Multi-robot localization

Miroslav Kulich

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

Wednesday 9/11/2016

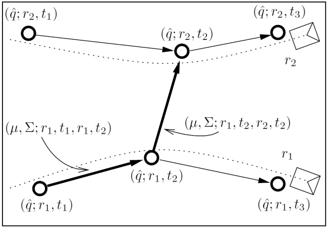
Localization in a team of robots

- Every robot localizes independently
- Maximum likelihood estimation
- Particle filter
- Extended Kalman filter

- We talk about localization, i.e. a map of the environment is known in advance
- SLAM approaches for multi-robot teams exist, but these are out of scope of the course

Maximum likelihood estimation

Howard, Matarić, Sukhatme (2002)



Input:

- The set of measurements: $O = \{o\}$, where $o = (\mu, \Sigma, r_a, t_a, r_b, t_b)$, μ is the measured robot position r_b at time t_b relatively to the robot t_a at time t_a.
  - Odometry: $o = (\mu, \Sigma, r_a, t_a, r_a, t_b)$
  - Measurement: $o = (\mu, \Sigma, r_a, t_a, r_a, t_b)$

Output:

- The set of positions estimates: $H = \{h\}$, where $h = (\hat{q}, r, t)$, $\hat{q}$ is estimate of robot's position r at time t.

Maximum likelihood estimation

- We want to determine a set of positions H, which maximizes probability of a measurement set O, i.e. maximizes $P(O|H)$.
- Assume the measurements are independent:

$$P(O|H) = \prod_{o \in O} P(o|H)$$

- After performing log **minimization**:

$$U(O|H) = \sum_{o \in O} U(o|H),$$

  where $U(O|H) = -\log P(O|H)$ and $U(o|H) = -\log P(o|H)$

- Assume normal distribution for measurement uncertainty:

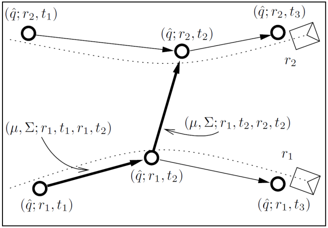
$$U(o|H) = \frac{1}{2}(\mu - \hat{\mu})^T \Sigma (\mu - \hat{\mu})$$

- Motion model: $\hat{\mu} = \Gamma(\hat{q}_a, \hat{q}_b)$
- Optimization by a standard numerical techniques (gradient descent, steepest descent)

Maximum likelihood estimation

Practical notes

- Dimensionality of the problem increases linearly with H and every step of the optimization process increases linearly with O.
- To decrease complexity we apply:
  - Filtering of old measurements.
  - Filtering of similar measurements.
  - Limiting of the rate at which pose estimates are generated.



Particle filter

- Extension of a standard particle filter.
- Integration of *detection* - one robot „sees“ the other one.
- Naive approach: state space incorporates positions of all robots:

$$x_t = x_t^1 \times x_t^2 \dots x_t^N$$

- Dimensionality increases linearly with the number of robots and the number of particles x_t exponentially.
- Factorization:

$$p(x_t^1, x_t^2, \dots, x_t^N | d^{(t)}) = p(x_t^1 | d^{(t)}) \cdot p(x_t^2 | d^{(t)}) \cdot \dots \cdot p(x_t^N | d^{(t)})$$

- Every robot keeps only its own position and only if the robot detect another one, information is exchanged.
- It is approximation only, positions of robots are not independent!

Particle filter

- Assume the following data:
  - Odometry - motion integration

$$Bel(x_t^n) = \int p(x_t^n | x_{t-1}^n, u_t^n) Bel(x_{t-1}^n)$$

  - Sensor measurement

$$Bel(x_t^n) = p(z | x_t^n) Bel(x_t^n)$$

  - Detection of other robots

Particle filter

Derivation of equations for detection

- The robot R_n detects another robot R_m by measuring r_t^m.

$$\begin{aligned} Bel(x_t^n) &= p(x_t^n | d_{(t)}^n) \\ &= p(x_t^n | d_{(t-1)}^n) p(x_t^n | d_t^m) \\ &= p(x_t^n | d_{(t-1)}^n) \int p(x_t^n | x_t^m, r_t^m) p(x_t^m | d_{(t-1)}^m) \end{aligned}$$

- Which leads to:

$$Bel(x_t^n) = Bel(x_{t-1}^n) \int p(x_t^n | x_t^m, r_t^m) Bel(x_t^m) dx_t^m$$

- Update of m-th robot's position is done symmetrically.

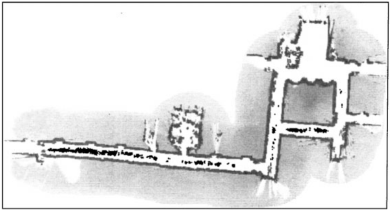
Particle filter

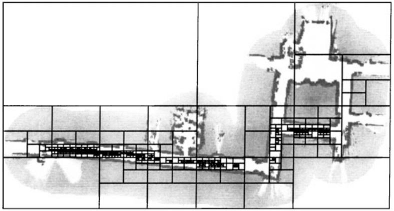
Implementation

- Extension of the particle filter for multiple robots is not straightforward – how to multiply two sets of particles?

$$Bel(x_t^n) = Bel(x_{t-1}^n) \int p(x_t^n | x_t^m, r_t^m) Bel(x_t^m) dx_t^m$$

- Idea: transform a set of particles for m into a *density tree*:
    - Recursive space division using piece-wise constant density functions.
    - Node (leaf) density is a sum of weights of particles divided by a volume of the node.
    - Weight of a particle R_n is multiplied by corresponding density.





Particle filter

Problems

- Frequency of detection is high $\rightsquigarrow$ a single detection is integrated many times.
- Identification of robots is needed.
- False-positive detection - robots „see“ each other with relatively low frequency $\rightsquigarrow$ small amount of false-positive plays a big role.
- Positive information only - negative information can be incorporated in general, but it is computationally demanding.
- Delayed integration - in case of high uncertainty of pose determination. It is necessary to keep information about all actions and measurements.

Extended Kalman filter

- Configuration of i-th robot $X_i = (x_i, y_i, \theta_i)$
- We aim to estimate the state

$$X = (X_1, X_2, \dots, X_N)$$

- Covariance matrix:

$$\Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} & \dots & \Sigma_{1N} \\ \Sigma_{21} & \Sigma_{22} & \dots & \Sigma_{2N} \\ \dots & \dots & \dots & \dots \\ \Sigma_{N1} & \Sigma_{N2} & \dots & \Sigma_{NN} \end{pmatrix}$$

Extended Kalman filter

Correction (measurement integration)
$K = \Sigma H^T (H \Sigma H^T + Q)^{-1}$
$\mu = \mu + K(z - h(\mu))$
$\Sigma = (E - KH)\Sigma$

- Detection (i-th robot „sees“ j-th)

$$z = h(X_i, X_j) + w$$

- Jacobian H:

$$H = (0, \ldots, 0, H_i, 0, \ldots, 0, H_j, 0, \ldots, 0),$$

- and

$$\begin{aligned}
H \Sigma H^T + Q &= H_i \Sigma_{ii} H_i^T + H_i \Sigma_{ij} H_j^T + H_j \Sigma_{ji} H_i^T + H_j \Sigma_{jj} H_j^T + Q = P_{zz} \\
\mu_l &= \mu_l + (\Sigma_{li} H_i^T + \Sigma_{lj} H_j^T) P_{zz}^{-1} (z - h(\mu_i, \mu_j)) \\
\Sigma_{lf} &= \Sigma_{lf} - (\Sigma_{li} H_i^T + \Sigma_{lj} H_j^T) P_{zz}^{-1} (H_i \Sigma_{if} + H_j \Sigma_{jf})
\end{aligned}$$

Extended Kalman filter

- Distance:

$$
\begin{aligned}
h_d(X_i, Y_i) &= \sqrt{\Delta x^2 + \Delta y^2} \\
H_i^d &= \left(\frac{-\Delta x}{\sqrt{\Delta x^2 + \Delta y^2}}, \frac{-\Delta y}{\sqrt{\Delta x^2 + \Delta y^2}}, 0 \right) \\
H_j^d &= \left(\frac{\Delta x}{\sqrt{\Delta x^2 + \Delta y^2}}, \frac{\Delta y}{\sqrt{\Delta x^2 + \Delta y^2}}, 0 \right)
\end{aligned}
$$

- Relative direction:

$$
\begin{aligned}
h_b(X_i, Y_i) &= \arctan \left(\frac{-\sin\theta_i \Delta x + \cos\theta_i \Delta y}{\cos\theta_i \Delta x + \sin\theta_i \Delta y} \right) \\
H_i^b &= \left(\frac{\Delta y}{\Delta x^2 + \Delta y^2}, \frac{-\Delta x}{\Delta x^2 + \Delta y^2}, -1 \right) \\
H_j^b &= \left(\frac{-\Delta y}{\Delta x^2 + \Delta y^2}, \frac{\Delta x}{\Delta x^2 + \Delta y^2}, 0 \right)
\end{aligned}
$$

Extended Kalman filter

- Relative orientation:

$$
\begin{aligned}
h_o(X_i, Y_i) &= \theta_j - \theta_i \\
H_i^o &= (0, 0, -1) \\
H_j^o &= (0, 0, 1)
\end{aligned}
$$

$$
\begin{aligned}
\Delta x &= x_j - x_i \\
\Delta y &= y_j - y_i
\end{aligned}
$$