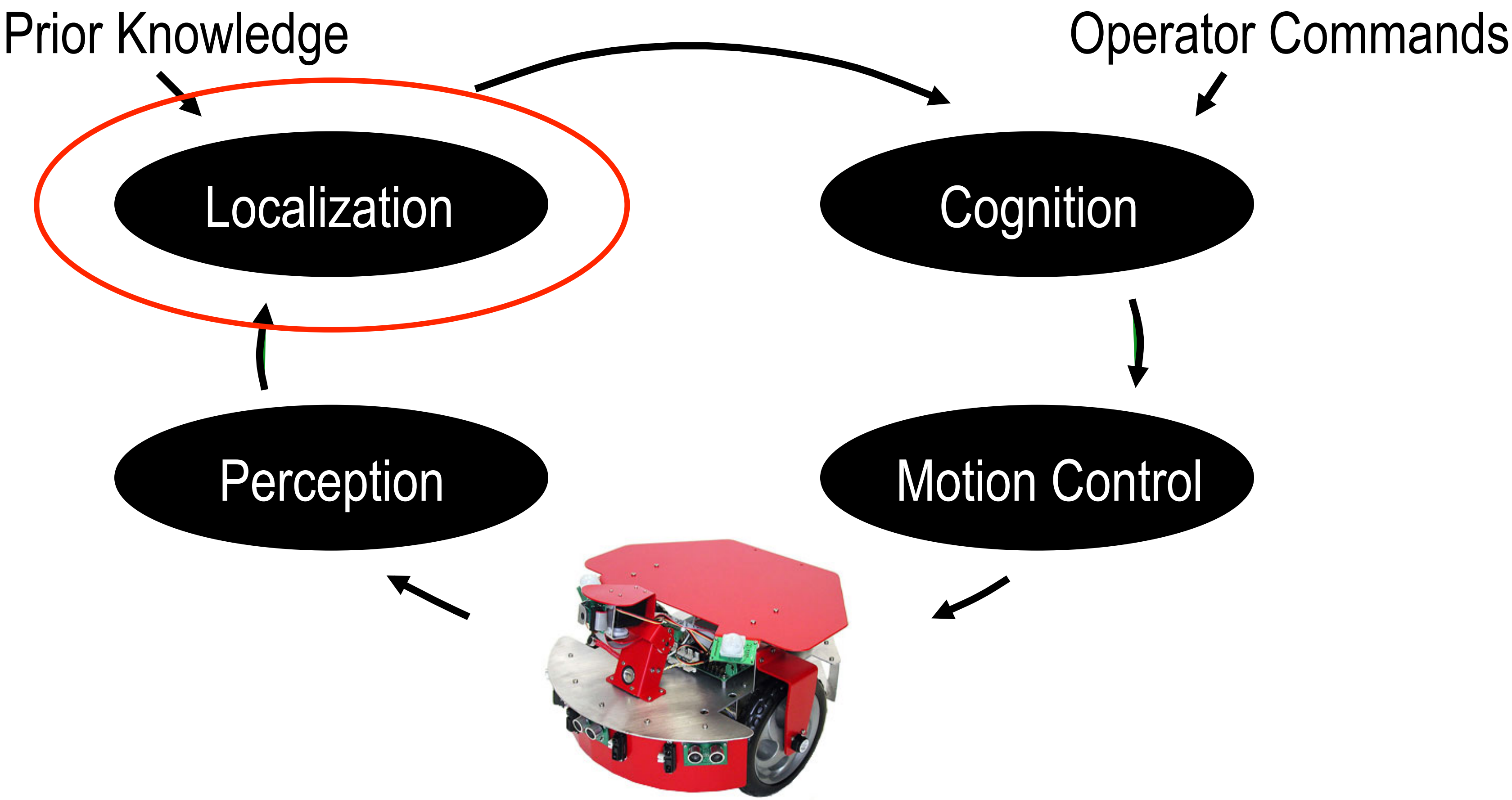
COS 495 - Lecture 15
Autonomous Robot Navigation

Instructor: Chris Clark

Semester: Fall 2011

Figures courtesy of Siegwart & Nourbakhsh

Control Structure



Particle Filter Localization: Outline

1. Particle Filters
    1. What are particles?
    2. Algorithm Overview
    3. Algorithm Example
2. PFL Application Example

What is a particle?

- Like Markov localization, Particle Filters represent the belief state with a set of possible states, and assigning a probability of being in each of the possible states.
- Unlike Markov localization, the set of possible states are not constructed by discretizing the configuration space, they are a randomly generated set of "particles".

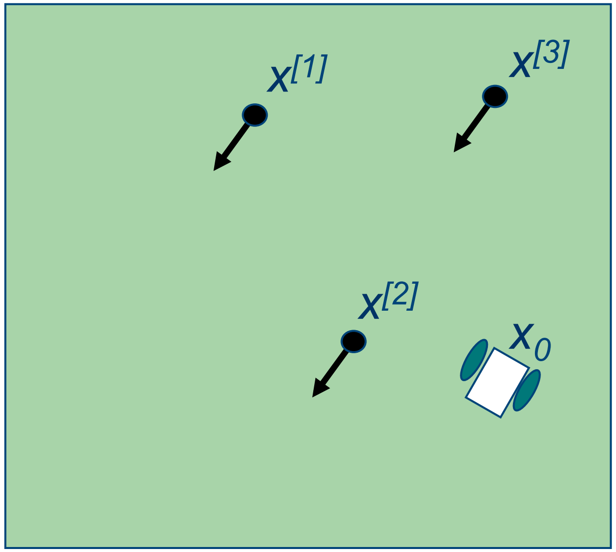
What is a particle?

- A particle is an individual state estimate.
- A particle is defined by its:
  1. State values that determine its location in the configuration space, e.g. *[x y θ]*
  2. A probability that indicates it's likelihood.
- Particle filters use many particles to for representing the belief state.

What is a particle?

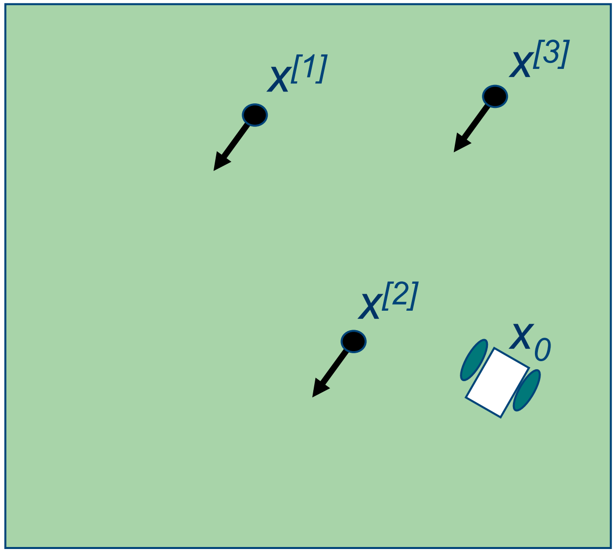
- Example:
  - A Particle filter uses 3 particles to represent the position of a (white) robot in a square room.
  - If the robot has a perfect compass, each particle is described as:

$$x^{[1]} = [x^1\ y^1]$$
$$x^{[2]} = [x^2\ y^2]$$
$$x^{[3]} = [x^3\ y^3]$$



What is a particle?

- Example:
  - Each of the particles $x^{[1]}$, $x^{[2]}$, $x^{[3]}$ also have associated weights $w^{[1]}$, $w^{[2]}$, $w^{[3]}$.
  - In the example below, $x^{[2]}$ should have the highest weight if the filter is working.



What is a particle?

- The user can choose how many particles to use:
  - More particles ensures a higher likelihood of converging to the correct belief state
  - Fewer particles may be necessary to ensure real-time implementation

Particle Filter Localization: Outline

1. Particle Filters
    1. What are particles?
    2. Algorithm Overview
    3. Algorithm Example
2. PFL Application Example

Markov Localization
Particle Filter

- Algorithm (Initialize at $t=0$):
  - Randomly draw N states in the work space and add them to the set X_0.
  - Iterate on these N states over time (see next slide).

Markov Localization
Particle Filter

- Algorithm (Loop over time step *t*):

1. For $i = 1 \ldots N$
2. Pick $x_{t-1}^{[i]}$ from X_{t-1}
3. Draw $x_t^{[i]}$ with probability $p(x_t^{[i]} \mid x_{t-1}^{[i]} , o_t)$
4. Calculate $w_t^{[i]} = p(z_t \mid x_t^{[i]})$
5. Add $x_t^{[i]}$ to X_t^{Temp}

Prediction

6. For $j = 1 \ldots N$
7. Draw $x_t^{[j]}$ from X_t^{Temp} with probability $w_t^{[j]}$
8. Add $x_t^{[j]}$ to X_t

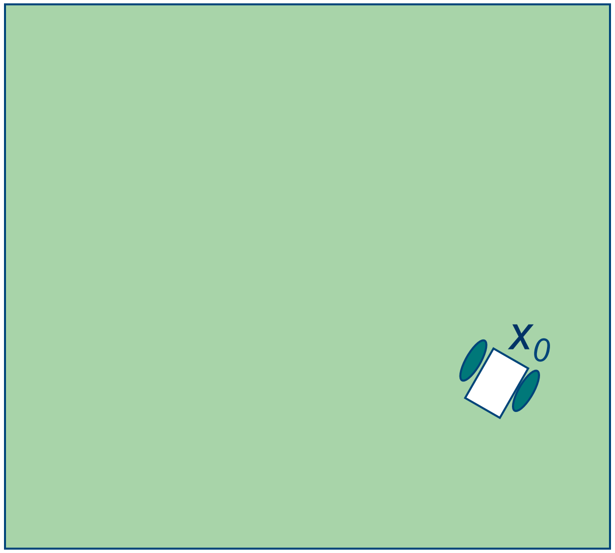
Correction

Particle Filter Localization: Outline

1. Particle Filters
    1. What are particles?
    2. Algorithm Overview
    3. Algorithm Example
2. PFL Application Example

Particle Filter Example

- Provided is an example where a robot (depicted below), starts at some unknown location in the bounded workspace.

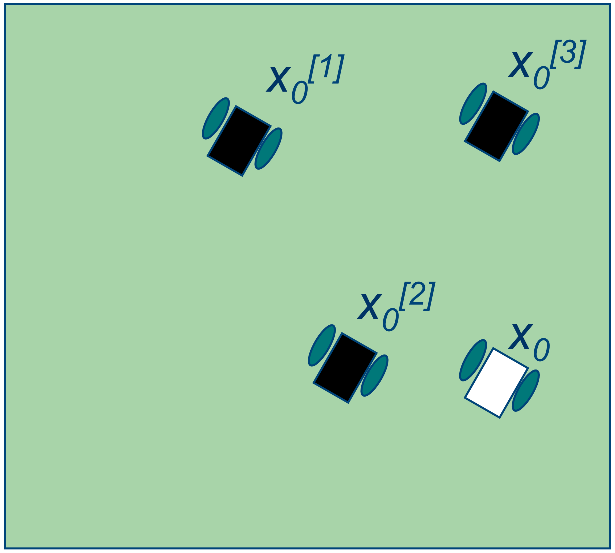


Particle Filter Example

- At time step t_0:
  - We randomly pick $N=3$ states represented as

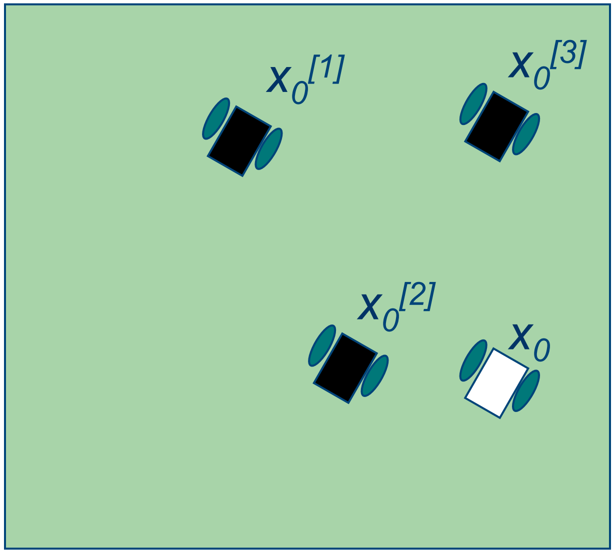
$$X_0 = \{x_0^{[1]}, x_0^{[2]}, x_0^{[3]}\}$$

  - For simplicity, assume known heading



Particle Filter Example

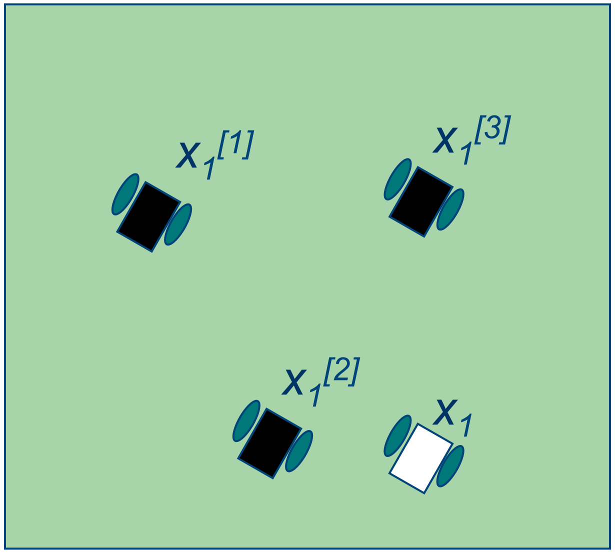
- The next few slides provide an example of one iteration of the algorithm, given X_0.
  - This iteration is for time step t_1.
  - The inputs are the measurement z_1, odometry o_1



Particle Filter Example

- For Time step t_1:
  - Randomly generate new states by propagating previous states X_0 with o_1

$$X_1^{Temp} = \{x_1^{[1]}, x_1^{[2]}, x_1^{[3]}\}$$



Particle Filter Example

- For Time step t_1:
  - To get new states, use the motion model from lecture 3 to randomly generate new state $x_1^{[i]}$.
  - Recall that given some Δs_r and Δs_l we can calculate the robot state in global coordinates:

$$\Delta x = \Delta s \cos(\theta + \Delta\theta / 2)$$

$$\Delta y = \Delta s \sin(\theta + \Delta\theta / 2)$$

$$\Delta\theta = \frac{\Delta s_r - \Delta s_l}{b}$$

$$\Delta s = \frac{\Delta s_r + \Delta s_l}{2}$$

Particle Filter Example

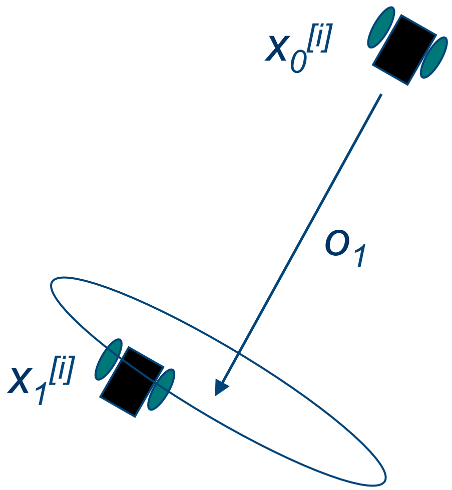
- For Time step t_1:
  - So, if you add some random errors ε_r and ε_l to Δs_r and Δs_l, you can generate a new random state that follows the probability distribution dictated by the motion model.
  - So, in the prediction step of the PF, the i^{th} particle can be randomly propagated forward using measured odometry $o_1 = \{\Delta s_r, \Delta s_l\}$ according to:

$$\Delta s_r^{[i]} = \text{rand}(\text{'norm'}, \Delta s_r, \sigma_s)$$

$$\Delta s_l^{[i]} = \text{rand}(\text{'norm'}, \Delta s_l, \sigma_s)$$

Particle Filter Example

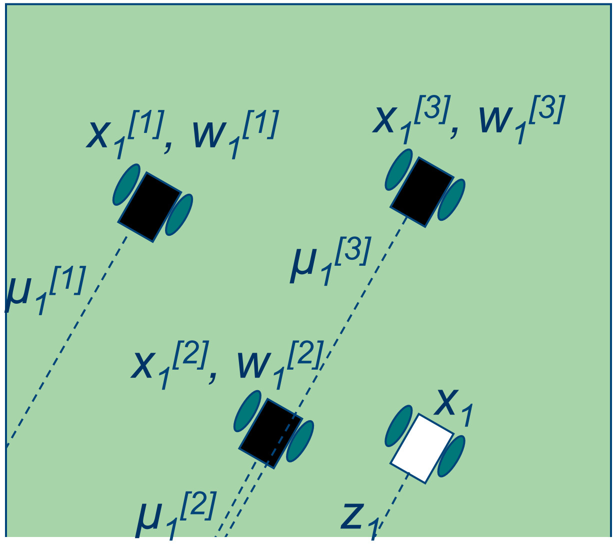
- For Time step t_1:
  - For example:



Particle Filter Example

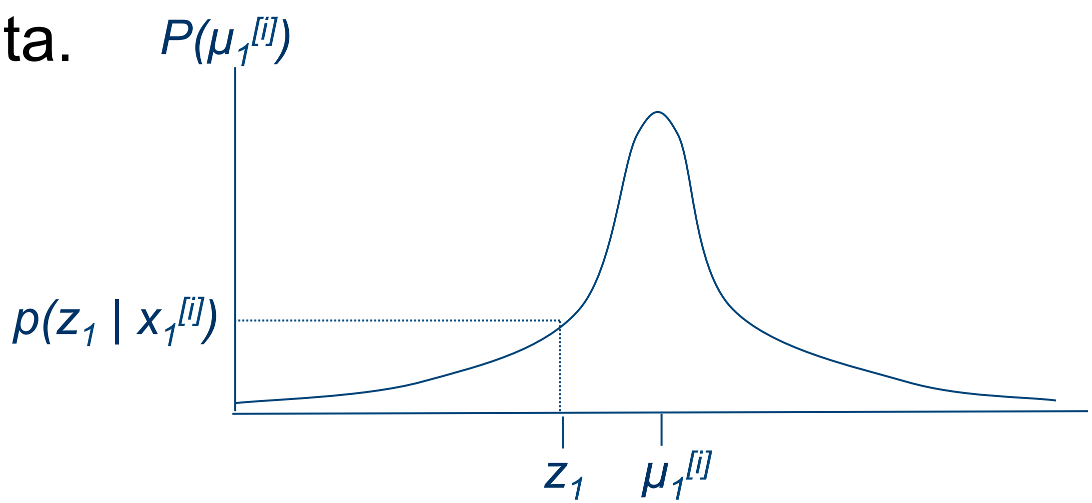
- For Time step t_1:
  - Using the measurement z_1, calculate the expected weights $w^{[i]} = p(z_1 \mid x_1^{[i]})$ for each state.

$$W_1 = \{w_1^{[1]}, w_1^{[2]}, w_1^{[3]}\}$$



Particle Filter Example

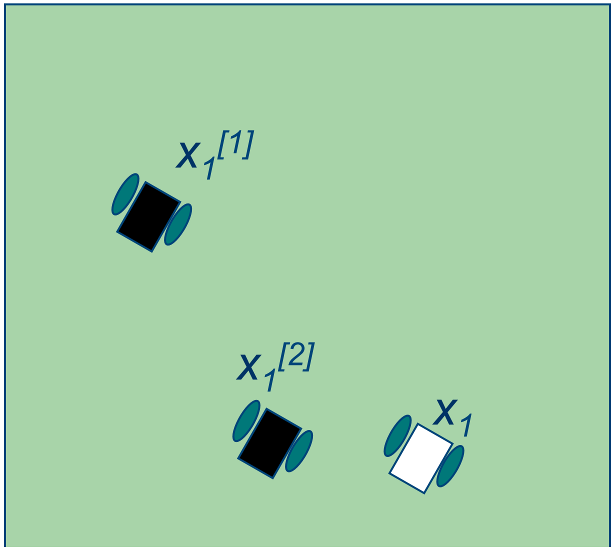
- For Time step t_1:
  - To calculate $p(z_1 \mid x_1^{[i]})$ we use the sensor probability distribution of a single Gaussian of mean $\mu_1^{[i]}$ that is the expected range for the particle
  - The gaussian variance can be taken from sensor data.



Particle Filter Example

- For Time step t_1:
  - Resample from the temporary state distribution based on the weights $w_1^{[2]} > w_1^{[1]} > w_1^{[3]}$

$$X_1 = \{x_1^{[2]}, x_1^{[2]}, x_1^{[1]}\}$$

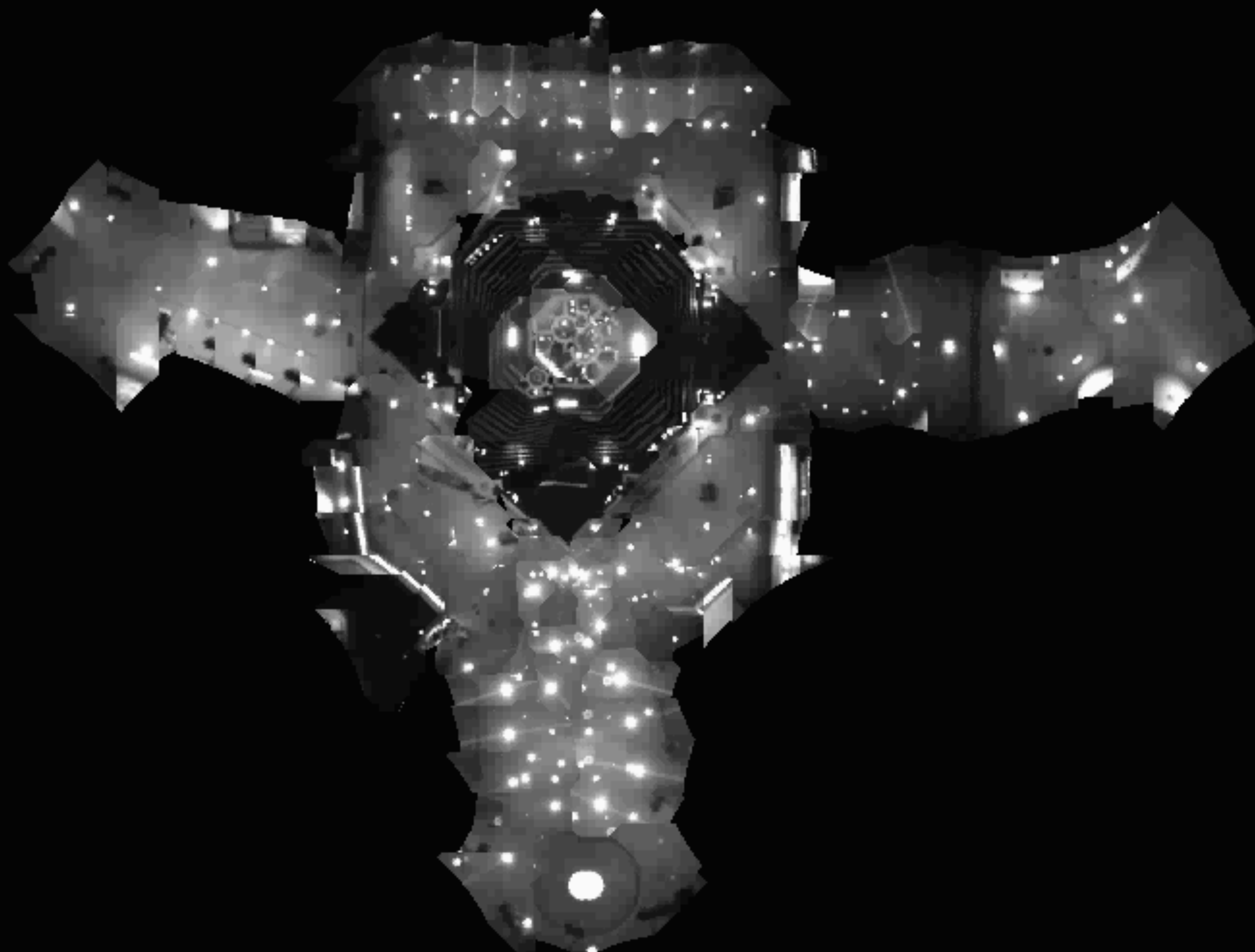


Particle Filter Example

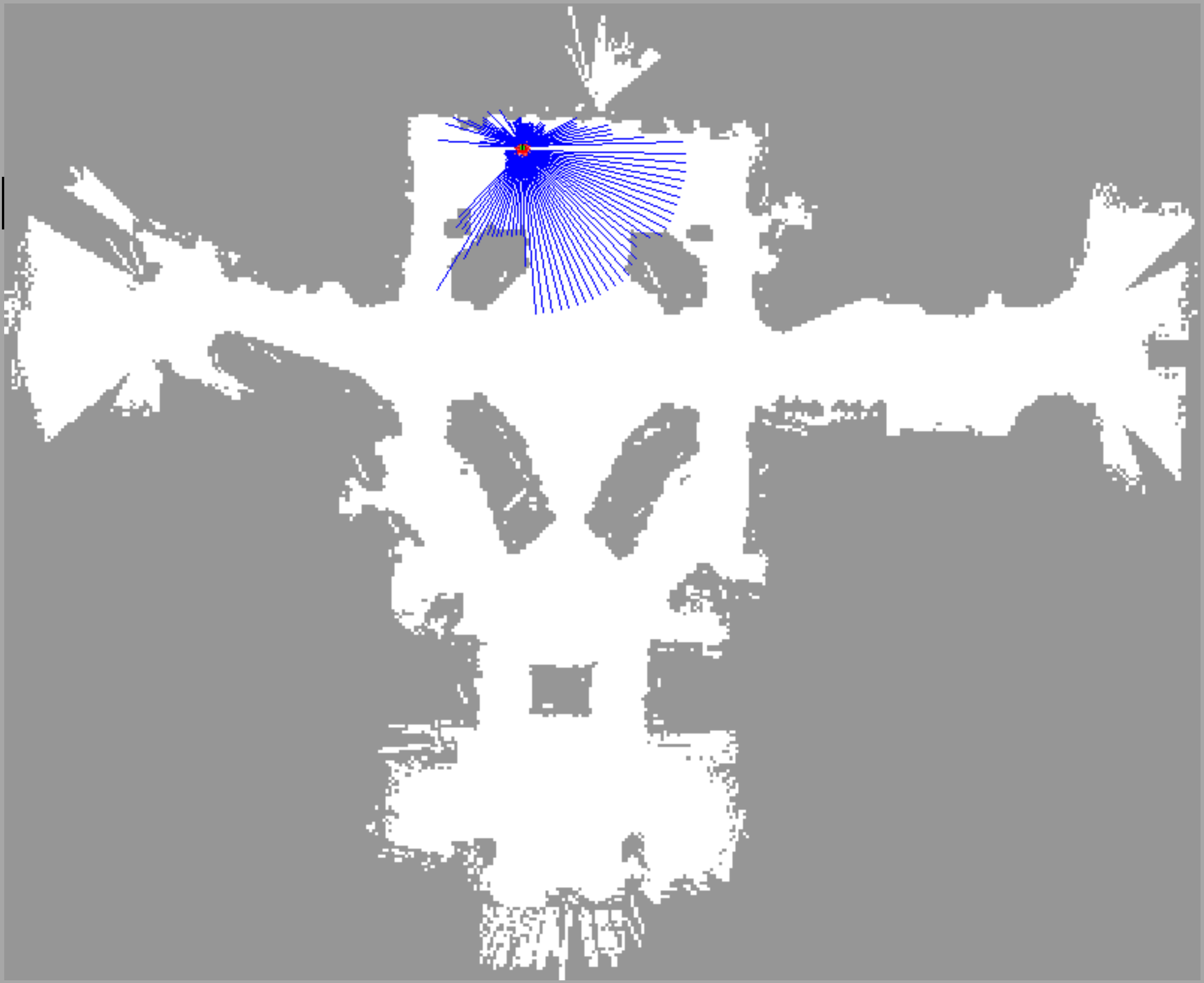
- For Time step t_2:
  - Iterate on previous steps to update state belief at time step t_2 given (X_1, o_2, z_2).

Particle Filter Localization: Outline

1. Particle Filters
    1. What are particles?
    2. Algorithm Overview
    3. Algorithm Example
2. PFL Application Example



Courtesy of S. Thrun



Courtesy of S. Thrun