Motivation

- Recall: 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
Sample-based Localization (sonar)
Mathematical Description

- Set of weighted samples

\[ S = \left\{ \langle s[i], w[i] \rangle \mid i = 1, \ldots, N \right\} \]

State hypothesis

Importance weight

- The samples represent the posterior

\[ p(x) = \sum_{i=1}^{N} w_i \cdot \delta_{s[i]}(x) \]
Particle sets can be used to approximate functions. The more particles fall into an interval, the higher the probability of that interval. How to draw samples from a function/distribution?
Rejection Sampling

- Let us assume that $f(x) < 1$ for all $x$
- Sample $x$ from a uniform distribution
- Sample $c$ from $[0,1]$
- if $f(x) > c$ keep the sample
  otherwise reject the sample
Importance Sampling Principle

- We can even use a different distribution \( g \) to generate samples from \( f \)
- By introducing an importance weight \( w \), we can account for the “differences between \( g \) and \( f \)”
- \( w = \frac{f}{g} \)
- \( f \) is often called target
- \( g \) is often called proposal
- Pre-condition: \( f(x) > 0 \Rightarrow g(x) > 0 \)
Importance Sampling with Resampling: Landmark Detection Example
Distributions
Distributions

Wanted: samples distributed according to \( p(x|z_1, z_2, z_3) \)
This is Easy!

We can draw samples from $p(x|z_i)$ by adding noise to the detection parameters.
Importance Sampling

Target distribution $f : p(x \mid z_1, z_2, \ldots, z_n) = \frac{\prod_{k} p(z_k \mid x) p(x)}{p(z_1, z_2, \ldots, z_n)}$

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

Importance weights $w : \frac{f}{g} = \frac{p(x \mid z_1, z_2, \ldots, z_n)}{p(x \mid z_l)} = \frac{p(z_l) \prod_{k \neq l} p(z_k \mid x)}{p(z_1, z_2, \ldots, z_n)}$
Importance Sampling with Resampling

Weighted samples

After resampling
Particle Filters
Sensor Information: Importance Sampling

\[ \text{Bel}(x) \leftarrow \alpha p(z \mid x) \text{Bel}^{-}(x) \]

\[ w \leftarrow \frac{\alpha p(z \mid x) \text{Bel}^{-}(x)}{\text{Bel}^{-}(x)} = \alpha p(z \mid x) \]
Robot Motion

\[ Bel^{-}(x) \leftarrow \int p(x \mid u, x') Bel(x') \, dx' \]
Sensor Information: Importance Sampling

\[
\begin{align*}
\text{Bel}(x) & \leftarrow \alpha p(z \mid x) \text{Bel}^{-}(x) \\
{w} & \leftarrow \frac{\alpha p(z \mid x) \text{Bel}^{-}(x)}{\text{Bel}^{-}(x)} = \alpha p(z \mid x)
\end{align*}
\]
Robot Motion

\[ Bel^{-}(x) \leftarrow \int p(x \mid u, x') Bel(x') \, dx' \]
Particle Filter Algorithm

- Sample the next generation for particles using the proposal distribution

- Compute the importance weights:
  \[ \text{weight} = \text{target distribution} / \text{proposal distribution} \]

- Resampling: “Replace unlikely samples by more likely ones”

[Derivation of the MCL equations on the blackboard]
Particle Filter Algorithm

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

\[ Bel(x_t) = \eta p(z_t | x_t) \int p(x_t | x_{t-1}, u_{t-1}) Bel(x_{t-1}) \, dx_{t-1} \]

- draw \( x_{t-1}^i \) from \( Bel(x_{t-1}) \)
- draw \( x_t^i \) from \( p(x_t | x_{t-1}^i, u_{t-1}) \)
- Importance factor for \( x_t^i \):

\[
\begin{align*}
    w_t^i &= \frac{\text{target distribution}}{\text{proposal distribution}} \\
          &= \frac{\eta p(z_t | x_t) p(x_t | x_{t-1}, u_{t-1}) Bel(x_{t-1})}{p(x_t | x_{t-1}, u_{t-1}) Bel(x_{t-1})} \\
          &\propto p(z_t | x_t)
\end{align*}
\]
Resampling

- **Given**: Set $S$ of 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 $S'$.
Resampling

- Roulette wheel
- Binary search, $n \log n$

- Stochastic universal sampling
- Systematic resampling
- Linear time complexity
- Easy to implement, low variance
Resampling Algorithm

1. Algorithm `systematic_resampling`(S, n):
   2. \( S' = \emptyset, c_1 = w^1 \)
   3. For \( i = 2 \ldots n \)
      Generate cdf
   4. \( c_i = c_{i-1} + w^i \)
   5. \( u_1 \sim U [0, n^{-1}], i = 1 \)
      Initialize threshold
   6. For \( j = 1 \ldots n \)
      Draw samples …
   7. While ( \( u_j > c_i \) )
      Skip until next threshold reached
   8. \( i = i + 1 \)
   9. \( S' = S' \cup \{ < x^i, n^{-1} > \} \)
      Insert
   10. \( u_{j+1} = u_j + n^{-1} \)
      Increment threshold
11. Return \( S' \)

Also called stochastic universal sampling
Mobile Robot Localization

- Each particle is a potential pose of the robot

- Proposal distribution is the motion model of the robot (prediction step)

- The observation model is used to compute the importance weight (correction step)

[For details, see PDF file on the lecture web page]
Motion Model Reminder
Proximity Sensor Model Reminder

Laser sensor

Sonar sensor
Sample-based Localization (sonar)
Initial Distribution
After Incorporating Ten Ultrasound Scans
After Incorporating 65 Ultrasound Scans
Estimated Path
Localization for AIBO robots
Using Ceiling Maps for Localization

[Dellaert et al. 99]
Vision-based Localization

\[ P(z|x) \]

\[ h(x) \]
Under a Light

Measurement $z$: $P(z|x)$:
Next to a Light

Measurement $z$: $P(z|x)$:
Elsewhere

Measurement $z$: \hspace{1cm} P(z|x):
Global Localization Using Vision
Limitations

- The approach described so far is able to
  - track the pose of a mobile robot and to
  - globally localize the robot.

- How can we deal with localization errors (i.e., the kidnapped robot problem)?
Approaches

- Randomly insert samples (the robot can be teleported at any point in time).
- Insert random samples proportional to the average likelihood of the particles (the robot has been teleported with higher probability when the likelihood of its observations drops).
Summary – Particle Filters

- Particle filters are an implementation of recursive Bayesian filtering.
- They represent the posterior by a set of weighted samples.
- They can model non-Gaussian distributions.
- Proposal to draw new samples.
- Weight to account for the differences between the proposal and the target.
- Monte Carlo filter, Survival of the fittest, Condensation, Bootstrap filter.
Summary – PF Localization

- In the context of localization, the particles are propagated according to the motion model.
- They are then weighted according to the likelihood of the observations.
- In a re-sampling step, new particles are drawn with a probability proportional to the likelihood of the observation.