AI Robotics

Particle Filter

Overview

Localization

Sensor Measurements:

$Z_{0:t} = \{z_0, \ldots, z_t\}$

Robot State (or pose):

$X_{0:t} = \{X_0, \ldots, X_t\}$

sense

Known

Unknown

act

Robot Controls:

$u_{0:t} = \{u_0, u_1, \ldots, u_t\}$

Probabilistic State Estimation

$$P(x_t|u_1, z_1, u_2, z_2, \cdots, u_t, z_t) = P(x_t|z_{1:t}, u_{1:t})$$

Robot State

sense

Sensor Model

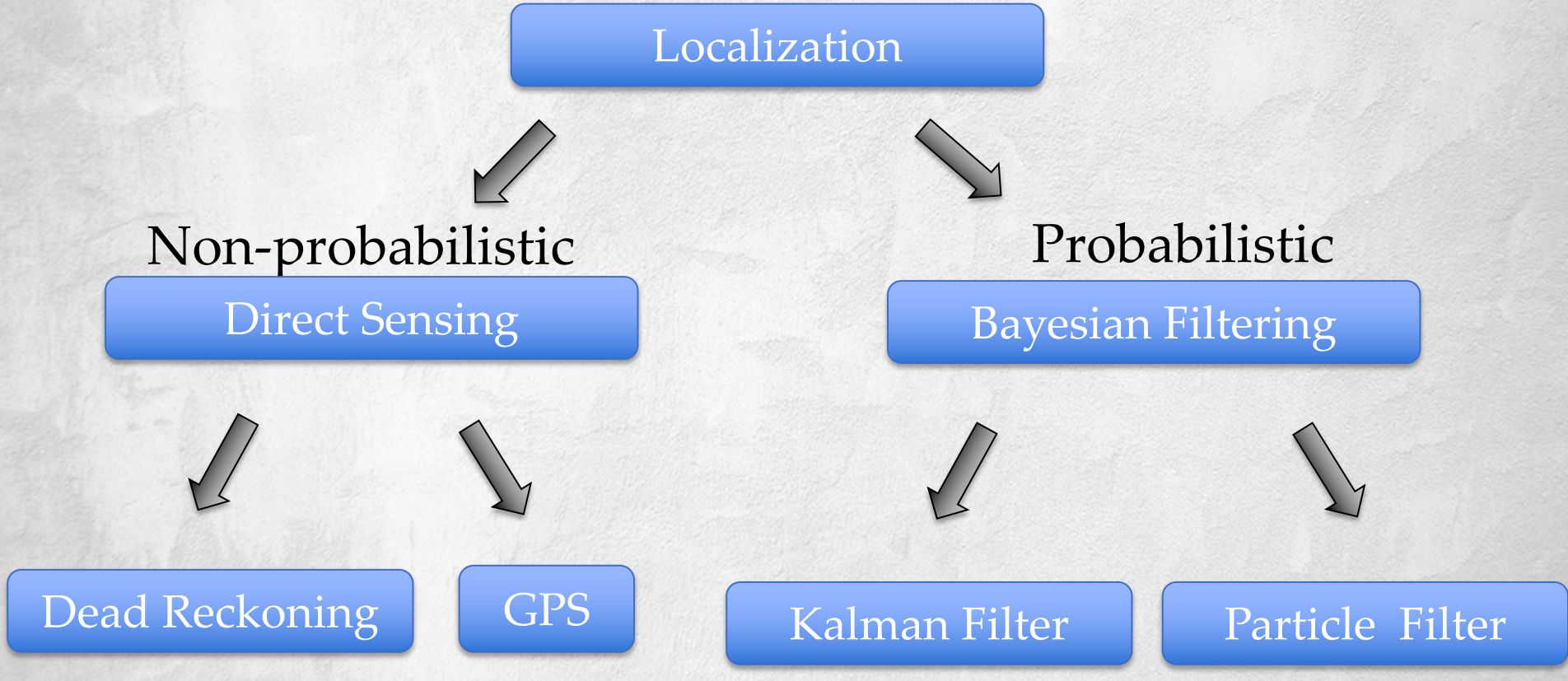
$$P(z_t|x_t)$$

act

Motion Model

$$P(x_t|x_{t-1}, u_t)$$

Types of Localization Algorithms



Particle Filter

Bayes Filter Algorithm

$$Bel(x_t) = \eta\ P(z_t \mid x_t) \int P(x_t \mid u_t, x_{t-1})\ Bel(x_{t-1})\ dx_{t-1}$$

Previous State

Current control input

Current sensor measurement

Algorithm **Bayes_filter**($Bel(x_{t-1})$, u_t, z_t):

$$Bel'(x_t) = \int P(x_t \mid u_t, x_{t-1})\ Bel\ (x_{t-1})dx$$

Prediction Step: Motion Update

$$Bel(x_t) = \eta\ P(z_t \mid x_t)\ Bel'(x_t)$$

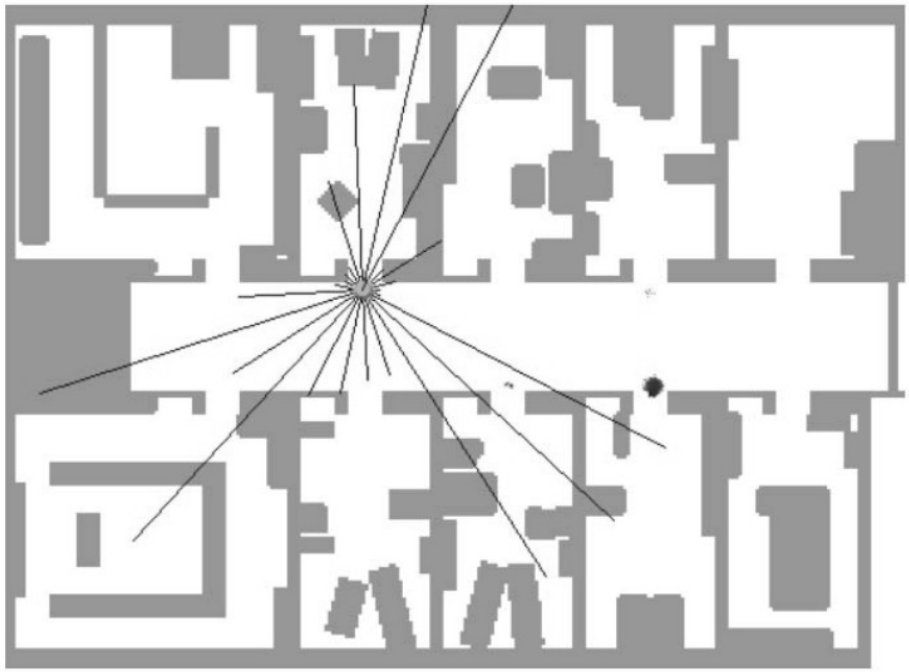
Correction Step: Sensor Update

$return \quad Bel(x_t)$

Current State

The Problem with Kalman Filters in Robot Localization

- Kalman Filters only represent state variables as single Gaussians
- What if robot could be in one of two places?



Particle Filters (aka sequential Monte Carlo)

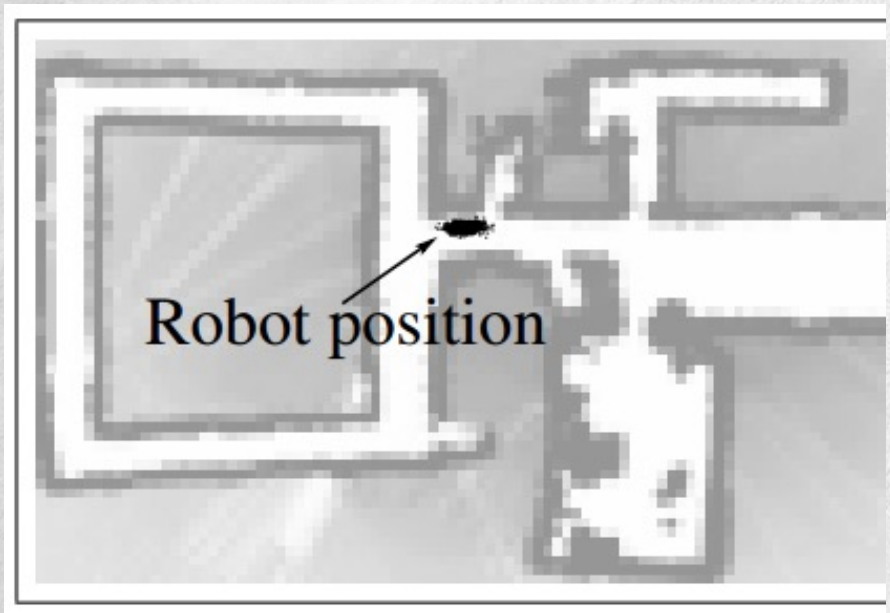
- Represents pdf as a set of samples (particles)
- Each particle contains one set of values for the state variables
- Good for non-Gaussian, multi-modal pdfs
- Find an approximate solution using a complex model (arbitary pdf) rather than an exact solution using a simplified model (Gaussians)

Particle Filters

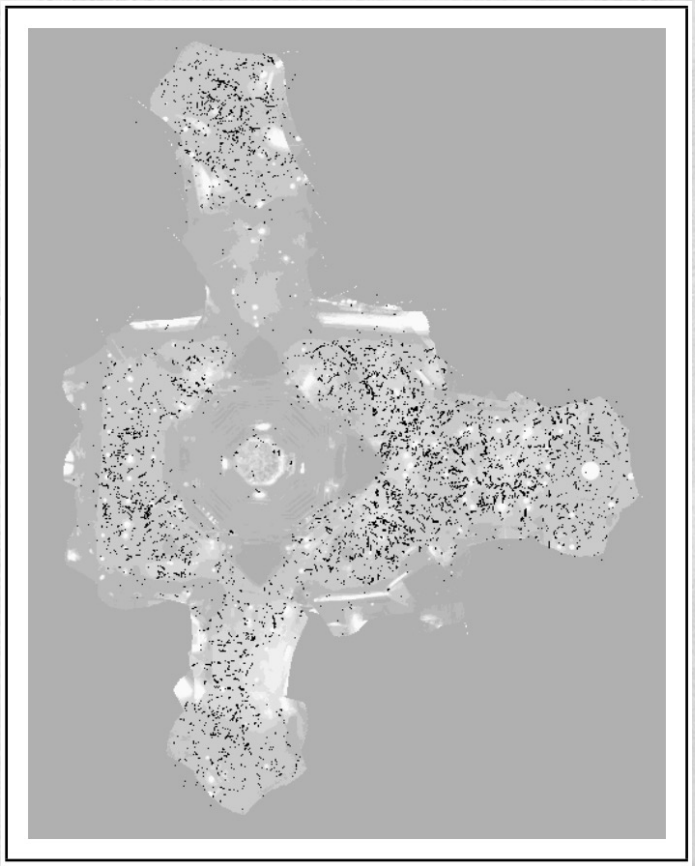
- Main idea: Use samples to represent belief distribution
- Each sample will be called a particle
  - Each particle, or sample, corresponds to a potential pose of the robot
  - Intuitively, each particle is a "guess" at the actual pose
- Highly certain beliefs?
  - All particles very close together, or even the same
- Uncertain beliefs?
  - Particles distributed evenly around the space
- Set of particles approximates the distribution over poses
  - More particles = better approximation
  - Fewer = less precise approximation

Example of Particle Distribution

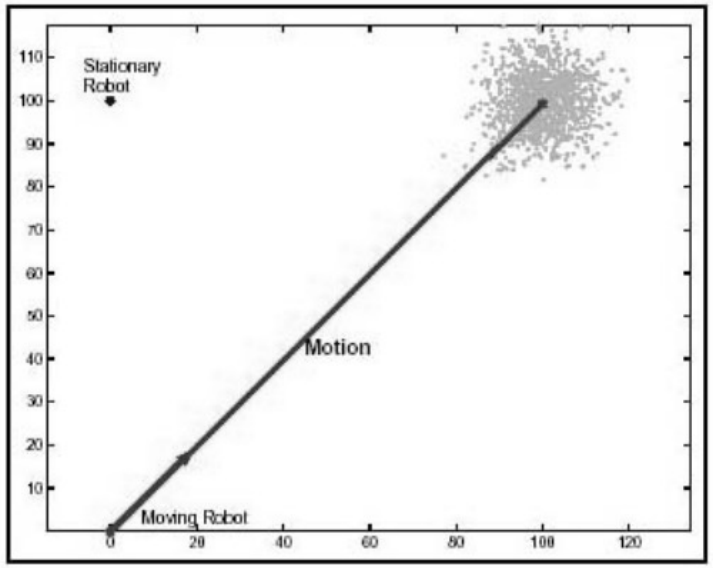
Low Uncertainty



High Uncertainty



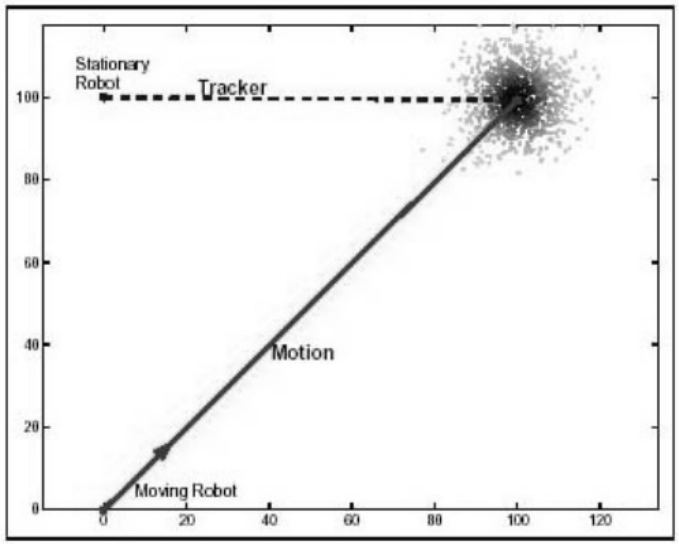
1) Prediction: for each particle, sample and add random, noisy values from action model



Resulting proposal distribution ($q(x)$) approximates

$$\int p(x_t \mid x_{t-1}, u_{t-1})\, p(x_{t-1} \mid d_{0\ldots t-1})\, d\, x_{t-1}$$

2) Update: each particle's weight is the likelihood of getting the sensor readings from that particle's hypothesis



The weight associated with each particle is

$w(x) = p(x)/q(x) = p(z_t \mid x_t)$, normalized so that all the weights sum to 1

3) Resample: new set of particles are chosen such that each particle survives in proportion to its weight

Resulting distribution is p(x):

$$\underbrace{p(x_t \mid d_{o\ldots t})}_{\text{p(x) – the posterior probability}} = \underbrace{\eta\, p(z_t \mid x_t)}_{\text{w(x) – the importance weights}} \underbrace{\int p(x_t \mid u_{t-1}, x_{t-1})\, p(x_{t-1} \mid d_{o\ldots t-1})\, dx_{t-1}}_{\text{q(x) – the prior probability}}$$

Particle Filter: Steps (Simplified)

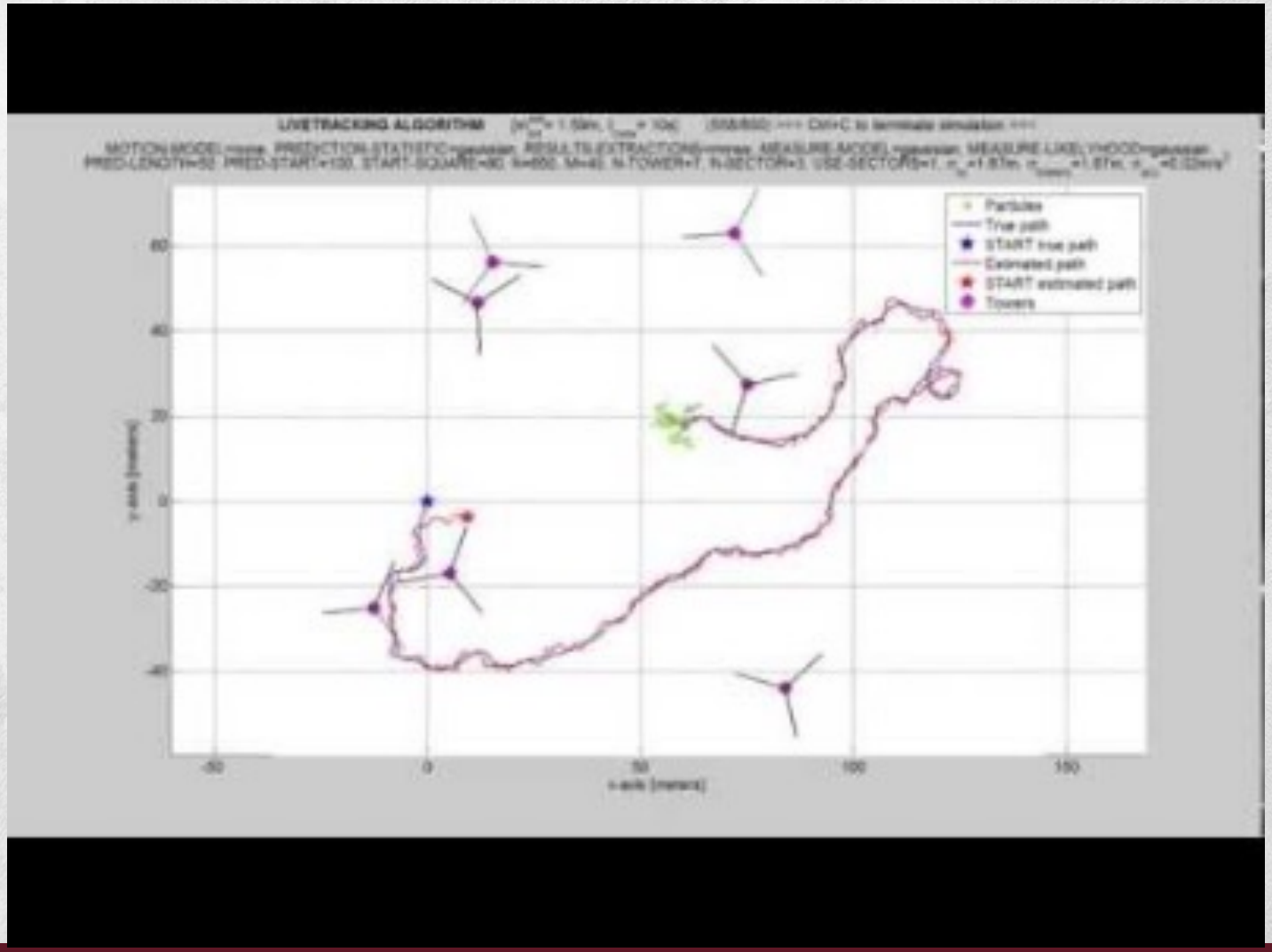
1. Initialize particles
2. At each time step
   1. Use motion model to propagate each particle forward (randomly)
      - (This will spread out particles that were the same)
   2. Use measurement model to assign each particle a weight
      - (Probability of sensor reading given particle's pose)
   3. Resample particles with replacement, with probability proportional to weight

At any point in time, our set of particles represents our current (approximate) beliefs

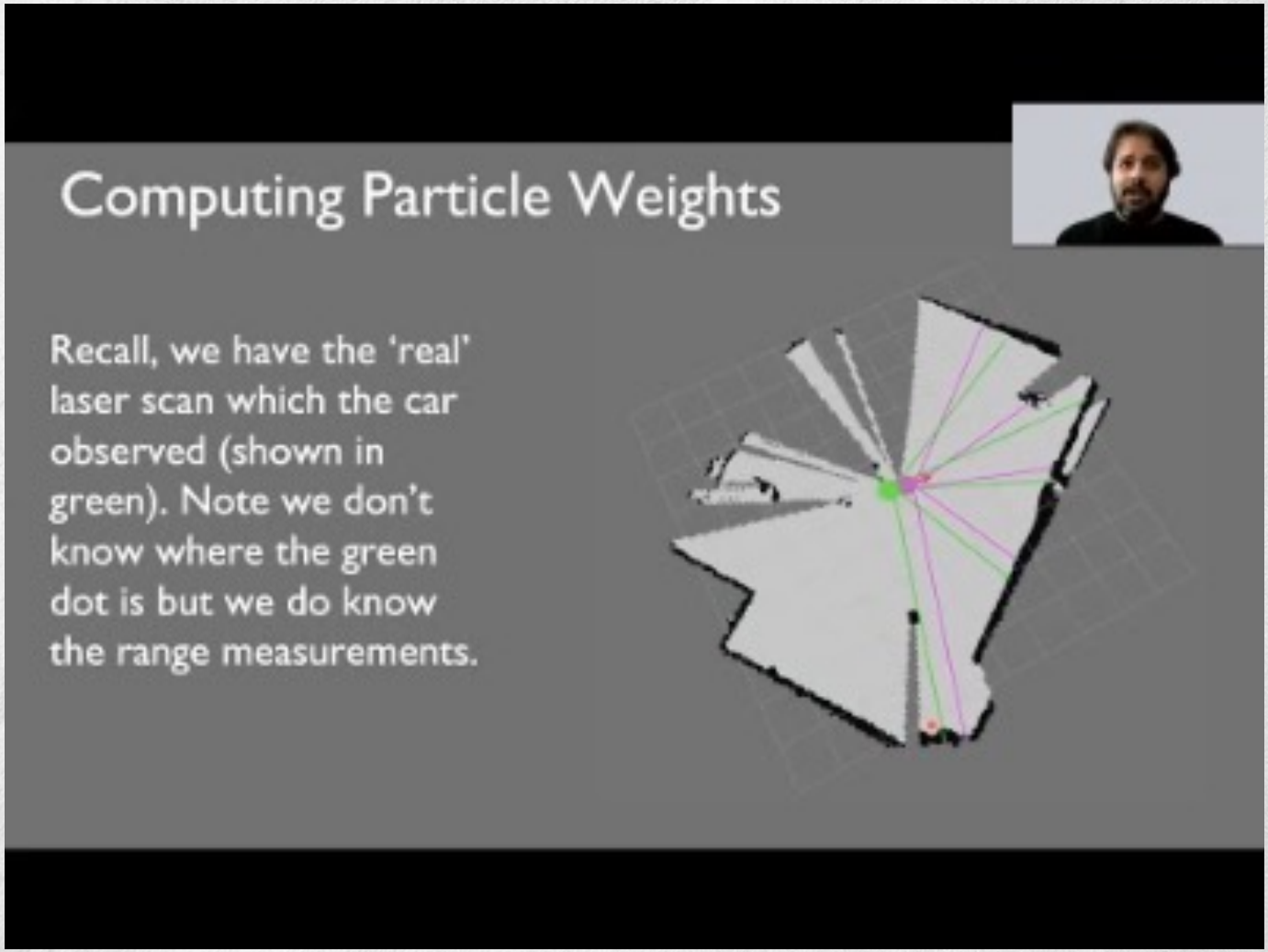
Particle Filter Algorithm

- Particle Filter(χ_{t-1}, u_t, z_t, m):
  - $\chi'_t = \chi_t = \emptyset$
  - for $m = 1$ to M do
    - $x_t^{[m]} =$ **sample_motion_model**($u_t, x_{t-1}^{[m]}$)
    - $w_t^{[m]} =$ **measurement_model**($z_t, x_t^{[m]}, m,$)
    - $\chi'_t = \chi'_t + < x_t^{[m]}, w_t^{[m]} >$
  - endfor
  - for $m = 1$ to M do
    - draw i with probability $\alpha\ w_i^{[m]}$
    - Assume that gives particle j
    - add copy of $x_t^{[j]}$ to χ_t
  - endfor
  - return χ_t

Particle Filter Demo



Particle Filter Demo



References

1. https://web.mit.edu/16.412j/www/html/Advanced%20lectures/Slides/Hsaio_plinval_miller_ParticleFiltersPrint.pdf
2. https://youtu.be/aUkBa1zMKv4
3. https://www.youtube.com/watch?v=HuWNhlpF4Ps