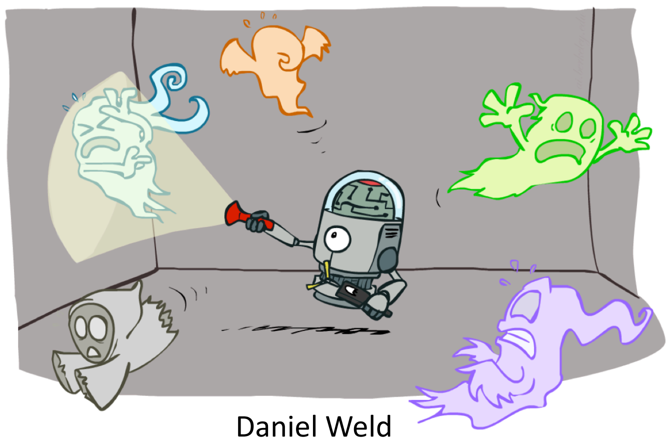
CSE 473: Artificial Intelligence

Hidden Markov Models

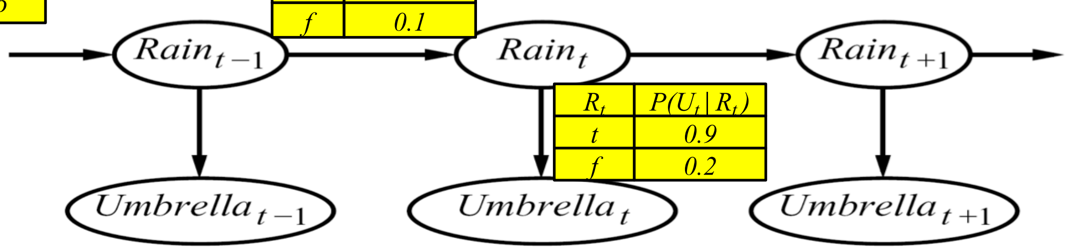


Daniel Weld

University of Washington

[Many of these slides were created by Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley. All CS188 materials are available at http://ai.berkeley.edu.]

Hidden Markov Model: Example



$P(R_1)$
0.6

R_{t-1}	$P(R_t \mid R_{t-1})$
t	0.7
f	0.1

R_t	$P(U_t \mid R_t)$
t	0.9
f	0.2

- An HMM is defined by:
  - Initial distribution: $P(X_1)$
  - Transitions: $P(X_t|X_{t-1})$
  - Emissions: $P(E|X)$

Filtering (aka Monitoring)

- Filtering, or monitoring, is the task of tracking the distribution B(X) (called "the belief state") over time

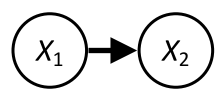
- We start with $B_0(X)$ in an initial setting, usually uniform

- We update $B_t(X)$ — *computing $B_{t+1}(X)$*
  1. As time passes, and — *using prob model of how ghosts move*
  2. As we get observations — *using prob model of how noisy sensors work*

Passage of Time

- Assume we have current belief P(X | evidence to date)

$$B(X_t) = P(X_t|e_{1:t})$$



- Then, after one time step passes:

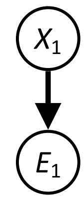
$$P(X_{t+1}|e_{1:t}) = \sum_{x_t} P(X_{t+1}, x_t|e_{1:t})$$
$$= \sum_{x_t} P(X_{t+1}|x_t, e_{1:t})P(x_t|e_{1:t})$$
$$= \sum_{x_t} P(X_{t+1}|x_t)P(x_t|e_{1:t})$$

- Or compactly:

$$B'(X_{t+1}) = \sum_{x_t} P(X'|x_t)B(x_t)$$

- Basic idea: beliefs get "pushed" through the transitions
  - With the "B" notation, we have to be careful about what time step t the belief is about, and what evidence it includes

Observation



- Assume we have current belief P(X | previous evidence):

$$B'(X_{t+1}) = P(X_{t+1}|e_{1:t})$$

- Then, after evidence comes in:

$$P(X_{t+1}|e_{1:t+1}) = P(X_{t+1}, e_{t+1}|e_{1:t})/P(e_{t+1}|e_{1:t})$$ *Defn cond prob*

$$= P(e_{t+1}|e_{1:t}, X_{t+1})P(X_{t+1}|e_{1:t}) \;/P(e_{t+1}|e_{1:t})$$ *Defn cond prob*

$$= P(e_{t+1}|X_{t+1})P(X_{t+1}|e_{1:t}) \;/P(e_{t+1}|e_{1:t})$$ *Independence*

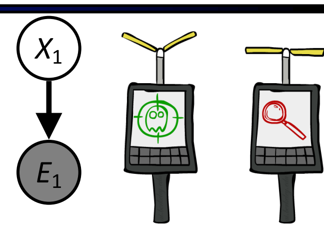
- Or, compactly:

$$B(X_{t+1}) = P(e_{t+1}|X_{t+1})B'(X_{t+1}) \;/P(e_{t+1}|e_{1:t})$$

- Basic idea: beliefs "reweighted" by likelihood of evidence
- Unlike passage of time, we have to normalize

Observation

Old version



- Assume we have current belief P(X | previous evidence):

$$B'(X_{t+1}) = P(X_{t+1}|e_{1:t})$$

- Then, after evidence comes in:

$$P(X_{t+1}|e_{1:t+1}) = P(X_{t+1}, e_{t+1}|e_{1:t})/P(e_{t+1}|e_{1:t})$$

$$\propto_{X_{t+1}} P(X_{t+1}, e_{t+1}|e_{1:t})$$

$$= P(e_{t+1}|e_{1:t}, X_{t+1})P(X_{t+1}|e_{1:t})$$

$$= P(e_{t+1}|X_{t+1})P(X_{t+1}|e_{1:t})$$

- Or, compactly:

$$B(X_{t+1}) \propto_{X_{t+1}} P(e_{t+1}|X_{t+1})B'(X_{t+1})$$

- Basic idea: beliefs "reweighted" by likelihood of evidence
- Unlike passage of time, we have to renormalize

Normalization to Account for Evidence

X	E	P
rain	U	0.4
rain	-	0.1
sun	U	0.2
sun	-	0.3

SELECT the joint probabilities matching the evidence →

X	E	P
rain	U	0.4
sun	U	0.2

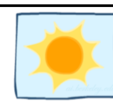
NORMALIZE the selection (make it sum to one) →

$P(W|T = c)$

X	P
rain	0.67
sun	0.33

Since could have seen other evidence, we normalize by dividing by the probability of the evidence we *did* see (in this case dividing by 0.5)…

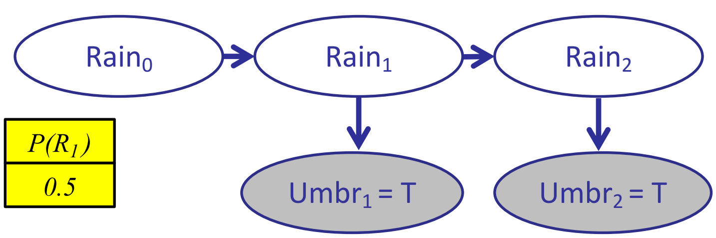
Example: Weather HMM




$B'(x_1{=}r) = P(x_1{=}r \mid x_0{=}r) * 0.5 + P(x_1{=}r \mid x_0{=}s) * 0.5$
$= 0.8*0.5 + 0.6*0.5$
$= 0.7$

$B(x_0{=}r) = 0.5$

$$B'(X_{t+1}) = \sum_{x_t} P(X'|x_t)B(x_t)$$



$P(R_1)$
0.5

R_{t-1}	$P(R_t \mid R_{t-1})$
t	0.8
f	0.6

R_t	$P(U_t \mid R_t)$
t	0.9
f	0.3

Example: Weather HMM

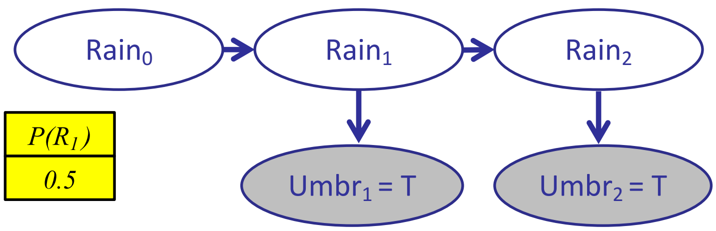
B'(x_1=r) = P(x_1=r | x_0=r) * 0.5 + P(x_1=r | x_0=s) * 0.5
= 0.8*0.5 + 0.6*0.5
= 0.7

B(x_1=r) $\propto$ 0.9 * 0.7 = 0.63
B(x_1=s) $\propto$ 0.3 * 0.3 = 0.09
Divide by 0.72 (=0.63+0.09) to normalize
B(x_1=r) =0.63/0.72 = 0.875

B(x_0=r) = 0.5

$$B'(X_{t+1}) = \sum_{x_t} P(X'|x_t)B(x_t)$$

$$B(X_{t+1}) \propto_{X_{t+1}} P(e_{t+1}|X_{t+1})B'(X_{t+1})$$



$P(R_1)$
0.5

R_{t-1}	$P(R_t \mid R_{t-1})$
t	0.8
f	0.6

R_t	$P(U_t \mid R_t)$
t	0.9
f	0.3

Example: Weather HMM

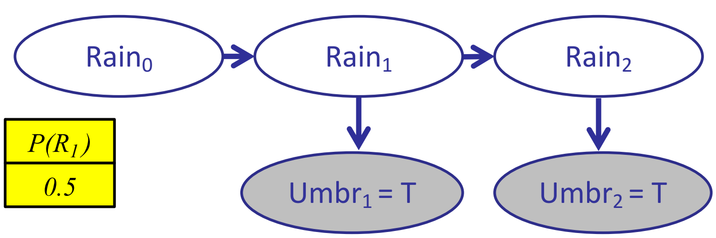
B'(x_2=r) = P(x_2=r | x_1=r)*0.875 + P(x_2=r | x_1=s)*0.125
= 0.8*0.875 + 0.6*0.125
= 0.775

B'(x_1=r) = 0.7

B(x_0=r) = 0.5

B(x_1=r) = 0.875

$$B'(X_{t+1}) = \sum_{x_t} P(X'|x_t)B(x_t)$$



$P(R_1)$
0.5

R_{t-1}	$P(R_t \mid R_{t-1})$
t	0.8
f	0.6

R_t	$P(U_t \mid R_t)$
t	0.9
f	0.3

Example: Weather HMM

$B'(x_2=r) = P(x_2=r \mid x_1=r)*0.875 + P(x_2=r \mid x_1=s)*0.125$
$= 0.8*0.875 + 0.6*0.125$
$= 0.775$

$B'(x_1=r) = 0.7$

$B(x_1=r) \propto 0.9 * 0.775 = 0.6975$
$B(x_1=s) \propto 0.3 * 0.225 = 0.0675$

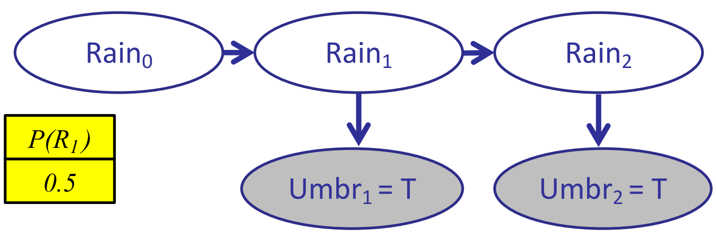
Divide by 0.765 to normalize

$B(x_1=r) = 0.912$

$B(x_0=r) = 0.5$ $B(x_1=r) = 0.875$

$$B'(X_{t+1}) = \sum_{x_t} P(X'|x_t)B(x_t)$$

$$B(X_{t+1}) \propto_{X_{t+1}} P(e_{t+1}|X_{t+1})B'(X_{t+1})$$

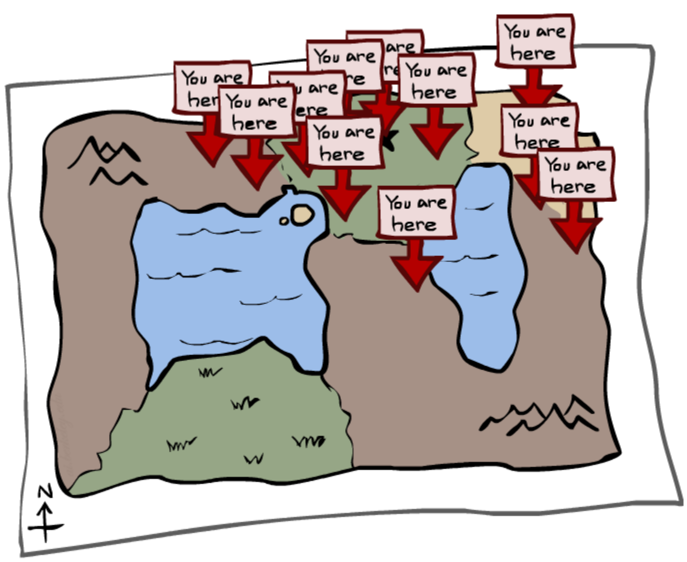


$P(R_1)$
0.5

R_{t-1}	$P(R_t \mid R_{t-1})$
t	0.8
f	0.6

R_t	$P(U_t \mid R_t)$
t	0.9
f	0.3

Particle Filtering



Particle Filtering Overview

- ***Approximation technique*** to solve filtering problem
- Represents P distribution with ***samples***
- Filtering still operates in two steps
  - Elapse time
  - Incorporate observations
    - (But this part has two sub-steps: weight & resample)

38

Particle Filtering

- Sometimes |X| is too big to use exact inference
  - |X| may be too big to even store B(X)
  - E.g. X is continuous
- Solution: approximate inference
  - Track ***samples of X***, not exact distribution of values
  - Samples are called ***particles***
  - Time per step is linear in the number of samples
  - But: number needed may be large
  - In memory: list of particles, not states
- Particle is just new name for ***sample***
- This is how robot localization works in practice

Remember…

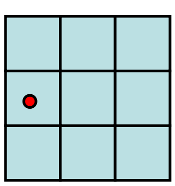
We'll start by looking at this

An HMM is defined by:

- Initial distribution: $P(X_1)$
- Transitions: $P(X_t|X_{t-1})$
- Emissions: $P(E|X)$

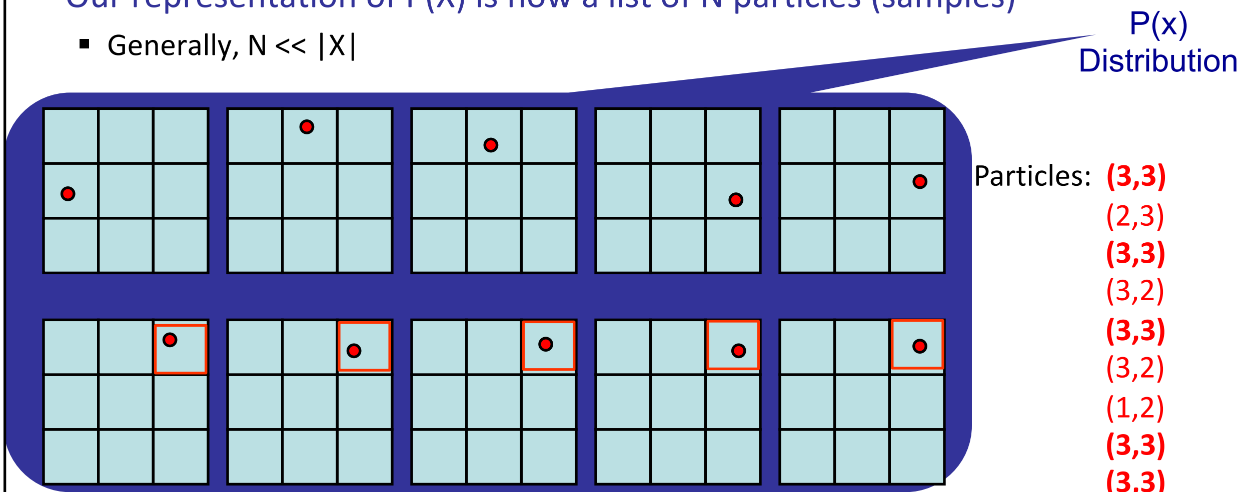
Here's a Single Particle

- It represents a hypothetical state where the robot is in (1,2)



Particles Approximate Distribution

- Our representation of P(X) is now a list of N particles (samples)
  - Generally, N << |X|

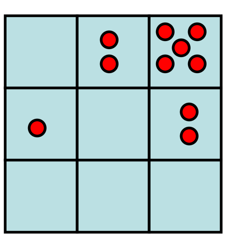


Particles: **(3,3)**
(2,3)
(3,3)
(3,2)
(3,3)
(3,2)
(1,2)
(3,3)
(3,3)
(2,3)

P(x=<3,3>) = 5/10 = 50%

Particle Filtering

A more compact view ***overlays*** the samples:

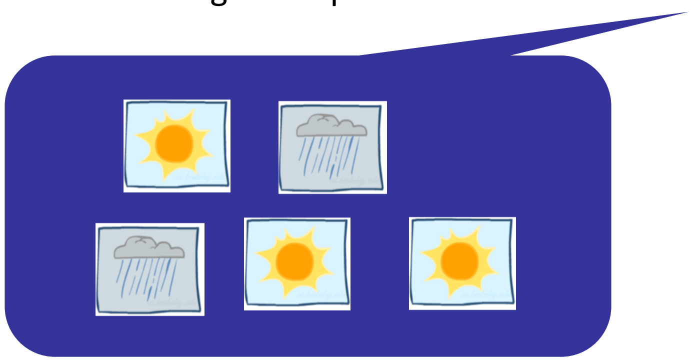


Encodes →

0.0	0.2	0.5
0.1	0.0	0.2
0.0	0.2	0.5

Another Example

In the weather HMM, suppose we decide to approximate the distributions with 5 particles. To initialize the filter, we draw 5 samples from $B(x_0=r) = 0.5$ and we might get the following set of particles:



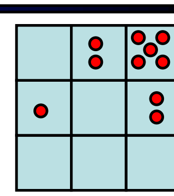
P(x)
Distribution

Particles: S
R
R
S
S

Not such a good approximation, but that's life.

44

Representation: Particles



- Our representation of P(X) is now a list of N particles (samples)
  - Generally, N << |X|
  - Storing map from X to counts would defeat the purpose
- P(x) approximated by **(number of particles with value x) / N**
  - More particles, more accuracy
- What is P((2,2))? 0/10 = 0%
- In fact, many x may have P(x) = 0!

Particles: (3,3)
(2,3)
(3,3)
(3,2)
(3,3)
(3,2)
(1,2)
(3,3)
(3,3)
(2,3)

Particle Filtering Algorithm

1. Elapse Time
2. Observe

   2a. Downweight samples based on evidence

   2b. Resample

47

Particle Filtering: Elapse Time

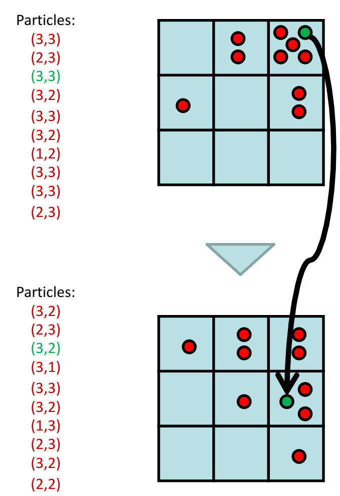
- For each particle, x, move x by sampling its next position from the transition model

$$x' = \text{sample}(P(X'|x))$$

Aka: sample($P(x_{t+1} \mid x_t)$)

  - This is like ***prior sampling*** – samples' frequencies reflect the transition probabilities
  - Here, most samples move clockwise, but some move in another direction or stay in place

- This captures the passage of time
  - If enough samples, close to exact values before and after (consistent)



Particle Filtering: Observe

- Slightly trickier:
  - Don't sample observation, fix it
    - Similar to likelihood weighting,
  - ***For each particle, x, down-weight x based on the evidence***

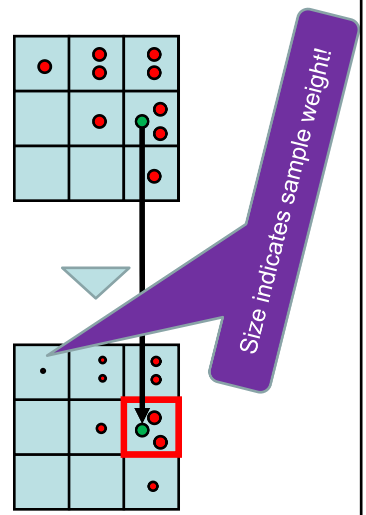
$$w(x) = P(e|x)$$

$$B(X) \propto P(e|X)B'(X)$$

  - As before, the probabilities don't sum to one, since all have been downweighted (in fact they now sum to (N times) an approximation of P(e))

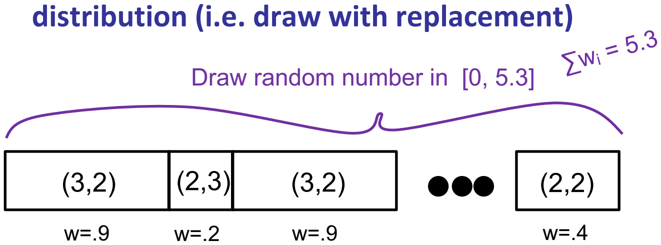
Particles:
(3,2)
(2,3)
(3,2)
(3,1)
(3,3)
(3,2)
(1,3)
(2,3)
(3,2)
(2,2)

Particles:
(3,2) w=.9
(2,3) w=.2
(3,2) w=.9
(3,1) w=.4
(3,3) w=.4
(3,2) w=.9
(1,3) w=.1
(2,3) w=.2
(3,2) w=.9
(2,2) w=.4



Particle Filtering Observe Part II: Resample

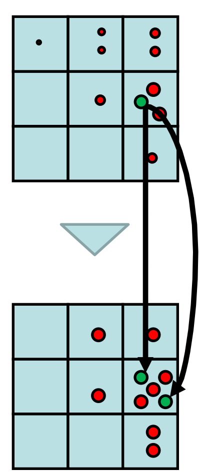
- Rather than tracking weighted samples, we resample
- **N times, we choose from our weighted sample distribution (i.e. draw with replacement)**



- This is equivalent to renormalizing the distribution
- Now the update is complete for this time step, continue with the next one

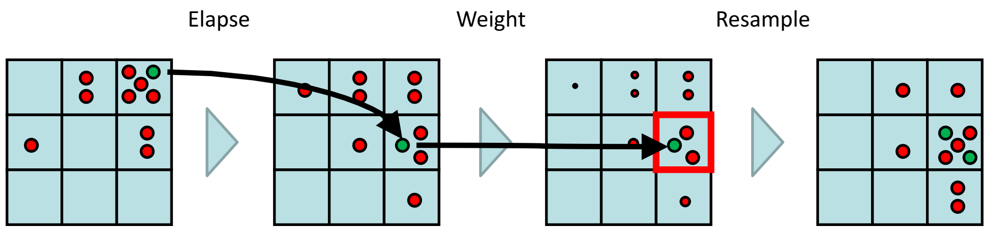
Particles:
(3,2) w=.9
(2,3) w=.2
(3,2) w=.9
(3,1) w=.4
(3,3) w=.4
(3,2) w=.9
(1,3) w=.1
(2,3) w=.2
(3,2) w=.9
(2,2) w=.4

(New) Particles:
(3,2)
(2,2)
(3,2)
(2,3)
(3,3)
(3,2)
(3,1)
(3,2)
(3,1)
(3,2)



Recap: Particle Filtering

- Particles: track samples of states rather than an explicit distribution



Particles:	Particles:	Particles:	(New) Particles:
(3,3)	(3,2)	(3,2) w=.9	(3,2)
(2,3)	(2,3)	(2,3) w=.2	(2,2)
(3,3)	(3,2)	(3,2) w=.9	(3,2)
(3,2)	(3,1)	(3,1) w=.4	(2,3)
(3,3)	(3,3)	(3,3) w=.4	(3,3)
(3,2)	(3,2)	(3,2) w=.9	(3,2)
(1,2)	(1,3)	(1,3) w=.1	(1,3)
(3,3)	(2,3)	(2,3) w=.2	(2,3)
(3,3)	(3,2)	(3,2) w=.9	(3,2)
(2,3)	(2,2)	(2,2) w=.4	(3,2)

[Demos: ghostbusters particle filtering (L15D3,4,5)]

Video of Demo – Moderate Number of Particles

Uniform initialization (!)

Circular dynamics

Video of Demo – One Particle

Uniform initialization (ha!)

Circular dynamics

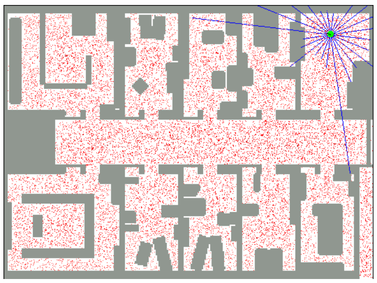
Video of Demo – Huge Number of Particles

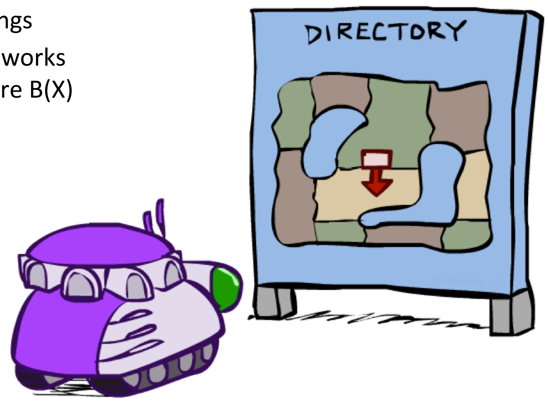
Actually looks uniform !
but look closely

Circular dynamics

Robot Localization

- In robot localization:
  - We know the map, but not the robot's position
  - Observations may be vectors of range finder readings
  - State space and readings are typically continuous (works basically like a very fine grid) and so we cannot store B(X)
  - Particle filtering is a main technique

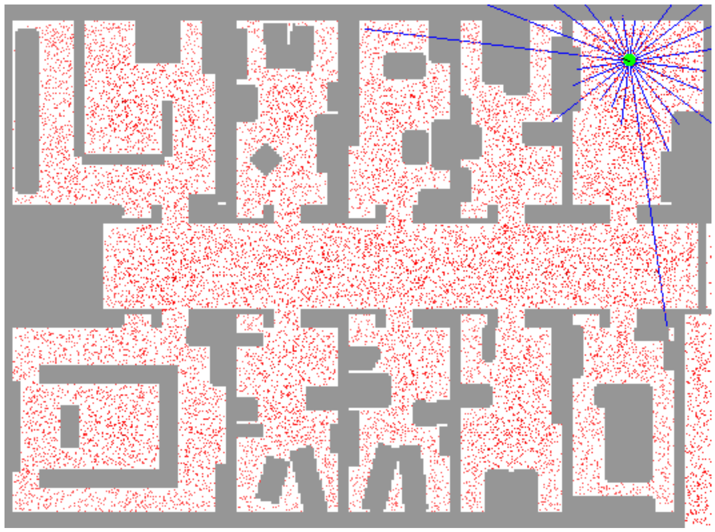




Particle Filter Localization (Sonar)

[Video: global-sonar-uw-annotated.avi]

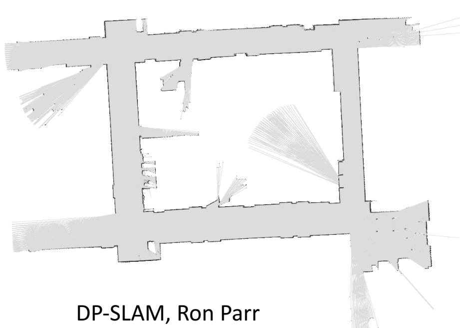
Particle Filter Localization (Laser)



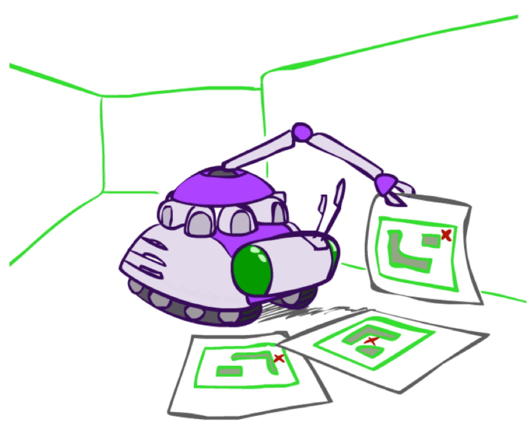
[Video: global-floor.gif]

Robot Mapping

- SLAM: Simultaneous Localization And Mapping
  - We do not know the map or our location
  - State consists of position AND map!
  - Main techniques: Kalman filtering (Gaussian HMMs) and particle methods

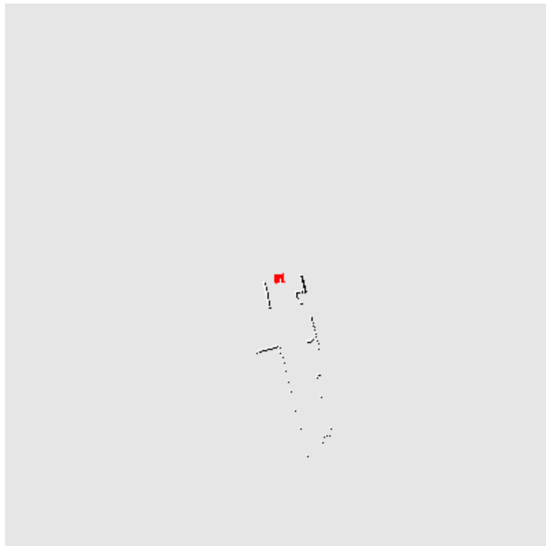


DP-SLAM, Ron Parr



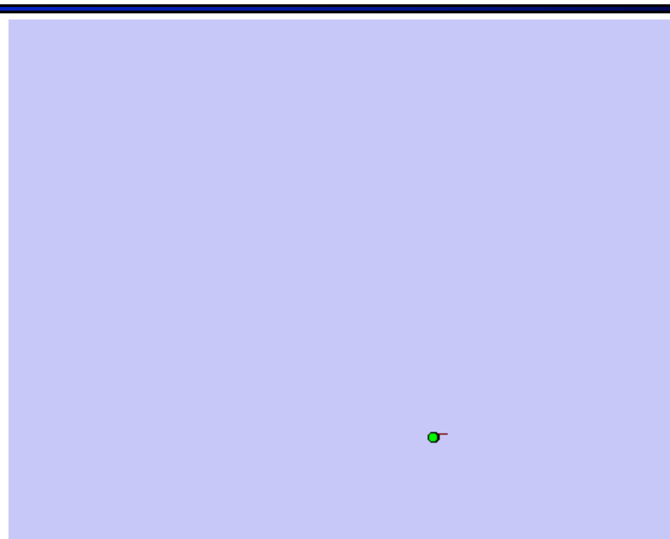
[Demo: PARTICLES-SLAM-mapping1-new.avi]

Particle Filter SLAM – Video 1



[Demo: PARTICLES-SLAM-mapping1-new.avi]

Particle Filter SLAM – Video 2



[Demo: PARTICLES-SLAM-fastslam.avi]