CSE-P590a
Robotics

**Bayes Filter Implementations**

Particle filters

1

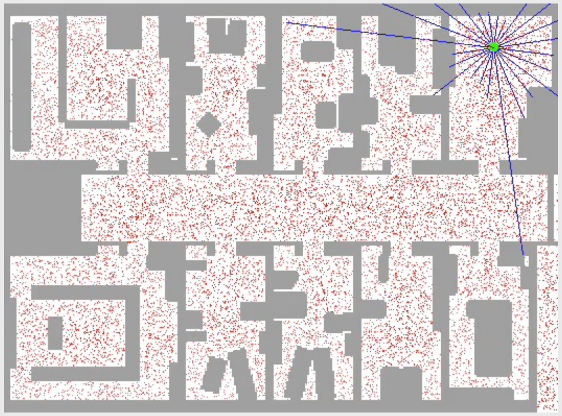
## Motivation

- 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)

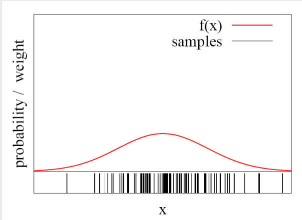


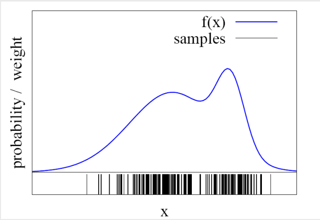
4/18/23 Probabilistic Robotics 3

3

## Density Approximation

- Particle sets can be used to approximate densities



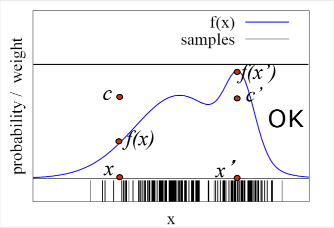

- The more particles fall into an interval, the higher the probability of that interval
- How to draw samples form a function/distribution?

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

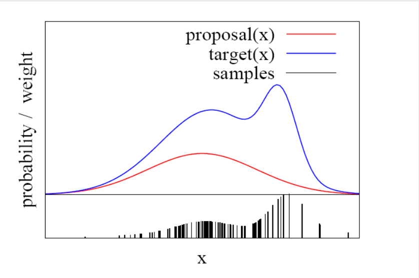



5

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

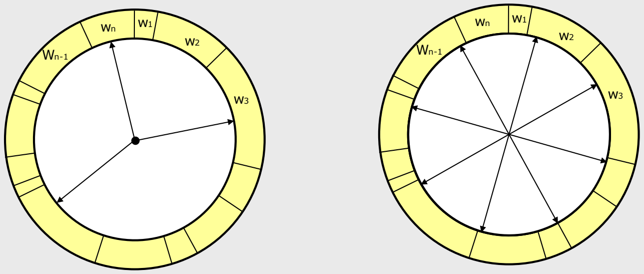
6

## 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'*.

7

## Resampling

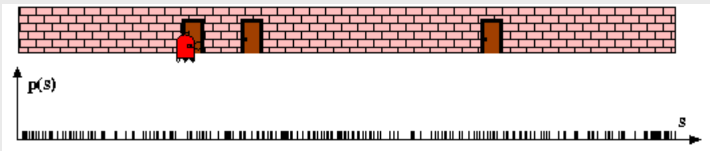



- Roulette wheel
- Binary search, n log n

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

8

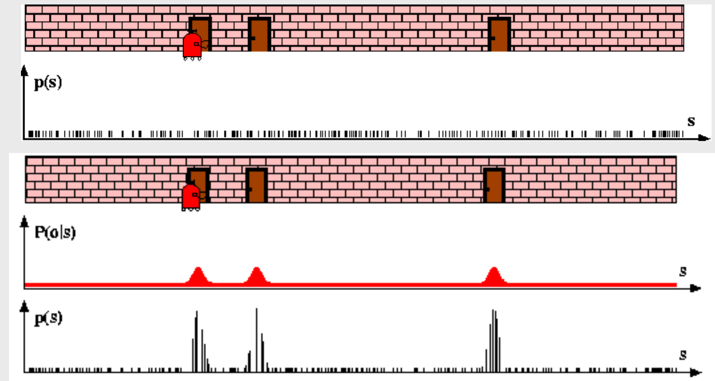
## Particle Filters




9

## Sensor Information: Importance Sampling

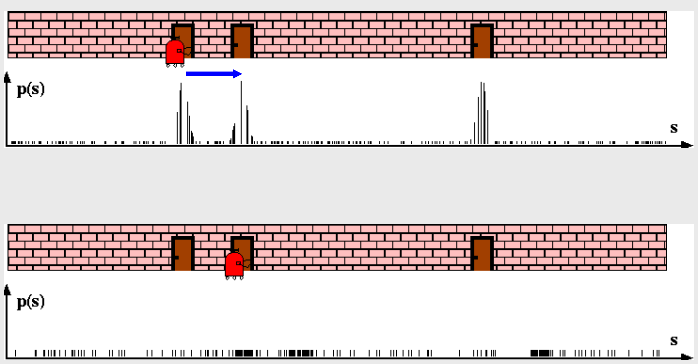
$$Bel(x) \leftarrow \alpha\, p(z\,|\,x)\, Bel^{-}(x)$$
$$w \leftarrow \frac{\alpha\, p(z\,|\,x)\, Bel^{-}(x)}{Bel^{-}(x)} = \alpha\, p(z\,|\,x)$$




10

## Robot Motion

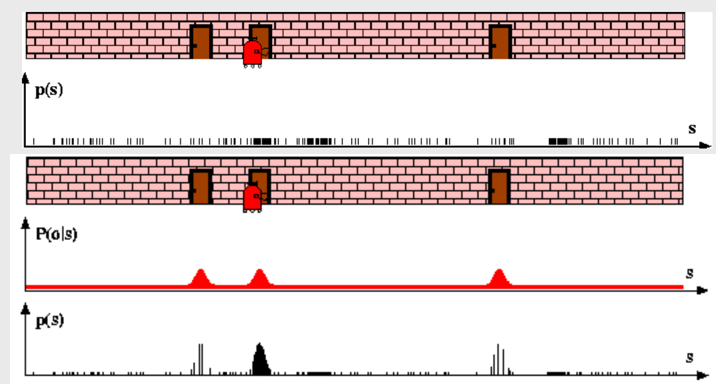
$$Bel^{-}(x) \leftarrow \int p(x\,|\,u, x')\, Bel(x')\; \mathrm{d}x'$$




11

## Sensor Information: Importance Sampling

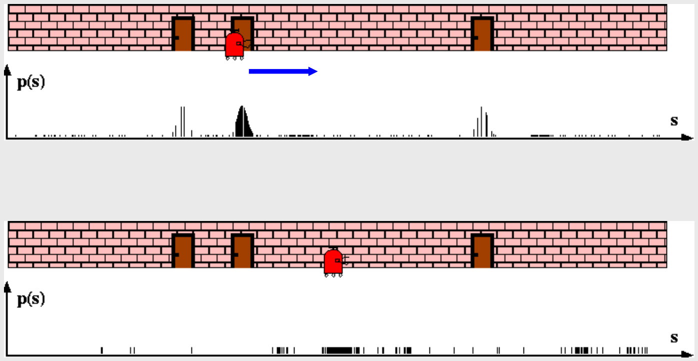
$$Bel(x) \leftarrow \alpha\, p(z\,|\,x)\, Bel^{-}(x)$$
$$w \leftarrow \frac{\alpha\, p(z\,|\,x)\, Bel^{-}(x)}{Bel^{-}(x)} = \alpha\, p(z\,|\,x)$$




12

## Robot Motion

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




13

## 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 $x^i_t$:

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

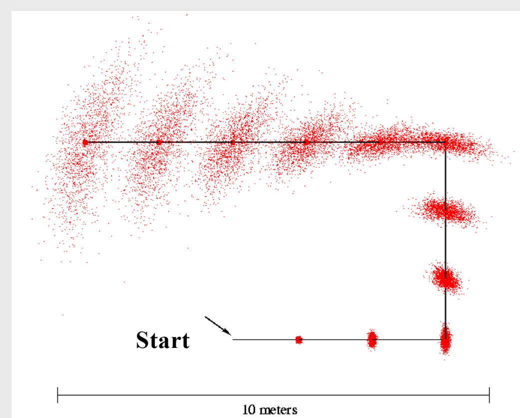
14

## 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 $x^i_t$ from $p(x_t \mid x_{t-1}, u_{t-1})$ using $x^{j(i)}_{t-1}$ and $u_{t-1}$
6. $w^i_t = p(z_t \mid x^i_t)$ — ***Compute importance weight***
7. $\eta = \eta + w^i_t$ — ***Update normalization factor***
8. $S_t = S_t \cup \{< x^i_t, w^i_t >\}$ — ***Insert***
9. **For** $i = 1 \ldots n$
10. $w^i_t = w^i_t / \eta$ — ***Normalize weights***

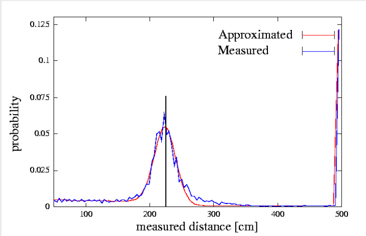
15

## Motion Model Reminder

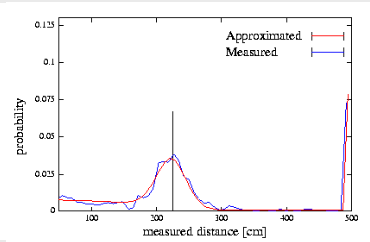



16

## Proximity Sensor Model Reminder

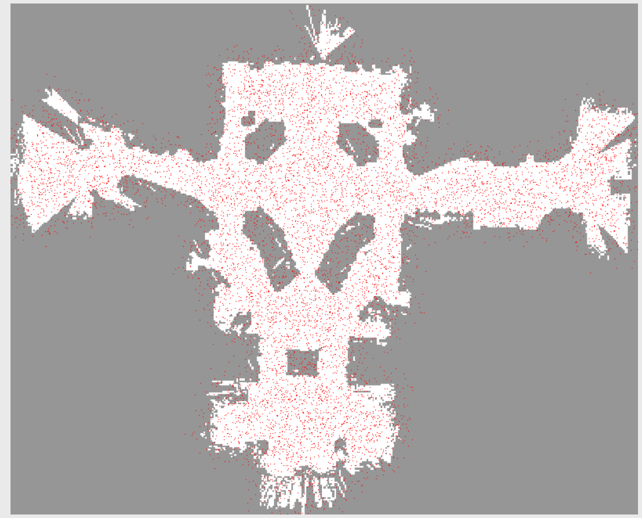



**Laser sensor**



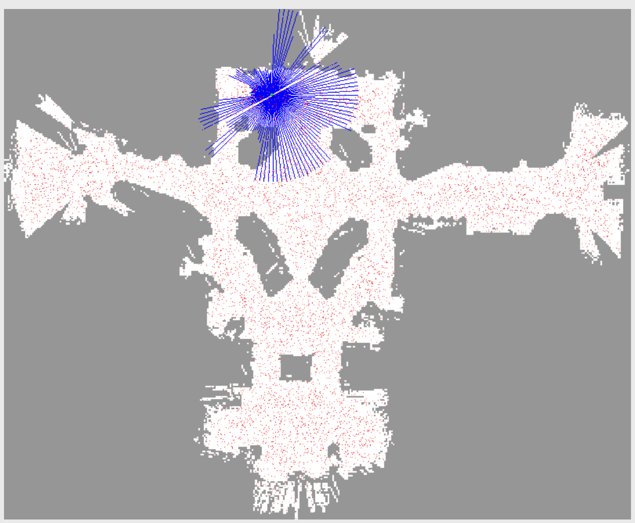

**Sonar sensor**

17



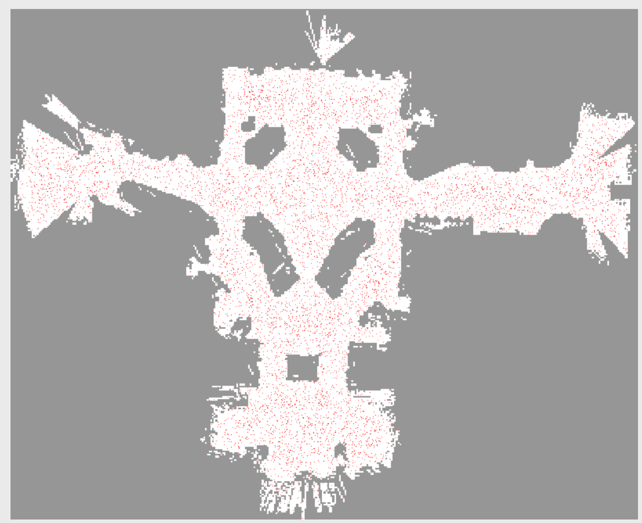
18

18



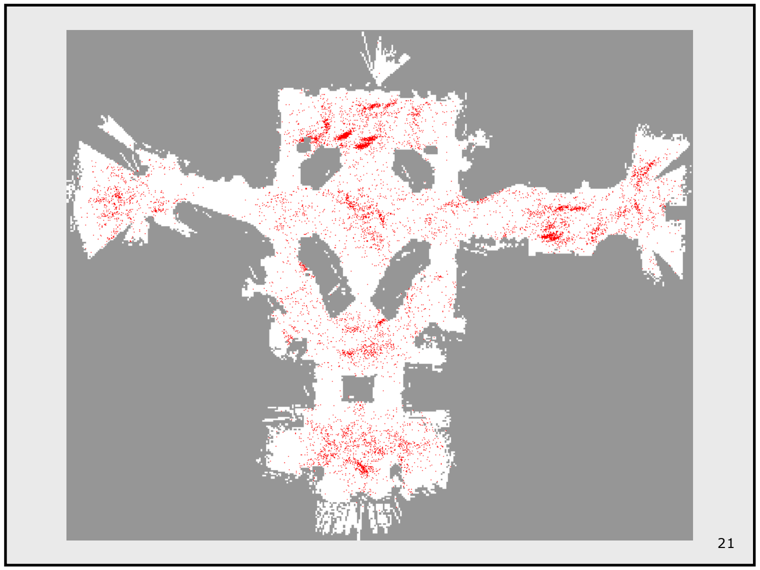
19

19

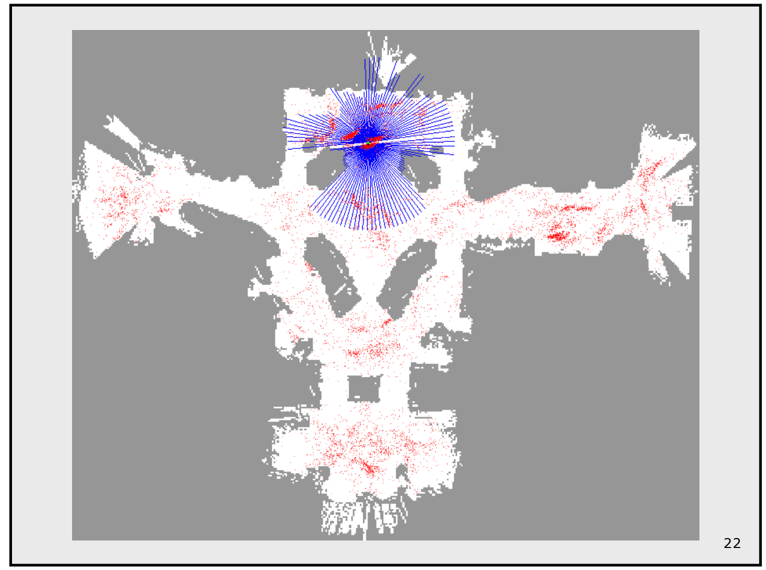


20

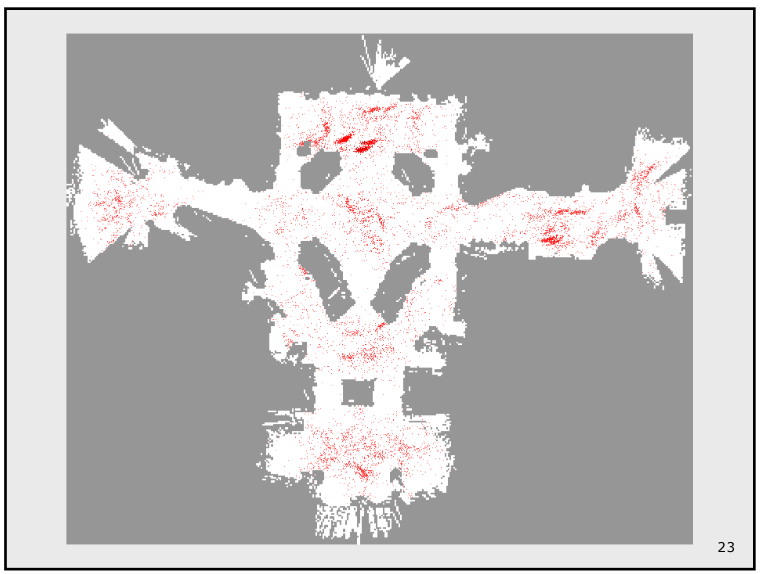
20




21

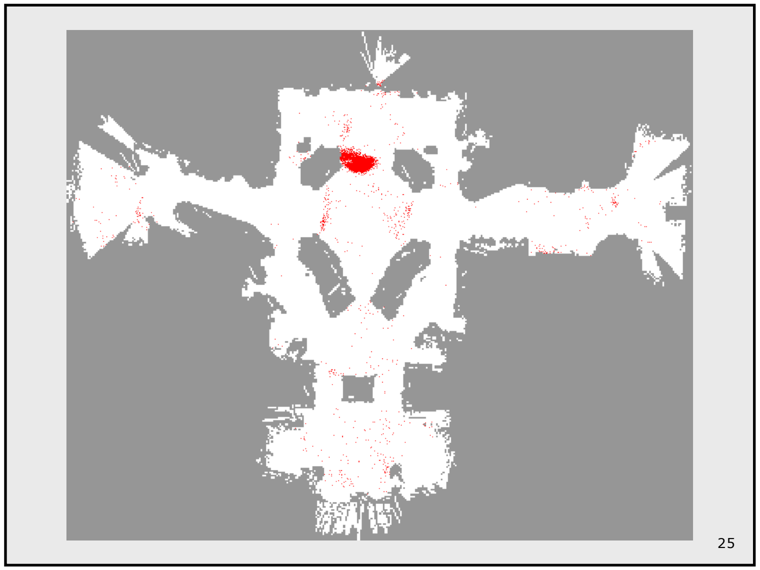



22

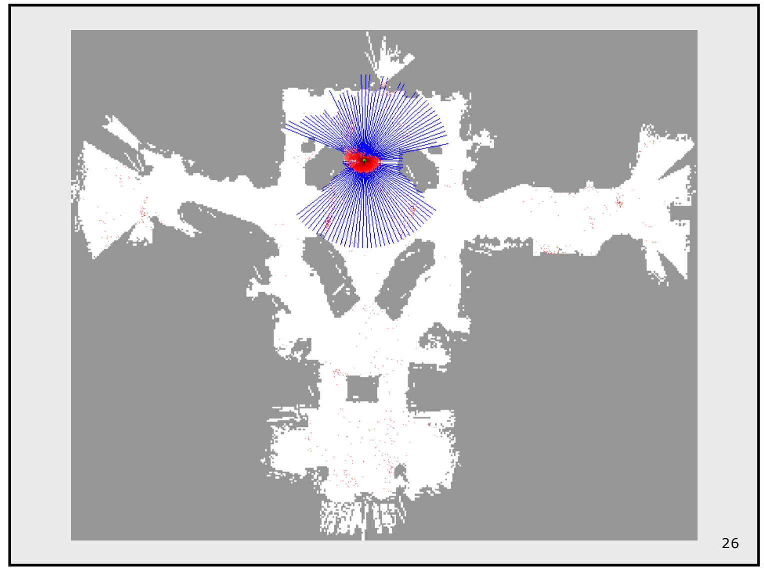



23

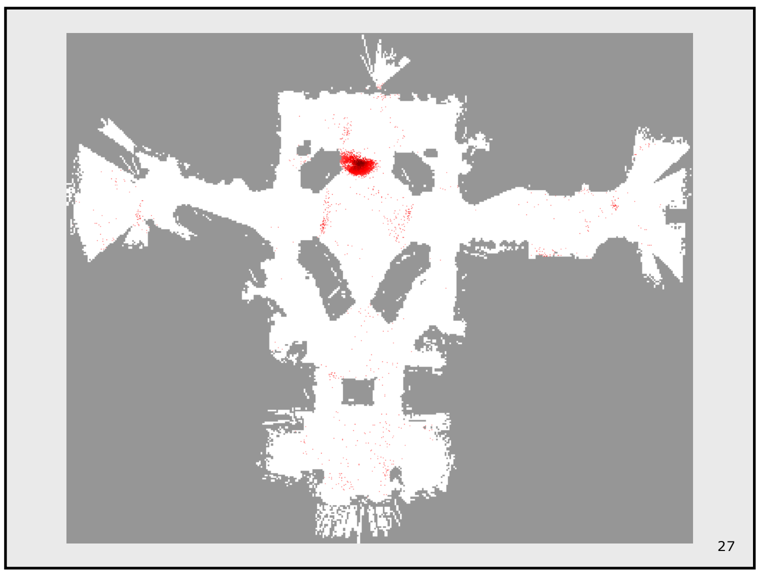

24



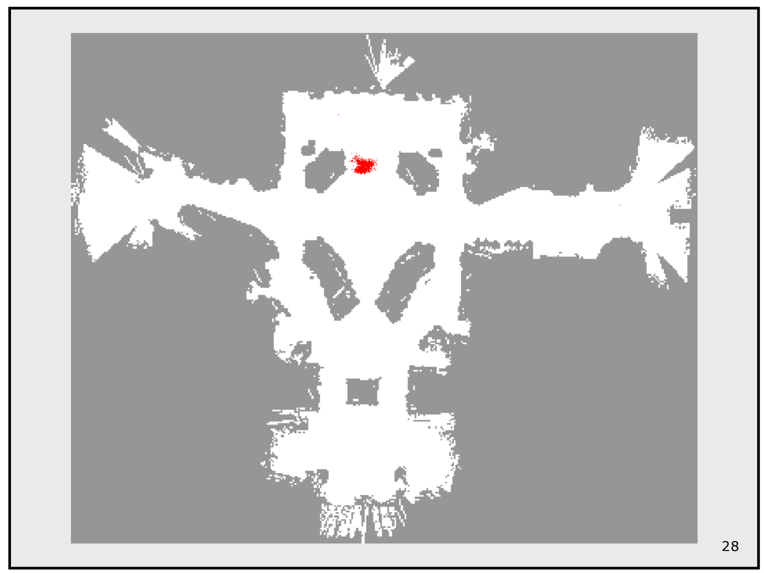

25



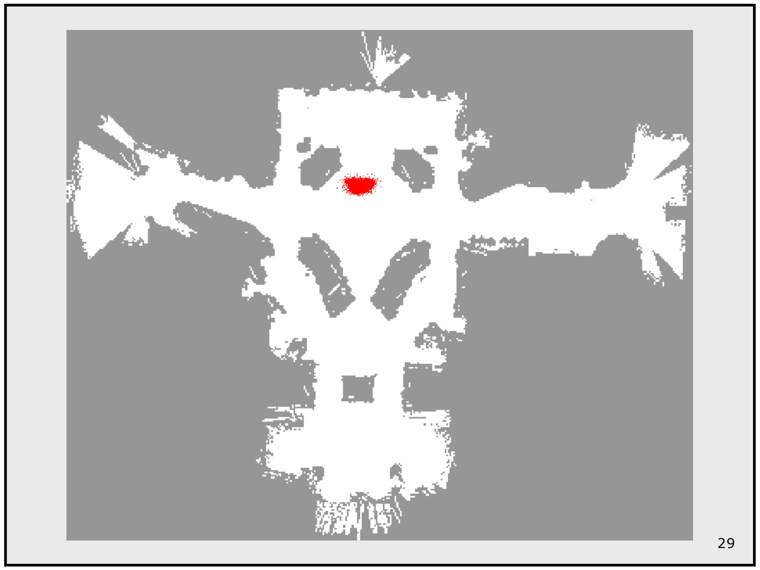

26



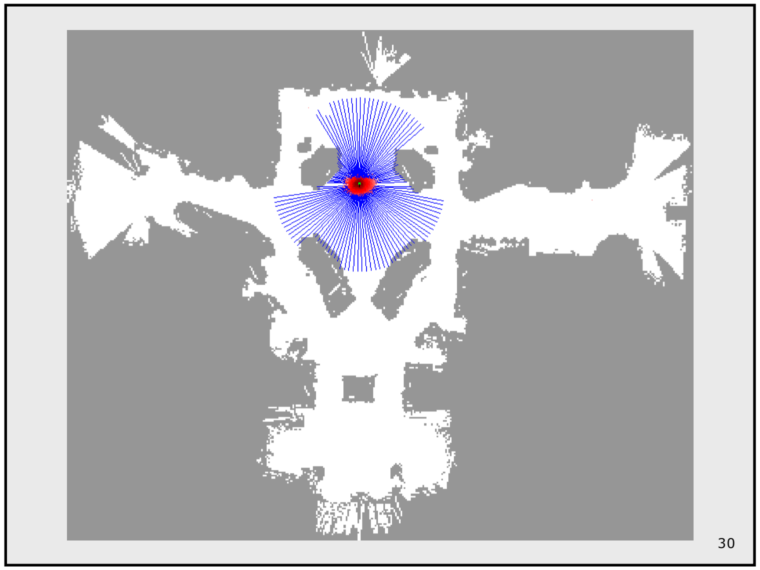

27



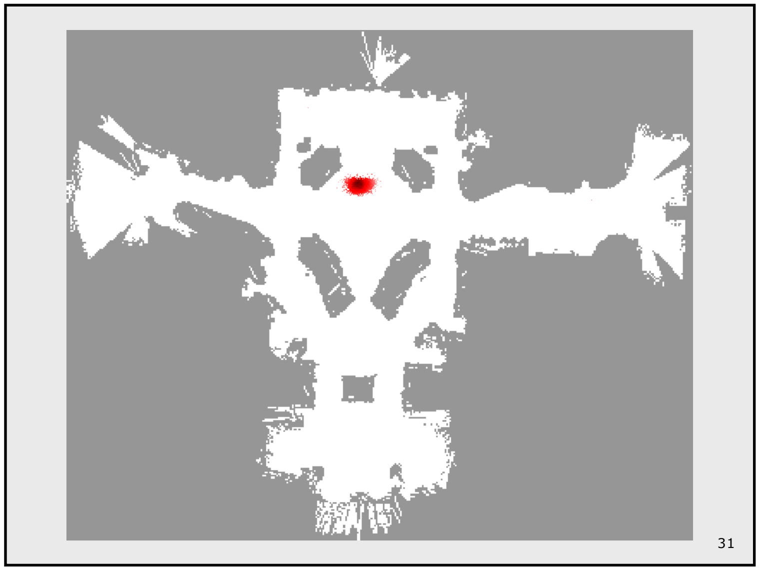

28



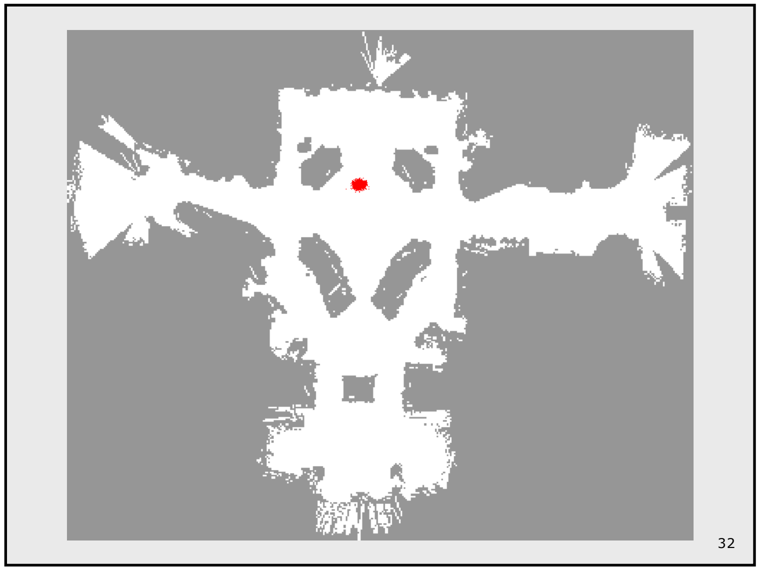

29



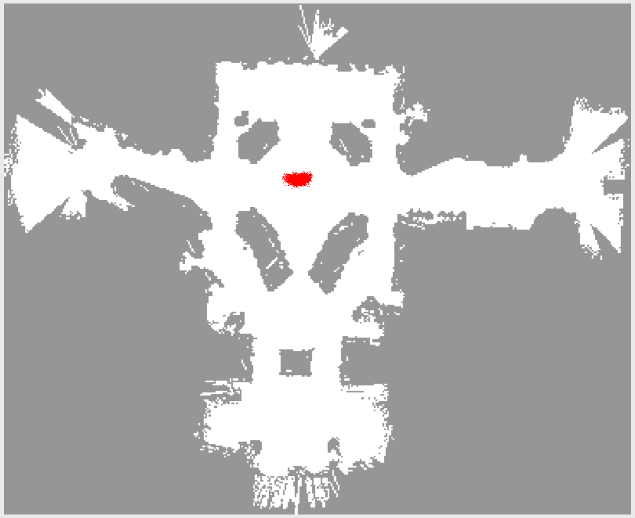

30




31

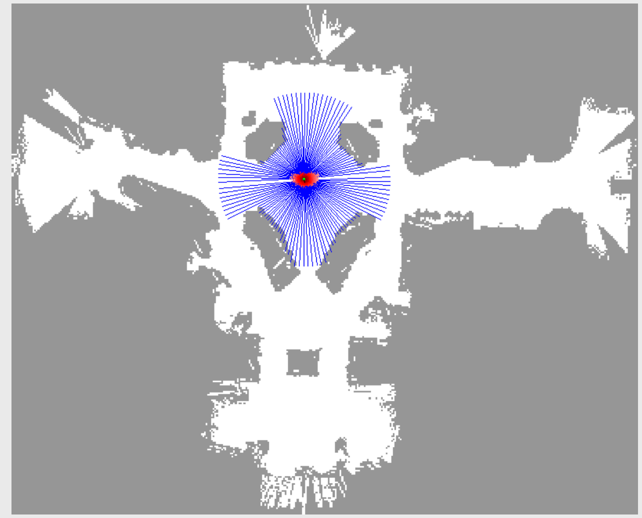



32



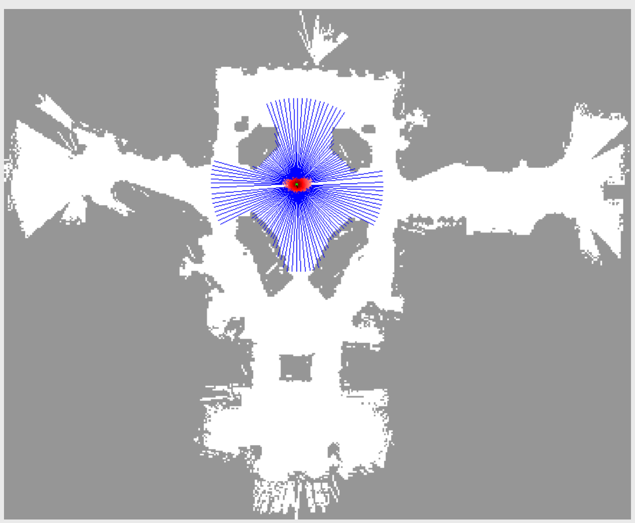
33

33



34

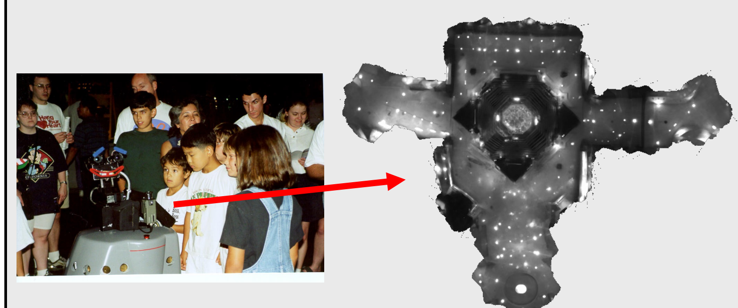
34



35

35

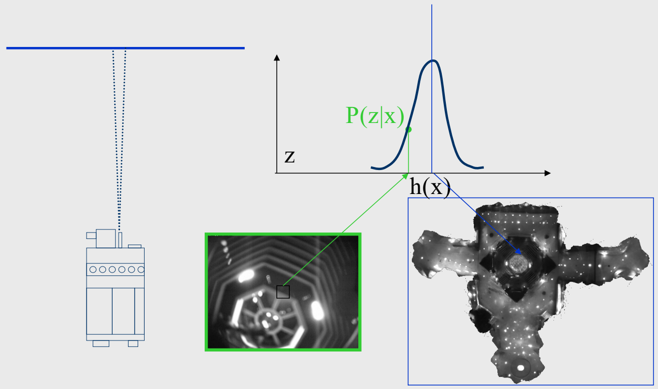
## Using Ceiling Maps for Localization



[Dellaert et al. 99]

36

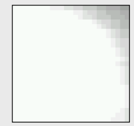
## Vision-based Localization



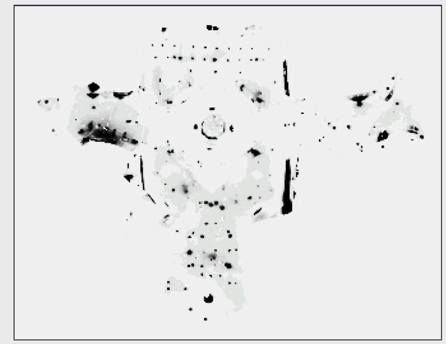

37

## Under a Light

**Measurement z:**



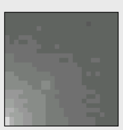
***P(z|x)*:**



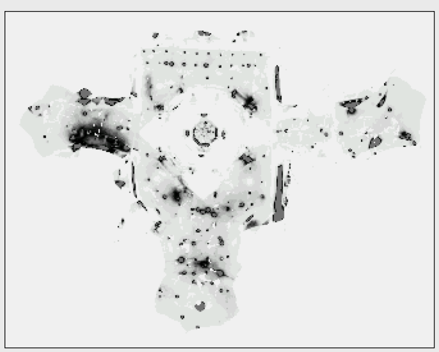
38

## Next to a Light

**Measurement z:**



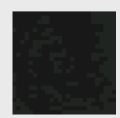
***P(z|x)*:**



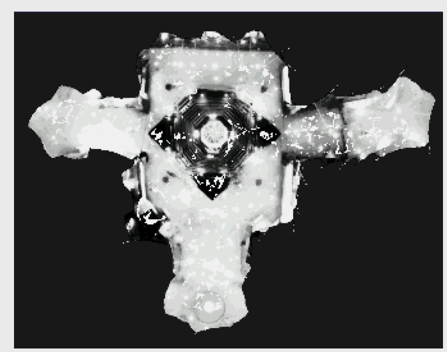
39

## Elsewhere

**Measurement z:**

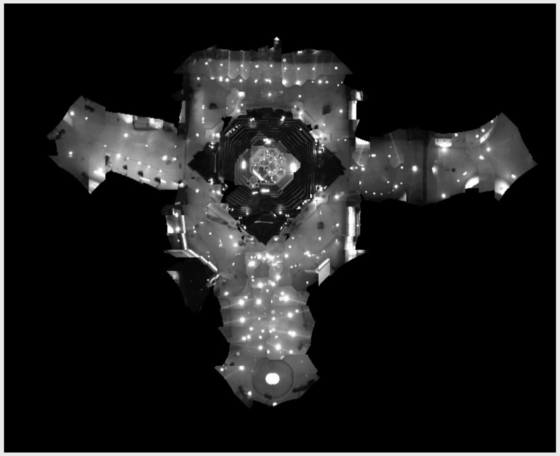


***P(z|x)*:**



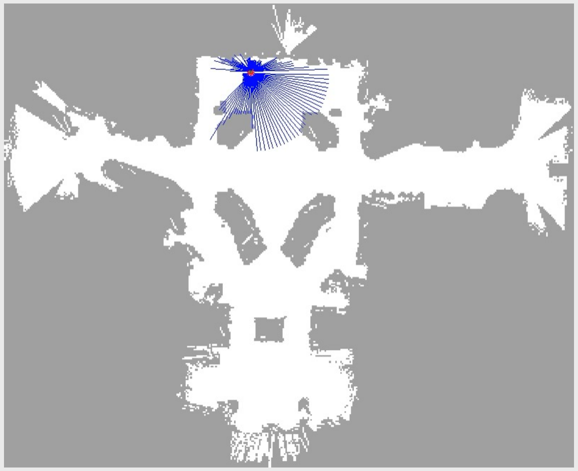
40

## Global Localization Using Vision



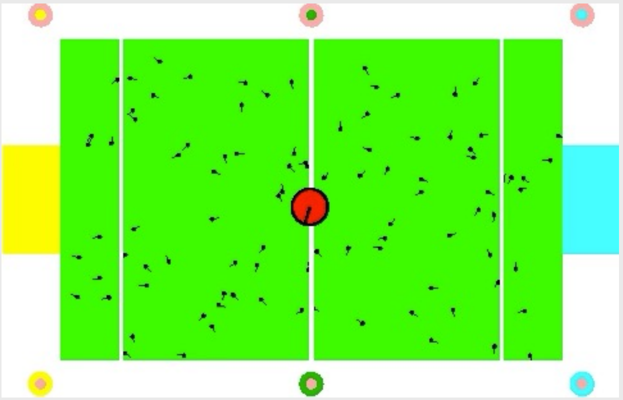
41

## Recovery from Failure



42

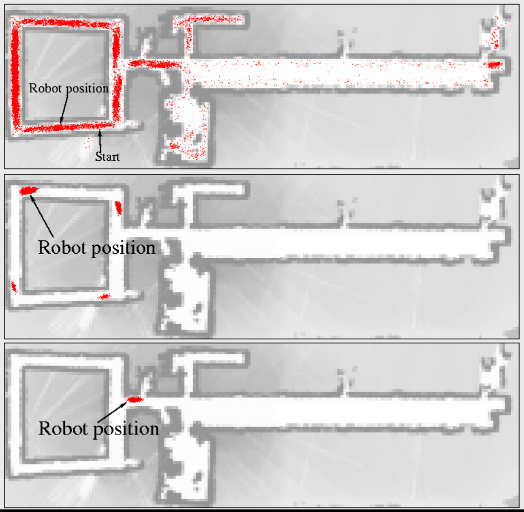
## Localization for AIBO robots



4/18/23 CSE-571 - AI-based Mobile Robotics 43

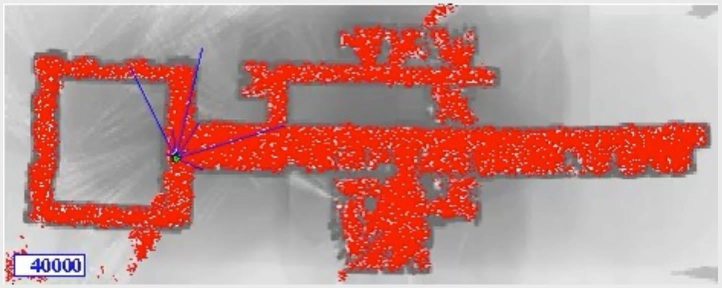
43

## Adaptive Sampling




44

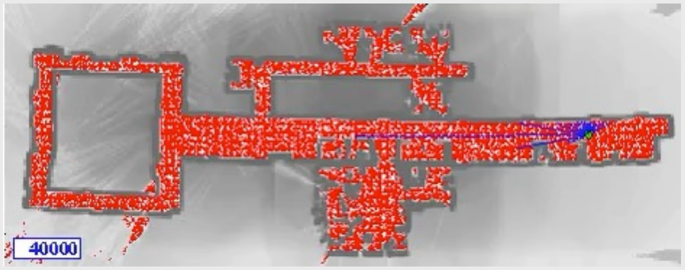
## KLD-Sampling Sonar




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

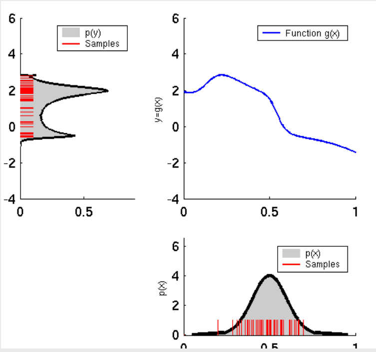
45

## KLD-Sampling Laser



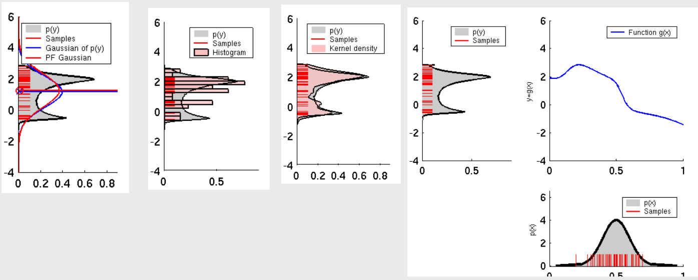

46

## Particle Filter Projection



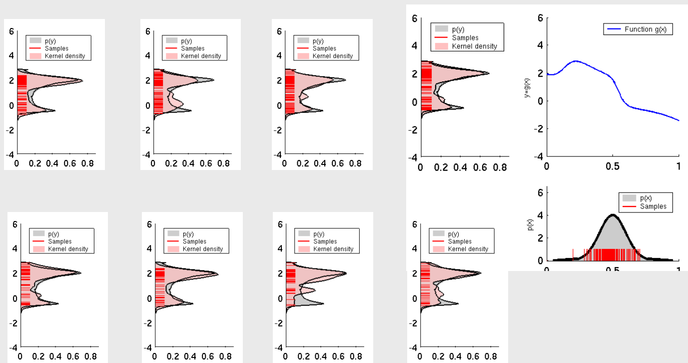

47

## Density Extraction




48

## Sampling Variance




49

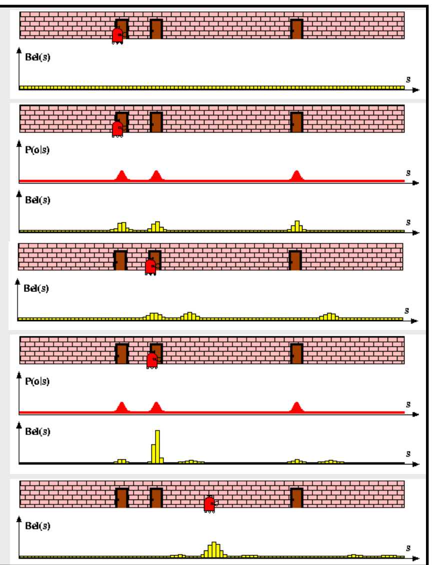
# CSE-P590a
# Robotics

## Bayes Filter Implementations

Discrete filters

50

Piecewise Constant






51

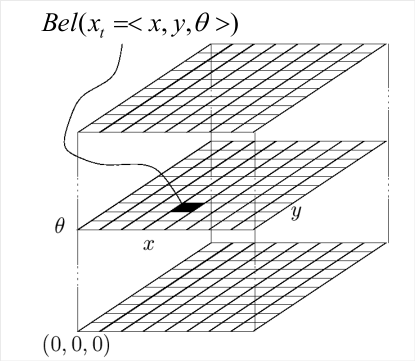
## Discrete Bayes Filter Algorithm

1. Algorithm **Discrete_Bayes_filter**( *Bel(x),d* ):
2. $\eta$=0
3. If *d* is a perceptual data item *z* then
4. For all *x* do
5. $Bel'(x) = P(z \mid x) Bel(x)$
6. $\eta = \eta + Bel'(x)$
7. For all *x* do
8. $Bel'(x) = \eta^{-1} Bel'(x)$
9. Else if *d* is an action data item *u* then
10. For all *x* do
11. $Bel'(x) = \sum_{x'} P(x \mid u, x')\, Bel(x')$
12. Return *Bel'(x)*



52

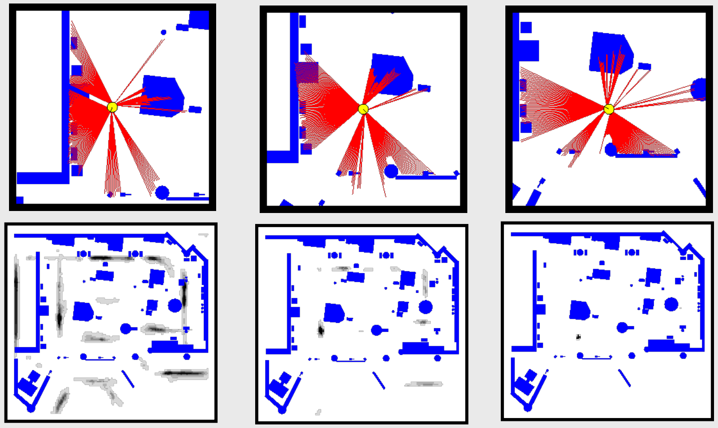
# Piecewise Constant Representation






53

# Grid-based Localization





54