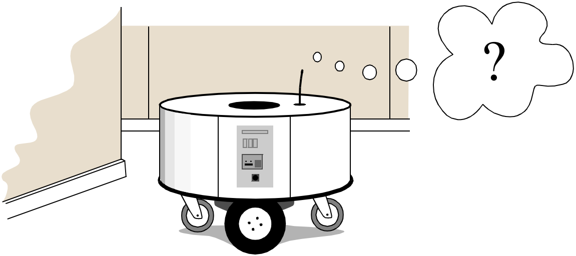
Localization and Mapping

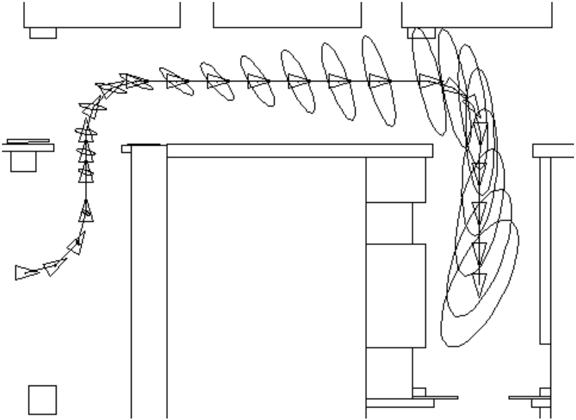
Brian Williams

Introduction

For many tasks, a mobile robot must know where it is relative to its environment.



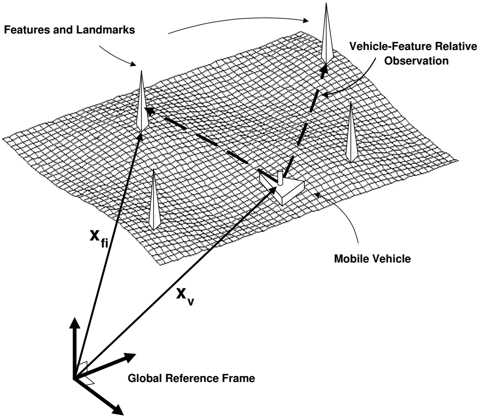
Introduction



Dead reckoning is insufficient since uncertainty grows very quickly.

Observations

Sensor measurements can help determine both the robots position and information about the environment that it is moving through.



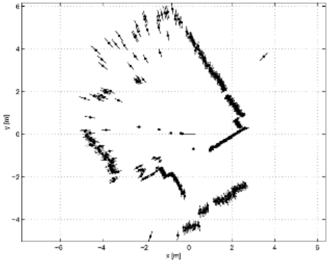
Absolute

- GPS
- Compass

Vehicle Relative

- Laser Rangefinder
- Camera
- Odometry

Observations



Sensors aren't perfect

- Noisy measurements
- Spurious measurements

Solution:

Probabilistic Estimation

$$p(\underbrace{\mathbf{x}_v}_{\text{Vehicle Pose}}, \underbrace{\mathbf{M}}_{\text{Map}} | \underbrace{\mathbf{Z}^k}_{\text{Observations}})$$

Estimate the robot pose and the map given the set of noisy observations.

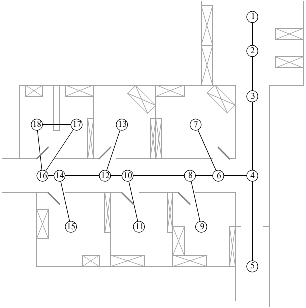
Map Types

Many different representations can be used to estimate the structure of the environment that a robot is moving through.

Map Types

- Topological
- Metric
  - Grid-Based
  - Pose-Based
  - Feature-Based

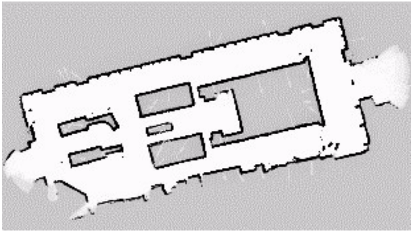
Map Types

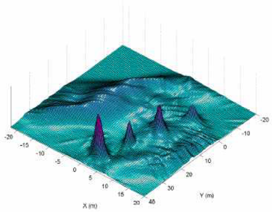


Topological Maps

- Map describes distinct places (nodes) and the connections between them (edges).
- Useful for high level path planning.
- Useless for obstacle avoidance.

Map Types

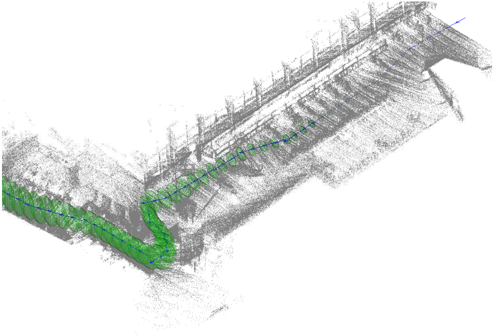




Grid-Based

- World is discretized into a grid of cells.
- Each cell maintains an estimate of its contents.
- Discretization must be sufficiently fine.
- Memory intensive.

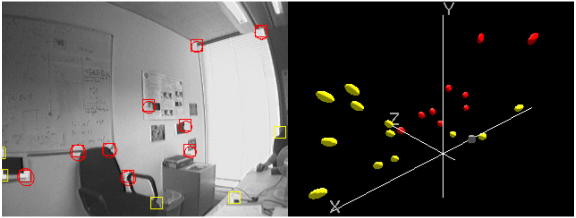
Map Types

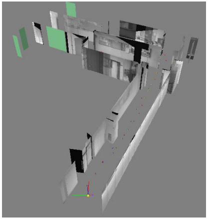


Pose-Based

- Estimator maintains the estimate of all past robot poses.
- Sensor scan is anchored to each pose.
- Efficient if $p < n$.

Map Types



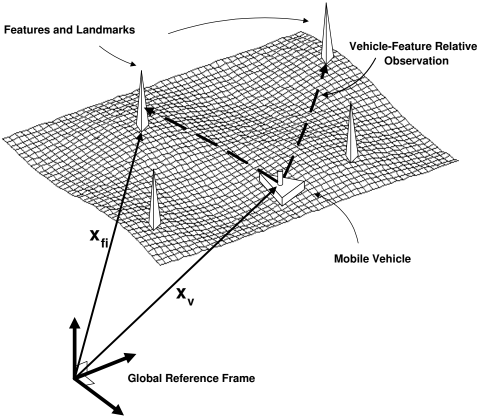


Feature-Based

- Estimator maintains the estimate of map features.
- Features can be parameterized as points, lines, planes, etc.
- Efficient if $n < p$.

Simple Example

To explain the concepts of localization and mapping we'll use a simple example.

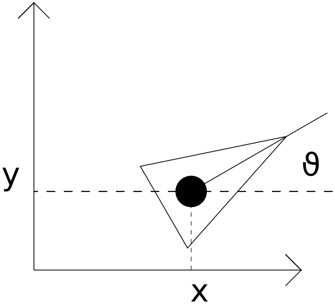


Simple Example

- 2D world
- Point features

Simple Example

The robot moves in 2D.

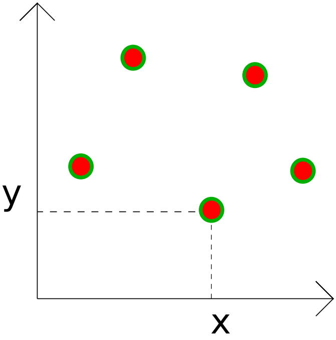


Robot Pose

$$\mathbf{x}_v = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$$

Simple Example

The world is a set of points.



Map Representation

A point feature can be described by its 2d coordinates

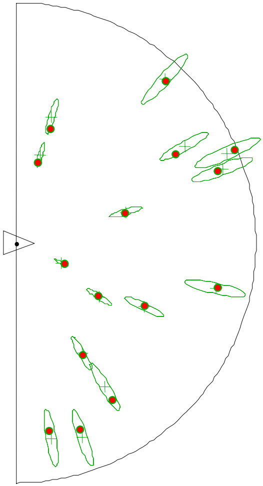
$$\mathbf{x}_{f_i} = \begin{bmatrix} x_i \\ y_i \end{bmatrix}$$

The map $\mathbf{M}$ is the concatenation of all of the feature parameters

$$\mathbf{M} = \begin{bmatrix} \mathbf{x}_{f_1} \\ \mathbf{x}_{f_2} \\ \dots \\ \mathbf{x}_{f_n} \end{bmatrix}$$

Simple Example

Range and bearing to point features from the robot is measured.



Observations

$$\mathbf{z} = \begin{bmatrix} r \\ \theta \end{bmatrix}$$

Probabilistic Localization

Determine

The vehicle pose $\mathbf{x}_v$

Given

- Map of features $\mathbf{M}$
- Observations $\mathbf{Z}^k$
- Observations likelihood model $p(\mathbf{Z}^k|\mathbf{M}, \mathbf{x}_v)$

Probabilistic Localization

$$p(\mathbf{x}_v|\mathbf{M},\mathbf{Z}^k) = \frac{p(\mathbf{Z}^k|\mathbf{M},\mathbf{x}_v)p(\mathbf{M},\mathbf{x}_v)}{p(\mathbf{M},\mathbf{Z}^k)}$$
$$= \frac{p(\mathbf{Z}^k|\mathbf{M},\mathbf{x}_v)p(\mathbf{x}_v|\mathbf{M})p(\mathbf{M})}{p(\mathbf{M},\mathbf{Z}^k)}$$
$$= \frac{p(\mathbf{Z}^k|\mathbf{M},\mathbf{x}_v)p(\mathbf{x}_v|\mathbf{M})p(\mathbf{M})}{\int_{-\infty}^{\infty} p(\mathbf{Z}^k|\mathbf{M},\mathbf{x}_v)p(\mathbf{x}_v|\mathbf{M})p(\mathbf{M})d\mathbf{x}_v}$$

but with a perfect map everything is independent of $\mathbf{M}$

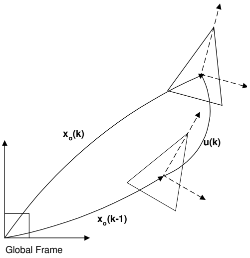
$$p(\mathbf{x}_v|\mathbf{M},\mathbf{Z}^k) = \frac{p(\mathbf{Z}^k|\mathbf{x}_v)p(\mathbf{x}_v)}{\int_{-\infty}^{\infty} p(\mathbf{Z}^k|\mathbf{x}_v)p(\mathbf{x}_v)d\mathbf{x}_v}$$
$$= \frac{p(\mathbf{Z}^k|\mathbf{x}_v)p(\mathbf{x}_v)}{C(\mathbf{Z}^k)}$$

Probabilistic Localization

Extended Kalman Filter

1. Predict State
2. Make Observations
3. Update State

Probabilistic Localization



- function f is the motion model
- $\mathtt{F}$ is Jacobian of the motion model
- $\mathbf{u}$ is the control inputs
- $\mathtt{Q}$ is noise on the control inputs

Motion Model Prediction

Predict new robot pose from odometry.

$$\hat{\mathrm{x}}(k|k-1) = f(\hat{\mathrm{x}}(k-1|k-1), \mathbf{u})$$

Noisy motion model increases the uncertainty.

$$\mathtt{P}(k|k-1) = \mathtt{F}_x \mathtt{P}(k-1|k-1)\mathtt{F}_x^\top + \mathtt{F}_u \mathtt{Q} \mathtt{F}_u^\top$$

Probabilistic Localization

Observation Model

Predict range and bearing observations given the predicted robot pose and the known feature position.

$$\begin{aligned}
\mathbf{z}(k|k-1) &= \underbrace{h(\hat{\mathbf{x}}(k|k-1))}_{\text{Observation Model}} \\
&= \begin{bmatrix} \sqrt{(x_i - x_v(k))^2 + (y_i - y_v(k))^2} \\ atan2\left(\frac{y_i - y_v(k)}{x_i - x_v(k)}\right) - \theta_v \end{bmatrix} \\
\mathtt{H}_x &= \begin{bmatrix} \frac{x_i - x_v(k)}{r} & \frac{y_i - y_v(k)}{r} & 0 \\ \frac{y_i - y_v(k)}{r^2} & -\frac{x_i - x_v(k)}{r^2} & -1 \end{bmatrix} \\
\mathtt{R} &= \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\theta^2 \end{bmatrix} \text{Assumed Sensor Noise}
\end{aligned}$$

Probabilistic Localization

EKF Update

$$\underbrace{\hat{\mathbf{x}}(k|k)}_{\text{new state estimate}} = \underbrace{\hat{\mathbf{x}}(k|k-1) + \mathtt{W}\boldsymbol{\nu}(k)}_{\text{prediction and correction}}$$

$$\underbrace{\mathtt{P}(k|k)}_{\text{new covariance estimate}} = \underbrace{\mathtt{P}(k|k-1) - \mathtt{W}\mathtt{S}\mathtt{W}^\top}_{\text{update decreases uncertainty}}$$

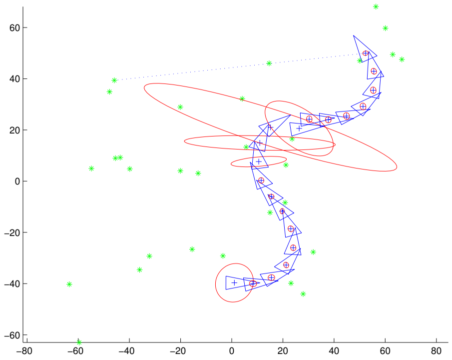
where

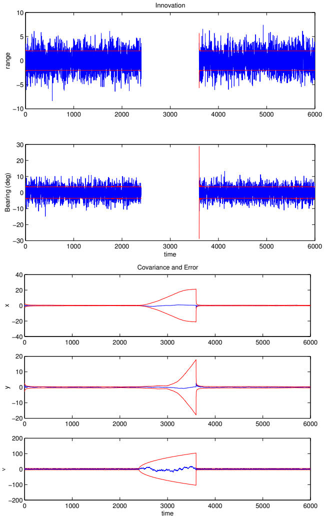
$$\overbrace{\boldsymbol{\nu}(k)}^{\text{innovation}} = \overbrace{\mathbf{z}(k)}^{\text{measurement}} - \overbrace{\mathbf{z}(k|k-1)}^{\text{predicted measurement}}$$

$$\mathtt{W} = \underbrace{\mathtt{P}(k|k-1)\mathtt{H}_x^\top \mathtt{S}^{-1}}_{\text{Kalman Gain}}$$

$$\mathtt{S} = \underbrace{\mathtt{H}_x \mathtt{P}(k|k-1)\mathtt{H}_x^\top + \mathtt{R}}_{\text{Innovation Covariance}}$$

Probabilistic Localization





Probabilistic Mapping

Determine

Map of features $\mathbf{M}$

Given

- The vehicle pose $\mathbf{x}_v$
- Observations $\mathbf{Z}^k$
- Observations likelihood model $p(\mathbf{Z}^k|\mathbf{M},\mathbf{x}_v)$

Probabilistic Mapping

$$
\begin{aligned}
p(\mathbf{M}|\mathbf{x}_v, \mathbf{Z}^k) &= \frac{p(\mathbf{Z}^k|\mathbf{M}, \mathbf{x}_v)p(\mathbf{M}, \mathbf{x}_v)}{p(\mathbf{x}_v, \mathbf{Z}^k)} \\
&= \frac{p(\mathbf{Z}^k|\mathbf{M}, \mathbf{x}_v)p(\mathbf{M}|\mathbf{x}_v)p(\mathbf{x}_v)}{p(\mathbf{x}_v, \mathbf{Z}^k)} \\
&= \frac{p(\mathbf{Z}^k|\mathbf{M}, \mathbf{x}_v)p(\mathbf{M}|\mathbf{x}_v)p(\mathbf{x}_v)}{\int_{-\infty}^{\infty} p(\mathbf{Z}^k|\mathbf{M}, \mathbf{x}_v)p(\mathbf{M}|\mathbf{x}_v)p(\mathbf{x}_v)d\mathbf{M}}
\end{aligned}
$$

but we have perfect localization so everything is independent of $\mathbf{x}_v$

$$
\begin{aligned}
p(\mathbf{M}|\mathbf{x}_v, \mathbf{Z}^k) &= \frac{p(\mathbf{Z}^k|\mathbf{M})p(\mathbf{M})}{\int_{-\infty}^{\infty} p(\mathbf{Z}^k|\mathbf{M})p(\mathbf{M})d\mathbf{M}} \\
&= \frac{p(\mathbf{Z}^k|\mathbf{M})p(\mathbf{M})}{C(\mathbf{Z}^k)}
\end{aligned}
$$

Probabilistic Mapping

Since all map features are stationary the prediction step leaves the estimate unchanged.

Prediction Model

$$\hat{\mathbf{x}}(k+1|k)_{map} = \hat{\mathbf{x}}(k|k)_{map}$$

Probabilistic Mapping

A predicted range and bearing observation only depends on the predicted state for the feature being observed. The robot pose is known with certainty.

Observation Model

$$\mathbf{z}(k) = \begin{bmatrix} r \\ \theta \end{bmatrix}$$

$$\mathtt{H}_x = \begin{bmatrix} \underbrace{\ldots 0 \ldots}_{\text{other features}} & \underbrace{\mathtt{H}_{x f_i}}_{\text{observed feature}} & \underbrace{\ldots 0 \ldots}_{\text{other features}} \end{bmatrix}$$

Probabilistic Mapping

As new features are observed they must be added to the state using thier initial measurement.

Adding New Features to the State

$$\hat{\mathbf{x}}(k|k)^* = \begin{bmatrix} \hat{\mathbf{x}}(k|k) \\ \hat{\mathbf{x}}_{f_{new}} \end{bmatrix}$$

$$\begin{aligned} \hat{\mathbf{x}}_{f_{new}} &= g(\mathbf{x}_v(k|k), \mathbf{z}(k|k)) \\ &= \begin{bmatrix} x_v + r\cos(\theta + \theta_v) \\ y_v + r\sin(\theta + \theta_v) \end{bmatrix} \end{aligned}$$

Probabilistic Mapping

The covariance for the new feature must also be calculated.

Adding New Features to the State

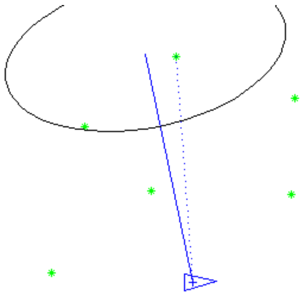
$$\mathtt{P}(k|k)^* = \mathtt{Y} \begin{bmatrix} \mathtt{P}(k|k) & 0 \\ 0 & \mathtt{R} \end{bmatrix} \mathtt{Y}^\top$$

$$\mathtt{Y} = \begin{bmatrix} \mathtt{I}_{n\times n} & 0_{n\times 2} \\ \mathtt{G}_x & \mathtt{G}_z \end{bmatrix}$$

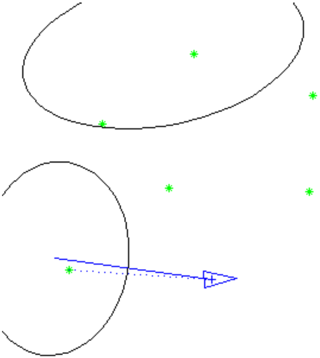
but $\mathtt{G}_x = 0$ since the state only contains features.

$$\mathtt{P}(k|k)^* = \begin{bmatrix} \mathtt{P}(k|k) & 0 \\ 0 & \mathtt{G}_z \mathtt{R} \mathtt{G}_z^\top \end{bmatrix}$$

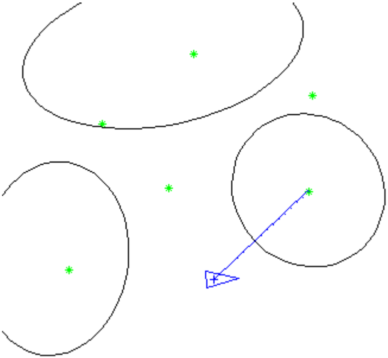
Probabilistic Mapping



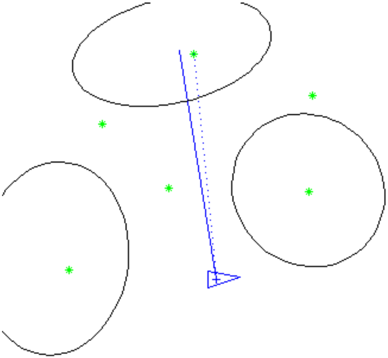
Probabilistic Mapping



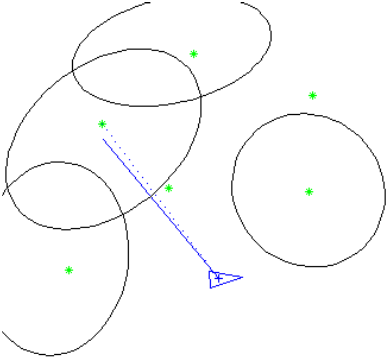
Probabilistic Mapping



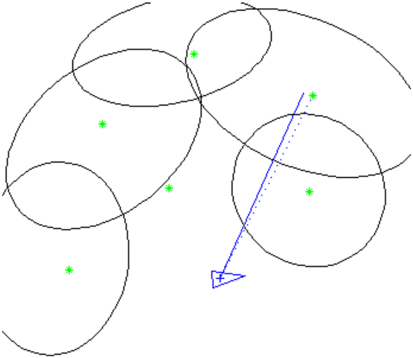
Probabilistic Mapping



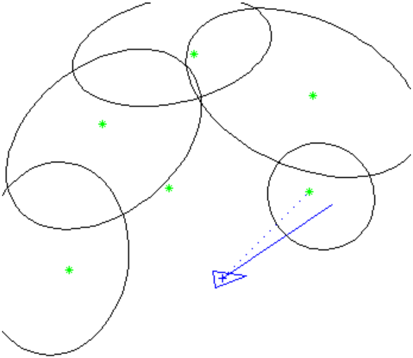
Probabilistic Mapping



Probabilistic Mapping



Probabilistic Mapping



Probabilistic Mapping

Mapping Video

Probabilistic Mapping

Covariance Matrix

The covariance matrix is block diagonal.

$$P = \begin{bmatrix} P_{f_1} & 0 & \dots & 0 \\ 0 & P_{f_2} & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & P_{f_n} \end{bmatrix}$$

Each feature estimate is independent.

Localization and Mapping

Localization

- Localization with a known map is easy.
- Small state to estimate (only the robot pose).

Localization and Mapping

Localization

- Localization with a known map is easy.
- Small state to estimate (only the robot pose).

Mapping

- Mapping from known poses is easy.
- Many small states to estimate independently (each map feature).

Localization and Mapping

Localization

- Localization with a known map is easy.
- Small state to estimate (only the robot pose).

Mapping

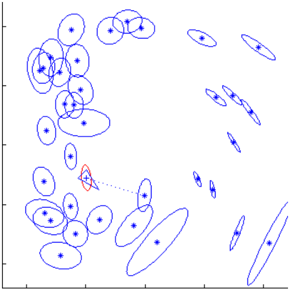
- Mapping from known poses is easy.
- Many small states to estimate independently (each map feature).

Simultaneous Localization and Mapping

Chicken and egg problem:

- a map is needed to localize the robot.
- a pose estimate is needed to build a map.

EKF SLAM



Simultaneous Localization and Mapping

- Noisy observations of uncertain map features are made from an uncertain vehicle position.
- Vehicle pose and map must be estimated *jointly*.
- The entire state becomes correlated making it more expensive to compute.

EKF SLAM

Since localization and mapping is being performed simultaneously, the state contains both the vehicle pose estimate and the map estimate.

SLAM State

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{\mathbf{x}}_v \\ \hat{\mathbf{x}}_{f_1} \\ \vdots \\ \hat{\mathbf{x}}_{f_n} \end{bmatrix}$$

$$\mathrm{P} = \begin{bmatrix} \mathrm{P}_{vv} & \mathrm{P}_{vm} \\ \mathrm{P}_{vm}^\top & \mathrm{P}_{mm} \end{bmatrix}$$

EKF SLAM

Motion model affects the robot pose estimate but not the stationary map.

Motion Model Prediction

$$\begin{bmatrix} \hat{\mathbf{x}}_v(k+1) \\ \hat{\mathbf{x}}_{f_1}(k+1) \\ \vdots \\ \hat{\mathbf{x}}_{f_n}(k+1) \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{x}}_v(k) \oplus \mathbf{u}(k) \\ \hat{\mathbf{x}}_{f_1}(k) \\ \vdots \\ \hat{\mathbf{x}}_{f_n}(k) \end{bmatrix}$$

$$\mathtt{P}(k|k-1) = \mathtt{F}_x \mathtt{P}(k-1|k-1)\mathtt{F}_x^\top + \mathtt{F}_u \mathtt{Q} \mathtt{F}_u^\top$$

$$\mathtt{F}_x = \begin{bmatrix} \mathtt{J1}(\mathbf{x}_v, \mathbf{u}) & 0 \\ 0 & \mathtt{I}_{2n\times 2n} \end{bmatrix}$$

$$\mathtt{F}_u = \begin{bmatrix} \mathtt{J2}(\mathbf{x}_v, \mathbf{u}) & 0 \\ 0 & 0_{2n\times 2n} \end{bmatrix}$$

EKF SLAM

A predicted observation depends on the both robot pose estimate and the estimated state for the feature being observed.

Observation Model Jacobian

$$\mathtt{H}_x = \left[\underbrace{\mathtt{H}_{x_v}}_{\text{vehicle}} \quad \ldots 0 \ldots \quad \underbrace{\mathtt{H}_{x_{f_i}}}_{\text{observed feature}} \quad \ldots 0 \right]$$

EKF SLAM

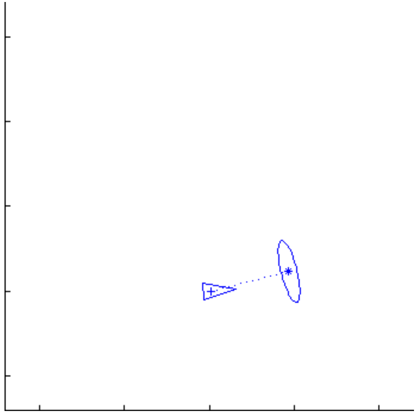
The initial state for a new feature depends on the both robot pose estimate and the observation.

New Feature Jacobian

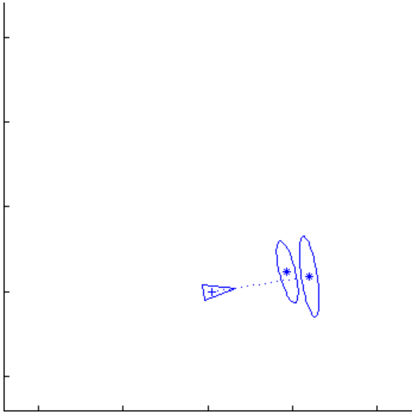
$$\mathtt{Y} = \begin{bmatrix} \mathtt{I}_{n\times n} & \mathtt{0}_{n\times 2} \\ \mathtt{G}_x & \mathtt{G}_z \end{bmatrix}$$

$$\mathtt{Y} = \begin{bmatrix} \mathtt{I}_{n\times n} & \mathtt{0}_{n\times 2} \\ \begin{bmatrix} \mathtt{G}_{x_v} & \ldots \mathtt{0} \ldots \end{bmatrix} & \mathtt{G}_z \end{bmatrix}$$

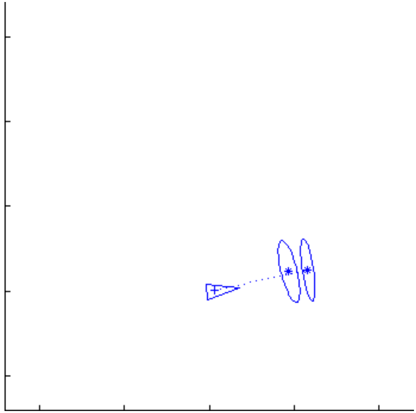
EKF SLAM



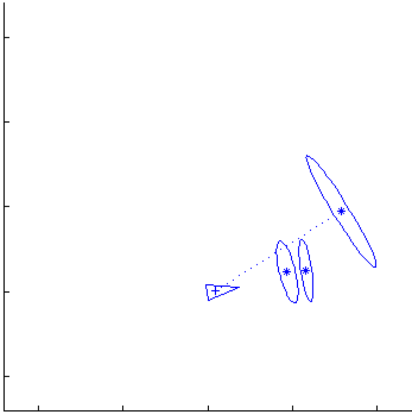
EKF SLAM



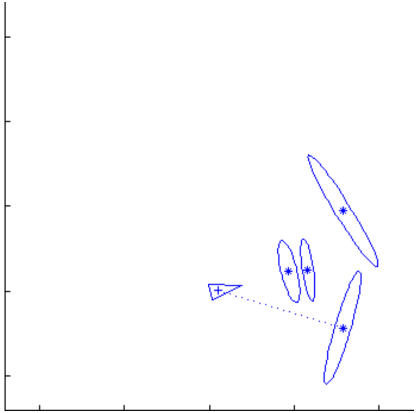
EKF SLAM



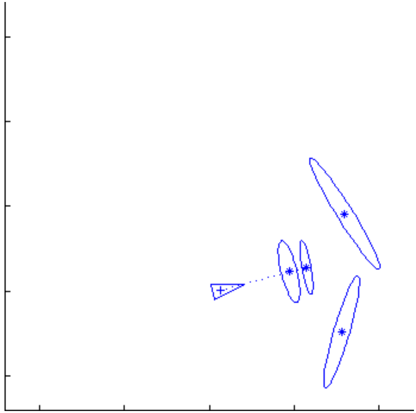
EKF SLAM



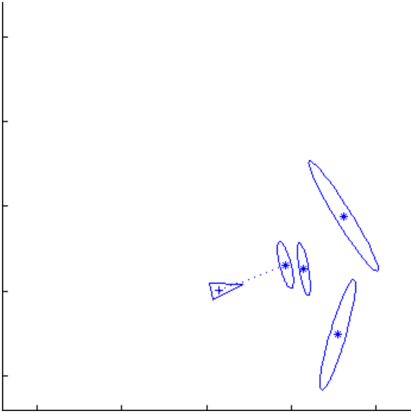
EKF SLAM



EKF SLAM



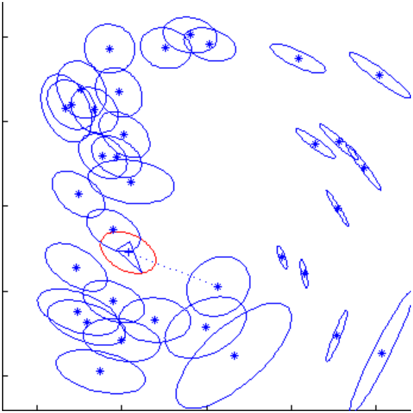
EKF SLAM



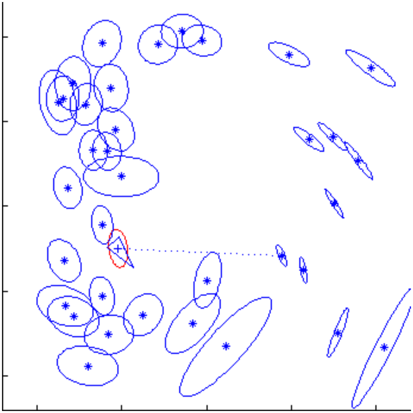
EKF SLAM

EKF SLAM Video

EKF SLAM



EKF SLAM



EKF SLAM

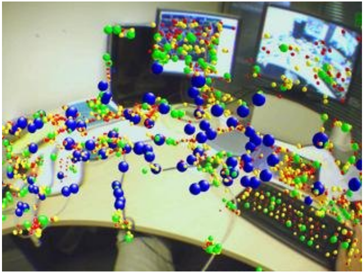
EKF SLAM

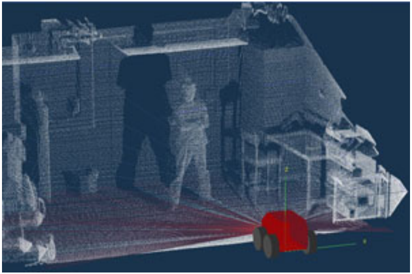
- Quadratic in number of landmarks $O(n^2)$.
- Convergence results for the linear case.
- Can diverge if nonlinearities are large!
- Has been applied successfully in large-scale environments.
- Approximations reduce the computational complexity.

Other Approaches to SLAM

Pose-Based

- Estimate full robot trajectory (rather than current pose).
  - Used when sensor scans are very dense.
  - Takes advantage of sparsity.
  - Vision-based bundle adjustment.
  - Laser scan matching.

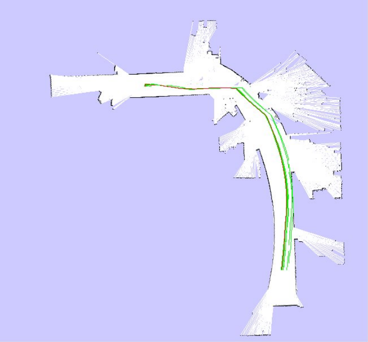




Other Approaches to SLAM

Rao-Blackwellized Particle Filtering (FastSLAM)

- Estimate robot trajectory using a particle filter.
- Estimate map features independently (one map for each particle).



FastSLAM

Why not use an ordinary particle filter?

- Localization: state space $< x_v >$
- SLAM: state space $< x_v, Map >$
  - For feature based maps $= < x_{f_1}, x_{f_2}, \ldots, x_{f_n} >$
  - For grid based maps $= < x_{c11}, x_{c12}, \ldots, x_{c1n}, x_{c21}, \ldots, x_{cnm} >$

Problem: The number of particles needed to represent a posterior grows exponentially with the dimension of the state space!

FastSLAM

Rao-Blackwellized Factorization

$$p(x_{v_{1:k}}, x_{f_{1:n}} | z_{1:k}, u_{1:k})$$

FastSLAM

Rao-Blackwellized Factorization

$$p(x_{v_{1:k}}, x_{f_{1:n}} | z_{1:k}, u_{1:k}) \quad = \quad p(x_{v_{1:k}} | z_{1:k}, u_{1:k}) \cdot p(x_{f_{1:n}} | x_{v_{1:k}}, z_{1:k})$$

FastSLAM

Rao-Blackwellized Factorization

$$
\begin{aligned}
p(x_{v_{1:k}}, x_{f_{1:n}} | z_{1:k}, u_{1:k}) &= p(x_{v_{1:k}} | z_{1:k}, u_{1:k}) \cdot p(x_{f_{1:n}} | x_{v_{1:k}}, z_{1:k}) \\
&= \underbrace{p(x_{v_{1:k}} | z_{1:k}, u_{1:k})}_{\substack{\text{Robot path posterior} \\ \text{(localization problem)}}} \cdot \underbrace{\prod_{i=1}^{n} p(x_{f_i} | x_{v_{1:k}}, z_{1:k})}_{\substack{\text{Conditionally independent} \\ \text{landmark positions}}}
\end{aligned}
$$

FastSLAM

Rao-Blackwellized Factorization

$$
\begin{aligned}
p(x_{v_{1:k}}, x_{f_{1:n}} | z_{1:k}, u_{1:k}) &= p(x_{v_{1:k}} | z_{1:k}, u_{1:k}) \cdot p(x_{f_{1:n}} | x_{v_{1:k}}, z_{1:k}) \\
&= \underbrace{p(x_{v_{1:k}} | z_{1:k}, u_{1:k})}_{\substack{\text{Robot path posterior} \\ \text{(localization problem)}}} \cdot \underbrace{\prod_{i=1}^{n} p(x_{f_i} | x_{v_{1:k}}, z_{1:k})}_{\substack{\text{Conditionally independent} \\ \text{landmark positions}}}
\end{aligned}
$$

- Each particle represents a different hypothesis for the entire robot trajectory.

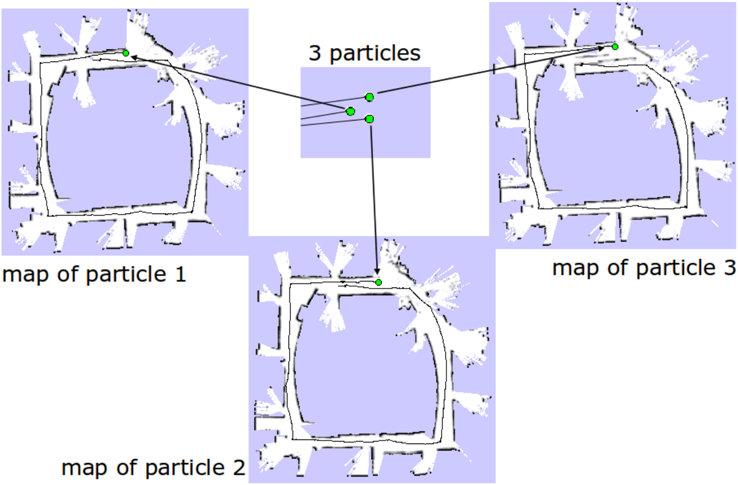
FastSLAM

Rao-Blackwellized Factorization

$$\begin{aligned} p(x_{v_{1:k}}, x_{f_{1:n}} | z_{1:k}, u_{1:k}) &= p(x_{v_{1:k}} | z_{1:k}, u_{1:k}) \cdot p(x_{f_{1:n}} | x_{v_{1:k}}, z_{1:k}) \\ &= \underbrace{p(x_{v_{1:k}} | z_{1:k}, u_{1:k})}_{\substack{\text{Robot path posterior} \\ \text{(localization problem)}}} \cdot \underbrace{\prod_{i=1}^{n} p(x_{f_i} | x_{v_{1:k}}, z_{1:k})}_{\substack{\text{Conditionally independent} \\ \text{landmark positions}}} \end{aligned}$$

- Each particle represents a different hypothesis for the entire robot trajectory.
- Conditioned on the trajectory, the map features are independent.

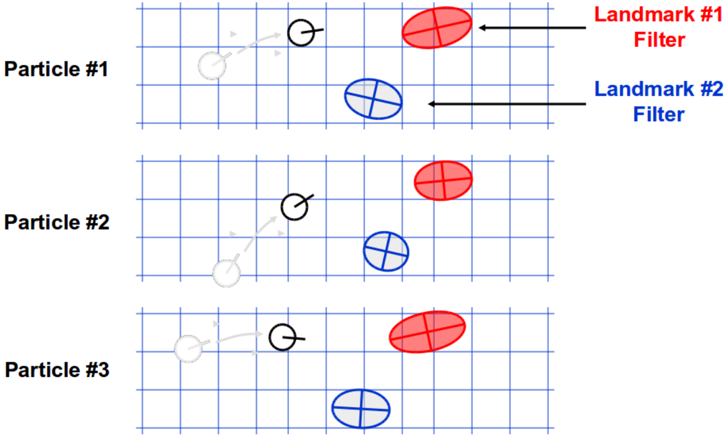
FastSLAM



3 particles
map of particle 1
map of particle 2
map of particle 3

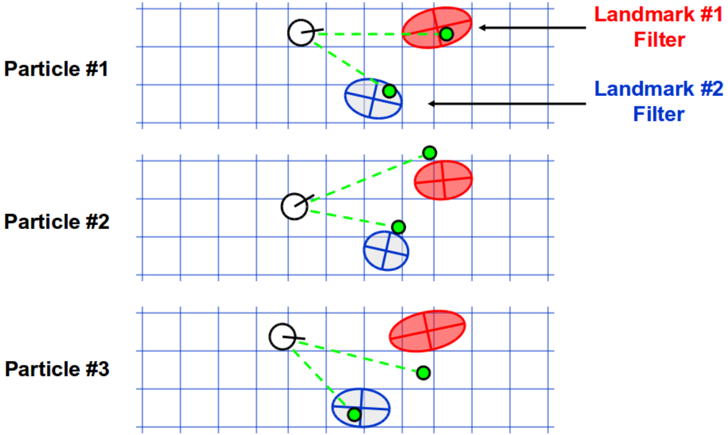
FastSLAM

Motion Prediction



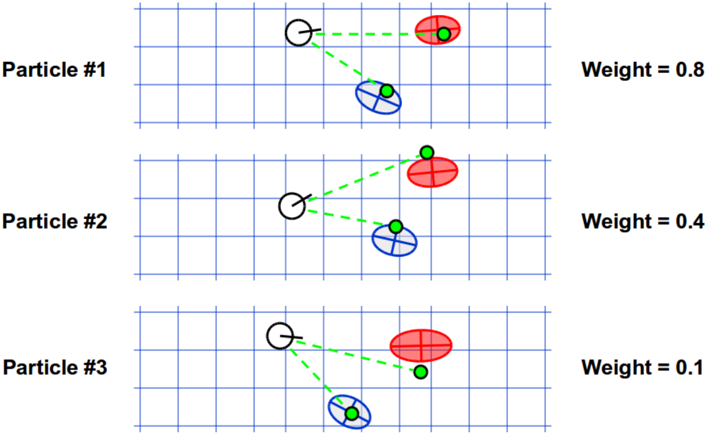
FastSLAM

Observation Update



FastSLAM

Observation Update



FastSLAM

FastSLAM Video

FastSLAM

FastSLAM

- Can handle non-linear motion models well.
- Can handle complex maps efficiently since features are estimated independently.
- Efficient map storage is necessary since one map is required per particle.
- A large number of particles is required for large environments.

Conclusion

Map types

- Topological
- Metric

Conclusion

Map types

- Topological
- Metric

Estimation

- Localization from a known map is easy.
- Mapping from known robot poses is easy.
- Simultaneous localization and mapping (SLAM) requires a joint estimate taking into account correlations.

Conclusion

Map types

- Topological
- Metric

Estimation

- Localization from a known map is easy.
- Mapping from known robot poses is easy.
- Simultaneous localization and mapping (SLAM) requires a joint estimate taking into account correlations.

Unresolved issues (next lecture)

- Which feature does an observation correspond to?
- How to recognise when the robot has gone around a loop?
- How to handle multi-robot mapping?