FreiCAR
Block 2: Localization
Lecture 2: Monte Carlo Localization Details

Lukas Luft
Monte Carlo Localization
Monte Carlo Localization Recap

- Belief representation
  \[ x_t = \left\{ (x_t^{[j]}, w_t^{[j]}) \right\}_{j=1,\ldots,J} \]

- Belief update
  1. Prediction step: shift particle poses
     \[ x_t^{[j]} \sim p(x_t^{[j]} | x_{t-1}^{[j]}, u_t) \]
  2. Correction step: update particle weights
     \[ w_t^{[j]} \propto p(z_t | x_t^{[j]}, m) \]
  3. Resampling: focus particles to relevant regions
     \[ x_t^{[j]} \sim \omega_t^{[j]} \]
Monte Carlo Localization Algorithm

\textbf{Particle\_filter}(x_{t-1}, u_t, z_t):

1: \quad \tilde{X}_t = X_t = \emptyset

2: \quad \text{for } j = 1 \text{ to } J \text{ do}

3: \quad \text{sample } x_t^{[j]} \sim p(x_t^{[j]} \mid u_t, x_{t-1}^{[j]})

4: \quad w_t^{[j]} = p(z_t \mid x_t^{[j]})

5: \quad \tilde{X}_t = \tilde{X}_t + \langle x_t^{[j]}, w_t^{[j]} \rangle

6: \quad \text{endfor}

7: \quad \text{for } j = 1 \text{ to } J \text{ do}

8: \quad \text{draw } i \in 1, \ldots, J \text{ with probability } \propto w_t^{[i]}

9: \quad \text{add } x_t^{[i]} \text{ to } X_t

10: \quad \text{endfor}

11: \quad \text{return } X_t
1. Prediction Step

- General formula Bayes filter
  \[
  \overline{bel}(x_t) = \int p(x_t | u_t, x_{t-1}) \, bel(x_{t-1}) \, dx_{t-1}
  \]

- Particle filter
  \[
  \approx \int p(x_t | u_t, x_{t-1}) \sum_{j=1}^{J} w_t^{[j]} \, \delta_{x_{t-1}^{[j]}}(x_{t-1}) \, dx_{t-1}
  \]
  \[
  = \sum_{j=1}^{J} p(x_t | u_t, x_{t-1}^{[j]}) \, w_t^{[j]}
  \]

- Sample from motion model
  \[
  x_t^{[j]} \sim p(x_t^{[j]} | x_{t-1}^{[j]}, u_t)
  \]
Simple Constant Velocity Model

- Perceived relative velocities $u_t = (v_x, v_y, v_\theta)^T$
- Observed velocities are corrupted by independent Gaussian noise $\epsilon = (\epsilon_x, \epsilon_y, \epsilon_\theta)$
- Constant velocity model:

$$x_t[j] = x_{t-1} + map T_{vehicle} (u_t - \epsilon) \Delta t$$

- Sample from $p(\epsilon)$ and apply motion model
Rejection Sampling

- Sample $x$ from a uniform distribution from $[-b,b]$.
- Sample $y$ from $[0, \text{max } f]$.
- if and only if $f(x) > y$ keep the sample $x$. 

![Diagram showing rejection sampling process with samples and probability/weight distribution.](image-url)
Further reading

- More on transformations:
  Lecture 02 of the Robot Mapping Course
  http://ais.informatik.uni-freiburg.de/teaching/ws19/mapping/index_en.php

- More motion models:
  Lecture 06 of the Introduction to Mobile Robotics Course
  http://ais.informatik.uni-freiburg.de/teaching/ss20/robotics/index_en.php
1. Correction Step

- General formula Bayes filter

\[ \text{bel}(x_t) = \eta \ p(z_t \mid x_t, m) \ \overline{\text{bel}}(x_t) \]

- Particle filter: assign weight proportional to measurement likelihood

\[ w_{t}^{[j]} \propto p(z_t \mid x_t^{[j]}, m) \]
Perceiving Relative Positions of Landmarks

- Observations:

- Map:
Perceiving Relative Positions of Landmarks

- Perceived relative position: \( z_t = (\delta_x, \delta_y)^T \)
- Observation of landmark at known location: \( l = (l_x, l_y)^T \)
  is corrupted by Gaussian noise: \( \nu = (\nu_x, \nu_y) \)
- Measurement model:
  \[
  z_t = \text{vehicle} T_{map}(l) + \nu
  \]
- Measurement likelihood is Gaussian distribution:
  \[
  p(z_t \mid x_t^{[j]}, m) = \mathcal{N} \left( z_t; \text{vehicle} T_{map}(l), \Sigma_{\nu} \right)
  \]
Perceiving Landmarks with Range-Bearing Sensors

- Range-bearing \( z_t^i = (r_t, \phi_t)^T \)
- Vehicle pose \( (x, y, \theta)^T \)
- Observation of landmark

\[
\begin{pmatrix}
  r_t \\
  \phi_t
\end{pmatrix}
= \begin{pmatrix}
  \sqrt{(l_x - x)^2 + (l_y - y)^2} \\
  \text{atan2}(l_y - y, l_x - x) - \theta
\end{pmatrix} + \nu
\]
Unknown Data Association

- If we do not know which out of $J$ landmarks, we observe

\[
p(z|x, m) = \sum_{j=1}^{J} p(z|j, x, m)p(j|x, m)
\]

\[
= \sum_{j=1}^{J} p(z|j, x, m_j)p(j|x, m)
\]
3. Resampling

- Draw sample $i$ with probability $w_i^{[i]}$. Repeat $J$ times.
- Informally: “Replace unlikely samples by more likely ones”
- Survival of the fittest
- “Trick” to avoid that many samples cover unlikely states
- Needed as we have a limited number of samples
3. Resampling

- **Given**: Set of J weighted samples.

- **Wanted**: Random sample, where the probability of drawing $x_i$ is given by $w_i$.

- Typically done $n$ times with replacement to generate new sample set $S'$.
3. Resampling

- Roulette wheel
- Binary search
- \(O(J \log J)\)

- Stochastic universal sampling
- Low variance
- \(O(J)\)
Low Variance Resampling

\textbf{Low\_variance\_resampling}(\mathcal{X}_t, \mathcal{W}_t):

1: \quad \tilde{\mathcal{X}}_t = \emptyset \\
2: \quad r = \text{rand}(0; J^{-1}) \\
3: \quad c = w_t^{[1]} \\
4: \quad i = 1 \\
5: \quad \text{for } j = 1 \text{ to } J \text{ do} \\
6: \quad \quad U = r + (j - 1)J^{-1} \\
7: \quad \quad \text{while } U > c \text{ do} \\
8: \quad \quad \quad i = i + 1 \\
9: \quad \quad \quad c = c + w_t^{[i]} \\
10: \quad \quad \text{endwhile} \\
11: \quad \quad \text{add } x_t^{[i]} \text{ to } \tilde{\mathcal{X}}_t \\
12: \quad \text{endfor} \\
13: \quad \text{return } \tilde{\mathcal{X}}_t
Thank you for your attention!