

B3M33MKR Particle Filter

Miroslav Kulich, Gaël Écorchard, Karel Košnar

Intelligent and Mobile Robotics Group
Czech Technical University in Prague

Winter semester 2017/2018



Assignment

- Implement Particle filter
 - Motion model
 - Sensor model
 - Integrate
- Input data from a recorded dataset
- Program template given



Particle filter - the algorithm

Particle_filter(S_{t-1}, u_{t-1}, z_t)

$S_t = \emptyset, \eta = 0$

for $i = 1, \dots, n$ **do**

Sample index j_i from the discrete distribution given by w_{t-1}

Sample $x_t^i \sim p(x_t | x_{t-1}, u_{t-1})$ using $x_{t-1}^{j_i}$ and u_{t-1}

$w_t^i = p(z_t | x_t^i)$

$\eta = \eta + w_t^i$

$S_t = S_t \cup \{ \langle x_t^i, w_t^i \rangle \}$

end for

for $i = 1, \dots, n$ **do**

$w_t^i = \frac{w_t^i}{\eta}$

end for

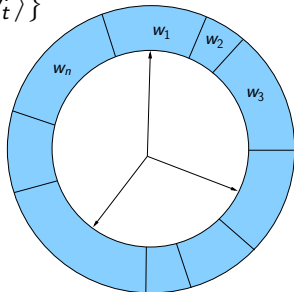
Generate new particles

Compute importance weight

Update normalization factor

Insert a particle

Normalize weights



Odometry-based model for sampling

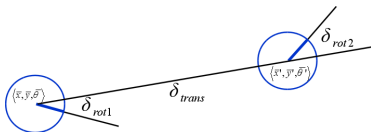
$$u = \langle \delta_{rot1}, \delta_{rot2}, \delta_{trans} \rangle, x = \langle x, y, \phi \rangle \Rightarrow x' = \langle x', y', \phi' \rangle$$

Random control

$$\begin{aligned}\hat{\delta}_{rot1} &= \delta_{rot1} + \text{sample}(\alpha_1 |\delta_{rot1}| + \alpha_2 |\delta_{trans}|) \\ \hat{\delta}_{trans} &= \delta_{trans} + \text{sample}(\alpha_3 |\delta_{trans}| + \alpha_4 (|\delta_{rot1}| + |\delta_{rot2}|)) \\ \hat{\delta}_{rot2} &= \delta_{rot2} + \text{sample}(\alpha_1 |\delta_{rot2}| + \alpha_2 |\delta_{trans}|)\end{aligned}$$

Position determination

$$\begin{aligned}x' &= x + \hat{\delta}_{trans} \cos(\phi + \hat{\delta}_{rot1}) \\ y' &= y + \hat{\delta}_{trans} \sin(\phi + \hat{\delta}_{rot1}) \\ \phi' &= \phi + \hat{\delta}_{rot1} + \hat{\delta}_{rot2} \\ &\text{return } \langle x', y', \phi' \rangle\end{aligned}$$



Naive (not good!) approach



Correct approach

α_2 is large



Correct approach

α_3 is large



Correct approach

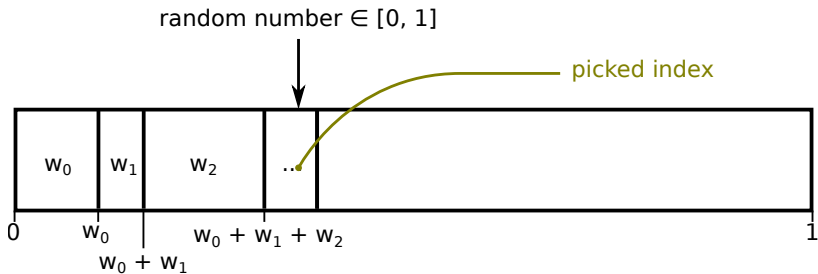
Good parameters



Correct approach

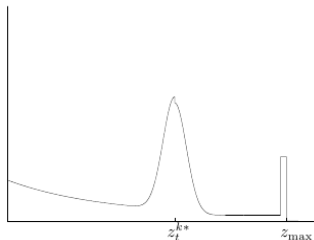


Roulette wheel



Sensor model

- The aim is to determine $p(z|m, x)$.
- Scan is composed from k measurements (beams):
 $z = \{z_1, z_2, \dots, z_k\}$
- Individual measurements are independent given the robot position (strong assumption): $P(z|x, m) = \prod_{k=1} P(z_k|x, m)$

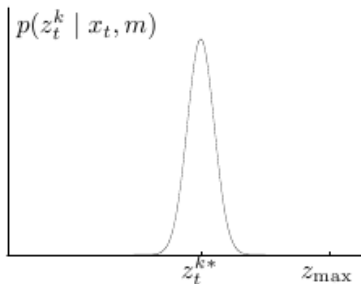


$$p(z|x, m) = \begin{pmatrix} \alpha_{hit} \\ \alpha_{short} \\ \alpha_{rand} \\ \alpha_{max} \end{pmatrix}^T \begin{pmatrix} p_{hit}(z|x, m) \\ p_{short}(z|x, m) \\ p_{rand}(z|x, m) \\ p_{max}(z|x, m) \end{pmatrix}$$



Beam-based model - components

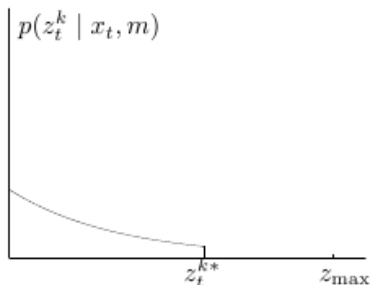
Measurement noise



$$p_{hit}(z|x, m) = \begin{cases} \eta N(z, z^*, \sigma_{hit}^2) & \text{if } 0 \leq z \leq z_{\max} \\ 0 & \text{otherwise} \end{cases}$$

Normal distribution

Unexpected obstacles



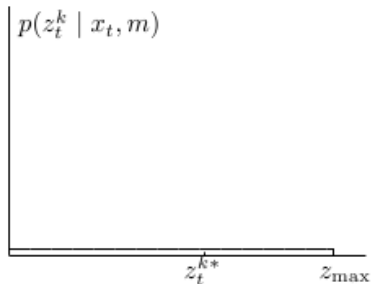
$$p_{short}(z|x, m) = \begin{cases} \eta \lambda_{short} e^{-\lambda_{short} z} & \text{if } 0 \leq z \leq z^* \\ 0 & \text{otherwise} \end{cases}$$

Exponential distribution



Beam-based model - components

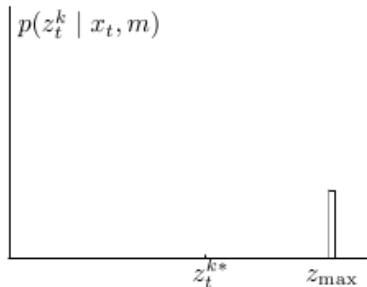
Random measurement



$$p_{\text{rand}}(z|x, m) = \begin{cases} \frac{1}{z_{\max}} & \text{if } 0 \leq z \leq z_{\max} \\ 0 & \text{otherwise} \end{cases}$$

Uniform distribution

Max range



$$p_{\text{max}}(z|x, m) = \begin{cases} 1 & \text{if } z = z_{\max} \\ 0 & \text{otherwise} \end{cases}$$

Discrete distribution :-)



Implementation - Motion Model

- optional implementation of the motion model separately
- archive `motion-model.tar.gz` on CourseWare
- implementation in `moveParticle`
- robot motion hard-recorded
- shows arrows or dots

