CSE-481O

A Gentle Introduction to Particle Filters

Bayes Filters: Framework

- **Given:**
  - Stream of observations *z* and action data *u:*

$$d_t = \{u_1, z_2 \ldots, u_{t-1}, z_t\}$$

  - Sensor model *P(z|x).*
  - Dynamics model *P(x|u,x')*.
  - Prior probability of the system state *P(x).*
- **Wanted:**
  - Estimate of the state *X* of a dynamical system.
  - The posterior of the state is also called **Belief**:

$$Bel(x_t) = P(x_t \mid u_1, z_2 \ldots, u_{t-1}, z_t)$$

Bayes Filters

z = observation
u = action
x = state

$$Bel(x_t) = P(x_t \mid u_1, z_1 \ldots, u_t, z_t)$$

Bayes
$$= \eta \; P(z_t \mid x_t, u_1, z_1, \ldots, u_t) \; P(x_t \mid u_1, z_1, \ldots, u_t)$$

Markov
$$= \eta \; P(z_t \mid x_t) \; P(x_t \mid u_1, z_1, \ldots, u_t)$$

Total prob.
$$= \eta \; P(z_t \mid x_t) \int P(x_t \mid u_1, z_1, \ldots, u_t, x_{t-1}) P(x_{t-1} \mid u_1, z_1, \ldots, u_t) \, dx_{t-1}$$

Markov
$$= \eta \; P(z_t \mid x_t) \int P(x_t \mid u_t, x_{t-1}) \; P(x_{t-1} \mid u_1, z_1, \ldots, u_t) \, dx_{t-1}$$

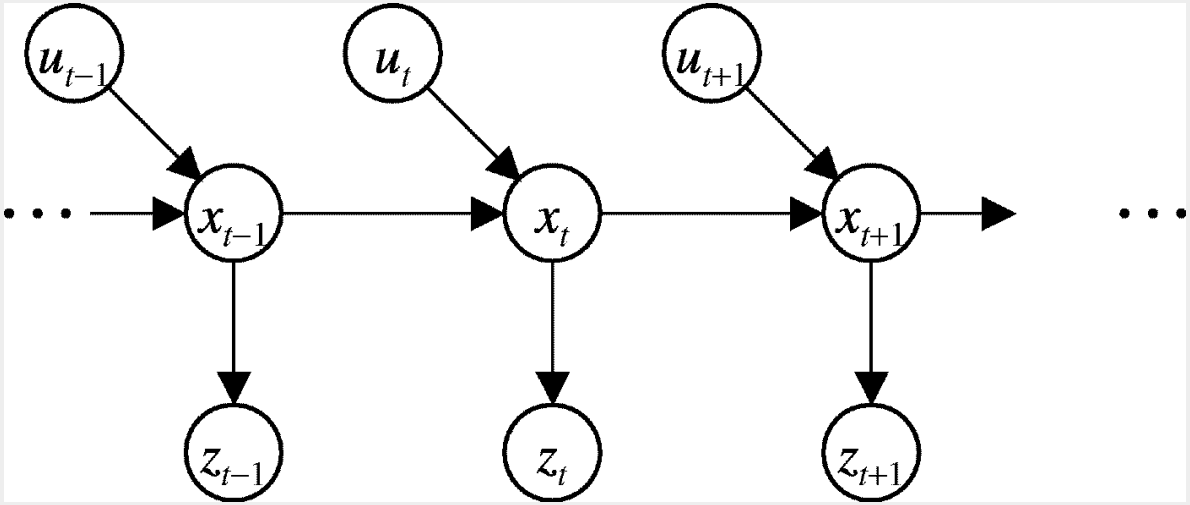
$$= \eta \; P(z_t \mid x_t) \int P(x_t \mid u_t, x_{t-1}) \; Bel(x_{t-1}) \, dx_{t-1}$$

Bayes Filters are Familiar!

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

- Kalman filters
- Particle filters
- Hidden Markov models
- Dynamic Bayesian networks
- Partially Observable Markov Decision Processes (POMDPs)

Graphical Model Representation



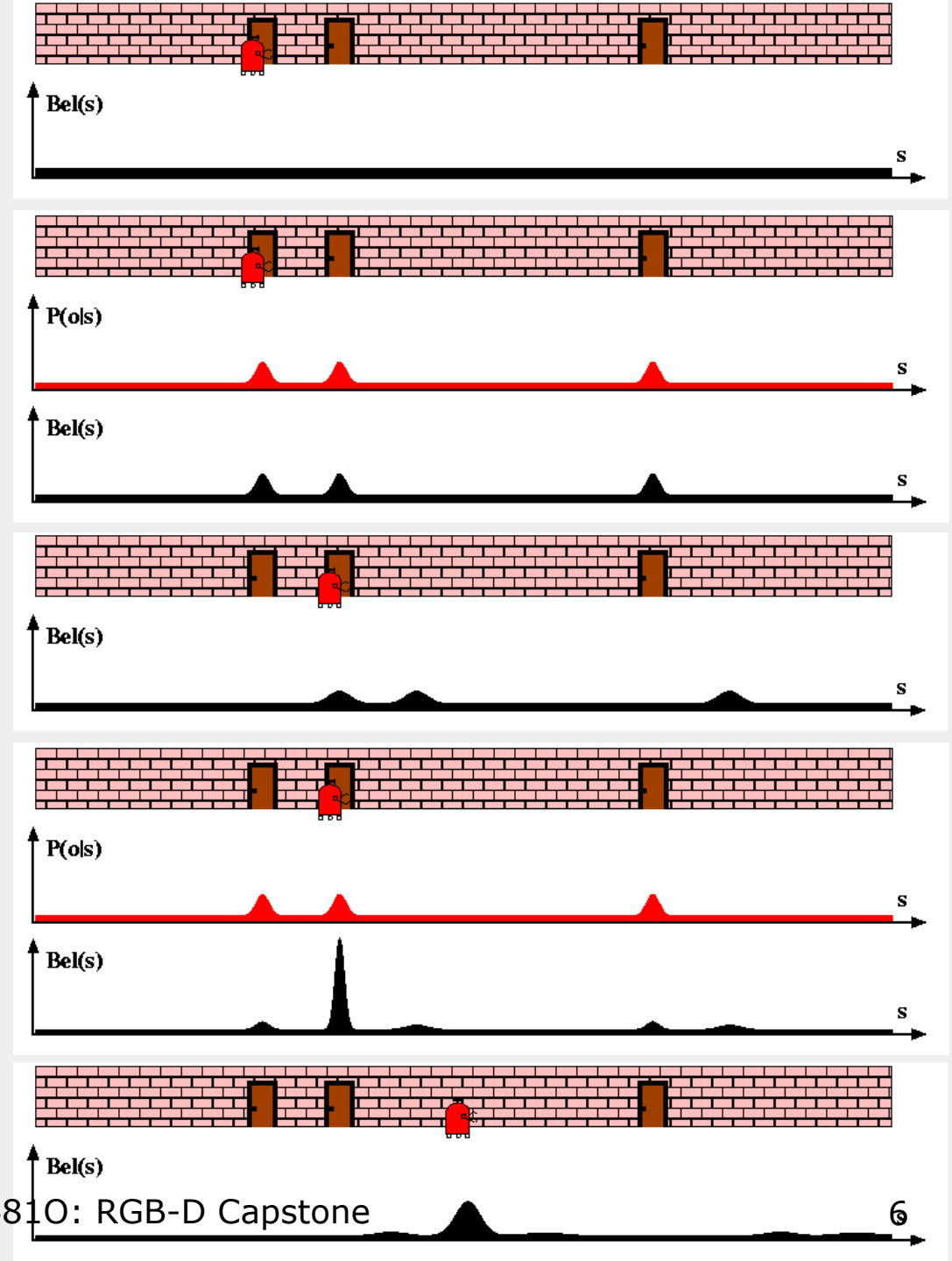
$$p(z_t \mid x_{0:t}, z_{1:t-1}, u_{1:t}) = p(z_t \mid x_t)$$

$$p(x_t \mid x_{1:t-1}, z_{1:t-1}, u_{1:t}) = p(x_t \mid x_{t-1}, u_t)$$

Underlying Assumptions

- Static world
- Independent noise
- Perfect model, no approximation errors

Bayes Filters for Robot Localization

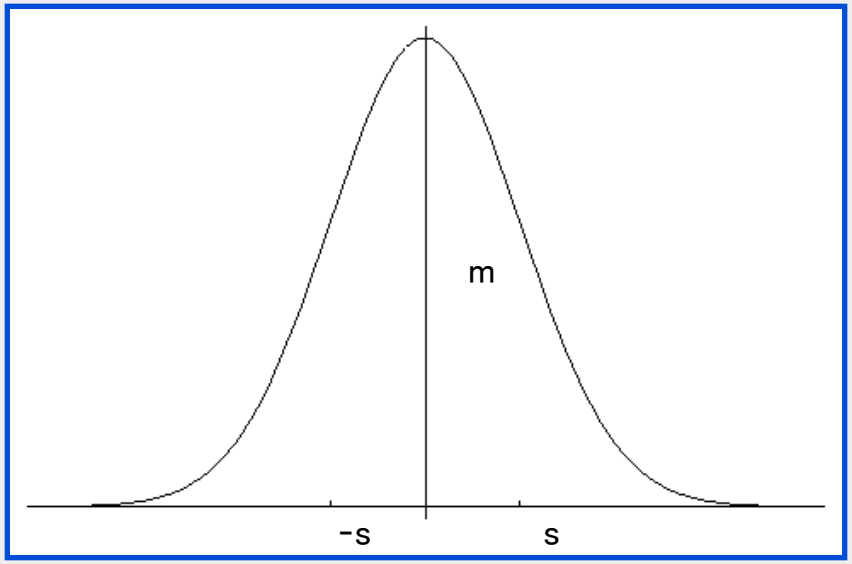


Gaussians

$$p(x) \sim N(\mu, \sigma^2):$$

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}}$$

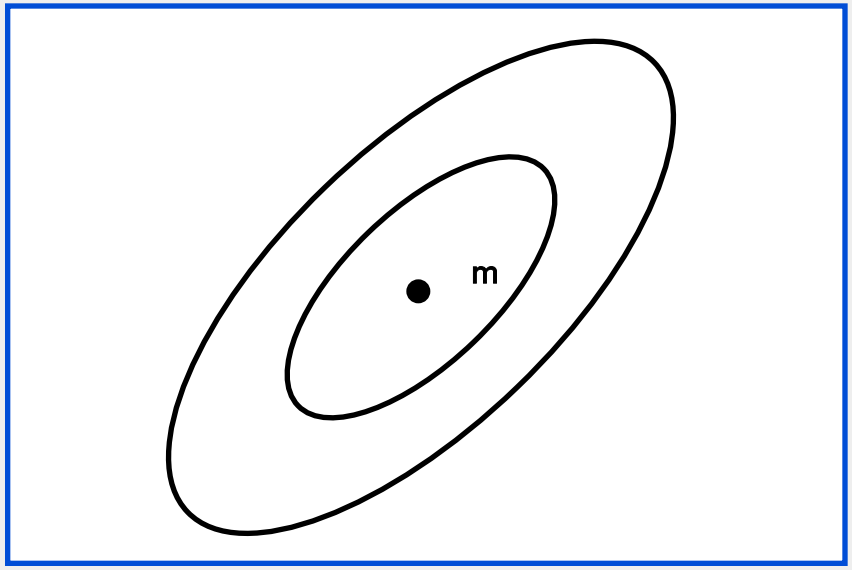
Univariate



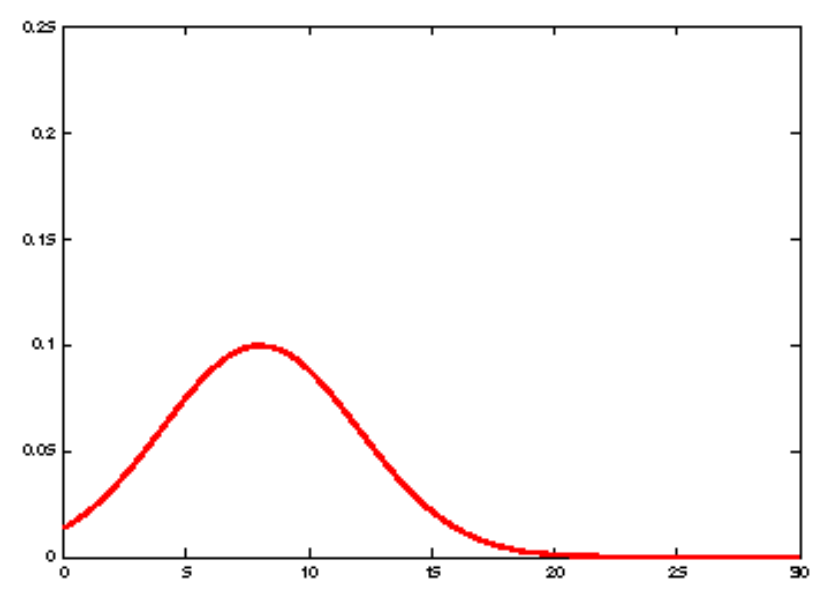
$$p(\mathbf{x}) \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma}):$$

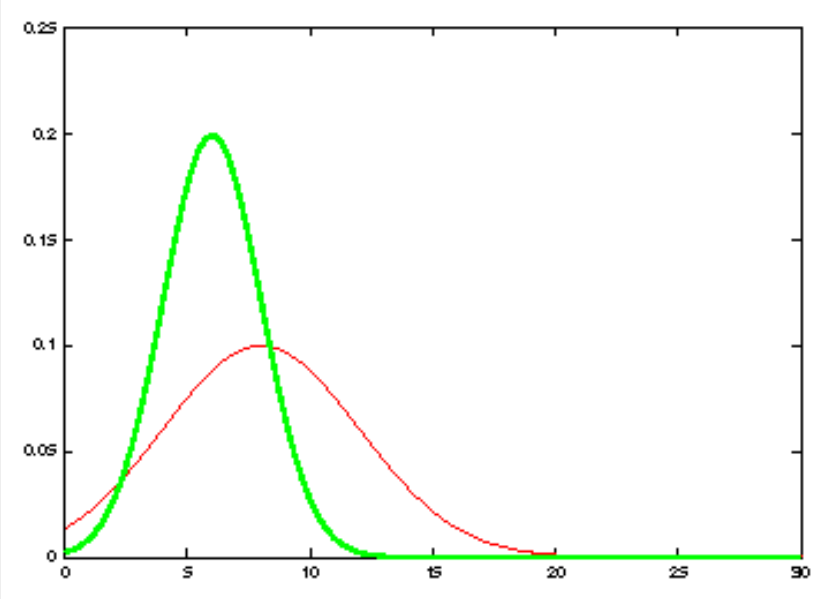
$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}\left|\boldsymbol{\Sigma}\right|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^t \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}$$

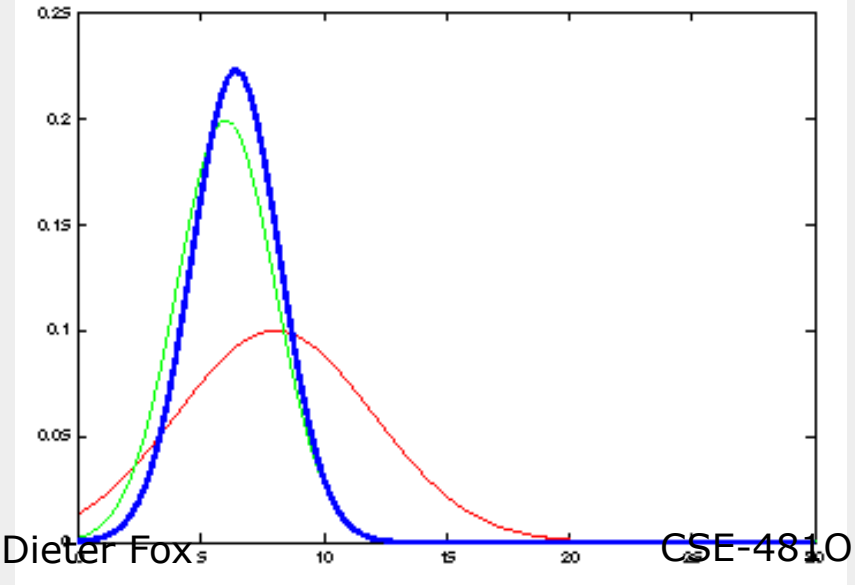
Multivariate



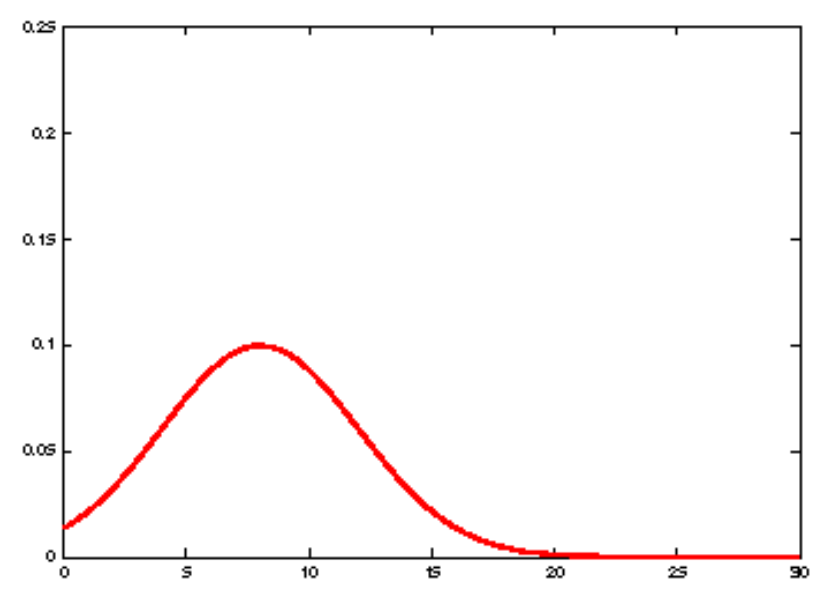
Kalman Filter Updates in 1D

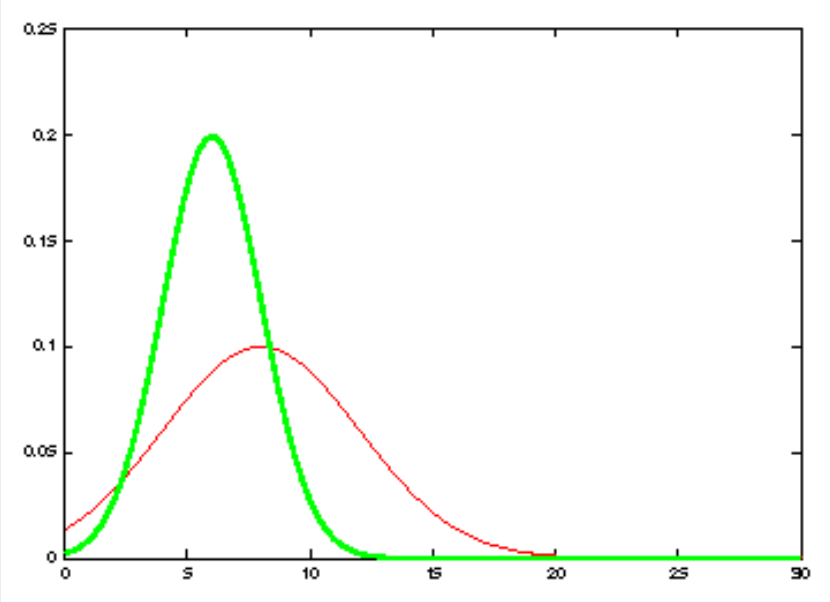


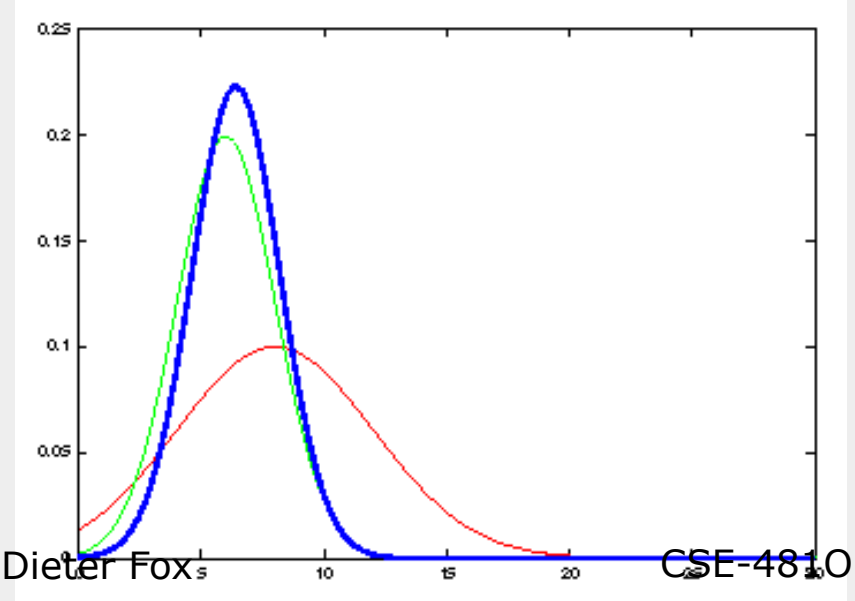


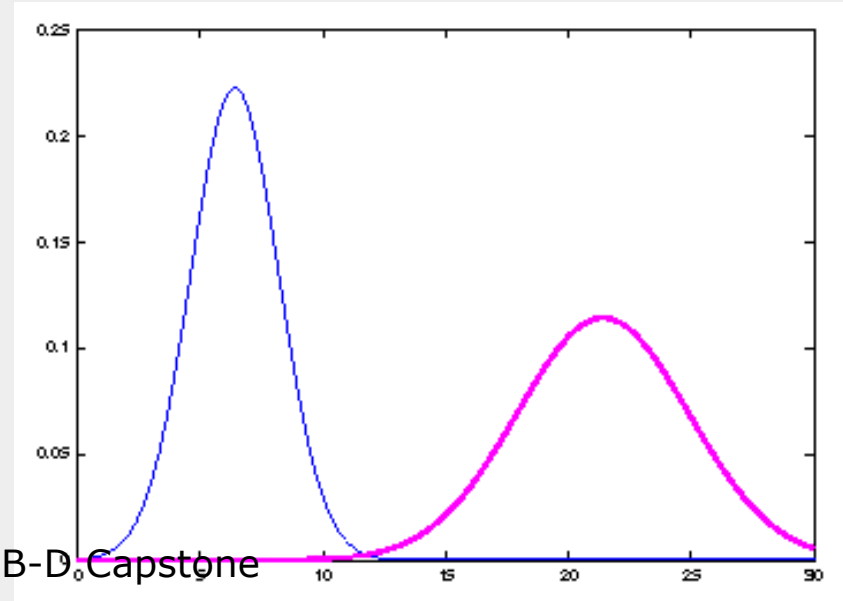


Kalman Filter Updates in 1D

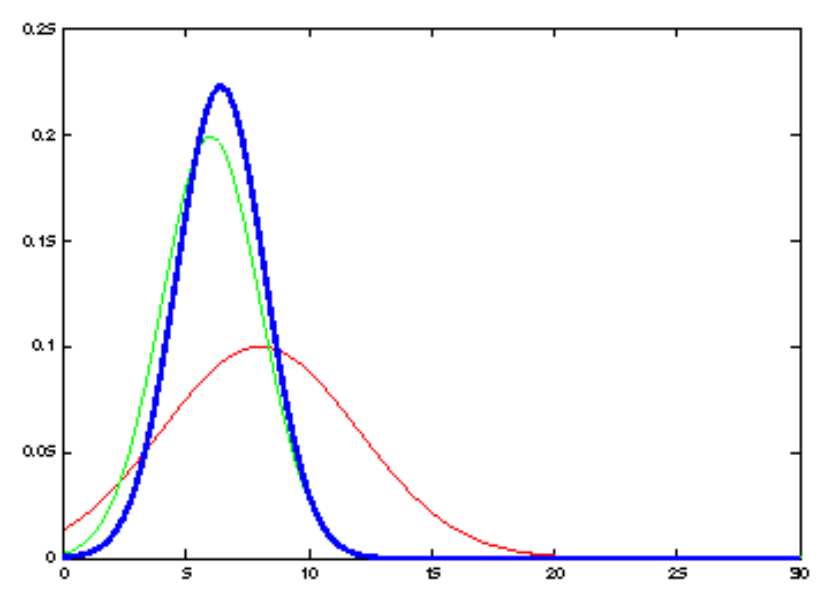


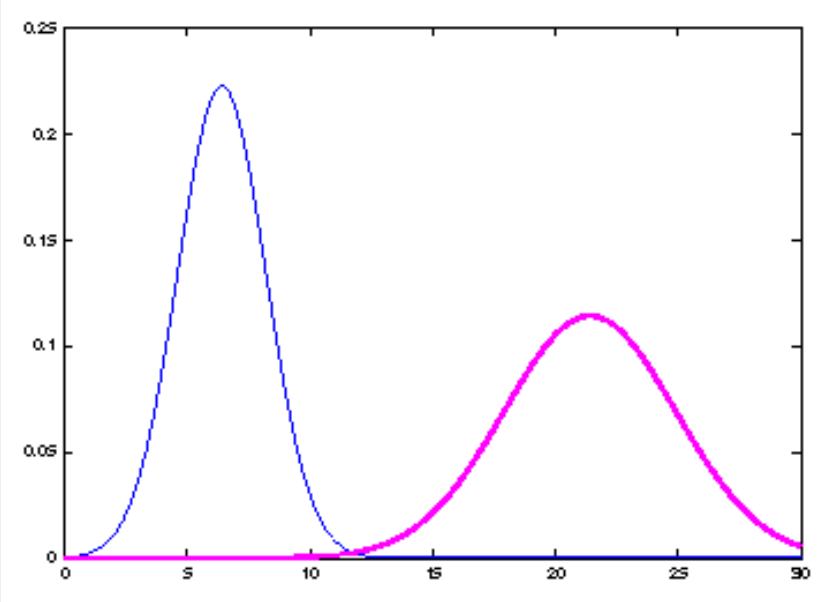


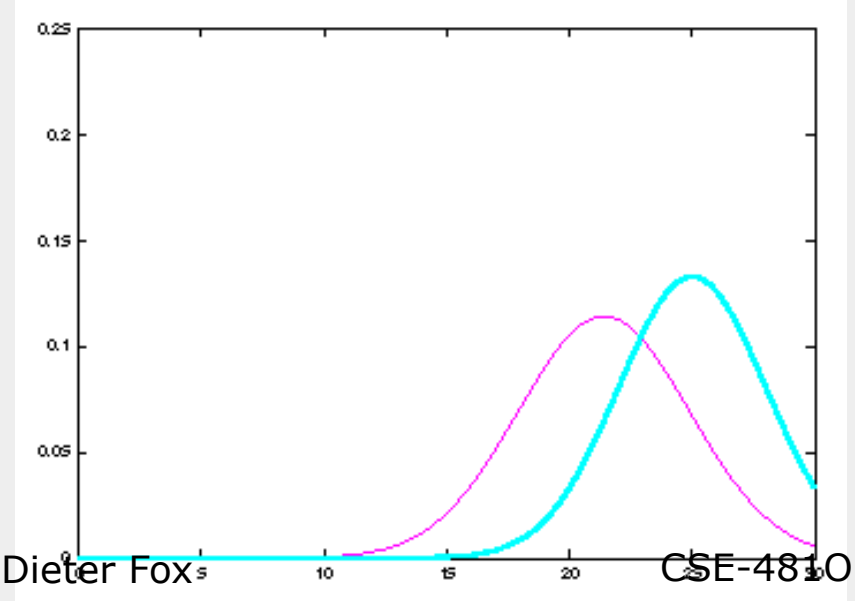


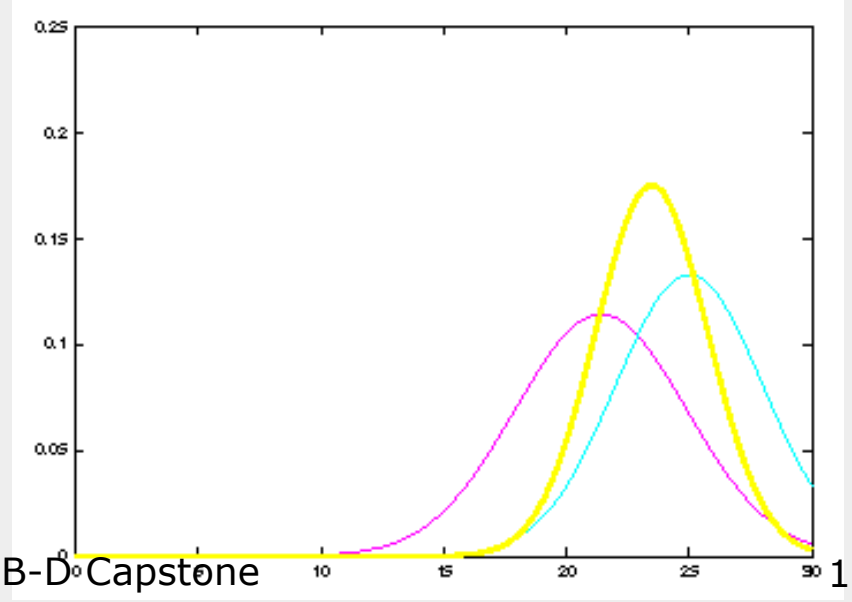


Kalman Filter Updates

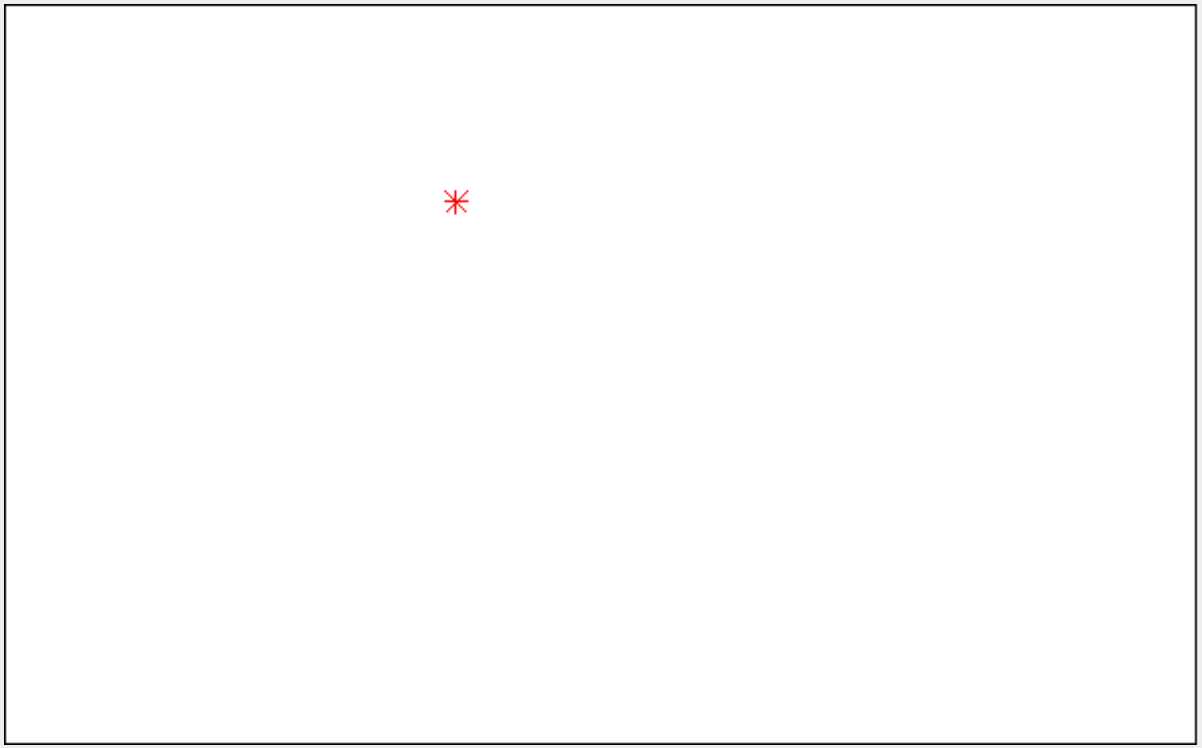




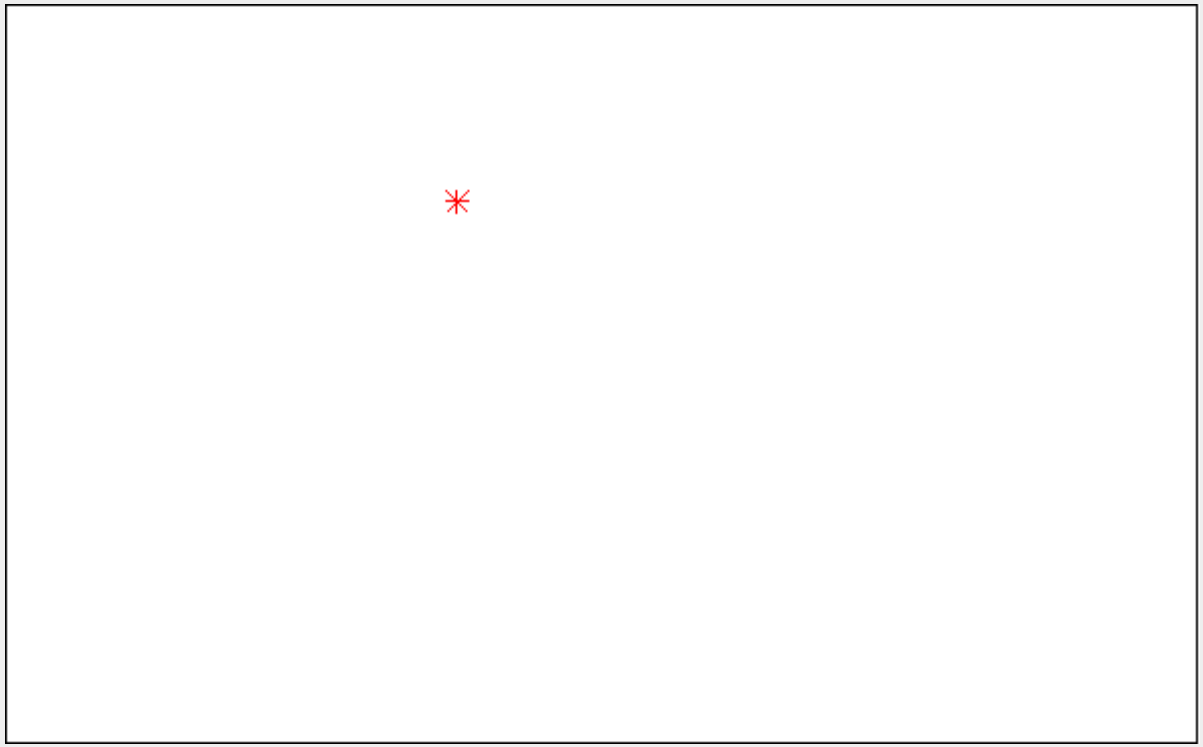




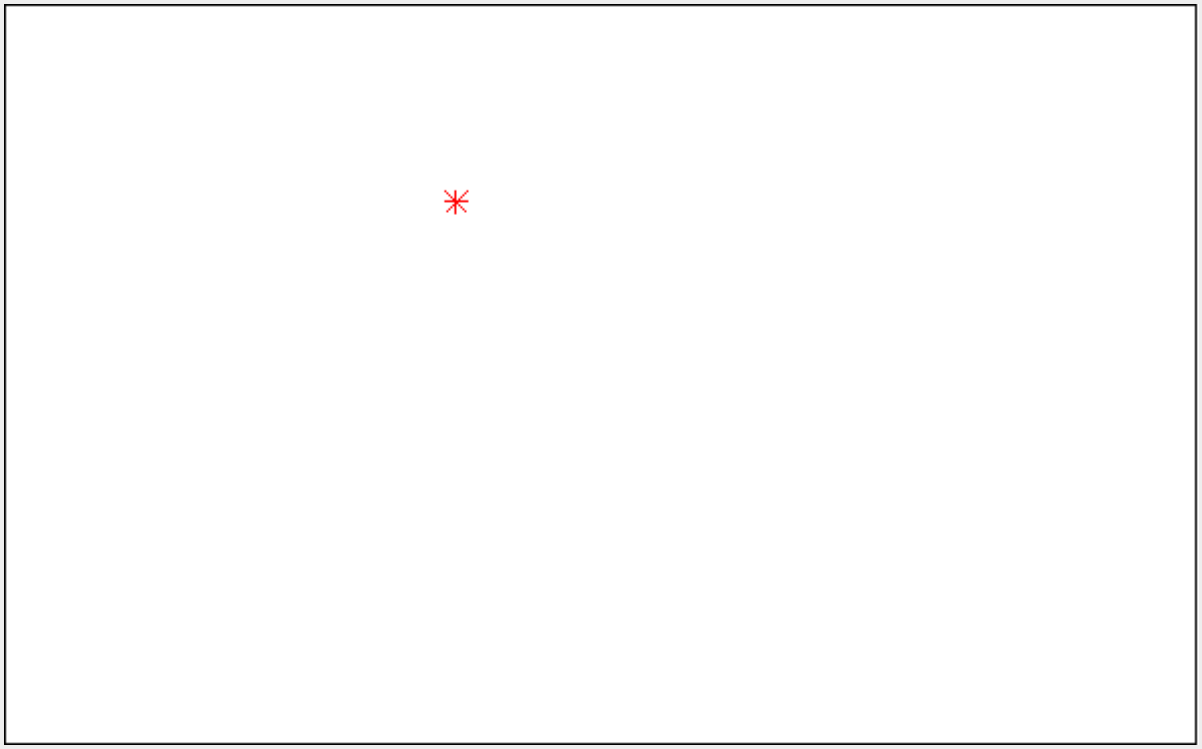
Simulation Run



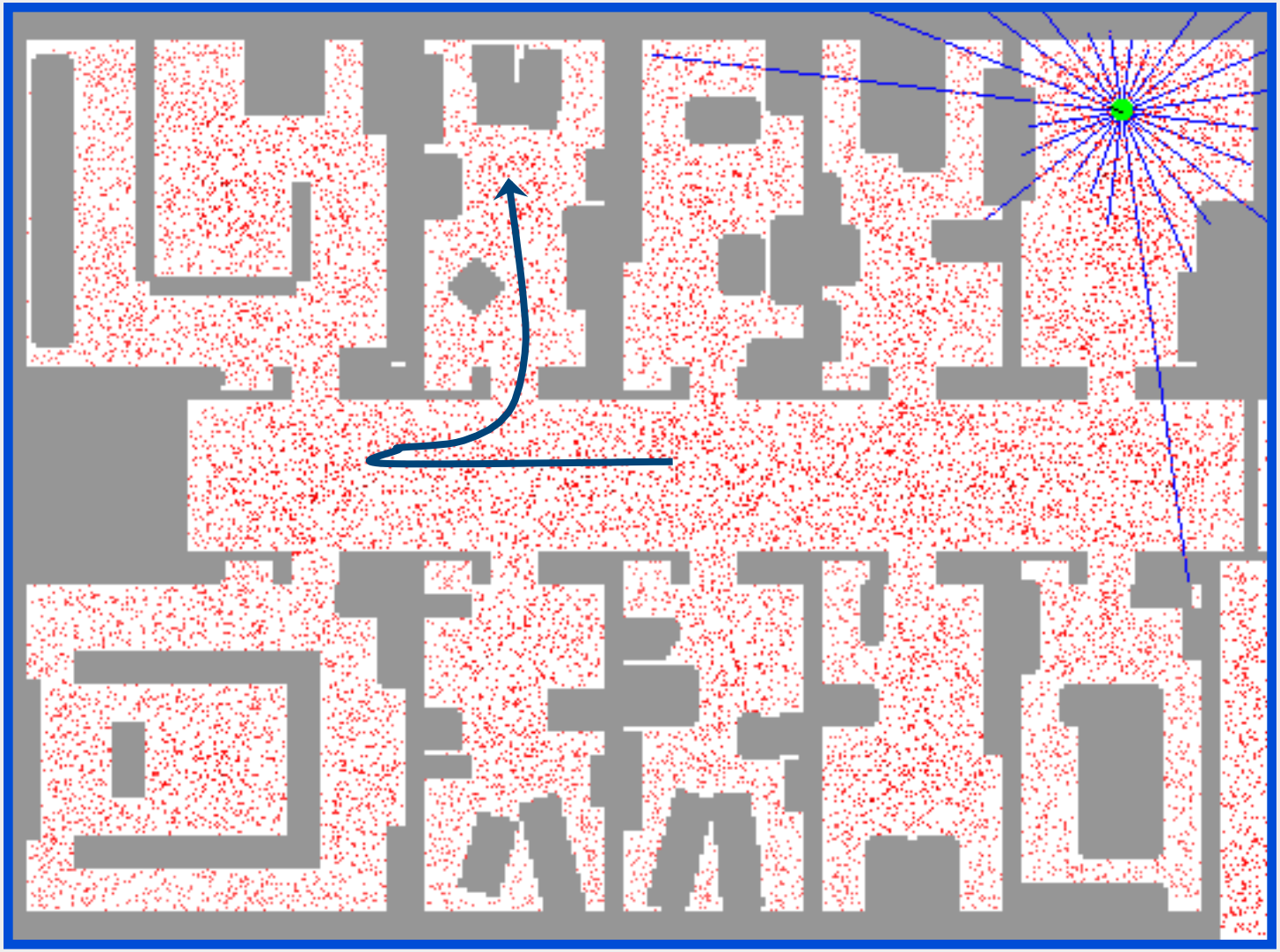
Kalman Filter (optimized for straight motion)



Kalman Filter (inflated prediction noise)

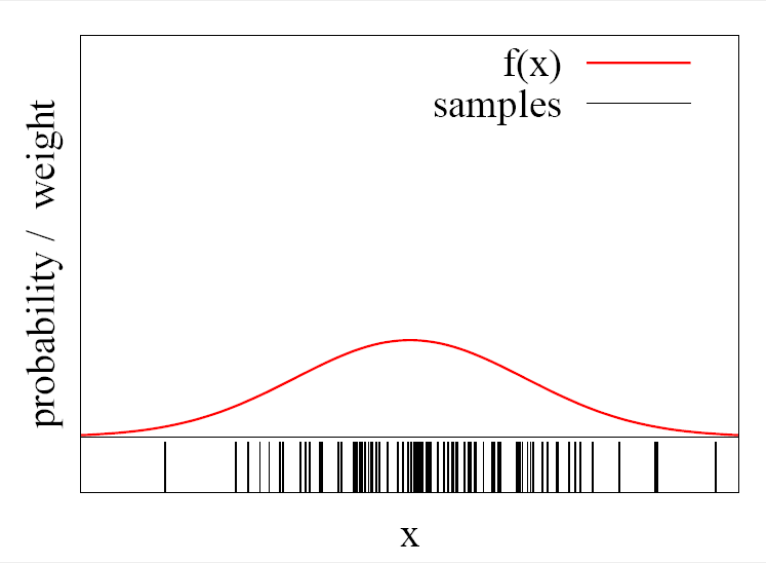


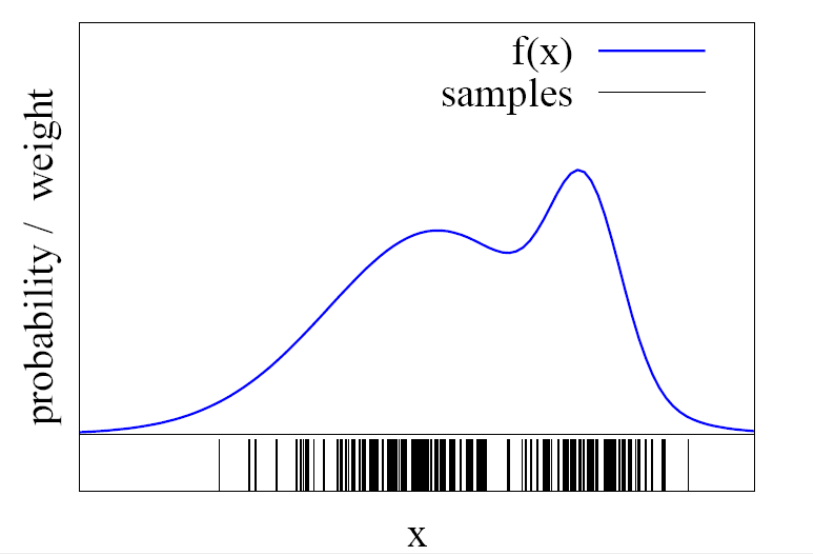
Sample-based Localization (sonar)



Sample-Based Density Approximation

- Particle sets can be used to approximate densities



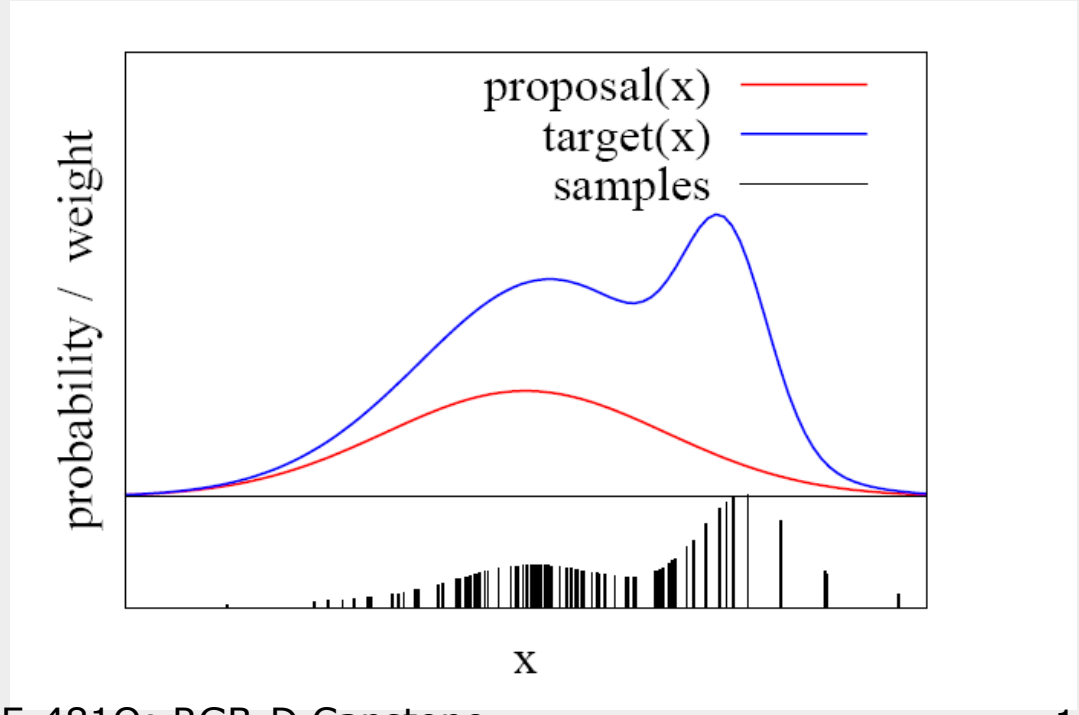


- The more particles fall into an interval, the higher the probability of that interval

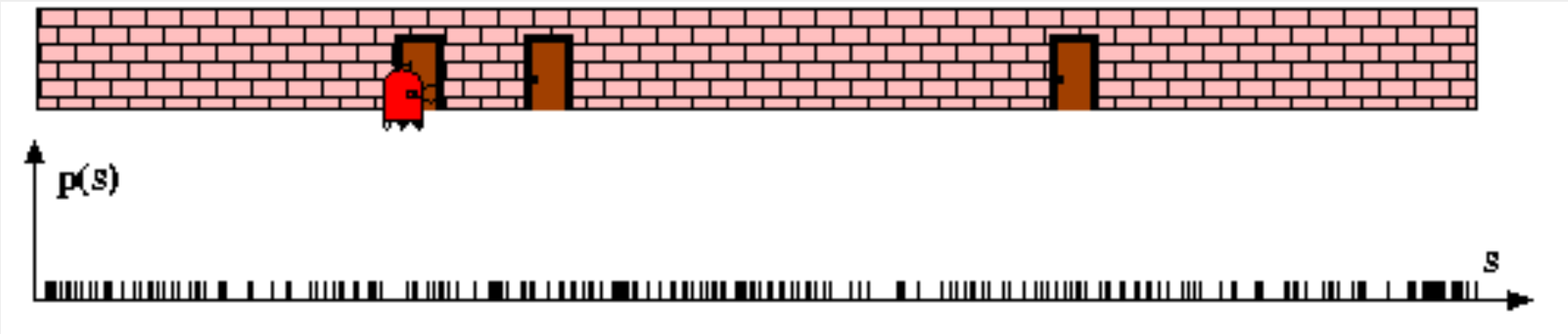
- How to draw samples form a function/distribution?

Importance Sampling Principle

- We can even use a different distribution g to generate samples from f
- By introducing an importance weight w, we can account for the "differences between g and f"
- $w = f / g$
- f is often called target
- g is often called proposal



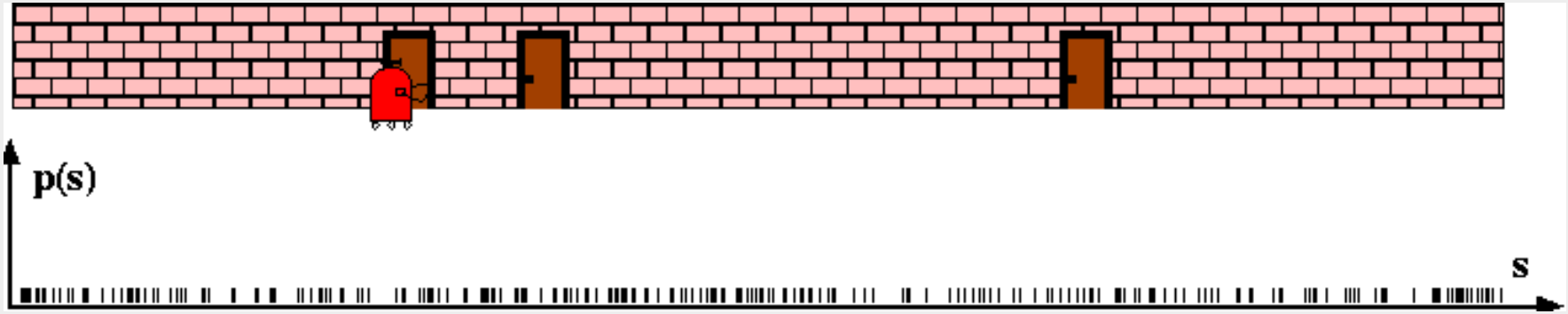
Particle Filters

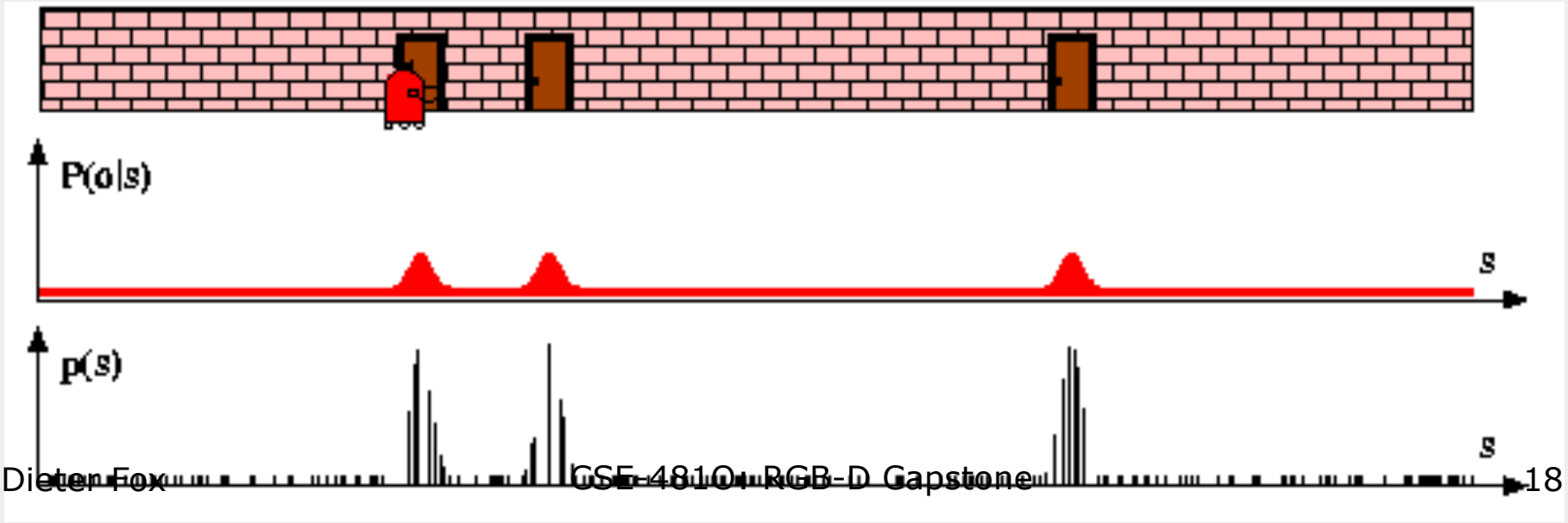


Sensor Information: Importance Sampling

$$Bel(x) \leftarrow \alpha\ p(z \mid x)\ Bel^{-}(x)$$

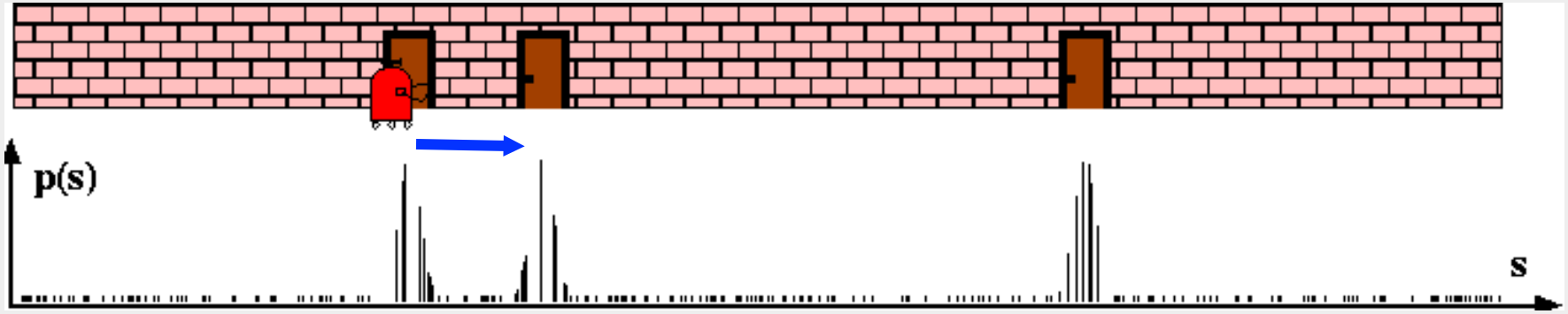
$$w \leftarrow \frac{\alpha\ p(z \mid x)\ Bel^{-}(x)}{Bel^{-}(x)} = \alpha\ p(z \mid x)$$

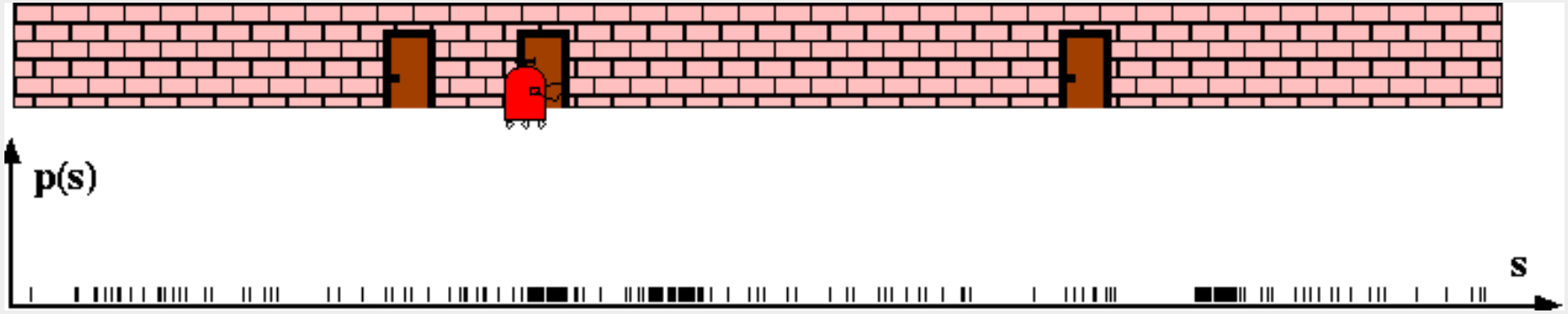




Robot Motion

$$Bel^{-}(x) \quad \leftarrow \quad \int p(x \mid u, x') \, Bel(x') \;\; \mathrm{d}x'$$

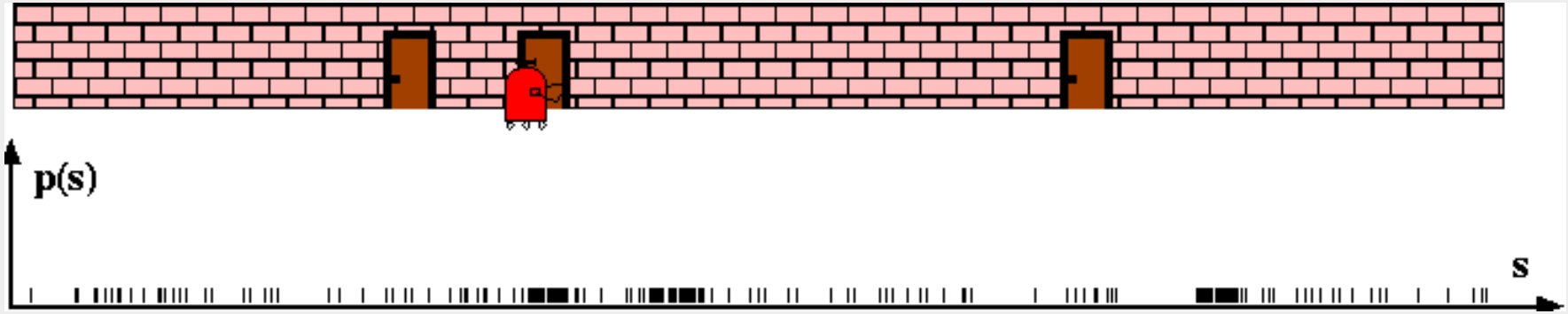


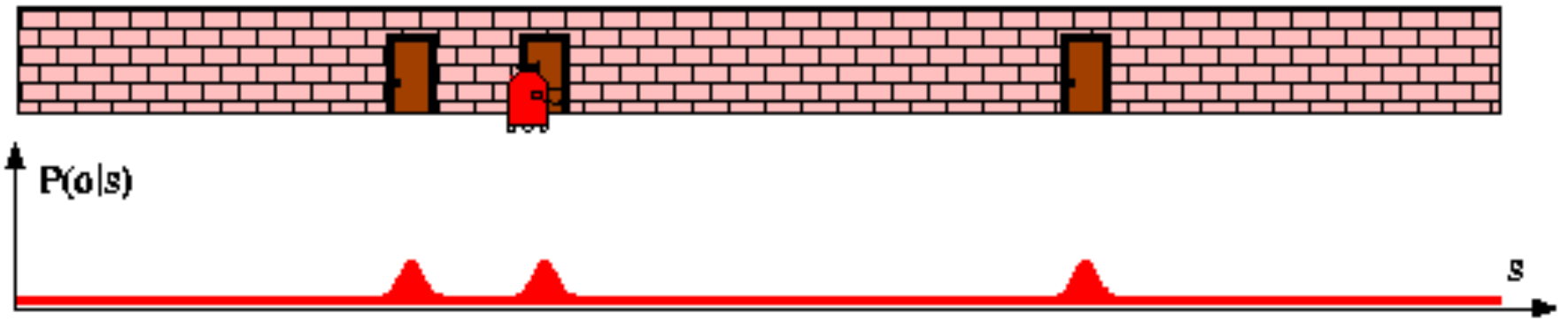


Sensor Information: Importance Sampling

$$Bel(x) \leftarrow \alpha\ p(z \mid x)\ Bel^{-}(x)$$

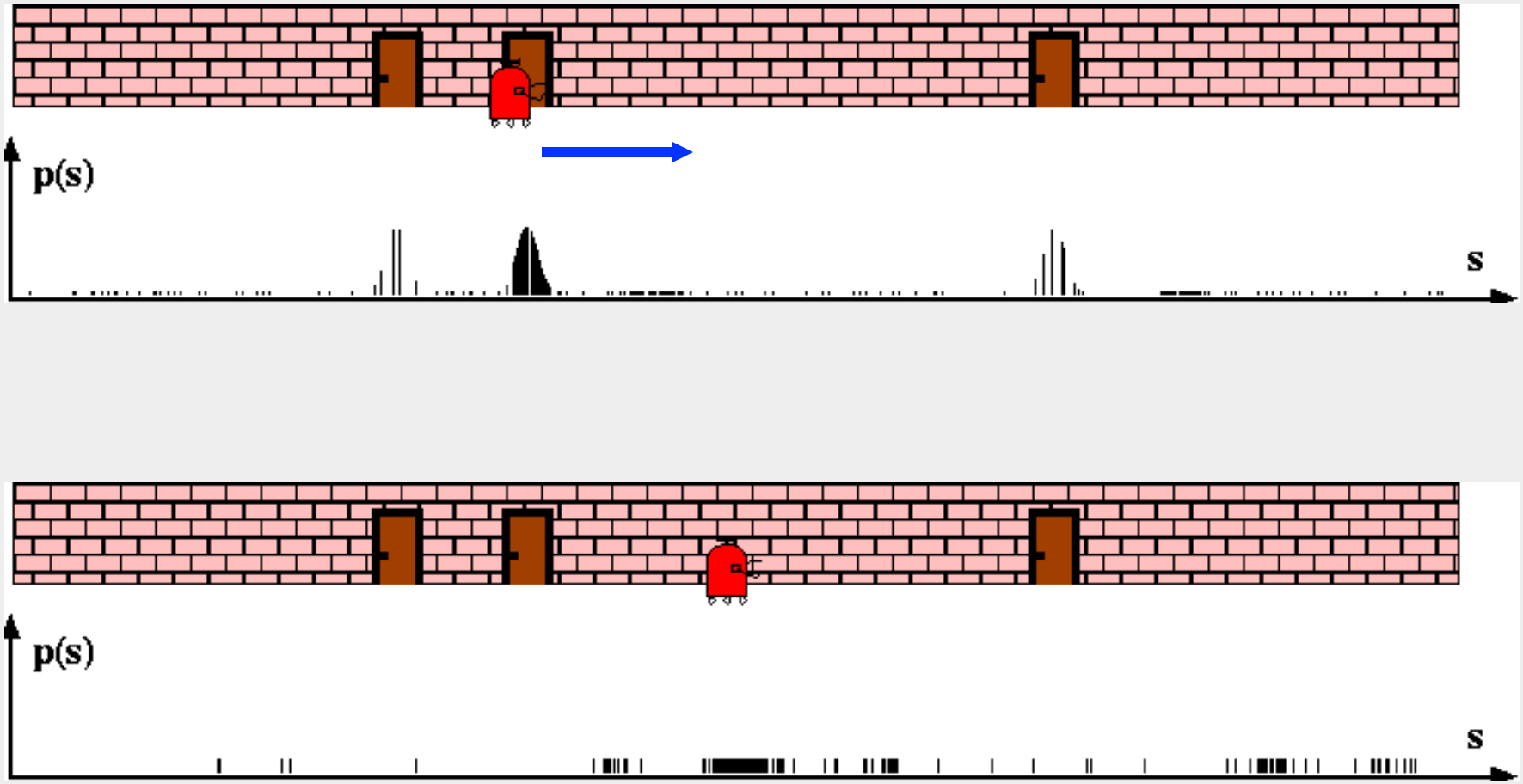
$$w \leftarrow \frac{\alpha\ p(z \mid x)\ Bel^{-}(x)}{Bel^{-}(x)} = \alpha\ p(z \mid x)$$





Robot Motion

$$Bel^{-}(x) \quad \leftarrow \quad \int p(x \mid u, x') \, Bel(x') \;\; \mathrm{d}x'$$



Particle Filter Algorithm

1. Algorithm **particle_filter**(S_{t-1}, u_{t-1} z_t):
2. $S_t = \emptyset, \quad \eta = 0$
3. **For** $i = 1 \ldots n$ ***Generate new samples***
4. Sample index $j(i)$ from the discrete distribution given by w_{t-1}
5. Sample x_t^i from $p(x_t \mid x_{t-1}, u_{t-1})$ using $x_{t-1}^{j(i)}$ and u_{t-1}
6. $w_t^i = p(z_t \mid x_t^i)$ ***Compute importance weight***
7. $\eta = \eta + w_t^i$ ***Update normalization factor***
8. $S_t = S_t \cup \{< x_t^i, w_t^i >\}$ ***Insert***
9. **For** $i = 1 \ldots n$
10. $w_t^i = w_t^i / \eta$ ***Normalize weights***

Particle Filter Algorithm

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

draw x^i_{t-1} from $Bel(x_{t-1})$

draw x^i_t from $p(x_t \mid x^i_{t-1}, u_{t-1})$

Importance factor for x^i_t:

$$w^i_t = \frac{\text{target distribution}}{\text{proposal distribution}}$$

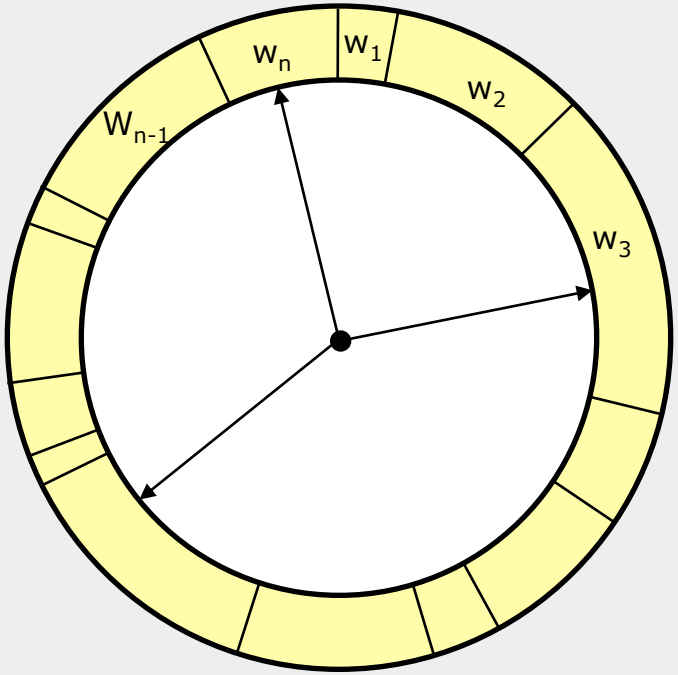
$$= \frac{\eta\; p(z_t \mid x_t)\; p(x_t \mid x_{t-1}, u_{t-1})\; Bel(x_{t-1})}{p(x_t \mid x_{t-1}, u_{t-1})\; Bel(x_{t-1})}$$

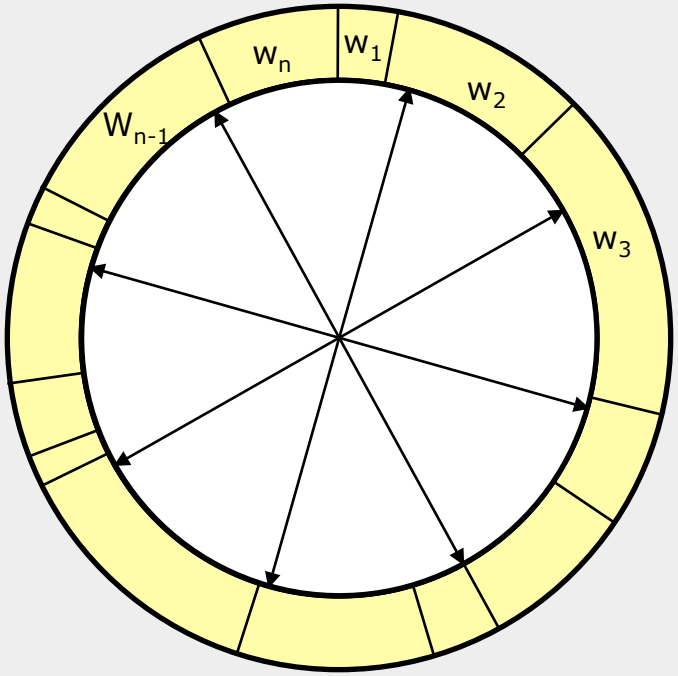
$$\propto p(z_t \mid x_t)$$

Resampling

- **Given**: Set S of weighted samples.
- **Wanted** : Random sample, where the probability of drawing x_i is given by w_i.
- Typically done n times with replacement to generate new sample set S'.

Resampling





- Roulette wheel
- Binary search, n log n

- Stochastic universal sampling
- Systematic resampling
- Linear time complexity
- Easy to implement, low variance

Resampling Algorithm

1. Algorithm **systematic_resampling**(*S,n*):

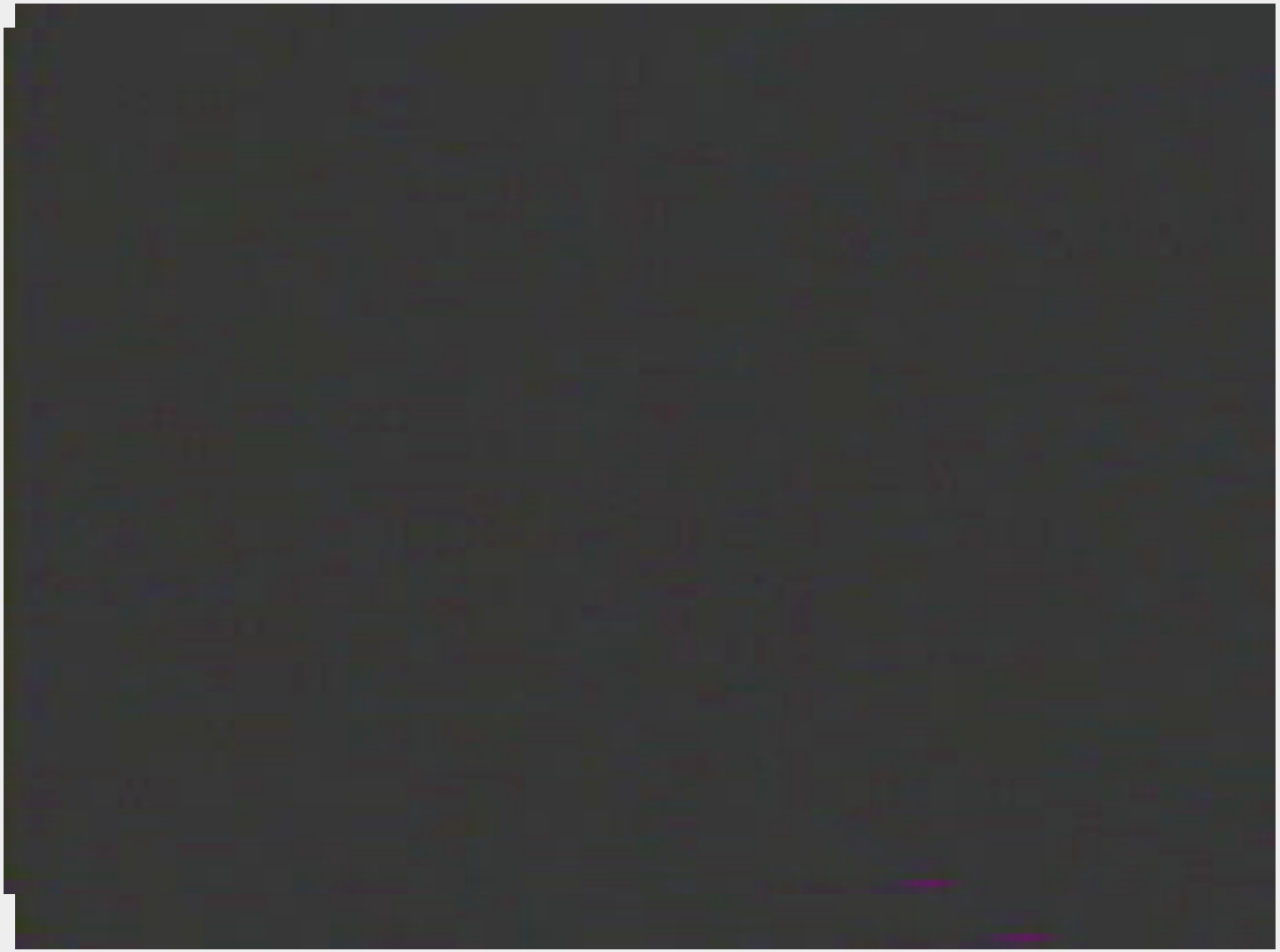
2. $S' = \emptyset, c_1 = w^1$
3. **For** $i = 2 \ldots n$ — ***Generate cdf***
4. $c_i = c_{i-1} + w^i$
5. $u_1 \sim U[0, n^{-1}], i = 1$ — ***Initialize threshold***

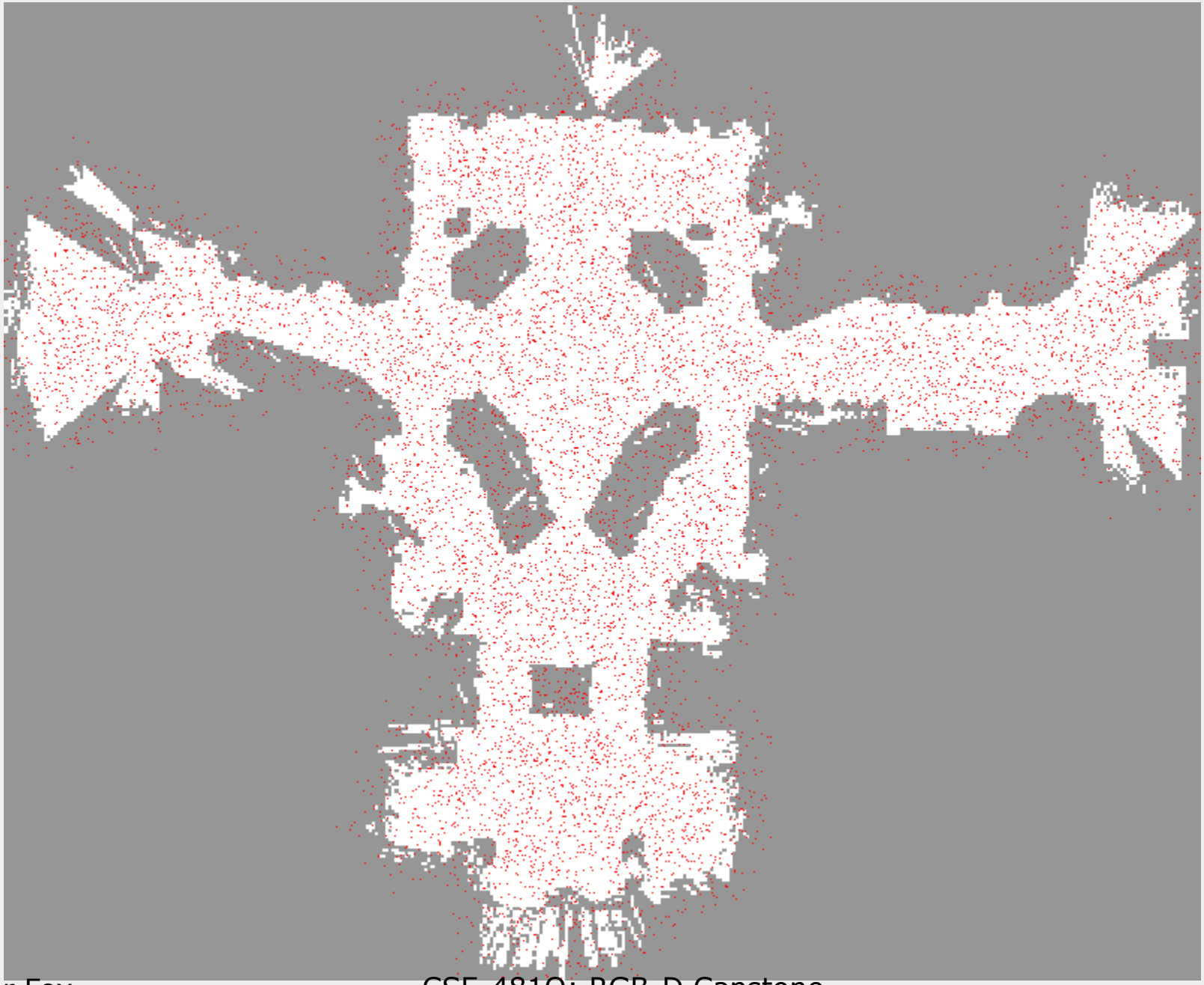
6. **For** $j = 1 \ldots n$ — ***Draw samples …***
7. **While** ($u_j > c_i$) — ***Skip until next threshold reached***
8. $i = i + 1$
9. $S' = S' \cup \{< x^i, n^{-1} >\}$ — ***Insert***
10. $u_j = u_j + n^{-1}$ — ***Increment threshold***

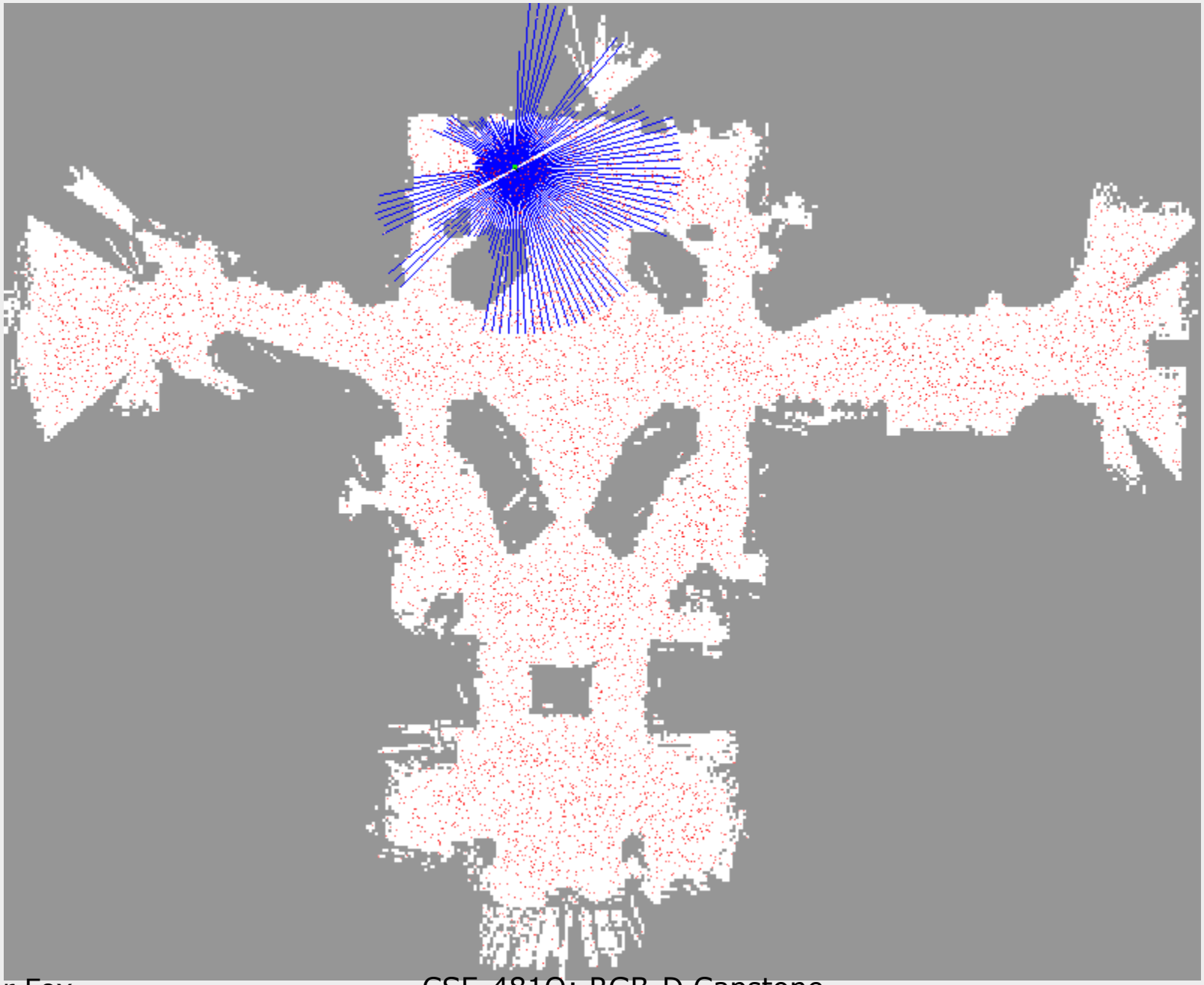
11. **Return** S'

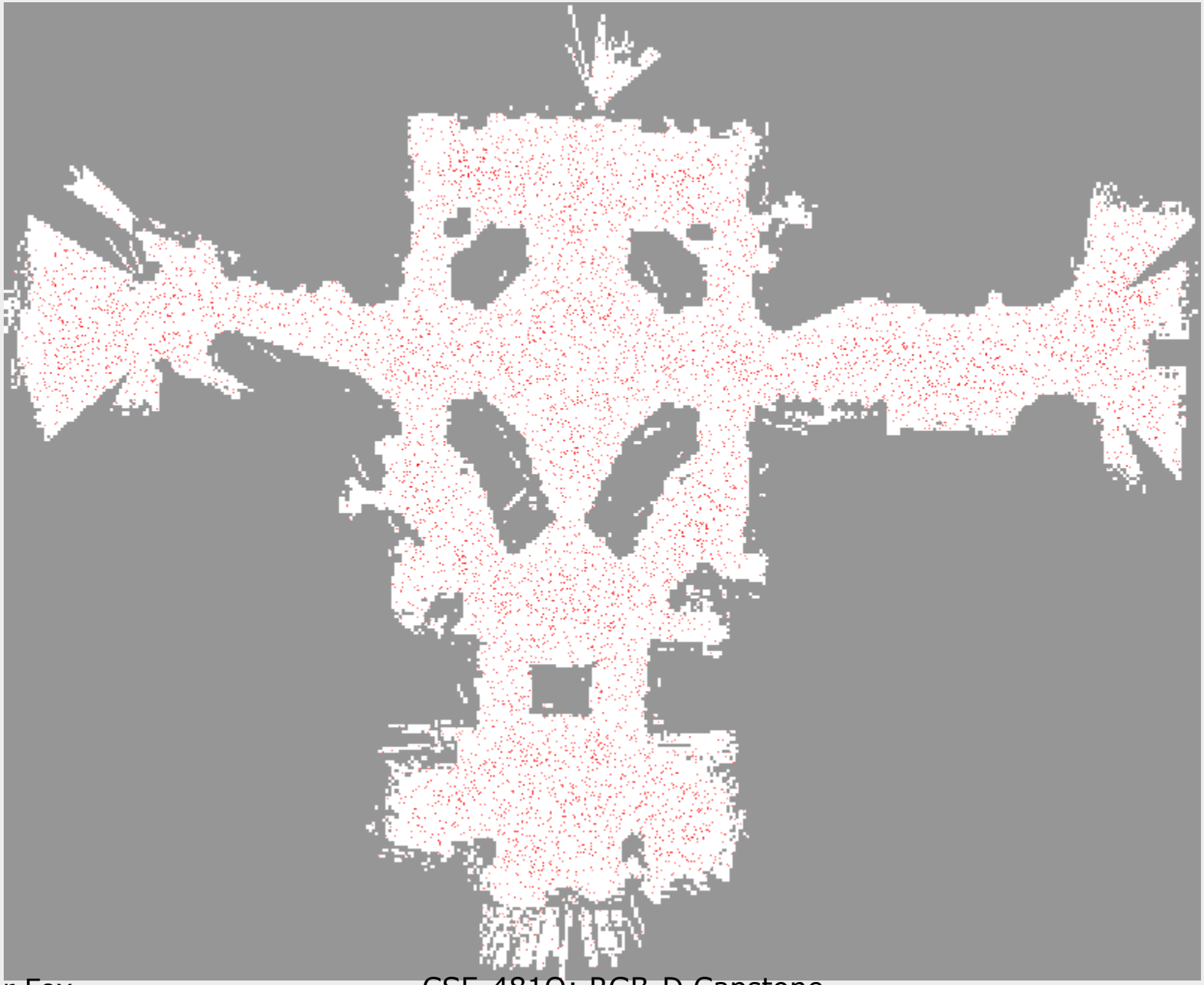
Also called **stochastic universal sampling**

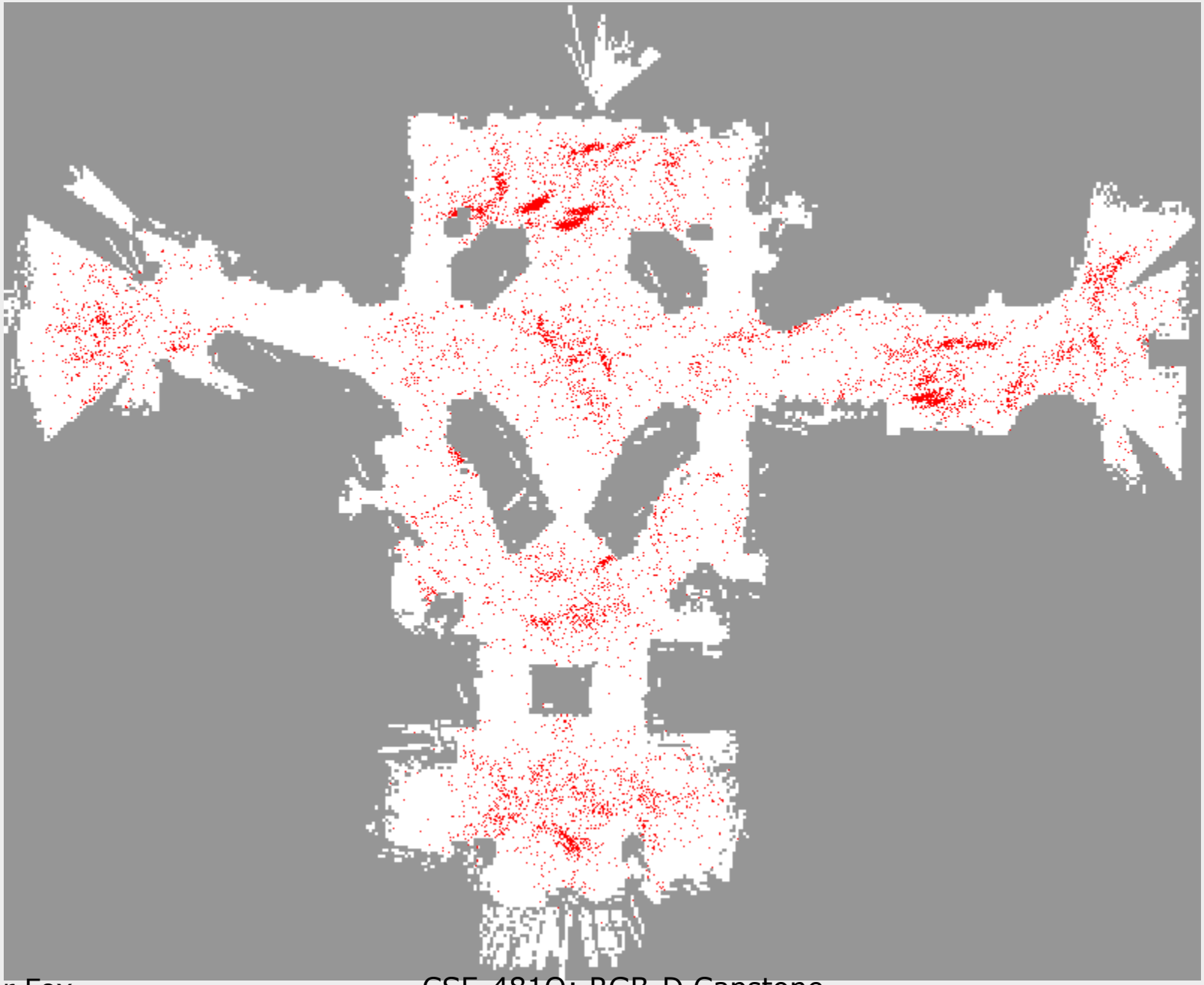
Museum Tour-Guide Minerva

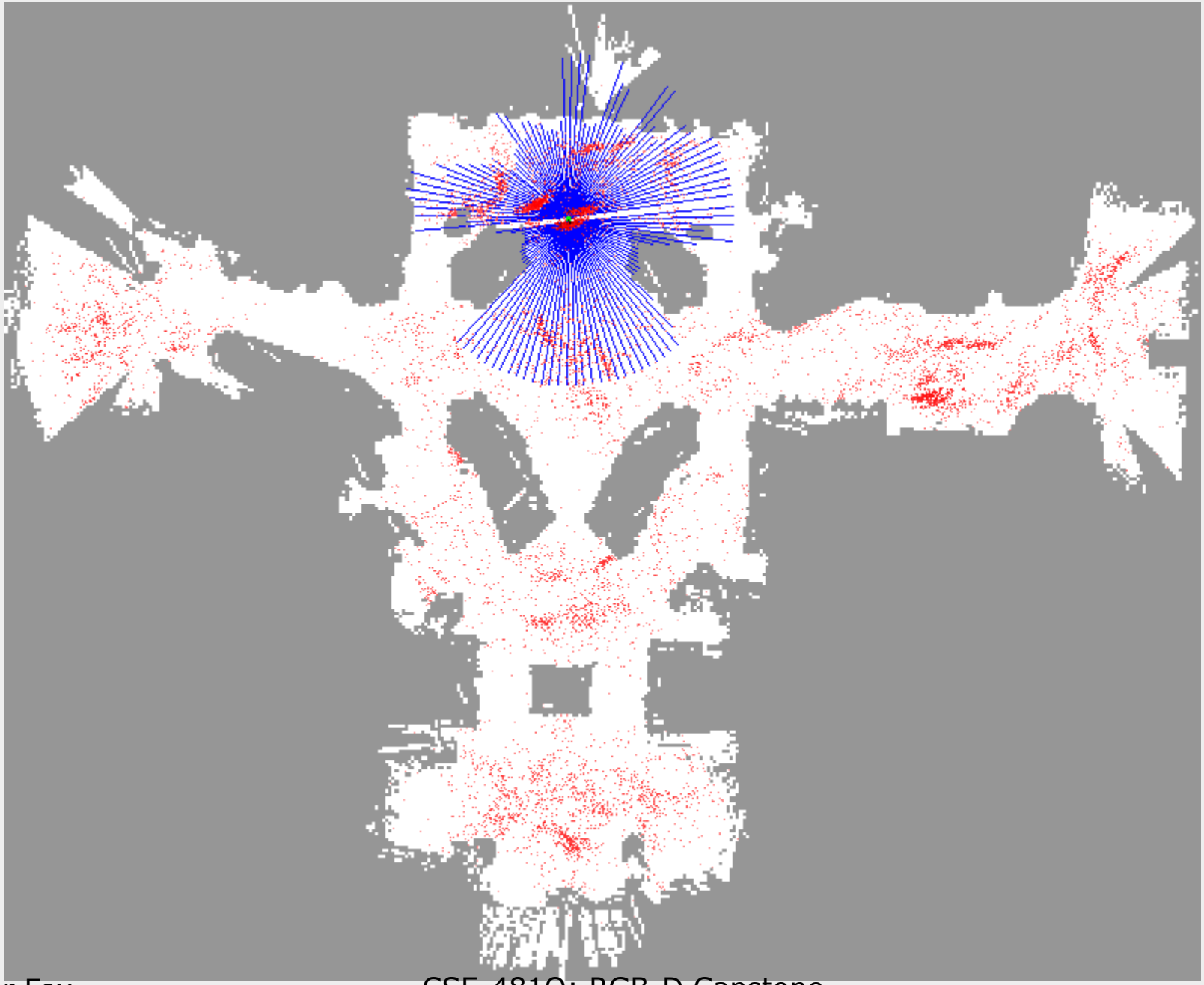


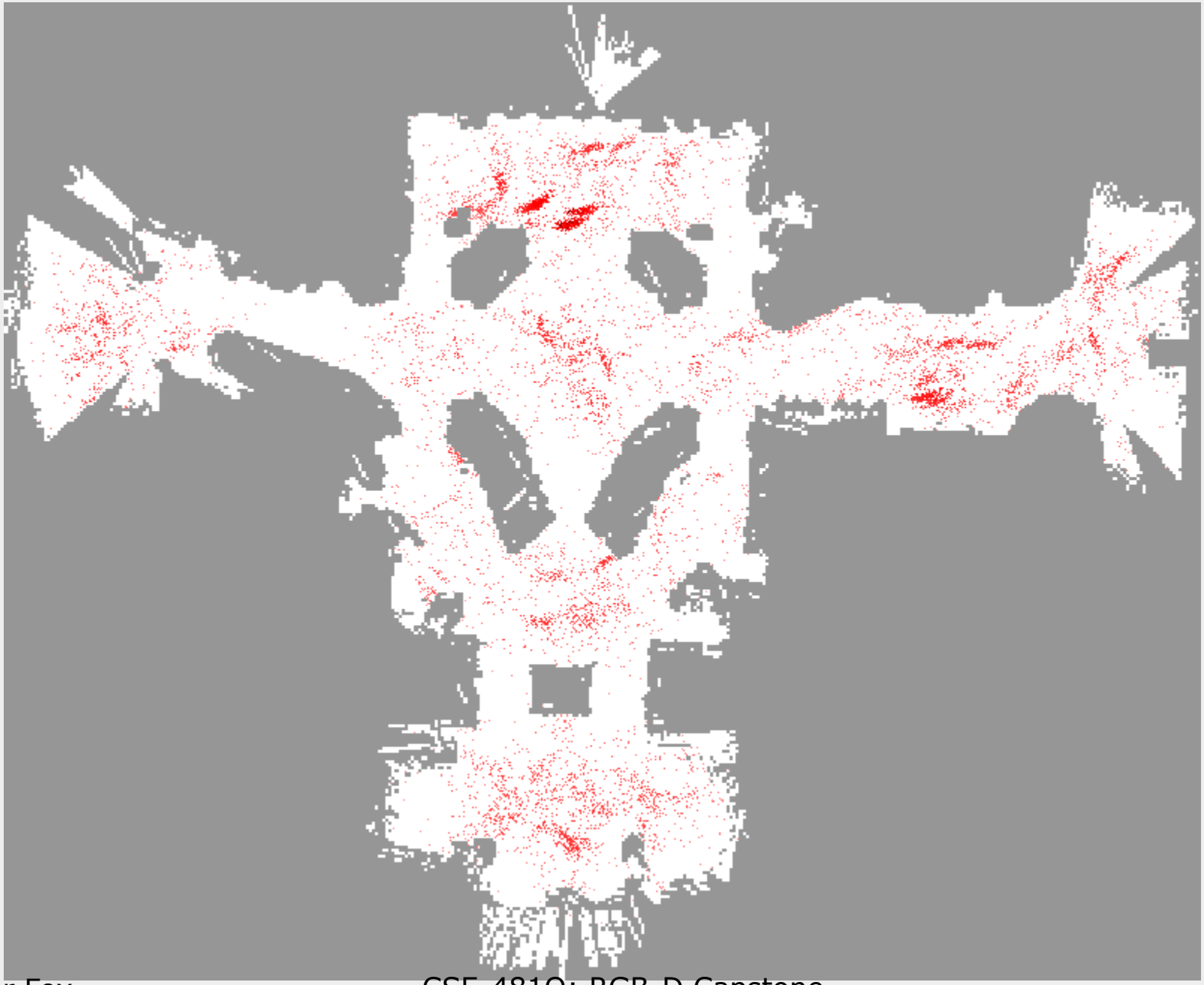


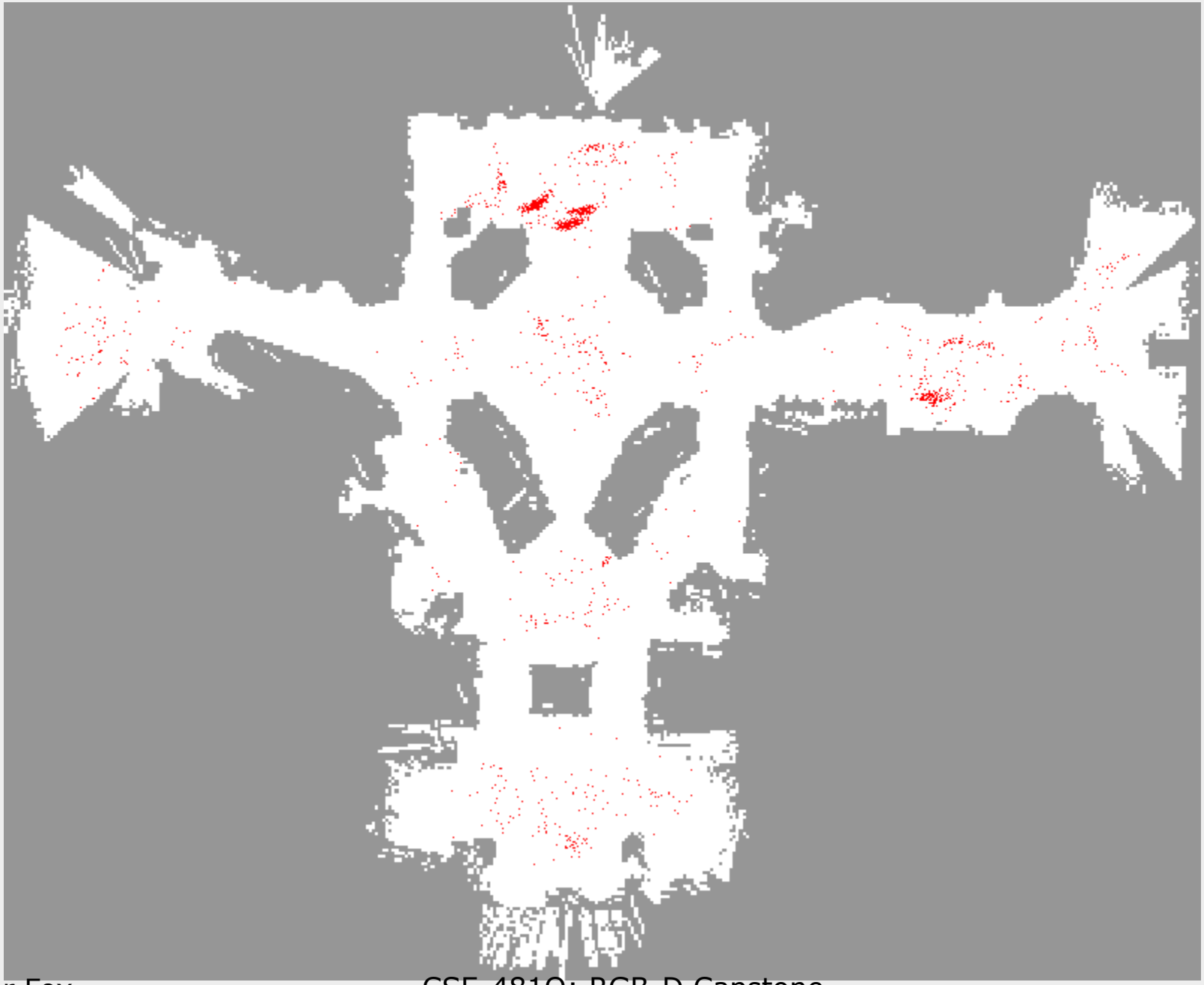


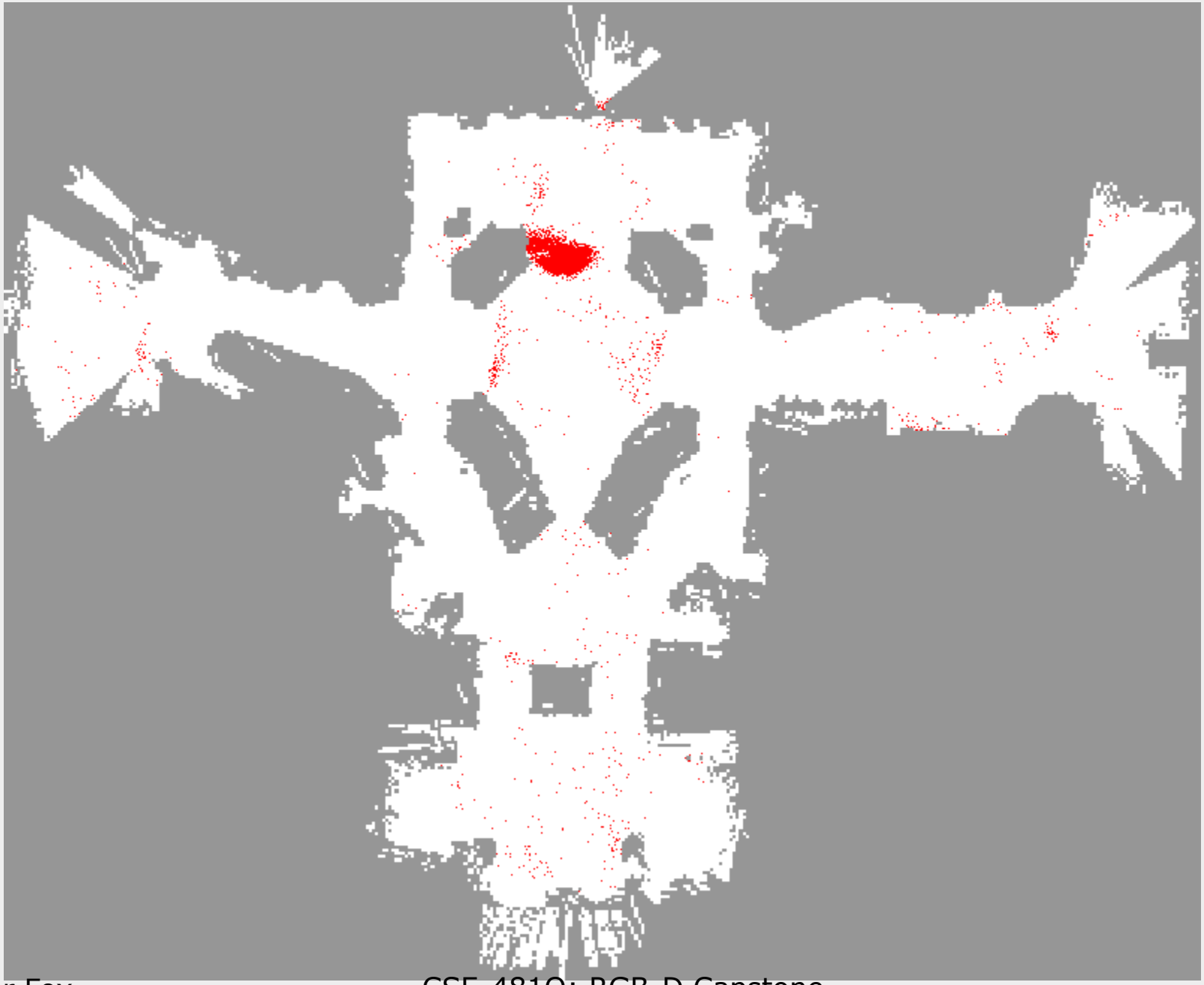


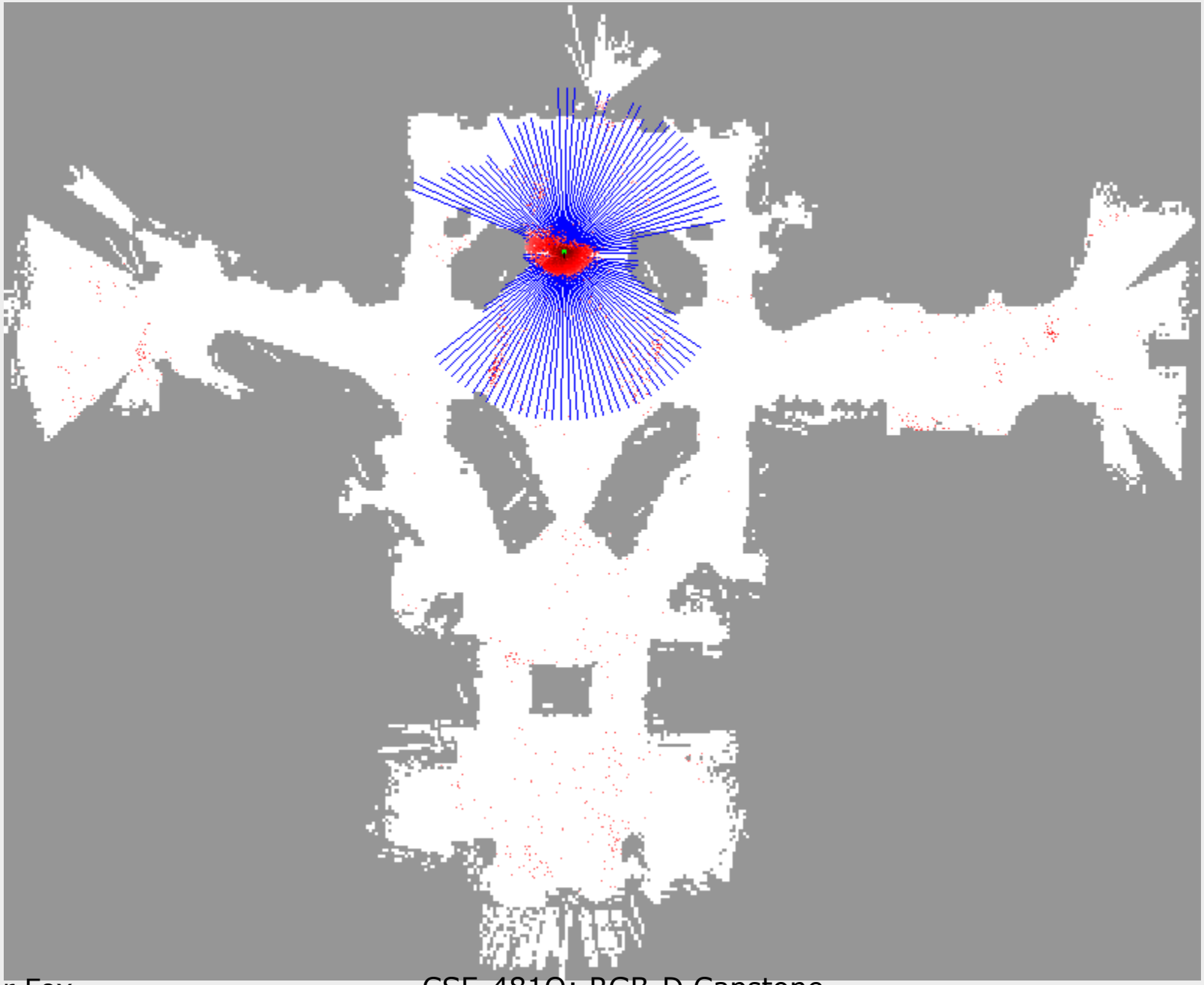


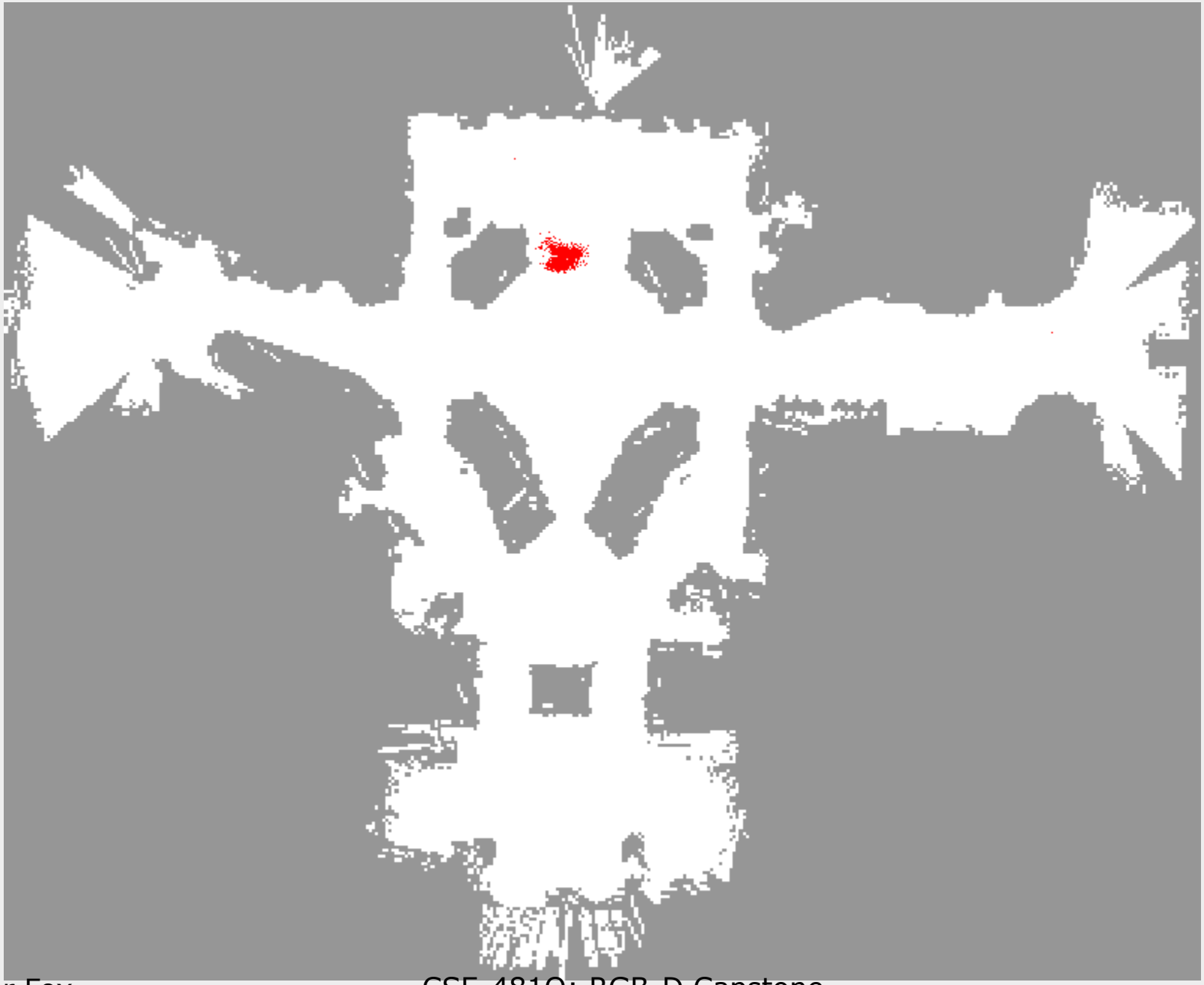


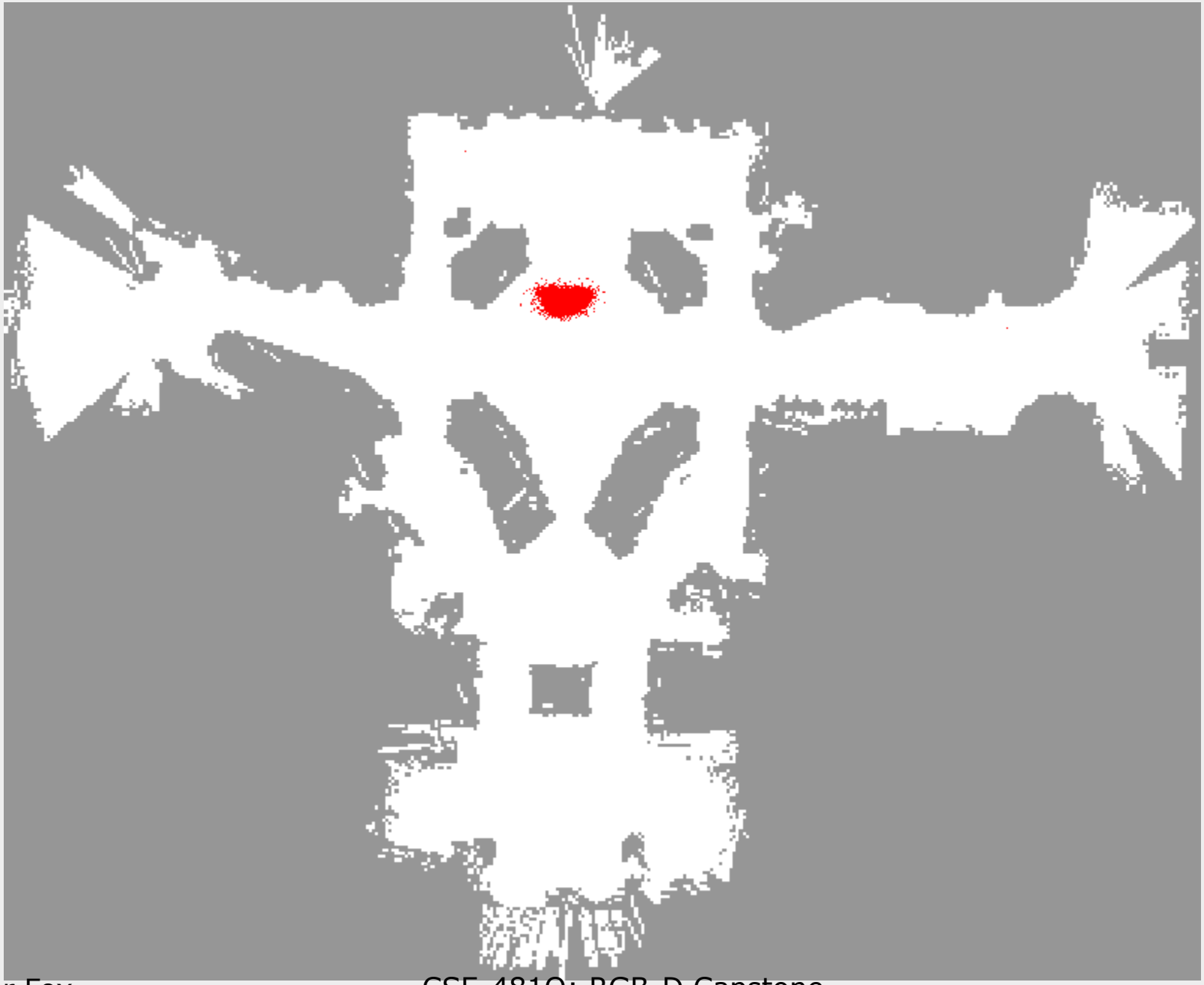


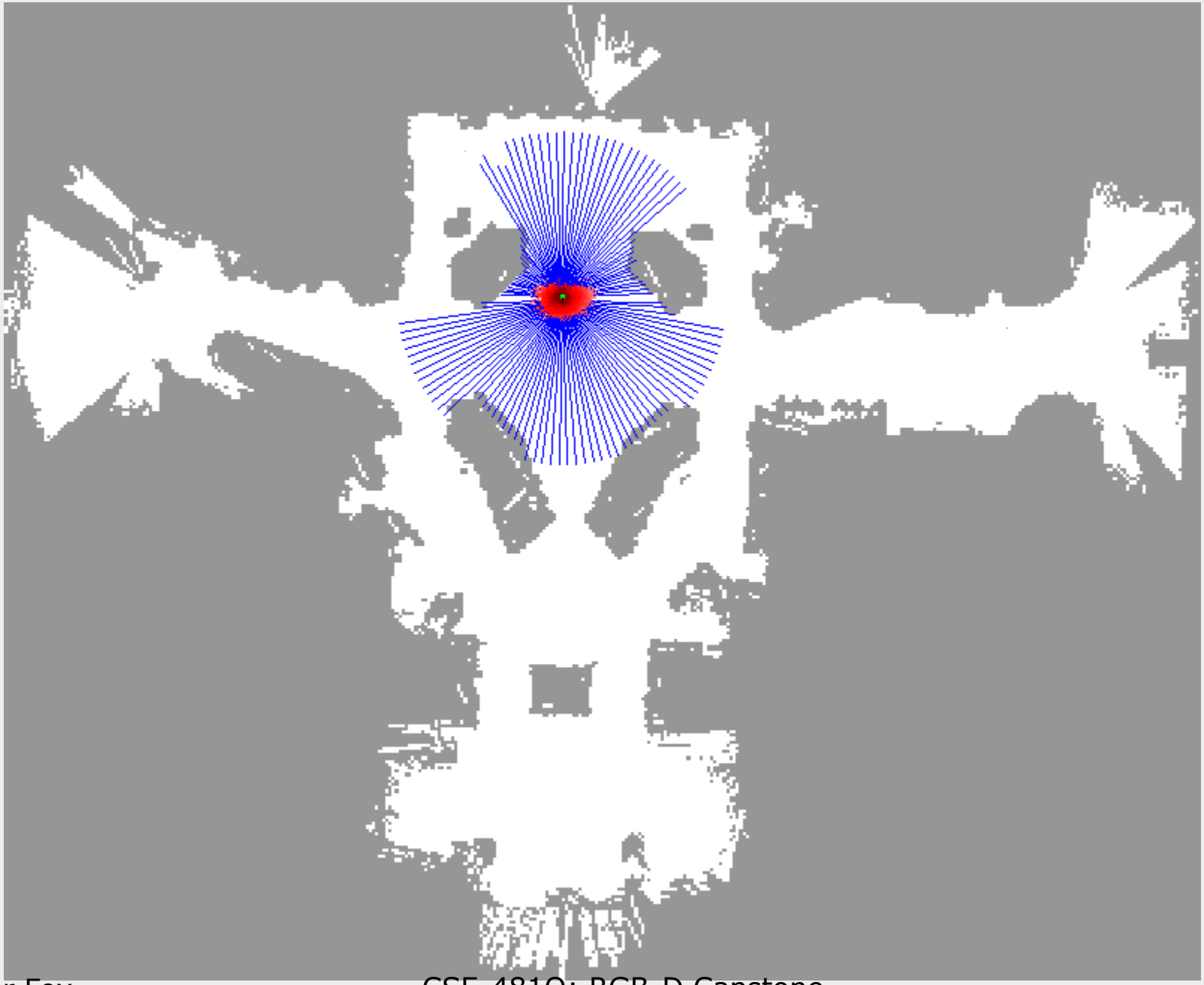


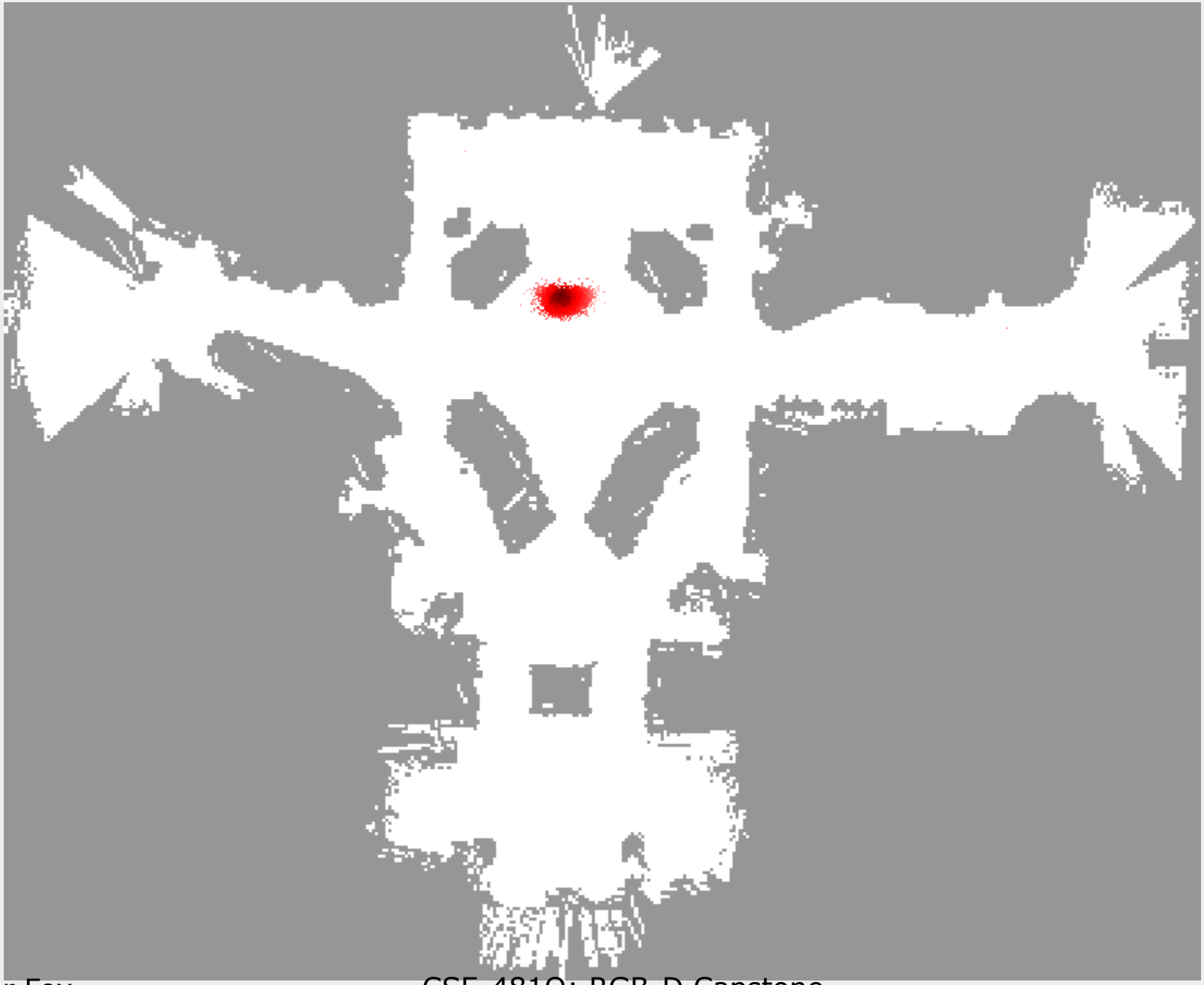


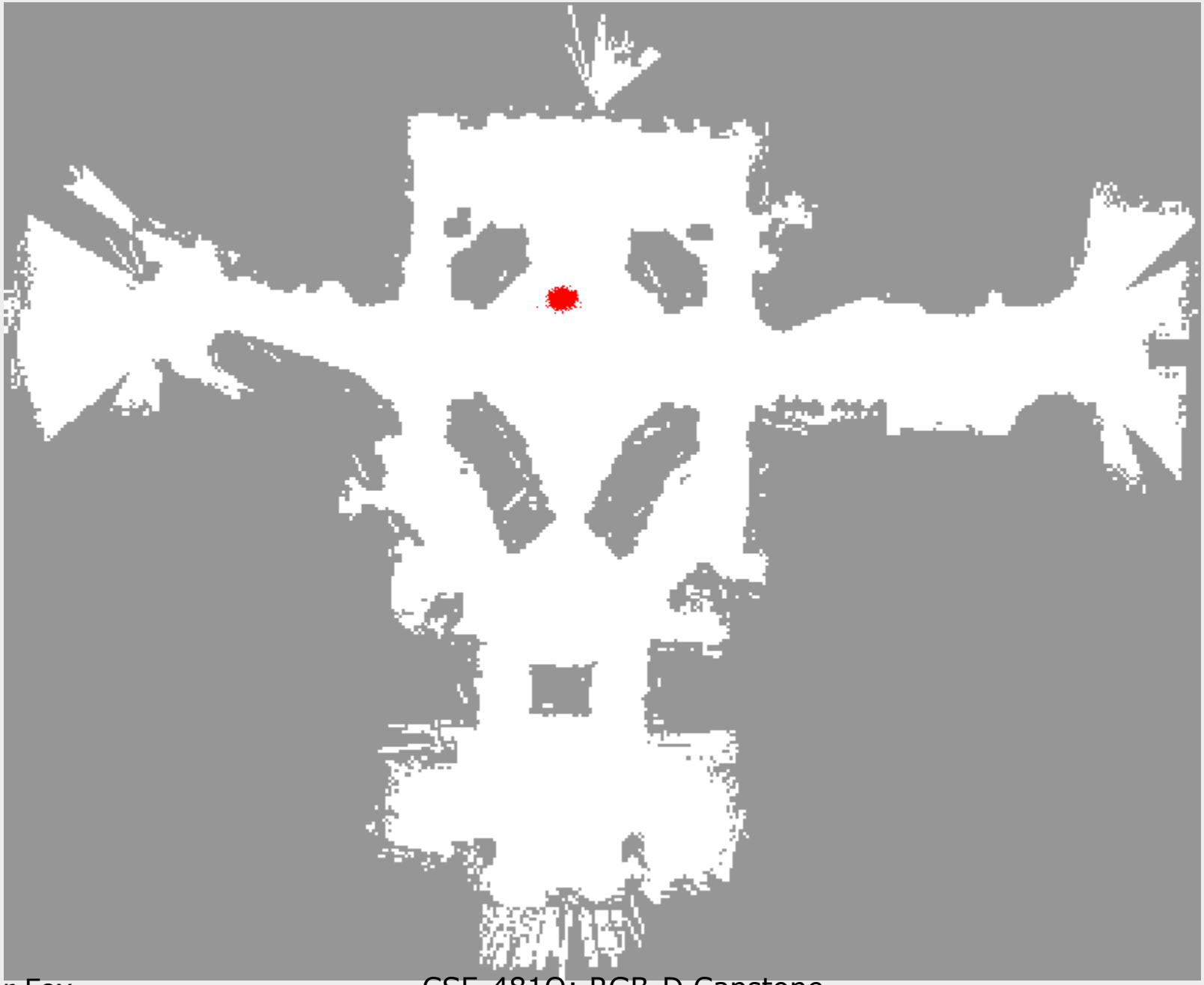


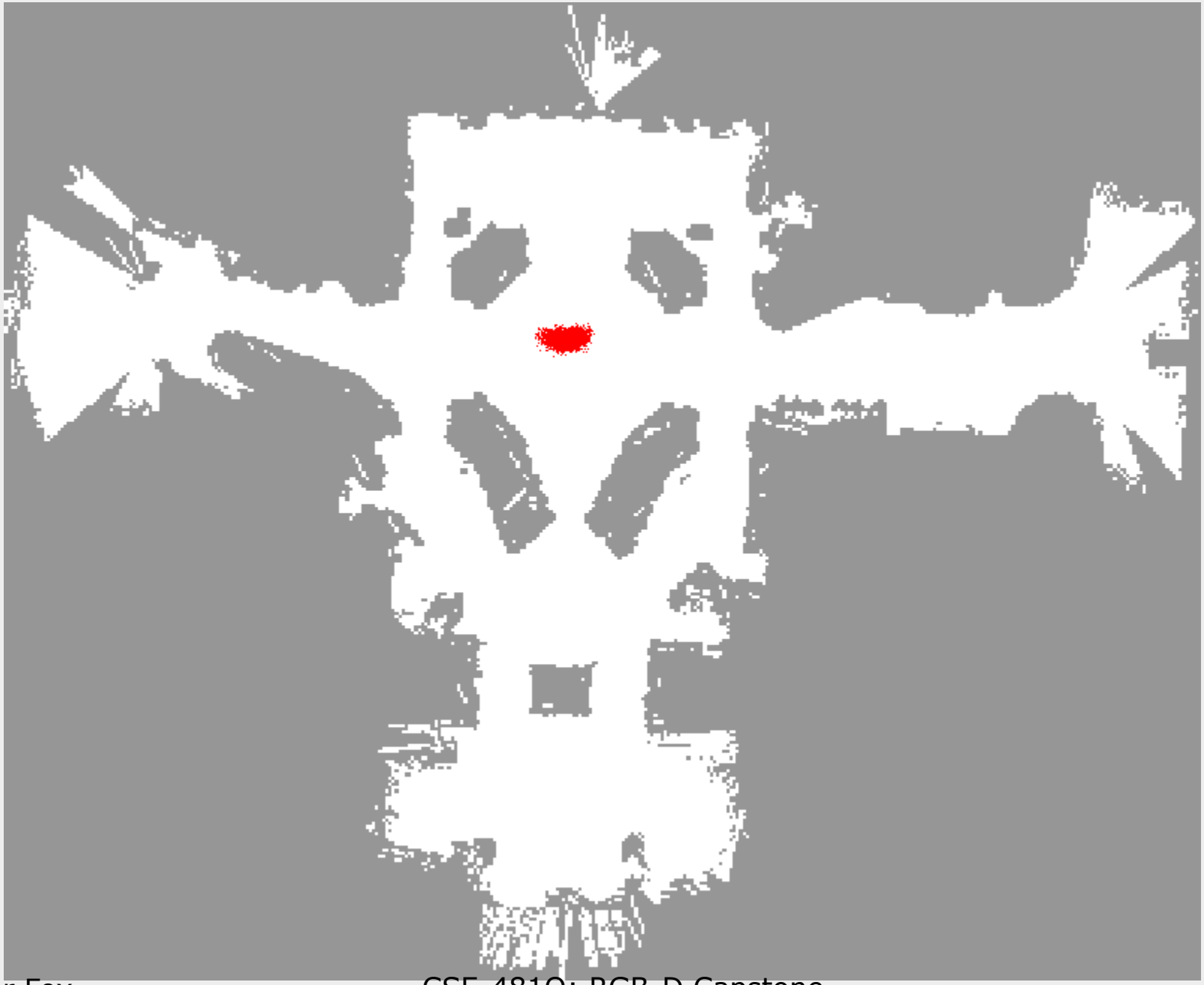


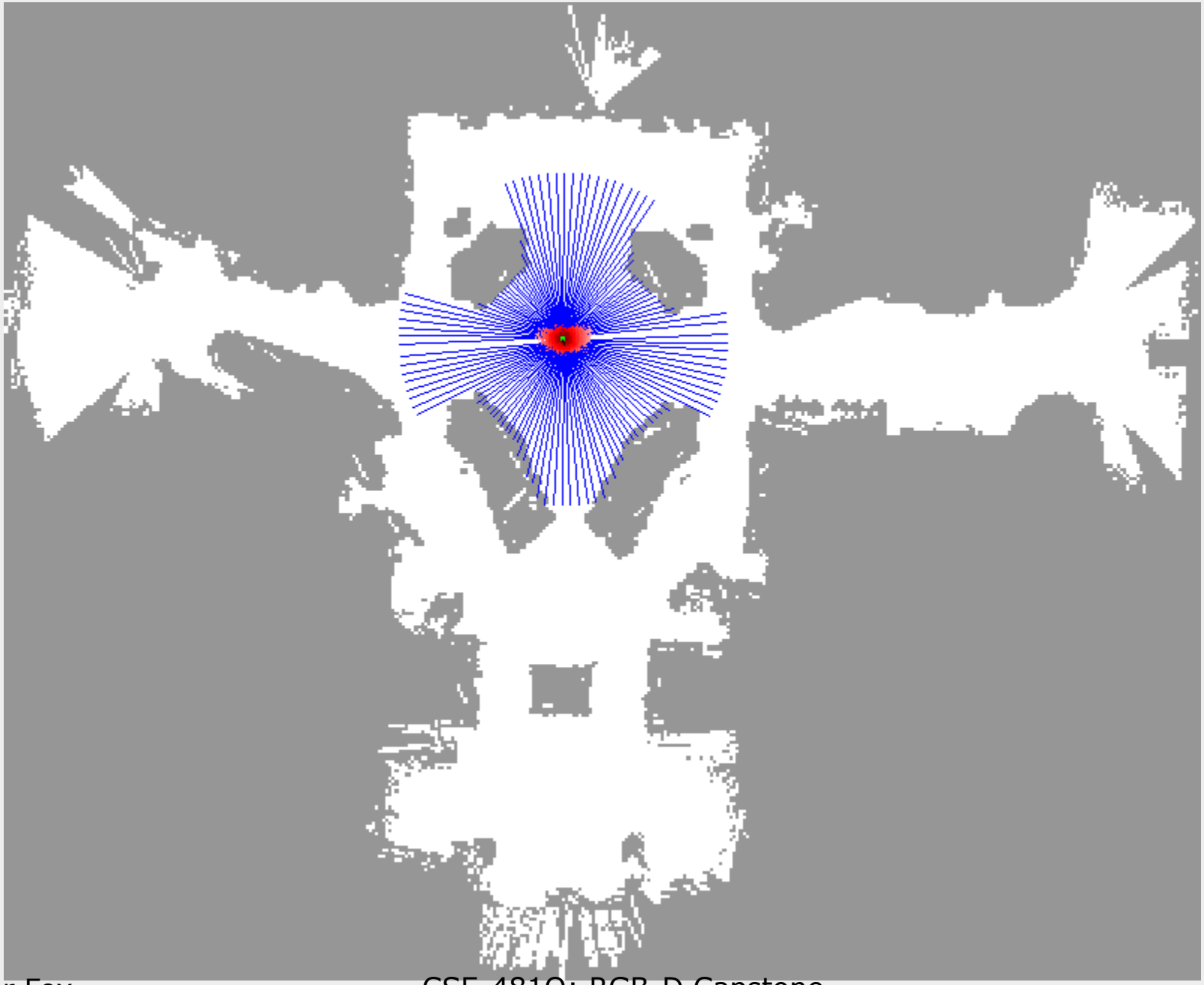


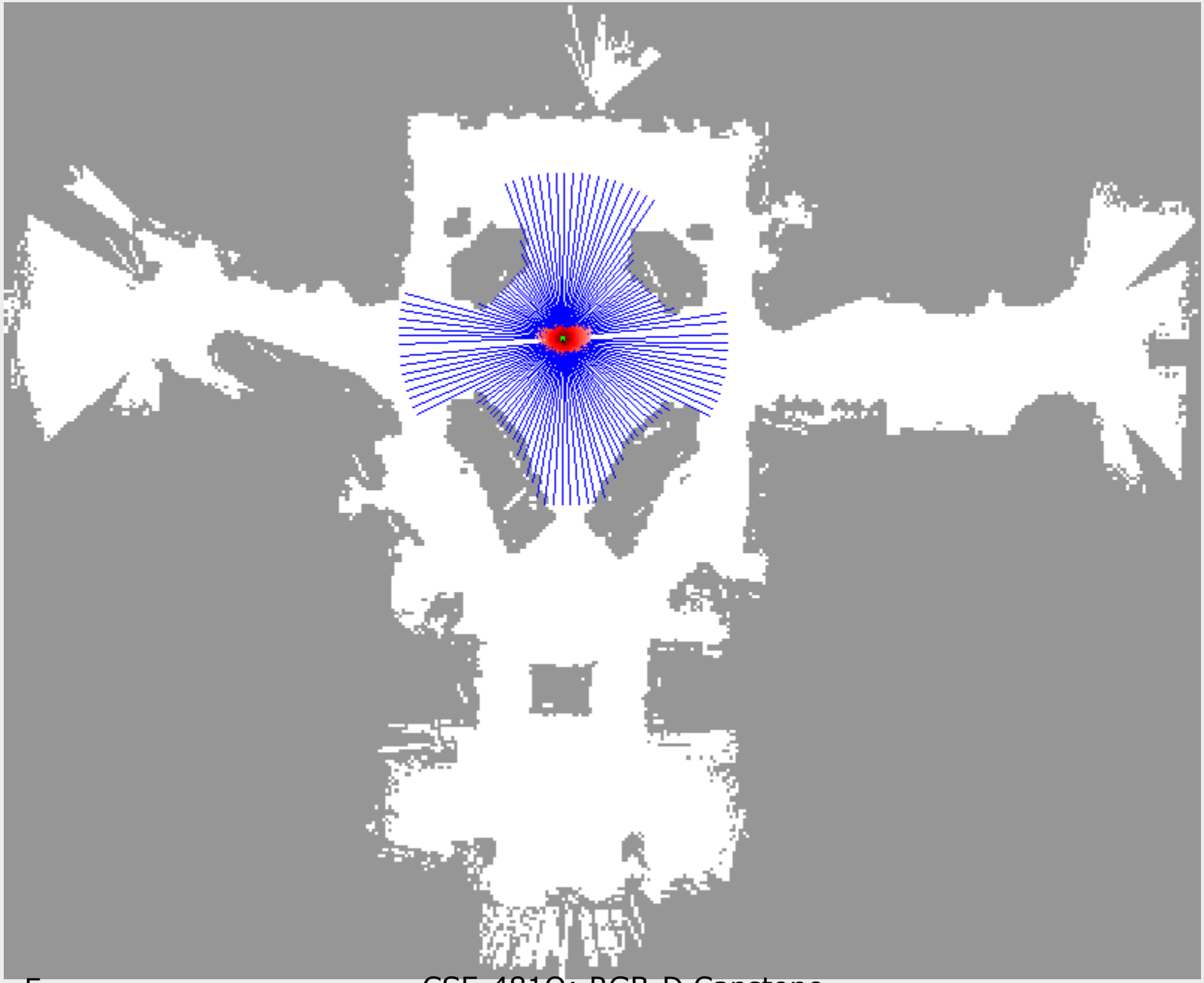




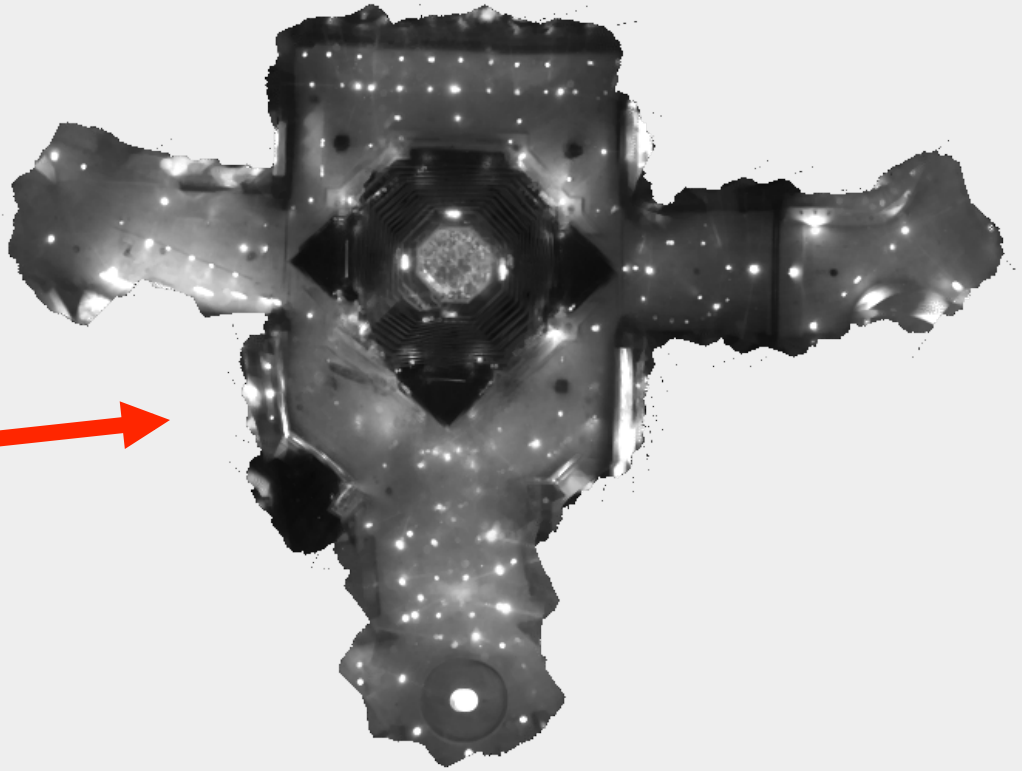






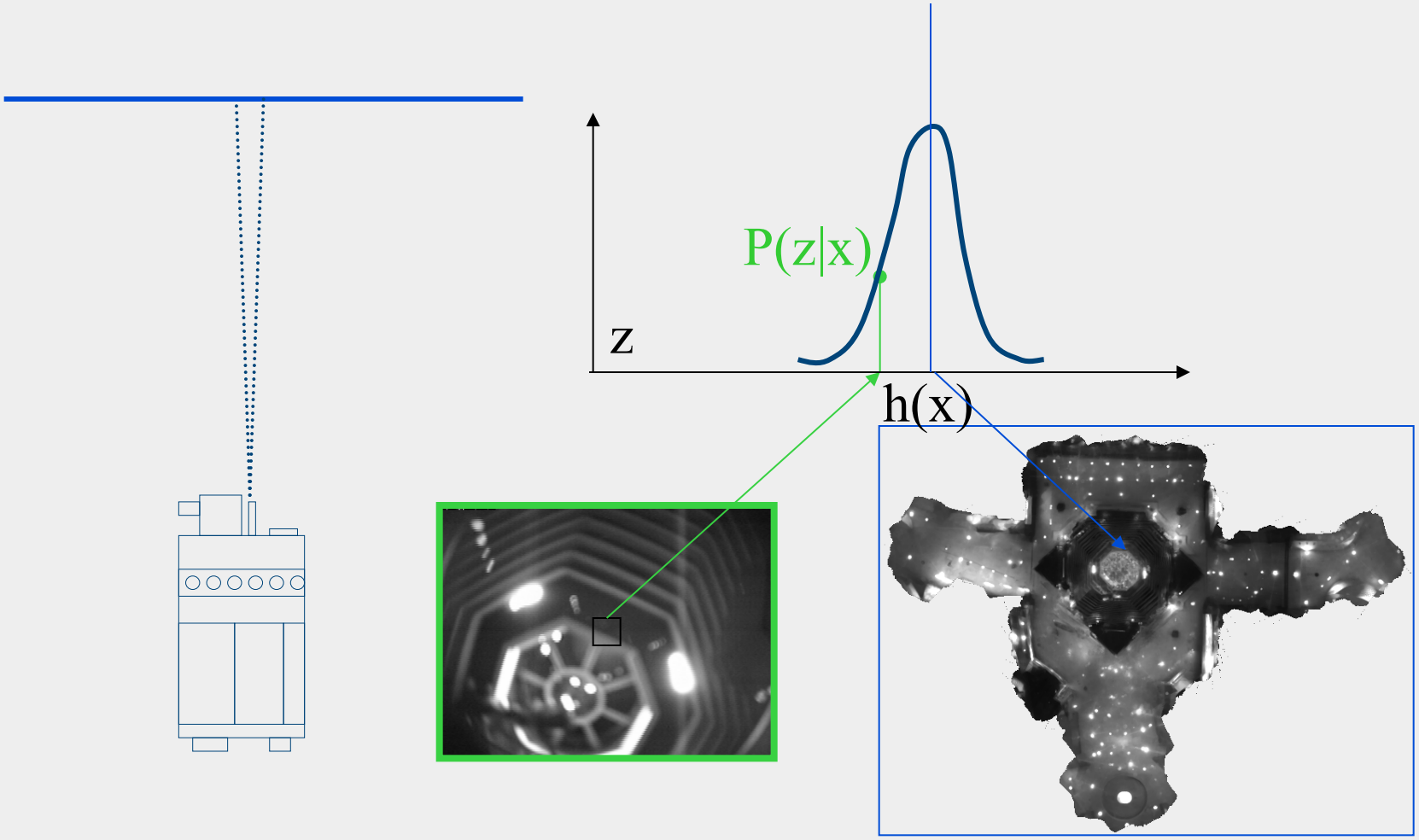


Using Ceiling Maps for Localization



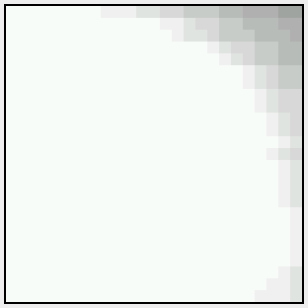
[Dellaert et al. 99]

Vision-based Localization

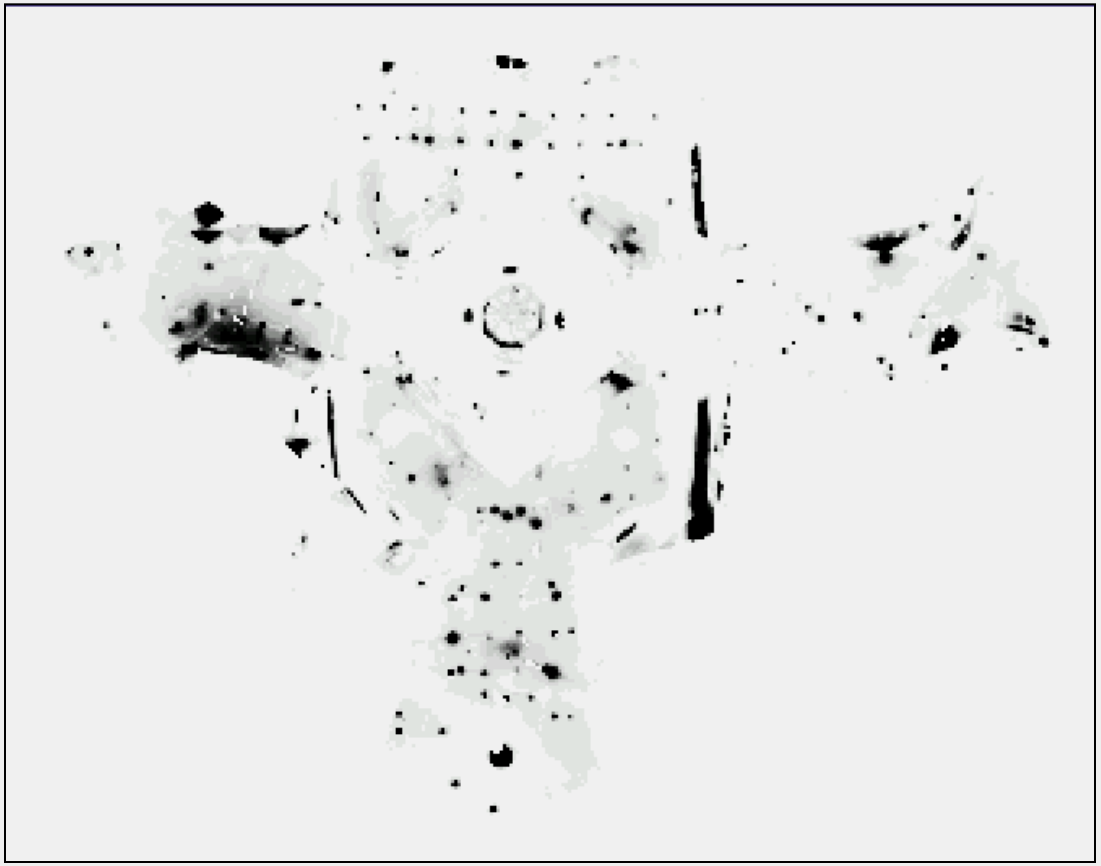


Under a Light

Measurement z:

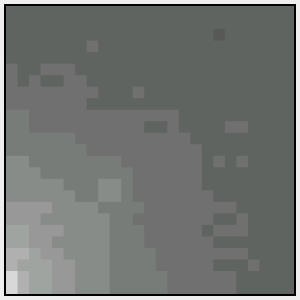


***P(z|x)*:**

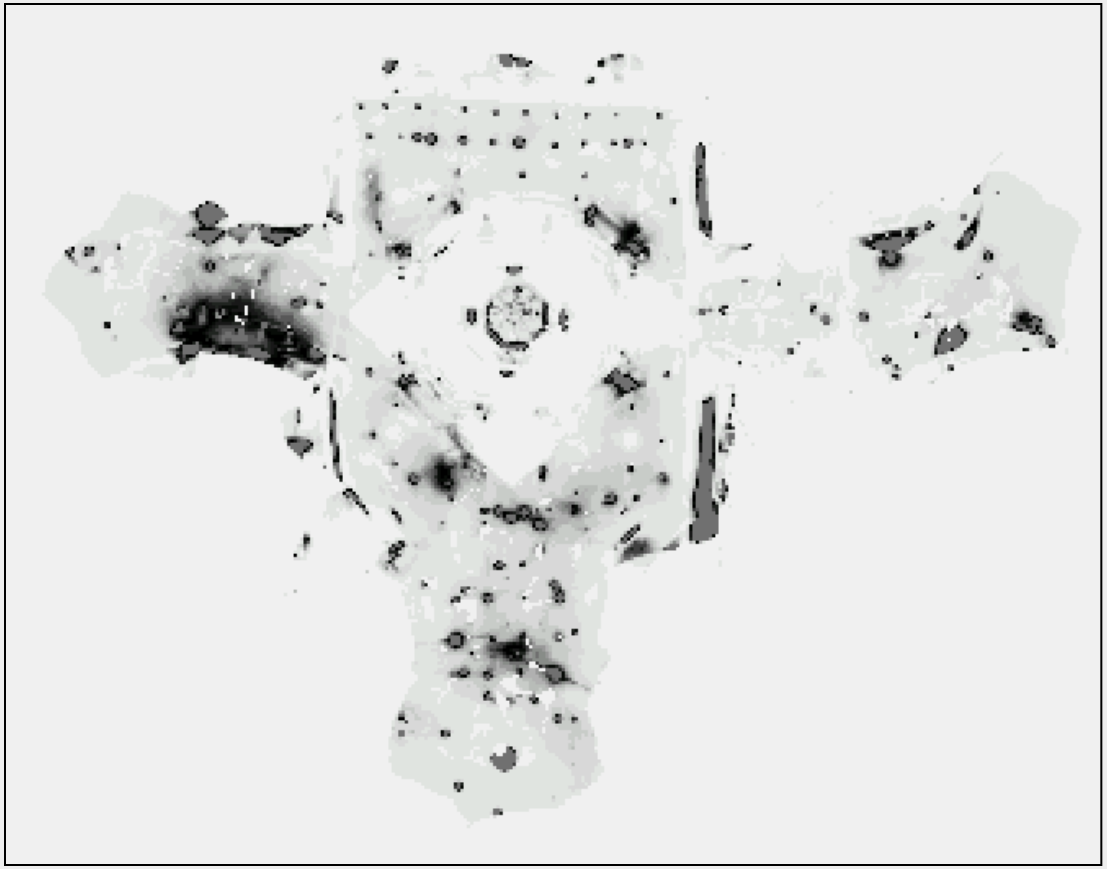


Next to a Light

Measurement z:



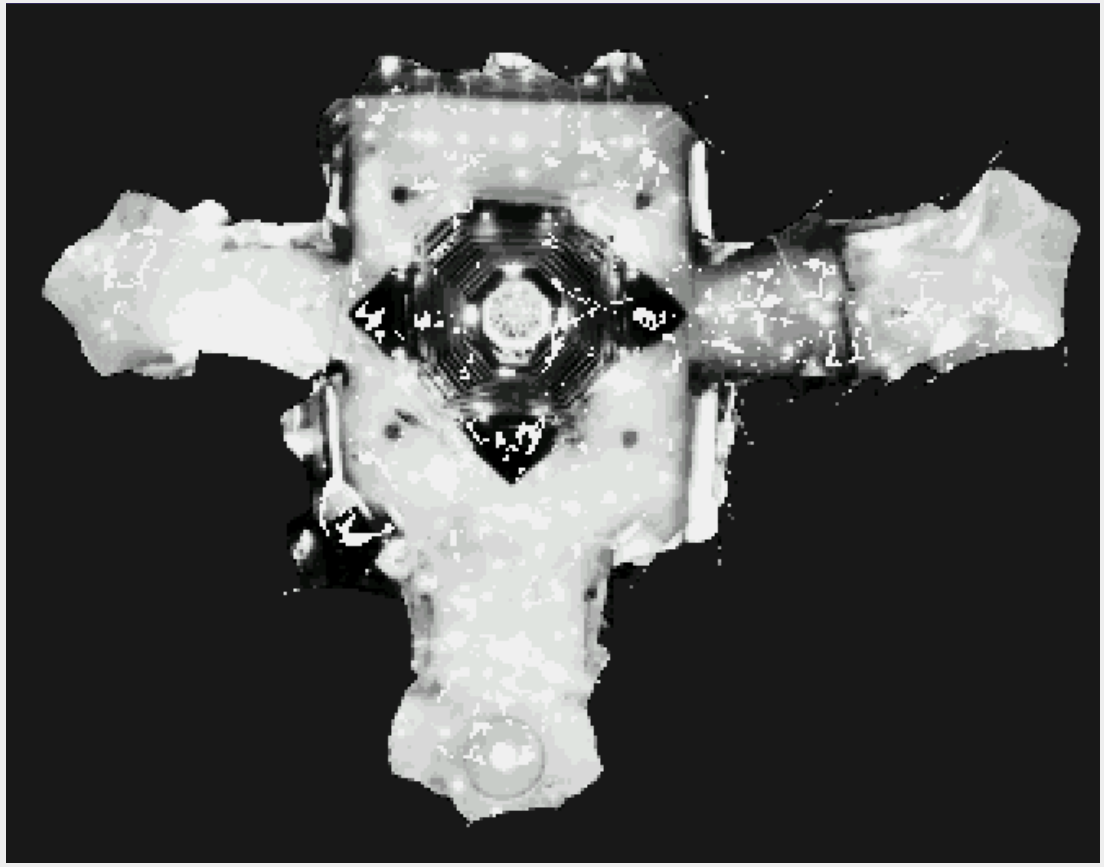
***P(z|x)*:**



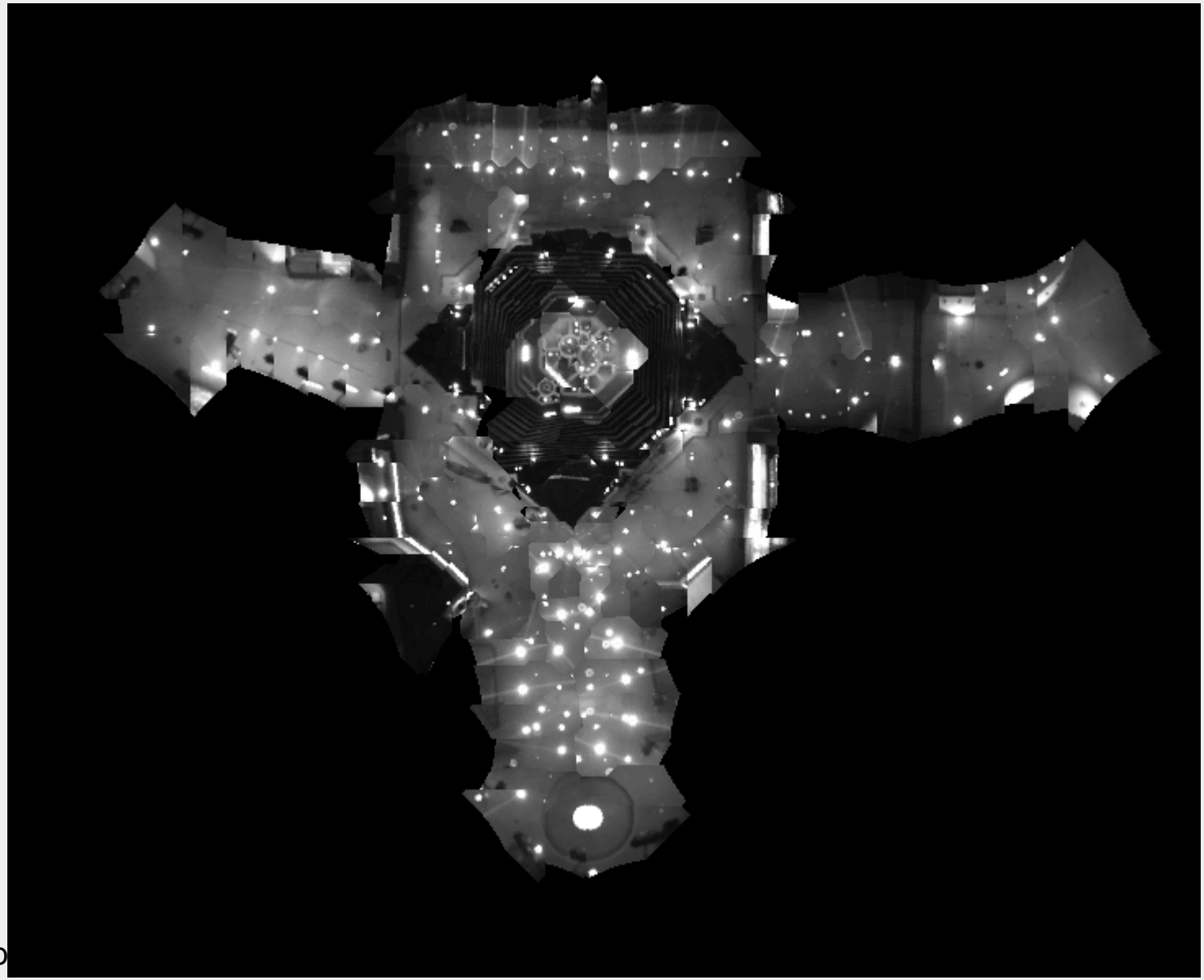
Elsewhere

Measurement z:

***P(z|x)*:**



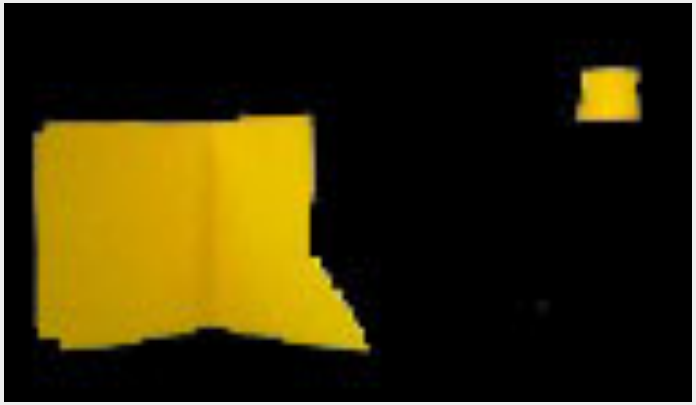
Global Localization Using Vision

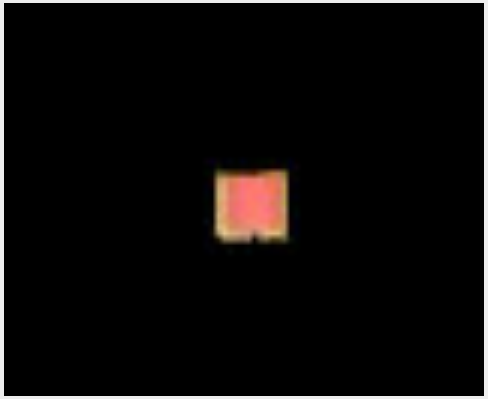


Localization for AIBO robots

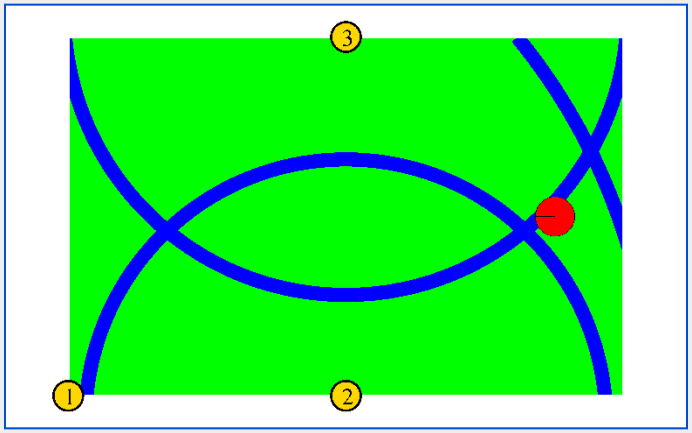
From Images to Objects

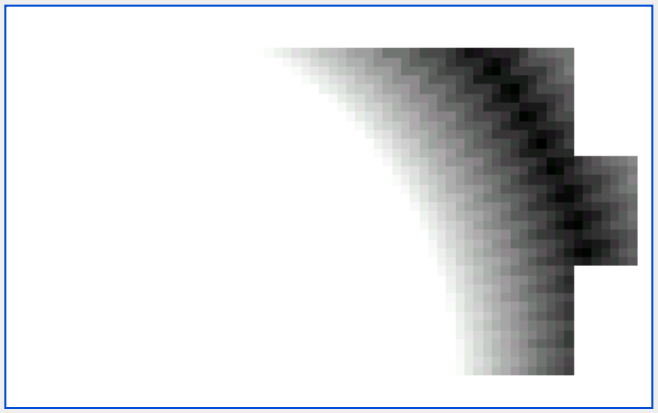
Approach: Extract relevant colors

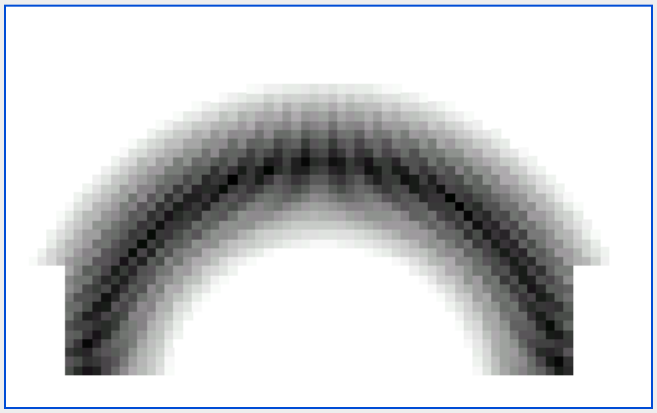


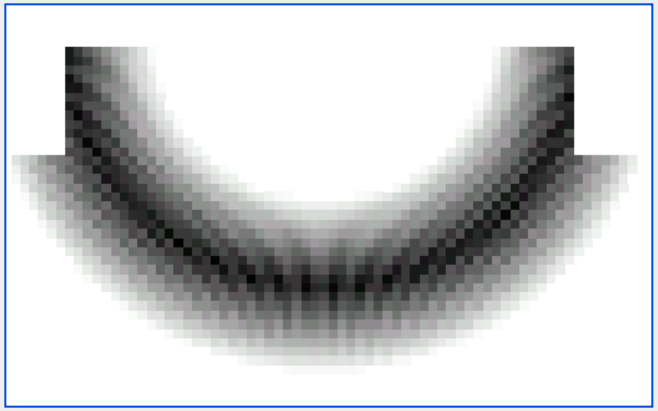


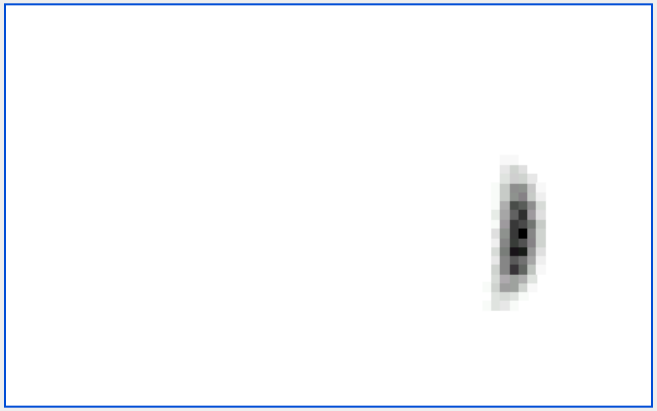
Distributions for P(z|x)



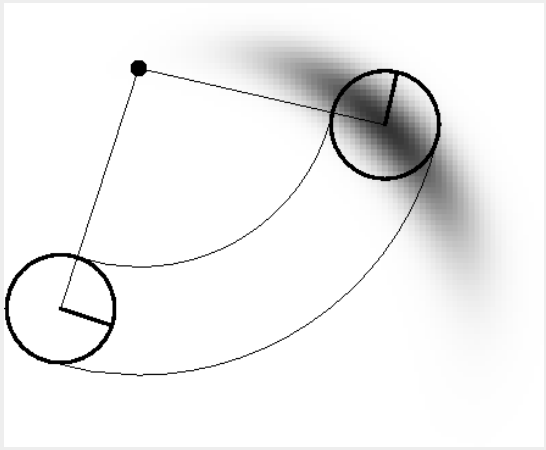
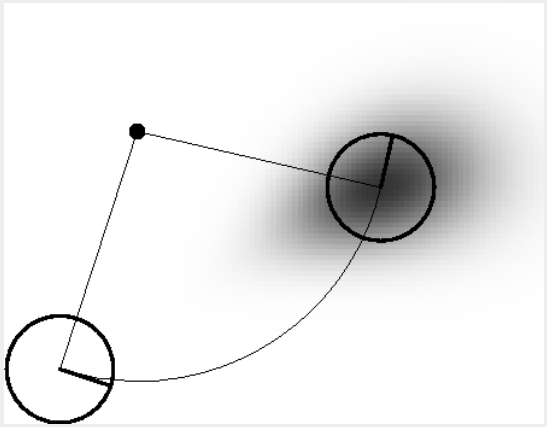
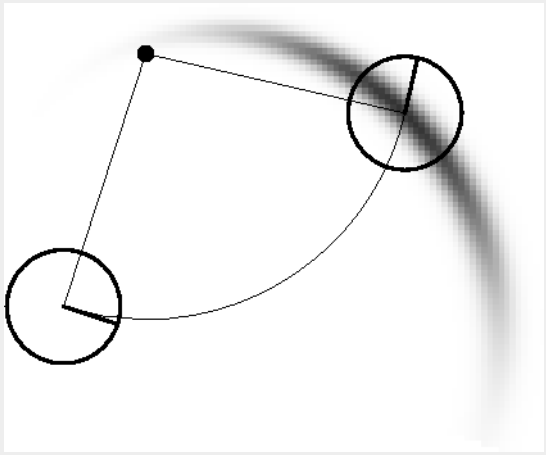
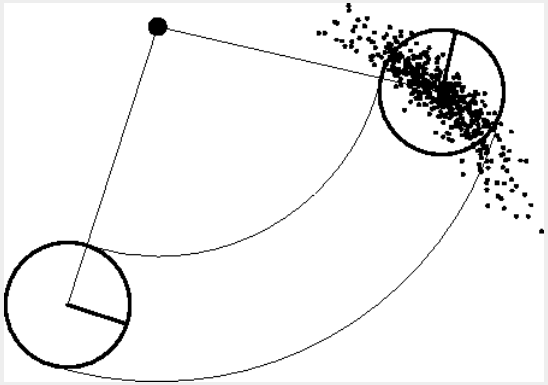
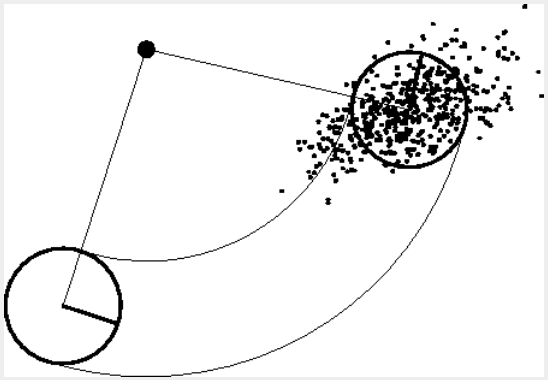
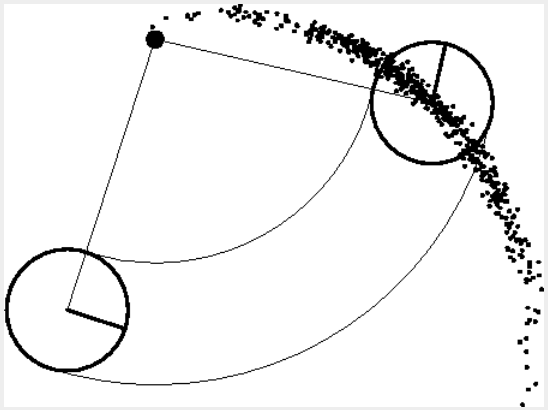




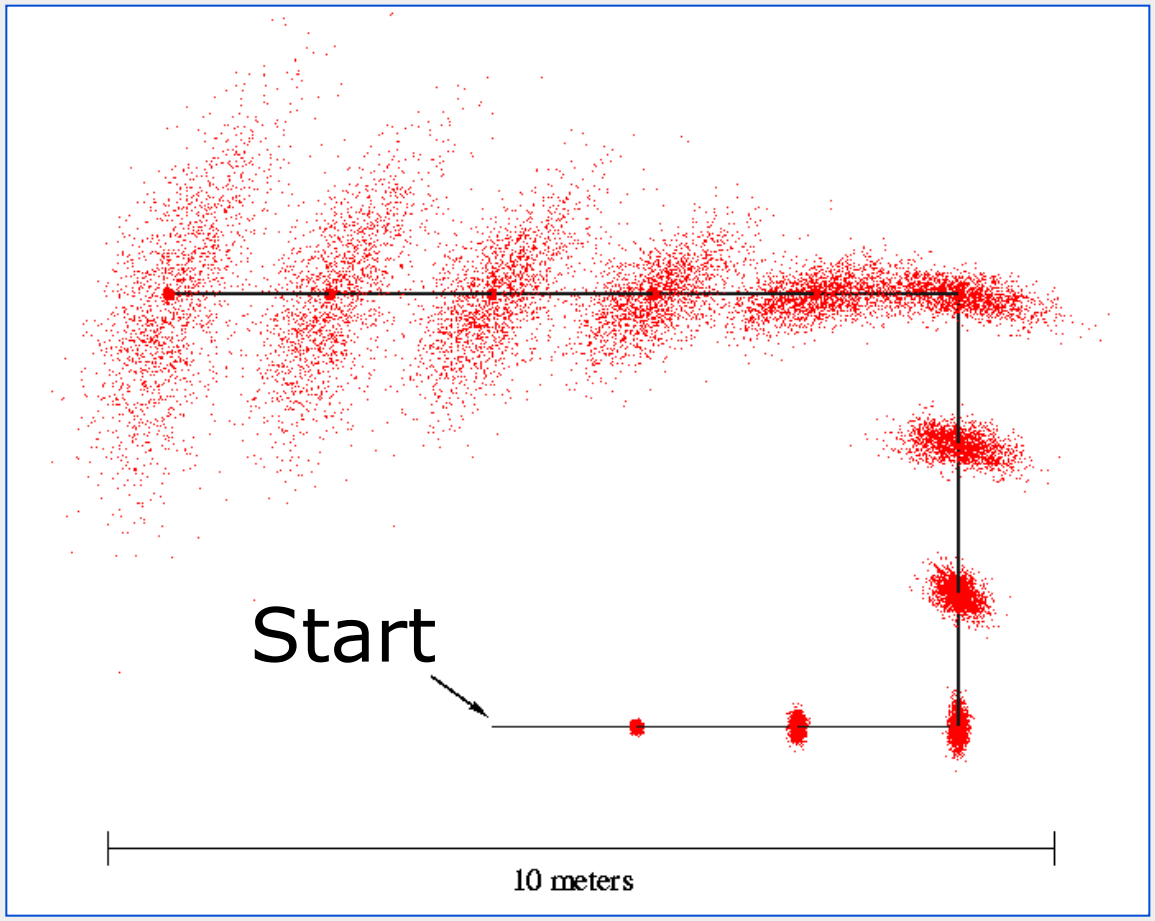




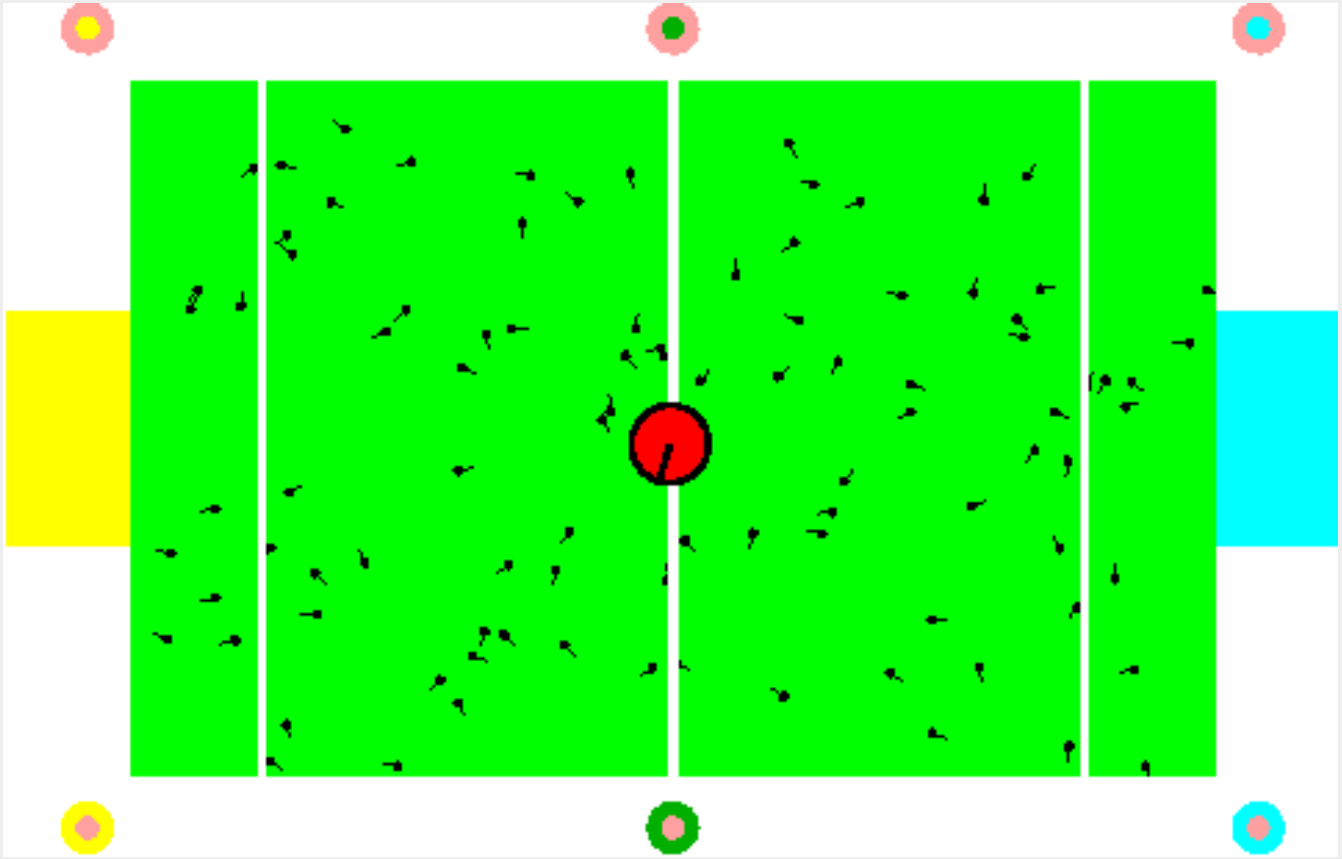
Velocity Based Motion Model

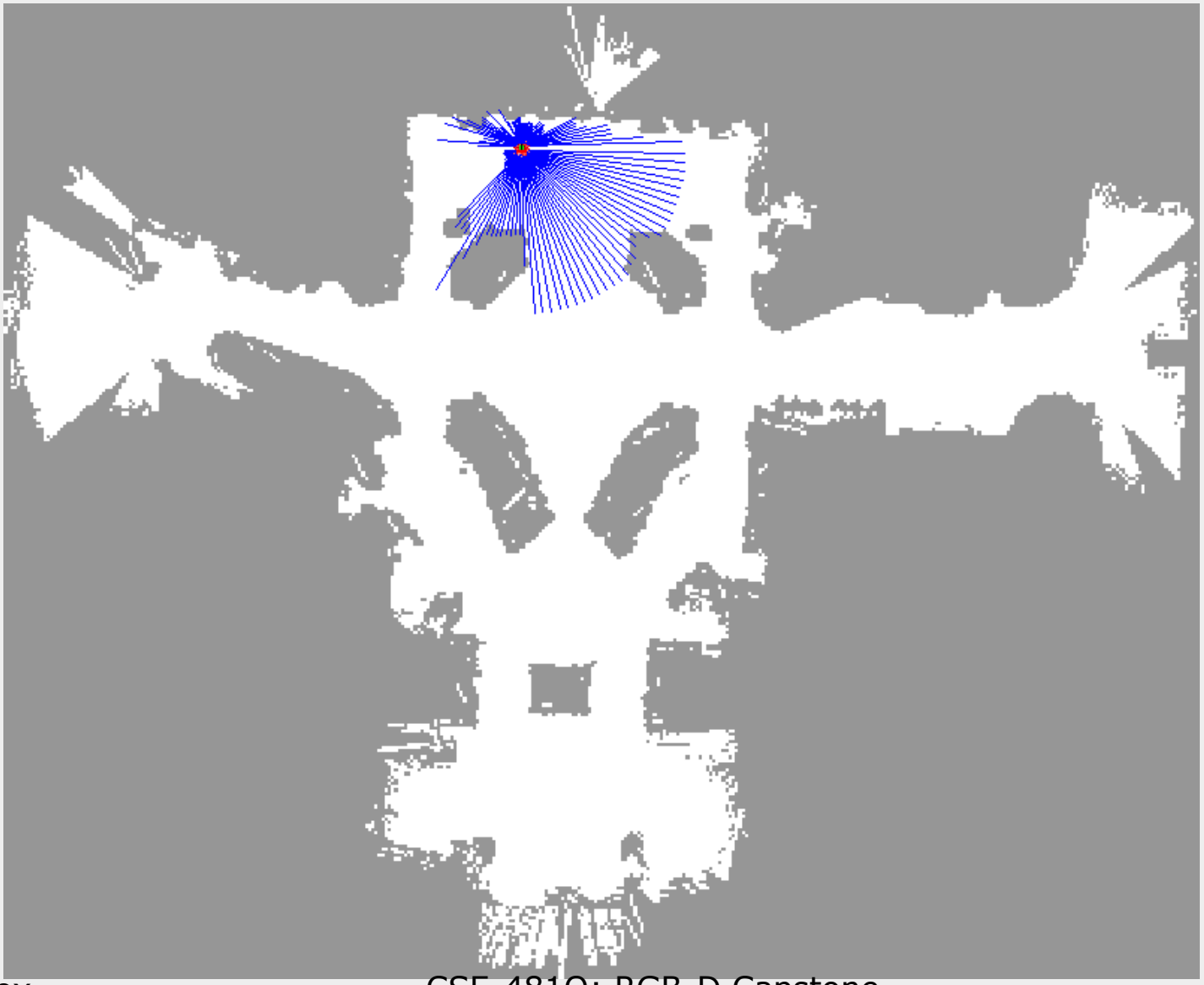
Multi-Step Motion



Example

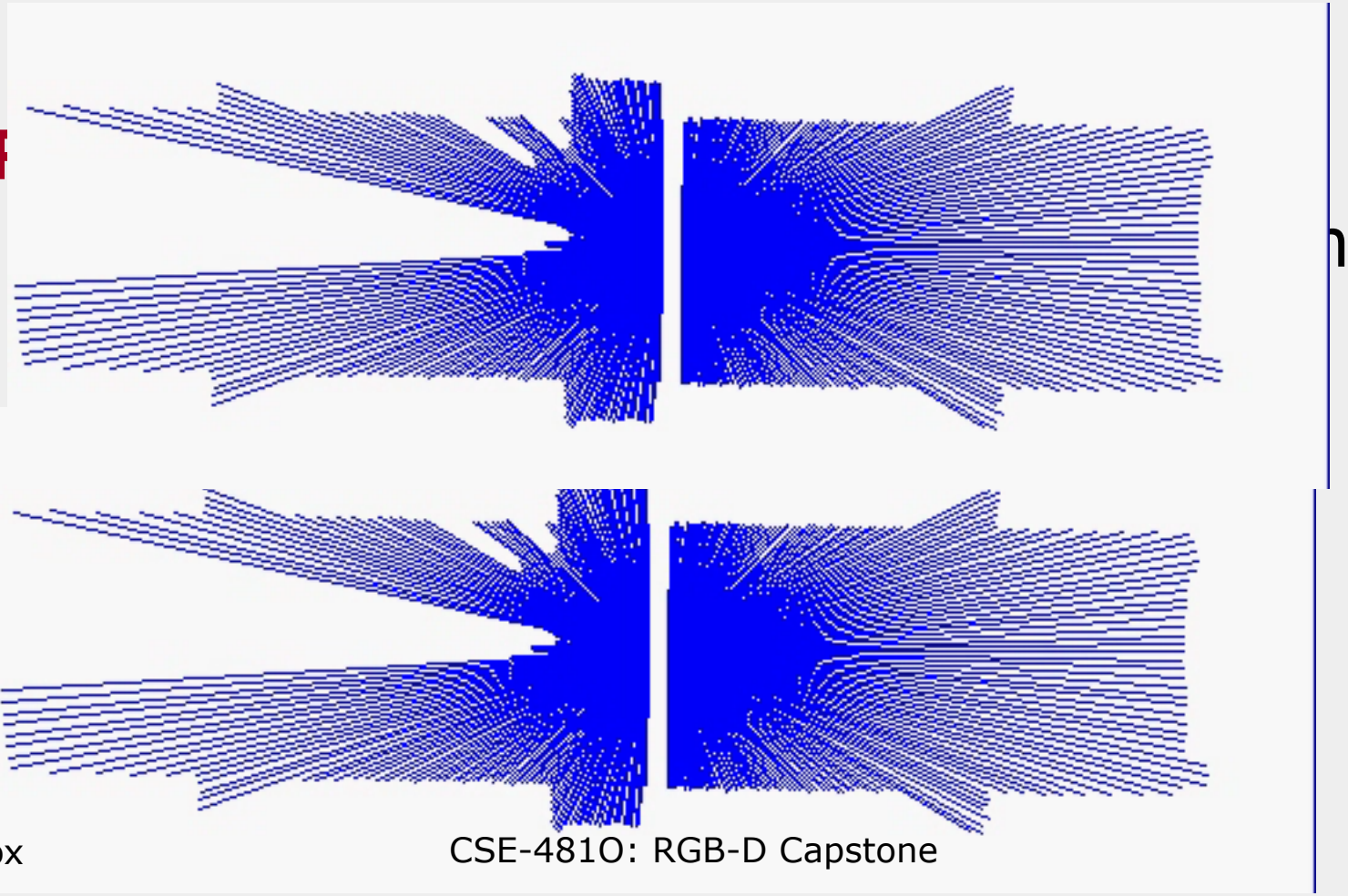


Recovery from Failure



Laser-Based People Tracking

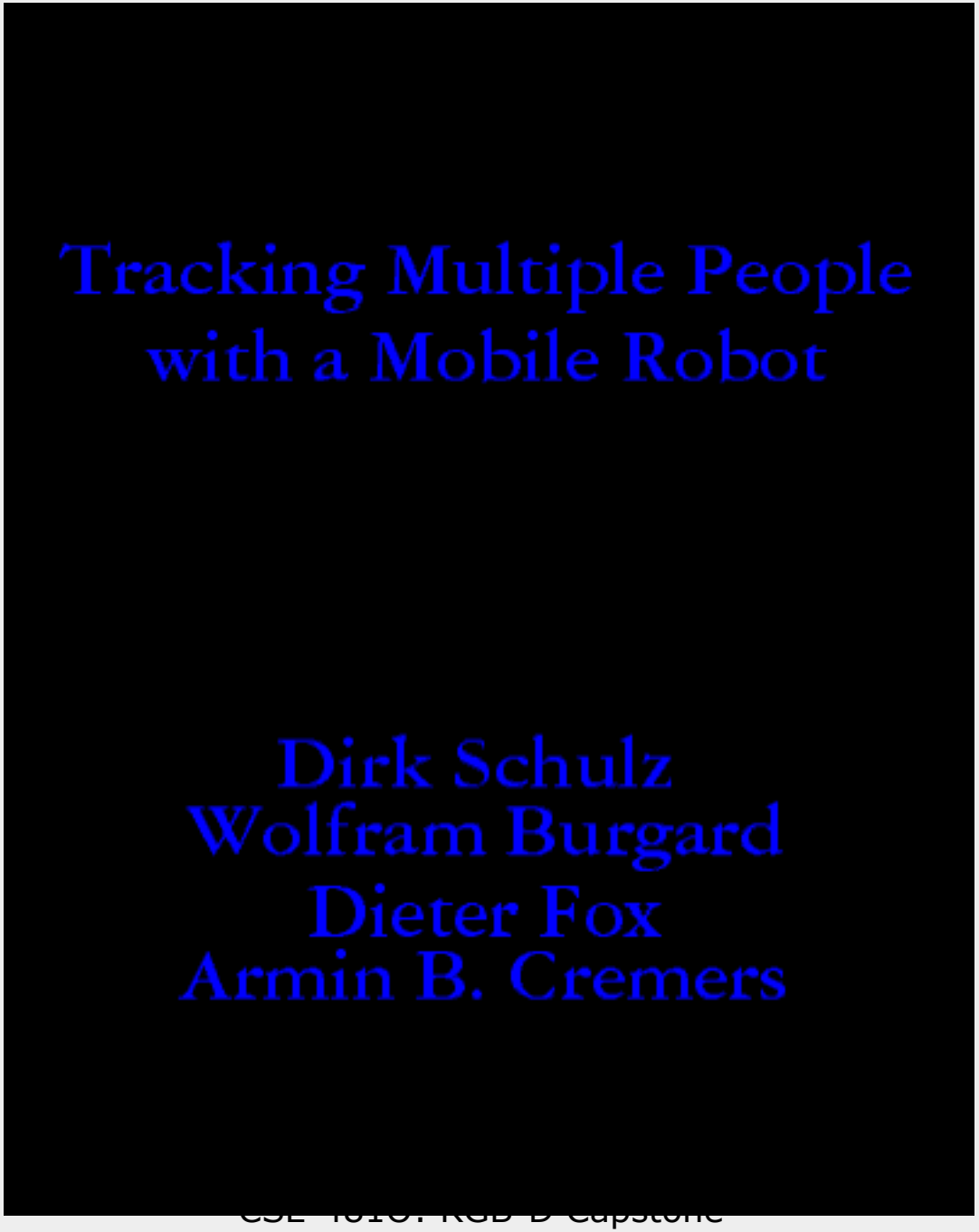
- **Questions**
  - How many people are there?
- **P**



nent

Experimental Results

- B21 robot
- two LRF (SICK PLS)
- 360 degree FOV
- angular resolution 1 degree
- perceptual range 8 m
- grid-resolution 10 cm


Tracking Multiple People
with a Mobile Robot
Dirk Schulz
Wolfram Burgard
Dieter Fox
Armin B. Cremers

Tracking with a Moving Robot

Representations for Bayes Filters

- Kalman Filter
  - Highly efficient, robust
  - Uni-modal, limited handling of nonlinearities
- Particle Filter
  - Less efficient, highly robust
  - Multi-modal, nonlinear, non-Gaussian
- Rao-Blackwellised Particle Filter, MHT
  - Combines PF with KF
  - Multi-modal, highly efficient

Task

[Liao-Fox-Kautz: AAAI-04, AIJ-07]

- Given data stream from a wearable GPS unit
  - Infer the user's location and mode of transportation (foot, car, bus, bike, …)
  - Predict where user will go
  - Detect novel behavior / user errors

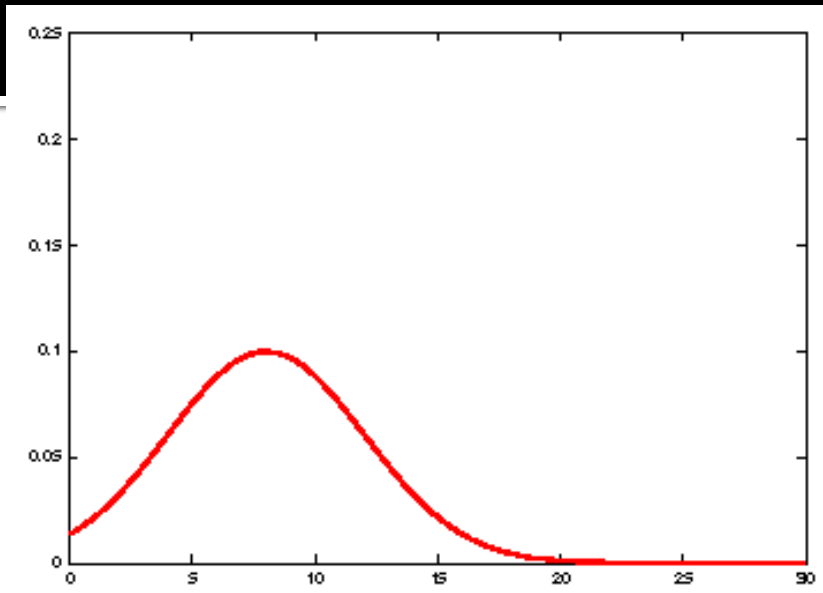
GPS-Tracking Is NOT Trivial

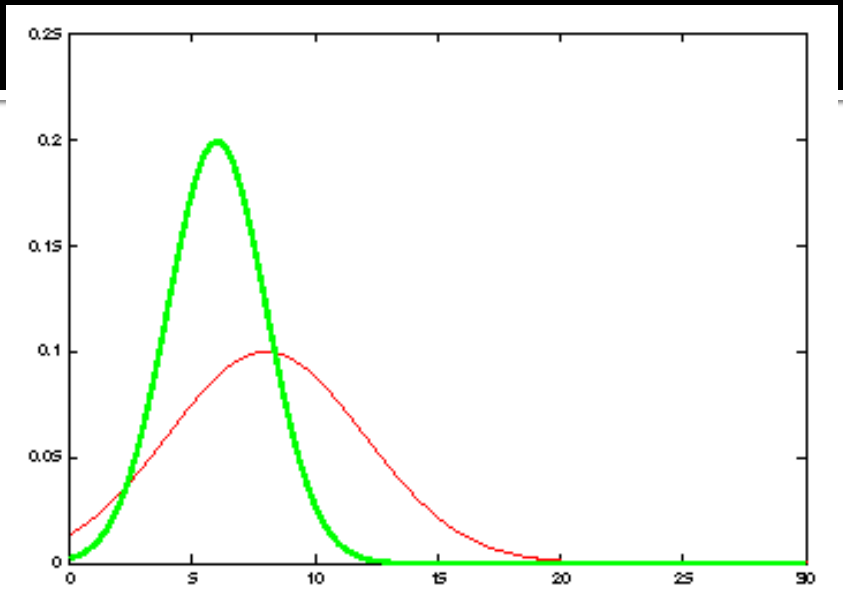
- Dead and semi-dead zones near buildings, trees, *etc*.
- Sparse measurements inside vehicles, especially bus
- Multi-path propagation
- Inaccurate street map
- …

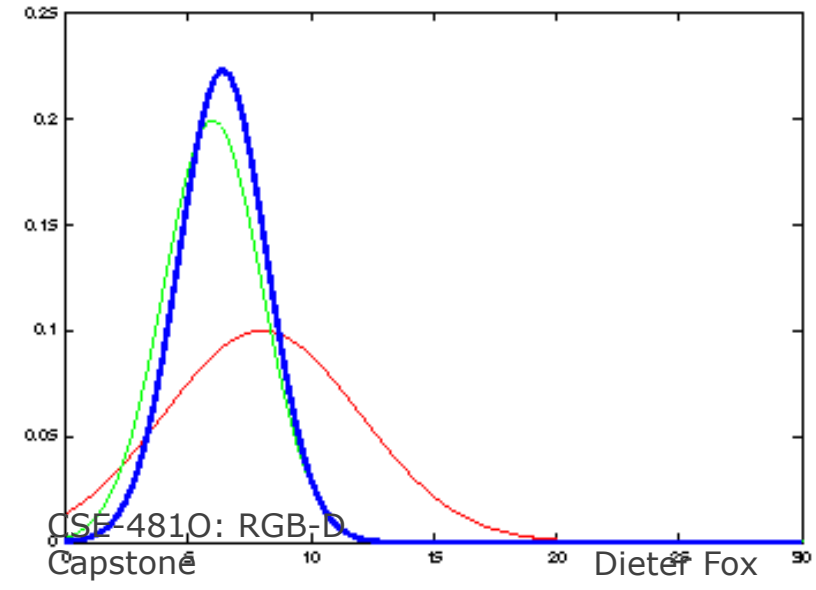
Graph-based Location Estimation

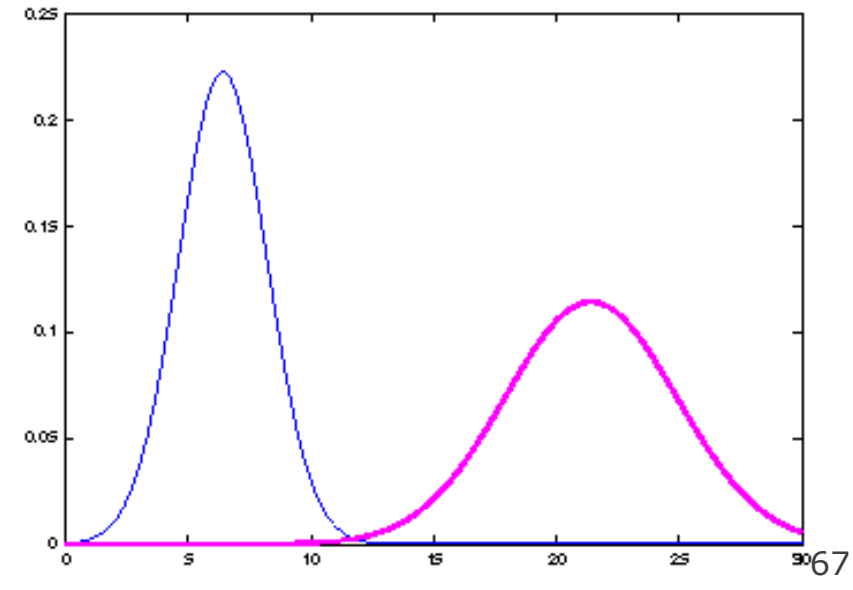
- Map is directed graph
- Location:
  - Edge e
  - Distance d from start of edge
- Prediction:
  - Move along edges according to velocity model
- Correction:
  - Update estimate based on GPS reading

Kalman Filter Updates in 1D

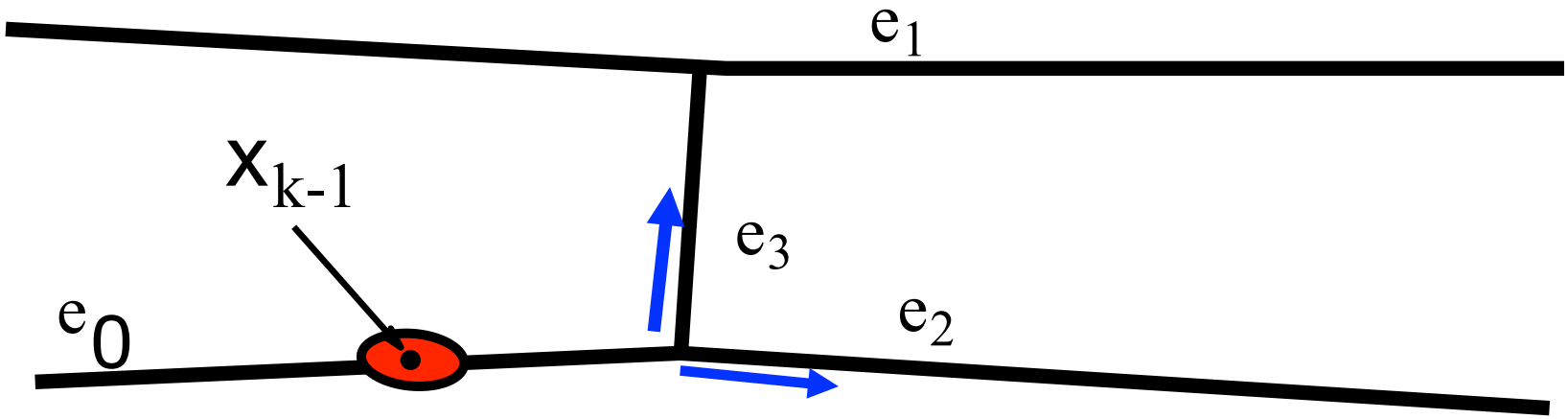






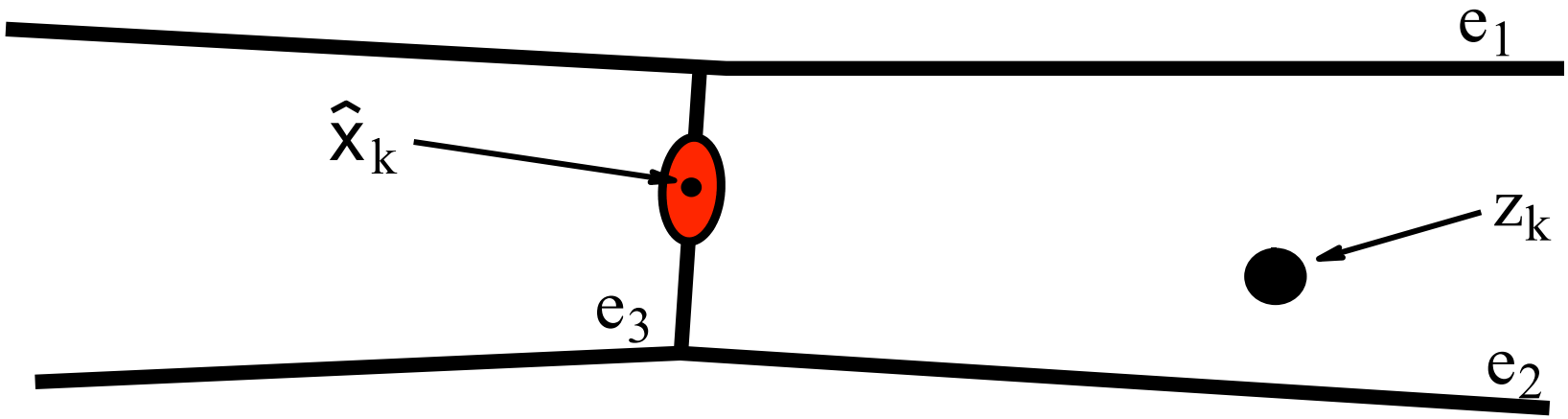


Kalman Filtering on a Graph: Prediction Step



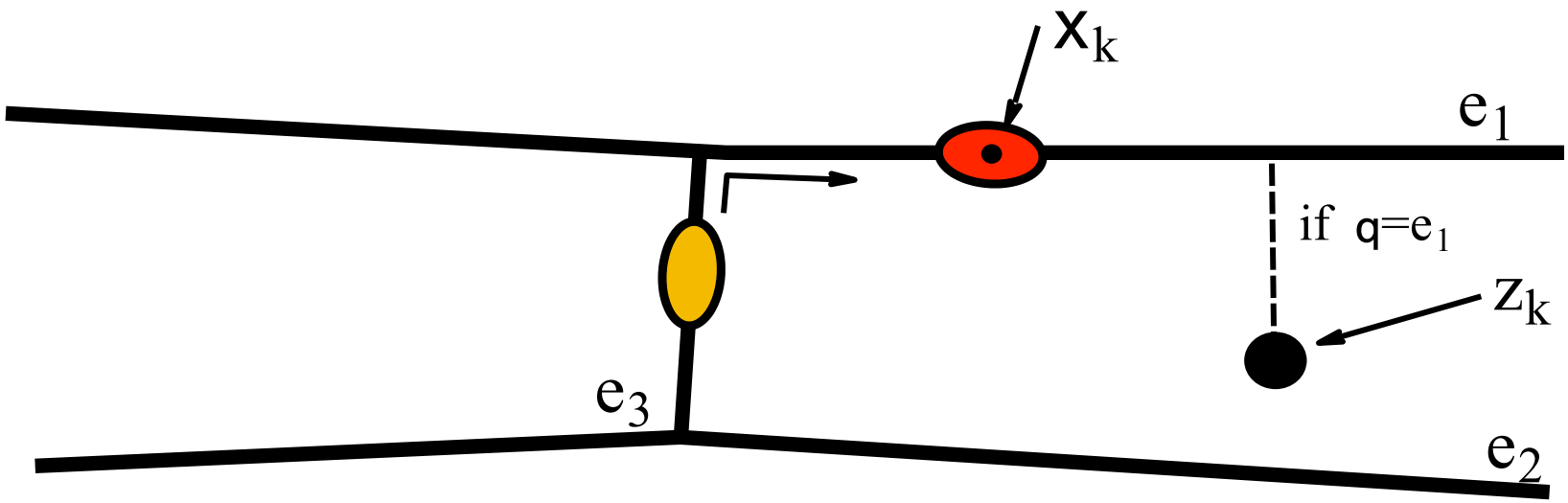
Problem: Predicted location is multi-modal

Kalman Filtering on a Graph: Correction Step



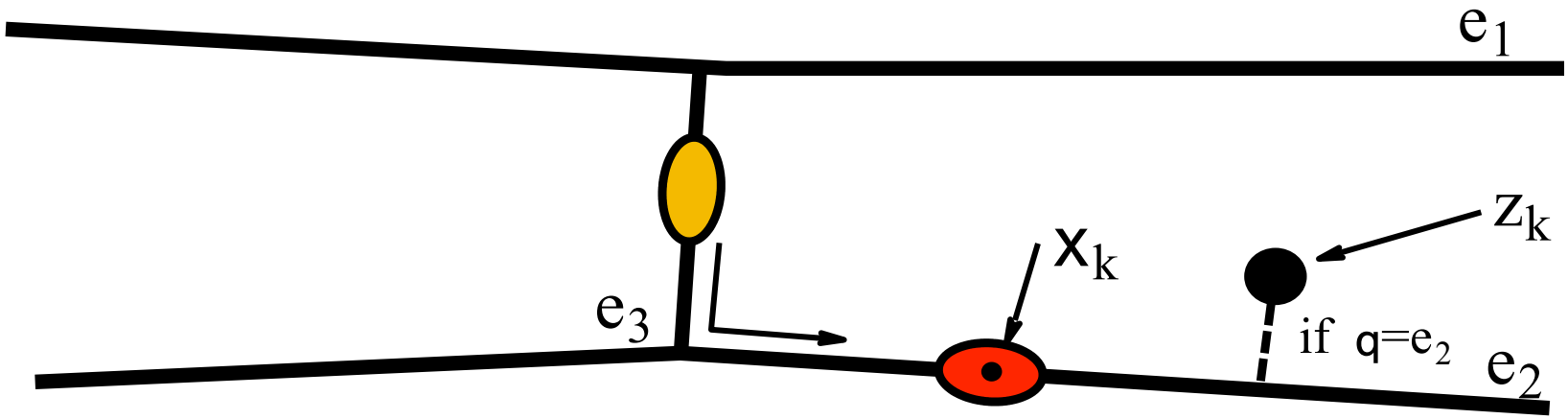
Problem: GPS reading is not on the graph

Kalman Filtering on a Graph: Correction Step



- Probabilistically "snap" GPS reading to the graph
- Perform A* search to compute innovation

Kalman Filtering on a Graph: Correction Step



- Probabilistically "snap" GPS reading to the graph
- Perform A* search to compute innovation

Location Tracking: Inference

- Rao-Blackwellised particle filter represents posterior by sets of weighted particles:

$$S_k = \{< s^{(i)}, w^{(i)} >, i = 1, \ldots, n\}$$

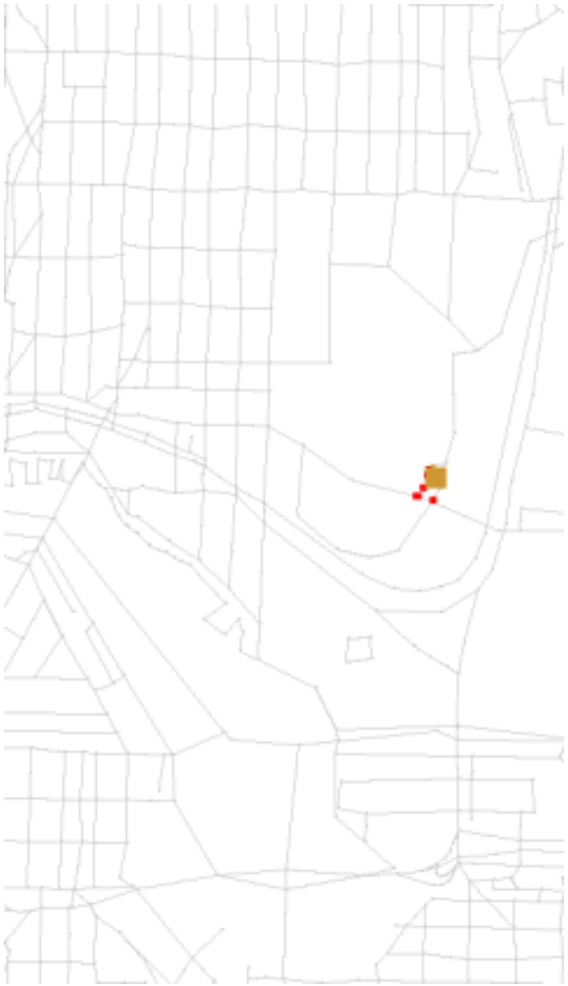
- Each particle contains Kalman filter for location:

$$s^{(i)} = \left\langle \underbrace{e^{(i)}, v^{(i)}, \theta^{(i)}}, \underbrace{N^{(i)}(\mu, \sigma^2)} \right\rangle$$

Edge transitions, velocities, edge associations

Gaussian for position

Tracking Example



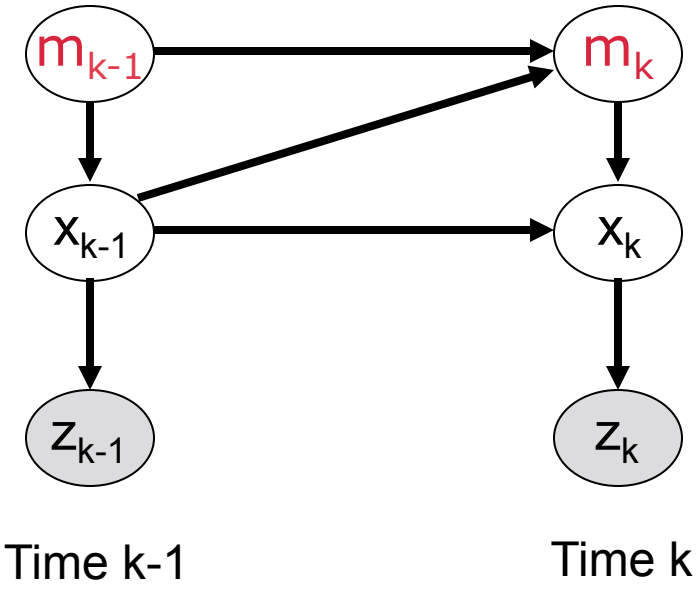
GPS measurements

Particles (Kalman filters)

Infer Mode of Transportation

- Encode prior knowledge into the model
  - Modes have different velocity distributions
  - Buses run on bus routes
  - Get on/off the bus near bus stops
  - Switch to car near car location

Dynamic Bayesian Network



Transportation mode

Edge, velocity, position

GPS reading

Particles: $s^{(i)} = \left\langle m^{(i)}, e^{(i)}, v^{(i)}, \theta^{(i)}, N^{(i)}(\mu, \sigma^2) \right\rangle$

Infer Location and Transportation

Measurements
Projections

Green Bus mode
Red Car mode
Blue Foot mode

Transportation Routines

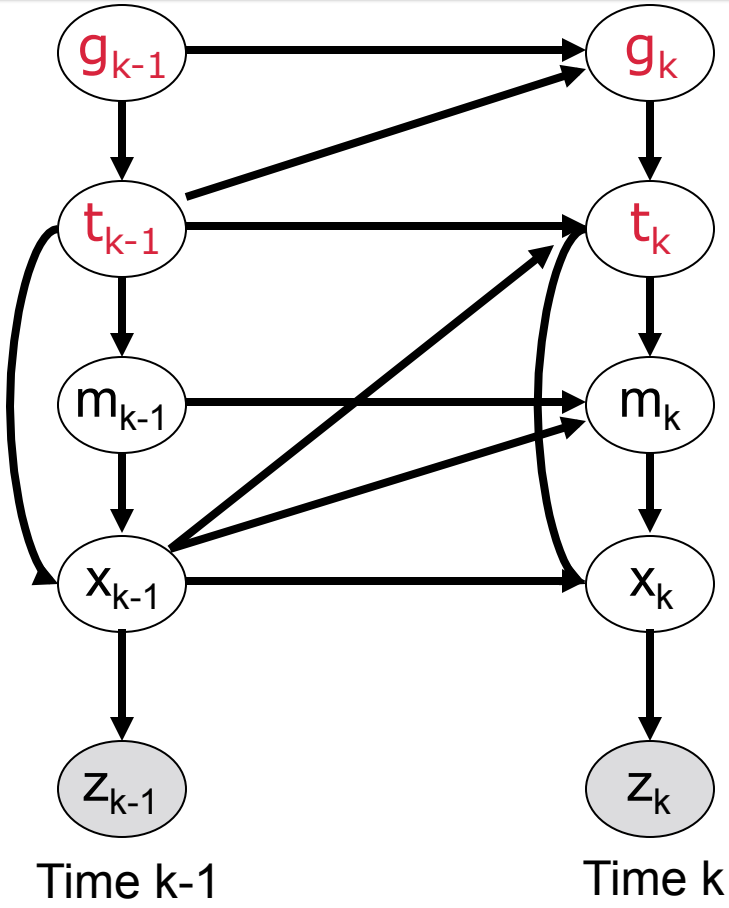
A

B

Workplace

- Goal (destination):
  - workplace (home, friends, restaurant, …)
- Trip segments: <start, end, transportation>
  - Home to Bus stop A on Foot
  - Bus stop A to Bus stop B on Bus
  - Bus stop B to workplace on Foot

Hierarchical Model



Goal

Trip segment

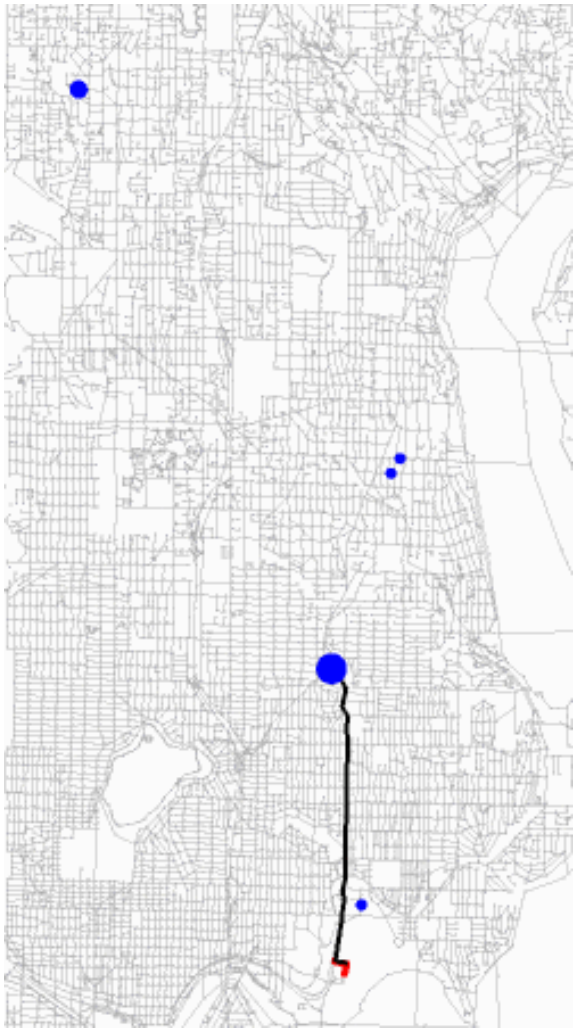
Transportation mode

Edge, velocity, position

GPS reading

Particles: $s^{(i)} = \left\langle \langle g, t \rangle^{(i)}, m^{(i)}, e^{(i)}, v^{(i)}, \theta^{(i)}, N^{(i)}(\mu, \sigma^2) \right\rangle$

Predict Goal and Path

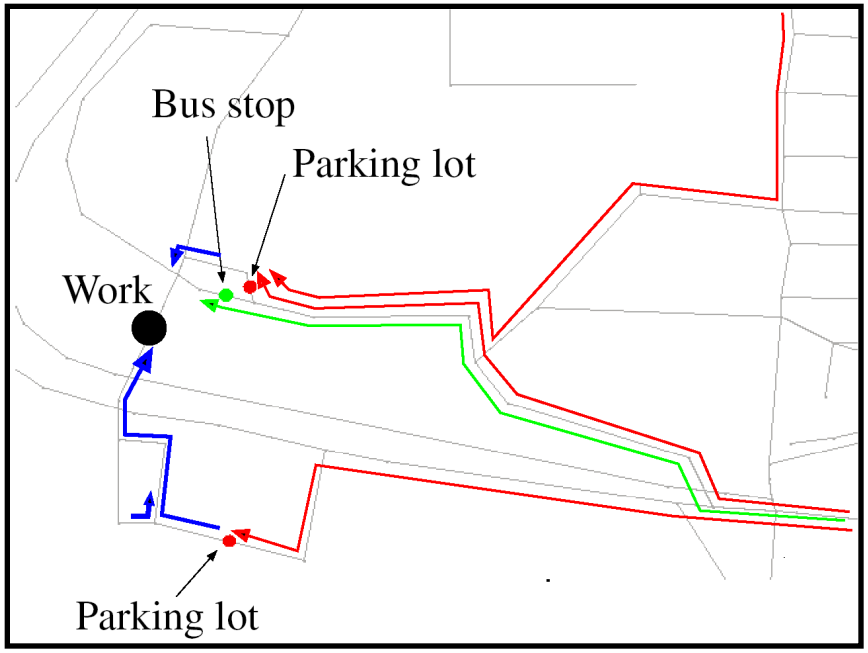


Predicted goal

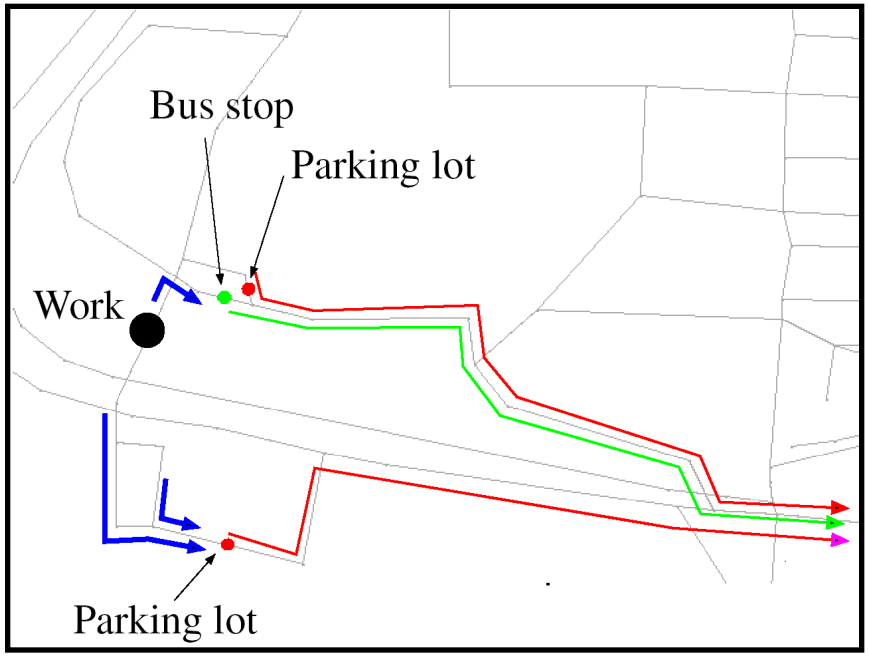
Predicted path

Learned Transition Parameters

GOING TO THE WORKPLACE



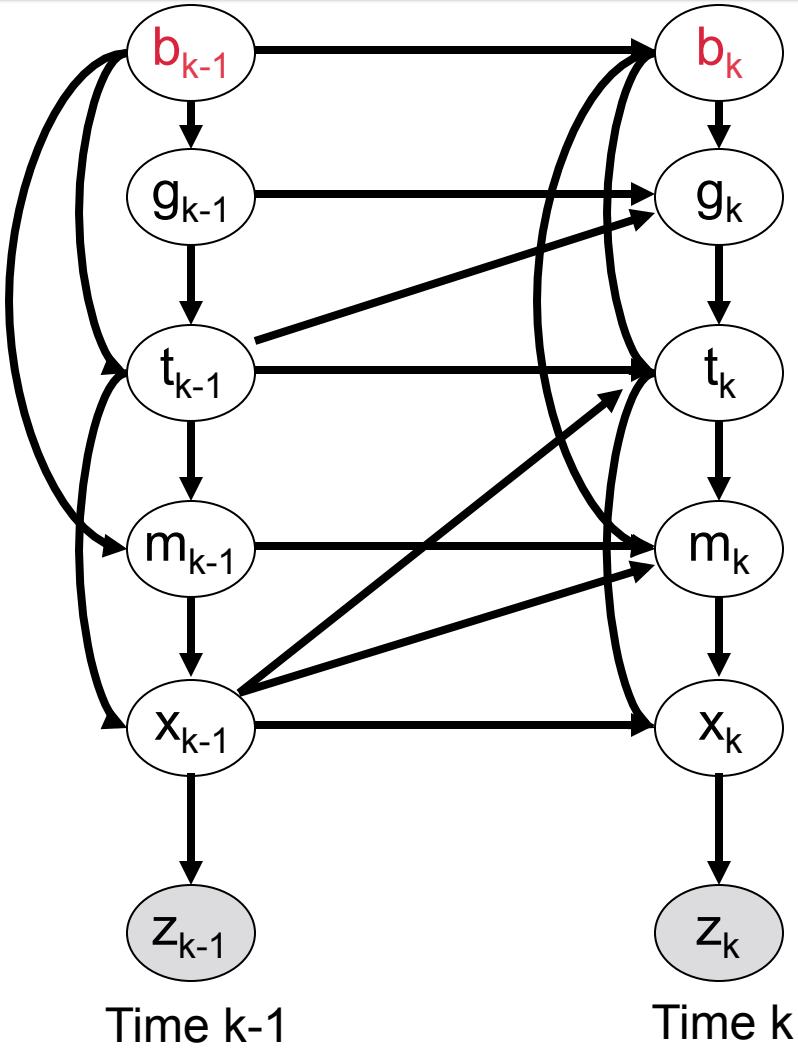
GOING HOME



High probability transitions: **bus** **car** **foot**

Detect Atypical Behavior and User Errors

[Patterson-Liao-etAl: Ubicomp-04]



Behavior mode

normal / unknown / error

Goal

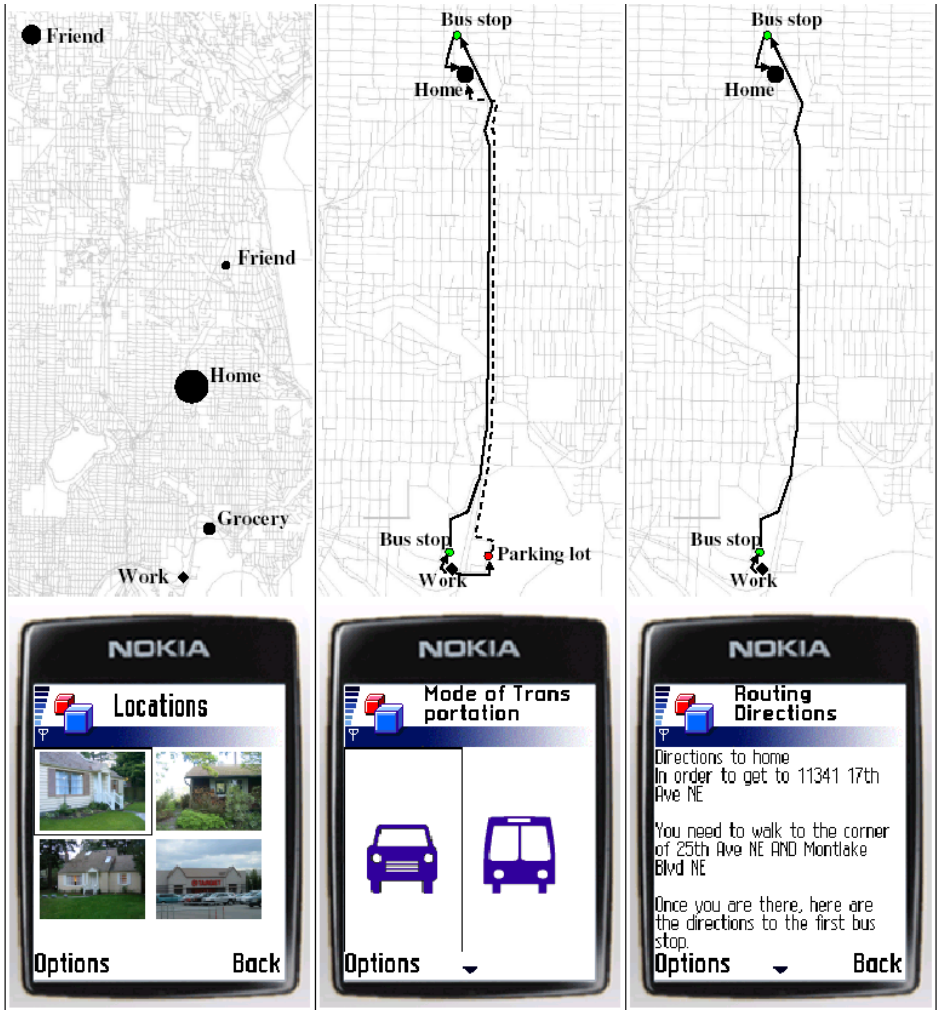
Trip segment

Transportation mode

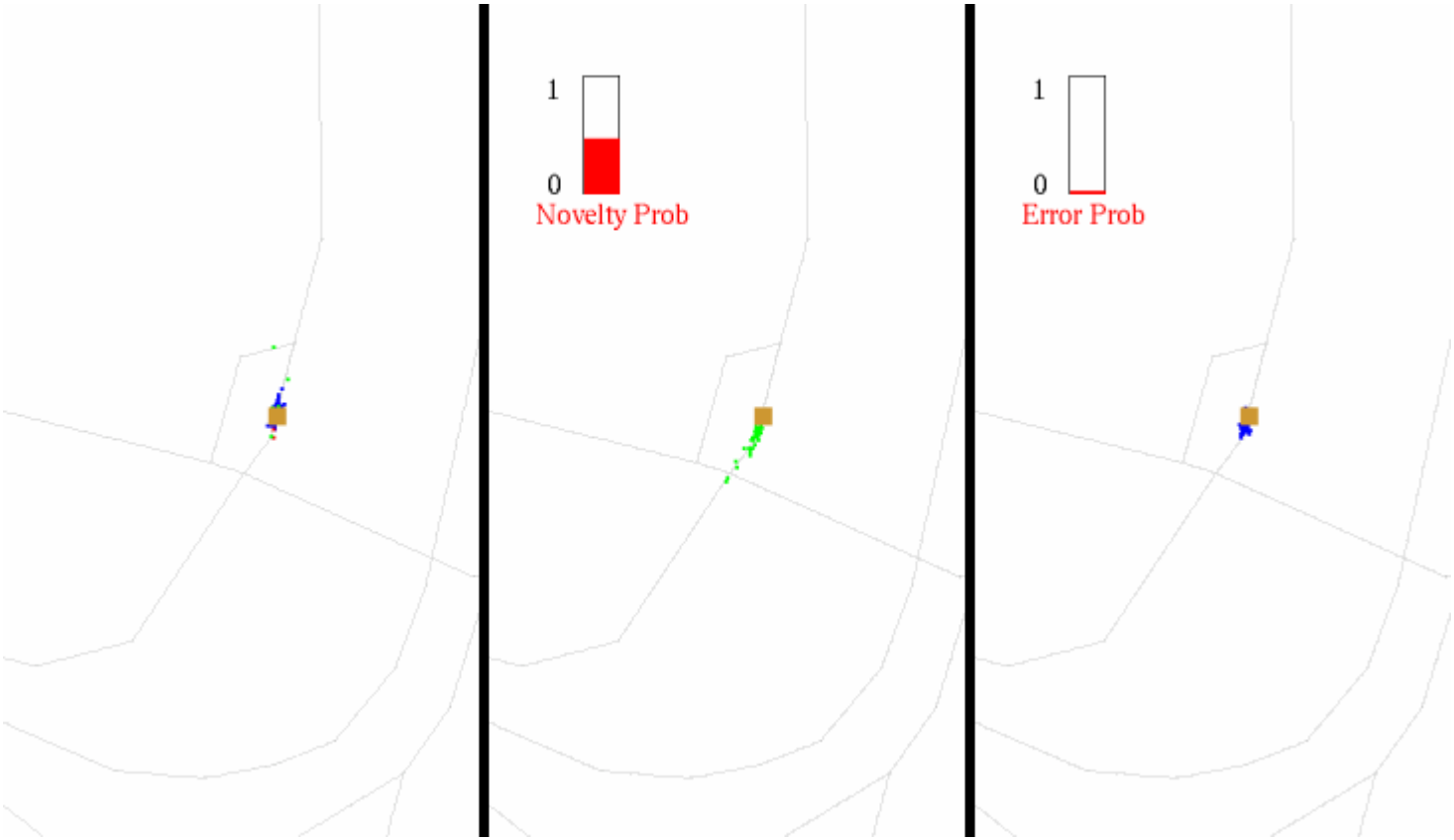
Edge, velocity, position

GPS reading

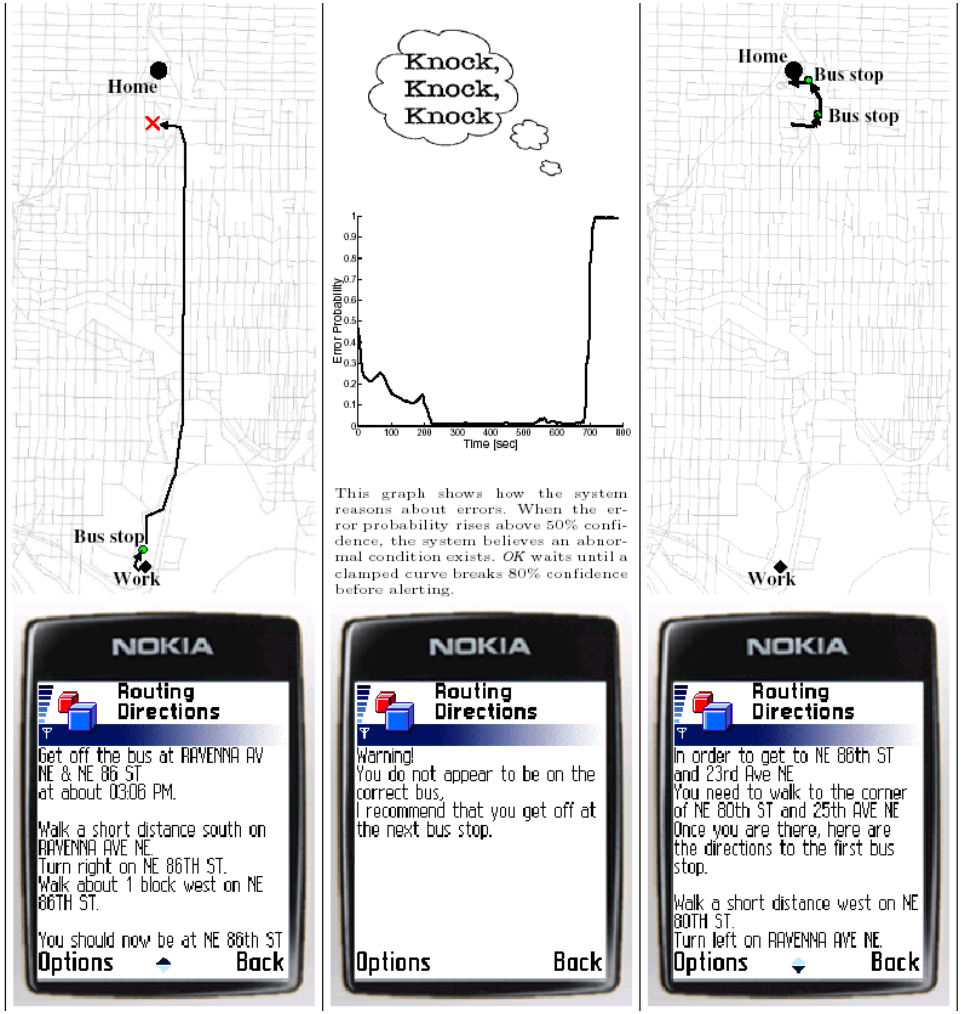
Application: Opportunity Knocks



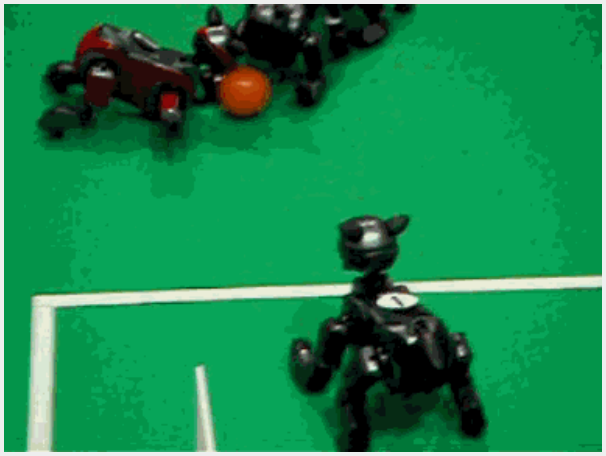
Detect User Errors

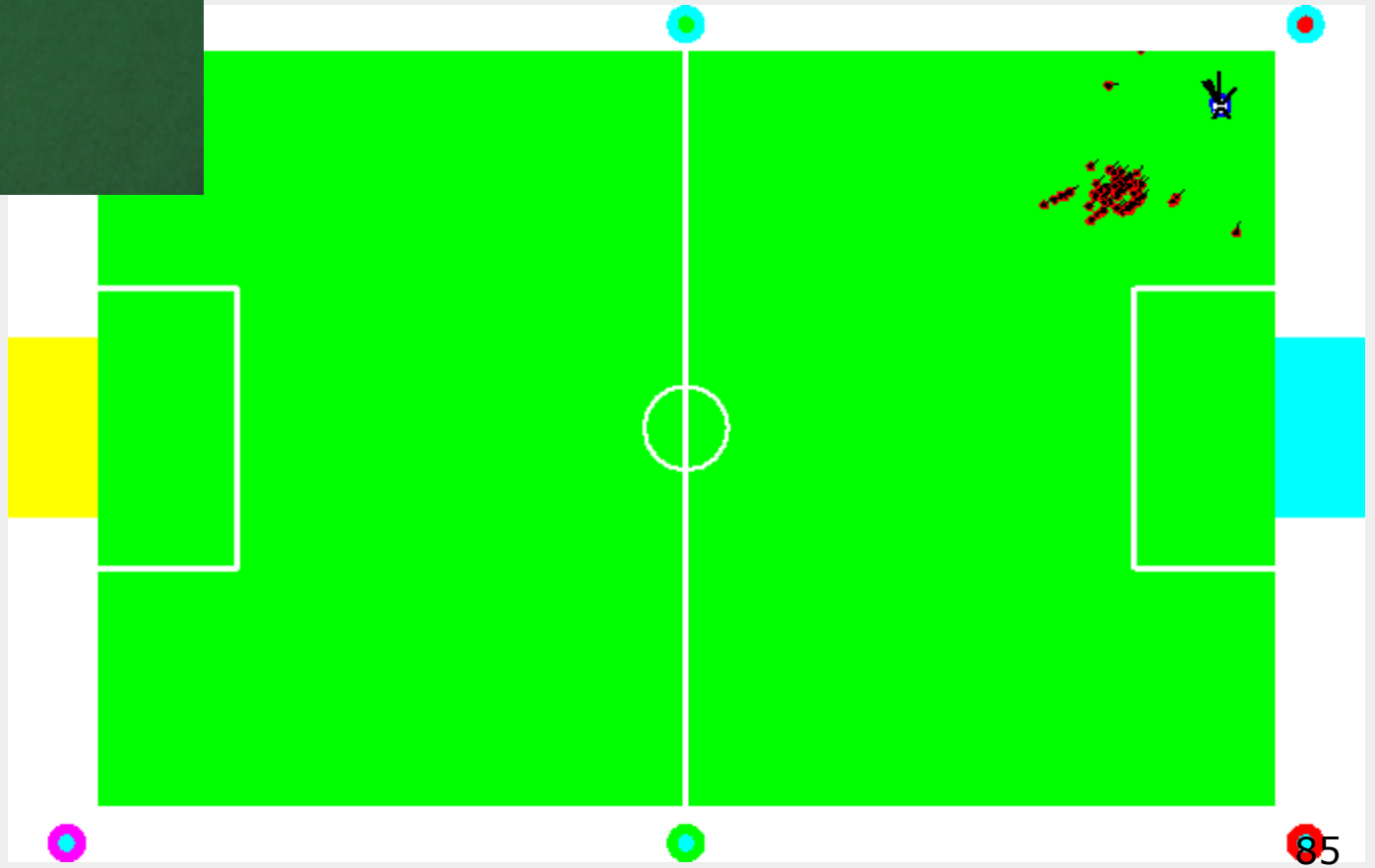


Application: Opportunity Knocks



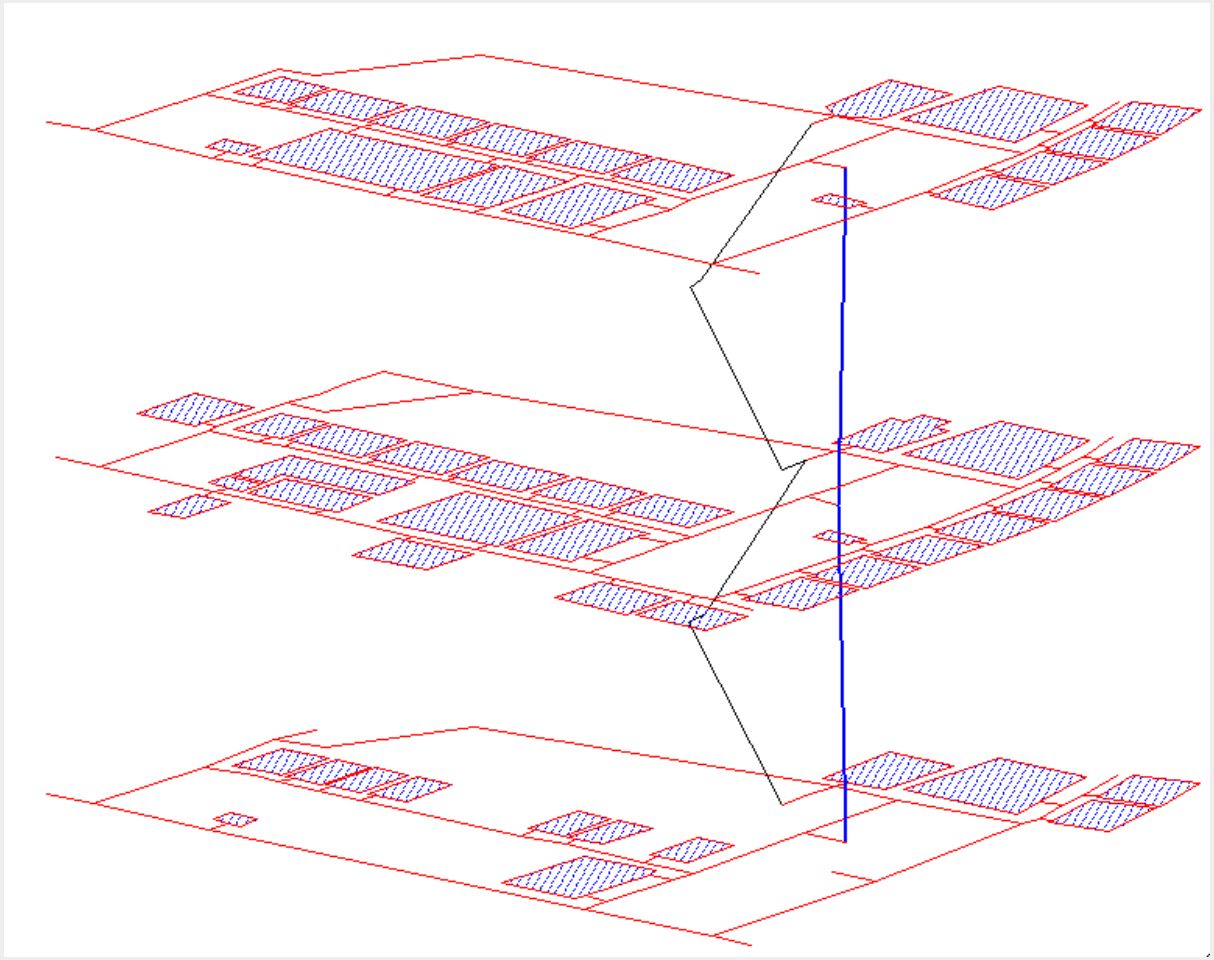
Ball Tracking



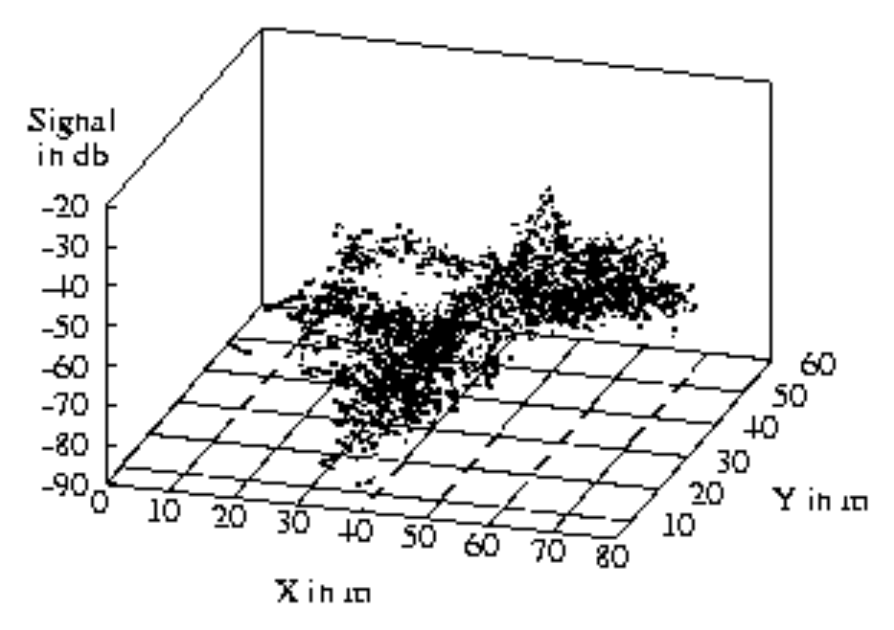


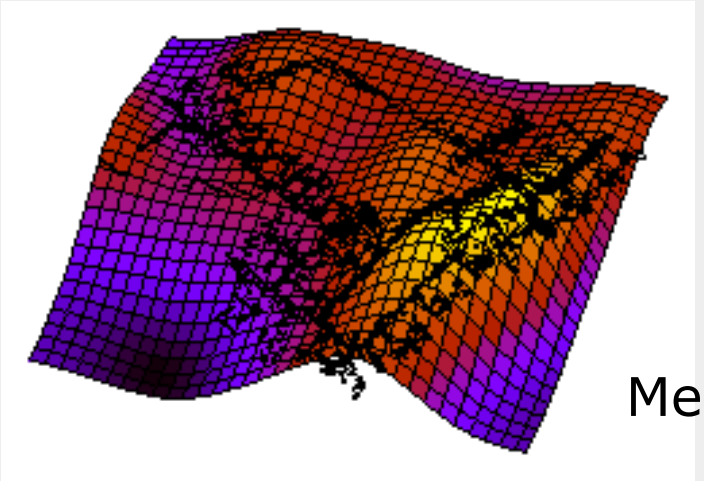
[Ferris-Haehnel-Fox: RSS-06]

Hybrid Model for People Tracking

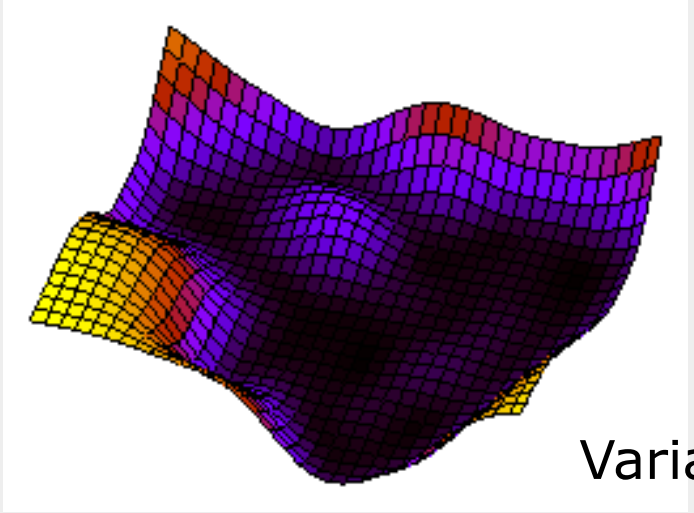


WiFi Sensor Model





Mean



Variance

Tracking Example