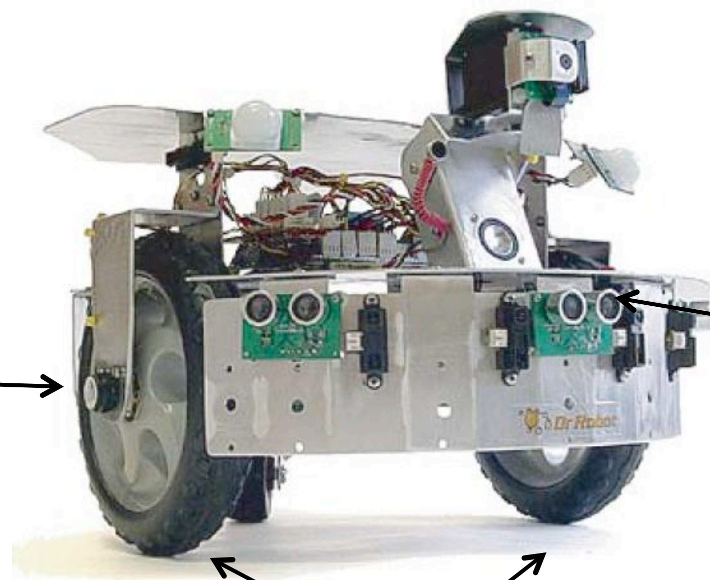
ARW – Lecture 02
Localization

Instructor: Chris Clark

Semester: Summer 2016

Figures courtesy of Siegwart & Nourbakhsh

Different Bots

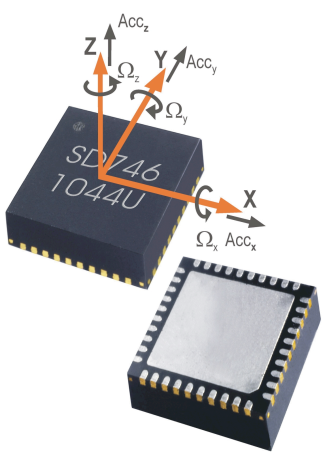


Wheel Encoders

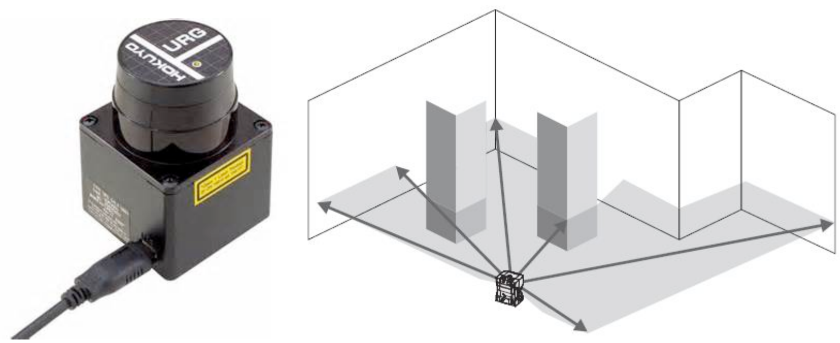
Range Sensors

Differential Drive Motor Configuration

Different Sensors

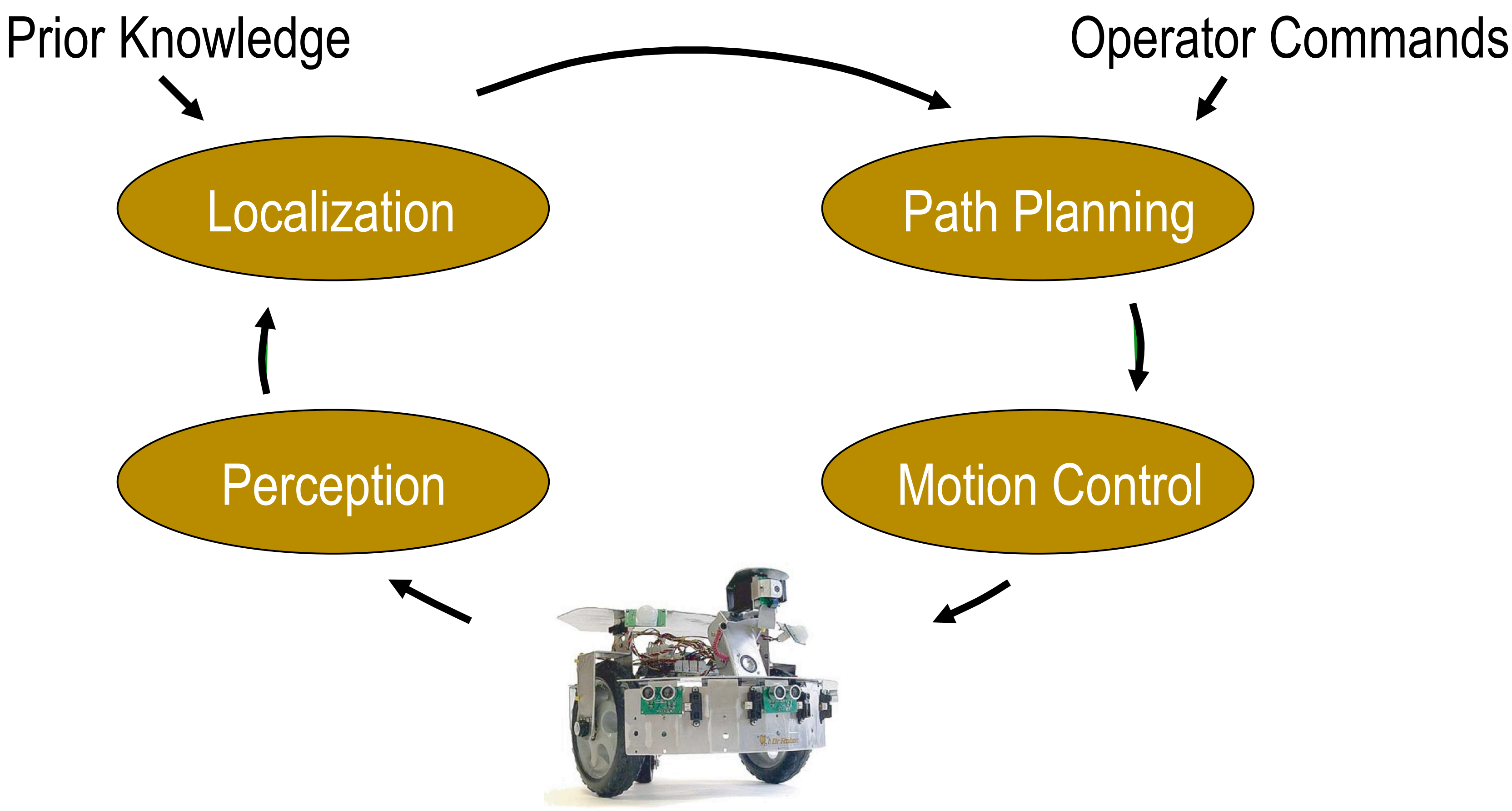


Proprioceptive Sensors

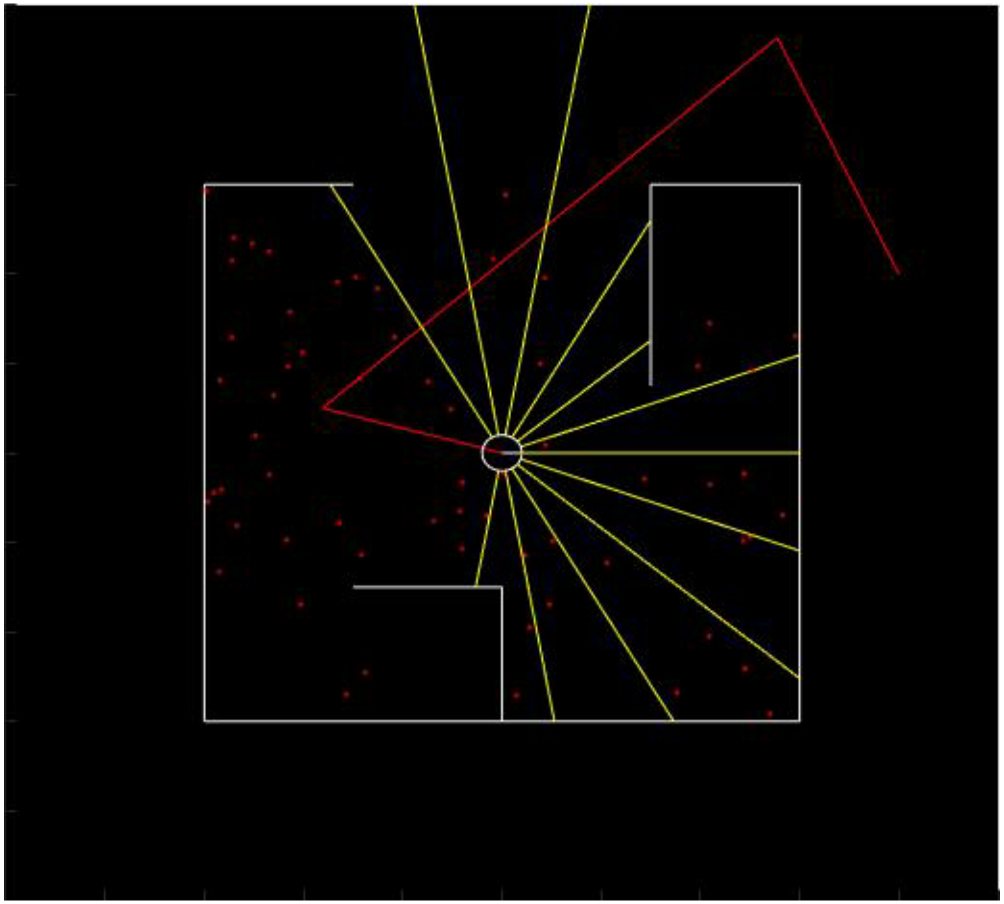


Range Sensors

Planning Based Control



ARW Goals



Odometry Kinematics

- Lecture Goal
  - Present an algorithm that estimates the current robot state given the previous estimated state, encoder measurements, and a map.

$$X_t = f(X_{t-1}, U_{t-1}, M)$$

Outline - Localization

1. Localization Tools
2. Overview of Algorithms
   - Typical Methods
   - Basic Structure
3. Markov Localization

Methods

Methods

- Strategy:
  - It might start to move from a known location, and keep track of its position using odometry.
  - However, the more it moves the greater the uncertainty in its position.
  - Therefore, it will update its position estimate using observation of its environment

Methods

- Method:
  - Fuse the odometric position estimate with the observation estimate to get best possible update of actual position

- This can be implemented with two main functions:
  1. Act
  2. See

Methods

- Action Update (Prediction)
  - Define function to predict position estimate based on previous state x_{t-1} and encoder measurement o_t or control inputs u_t

$$x'_t = Act\ (o_t\ ,\ x_{t-1})$$

  - Increases uncertainty

Methods

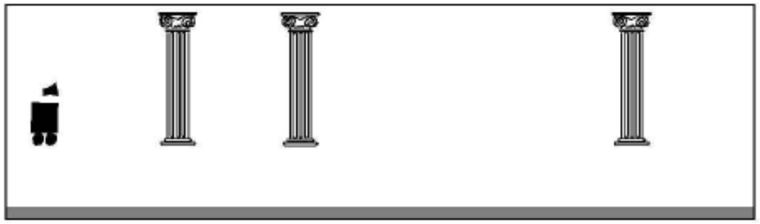
- Perception Update (Correction)
  - Define function to correct position estimate x'_t using exteroceptive sensor inputs z_t

$$x_t = See\ (z_t\ ,\ x'_t)$$

  - Decreases uncertainty

Methods

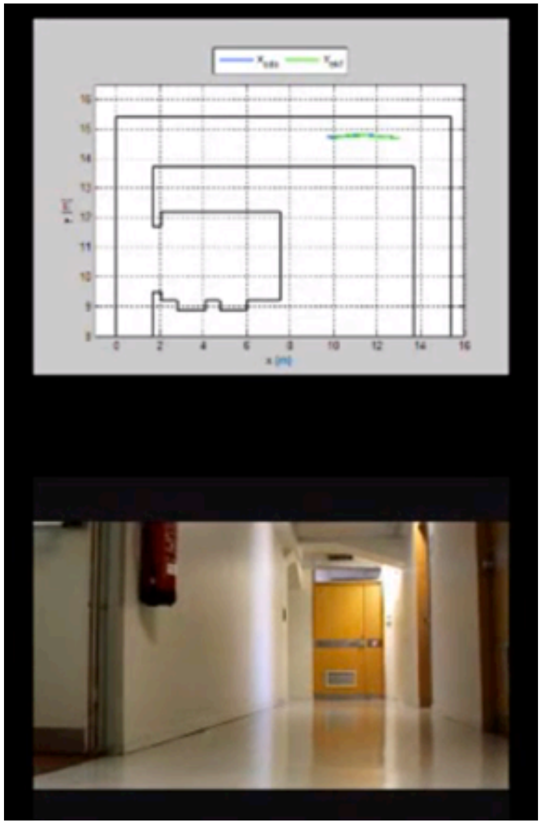
- Motion generally improves the position estimate.



Kalman Filtering vs. Monte Carlo

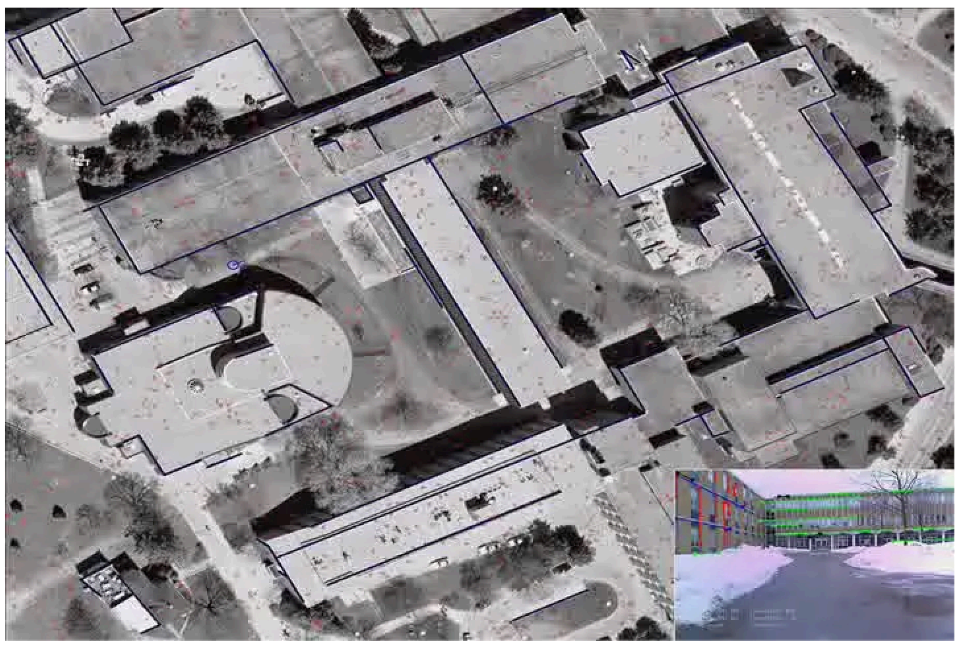
- MC Localization
  - Can localize from **any unknown** position in map
  - **Recovers** from ambiguous situation
  - However, to update the probability of all positions within the whole state space requires discrete representation of space. This can require large amounts of **memory** and **processing** power.
- Kalman Filter Localization
  - Tracks the robot and is inherently **precise** and **efficient**
  - However, if uncertainty grows too large, the KF will fail and the robot will get **lost**.

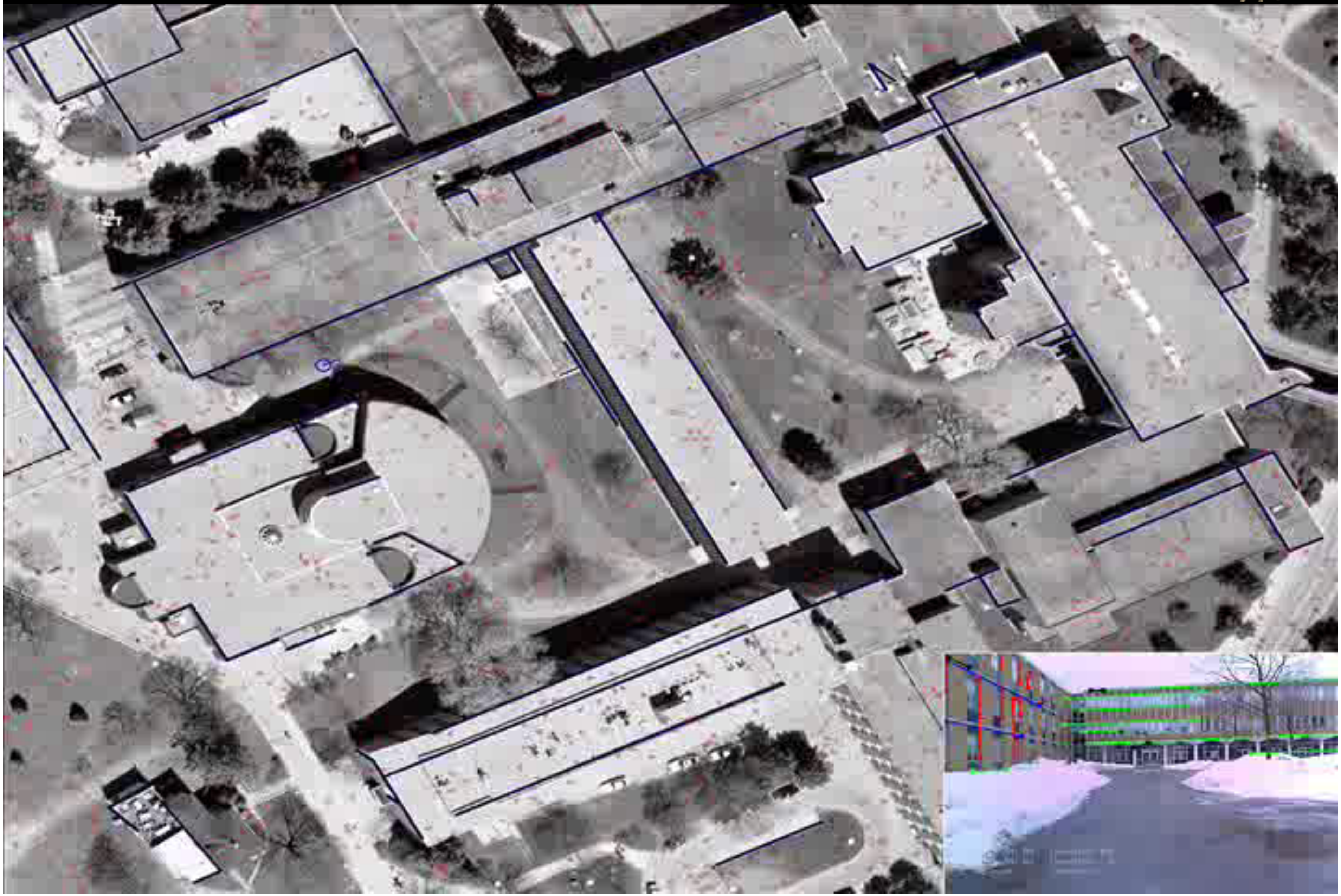
Kalman Filtering



http://www.youtube.com/watch?v=AXGXfD1GMY4

Particle Filter Localization





Particle Filter Localization

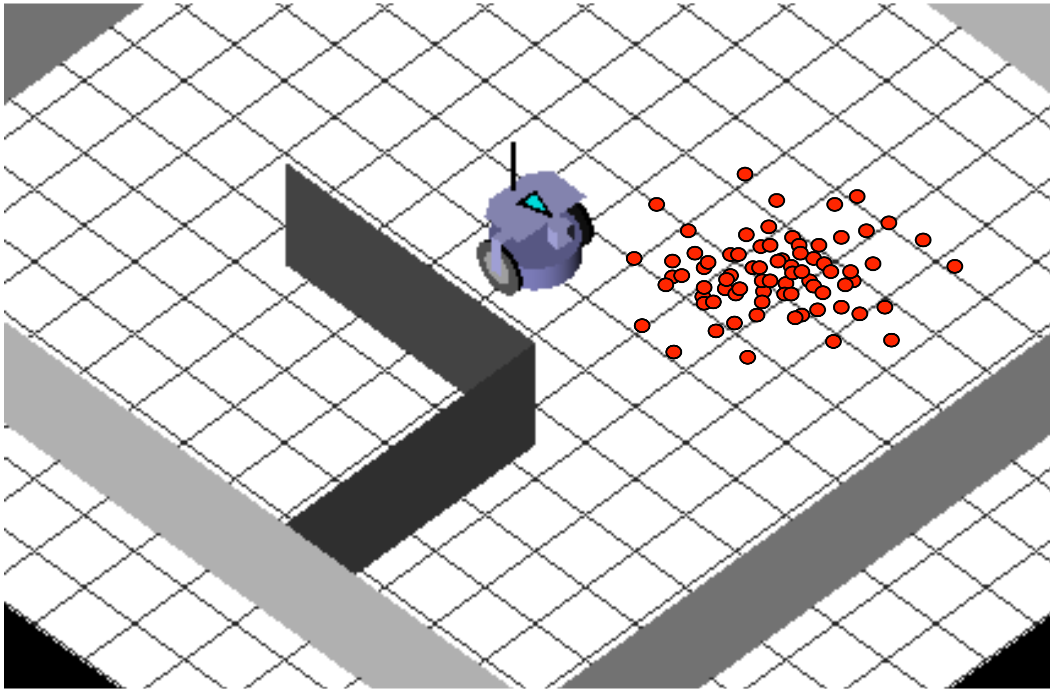
1. Particle Filters
   1. What are particles?
   2. Algorithm Overview
   3. Algorithm Example
   4. Using the particles
2. PFL Application Example

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. $\mathbf{x} = [\, x \; y \; \theta \,]$
  2. A probability that indicates it's likelihood.

What is a particle?

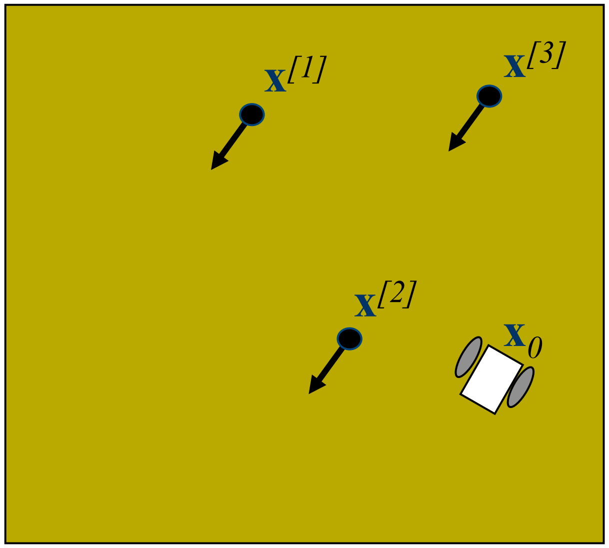
- Particle filters use many particles to for representing the belief state.



What is a particle?

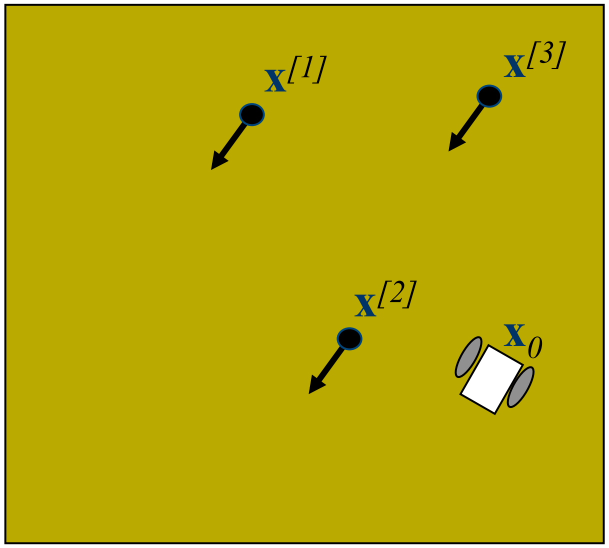
- Example:
  - A Particle filter uses *3* particles to represent the position of a (white) robot in a square room.
  - If the robot has a perfect compass, each particle is described as:

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



What is a particle?

- Example:
  - Each of the particles $\mathbf{x}^{[1]}$, $\mathbf{x}^{[2]}$, $\mathbf{x}^{[3]}$ also have associated weights $w^{[1]}$, $w^{[2]}$, $w^{[3]}$.
  - In the example below, $\mathbf{x}^{[2]}$ 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

1. Particle Filters
   1. What are particles?
   2. Algorithm Overview
   3. Algorithm Example
   4. Using the particles
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 $\mathbf{X}_0$.

$$\mathbf{X}_0 = \{\mathbf{x}_0^{[1]},\ \mathbf{x}_0^{[2]},\ \ldots,\ \mathbf{x}_0^{[N]}\}$$

  - Iterate on these N states over time (see next slide).

Markov Localization
Particle Filter

- Algorithm (Loop over time step t):

1. For $i = 1 \dots N$
2. Pick $\mathbf{x}_{t-1}^{[i]}$ from $\mathbf{X}_{t-1}$
3. Draw $\mathbf{x}_t^{[i]}$ with probability $P(\mathbf{x}_t^{[i]} \mid \mathbf{x}_{t-1}^{[i]} , o_t)$
4. Calculate $w_t^{[i]} = P(z_t \mid \mathbf{x}_t^{[i]})$
5. Add $\mathbf{x}_t^{[i]}$ to $\mathbf{X}_t^{Predict}$

Prediction

6. For $j = 1 \dots N$
7. Draw $\mathbf{x}_t^{[j]}$ from $\mathbf{X}_t^{Predict}$ with probability $w_t^{[j]}$
8. Add $\mathbf{x}_t^{[j]}$ to $\mathbf{X}_t$

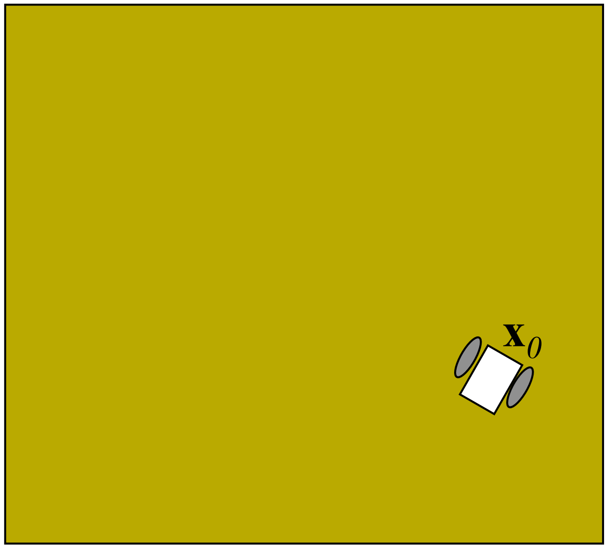
Correction

Particle Filter Localization

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

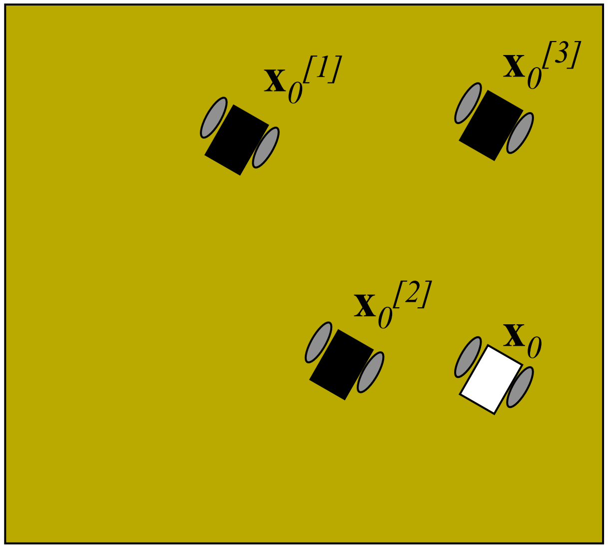
Particle Filter Example

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



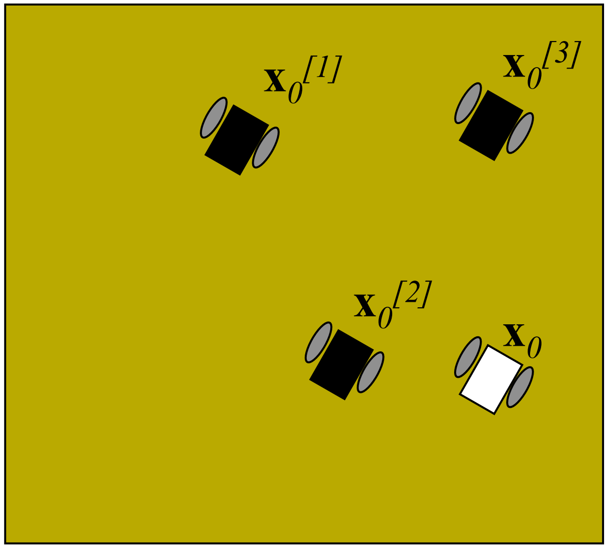
Particle Filter Example

- At time step t_0:
  - We randomly pick $N=3$ states represented as
    $$\mathbf{X}_0 = \{\mathbf{x}_0^{[1]}, \mathbf{x}_0^{[2]}, \mathbf{x}_0^{[3]}\}$$
  - For simplicity, assume known heading



Particle Filter Example

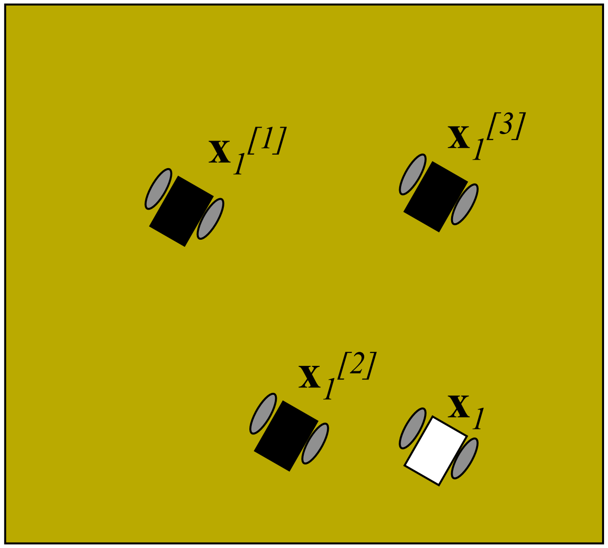
- The next few slides provide an example of one iteration of the algorithm, given $\mathbf{X}_0$.
  - This iteration is for time step t_1.
  - The inputs are the measurement z_1, odometry o_1



Particle Filter Example

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

$$\mathbf{X}_1^{\,Predict} = \{\mathbf{x}_1^{[1]}, \mathbf{x}_1^{[2]}, \mathbf{x}_1^{[3]}\}$$



Particle Filter Example

- For Time step t_1:
  - To get new states, use the motion model from lecture *3* to randomly generate new state $\mathbf{x}_1^{[i]}$.
  - 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

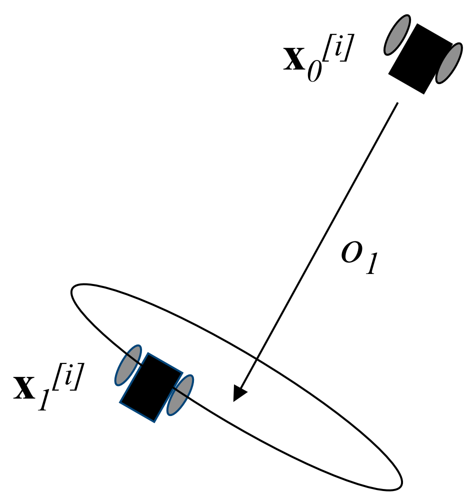
- For Time step t_1:
  - If you add some random errors ε_r and ε_l to Δs_r and Δs_l, you can generate a new random state that follows the probability distribution dictated by the motion model.
  - So, in the prediction step of the PF, the i^{th} particle can be randomly propagated forward using measured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

$$\Delta s_r^{[i]} = \Delta s_r + \text{rand}(\text{'norm'}, 0, \sigma_s)$$

$$\Delta s_l^{[i]} = \Delta s_l + \text{rand}(\text{'norm'}, 0, \sigma_s)$$

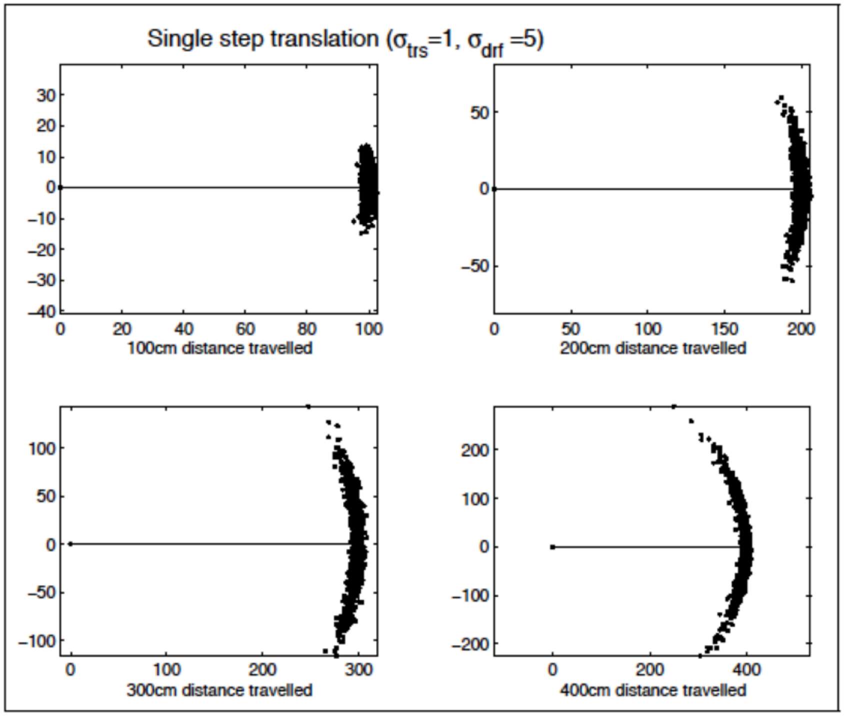
Particle Filter Example

- For Time step t_1:
  - For example:



Particle Filter Example

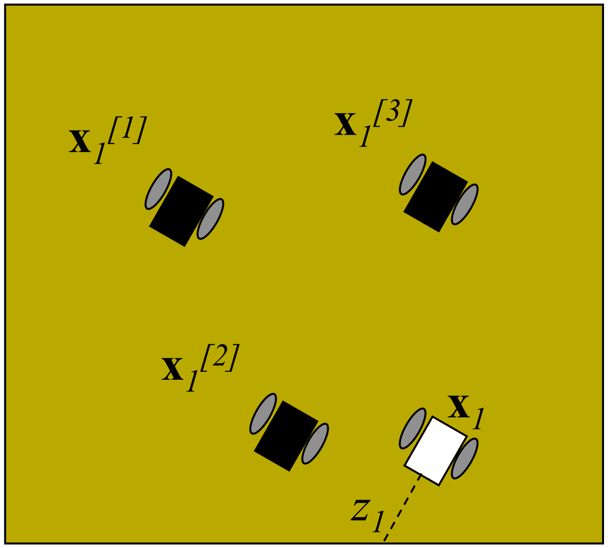
- Example Prediction Steps



Yiannis, McGill University, PF Tutorial

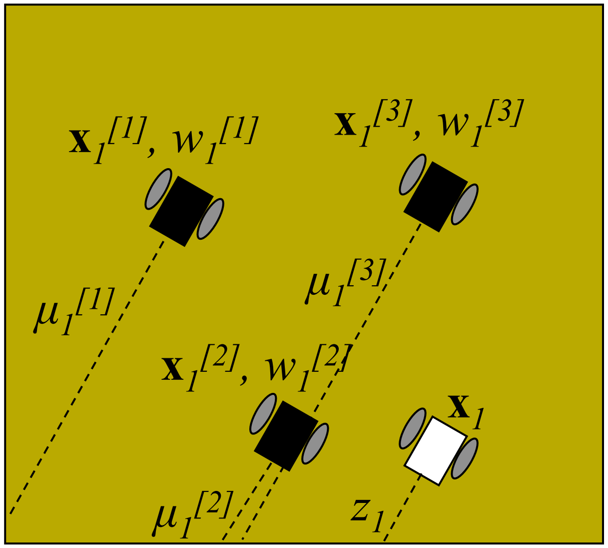
Particle Filter Example

- For Time step t_1:
  - We get a new measurement z_1, e.g. a forward facing range measurement.



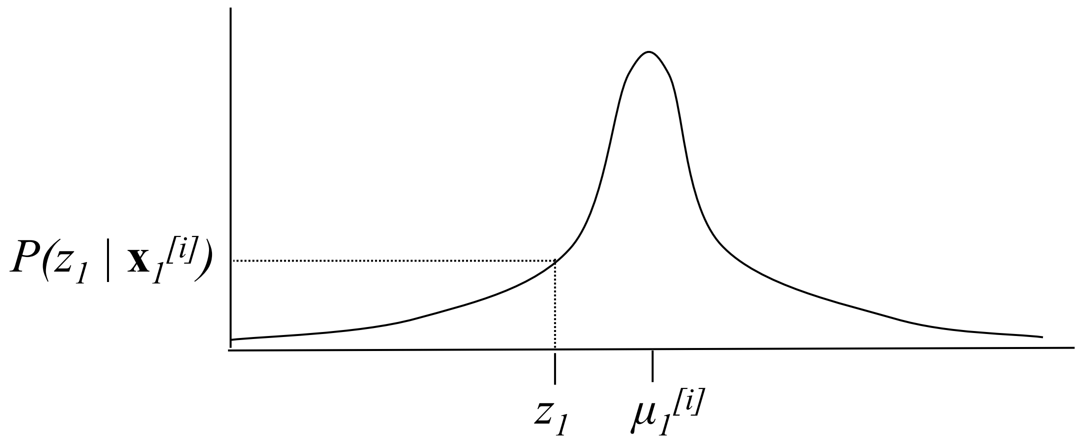
Particle Filter Example

- For Time step t_1:
  - Using the measurement z_1, and expected measurements $\mu_1^{[i]}$, calculate the weights $w^{[i]} = P(z_1 \mid \mathbf{x}_1^{[i]})$ for each state.



Particle Filter Example

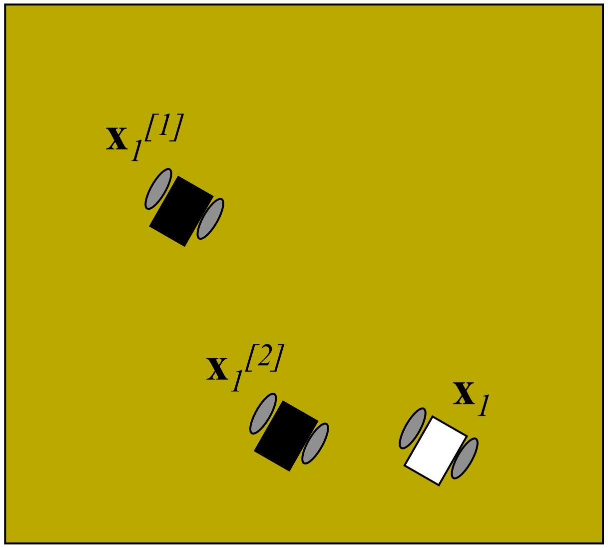
- For Time step t_1:
  - To calculate $P(z_1 \mid \mathbf{x}_1^{[i]})$ we use the sensor probability distribution of a single Gaussian of mean $\mu_1^{[i]}$ that is the expected range for the particle
  - The Gaussian variance is obtained from experiment.



Particle Filter Example

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

$$\mathbf{X}_1 = \{\mathbf{x}_1^{[2]}, \mathbf{x}_1^{[2]}, \mathbf{x}_1^{[1]}\}$$



Particle Filter Example

- For Time step t_1:
  - How do we resample?
    - Exact Method
    - Approximate Method
    - Others…

Particle Filter Example

- An Exact Method

$w_{tot} = \sum_j w_j$

for $i=1..N$

$r = rand(\text{'uniform'}) * w_{tot}$

$j = 1$

$w_{sum} = w_1$

$while\ (w_{sum} < r)$

$j = j+1$

$w_{sum} = w_{sum} + w_j$

$\mathbf{x}_i = \mathbf{x}_j^{Predict}$

Particle Filter Example

- An Approximate Method

$w_{tot} = max_j \ w_j$

for $i = 1..N$

$w_i = w_i / w_{tot}$

if $w_i < 0.25$

add 1 copy of $\mathbf{x}_i^{Predict}$ to $\mathbf{X}^{TEMP}$

else if $w_i < 0.50$

add 2 copies of $\mathbf{x}_i^{Predict}$ to $\mathbf{X}^{TEMP}$

else if $w_i < 0.75$

add 3 copies of $\mathbf{x}_i^{Predict}$ to $\mathbf{X}^{TEMP}$

else if $w_i < 1.00$

add 4 copies of $\mathbf{x}_i^{Predict}$ to $\mathbf{X}^{TEMP}$

Particle Filter Example

- An Approximate Method (cont')

for $i = 1..N$

$r = (int)\ rand('uniform')*size(\mathbf{X}^{TEMP})$

$\mathbf{x}_i = \mathbf{x}_r^{TEMP}$

Particle Filter Example

- NOTE:

  We should only resample when we get NEW measurements.

Particle Filter Example

- For Time step t_2:
  - Iterate on previous steps to update state belief at time step t_2 given $(\mathbf{X}_1, o_2, z_2)$.

Particle Filter Localization

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

Additional Notes

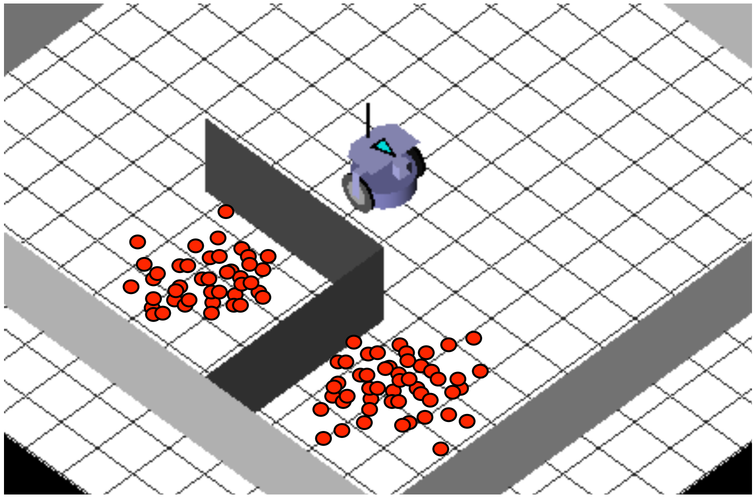
- How do we use the belief?
  - To control the robot, we often distill the belief into a lower dimension representation.
  - Examples:

$$\hat{\mathbf{x}}_1 = \frac{\sum_i w_1^{[i]}\, \mathbf{x}_1^{[i]}}{\sum_i w_1^{[i]}}$$

$$\hat{\mathbf{x}}_1 = \{\, \mathbf{x}_1^{[i]} \mid w_1^{[i]} > w_1^{[j]} \;\; \forall\, j \neq i \,\}$$

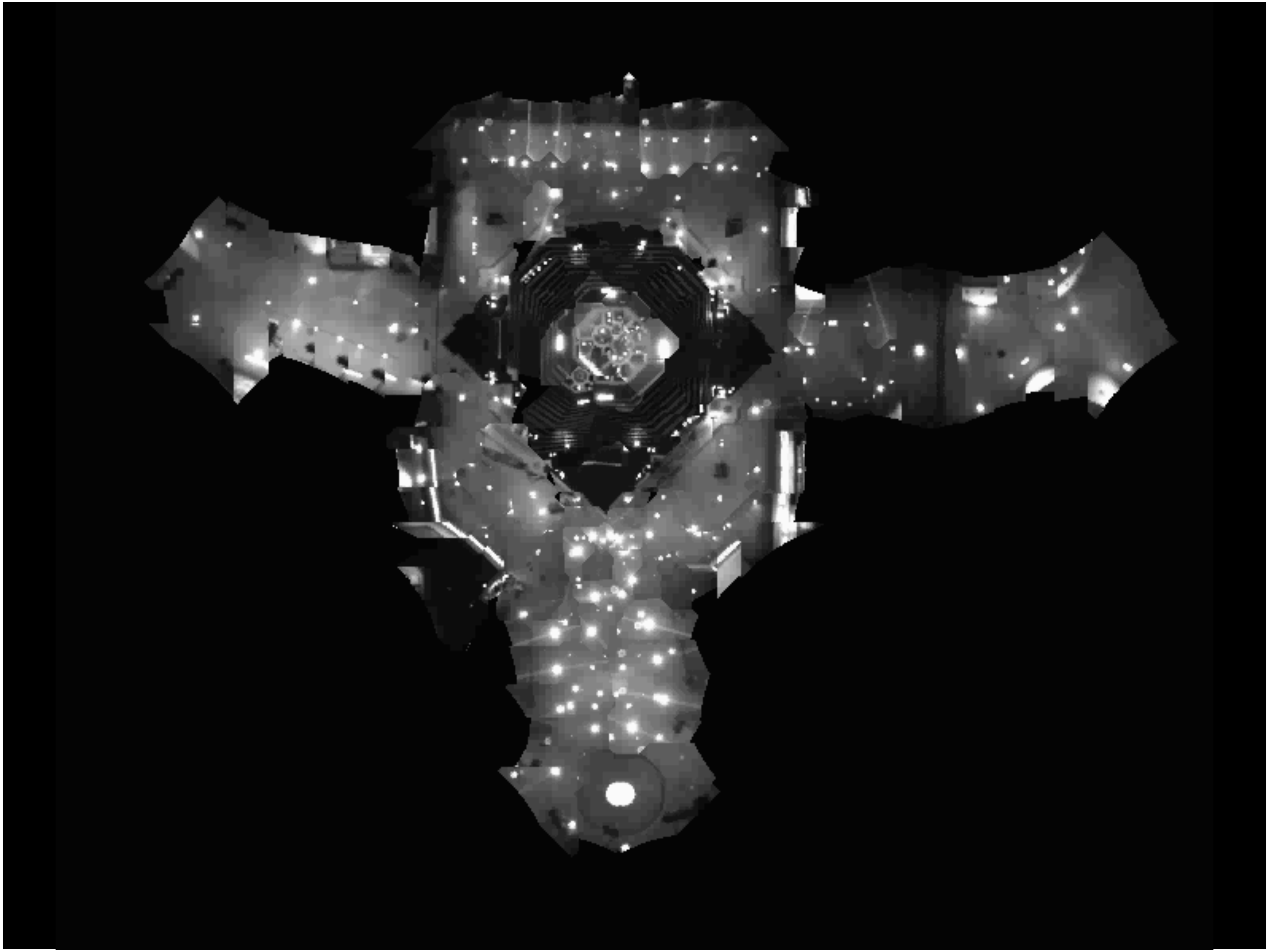
Additional Notes

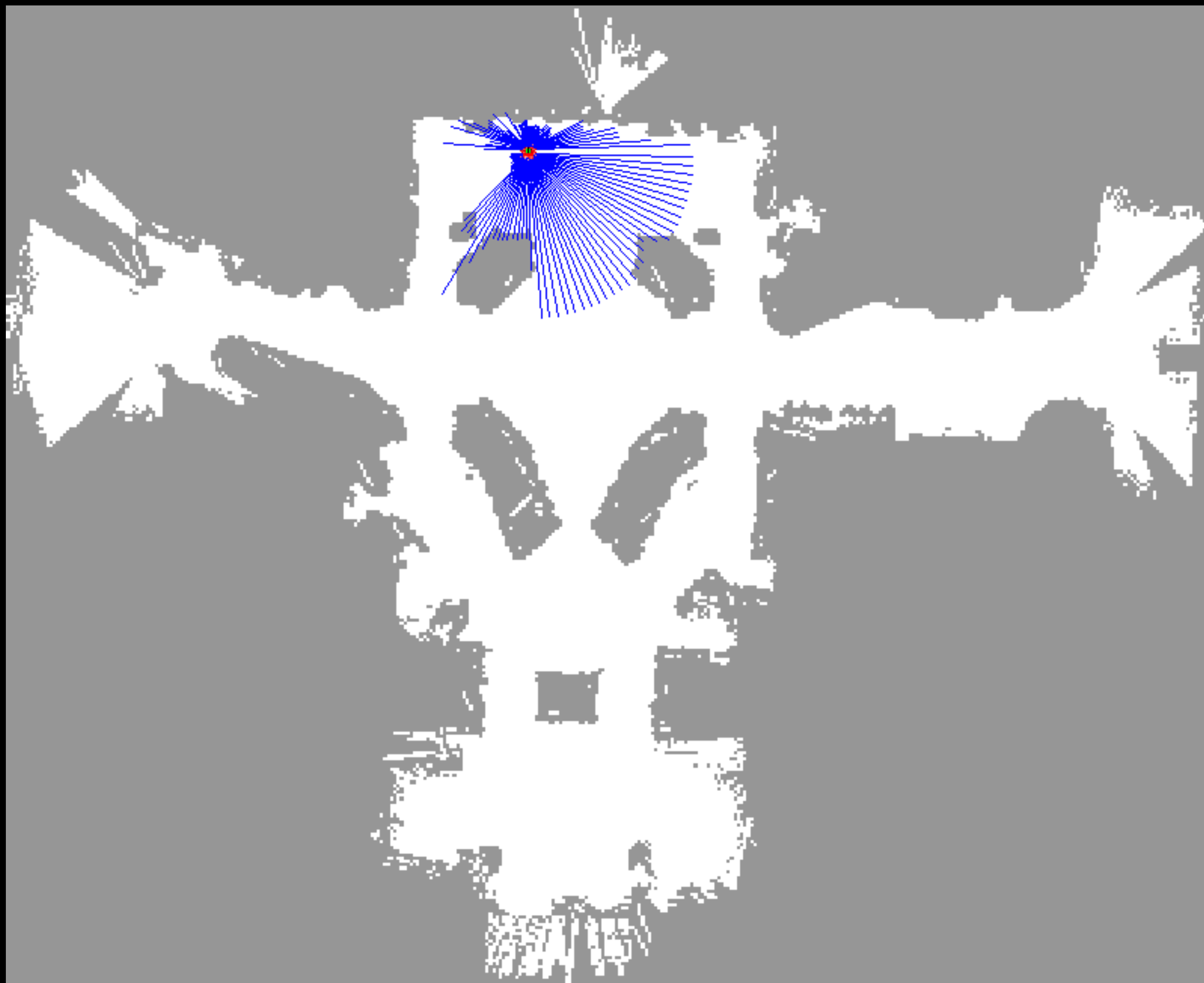
- How do we use the belief?
  - Sometimes we have several clusters
  - There are clustering algorithms to help …



Particle Filter Localization: Outline

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





Courtesy of S. Thrun