MCL part and apply mapping with known posses.

Grid-based FastSLAM

Algorithm

- for $k = 1$ to m
  - Select a sample from the previous generation.
  - Update the sample using motion model.
  - Compute the weight according to sensor model.
  - Update the map for the corresponding particle.



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 is smaller.

[Haehnel et al., 2003]

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 seriously be 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.

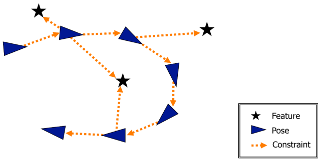
GraphSLAM

- Use a graph to represent the problem
- Every node in the graph corresponds to a pose of the robot/landmark during mapping
- Every edge between two nodes corresponds to a spatial constraint between them
- Graph-Based SLAM: Build the graph and find a node configuration that minimizes the error introduced by the constraints

GraphSLAM

Find a configuration of the nodes so that the real and predicted observations are as similar as possible

- Nodes can represent:
  - Robot poses x_t
  - Landmark locations m_j
- Edges can represent:
  - Landmark observations z_t^i
  - Odometry measurements u_t
- The minimization optimizes the landmark locations and robot poses



GraphSLAM

Building up the graph

- Information about measurements/controls is mapped into constraints between nodes.
- Edge ≈ spring in a spring-mass model
- Measurement:

$$(z_t^i - h(x_t, m_j))^T Q_t^{-1} (z_t^i - h(x_t, m_j))$$

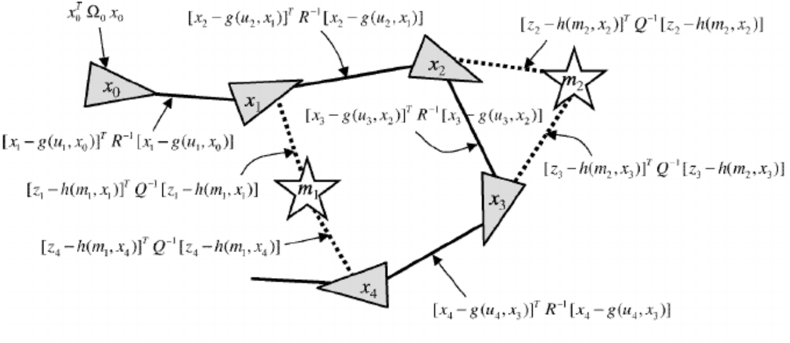
- Control (robot movement)

$$(x_t - g(x_{t-1}, u_t))^T R_t^{-1} (x_t - g(x_{t-1}, u_t))$$

$$\begin{aligned} J_G &= x_0^T \Omega_0 x_0 + \sum_t (x_t - g(x_{t-1}, u_t))^T R_t^{-1} (x_t - g(x_{t-1}, u_t)) \\ &+ \sum_t \sum_i (z_t^i - h(x_t, m_j))^T Q_t^{-1} (z_t^i - h(x_t, m_j)) \end{aligned}$$

GraphSLAM

Example



$$
\begin{aligned}
J_G &= x_0^T \Omega_0 x_0 + \sum_t (x_t - g(x_{t-1}, u_t))^T R_t^{-1} (x_t - g(x_{t-1}, u_t)) \\
&+ \sum_t \sum_i (z_t^i - h(x_t, m_j))^T Q_t^{-1} (z_t^i - h(x_t, m_j))
\end{aligned}
$$

Canonical parametrization of a Gaussian

$$p(x) = \det(2\pi\Sigma)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right\}$$

Canonical parametrization of a Gaussian

$$p(x) = \det(2\pi\Sigma)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right\}$$

$$= \det(2\pi\Sigma)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}x^T\Sigma^{-1}x + x^T\Sigma^{-1}\mu - \frac{1}{2}\mu^T\Sigma^{-1}\mu\right\}$$

Canonical parametrization of a Gaussian

$$
\begin{aligned}
p(x) &= \det(2\pi\Sigma)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right\} \\
&= \det(2\pi\Sigma)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}x^T\Sigma^{-1}x + x^T\Sigma^{-1}\mu - \frac{1}{2}\mu^T\Sigma^{-1}\mu\right\} \\
&= \underbrace{\det(2\pi\Sigma)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}\mu^T\Sigma^{-1}\mu\right\}}_{const.} \exp\left\{-\frac{1}{2}x^T\Sigma^{-1}x + x^T\Sigma^{-1}\mu\right\}
\end{aligned}
$$

Canonical parametrization of a Gaussian

$$
\begin{aligned}
p(x) &= \det(2\pi\Sigma)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right\} \\
&= \det(2\pi\Sigma)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}x^T\Sigma^{-1}x + x^T\Sigma^{-1}\mu - \frac{1}{2}\mu^T\Sigma^{-1}\mu\right\} \\
&= \underbrace{\det(2\pi\Sigma)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}\mu^T\Sigma^{-1}\mu\right\}}_{const.} \exp\left\{-\frac{1}{2}x^T\Sigma^{-1}x + x^T\Sigma^{-1}\mu\right\} \\
&= \eta \exp\left\{-\frac{1}{2}x^T\Sigma^{-1}x + x^T\Sigma^{-1}\mu\right\}
\end{aligned}
$$

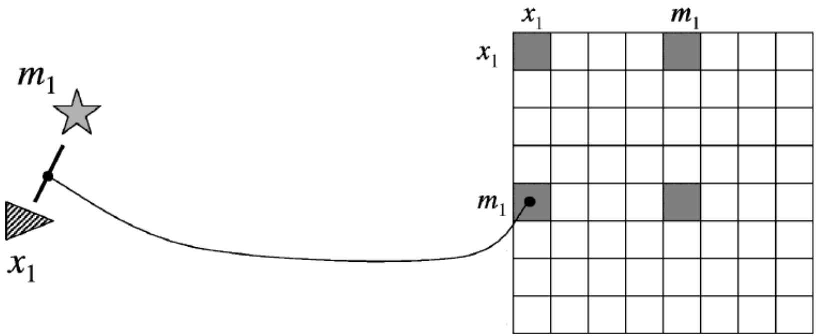
Canonical parametrization of a Gaussian

$$
\begin{aligned}
p(x) &= \det(2\pi\Sigma)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right\} \\
&= \det(2\pi\Sigma)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}x^T\Sigma^{-1}x + x^T\Sigma^{-1}\mu - \frac{1}{2}\mu^T\Sigma^{-1}\mu\right\} \\
&= \underbrace{\det(2\pi\Sigma)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}\mu^T\Sigma^{-1}\mu\right\}}_{const.} \exp\left\{-\frac{1}{2}x^T\Sigma^{-1}x + x^T\Sigma^{-1}\mu\right\} \\
&= \eta \exp\left\{-\frac{1}{2}x^T\Sigma^{-1}x + x^T\Sigma^{-1}\mu\right\} \\
&= \eta \exp\left\{-\frac{1}{2}x^T\Omega x + x^T\xi\right\}
\end{aligned}
$$

- Information matrix: $\Omega = \Sigma^{-1}$
- Information vector: $\xi = \Sigma^{-1}\mu$

GraphSLAM

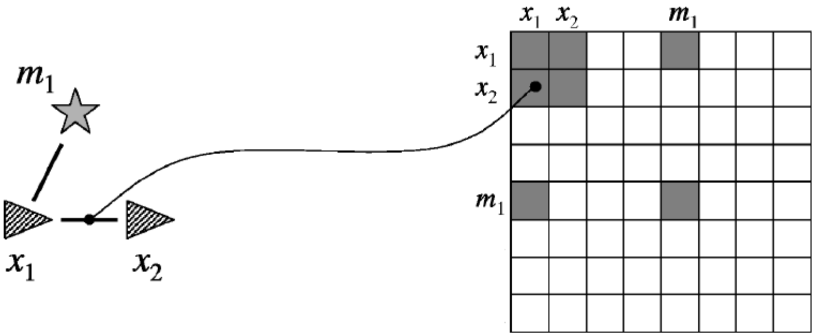
Information matrix is sparse!



Observation of landmark m_1.

GraphSLAM

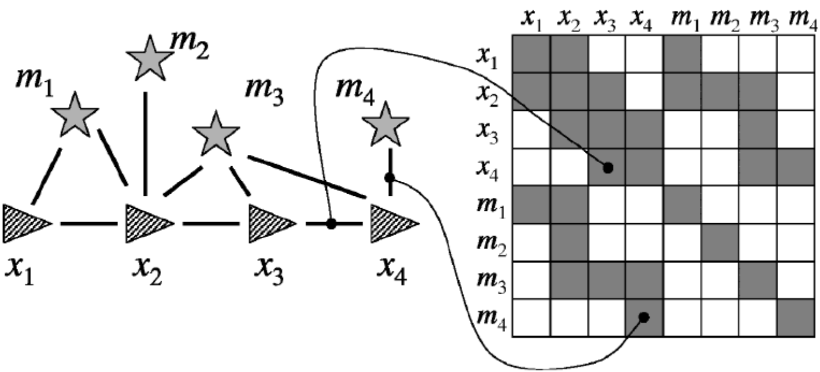
Information matrix is sparse!



Robot moves from x_1 to x_2.

GraphSLAM

Information matrix is sparse!



everal steps later . . .

GraphSLAM

Solving the problem

- Goal: minimize J_G
- General (non-linear) least squares form which can be solved by a number of algorithms:
  - Gradient descent
  - Levenberg-Marquardt
  - Conjugate gradient
- g2o (`http://openslam.org/g2o.html`): A General Framework for Graph Optimization
- These techniques compute mode only (not covariance).

Acknowledgement

I was inspired by lessons of Sebastian Thrun, from which majority of the presented figures comes. These (and many others) can be found and download from

http://www.probabilistic-robotics.org/.

I also recommend the book

S. Thrun, W. Burgard, and D. Fox: *Probabilistic Robotics*. MIT Press, Cambridge, MA, 2005.

and slides from SLAM Tutorial@ICRA 2016

www.dis.uniroma1.it/~labrococo/tutorial_icra_2016/