Monte Carlo Localization

Ximing Yu

March 24, 2009

ARIZONA

Localization Problem

Estimate the pose (including location and orientation) of a robot:

- Unmanned Aerial Vehicle: Longitude, Latitude, Altitude, Roll, Pitch, Yaw.
- Unmanned Ground Vehicle: Longitude, Latitude, Yaw.
- General Mobile Robot: x, y, θ.

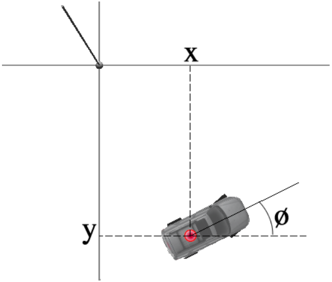

Figure: A demonstration of the pose of a general mobile robot

A Taxonomy of Localization Problems

- Position Tracking:
  - With known initial robot pose.
  - Recursively update state estimation.
- Global Localization:
  - Without initial robot pose.
  - The robot finds landmarks sequentially to deduce its location based on known map.
- Kidnapped Robot Problem:
  - Well-localized robot teleported.
  - Can test robot's ability to recover from catastrophic localization failure.

Generative Model for State Estimation

Denote:

- State (pose) of a robot at time t as x_t.
- Measurement (observation) at time t as o_t.
- Control input (odometry reading) between $[t-1, t]$ as a_{t-1}.

the state estimation of a robot can be modeled as a hidden Markov model:

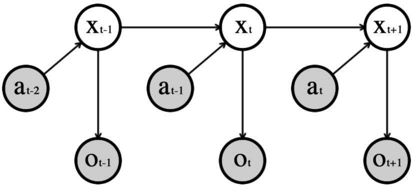

Figure: The hidden Markov model for state estimation

Bayes Filter

Bayes filter estimates the posterior probability distribution over the state space conditioned on all measurements and control inputs. The posterior is usually called *belief*:

$$Bel(x_t) = p(x_t | o_t, a_{t-1}, o_{t-1}, a_{t-2}, \ldots, o_0)$$

The belief without the latest measurement o_t is termed as:

$$\overline{Bel}(x_t) = p(x_t | a_{t-1}, o_{t-1}, a_{t-2}, \ldots, o_0)$$

Bayes Filter

The relationship between $Bel(x_t)$ and $\overline{Bel}(x_t)$ is:

$$\overline{Bel}(x_t) = \int p(x_t|x_{t-1}, a_{t-1}, \ldots, o_0) Bel(x_{t-1}) \mathrm{d}x_{t-1}$$

$$Bel(x_t) = \eta p(o_t|x_t, a_{t-1}, \ldots, o_0) \overline{Bel}(x_t)$$

$$\eta = p(o_t|a_{t-1}, \ldots, o_0)^{-1}$$

Bayes Filter

Applying the Markov assumption, the relationship is simplified to:

$$\overline{Bel}(x_t) = \int p(x_t|x_{t-1}, a_{t-1})Bel(x_{t-1})\mathrm{d}x_{t-1}$$

$$Bel(x_t) = \eta p(o_t|x_t)\overline{Bel}(x_t)$$

With the given initial state, we can get state at each time t by using the above formula recursively.

Motion Model and Perceptual Model

Notice the two conditional densities in previous slide:

- $p(x_t|x_{t-1}, a_{t-1})$: The motion model or kinematic model.
- $p(o_t|x_t)$: The perceptual model or environment measurement model.

Typically these two models are regarded as *stationary*.

Particle Filter

The idea of particle filter is to represent the belief $Bel(x)$ by a set of M weighted samples distributed according to $Bel(x)$:

$$Bel(x) \approx \left\{ x^{(i)}, w^{(i)} \right\}_{i=1,\dots,M}$$

$w^{(i)}$ are called *importance factors* and $\sum_{i=1}^{M} w^{(i)} = 1$.
For initial samples of $Bel(x_0)$:

- Position tracking problem: samples drawn from a narrow Gaussian centered on the known initial pose.
- Global localization problem: samples drawn from a uniform distribution over the robot's possible universe, with $w^{(i)} = \frac{1}{M}$, $i = 1, \dots, M$.

Particle Filter

Particle filter works with three steps recursively:

1. Sample $x_{t-1}^{(i)} \sim Bel(x_{t-1})$ from the weighted sample set representing $Bel(x_{t-1})$.
2. Sample $x_t^{(i)} \sim p(x_t|x_{t-1}^{(i)}, a_{t-1})$ from the motion model.
3. Sample $w^{(i)} \sim p(o_t|x_t^{(i)})$ from the perceptual model.

Comments on Particle Filter

With respect to the three recursive steps:

- The second step is sampling from *a proposal distribution*

$$q_t = p(x_t|x_{t-1}^{(i)}, a_{t-1})Bel(x_{t-1}^{(i)}),$$

while the target distribution is

$$\begin{aligned} Bel(x_t^{(i)}) &= \eta p(o_t|x_t^{(i)})\overline{Bel}(x_t^{(i)}) \\ &= \eta p(o_t|x_t^{(i)})p(x_t^{(i)}|x_{t-1}^{(i)}, a_{t-1})Bel(x_{t-1}^{(i)}) \end{aligned}$$

(The absence of integral sign is due to some trick on $Bel(x_t)$ and $Bel(x_{0,\ldots,t})$)

Comments on Particle Filter

Continue. . .

- The third step realizes an *importance sampling* by drawing the *importance factor* $w^{(i)} = p(o_t|x_t^{(i)})$, which is proportional to the quotient of the target distribution and the proposal distribution $(\eta p(o_t|x_t^{(i)}))$.

Comments on Particle Filter

Continue...

- The first step is the real "trick" of particle filter.
  - Before step 1: the weighted sample set $\{x^{(i)}, w^{(i)}\}_{i=1,\ldots,M}$ are actually distributed to $\overline{Bel}(x)$, with some compensating weights assigned to each particle.
  - After step 1: particles are distributed to $Bel(x)$, each with equal weight factor of $\frac{1}{M}$.

With the first step, particle filter is actually using *Sampling Importance Resampling* (SIR).

Comments on Particle Filter

Continue. . .

- The disadvantage of resampling is the diversity of particles will reduce, leaving several copies of those particles with higher weight factor. However, without the resampling step it is possible that after several recursion, only one particle with weight factor 1 is left. Then the state estimation fails.

The Algorithm of MCL

Algorithm MCL(X,a,o)

$X' = \emptyset$

for $i = 0$ to M do

  generate random x from X according to $w_1, \ldots, w_M$

  generate random $x' \sim p(x'|a, x)$

  $w' = p(o|x')$

  add $< x', w' >$ to X'

endfor

normalize the importance factors w' in X'

return X'

Limitation of MCL

MCL with accurate sensors may perform *worse* than MCL with inaccurate sensors.

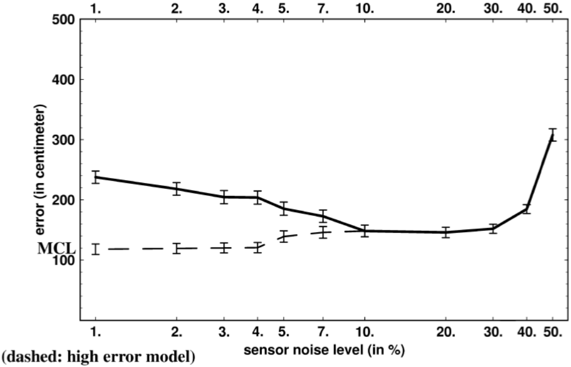

Figure: Error of MCL

Limitation of MCL

Reason

- The limitation of MCL stems from the difference between *target distribution* and *proposal distribution*.
- The difference is accounted by the perceptual model $p(o_t|x_t)$, which produce the *importance factor*.
- For accurate sensors, the density of perceptual model is narrow with sharp peak, resulting in great difference between *target distribution* and *proposal distribution*.
- If the peaks of motion model and perceptual model do not match, some "accurate" particles are neglected.

Dual-MCL

- The idea of Dual-MCL is to take density of *perceptual model* as proposal distribution and find corresponding importance factor with regard to the difference of target distribution and proposal distribution.
- Generally the proposal distribution takes the form:

$$\bar{q}_t = \frac{p(o_t|x_t)}{\pi(o_t)} \quad \text{with } \pi(o_t) = \int p(o_t|x_t)\mathrm{d}x_t$$

Dual-MCL

Three approaches to find proposal distribution and importance factor for Dual-MCL:

1. $\bar{q}_{1,t} = \frac{p(o_t|x_t)}{\pi(o_t)} \times Bel(x_{t-1})$
   $w^{(i)} = \eta p(x_t^{(i)}|x_{t-1}^{(i)}, a_{t-1})\pi(o_t) \propto p(x_t^{(i)}|x_{t-1}^{(i)}, a_{t-1})$
2. $\bar{q}_{2,t} = \bar{q}_t$
   $w^{(i)} = \eta\pi(o_t)p(x_t^{(i)}|a_{t-1}, d_{0\ldots t-1}) = \eta\pi(o_t)\overline{Bel}(x_t^{(i)})$
3. $\bar{q}_{3,t} = \frac{p(o_t|x_t)}{\pi(o_t)} \times \frac{p(x_t|x_{t-1},a_{t-1})}{\pi(x_t|a_{t-1})}$
   $w^{(i)} = \eta\pi(o_t)\pi(x_t^{(i)}|a_{t-1})Bel(x_{t-1}^{(i)})$

Performance of Dual-MCL

The Dual-MCL alone does not produce localization accurately, due to its vulnerability to perceptual noise. However, its accuracy is *monotonic* in perceptual noise.

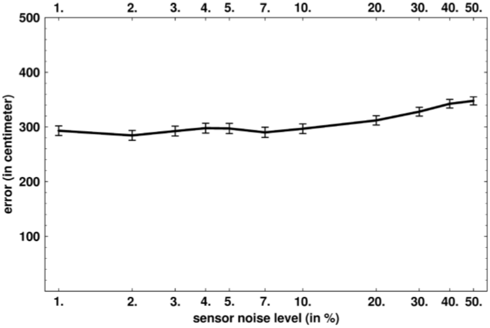

Figure: Error of Dual-MCL

Mixture-MCL Algorithm

By Mixture-MCL, each sample is generated with probability $1 - \phi$ using standard MCL and with probability ϕ using a Dual-MCL. The performance of Mixture-MCL is desirable when perceptual noise is low and its accuracy is *monotonic* in perceptual noise.

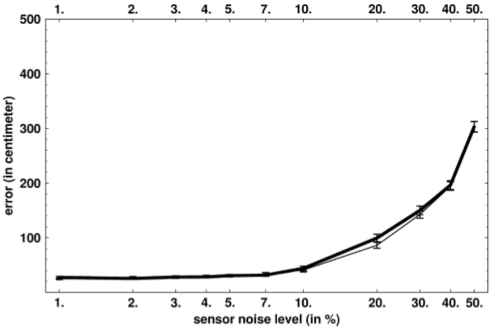

Figure: Error of Mixture-MCL

Thank you!