MONTE CARLO LOCALIZATION (MCL):
& INTRODUCTION TO SLAM

CS486
Introduction to AI
Spring 2006
University of Waterloo

Speaker: Martin Talbot

DEFINITION OF LOCALIZATION

Process where the robot finds its position in its environment using a global coordinate scheme (map).

2 PROBLEMS:
- GLOBAL LOCALIZATION
- POSITION TRACKING

EARLY WORK

* Initially, work was focused on tracking using Kalman Filters

* Then MARKOV LOCALIZATION came along and global localization could be addressed successfully.

KALMAN FILTERS BASIC IDEA

SINGLE POSITION HYPOTHESIS

The uncertainty in the robot's position is represented by an unimodal Gaussian distribution (bell shape).

KALMAN FILTERS

WHAT IF THE ROBOT GETS LOST?

MARKOV LOCALIZATION

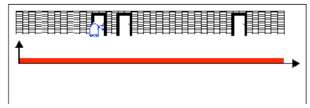
MULTIPLE POSITION HYPOTHESIS.

ML maintains a **probability distribution** over the entire space, multimodal Gaussian distribution.

SIMPLE EXAMPLE…………..

Let's assume the space of robot positions is **one-dimensional**, that is, the robot can only move horizontally…

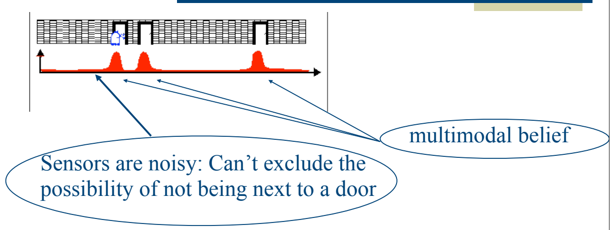
ROBOT PLACED SOMEWHERE



Agent ignores its location & did not sense its environment yet…notice the flat distribution (red bar.)

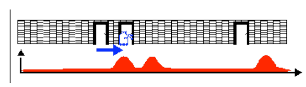
Markov localization represents this state of uncertainty by a *uniform distribution* over all positions.

ROBOT QUERIES ITS SENSORS



The robot queries its sensors and finds out that it is next to a door. **Markov localization modifies the belief by raising the probability for places next to doors**, and lowering it anywhere else.

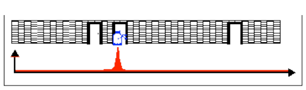
ROBOT MOVES A METER FORWARD



The robot moves a meter forward. Markov localization incorporates this information by **shifting the belief distribution** accordingly.

The noise in robot motion leads to a loss of information, the new belief is smoother (and less certain) than the previous one, variances are larger.

ROBOT SENSES A SECOND TIME



The robot senses a second time

This observation is **multiplied** into the current belief, which leads to the final belief.

At this point in time, most of the probability is centred around a single location. The robot is now quite certain about its position.

MATH BEHIND MARKOV LOCALIZATION

In the context of Robotics: Markov Localization = Bayes Filters

Next we derive the recursive update equation in Bayes filter

MATH BEHIND MARKOV LOCALIZATION

Bayes Filters address the problem of **estimating the robot's pose** in its environment, where the pose is represented by a **2-dimensional Cartesian** space with an **angular direction**.

Example: pos(x,y, θ)

MATH BEHIND MARKOV LOCALIZATION

Bayes filters assume that the environment is **Markov**, that is **past and future data are conditionally independent if one knows the current state**

MATH BEHIND MARKOV LOCALIZATION

TWO TYPES OF MODEL:

Perceptual data such as laser range measurements, sonar, camera is denoted by ***o*** (observed)

Odometer data which carry information about robot's motion is denoted by ***a*** (action)

MATH BEHIND MARKOV LOCALIZATION

$$Bel(x_t) = p(x_t | d_{0 \ldots t})$$

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

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

MATH BEHIND MARKOV LOCALIZATION

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

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

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

MATH BEHIND MARKOV LOCALIZATION

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

MARKOV ASSUMPTION

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

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

MATH BEHIND MARKOV LOCALIZATION

We are working to obtain a **recursive form**…we integrate out x_{t-1} at time t-1

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

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

Using the Theorem of Total Probability

MATH BEHIND MARKOV LOCALIZATION

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

$$p(x_t|x_{t-1},a_{t-1},\ldots,o_0) = p(x_t|x_{t-1},a_{t-1})$$

The **Markov assumption** also implies that given the knowledge of x_{t-1} and a_{t-1}, the state x_t is conditionally independent of past measurements up to time t-2

MATH BEHIND MARKOV LOCALIZATION

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

$$Bel(x_t) = \eta\ p(o_t|x_t) \int p(x_t|x_{t-1},a_{t-1}) Bel(x_{t-1}) dx_{t-1}$$

Using the definition of the belief *Bel*() we obtain a recursive estimation known as **Bayes Filters**.

The top equation is of an incremental form.

MATH BEHIND MARKOV LOCALIZATION

Since both our motion model (*a*) and our perceptual model (*o*) are typically **stationary** (models don't depend on specific time t) we can **simplify** the notation by using **p(x' | *x, a*)** and **p(*o'*|*x'*)**.

$p(x_t|x_{t-1},a_{t-1})$

$p(o_t|x_t)$

MATH BEHIND MARKOV LOCALIZATION

Bel(x') = η p(o'|x') ∫ p(x'|x, a) Bel(x) dx

MONTE CARLO LOCALIZATION

KEY IDEA:

Represent the belief *Bel*(*x*) by a set of discrete weighted samples.

MONTE CARLO LOCALIZATION

KEY IDEA:

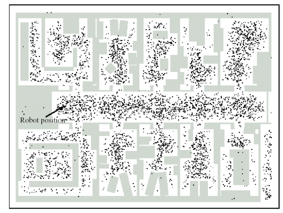
Bel(x) = {(l_1,w_1), (l_2, w_2), ….(l_m, w_m)}

Where each l_i, $1 \le i \le m$, represents a location (x,y,θ)

Where each $w_i \ge 0$ is called the **importance factor**

MONTE CARLO LOCALIZATION

In global localization, the initial belief is a set of locations drawn according a uniform distribution, each sample has **weight = 1/m**.



MCL: THE ALGORITHM

Algorithm MCL(X, a, o):
- $X' = \emptyset$ ← X' = {(l_1,w_1)', ….(l_m, w_m)'}
- *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* $\langle x', w' \rangle$ *to* X'
- *endfor*
- *normalize the importance factors* w' *in* X'
- *return* X'

The Recursive Update Is Realized in Three Steps

MCL: THE ALGORITHM

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* $\langle x', w' \rangle$ *to* X'
- *endfor*
- *normalize the importance factors* w' *in* X'
- *return* X'

Step 1: Using **importance sample** from the weighted sample set representing ~ Bel(x) pick a sample x_i: **x_i ~ Bel(x)**

MCL: THE ALGORITHM

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* $\langle x', w' \rangle$ *to* X'
- *endfor*
- *normalize the importance factors* w' *in* X'
- *return* X'

Step 2: Sample x_i' ~ p(x'| a, x_i). Since x_i and a together belong to a distribution, we pick x_i^t according to this distribution, the one that has the highest probability is more likely to be picked.

MCL: THE ALGORITHM

Step 2: With x_i' ~ p(x'| x_i, a) and x_i ~ Bel(x) we compute

qt := x_i' * x_i note that this is: **p(x'| x_i, a)** * **Bel(x)**

We **propose** this distribution

$$Bel(x_t) = \eta \; p(o_t|x_t) \int p(x_t|x_{t-1}, a_{t-1}) \, Bel(x_{t-1}) \, dx_{t-1}$$

MARKOV LOCALIZATION FORMULA!

MCL: THE ALGORITHM

The role of **qt** is to propose samples of the posterior distribution.
This is not equivalent to the desired posterior.

MCL: THE ALGORITHM

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 $\langle x', w' \rangle$ to X'
endfor
normalize the importance factors w' in X'
return X'

Step 3: The importance factor w is obtained as the quotient of the target distribution and the proposal distribution.

MCL: THE ALGORITHM

Notice the proportionality between the quotient and w_i (η is constant)

Target distribution

$$\frac{\eta\, p(o'|x_i')\, p(x'|x_i, a)\, Bel(x)}{p(x'|x_i, a)\, Bel(x)} = \eta\, \mathbf{p(o'|x_i')} \propto \mathbf{w_i}$$

Proposal distribution

MCL: THE ALGORITHM

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 $\langle x', w' \rangle$ to X'
endfor
normalize the importance factors w' in X'
return X'

After m iterations, normalize w'

ADAPTIVE SAMPLE SET SIZES

KEY IDEA:

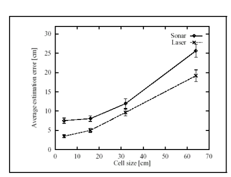
Use **divergence** of weight **before** and **after** sensing.

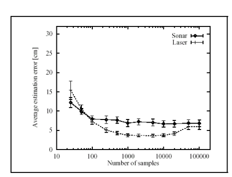
Sampling is stopped when the sum of weights (action and observation data) exceeds some **threshold.**

If action and observations are in tune together, each individual weight is large and the sample set remains small.

If the opposite occurs, individual weights are small and the sample set is large.

ADAPTIVE SAMPLE SET SIZES





Global localization: ML ⇒ 120 sec, MCL ⇒ 3sec.

Resolution 20 cm ~10 times more space than MCL with m=5000 samples

CONCLUSION

* Markov Localization method is a foundation for MCL

* MCL uses random weighted samples to decide which states it evaluates

    * "unlikely" states (low weight) are less probable to be evaluated

* MCL is a more efficient, more effective method especially when used with adaptive sample set size.

REFERENCES

D. Fox, W. Burgard, and S. Thrun. Markov localization for mobile robots in dynamic environments. *Journal of Artificial Intelligence Research*, 11:391–427, 1999.

S. Thrun, D. Fox, and W. Burgard. Monte carlo localization with mixture proposal distribution. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, Austin, TX, 2000. AAAI.

D. Fox, W. Burgard, F. Dellaert, and S. Thrun. Monte carlo localization: Efficient position estimation for mobile robots. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, Orlando, FL, 1999. AAAI.

D. Fox, W. Burgard, and S. Thrun. Active markov localization for mobile robots. *Robotics and Autonomous Systems*, 25(3-4):195–207, 1998.