CSE-P590a
Robotics

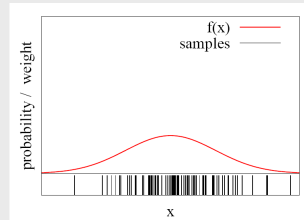
Bayes Filter Implementations

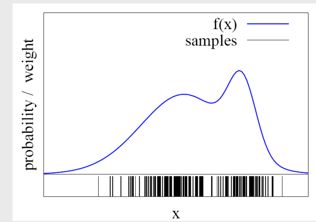
Particle filters

1

Density Approximation

- Particle sets can be used to approximate densities





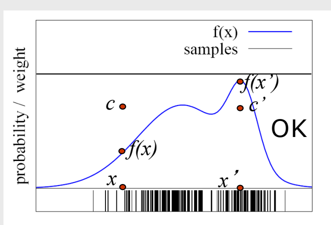
- The more particles fall into an interval, the higher the probability of that interval
- How to draw samples form a function/distribution?

2

2

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

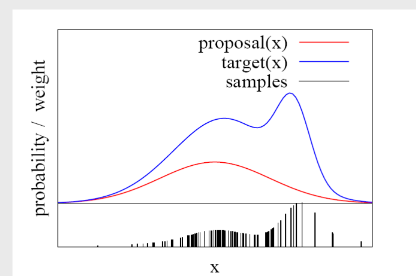


3

3

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



4

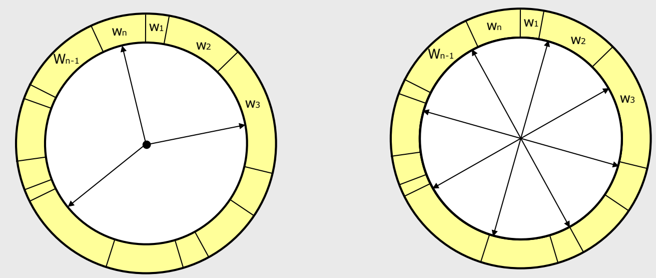
4

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'.

5

Resampling

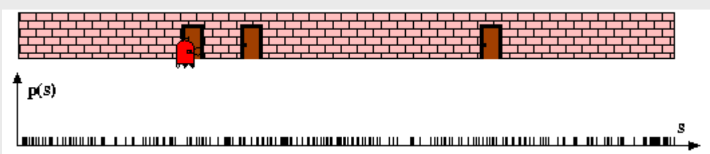


- Roulette wheel
- Binary search, n log n

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

6

Particle Filters

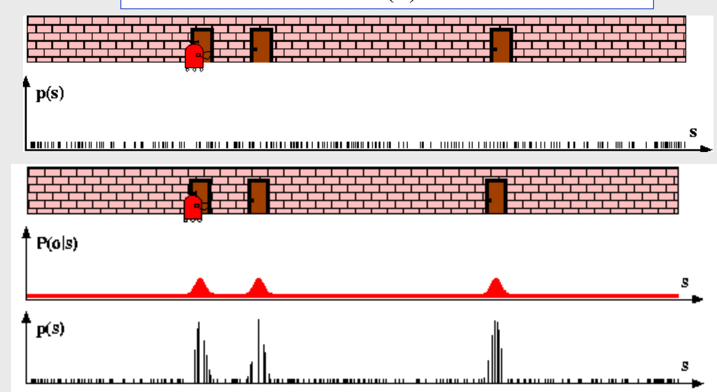


7

Sensor Information: Importance Sampling

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

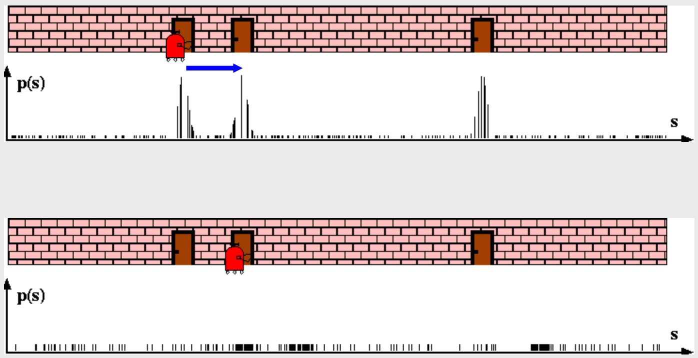
$$w \leftarrow \frac{\alpha\ p(z\,|\,x)\, Bel^{-}(x)}{Bel^{-}(x)} = \alpha\ p(z\,|\,x)$$



8

Robot Motion

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

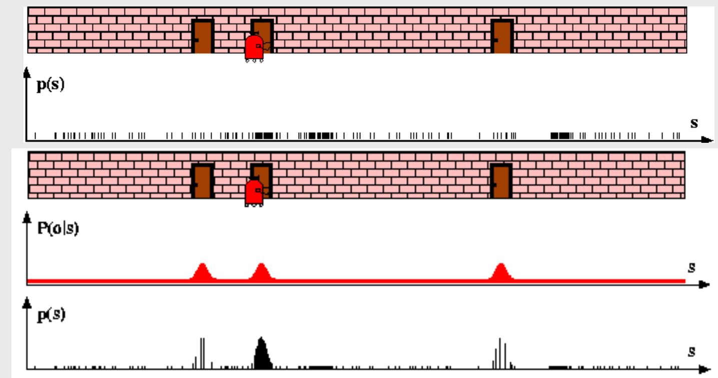


9

Sensor Information: Importance Sampling

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

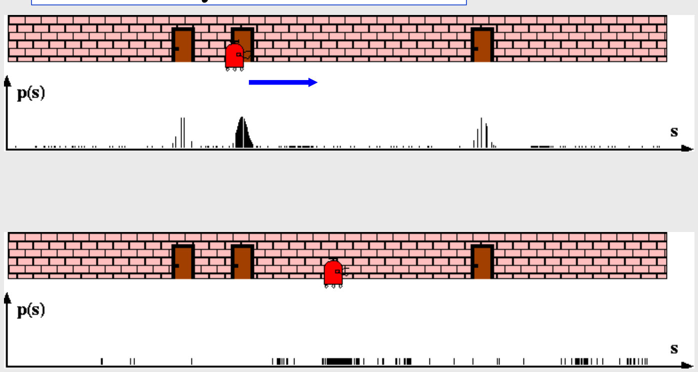
$$w \leftarrow \frac{\alpha\, p(z \mid x)\, Bel^-(x)}{Bel^-(x)} = \alpha\, p(z \mid x)$$



10

Robot Motion

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



11

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:

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

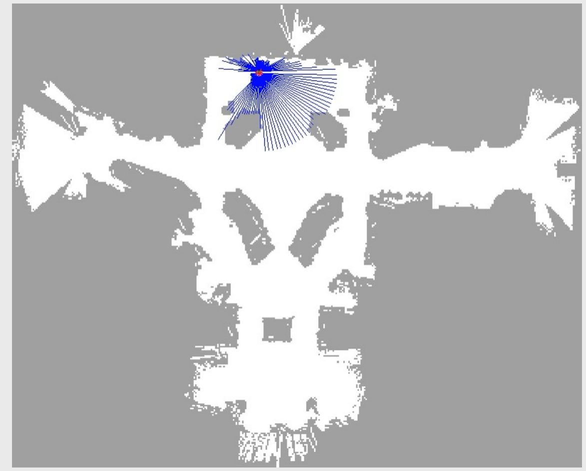
12

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***

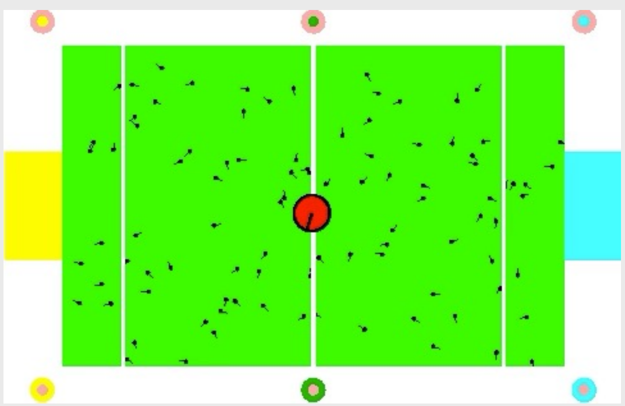
13

Recovery from Failure



14

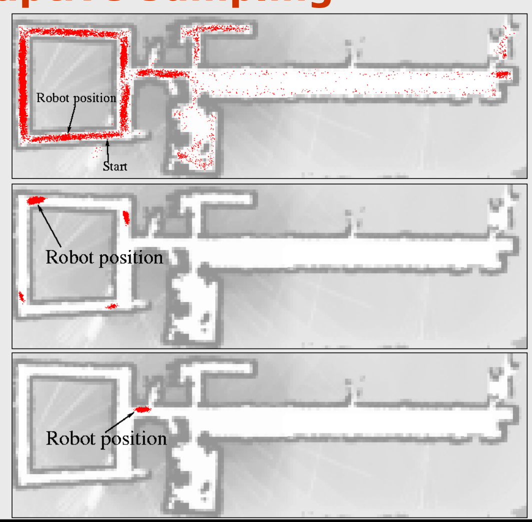
Localization for AIBO robots



4/3/24 CSE-571 - AI-based Mobile Robotics 15

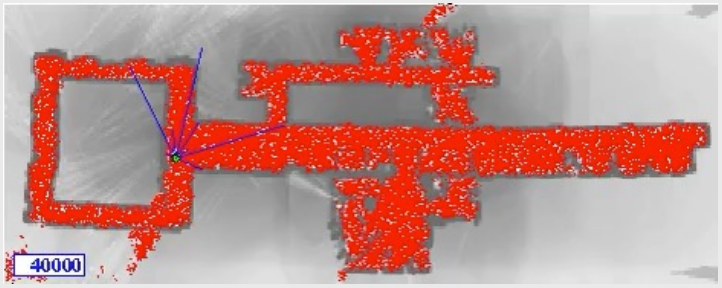
15

Adaptive Sampling



16

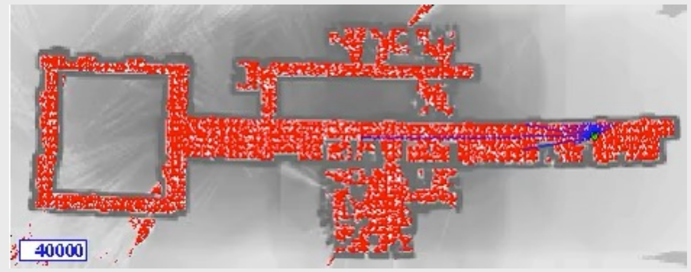
KLD-Sampling Sonar



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

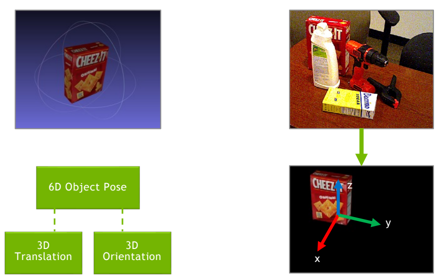
17

KLD-Sampling Laser



18

6D OBJECT POSE ESTIMATION

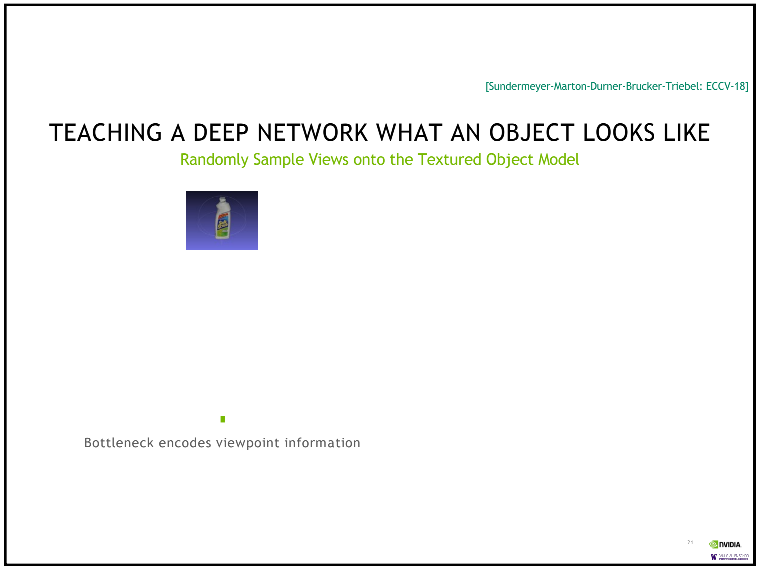


19

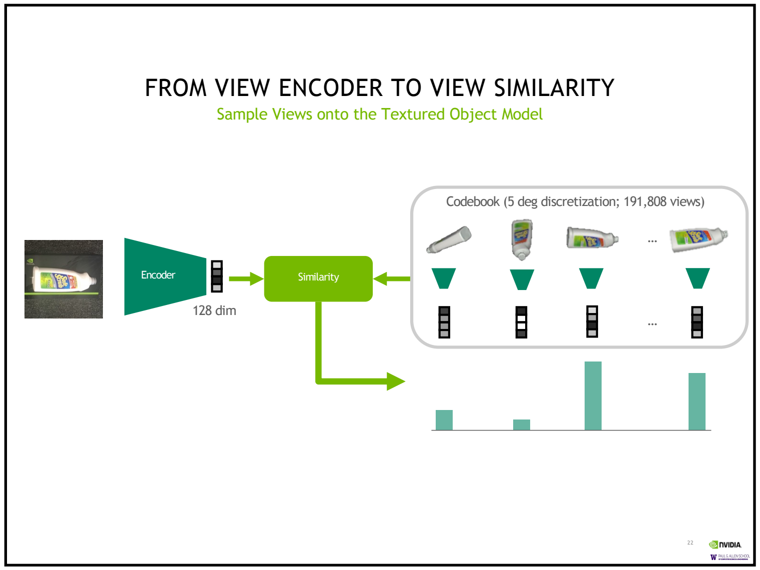
ORIENTATION UNCERTAINTY

Depends on context, shape, sensor

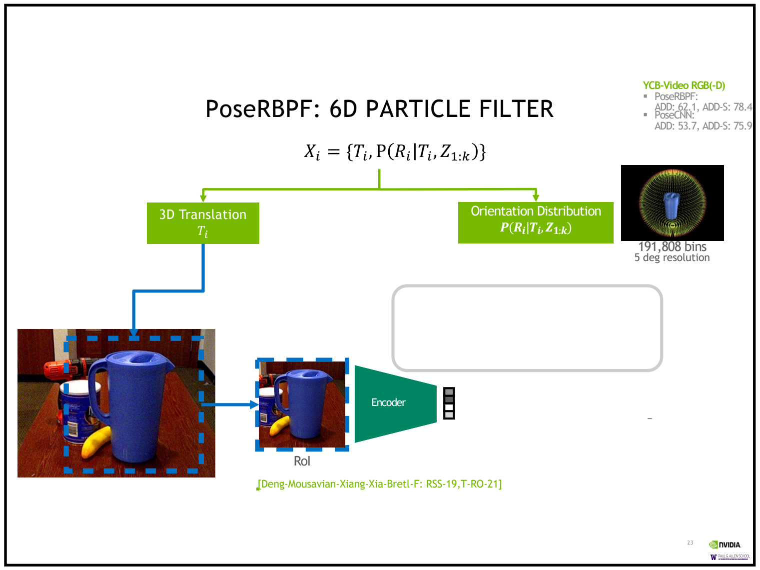
20


[Sundermeyer-Marton-Durner-Brucker-Triebel: ECCV-18]
TEACHING A DEEP NETWORK WHAT AN OBJECT LOOKS LIKE
Randomly Sample Views onto the Textured Object Model
Bottleneck encodes viewpoint information
NVIDIA

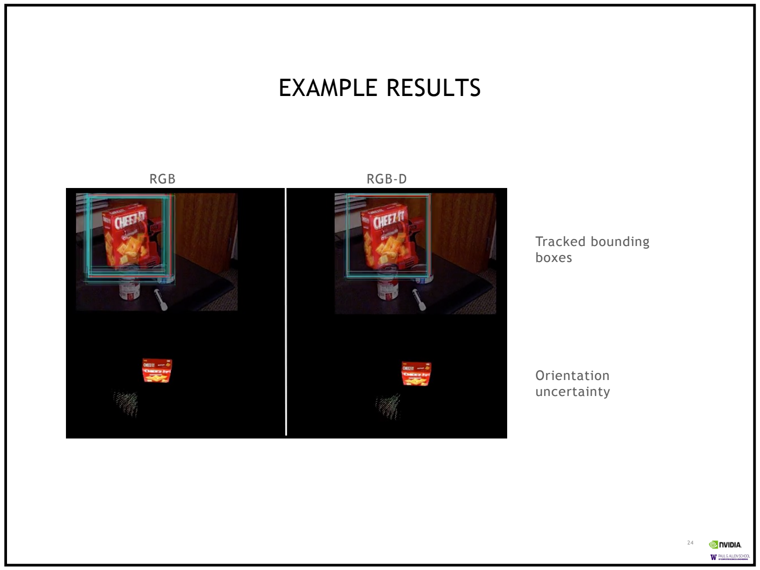
21


FROM VIEW ENCODER TO VIEW SIMILARITY
Sample Views onto the Textured Object Model
Codebook (5 deg discretization; 191,808 views)
Encoder
128 dim
Similarity
NVIDIA

22


PoseRBPF: 6D PARTICLE FILTER
YCB-Video RGB(-D)
PoseRBPF: ADD: 62.1, ADD-S: 78.4
PoseCNN: ADD: 53.7, ADD-S: 75.9
$X_i = \{T_i, P(R_i|T_i, Z_{1:k})\}$
3D Translation T_i
Orientation Distribution $P(R_i|T_i, Z_{1:k})$
191,808 bins
5 deg resolution
Encoder
RoI
[Deng-Mousavian-Xiang-Xia-Bretl-F: RSS-19,T-RO-21]
NVIDIA

23


EXAMPLE RESULTS
RGB
RGB-D
Tracked bounding boxes
Orientation uncertainty
NVIDIA

24

GLOBAL LOCALIZATION EXAMPLE

Sample Uniformly in Translation Space

1st frame: 5,000 particles, then 500 particles until strong match, then 50 particles
500 particles: 2.6 fps; 50 particles: 20 fps