CSE-571
Robotics

Bayes Filter Implementations

Particle filters

1

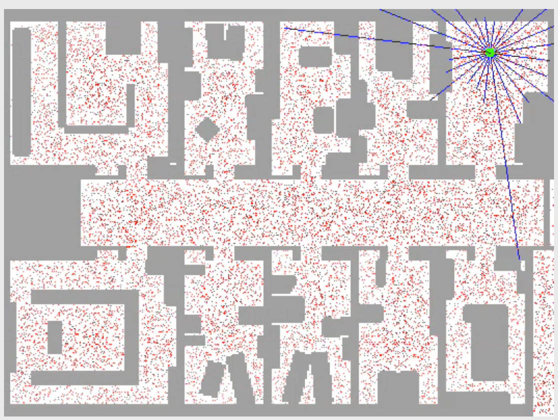
Motivation

- So far, we discussed the
  - Kalman filter: Gaussian, linearization problems, multi-modal beliefs
- Particle filters are a way to **efficiently** represent **non-Gaussian distributions**
- Basic principle
  - Set of state hypotheses (“particles”)
  - Survival-of-the-fittest

2

2

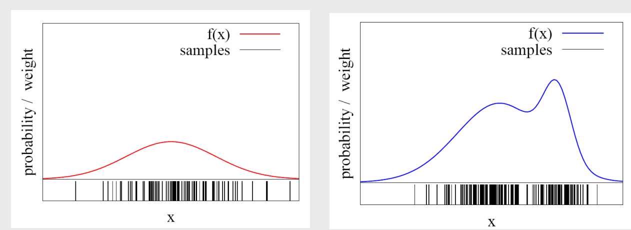
Sample-based Localization (sonar)



3

Density Approximation

- Particle sets can be used to approximate densities



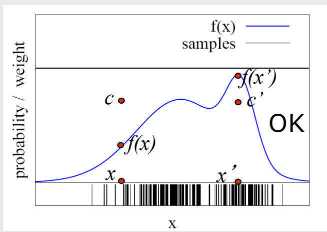
- The more particles fall into an interval, the higher the probability of that interval
- How to draw samples form a function/distribution?

4

4

Rejection Sampling

- Let us assume that $f(x) <= 1$ for all x
- Sample x from a uniform distribution
- Sample c from [0,1]
- if $f(x) > c$ keep the sample
  otherwise reject the sampe

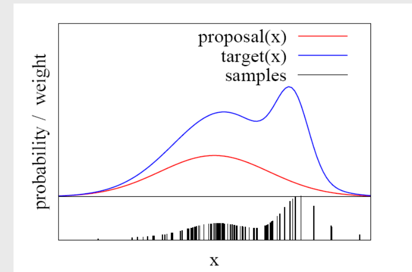


5

5

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



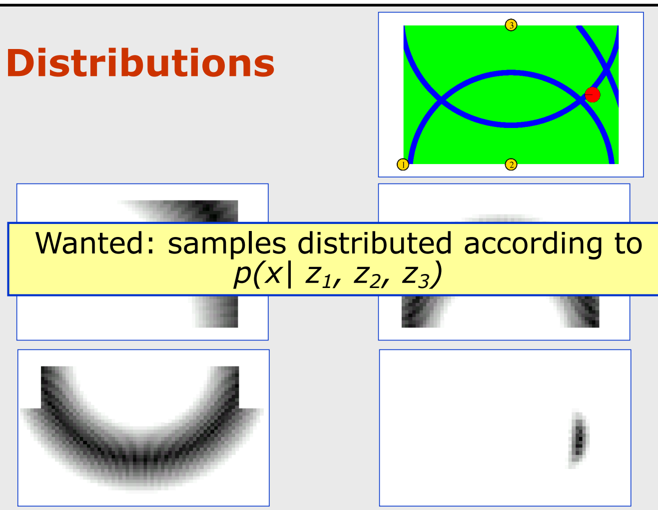
6

6

Importance Sampling with Resampling: Landmark Detection Example

7

Distributions

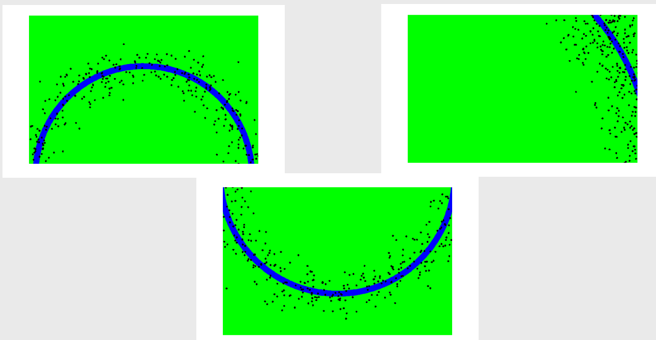


Wanted: samples distributed according to $p(x| z_1, z_2, z_3)$

8

This is Easy!

We can draw samples from $p(x|z_l)$ by adding noise to the detection parameters.



9

Importance Sampling with Resampling

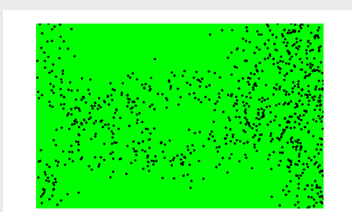
$$\text{Target distribution f}: p(x \mid z_1, z_2, \ldots, z_n) = \frac{\prod_k p(z_k \mid x) \quad p(x)}{p(z_1, z_2, \ldots, z_n)}$$

$$\text{Sampling distribution g}: p(x \mid z_l) = \frac{p(z_l \mid x) p(x)}{p(z_l)}$$

$$\text{Importance weights w}: \frac{f}{g} = \frac{p(x \mid z_1, z_2, \ldots, z_n)}{p(x \mid z_l)} = \frac{p(z_l) \prod_{k \neq l} p(z_k \mid x)}{p(z_1, z_2, \ldots, z_n)}$$

10

Importance Sampling with Resampling



Weighted samples

After resampling

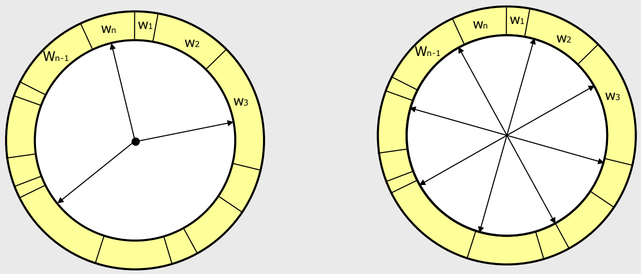
11

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'.

12

Resampling

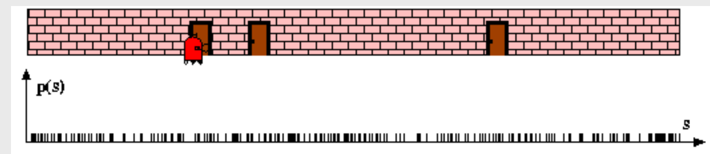


- Roulette wheel
- Binary search, n log n

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

13

Particle Filters

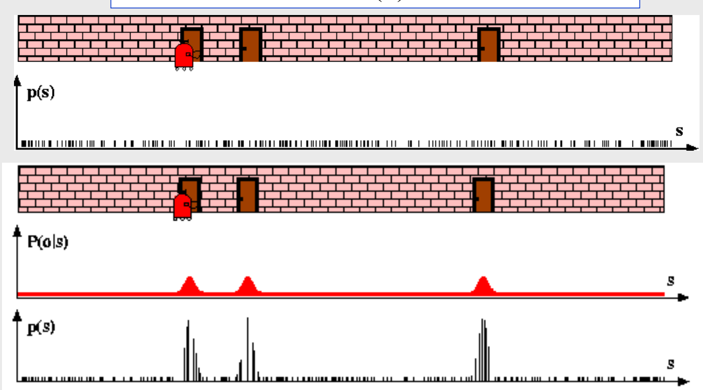


14

Sensor Information: Importance Sampling

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

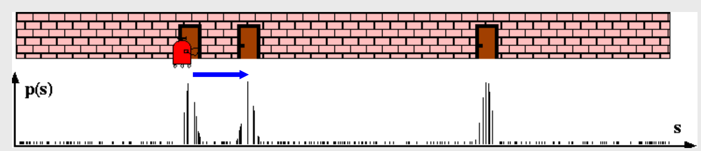
$$w \leftarrow \frac{\alpha\, p(z \mid x)\, Bel^{-}(x)}{Bel^{-}(x)} = \alpha\, p(z \mid x)$$

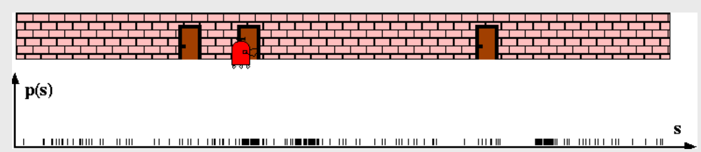


15

Robot Motion

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

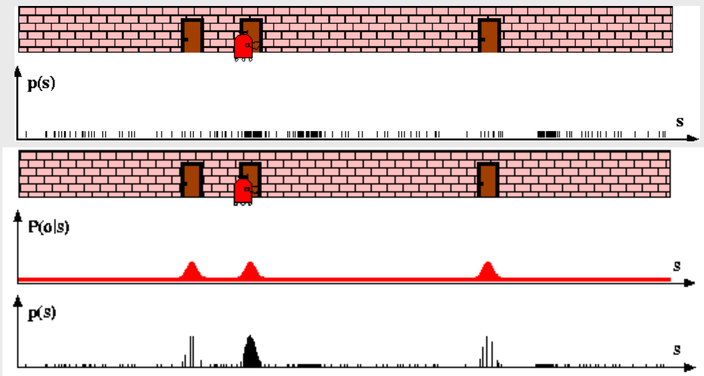




16

Sensor Information: Importance Sampling

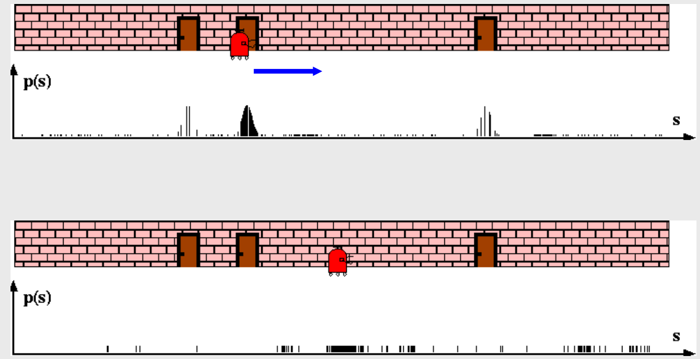
$$Bel(x) \leftarrow \alpha\, p(z|x)\, Bel^-(x)$$
$$w \leftarrow \frac{\alpha\, p(z|x)\, Bel^-(x)}{Bel^-(x)} = \alpha\, p(z|x)$$



17

Robot Motion

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



18

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. $S_t = S_t \cup \{\langle x_t^i, w_t^i \rangle\}$ ***Insert***
9. **For** $i = 1 \ldots n$
10. $w_t^i = w_t^i / \eta$ ***Normalize weights***

19

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 $Bel(x_{t-1})$

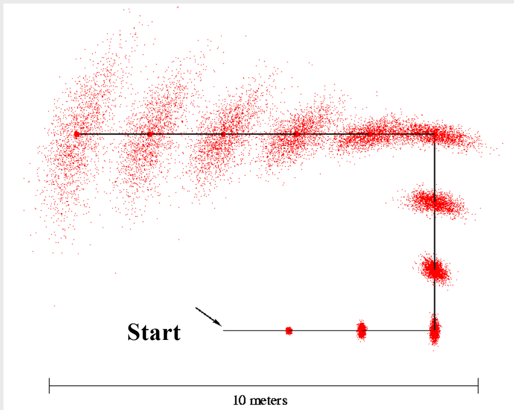
draw x^i_t from $p(x_t \mid x^i_{t-1}, u_{t-1})$

Importance factor for x^i_t:

$$w_t^i = \frac{\text{target distribution}}{\text{proposal distribution}} = \frac{\eta\, p(z_t \mid x_t)\, p(x_t \mid x_{t-1}, u_{t-1})\, Bel(x_{t-1})}{p(x_t \mid x_{t-1}, u_{t-1})\, Bel(x_{t-1})} \propto p(z_t \mid x_t)$$

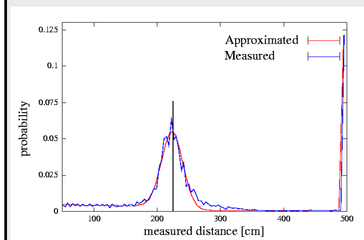
20

Motion Model Reminder

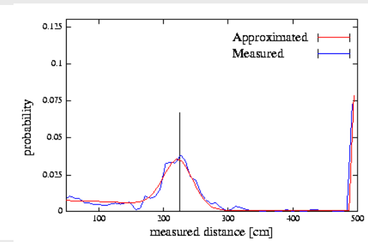


21

Proximity Sensor Model Reminder

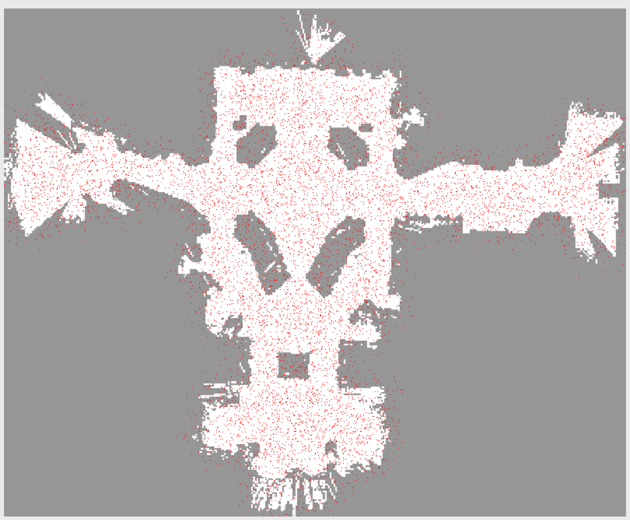


Laser sensor



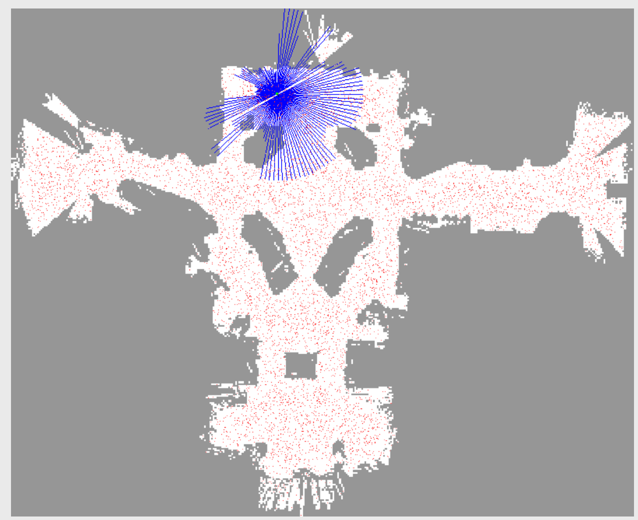
Sonar sensor

22



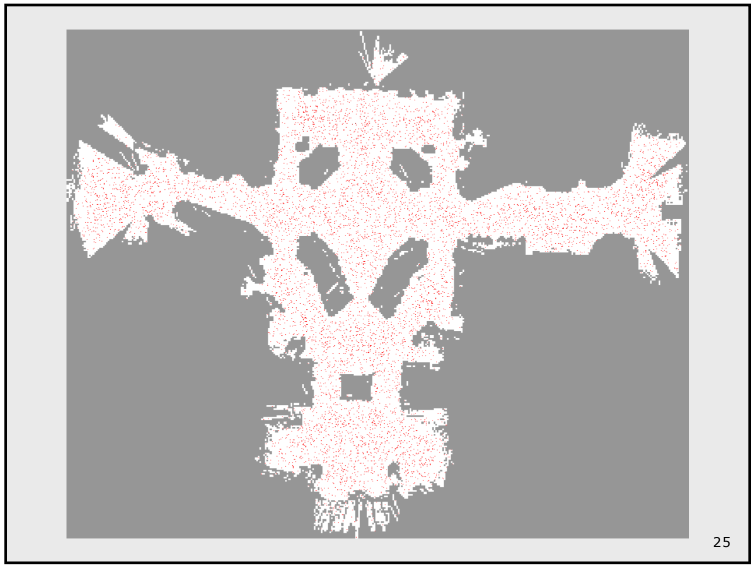
23

23

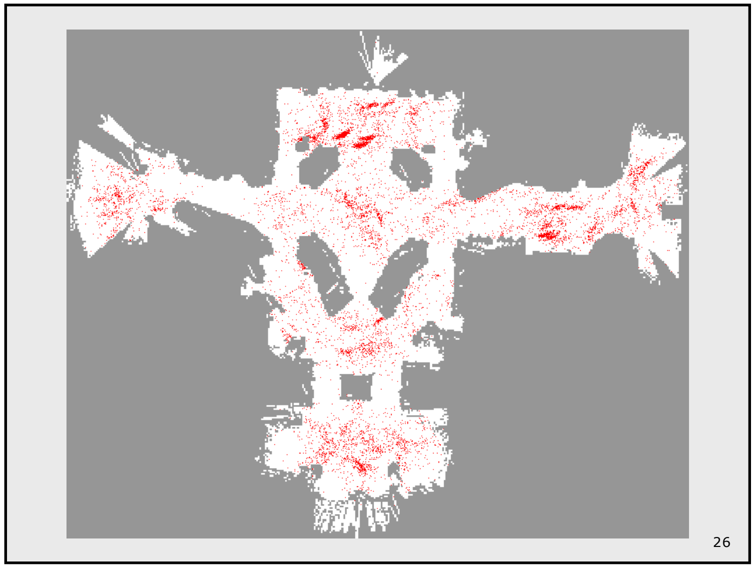


24

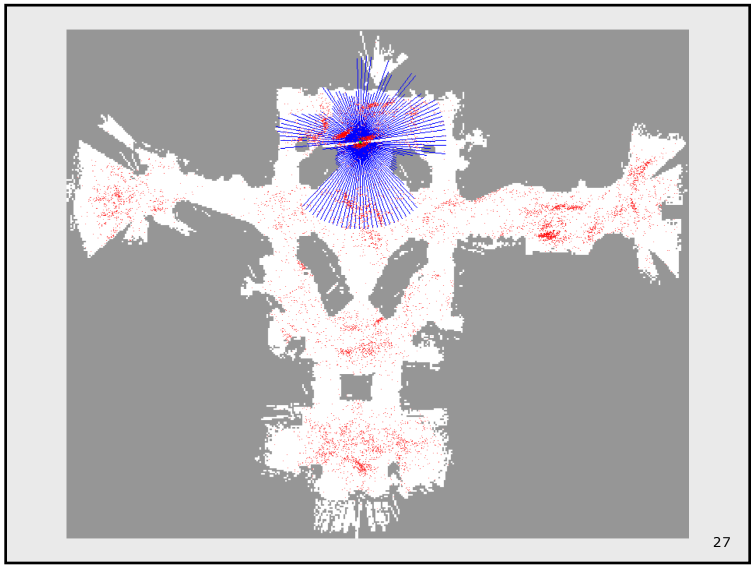
24



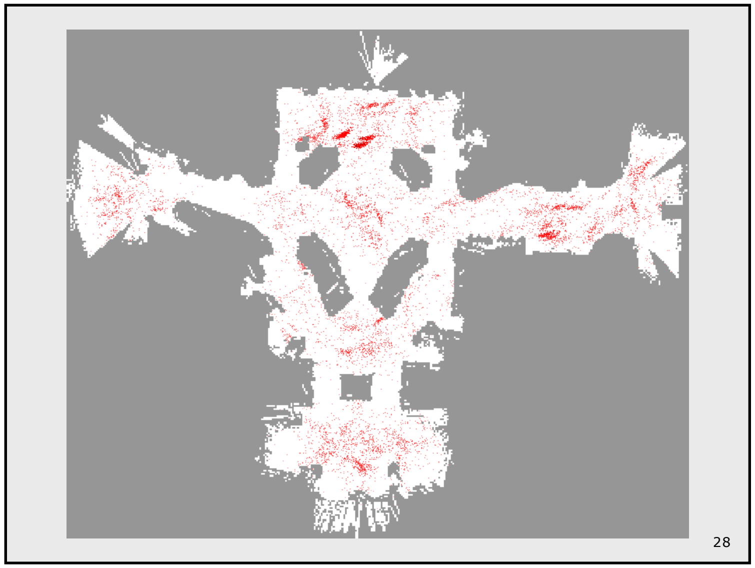
25



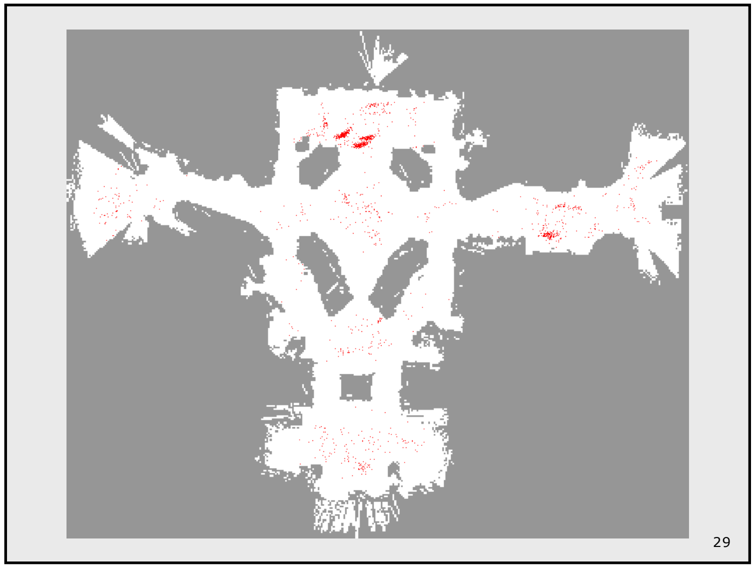
26



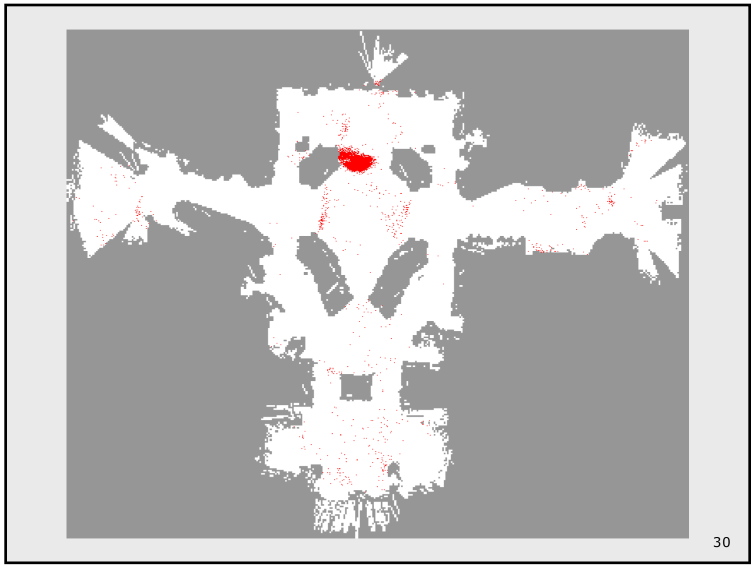
27



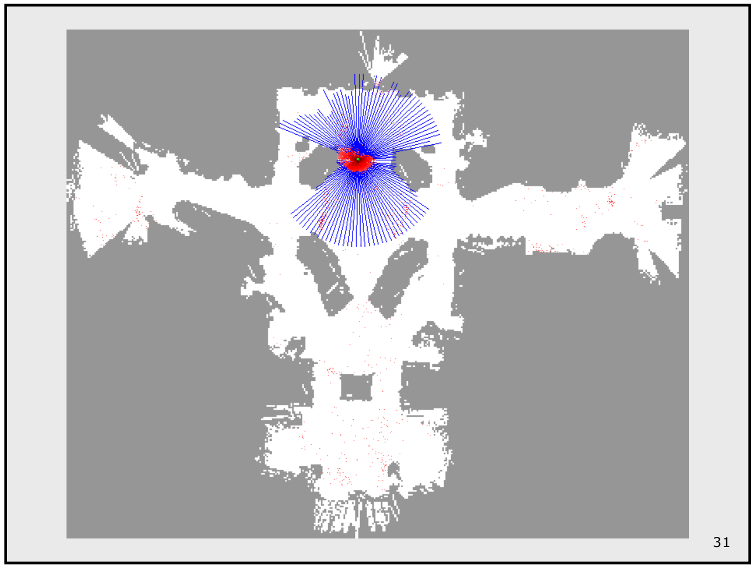
28



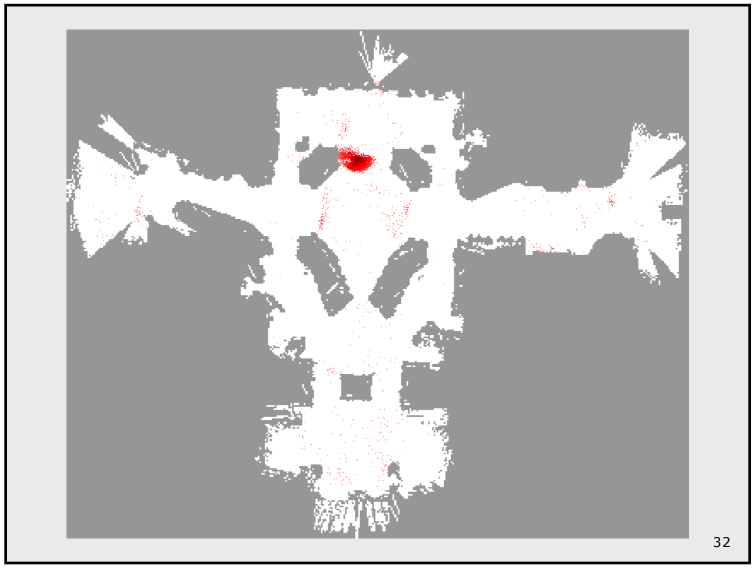
29



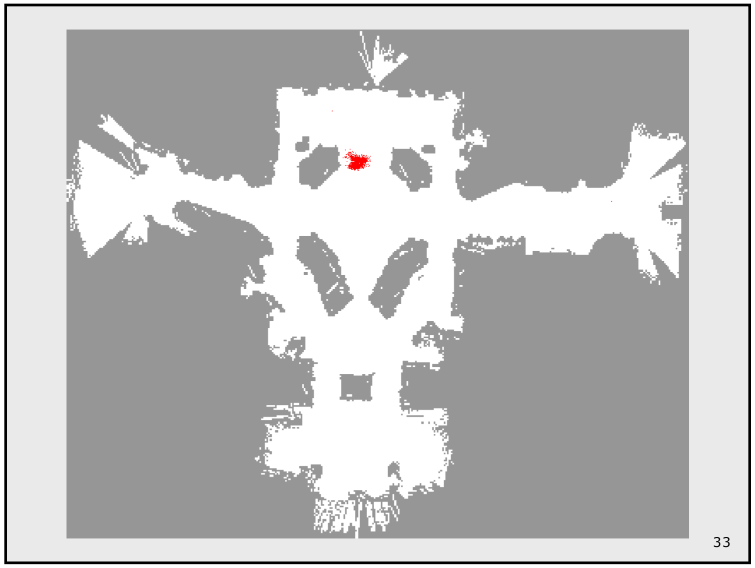
30



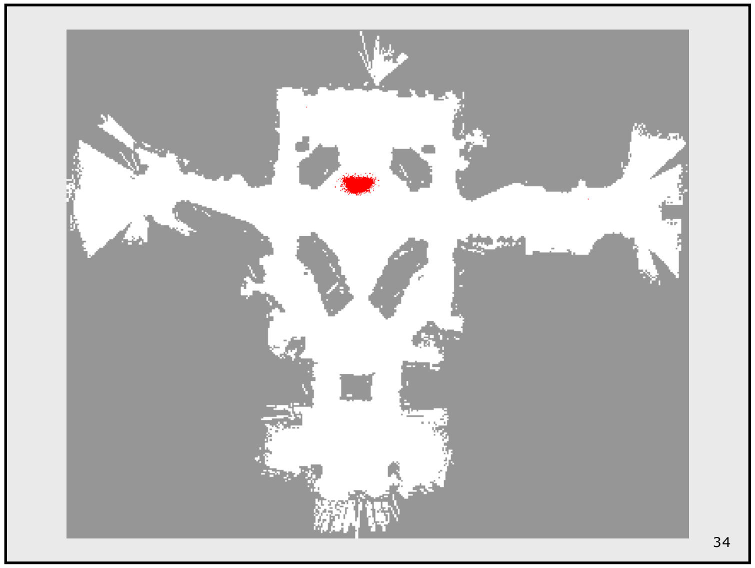
31



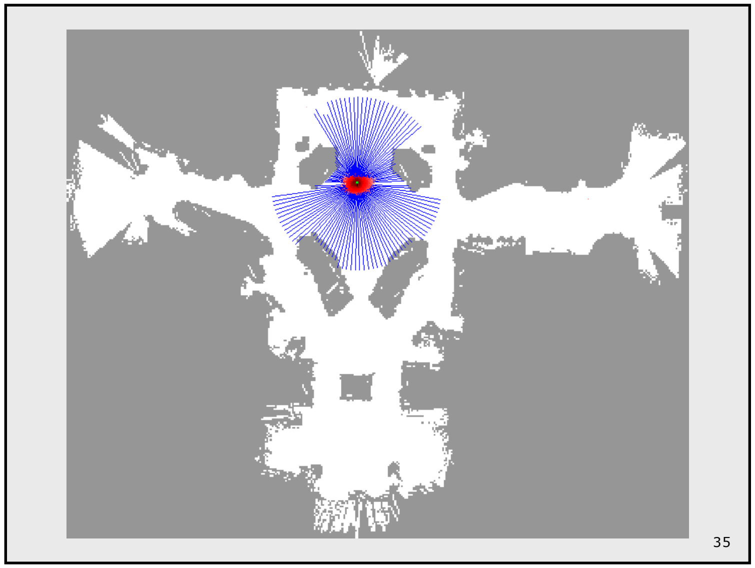
32



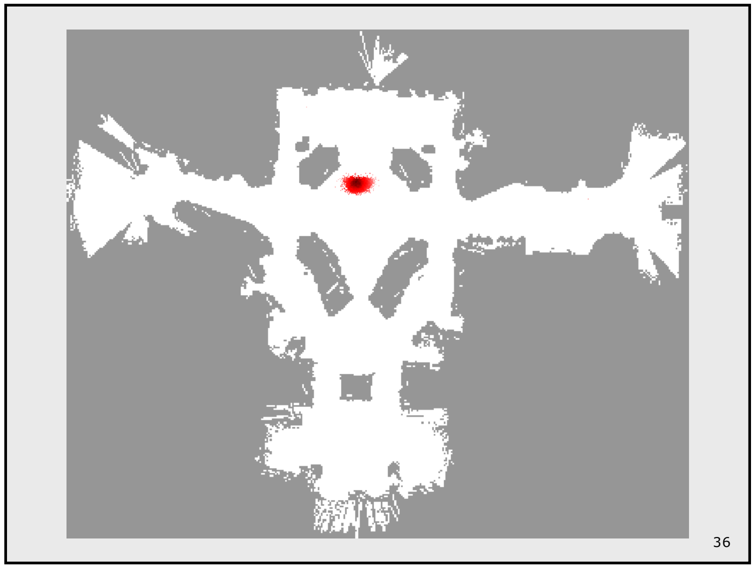
33



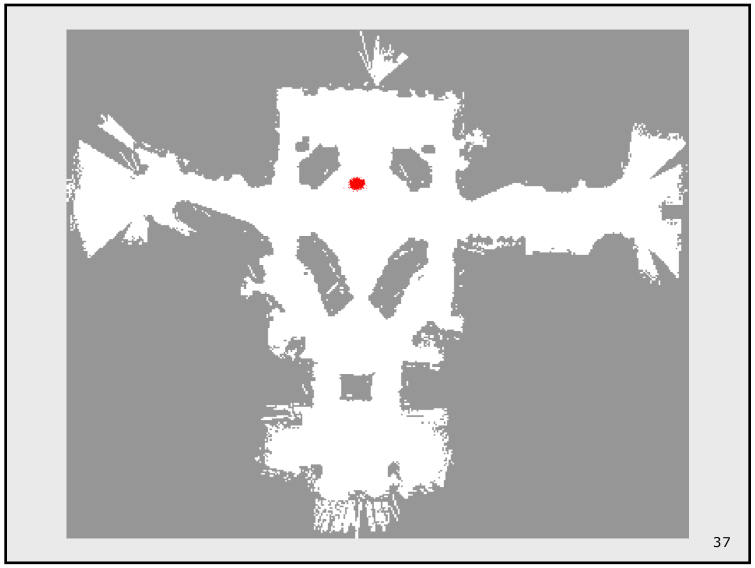
34



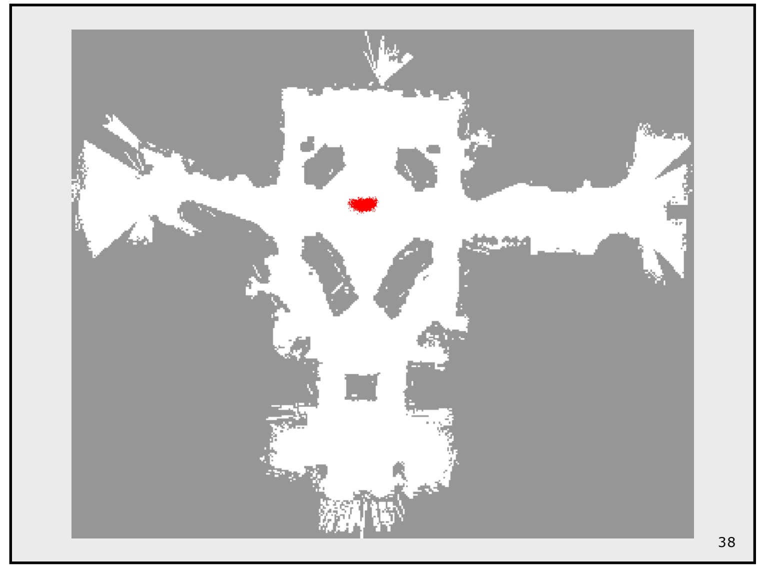
35



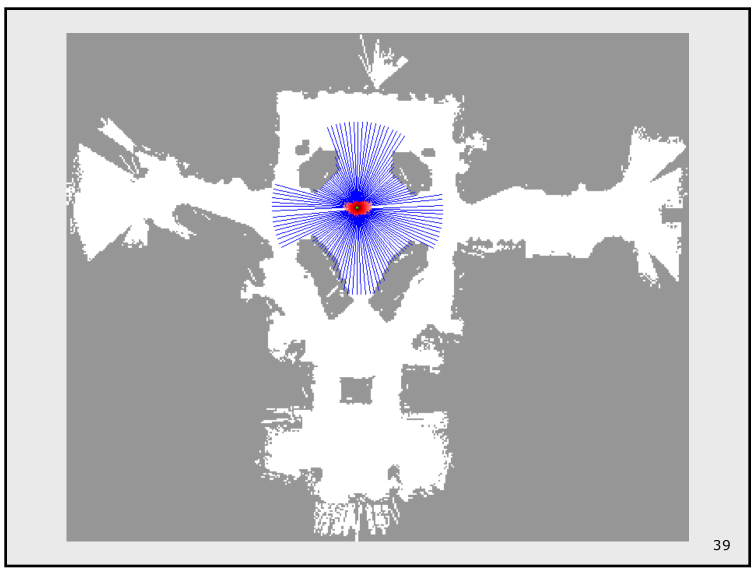
36



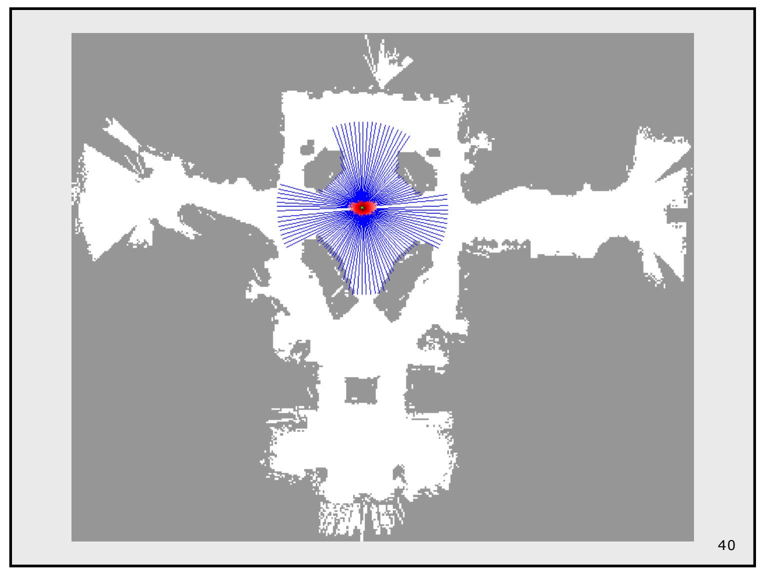
37



38

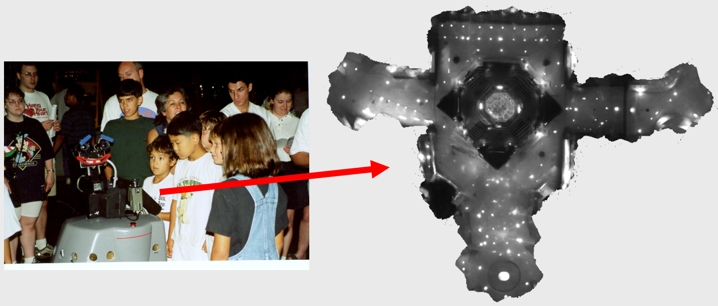


39



40

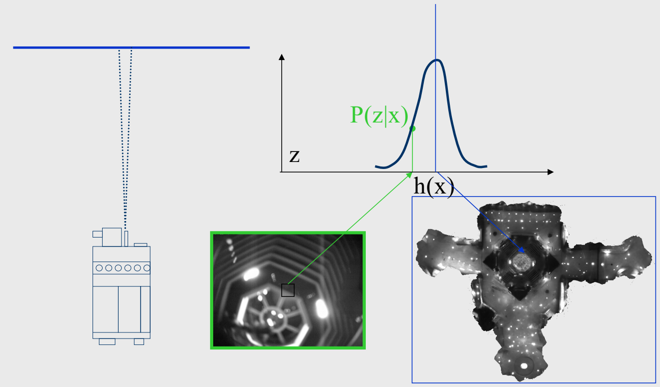
Using Ceiling Maps for Localization



[Dellaert et al. 99]

41

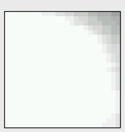
Vision-based Localization



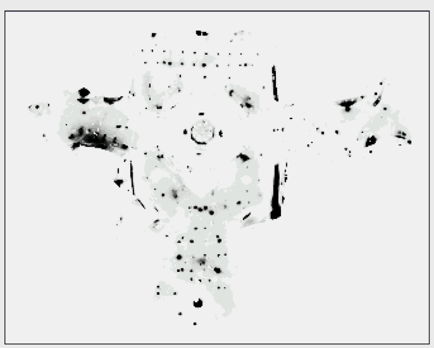
42

Under a Light

Measurement z:



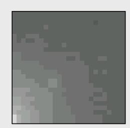
***P(z|x)*:**



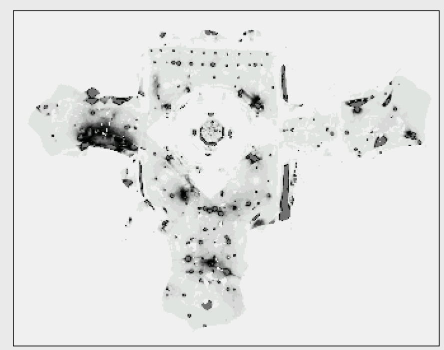
43

Next to a Light

Measurement z:



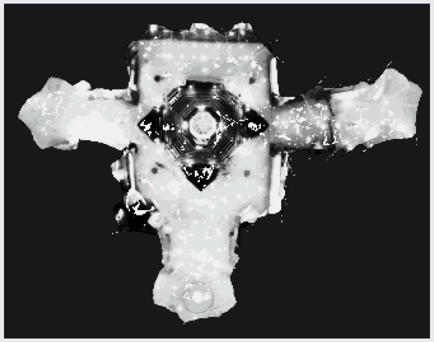
***P(z|x)*:**



44

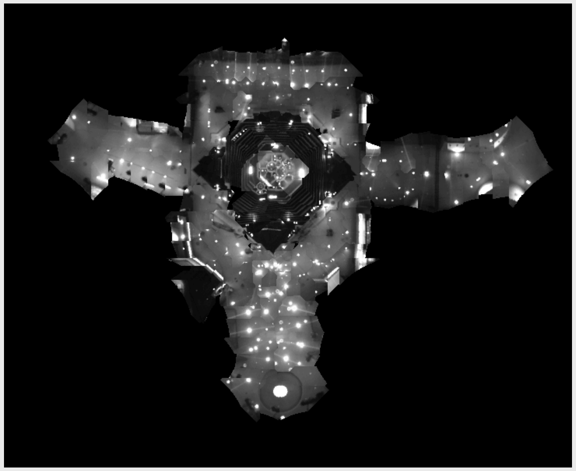
Elsewhere

Measurement z: 

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

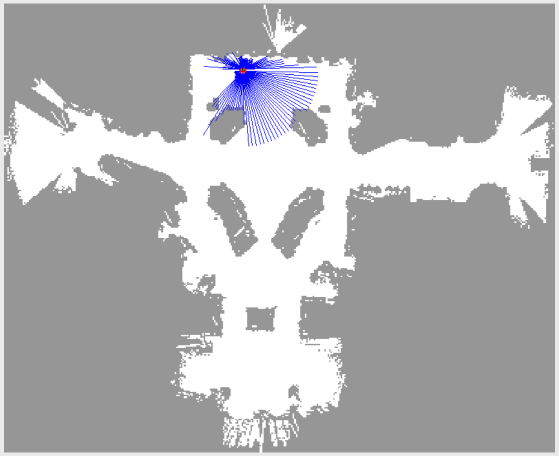
45

Global Localization Using Vision



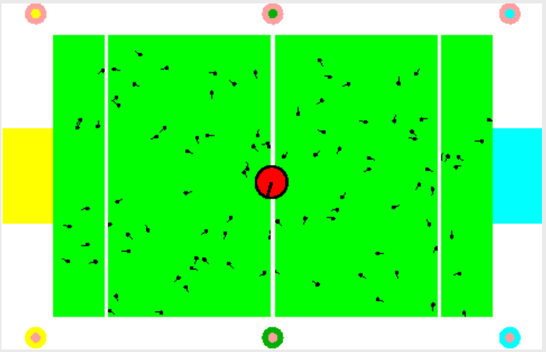
46

Recovery from Failure



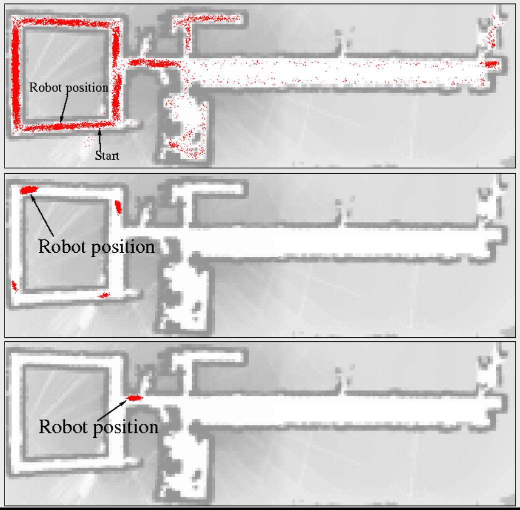
47

Localization for AIBO robots



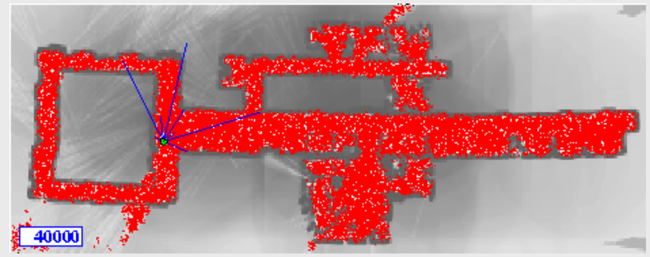
48

Adaptive Sampling



49

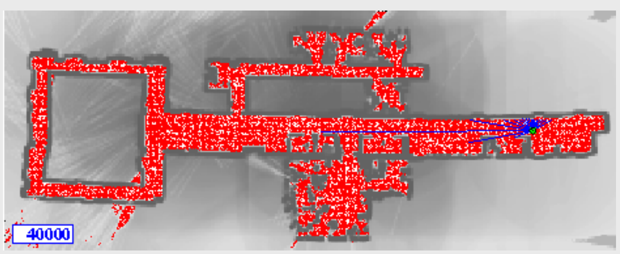
KLD-Sampling Sonar



Adapt number of particles on the fly based on statistical approximation measure

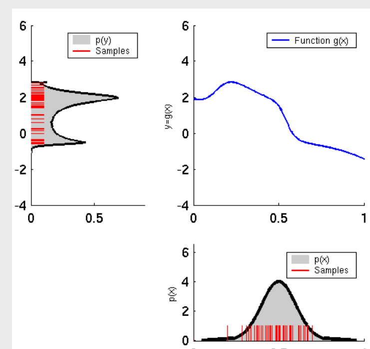
50

KLD-Sampling Laser



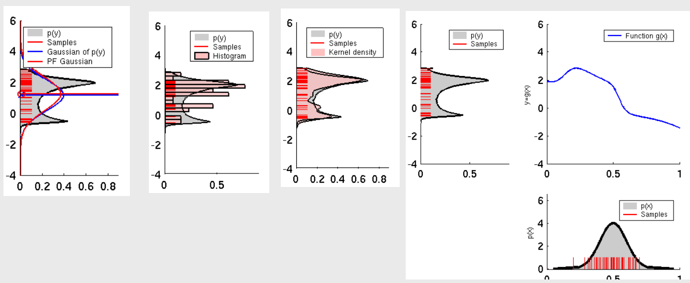
51

Particle Filter Projection



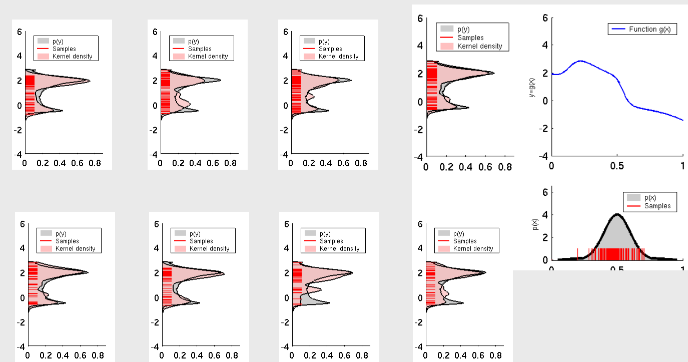
52

Density Extraction



53

Sampling Variance



54

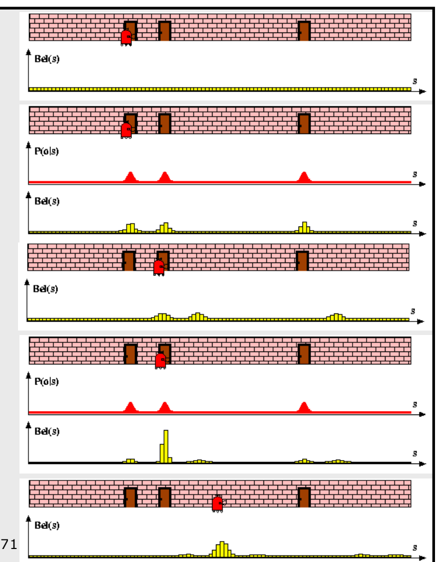
CSE-571
Robotics

Bayes Filter Implementations

Discrete filters

55

Piecewise Constant



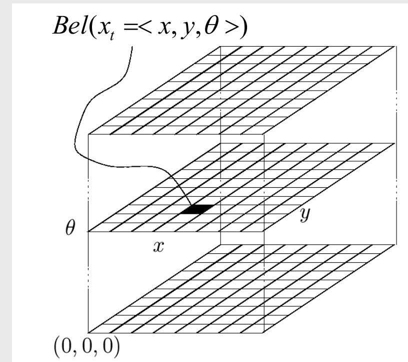
4/22/20 CSE-571

56

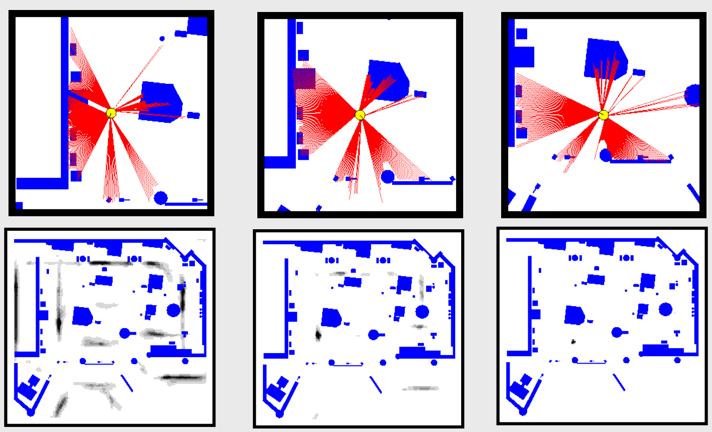
Discrete Bayes Filter Algorithm

Algorithm **Discrete_Bayes_filter**(*Bel(x),d*):
η=0
If *d* is a perceptual data item *z* then
    For all *x* do
        $Bel'(x) = P(z \mid x) Bel(x)$
        $\eta = \eta + Bel'(x)$
    For all *x* do
        $Bel'(x) = \eta^{-1} Bel'(x)$
Else if *d* is an action data item *u* then
    For all *x* do
        $Bel'(x) = \sum_{x'} P(x \mid u, x')\, Bel(x')$
Return *Bel'(x)*

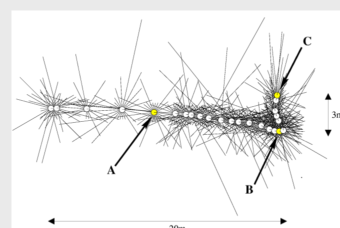
Piecewise Constant Representation



Grid-based Localization

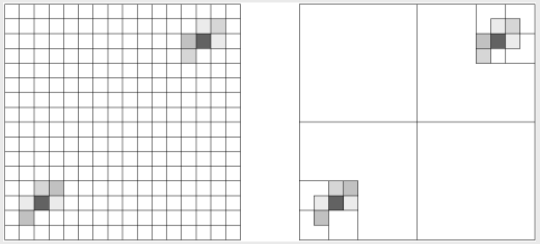


Sonars and Occupancy Grid Map



Tree-based Representation

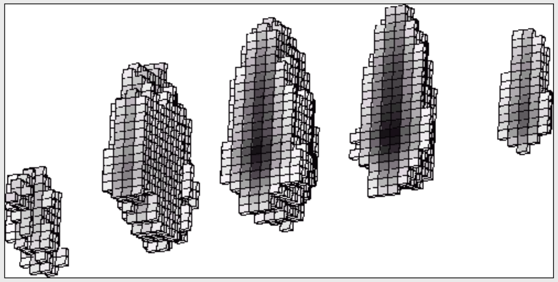
Idea: Represent density using a variant of Octrees



61

Tree-based Representations

- Efficient in space and time
- Multi-resolution



62