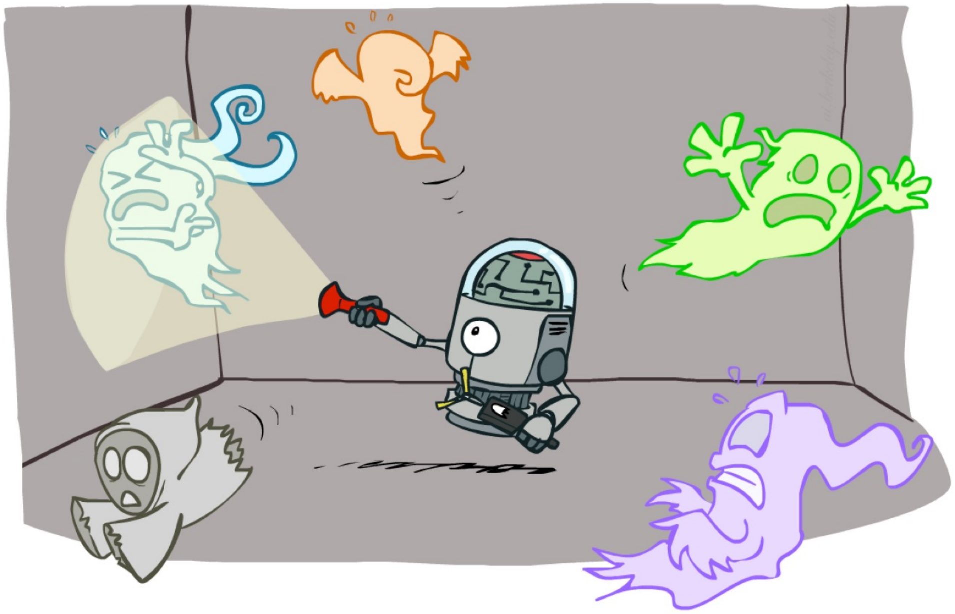
AI: Representation and Problem Solving

Particle Filtering



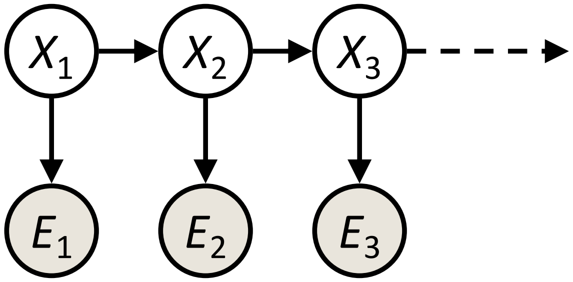
Instructors: Nihar B. Shah and Tuomas Sandholm

Slide credits: CMU AI and ai.berkeley.edu

Hidden Markov Models

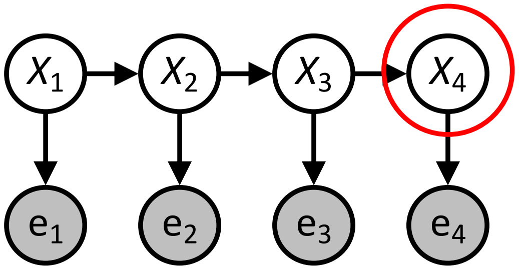
- In many applications, the true state is not observed directly

- Hidden Markov models (HMMs)
  - Underlying Markov chain over states X
  - You observe evidence E at each time step
  - X_t is a single discrete variable; E_t may be continuous and may consist of several variables



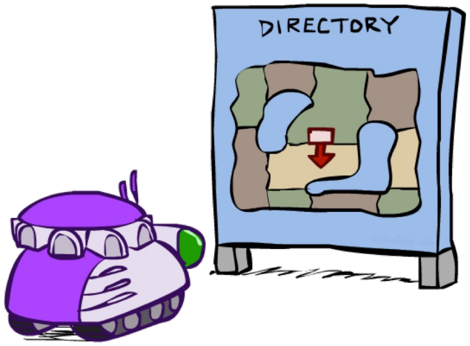
Filtering

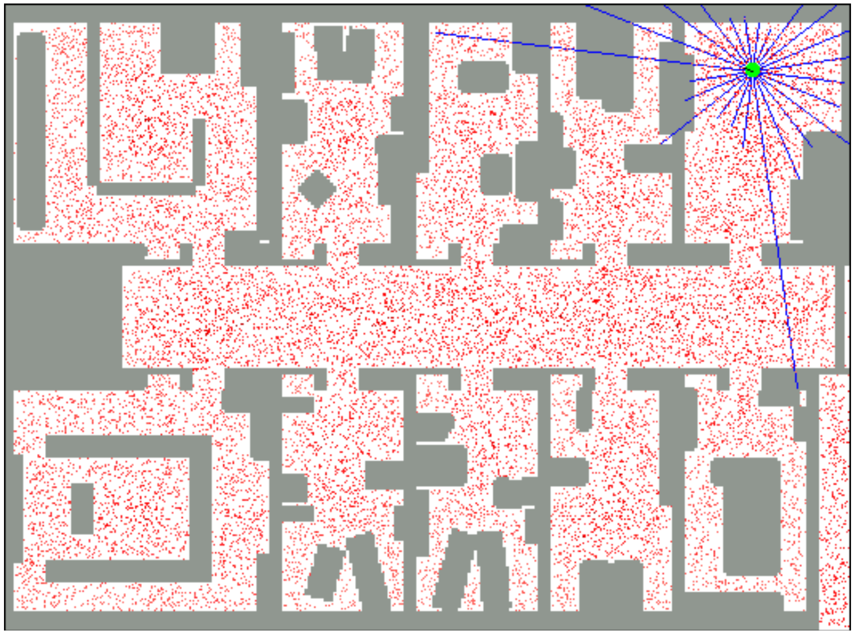
$$P(X_T | e_{1:T})=?$$



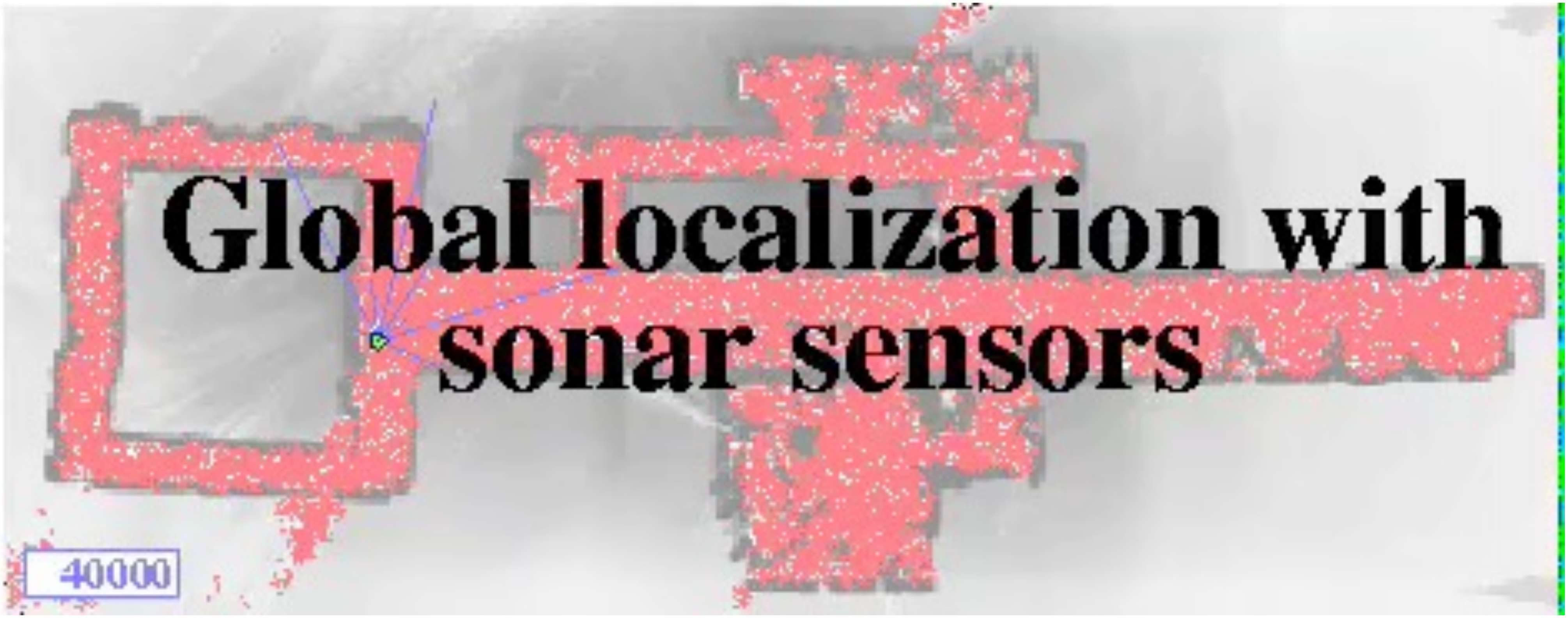
Robot Localization

- We know the map, but not the robot's position
- Observations are some sensor readings
- Another challenge: State space and readings are typically continuous (works basically like a very fine grid) and so we cannot store P(X)





Particle Filter Localization (Sonar)

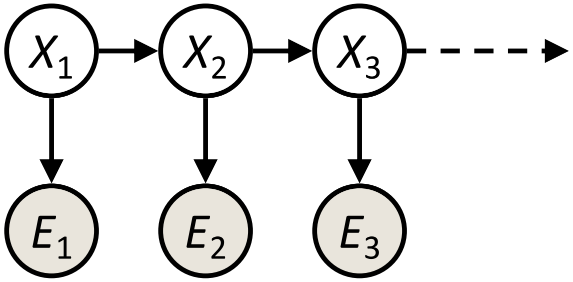


[Dieter Fox, et al.]

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

What about our current algorithms?

Previous lecture: Exact inference



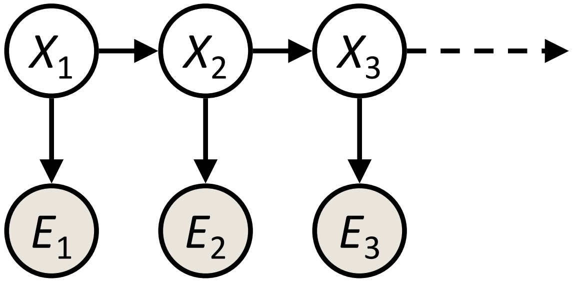
$$P(X_t \mid e_{1:t}) = \underbrace{\alpha}_{\text{Normalize}} \; \underbrace{P(e_t|X_t)}_{\text{Update}} \; \underbrace{\sum_{x_{t-1}} P(X_t|x_{t-1}) \, P(x_{t-1}| \, e_{1:t-1})}_{\text{Predict}}$$

Computational cost per time step:
$O(|X|^2)$ where $|X|$ is the number of states

$O(|X|^2)$ is infeasible for models with many state variables

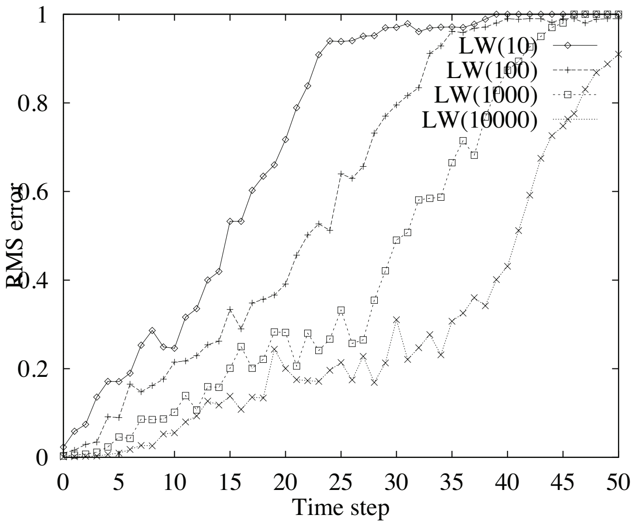
What about our current algorithms?

Want to compute P(X | E). What algorithm have we learnt for this?



- **Likelihood weighted sampling!**
  - For t in 1,2,….
    - Draw x_t~ $P(X_t | X_{t-1}=x_{t-1})$
    - $w=w*P(e_t | x_t)$
  - Take many such samples (say, N samples) and consider their weighted average

- Fails – number of samples needed grows ***exponentially*** with *T*
- ***Why??!!***



Failure of likelihood weighted sampling

- Want to compute P(X | E=e), so want samples which have high probability of E=e

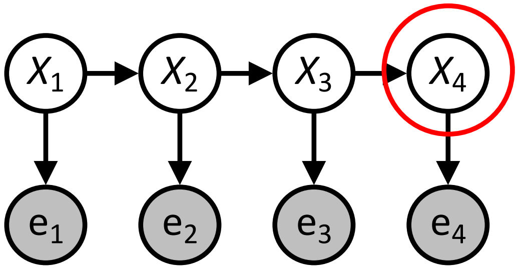
- We sample from $P(X_t \mid X_{t-1})$

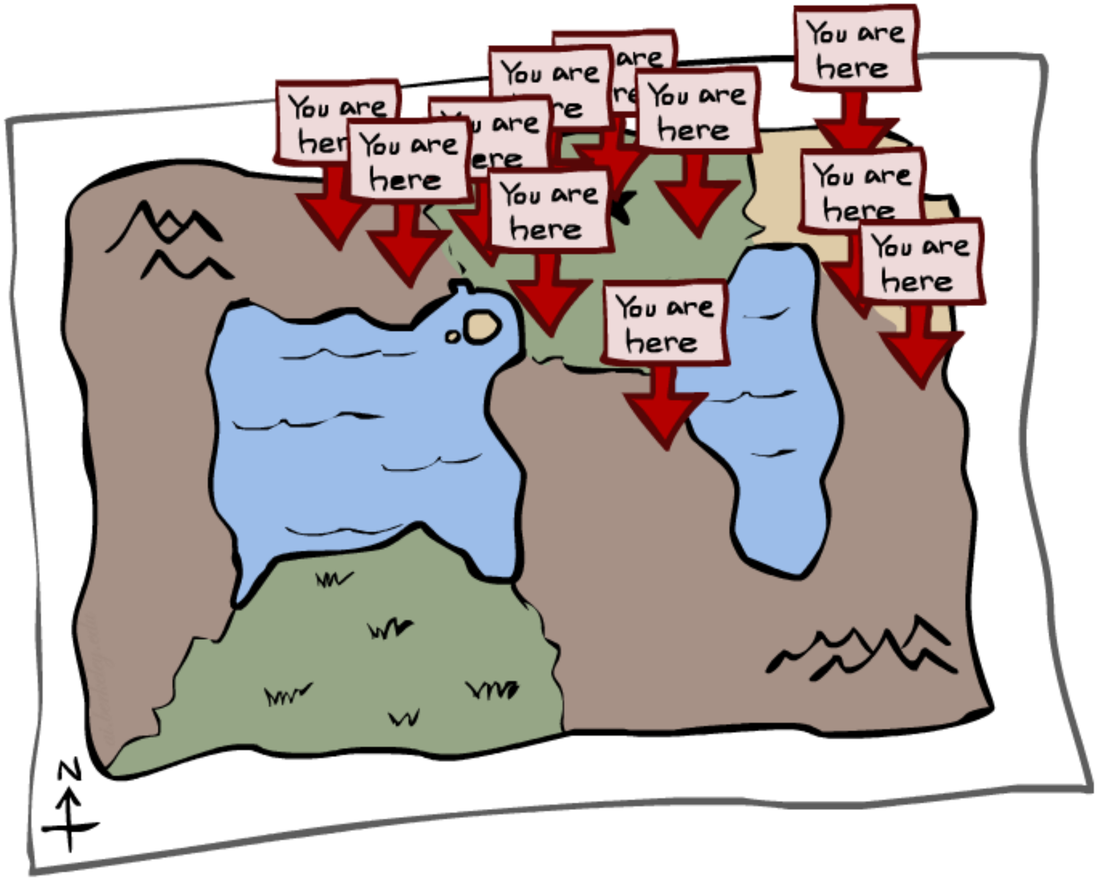
Mismatch!

- Values of X which have high probability in $P(X_t|X_{t-1})$ may have low probability of evidence E=e, i.e., low P(E=e|X=sample)
- As t increases, samples eventually have very low probability of E=e

Particle Filtering

Filtering: $P(X_T | e_{1:T})$=?





New algorithm: **Particle filtering**

- Builds on likelihood weighted sampling

Idea #1!

- Recall that the probability distribution of the samples are based on $P(X_t | X_{t-1})$.
- But we are interested in samples with high probability of $P(E=e | X)$
- What captures this distribution?
  - **The weights!** $(w=w^*P(e_t | x_t))$
  - Let's use weights to somehow create a distribution to draw the samples from… (but how?)

New algorithm: **Particle filtering**

Idea # 2!

- Let's use weights to somehow create a distribution to draw the samples from… (but how?)
- For any sample, we have one weight. How can we really use that as a distribution?
- We are actually drawing N samples. This set of N weights can approximate a distribution!
- Don't draw samples one at a time, but instead in parallel.
  - First draw N samples from $P(X_1)$
  - Set weights for all N samples
  - Then move on to X_2 for all N samples, and so on.
- But how are we using it as a distribution?

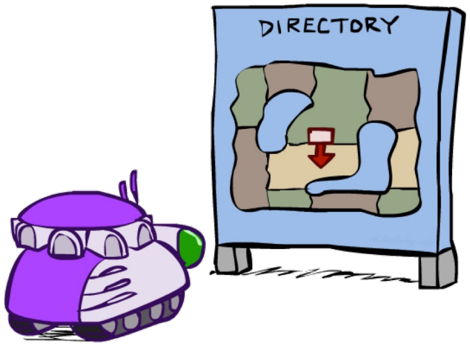
New algorithm: **Particle filtering**

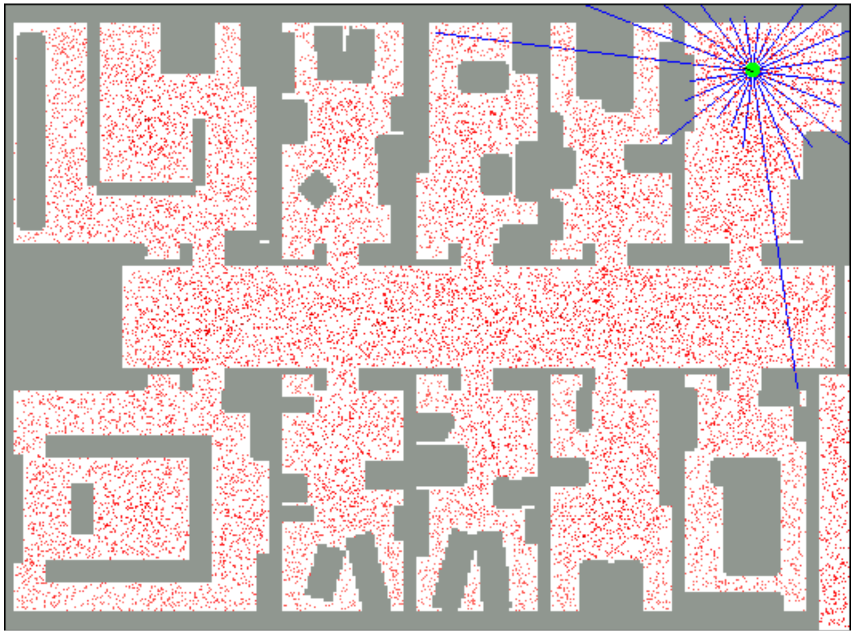
Idea #3 (the algorithm)!

- For each of the N samples, draw next state X_t from $P(X_{t}|X_{t-1})$
- Set weights for all N samples according to $P(E_t|X_t)$
- Denote the N samples as $s_1,s_2,\ldots,s_N$ and their respective weights as $w_1,w_2,\ldots,w_N$
- Normalize the weights: $w_i \leftarrow \frac{w_i}{\sum_{j=1}^{N} w_j}$
- **Resample** each of the N samples
  - For each i ϵ 1,…,N:
    - Draw $\tilde{s}_i$ at random: $\tilde{s}_i = s_j$ with probability w_j across j=1,…,N
  - Set all weights to 1
- Repeat for X_{t+1} and so on

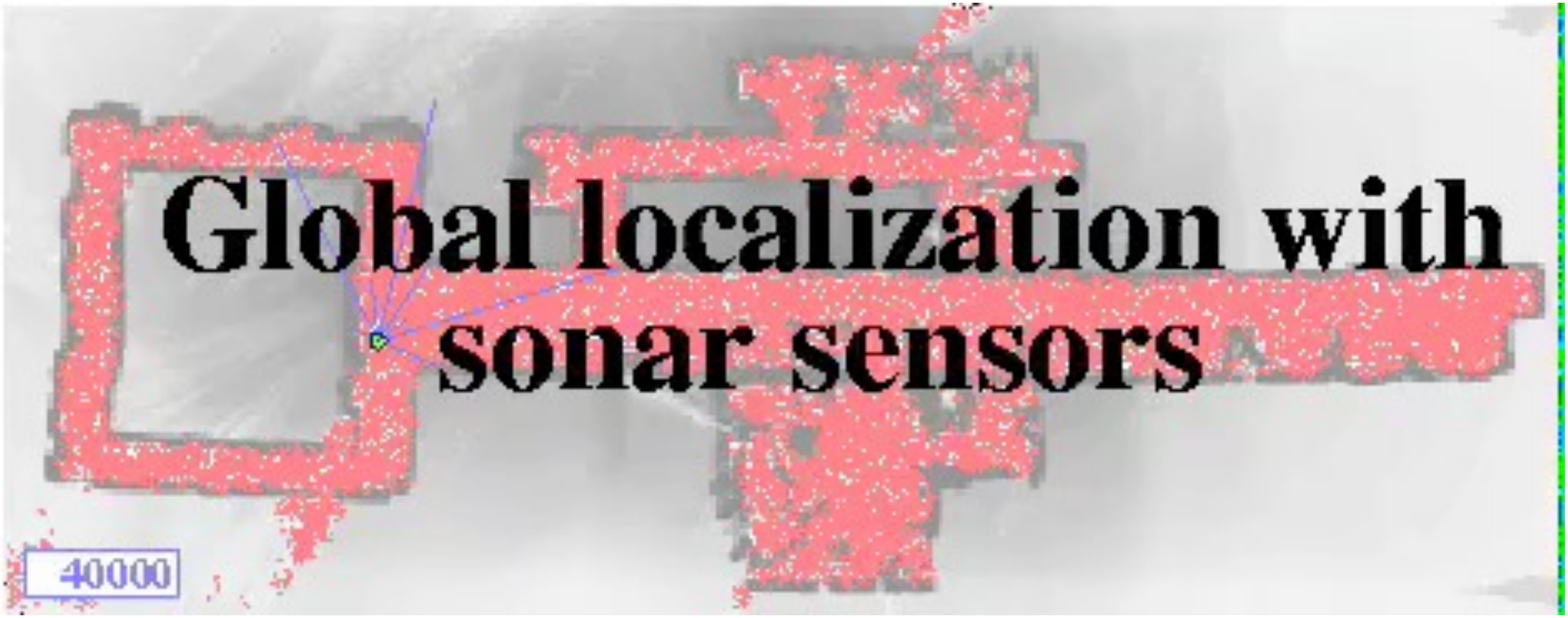
Robot Localization

- We know the map, but not the robot's position
- Observations are some sensor readings
- Another challenge: State space and readings are typically continuous (works basically like a very fine grid) and so we cannot store P(X)





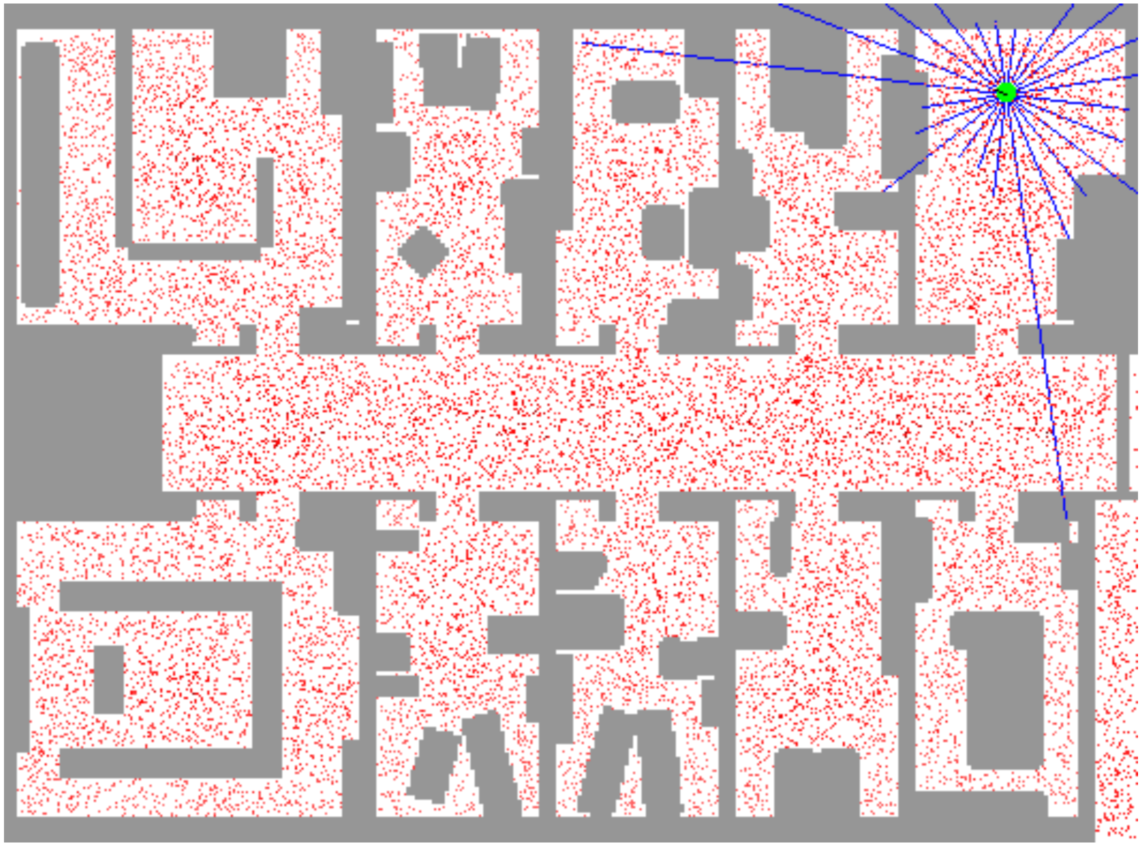
Particle Filter Localization (Sonar)



[Dieter Fox, et al.]

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

Particle Filter Localization (Laser)



[Dieter Fox, et al.]

[Video: global-floor.gif]

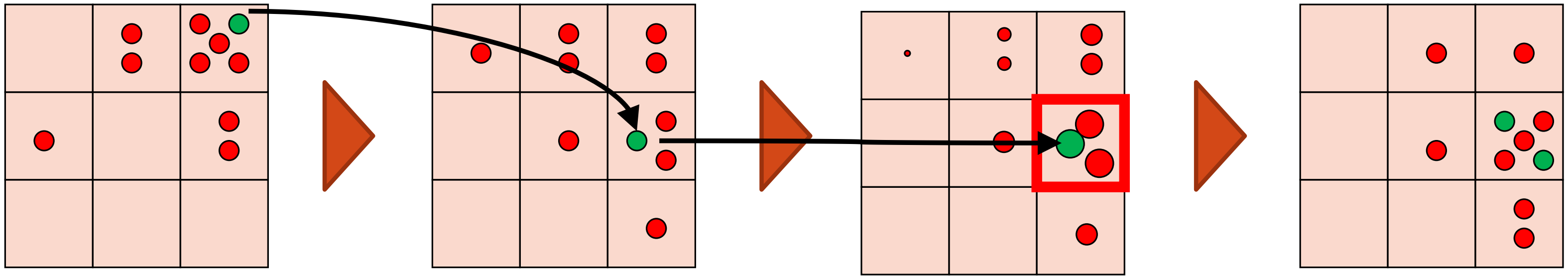
Summary: Particle Filtering

Particles: track samples of states rather than an explicit distribution

Propagate forward based on transition function

Weight based on observation function

Resample using weighted particles



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

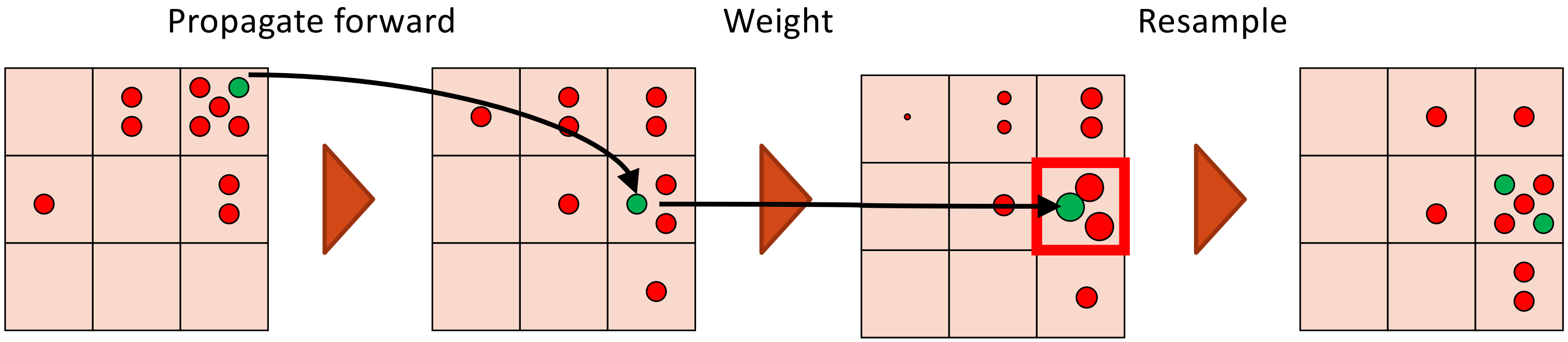
Particles:
(3,2)
(2,3)
(3,2)
(3,1)
(3,3)
(3,2)
(1,3)
(2,3)
(3,2)
(2,2)

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
(3,3) w=.4
(2,2) w=.4

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

Question

If we only have one particle which of these steps are unnecessary?



Select all that are unnecessary.

A. Propagate forward

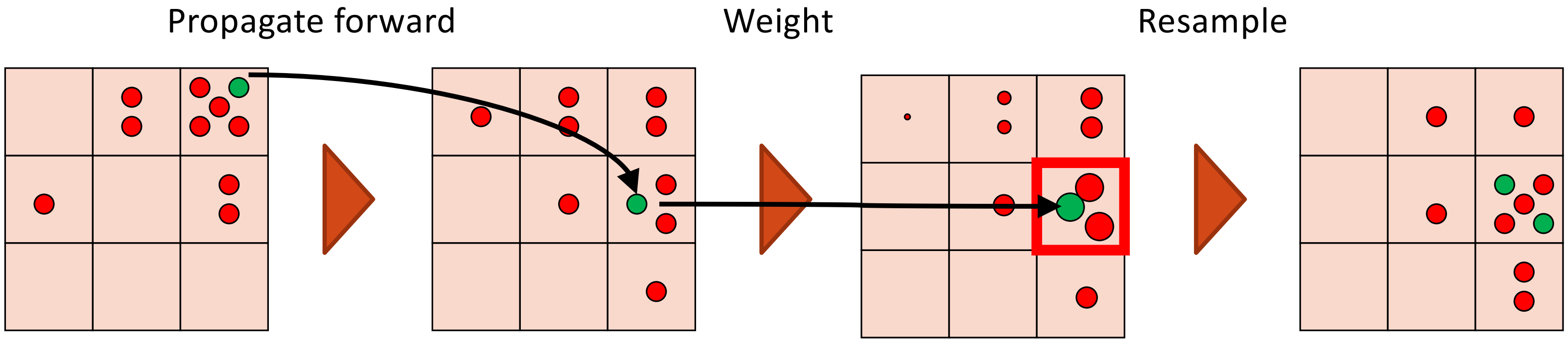
B. Weight

C. Resample

D. None of the above

Question

If we only have one particle which of these steps are unnecessary?



Select all that are unnecessary.

A. Propagate forward

B. Weight

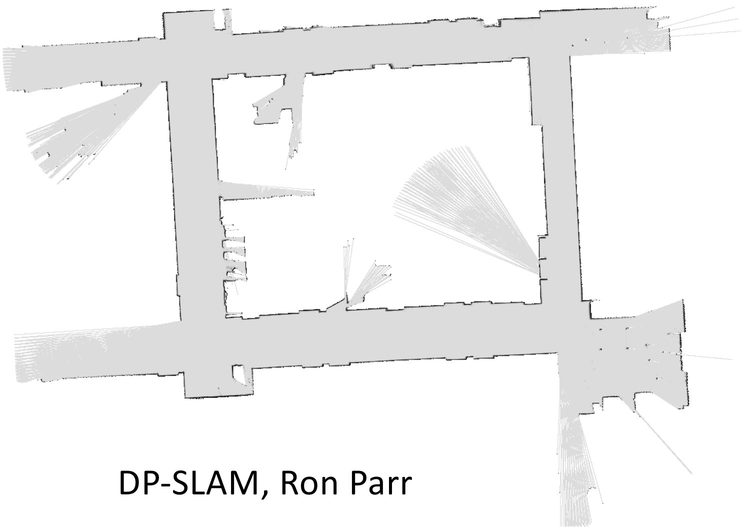
C. Resample

D. None of the above

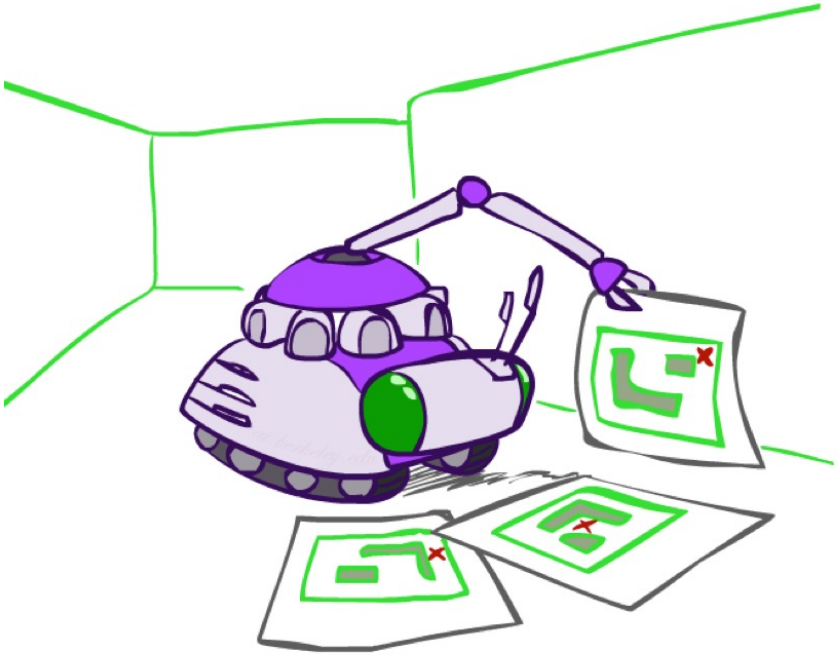
Unless the weight is zero, in which case, you'll want to resample from the beginning ☹

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

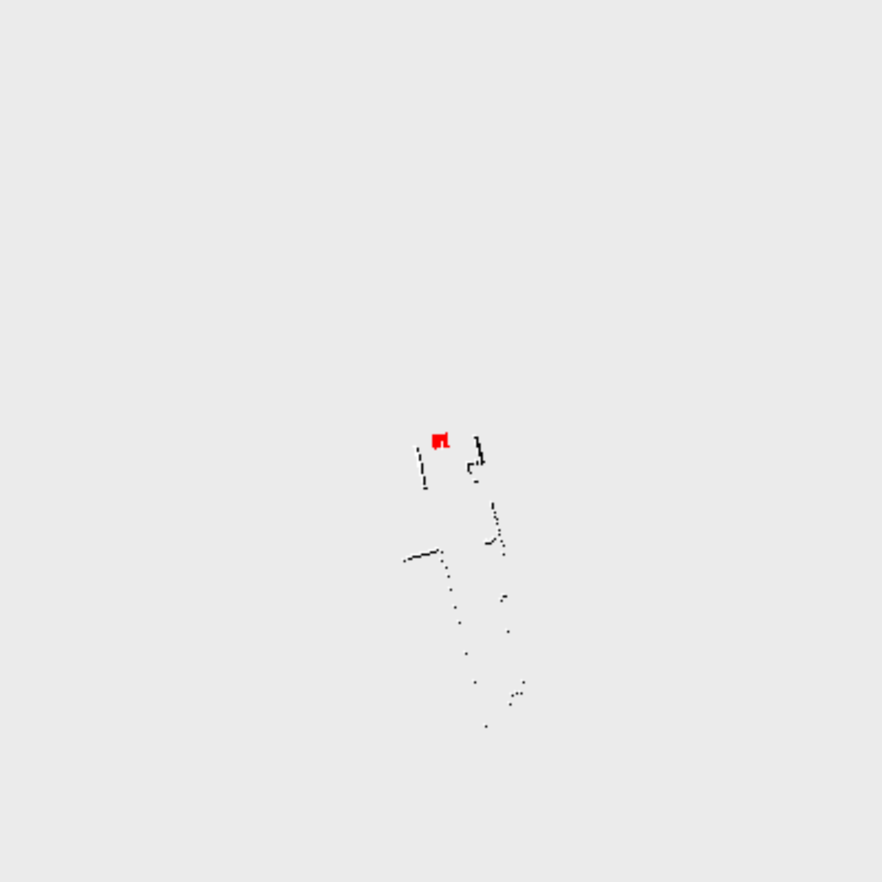


DP-SLAM, Ron Parr



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

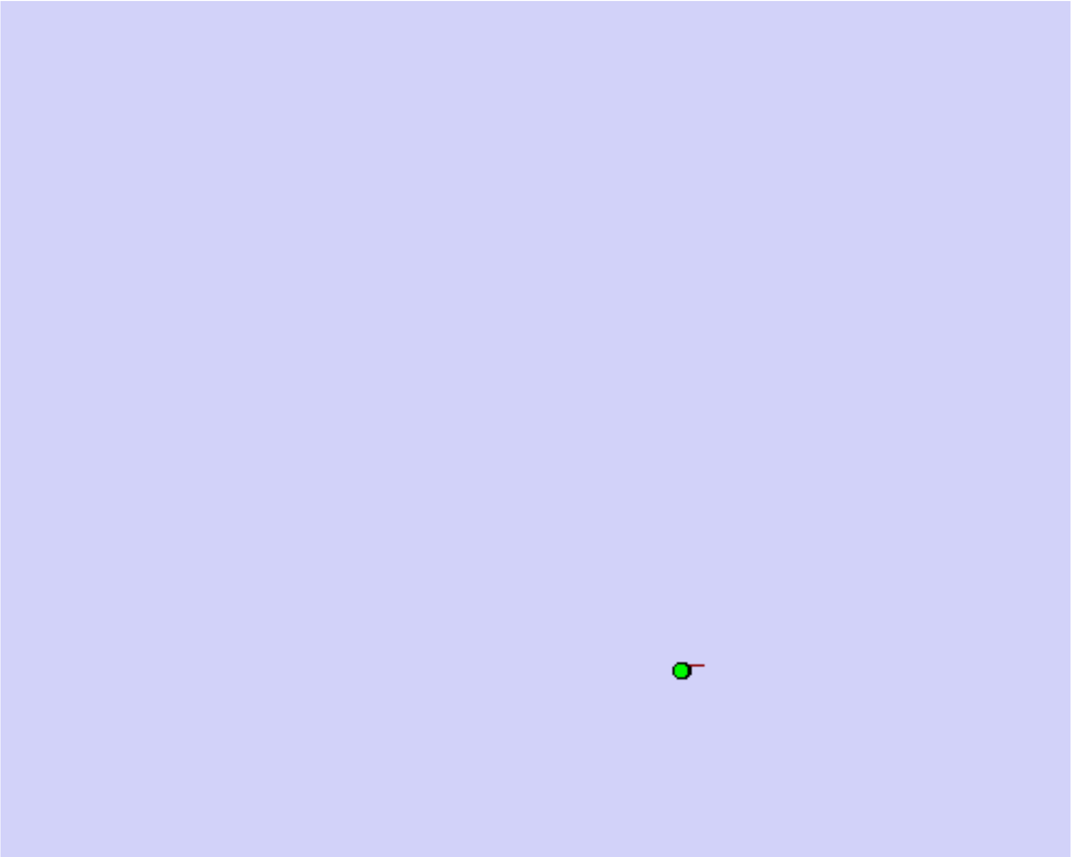
Particle Filter SLAM – Video 1



[Sebastian Thrun, et al.]

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

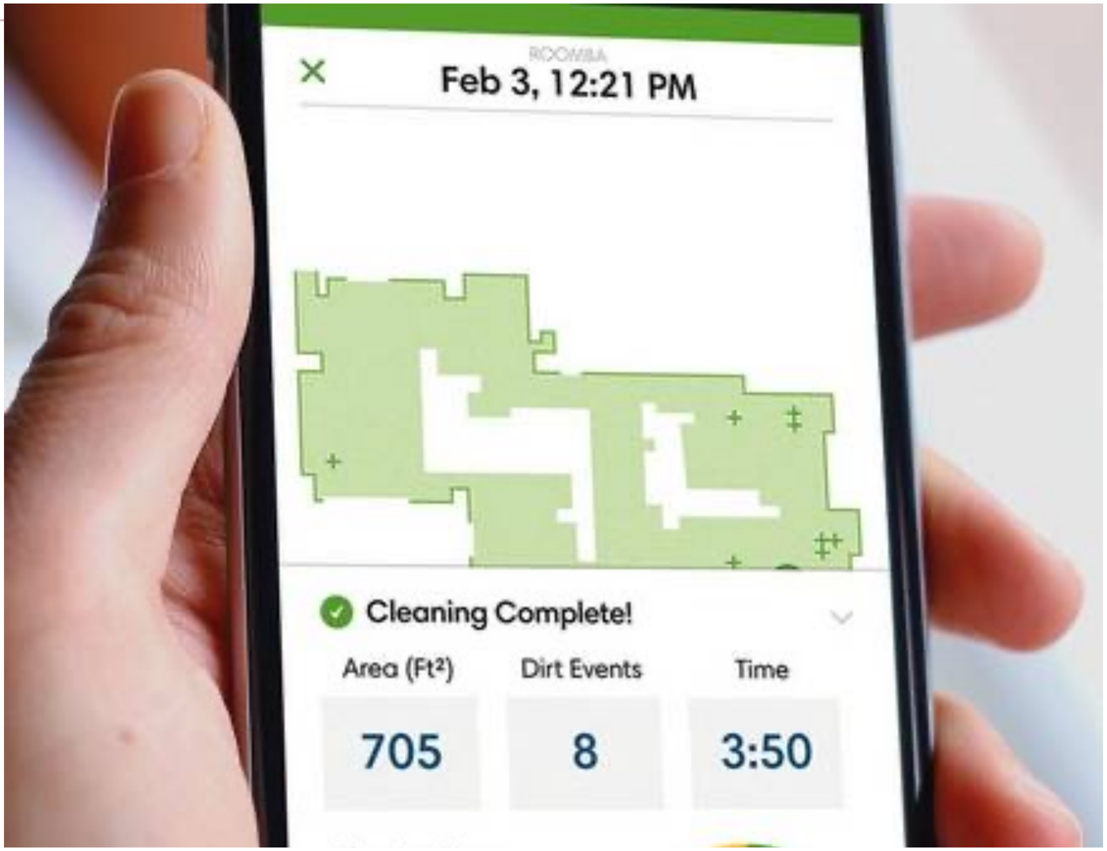
Particle Filter SLAM – Video 2



[Dirk Haehnel, et al.]

[Demo: PARTICLES-SLAM-fastslam.avi]

SLAM


ROOMBA

Feb 3, 12:21 PM

Cleaning Complete!

Area (Ft²)	Dirt Events	Time
705	8	3:50

https://www.irobot.com/

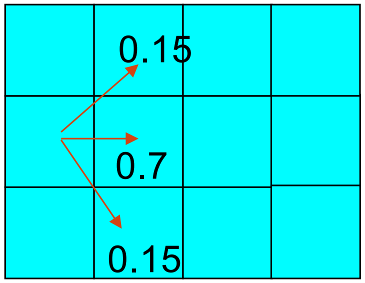
In Class Activity

Given the following starting particles, transition model, and e_1 and e_2 observed at time 1 and time 2, estimate $\mathbf{P(X_2 \mid e_1,e_2)}$.

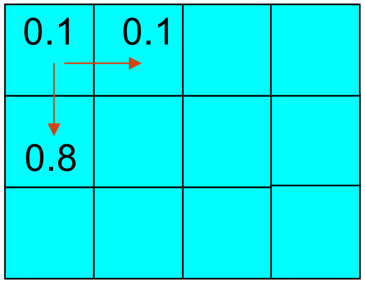
T=0

● ●	●	●	
●	●	●	
●	●	●	

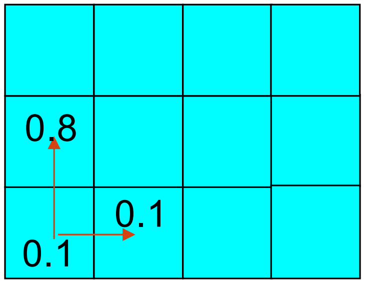
$P(X_{t+1}|X_t \text{ in middle row})$



$P(X_{t+1}|X_t \text{ in top row})$



$P(X_{t+1}|X_t \text{ in bottom row})$



$P(e_1|x_1)$

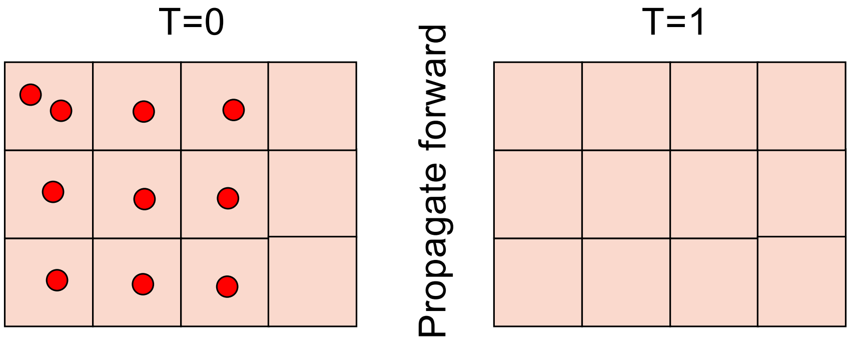
.3	.5		
.5	.5		
.2	.5		

$P(e_2|x_2)$

	.05	.4	
	.3	.5	
	.05	.2	

In Class Activity

Given the following starting particles, transition model, and e_1 and e_2 observed at time 1 and time 2, estimate $\mathbf{P(X_1 \mid e_1)}$.

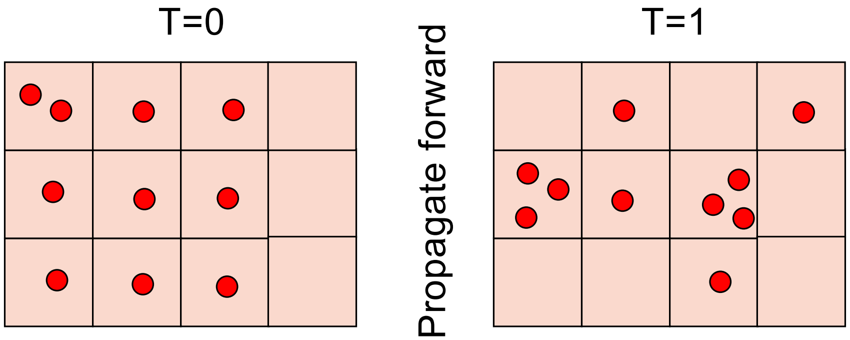


$P(e_1|x_1)$

.3	.5		
.5	.5		
.2	.5		

In Class Activity

Given the following starting particles, transition model, and e_1 and e_2 observed at time 1 and time 2, estimate $\mathbf{P(X_1 \mid e_1)}$.



Weight based on e_1

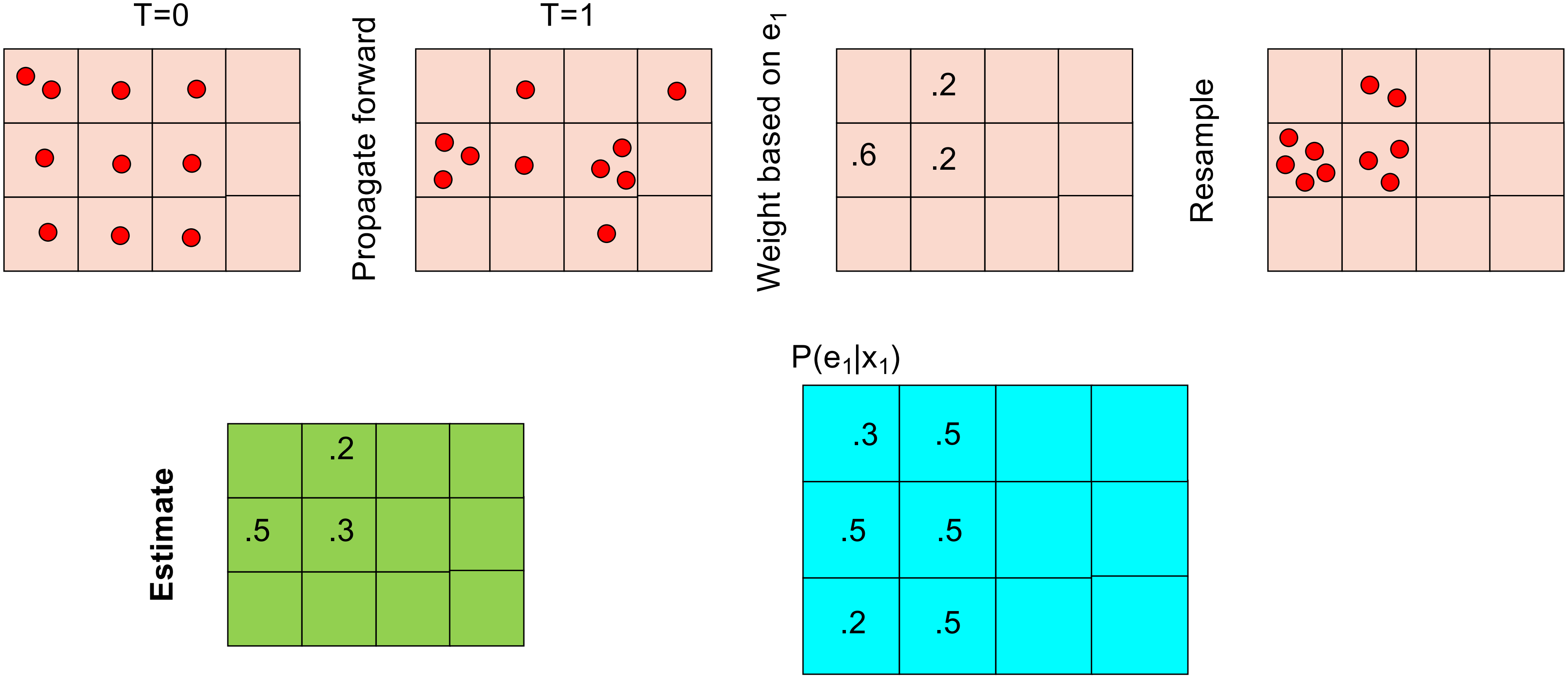
	.5		
1.5	.5		

$P(e_1|x_1)$

.3	.5		
.5	.5		
.2	.5		

In Class Activity

Given the following starting particles, transition model, and e_1 and e_2 observed at time 1 and time 2, estimate $\mathbf{P(X_1 \mid e_1)}$.



Weight based on e_1

	.2		
.6	.2		

Estimate

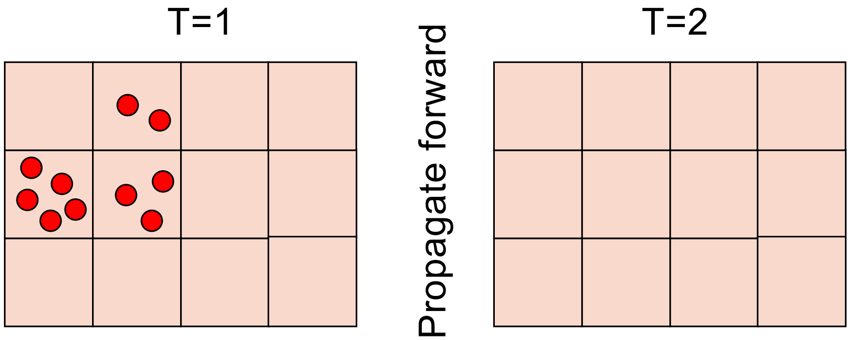
	.2		
.5	.3		

$P(e_1|x_1)$

.3	.5		
.5	.5		
.2	.5		

In Class Activity

Given the particles at T=1, transition model, and e_2 observed at time 2, estimate **$P(X_2 \mid e_1, e_2)$**.

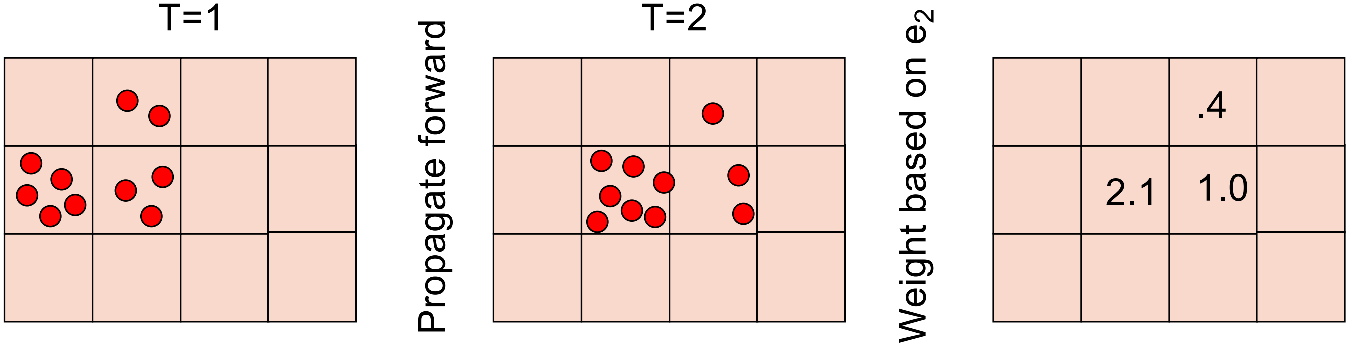


$P(e_2|x_2)$

	.05	.4	
	.3	.5	
	.05	.2	

In Class Activity

Given the particles at T=1, transition model, and e_2 observed at time 2, estimate $\mathbf{P(X_2 \mid e_1,e_2)}$.

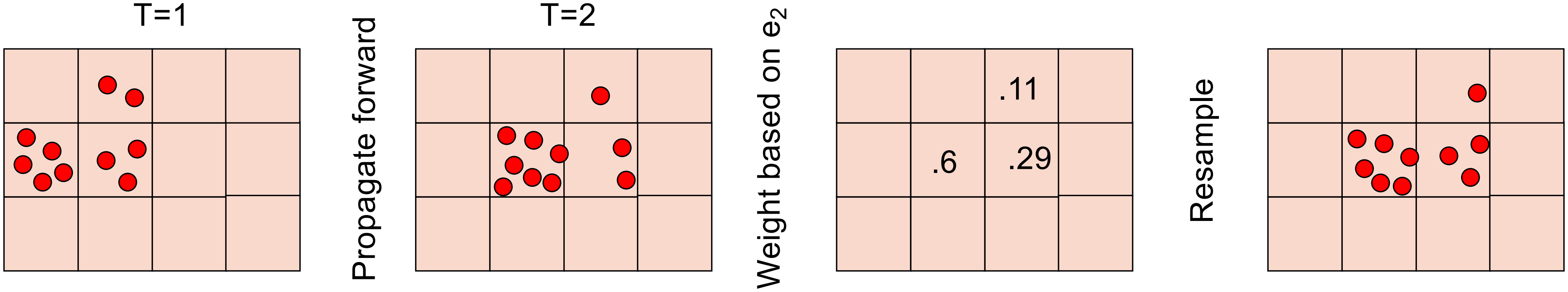


$P(e_2|x_2)$

	.05	.4	
	.3	.5	
	.05	.2	

In Class Activity

Given the particles at T=1, transition model, and e_2 observed at time 2, estimate $\mathbf{P(X_2 \mid e_1, e_2)}$.



		.11	
	.6	.29	

Estimate

		.1	
	.6	.3	

$P(e_2|x_2)$

	.05	.4	
	.3	.5	
	.05	.2	