Introduction to Mobile Robotics

EKF Localization

Wolfram Burgard, Cyrill Stachniss, Maren Bennewitz, Kai Arras

Slides by Kai Arras and Wolfram Burgard
Last update: June 2010
Localization

“Using sensory information to locate the robot in its environment is the most fundamental problem to providing a mobile robot with autonomous capabilities.” [Cox '91]

• **Given**
  - Map of the environment.
  - Sequence of sensor measurements.

• **Wanted**
  - Estimate of the robot’s position.

• **Problem classes**
  - Position tracking
  - Global localization
  - Kidnapped robot problem (recovery)
Landmark-based Localization

**EKF Localization:** Basic Cycle
Landmark-based Localization

**EKF Localization: Basic Cycle**
Landmark-based Localization

EKF Localization: Basic Cycle

- **Encoder Measurements**: Odometry or IMU
- **State Prediction**: Predicted state
- **Measurement Prediction**: Predicted measurements in sensor coordinates
- **Data Association**: Innovation from matched landmarks
- **Update**: Posterior state

**Landmarks in Global Coordinates**: Map

**Raw Sensory Data**: Sensors

**Feature/Landmark Extraction**: Landmarks in sensor coordinates
Landmark-based Localization

State Prediction (Odometry)

\[
\hat{x}_k = f(x_{k-1}, u_k)
\]
\[
\hat{C}_k = F_x C_k F_x^T + F_u U_k F_u^T
\]

Control \( u_k \): wheel displacements \( s_l, s_r \)

\[
u_k = (s_l \ s_r)^T \quad U_k = \begin{bmatrix} \sigma_l^2 & 0 \\ 0 & \sigma_r^2 \end{bmatrix}
\]

Error model: linear growth

\[
\sigma_l = k_l \ |s_l| \\
\sigma_r = k_r \ |s_r|
\]

Nonlinear process model \( f \):

\[
x_k = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \end{bmatrix} + \begin{bmatrix} \frac{b}{2} \frac{s_l + s_r}{s_r - s_l} (-\sin \theta_{k-1} + \sin(\theta_{k-1} + \frac{s_r - s_l}{b})) \\ \frac{b}{2} \frac{s_l + s_r}{s_r - s_l} (\cos \theta_{k-1} - \cos(\theta_{k-1} + \frac{s_r - s_l}{b})) \\ \frac{s_r - s_l}{b} \end{bmatrix}
\]
Landmark-based Localization

State Prediction (Odometry)

\[
\hat{x}_k = f(x_{k-1}, u_k)
\]

\[
\hat{C}_k = F_x C_k F_x^T + F_u U_k F_u^T
\]

Control \( u_k \): wheel displacements \( s_l, s_r \)

\[
u_k = (s_l \ s_r)^T \quad U_k = \begin{bmatrix} \sigma_l^2 & 0 \\ 0 & \sigma_r^2 \end{bmatrix}
\]

Error model: linear growth

\[
s_l = k_l |s_l| \\
\sigma_r = k_r |s_r|
\]

Nonlinear process model \( f \):

\[
x_k = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \end{bmatrix} + \begin{bmatrix} \frac{b}{2} (s_l + s_r) \\ \frac{b}{2} (s_l - s_r) \\ \frac{b}{2} (s_l + s_r) \end{bmatrix} \left( \begin{array}{c}
- \sin(\theta_{k-1} + \frac{s_r - s_l}{b}) \\
\cos(\theta_{k-1} - \cos(\theta_{k-1} + \frac{s_r - s_l}{b})) \\
\frac{s_r - s_l}{b} 
\end{array} \right)
\]
Landmark-based Localization

Landmark Extraction (Observation)

Hessian line model

\[ x \cos(\alpha) + y \sin(\alpha) - r = 0 \]
Landmark-based Localization

Measurement Prediction

- ...is a coordinate frame transform world-to-sensor
- Given the predicted state (robot pose), predicts the location $\hat{z}_k$ and location uncertainty $H \hat{C}_k H^T$ of expected observations in sensor coordinates

$$\hat{z}_k = h(\hat{x}_k, m)$$

[Diagram showing map and model space]
Data Association (Matching)

- Associates predicted measurements $\hat{z}_k^i$ with observations $z_k^j$
  \[
  \nu_k^{ij} = z_k^j - \hat{z}_k^i \\
  S_k^{ij} = R_k^j + H_k^i \hat{C}_k H_k^i T
  \]

- Innovation $\nu_k^{ij}$ and innovation covariance $S_k^{ij}$

- Matching on significance level alpha

Green: observation
Magenta: measurement prediction

No match!!
Wall was not observed.
Landmark-based Localization

Update

- Kalman gain

\[ K_k = \hat{C}_k H^T S_k^{-1} \]

- State update (robot pose)

\[ x_k = \hat{x}_k + K_k \nu_k \]

- State covariance update

\[ C_k = (I - K_k H) \hat{C}_k \]

Red: posterior estimate
Landmark-based Localization

- EKF Localization with Point Features
1. **EKF_localization** \((\mu_{t-1}, \Sigma_{t-1}, u_t, z_t, m)\):

**Prediction:**

2. \(G_t = \frac{\partial g(u_t, \mu_{t-1})}{\partial x_{t-1}} = \begin{pmatrix} \frac{\partial x'}{\partial \mu_{t-1,x}} & \frac{\partial x'}{\partial \mu_{t-1,y}} & \frac{\partial x'}{\partial \theta'} \\ \frac{\partial y'}{\partial \mu_{t-1,x}} & \frac{\partial y'}{\partial \mu_{t-1,y}} & \frac{\partial y'}{\partial \theta'} \\ \frac{\partial \theta'}{\partial \mu_{t-1,x}} & \frac{\partial \theta'}{\partial \mu_{t-1,y}} & \frac{\partial \theta'}{\partial \theta'} \end{pmatrix}\) Jacobian of \(g\) w.r.t location

3. \(B_t = \frac{\partial g(u_t, \mu_{t-1})}{\partial u_t} = \begin{pmatrix} \frac{\partial x'}{\partial v_t} & \frac{\partial x'}{\partial \omega_t} \\ \frac{\partial y'}{\partial v_t} & \frac{\partial y'}{\partial \omega_t} \\ \frac{\partial \theta'}{\partial v_t} & \frac{\partial \theta'}{\partial \omega_t} \end{pmatrix}\) Jacobian of \(g\) w.r.t control

4. \(Q_t = \begin{pmatrix} (\alpha_1 |v_t| + \alpha_2 |\omega_t|)^2 & 0 \\ 0 & (\alpha_3 |v_t| + \alpha_4 |\omega_t|)^2 \end{pmatrix}\) Motion noise

5. \(\bar{\mu}_t = g(u_t, \mu_{t-1})\) Predicted mean

6. \(\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + B_t Q_t B_t^T\) Predicted covariance
1. **EKF_localization** \( (\mu_{t-1}, \Sigma_{t-1}, u_t, z_t, m) \):

   **Correction:**

   2. \[ \hat{z}_t = \left( \sqrt{(m_x - \mu_{t,x})^2 + (m_y - \mu_{t,y})^2} \right) / \left( \tan^{-1} \left( \frac{m_y - \mu_{t,y}}{m_x - \mu_{t,x}} \right) - \mu_{t,\theta} \right) \]  

   Predicted measurement mean

   3. \[ H_t = \frac{\partial h(\mu_t, m)}{\partial x_t} = \begin{pmatrix} \frac{\partial r_t}{\partial \mu_{t,x}} & \frac{\partial r_t}{\partial \mu_{t,y}} & \frac{\partial r_t}{\partial \mu_{t,\theta}} \\ \frac{\partial q_t}{\partial \mu_{t,x}} & \frac{\partial q_t}{\partial \mu_{t,y}} & \frac{\partial q_t}{\partial \mu_{t,\theta}} \end{pmatrix} \]  

   Jacobian of \( h \) w.r.t location

   4. \[ R_t = \begin{pmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_r^2 \end{pmatrix} \]

   5. \[ S_t = H_t \Sigma_t H_t^T + R_t \]  

   Innovation covariance

   6. \[ K_t = \Sigma_t H_t^T S_t^{-1} \]  

   Kalman gain

   7. \[ \mu_t = \mu_t + K_t (z_t - \hat{z}_t) \]  

   Updated mean

   8. \[ \Sigma_t = (I - K_t H_t) \Sigma_t \]  

   Updated covariance
EKF Prediction Step
EKF Observation Prediction Step
EKF Correction Step
Estimation Sequence (1)
Estimation Sequence (2)
Comparison to GroundTruth
EKF Localization Example

- [Arras et al. 98]:
  - Laser range-finder and vision
  - High precision (<1cm accuracy)

Courtesy of K. Arras
EKF Localization Example

• Line and point landmarks
EKF Localization Example

- Line and point landmarks
EKF Localization Example

- Expo.02: Swiss National Exhibition 2002
- Pavilion "Robotics"
- 11 fully autonomous robots
- tour guides, entertainer, photographer
- 12 hours per day
- 7 days per week
- 5 months
- 3,316 km travel distance
- almost 700,000 visitors
- 400 visitors per hour

Localization method: Line-Based EKF
EKF Localization Example

“Robotics”

Expo.02 Switzerland

May 15th - October 20th, 2002

Autonomous Systems Lab
EPFL

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE
Global EKF Localization

Interpretation tree

$S_{h_2} = \{\{l_1, g_3\}, \{l_2, g_7\}, \{l_3, g_2\}\}$
Global EKF Localization

Env. Dynamics

\[ S_h = \{ \{l_1, g_4\}, \{l_2, g_8\}, \{l_3, *\}\} \]
Global EKF Localization

Geometric constraints we can exploit

Location independent constraints

*Unary constraint:* intrinsic property of feature
e.g. type, color, size

*Binary constraint:* relative measure between features
e.g. relative position, angle

Location dependent constraints

*Rigidity constraint:* "is the feature where I expect it given my position?"

*Visibility constraint:* "is the feature visible from my position?"

*Extension constraint:* "do the features overlap at my position?"

All decisions on a significance level $\alpha$
Global EKF Localization

Interpretation Tree
[Grimson 1987], [Drumheller 1987], [Castellanos 1996], [Lim 2000]

Algorithm

- backtracking
- depth-first
- recursive
- uses geometric constraints
- worst-case exponential complexity

```
function generate_hypotheses(h, L, G)
    H ← {}
    if L = {} then
        H ← H ∪ {h}
    else
        l ← select_observation(L)
        for g ∈ G do
            p ← {l, g}
            if satisfy_unary_constraints(p) then
                if location_available(h) then
                    accept ← satisfy_location_dependent_cnstr(L_h, p)
                    if accept then
                        H' ← H
                        S_h ← S_h ∪ {p}
                        L_h ← estimate_robot_location(S_h)
                    end
                else
                    accept ← true
                    for p_h ∈ S_h while accept
                        accept ← satisfy_binary_constraints(p_h, p)
                    end
                    if accept then
                        H' ← H
                        S_h ← S_h ∪ {p}
                        L_h ← estimate_robot_location(S_h)
                        if location_available(h') then
                            for p_h ∈ S_h while accept
                                accept ← satisfy_location_dependent_cnstr(L_h, p)
                            end
                        end
                    end
                end
            end
        end
        if accept then
            generate_hypotheses(H', L\{l}, G)
        end
        end
    generate_hypotheses(h, L\{l}, G)
end

return H
end
```
Global EKF Localization

\[ \alpha = 0.95, \quad p = 2 \]
Global EKF Localization

\[ \alpha = 0.95, \quad p = 3 \]
Global EKF Localization

\[ \alpha = 0.95 , \quad p = 4 \]
Global EKF Localization

\[ \alpha = 0.95, \quad p = 5 \]

\[ t_{\text{exe}}: 633 \text{ ms} \quad \text{(PowerPC at 300 MHz)} \]
Global EKF Localization

At Expo.02

05.07.02, 17.23 h

$\alpha = 0.999$

[Arras et al. 03]
Global EKF Localization

At Expo.02

05.07.02, 17.23 h

\[ \alpha = 0.999 \]

\[ t_{\text{exe}} = 105 \text{ ms} \]

[Arras et al. 03]
Global EKF Localization

At Expo.02

05.07.02, 17.32 h

$\alpha = 0.999$

[Arras et al. 03]
Global EKF Localization

At Expo.02

\( \alpha = 0.999 \)

\( t_{\text{exe}} = 446 \text{ ms} \)

[Arras et al. 03]
EKF Localization Summary

• **EKF localization** implements **pose tracking**
• Very **efficient** and **accurate** (positioning error down to subcentimeter)
• Filter divergence can cause lost situations from which the EKF **cannot recover**
• Industrial applications

• **Global EKF localization** can be achieved using interpretation tree-based data association
• Worst-case complexity is **exponential**
• **Fast** in practice for **small** maps