Particle filter, Motivation Discrete filter: Discretize the continuous state space; High memory complexity; Fixed resolution (does not adapt to the belief). Particle filters are a way to efficiently represent non-Gaussian distribution. Basic principle: Set of state hypotheses ("particles") Survival-of-the-fittest. Particle Filter: Sensor Information, 图 1. **Mapping** Problem Statement Formally, mapping involves, given the sensor data, d={x_1 , z_1 , x_2 , z_2 ,⋯, x_n , z_n } to calculate the most likely map m_*=arg max P(m | d). Occupancy GridMap , Occupancy of individual cells (m[xy]) is independent, 图 2. Updating Occupancy Map, 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 $Bel(m_t^{[xy]})=\eta\, p(z_t\mid m_t^{[xy]})Bel(m_{t-1}^{[xy]})$. Occupancy Probability 图 3. Reflection Maps {暂无}. Measurement Model 图 11. Summary Occupancy grid maps are a popular approach to represent the environment of a mobile robot given known poses. In this approach each cell is considered independently from all others. It stores the posterior probability that the corresponding area in the environment is occupied. Occupancy grid maps can be learned efficiently using a probabilistic approach. Reflection maps are an alternative representation. They store in each cell the probability that a beam is reflected by this cell. We provided a sensor model for computing the likelihood of measurements and showed that the counting procedure underlying reflection maps yield the optimal map. **EKF**, Localization Problem Statement, 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 图 4. Landmark Extraction 图 5. Measurement Prediction 图 6. Data Association (Matching) 图 7. Update 图 8. EKF Summary O(k^2.376 + n^2). **Planning**, Dynamic Window Approach, **Collision avoidance**: determine collision- free trajectories using geometric operations; Here: robot moves on circular arcs; Motion commands (v,ω); Which (v,ω) are admissible and reachable? 图 9. 5D planning Main Steps of the Algorithm, 1. Update (static) grid map based on sensory input. 2. Use A * to find a trajectory in the <x,y>-space using the updated grid map. 3. Determine a restricted 5d-configuration space based on step 2. 4. Find a trajectory by planning in the restricted <x,y,θ,v,ω>-space. **SLAM**, 图 10. SLAM Loop Closure • By revisiting already mapped areas, uncertainties in robot and landmark estimates can be reduced • This can be exploited to "optimally" explore an environment for the sake of better (e.g. more accurate) maps • Exploration: the problem of where to acquire new information (e.g. depth-first vs. breadth first). SLAM Summary • Convergence proof for linear case! • Can diverge if nonlinearities are large (and the reality is nonlinear...) • However, has been applied successfully in large-scale environments • Approximations reduce the computational complexity

1° motion model

$P(x_{t+1}\mid x_t, u_t)$

$\begin{bmatrix}x\\y\end{bmatrix}_{t+1}=\begin{bmatrix}1&\\&1\end{bmatrix}\begin{bmatrix}x\\y\end{bmatrix}_t+\begin{bmatrix}\Delta x\\\Delta y\end{bmatrix}+w$

$x_{t+1}=Ax+u+w$

$x_{t+1}\sim N(\bar\mu_{t+1},\bar\Sigma_{t+1})$

$\begin{cases}\bar\mu_{t+1}=\hat\mu_t+u\\ \bar\Sigma_{t+1}=A\hat\Sigma_{t+1}A^T+\Sigma_u+R\end{cases}$

2° measurement

$P(z_{t+1}\mid x_{t+1}, m)$

$\begin{bmatrix}d_1\\d_2\\d_3\\d_4\end{bmatrix}=\begin{bmatrix}1&0\\0&1\\-1&0\\0&-1\end{bmatrix}\begin{bmatrix}x\\y\end{bmatrix}_{t+1}+\begin{bmatrix}-1&0&0\\0&-1&0\\1&0&1\\0&1&1\end{bmatrix}\begin{bmatrix}x_0\\y_0\\a\end{bmatrix}$

~~$x_{t+1}\sim N$~~

$z_{t+1}=Cx_{t+1}+Gm+v$

$\hat x_{t+1}=(C^TQ^{-1}C)^{-1}C^TQ^{-1}(z_{t+1}-Gm)$

① $P(\hat x_{t+1}\mid z_{t+1},m)\sim N(\hat\mu_{t+1},\hat\Sigma_{t+1})$

$\begin{cases}\hat\mu_{t+1}=\underbrace{(C^TQ^{-1}C)^{-1}C^TQ^{-1}}_{H}(z-Gm)\\ \hat\Sigma_{t+1}=HQH^T\end{cases}$

$=(C^TQ^{-1}C)^{-1}C^TQ^{-1}C(C^TQ^{-1}C)^{-1}$

$=(C^TQ^{-1}C)^{-1}+H'\Sigma_mH'^T$

$H'=(C^TQ^{-1}C)^{-1}C^TQ^{-1}G$

② $P(\hat m_{t+1}\mid z_{t+1},x_{t+1})\sim N(m_{t+1},\Sigma_m)$

$\begin{cases}m_{t+1}=(G^TQ^{-1}G)^{-1}G^TQ^{-1}(z_{t+1}-C\mu_{t+1})\\ \Sigma_m=(G^TQ^{-1}G)^{-1}+L'\hat\Sigma_{t+1}L'^T\end{cases}$

$L'=(G^TQ^{-1}G)^{-1}G^TQ^{-1}C$

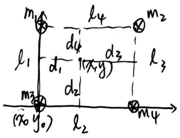
3° correction

$\begin{cases}\Sigma_{t+1}^{-1}=\bar\Sigma_{t+1}^{-1}+\hat\Sigma_{t+1}^{-1}\\ \Sigma_{t+1}^{-1}\mu_{t+1}=\bar\Sigma_{t+1}^{-1}\bar\mu_{t+1}+\hat\Sigma_{t+1}^{-1}\hat\mu_{t+1}\end{cases}$

① Localization $P(x\mid z,m,u)$

② mapping $P(m\mid z,x)$

③ SLAM $P(m,x\mid z,u)=P(x\mid z,m,u)P(m\mid z,x)$



$\begin{cases}d_1=x-x_0\\d_2=y-y_0\\d_3=l_2-x_0+x\\d_4=l_1-y_0+y\end{cases}$

$l_1=l_2=l_3=l_4=a$

① $P(z\mid x,m)=\left(\frac{1}{\sqrt{2\pi}\sigma}\right)^4e^{-\frac{1}{2\sigma^2}\bar A}$

$\bar A=[(\hat d-d_1)^2+(\hat d_2-d_2)^2+(\hat d_3-d_3)^2+(\hat d_4-d_4)^2]$

$=[(\hat d_1-x+x_0)^2+(\hat d_2-y+y_0)^2+(\hat d_3-a+x_0-x)^2+(\hat d_4-a+y_0-y)^2]$

② $P(x\mid m,z)=\left(\frac{1}{\sqrt{2\pi}\sigma}\right)^4e^{-\frac{1}{2\sigma^2}\bar B}$

a) $\bar B=[(\hat x-x)^2+(\hat y-y)^2]$

其中: $\begin{cases}x=d_1+x_0=d_3+x_0-l_2\\y=d_2+y_0=d_4+y_0-l_1\end{cases}$

b) $y=Ax+v$ ← $\sim N(0,Q)$

$x=(A^TA)^{-1}A^Ty$

$x=(A^TQ^{-1}A)^{-1}A^TQ^{-1}y$ (Q:噪声)

Algorithm EKF_localization_known_correspondences($\mu_{t-1},\Sigma_{t-1},u_t,z_t,c_t,m$):

$\theta=\mu_{t-1,\theta}$

$G_t=\begin{pmatrix}1&0&-\frac{v_t}{\omega_t}\cos\theta+\frac{v_t}{\omega_t}\cos(\theta+\omega_t\Delta t)\\0&1&-\frac{v_t}{\omega_t}\sin\theta+\frac{v_t}{\omega_t}\sin(\theta+\omega_t\Delta t)\\0&0&1\end{pmatrix}$

$V_t=\begin{pmatrix}\frac{-\sin\theta+\sin(\theta+\omega_t\Delta t)}{\omega_t}&\frac{v_t(\sin\theta-\sin(\theta+\omega_t\Delta t))}{\omega_t^2}+\frac{v_t\cos(\theta+\omega_t\Delta t)\Delta t}{\omega_t}\\\frac{\cos\theta-\cos(\theta+\omega_t\Delta t)}{\omega_t}&-\frac{v_t(\cos\theta-\cos(\theta+\omega_t\Delta t))}{\omega_t^2}+\frac{v_t\sin(\theta+\omega_t\Delta t)\Delta t}{\omega_t}\\0&\Delta t\end{pmatrix}$

$M_t=\begin{pmatrix}\alpha_1v_t^2+\alpha_2\omega_t^2&0\\0&\alpha_3v_t^2+\alpha_4\omega_t^2\end{pmatrix}$

$\bar\mu_t=\mu_{t-1}+\begin{pmatrix}-\frac{v_t}{\omega_t}\sin\theta+\frac{v_t}{\omega_t}\sin(\theta+\omega_t\Delta t)\\\frac{v_t}{\omega_t}\cos\theta-\frac{v_t}{\omega_t}\cos(\theta+\omega_t\Delta t)\\\omega_t\Delta t\end{pmatrix}$

$\bar\Sigma_t=G_t\,\Sigma_{t-1}\,G_t^T+V_t\,M_t\,V_t^T$

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

for all observed features $z_t^i=(r_t^i\ \phi_t^i\ s_t^i)^T$ do

$j=c_t^i$

$q=(m_{j,x}-\bar\mu_{t,x})^2+(m_{j,y}-\bar\mu_{t,y})^2$

$\hat z_t^i=\begin{pmatrix}\sqrt q\\\operatorname{atan2}(m_{j,y}-\bar\mu_{t,y},m_{j,x}-\bar\mu_{t,x})-\bar\mu_{t,\theta}\\m_{j,s}\end{pmatrix}$

$H_t^i=\begin{pmatrix}-\frac{m_{j,x}-\bar\mu_{t,x}}{\sqrt q}&-\frac{m_{j,y}-\bar\mu_{t,y}}{\sqrt q}&0\\\frac{m_{j,y}-\bar\mu_{t,y}}{q}&-\frac{m_{j,x}-\bar\mu_{t,x}}{q}&-1\\0&0&0\end{pmatrix}$

$S_t^i=H_t^i\,\bar\Sigma_t\,[H_t^i]^T+Q_t$

$K_t^i=\bar\Sigma_t\,[H_t^i]^T[S_t^i]^{-1}$

$\bar\mu_t=\bar\mu_t+K_t^i(z_t^i-\hat z_t^i)$

$\bar\Sigma_t=(I-K_t^i\,H_t^i)\,\bar\Sigma_t$

endfor

$\mu_t=\bar\mu_t$

$\Sigma_t=\bar\Sigma_t$

$p_{z_t}=\prod_i\det(2\pi S_t^i)^{-\frac12}\exp\{-\frac12(z_t^i-\hat z_t^i)^T[S_t^i]^{-1}(z_t^i-\hat z_t^i)\}$

return μ_t,Σ_t,p_{z_t}

System: $x_{t+1}=Ax_t+w$ $x_t\sim G(\mu_t,\Sigma_t)$

$z_{t+1}=Cx_{t+1}+v$ $w\sim G(0,R)$ $v\sim G(0,Q)$

$x_{t+1}\sim G(\mu_{t+1},\Sigma_{t+1})$

Kalman Filter

(1) prediction: $\bar x_{t+1}\sim G(\bar\mu_{t+1},\bar\Sigma_{t+1})$ $\begin{cases}\bar\mu_{t+1}=A\mu_t\\\bar\Sigma_{t+1}=A\Sigma_tA^T+R\end{cases}$

(2) observation: $\hat x_{t+1}=(C^TQ^{-1}C)^{-1}C^TQ^{-1}z_{t+1}$ $VAR\{z_{t+1}\mid x_{t+1}\}=Q$

$\hat\mu_{t+1}=E\{\hat x_{t+1}\mid z_{t+1}\}=(C^TQ^{-1}C)^{-1}C^TQ^{-1}z_{t+1}$

$\hat\Sigma_{t+1}=VAR\{\hat x_{t+1}\mid z_{t+1}\}=(C^TQ^{-1}C)^{-1}$

$\hat x_{t+1}\sim G(\hat\mu_{t+1},\hat\Sigma_{t+1})$

(3) correction: $x_{t+1}\sim G(\mu_{t+1},\Sigma_{t+1})$ $\begin{cases}\Sigma_{t+1}=(\bar\Sigma_{t+1}^{-1}+\hat\Sigma_{t+1}^{-1})^{-1}\\\Sigma_{t+1}^{-1}\mu_{t+1}=\bar\Sigma_{t+1}^{-1}\bar\mu_{t+1}+\hat\Sigma_{t+1}^{-1}\hat\mu_{t+1}\end{cases}$

From the motion in the plane

to the 2N+3 dimensional space

Extended Kalman filter($\mu_{t-1},\Sigma_{t-1},u_t,z_t$):

$\bar\mu_t=g(u_t,\mu_{t-1})$

$\bar\Sigma_t=G_t\,\Sigma_{t-1}\,G_t^T+R_t$

$K_t=\bar\Sigma_t\,H_t^T(H_t\,\bar\Sigma_t\,H_t^T+Q_t)^{-1}$

$\mu_t=\bar\mu_t+K_t(z_t-h(\bar\mu_t))$

$\Sigma_t=(I-K_t\,H_t)\,\bar\Sigma_t$

return μ_t,Σ_t

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

$F_x=\begin{pmatrix}1&0&0&0\cdots0\\0&1&0&0\cdots0\\0&0&1&0\cdots0\end{pmatrix}$

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

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

$\bar\Sigma_t=G_t\,\Sigma_{t-1}\,G_t^T+\underbrace{F_x^TR_t^xF_x}_{R_t}$

$\bar\Sigma_t=G_t\Sigma_{t-1}G_t^T+R_t=\begin{pmatrix}G_t^x&0\\0&I\end{pmatrix}\begin{pmatrix}\Sigma_{xx}&\Sigma_{xm}\\\Sigma_{mx}&\Sigma_{mm}\end{pmatrix}\begin{pmatrix}(G_t^x)^T&0\\0&I\end{pmatrix}+R_t=\begin{pmatrix}G_t^x\Sigma_{xx}(G_t^x)^T&G_t^x\Sigma_{xm}\\(G_t^x\Sigma_{xm})^T&\Sigma_{mm}\end{pmatrix}+R_t$

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

Importance Sampling

Target distribution f : $p(x \mid z_1, z_2, ..., z_n) = \frac{\prod_k p(z_k \mid x) \quad p(x)}{p(z_1, z_2, ..., z_n)}$

Sampling distribution g : $p(x \mid z_l) = \frac{p(z_l \mid x) p(x)}{p(z_l)}$

Importanceweights w : $\frac{f}{g} = \frac{p(x \mid z_1, z_2, ..., z_n)}{p(x \mid z_l)} = \frac{p(z_l) \prod_{k \neq l} p(z_k \mid x)}{p(z_1, z_2, ..., z_n)}$

1. Algorithm **particle_filter**(S_{t-1}, u_{t-1} z_t):
2. $S_t = \emptyset$, $\eta = 0$
3. **For** $i = 1 \ldots n$ — *Generate new samples*
4. Sample index $j(i)$ from the discrete distribution given by w_{t-1}
5. Sample x_t^i from $p(x_t \mid x_{t-1}, u_{t-1})$ using $x_{t-1}^{j(i)}$ and u_{t-1}
6. $w_t^i = p(z_t \mid x_t^i)$ — *Compute importance weight*
7. $\eta = \eta + w_t^i$ — *Update normalization factor*
8. $S_t = S_t \cup \{< x_t^i, w_t^i >\}$ — *Insert*
9. **For** $i = 1 \ldots n$
10. $w_t^i = w_t^i / \eta$ — *Normalize weights*

图 1

$$Bel(x) \quad \leftarrow \quad \alpha\, p(z \mid x)\, Bel^-(x)$$

$$w \quad \leftarrow \quad \frac{\alpha\, p(z \mid x)\, Bel^-(x)}{Bel^-(x)} \quad = \quad \alpha\, p(z \mid x)$$

图 2

$$Bel(m_t) = P(m_t \mid x_1, z_2 \ldots, x_{t-1}, z_t) = \prod_{x,y} Bel(m_t^{[xy]})$$

Algorithm Kalman_filter($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$):

$\bar{\mu}_t = A_t \mu_{t-1} + B_t u_t$

$\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t$

$K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$

$\mu_t = \bar{\mu}_t + K_t(z_t - C_t \bar{\mu}_t)$

$\Sigma_t = (I - K_t C_t) \bar{\Sigma}_t$

return μ_t, Σ_t

Updating Occupancy Map

Update the map cells using the inverse sensor mod

$$Bel(m_t^{[xy]}) = 1 - \left(1 + \frac{P(m_t^{[xy]} \mid z_t, u_{t-1})}{1 - P(m_t^{[xy]} \mid z_t, u_{t-1})} \cdot \frac{1 - P(m_t^{[xy]})}{P(m_t^{[xy]})} \cdot \frac{Bel(m_{t-1}^{[xy]})}{1 - Bel(m_{t-1}^{[xy]})}\right)^{-1}$$

Or use the log-odds representation

$$\bar{B}(m_t^{[xy]}) = \log odds(m_t^{[xy]} \mid z_t, u_{t-1}) - \log odds(m_t^{[xy]}) + \bar{B}(m_{t-1}^{[xy]})$$

$$\bar{B}(m_t^{[xy]}) := \log odds(m_t^{[xy]}) \qquad odds(x) := \left(\frac{P(x)}{1 - P(x)}\right)$$

图 3

$$P(m_{d,\theta}(x(k)) \mid y(k), x(k)) = P(m_{d,\theta}(x(k))) + \begin{cases} -s(y(k), \theta) & d < y(k) - d_1 \\ -s(y(k), \theta) + \frac{s(y(k),\theta)}{d_1}(d - y(k) + d_1) & d < y(k) + d_1 \\ s(y(k), \theta) & d < y(k) + d_2 \\ s(y(k), \theta) - \frac{s(y(k),\theta)}{d_3 - d_2}(d - y(k) - d_2) & d < y(k) + d_3 \\ 0 & \text{otherwise.} \end{cases}$$

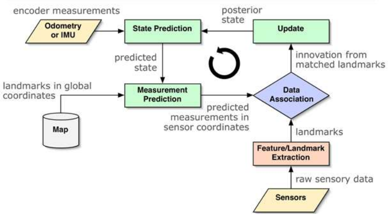
Example

- Out of 1000 beams only 60% are reflected from a cell and 40% intercept it without ending in it.
- Accordingly, the reflection probability will be 0.6.
- Suppose $p(occ \mid z) = 0.55$ when a beam ends in a cell and $p(occ \mid z) = 0.45$ when a cell is intercepted by a beam that does not end in it.
- Accordingly, after n measurements we will have

$$\left(\frac{0.55}{0.45}\right)^{n*0.6} * \left(\frac{0.45}{0.55}\right)^{n*0.4} = \left(\frac{11}{9}\right)^{n*0.6} * \left(\frac{11}{9}\right)^{-n*0.4} = \left(\frac{11}{9}\right)^{n*0.2}$$

- Whereas the reflection map yields a value of 0.6, the occupancy grid value converges to 1.

Landmark-based Localization



State Prediction (Odometry)

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

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

Control u_k : wheel displacements s_l , s_r

$\mathbf{u}_k = (s_l \; s_r)^T$

$U_k = \begin{bmatrix} \sigma_l^2 & 0 \\ 0 & \sigma_r^2 \end{bmatrix}$

图 4

Error model: linear growth

$\sigma_l = k_l |s_l|$

$\sigma_r = k_r |s_r|$

Nonlinear process model f :

$$\mathbf{x}_k = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \end{bmatrix} + \begin{bmatrix} \frac{b}{2}\frac{s_l + s_r}{s_r - s_l}\left(-\sin\theta_{k-1} + \sin(\theta_{k-1} + \frac{s_r - s_l}{b})\right) \\ \frac{b}{2}\frac{s_l + s_r}{s_r - s_l}\left(\cos\theta_{k-1} - \cos(\theta_{k-1} + \frac{s_r - s_l}{b})\right) \\ \frac{s_r - s_l}{b} \end{bmatrix}$$

图 5

Hessian line model

$$x\cos(\alpha) + y\sin(\alpha) - r = 0$$

$$\mathbf{z}_k = \begin{bmatrix} \alpha \\ r \end{bmatrix} \qquad R_k = \begin{bmatrix} \sigma_\alpha^2 & \sigma_{\alpha r} \\ \sigma_{r\alpha} & \sigma_r^2 \end{bmatrix}$$

图 6

Given the predicted state (robot pose),

predicts the location $\hat{\mathbf{z}}_k$ and location uncertainty $H\,\hat{C}_k\,H^T$ of expected observations in sensor coordinates

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

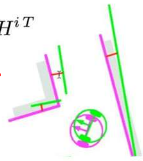
Associates predicted measurements $\hat{\mathbf{z}}_k^i$ with observations $\mathbf{z}_k^j$

$$\nu_k^{ij} = \mathbf{z}_k^j - \hat{\mathbf{z}}_k^i$$

$$S_k^{ij} = R_k^j + H^i\,\hat{C}_k\,H^{i\,T}$$

Innovation ν_k^{ij} and innovation covariance S_k^{ij}

图 7



- Kalman gain

$$K_k = \hat{C}_k\,H^T S_k^{-1}$$

图 8

- State update (robot pose)

$$\mathbf{x}_k = \hat{\mathbf{x}}_k + K_k\,\nu_k$$

- State covariance update

$$C_k = (I - K_k\,H)\,\hat{C}_k$$

$$p(m_i \mid z_{1:t}, x_{1:t}) \overset{\text{Bayes rule}}{=} \frac{p(z_t \mid m_i, z_{1:t-1}, x_{1:t})\, p(m_i \mid z_{1:t-1}, x_{1:t})}{p(z_t \mid z_{1:t-1}, x_{1:t})}$$

$$\overset{\text{Markov}}{=} \frac{p(z_t \mid m_i, x_t)\, p(m_i \mid z_{1:t-1}, x_{1:t-1})}{p(z_t \mid z_{1:t-1}, x_{1:t})}$$

$$\overset{\text{Bayes rule}}{=} \frac{p(m_i \mid z_t, x_t)\, p(z_t \mid x_t)\, p(m_i \mid z_{1:t-1}, x_{1:t-1})}{p(m_i \mid x_t)\, p(z_t \mid z_{1:t-1}, x_{1:t})}$$

$$\overset{\text{Markov}}{=} \frac{p(m_i \mid z_t, x_t)\, p(z_t \mid x_t)\, p(m_i \mid z_{1:t-1}, x_{1:t-1})}{p(m_i)\, p(z_t \mid z_{1:t-1}, x_{1:t})}$$

- Recursive rule

$$\frac{p(m_i \mid z_{1:t}, x_{1:t})}{1 - p(m_i \mid z_{1:t}, x_{1:t})} = \underbrace{\frac{p(m_i \mid z_t, x_t)}{1 - p(m_i \mid z_t, x_t)}}_{\text{uses } z_t} \underbrace{\frac{p(m_i \mid z_{1:t-1}, x_{1:t-1})}{1 - p(m_i \mid z_{1:t-1}, x_{1:t-1})}}_{\text{recursive term}} \underbrace{\frac{1 - p(m_i)}{p(m_i)}}_{\text{prior}}$$

- Often written as

$$Bel(m_t^i) = \left[1 + \frac{1 - p(m_t^i \mid z_t, u_{t-1})}{p(m_t^i \mid z_t, u_{t-1})} \frac{p(m_t^i)}{1 - p(m_t^i)} \frac{1 - Bel(m_{t-1}^i)}{Bel(m_{t-1}^i)}\right]^{-1}$$

Do exactly the same for the opposite event:

$$p(\neg m_i \mid z_{1:t}, x_{1:t}) \overset{\text{the same}}{=} \frac{p(\neg m_i \mid z_t, x_t)\, p(z_t \mid x_t)\, p(\neg m_i \mid z_{1:t-1}, x_{1:t-1})}{p(\neg m_i)\, p(z_t \mid z_{1:t-1}, x_{1:t})}$$

Dynamic Window Approach

Speeds are admissible if

$$V_a = \{(v, \omega) \;\mid\; v \leq \sqrt{2\text{dist}(v,\omega) a_{trans}} \;\wedge\; \omega \leq \sqrt{2\text{dist}(v,\omega) a_{rot}}\}$$

$$V_d = \{(v, \omega) \;\mid\; v \in [v - a_{trans}t, v + a_{trans}t] \;\wedge\; \omega \in [\omega - a_{rot}t, \omega + a_{rot}t]\}$$

V_s = all possible speeds of the robot.

V_a = obstacle free area.

V_d = speeds reachable within a certain time frame based on possible accelerations.

$$Space = V_s \cap V_a \cap V_d$$

图 9

Navigation Function: [Brock & Khatib, 99]

$$NF = \alpha \cdot vel + \beta \cdot nf + \gamma \cdot \Delta nf + \delta \cdot goal$$

Maximizes velocity. — Considers cost to reach the goal. — Follows grid based path computed byA*. — Goal Nearness

A*: Minimize the Estimated Path Costs

- $g(n)$ = actual cost from the initial state to n.
- $h(n)$ = estimated cost from n to the next goal.
- $f(n) = g(n) + h(n)$, the estimated cost of the cheapest solution through n.
- Let $h^*(n)$ be the actual cost of the optimal path from n to the next goal.
- h is admissible if the following holds for all n :
  - $h(n) \leq h^*(n)$
- We require that for A*, h is admissible (the straight-line distance is admissible in the Euclidean Space).

The SLAM Problem

Given:

- The robot's controls

$$\mathbf{U}_{0:k} = \{\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_k\}$$

- Relative observations

$$\mathbf{Z}_{0:k} = \{\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_k\}$$

Wanted:

- Map of features

$$\mathbf{m} = \{\mathbf{m}_1, \mathbf{m}_2, \cdots, \mathbf{m}_n\}$$

- Path of the robot

$$\mathbf{X}_{0:k} = \{\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_k\}$$

SLAM

- Full SLAM:

$$p(x_{0:t}, m \mid z_{1:t}, u_{1:t})$$

图 10

Estimates entire path and map!

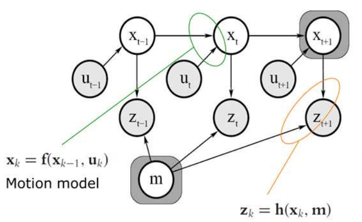
- Online SLAM:

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

Integrations (marginalization) typically done recursively, one at a time

Estimates most recent pose and map!

Graphical Model: Models



$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k)$

Motion model

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

Observation model

(1) Bayes Rule

$p(x \mid z) = \frac{p(z \mid x) p(x)}{p(z)}$

(2) Bayes Filter

$p(x_{t+1} \mid z_{t+1}) = \frac{p(z_{t+1} \mid x_{t+1}) \int p(x_{t+1} \mid x_t) p(x_t) dx_t}{p(z_{t+1})}$

(3) Bayes Filter with Map & Odometry

$p(x_{t+1} \mid z_{t+1}, m, u) = \frac{p(z_{t+1} \mid x_{t+1}, m) \int p(x_{t+1} \mid x_t, u) p(x_t) dx_t}{p(z_{t+1})}$

(4) prediction $p(\bar{x}_{t+1} \mid u) = \int p(x_{t+1} \mid x_t, u) p(x_t) dx_t$

observation $p(\bar{x}_{t+1} \mid z_{t+1}, m) \longleftarrow p(z_{t+1} \mid x_{t+1}, m)$

correction $p(x_{t+1} \mid z_{t+1}, u, m) = p(\bar{x}_{t+1} \mid u)\, p(\bar{x}_{t+1} \mid z_{t+1}, m)$

State Prediction

Odometry:

$\hat{\mathbf{x}}_R = f(\mathbf{x}_R, \mathbf{u})$

$\hat{C}_R = F_x C_R F_x^T + F_u U F_u^T$

Robot-landmark cross-covariance prediction:

$\hat{C}_{RM_i} = F_x C_{RM_i}$

(skipping time index k)

Measurement Prediction

Global-to-local frame transform h

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

Observation

(x,y)-point landmarks

$\mathbf{z}_k = \begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ \vdots \end{bmatrix} = \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \\ \vdots \end{bmatrix}$

$R_k = \begin{bmatrix} R_1 & 0 & \\ 0 & R_2 & \\ & & \ddots \end{bmatrix}$

Data Association

Associates predicted measurements $\hat{\mathbf{z}}_k^i$ with observation $\mathbf{z}_k^j$

$\nu_k^{ij} = \mathbf{z}_k^j - \hat{\mathbf{z}}_k^i$

$S_k^{ij} = R_k^j + H^i \hat{C}_k H^{i\,T}$

Filter Update

The usual Kalman filter expressions

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

$\mathbf{x}_k = \hat{\mathbf{x}}_k + K_k \nu_k$

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

Integrating New Landmarks

State augmented by

$\mathbf{m}_{n+1} = g(\mathbf{x}_R, \mathbf{z}_j)$

$C_{M_{n+1}} = G_R C_R G_R^T + G_z R_j G_z^T$

Cross-covariances:

$C_{M_{n+1}M_i} = G_R C_{RM_i}$

$C_{M_{n+1}R} = G_R C_R$

EKF SLAM: State Representation

- **Localization**

3x1 pose vector 3x3 cov. matrix

$$\mathbf{x}_k = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} \qquad C_k = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} & \sigma_{x\theta} \\ \sigma_{yx} & \sigma_y^2 & \sigma_{y\theta} \\ \sigma_{\theta x} & \sigma_{\theta y} & \sigma_\theta^2 \end{bmatrix}_k$$

- **SLAM**

Landmarks are **simply added** to the state. **Growing** state vector and covariance matrix!

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{x}_R \\ \mathbf{m}_1 \\ \mathbf{m}_2 \\ \vdots \\ \mathbf{m}_n \end{bmatrix}_k \qquad C_k = \begin{bmatrix} C_R & C_{RM_1} & C_{RM_2} & \cdots & C_{RM_n} \\ C_{M_1R} & C_{M_1} & C_{M_1M_2} & \cdots & C_{M_1M_n} \\ C_{M_2R} & C_{M_2M_1} & C_{M_2} & \cdots & C_{M_2M_n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ C_{M_nR} & C_{M_nM_1} & C_{M_nM_2} & \cdots & C_{M_n} \end{bmatrix}_k$$

First-Order Propagation

$$\begin{aligned} \mu_x &= E(x) \\ &= E(Au + b) \\ &= AE(u) + b \\ &= A\mu_u + b \end{aligned}$$

$$\begin{aligned} \Sigma_x &= E((x - E(x))(x - E(x))^T) \\ &= E((Au + b - AE(u) - b)(Au + b - AE(u) - b)^T) \\ &= E((A(u - E(u)))(A(u - E(u)))^T) \\ &= E((A(u - E(u)))((u - E(u))^T A^T)) \\ &= AE((u - E(u))(u - E(u))^T)A^T \\ &= A\Sigma_u A^T \end{aligned}$$

Error Propagation Law

$$C_Y = F_X C_X F_X^T$$

$$p(z_{t,n} \mid x_t, m) = \begin{cases} \prod_{k=0}^{z_{t,n}-1} (1 - m_{f(x_t,n,k)}) & \text{if } \varsigma_{t,n} = 1 \\ m_{f(x_t,n,z_{t,n})} \prod_{k=0}^{z_{t,n}-1} (1 - m_{f(x_t,n,k)}) & \text{if } \varsigma_{t,n} = 0 \end{cases}$$

图 11

1. pose at time t: x_t
2. beam n of scan t: $z_{t,n}$
3. maximum range reading: $\varsigma_{t,n} = 1$
4. beam reflected by an object: $\varsigma_{t,n} = 0$