PARTICLE FILTERING

CSE 511A: Introduction to Artificial Intelligence

Some content and images are from slides 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.

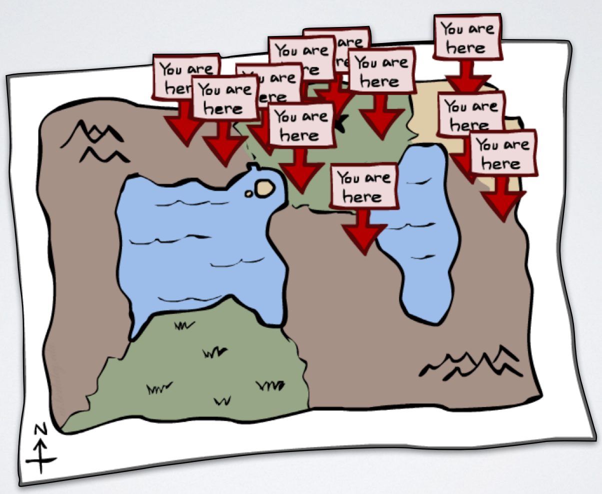
1

INFERENCE TASKS

- Estimation: $P(X_t \mid E_t)$
  - e.g., likelihood that you have COVID-19 today given than you have a fever today
- Prediction: $P(X_{t+1} \mid E_t,\ldots,E_1)$
  - e.g., likelihood that you have COVID-19 tomorrow in the future given your fever history
- Filtering: $P(X_t \mid E_t,\ldots,E_1)$
  - e.g., likelihood that you have COVID-19 today given your fever history
- Smoothing: $P(X_k \mid E_t,\ldots,E_1)$
  - e.g., likelihood that you have COVID-19 in the past given your fever history
- Most likely sequence: $\text{argmax}_{X_1,\ldots,X_t} P(X_1,\ldots,X_t \mid E_t,\ldots,E_1)$
  - e.g., most likely COVID-19 history given your fever history

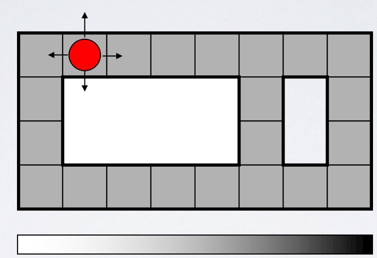
2

ROBOT LOCALIZATION



3

ROBOT LOCALIZATION



Probability 0 1

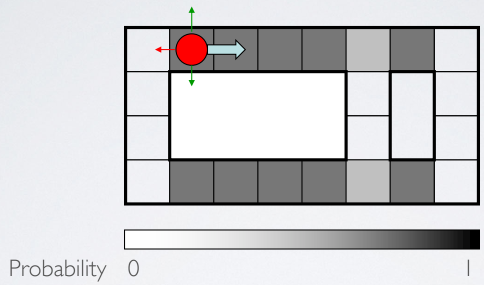
t=0

- Robot knows the map but doesn't know where it is.
- Initially assumes that it can be anywhere on the map with equal probability.
- It is able to reasonably detect in which direction there is a wall.
- It is reasonably confident that it will move in the direction that it wants to move.

Source: Michael Pfeiffer

4

ROBOT LOCALIZATION



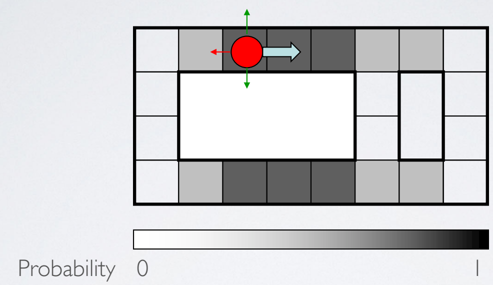
t=1

- Robot detects two walls on opposite sides.
- Updates its belief of where it is based on its observation and its prior belief.
- Lighter grey cells are due to the possibility that it received a wrong reading in one of the directions.

Source: Michael Pfeiffer

5

ROBOT LOCALIZATION



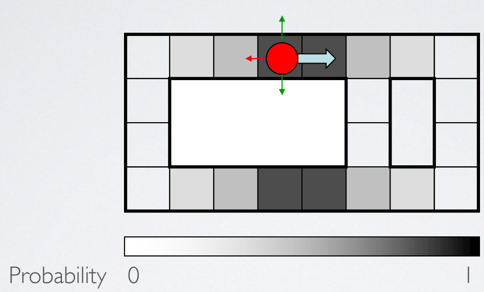
t=2

- Robot detects two walls on opposite sides.
- Updates its belief of where it is based on its observation and its prior belief and its movement.

Source: Michael Pfeiffer

6

ROBOT LOCALIZATION



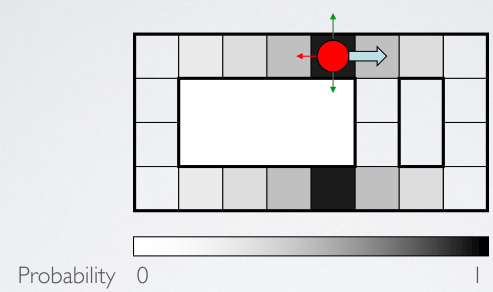
t=3

- Robot detects two walls on opposite sides.
- Updates its belief of where it is based on its observation and its prior belief and its movement.

Source: Michael Pfeiffer

7

ROBOT LOCALIZATION



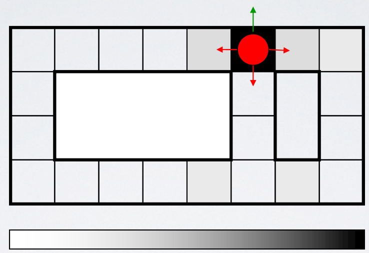
t=4

- Robot detects two walls on opposite sides.
- Updates its belief of where it is based on its observation and its prior belief and its movement.

Source: Michael Pfeiffer

8

ROBOT LOCALIZATION



Probability 0 1

t=5

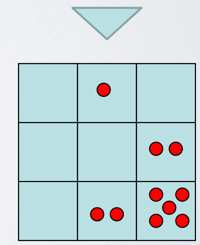
- Robot detects two walls on opposite sides.
- Updates its belief of where it is based on its observation and its prior belief and its movement.
- It is now highly confident of its location.

Source: Michael Pfeiffer

PARTICLE FILTERING

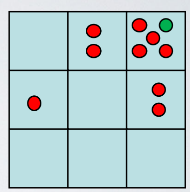
- Particle filtering: An approximation approach for filtering
- Problem:
  - State space |X| may be too large to store the belief probability *B(X)* over all states
  - State space may be continuous
- Solution:
  - Track samples (called particles) instead of values
  - Collectively, they approximate the belief over all states
  - Update particles based on sensor and motion models
- This is how robot localization works in practice

0.0	0.1	0.0
0.0	0.0	0.2
0.0	0.2	0.5



PARTICLE FILTERING

- Representation of *P(X)* is now a list of *N* particles
  - Generally, *N* << |X|
  - Don't store mapping of *X* to counts of particles. Would defeat the purpose of using less space
- The belief *P(x)* over a state *x* is approximated by the number of particles with value *x*
  - So, many states *x* may have *P(x)* = *0*!
  - More particles, more accuracy
- For now, all particles have a weight of 1

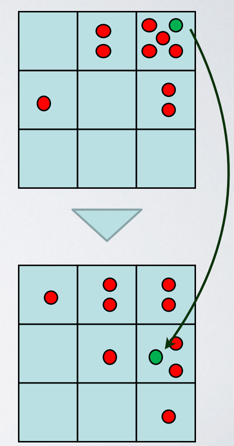


Particles:
(3,3) (3,2)
(2,3) (1,2)
(3,3) (3,3)
(3,2) (3,3)
(3,3) (2,3)

PARTICLE FILTERING

- Assume that the robot took action *a* (= clockwise in this example).
- Each particle is moved by sampling its next position from the transition function
  - i.e., it moves to its next position *x'* with probability P(*x'* | *a*, *x*)

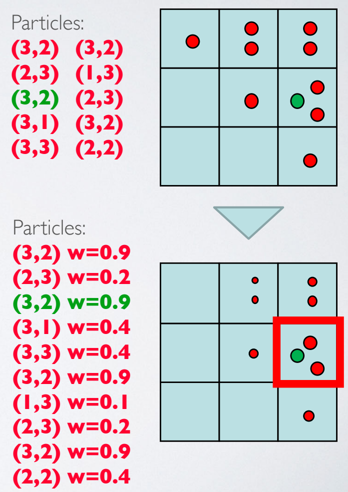
Particles:
(3,3) (3,2)
(2,3) (1,2)
(3,3) (3,3)
(3,2) (3,3)
(3,3) (2,3)



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

PARTICLE FILTERING

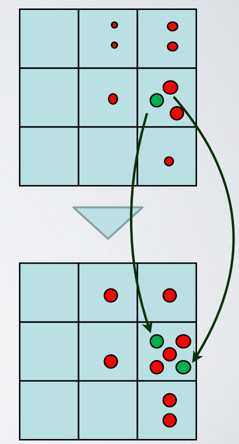
- Assume that the robot observed observation *o*. (= observed the highlighted cell in this example)
- Update weight of particle based on observation.
  - i.e., it's weight $w(x) = P(o \mid x)$
- Note that probabilities don't sum to one here, which is okay. They actually sum up to *N* times an approximation of $P(o)$.



13

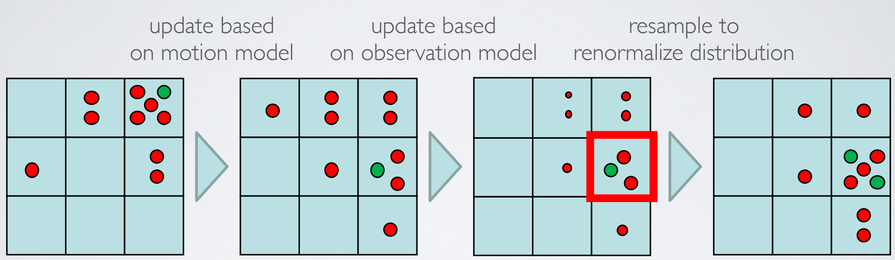
PARTICLE FILTERING

- Instead of tracking weighted samples, we resample so weights of all samples = 1.
- *N* times, choose from our weighted sample distribution
  - Possible to sample the same sample repeatedly.
- Equivalent to renormalizing the distribution
- Update is complete for this time step, continue with next time step (move, then observe, then resample)

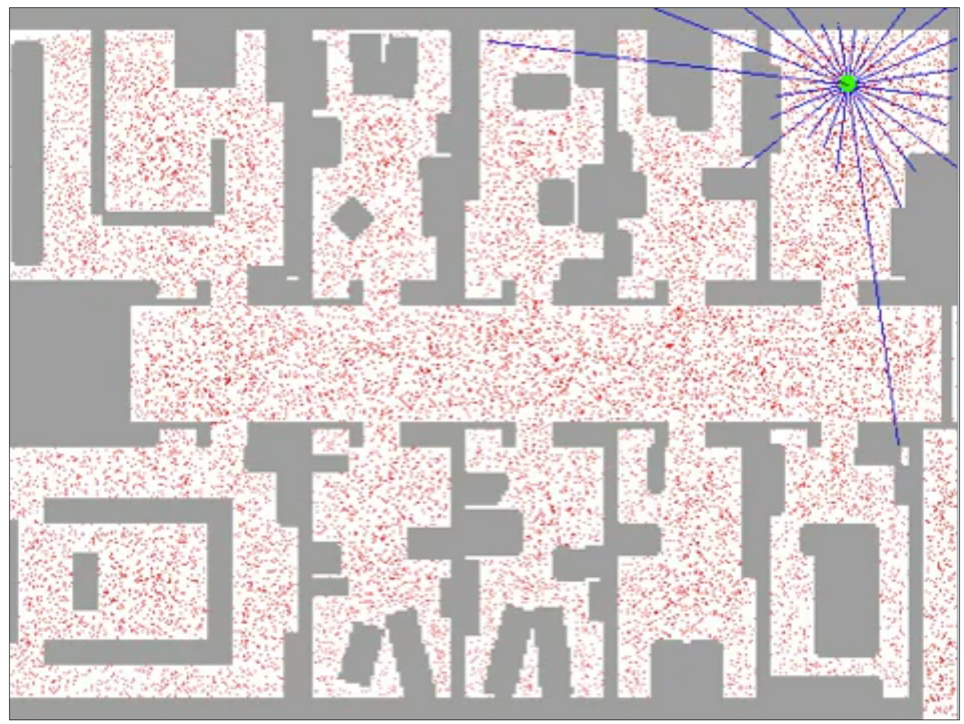


14

PARTICLE FILTERING



15



16