

# **Rao-Blackwellized Particle Filtering**

Pieter Abbeel  
UC Berkeley EECS

Many slides adapted from Thrun, Burgard and Fox, Probabilistic Robotics

# Particle Filters Recap

1. Algorithm `particle_filter`(  $S_{t-1}, u_t, z_t$ ):

2.  $S_t = \emptyset, \eta = 0$

3. **For**  $i = 1 \dots n$

*Generate new samples*

4. Sample index  $j(i)$  from the discrete distribution given by  $w_{t-1}$

5. Sample  $x_t^i$  from  $\pi(x_t | x_{t-1}^{j(i)}, u_t, z_t)$

6.  $w_t^i = \frac{p(z_t | x_t^i) p(x_t^i | x_{t-1}^{j(i)}, u_t)}{\pi(x_t^i | x_{t-1}^{j(i)}, u_t, z_t)}$

*Compute importance weight*

7.  $\eta = \eta + w_t^i$

*Update normalization factor*

8.  $S_t = S_t \cup \{ \langle x_t^i, w_t^i \rangle \}$

*Insert*

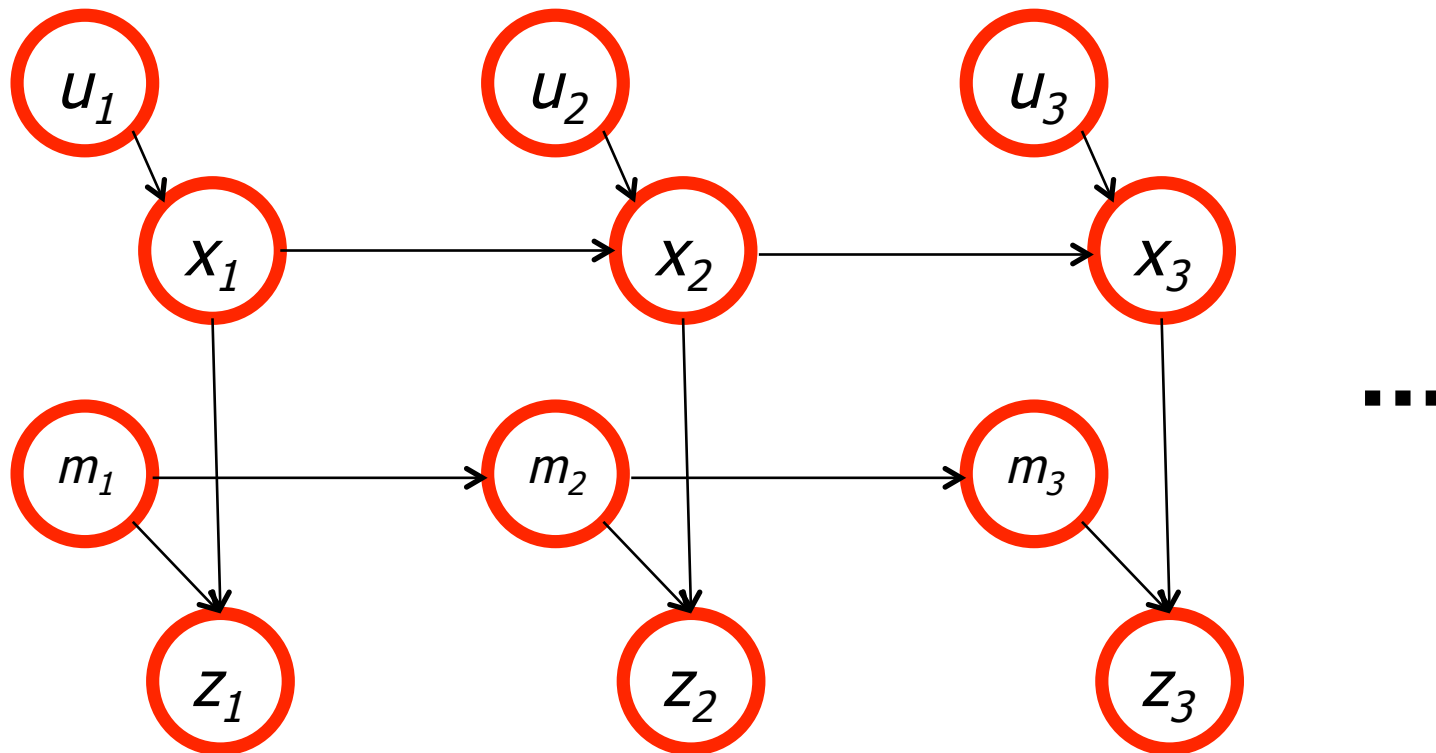
9. **For**  $i = 1 \dots n$

10.  $w_t^i = w_t^i / \eta$

*Normalize weights*

11. **Return**  $S_t$

# Motivating Example: Simultaneous Localization and Mapping (SLAM)



- $m_t$ : map at time  $t$ 
  - Often map is assumed static, then denoted by  $m$

# Naive Particle Filter for SLAM

- Each particle  $\langle (x^i, m^i), w^i \rangle$  encodes a weighted hypothesis of robot pose and map
- E.g., 20m x 10m space, mapped at 5cm x 5cm resolution
  - $400 \times 200 = 80,000$  cells
  - $2^{80,000}$  possible maps
- Impractical to get sufficient coverage of such a large state space

# Particle Filter Revisited

Let's consider just the robot pose:

- Sample from  $\pi(x_t | x_{t-1}^i, u_t, z_t)$

- Reweight  $w_t^i = \frac{p(z_t | x_t^i) p(x_t^i | x_{t-1}^i, u_t)}{\pi(x_t^i | x_{t-1}^i, u_t, z_t)}$

Recall a particle really corresponds to an entire history, this will matter going forward, so let's make this explicit, also account for the fact that by ignoring the other state variable, we lost Markov property:

- Reweight  $w_t^i = \frac{p(z_t | x_{1:t}^i, z_{1:t-1}) p(x_t^i | x_{1:t-1}^i, u_t, z_{1:t-1})}{\pi(x_t^i | x_{t-1}^i, u_t, z_t)}$

Still defines a valid particle filter just for  $x$ , BUT as  $z$  depends both on  $x$  and  $m$ , some quantities are not readily available (yet).

# Weights Computation

$$w_t^i = \frac{p(z_t | x_{1:t}^i, z_{1:t-1}) p(x_t^i | x_{1:t-1}^i, u_t, z_{1:t-1})}{\pi(x_t^i | x_{t-1}^i, u_t, z_t)}$$

- $$p(z_t | x_{1:t}^i, z_{1:t-1}) = \int p(z_t | x_{1:t}^i, m_t, z_{1:t-1}) p(m_t | z_{1:t-1}, x_{1:t-1}^i) dm_t$$
$$= \int p(z_t | x_t^i, m_t) p(m_t | z_{1:t-1}, x_{1:t-1}^i) dm_t$$

sensor model

mapping with KNOWN poses

This integral is over large space, but we'll see how to still compute it efficiently (sometimes approximately).

- $$p(x_t^i | x_{1:t-1}^i, u_t, z_{1:t-1}) = p(x_t^i | x_{t-1}^i, u_t)$$

motion model

# Examples

- We'll consider  $\pi(x_t | x_{t-1}^i, u_t, z_t) = p(x_t | x_{t-1}^i, u_t)$

hence  $w_t^i = \int p(z_t | x_t^i, m_t) p(m_t | z_{1:t-1}, x_{1:t-1}^i) dm_t$

- Examples for which  $w_t^i$  can be computed efficiently
  - “Color-tile” SLAM
  - FastSLAM:
    - Not in this lecture.
  - SLAM with gridmaps

# “Color-tile” SLAM

- Robot lives in  $M \times N$  discrete grid:
  - Robot pose space =  $\{1, \dots, M\} \times \{1, \dots, N\}$
- Every grid-cell can be red or green
  - Map space =  $\{R, G\}^{MN}$
- Motion model: robot can try to move to any neighboring cell, and succeeds with probability  $a$ , stays in place with probability  $1-a$ .
- Sensor model: robot can measure the color of the cell it is currently on. Measurement is correct with probability  $b$ , incorrect with probability  $1-b$ .



# “Color-tile” SLAM

- Challenge in running the Rao-Blackwellized Particle Filter:

efficiently evaluate  $w_t^i = \int p(z_t | x_t^i, m_t) p(m_t | z_{1:t-1}, x_{1:t-1}^i) dm_t$

$$\begin{aligned} w_t^i &= \sum_m p(z_t | x_t^i, m) p(m | z_{1:t-1}, x_{1:t-1}^i) \\ &= \sum_m p(z_t | x_t^i, m^{x_t^i}) p(m | z_{1:t-1}, x_{1:t-1}^i) \\ &= \sum_{m^{x_t^i}} p(z_t | x_t^i, m^{x_t^i}) p(m^{x_t^i} | z_{1:t-1}, x_{1:t-1}^i) \end{aligned}$$

sensor model

posterior for the coloring of the cell the robot is currently at, which we can efficiently keep track of over time (mapping w/known poses)

Note: FastSLAM follows same derivation, difference being that (gridcell  $\rightarrow$  landmark), (gridcell color  $\rightarrow$  landmark location), (multinomial over color  $\rightarrow$  Gaussian over location)

# SLAM with Gridmaps

- Robot state  $(x, y, \theta)$
- Map space  $\{0, 1\}^{MN}$  where  $M$  and  $N$  is number of grid cells considered in  $X$  and  $Y$  direction
- Challenge in running the Rao-Blackwellized Particle Filter:

efficiently evaluate  $w_t^i = \int p(z_t | x_t^i, m_t) p(m_t | z_{1:t-1}, x_{1:t-1}^i) dm_t$

- Let  $m^{*i} = \arg \max_m p(m | z_{1:t-1}, x_{1:t-1}^i)$

then assuming a peaked posterior for the map, we have

$w_t^i \approx p(z_t | x_t^i, m^{*i})$  which is a sensor model evaluation