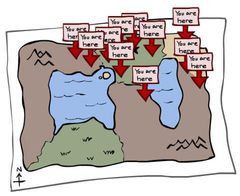
CSE 473: Artificial Intelligence

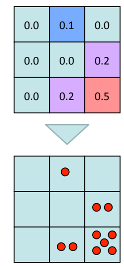
Particle Filters



Dieter Fox --- University of Washington

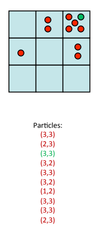
Particle Filtering

- Filtering: approximate solution
- Sometimes |X| is too big to use exact inference
  - |X| may be too big to even store B(X)
  - E.g. X is continuous
  - $|X|^2$ may be too big to do updates
- Solution: approximate inference
  - Track samples of X, not all values
  - Samples are called particles
  - Time per step is linear in the number of samples
  - But: number needed may be large
  - In memory: list of particles, not states
- This is how robot localization works in practice



Representation: Particles

- Our representation of P(X) is now a list of N particles (samples)
  - Generally, N << |X|
  - Storing map from X to counts would defeat the point
- P(x) approximated by number of particles with value x
  - So, many x may have P(x) = 0!
  - More particles, more accuracy
- For now, all particles have a weight of 1

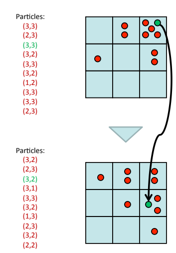


Particle Filtering: Elapse Time

- Each particle is moved by sampling its next position from the transition model

$$x' = \text{sample}(P(X'|x))$$

  - This is like prior sampling – samples' frequencies reflect the transition probabilities
  - Here, most samples move clockwise, but some move in another direction or stay in place
- This captures the passage of time
  - If enough samples, close to exact values before and after (consistent)



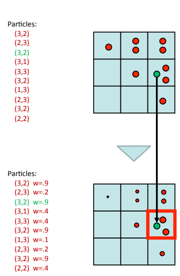
Particle Filtering: Observe

- Slightly trickier:
  - Don't sample observation, fix it
  - Similar to likelihood weighting, downweight samples based on the evidence

$$w(x) = P(e|x)$$

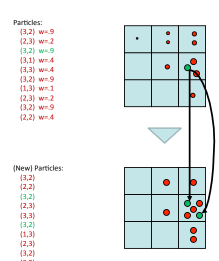
$$B(X) \propto P(e|X)B'(X)$$

  - As before, the probabilities don't sum to one, since all have been downweighted (in fact they now sum to (N times) an approximation of P(e))



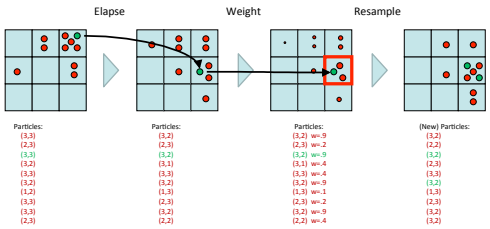
Particle Filtering: Resample

- Rather than tracking weighted samples, we resample
- N times, we choose from our weighted sample distribution (i.e. draw with replacement)
- This is equivalent to renormalizing the distribution
- Now the update is complete for this time step, continue with the next one



Recap: Particle Filtering

- Particles: track samples of states rather than an explicit distribution



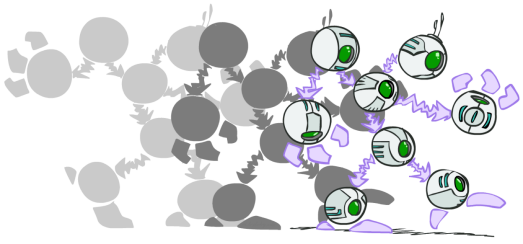
[Demos: ghostbusters particle filtering (L15D3,4,5)]

Video of Demo – Moderate Number of Particles

Video of Demo – One Particle

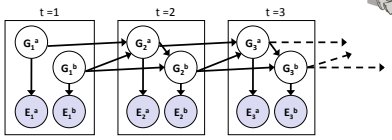
Video of Demo – Huge Number of Particles

Dynamic Bayes Nets



Dynamic Bayes Nets (DBNs)

- We want to track multiple variables over time, using multiple sources of evidence
- Idea: Repeat a fixed Bayes net structure at each time
- Variables from time *t* can condition on those from *t-1*



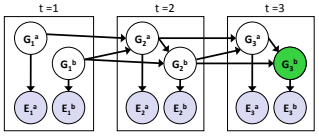
- Dynamic Bayes nets are a generalization of HMMs

[Demo: pacman sonar ghost DBN model (L15D6)]

Video of Demo Pacman Sonar Ghost DBN Model

Exact Inference in DBNs

- Variable elimination applies to dynamic Bayes nets
- Procedure: "unroll" the network for T time steps, then eliminate variables until $P(X_T|e_{1:T})$ is computed



- Online belief updates: Eliminate all variables from the previous time step; store factors for current time only

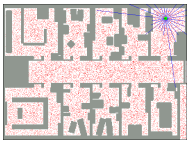
DBN Particle Filters

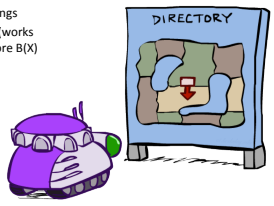
- A particle is a complete sample for a time step
- **Initialize**: Generate prior samples for the t=1 Bayes net
  - Example particle: $\mathbf{G_1^a} = (3,3)$ $\mathbf{G_1^b} = (5,3)$
- **Elapse time**: Sample a successor for each particle
  - Example successor: $\mathbf{G_2^a} = (2,3)$ $\mathbf{G_2^b} = (6,3)$
- **Observe**: Weight each *entire* sample by the likelihood of the evidence conditioned on the sample
  - Likelihood: $P(\mathbf{E_1^a}|\mathbf{G_1^a}) * P(\mathbf{E_1^b}|\mathbf{G_1^b})$
- **Resample:** Select prior samples (tuples of values) in proportion to their likelihood

Some More Thoughts on Particle Filters and Sampling

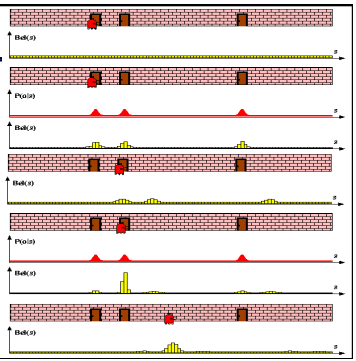
Robot Localization

- In robot localization:
  - We know the map, but not the robot's position
  - Observations may be vectors of range finder readings
  - State space and readings are typically continuous (works basically like a very fine grid) and so we cannot store B(X)
  - Particle filtering is a main technique

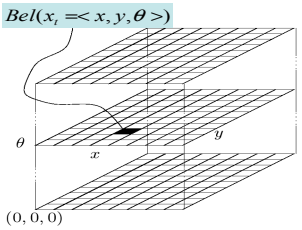




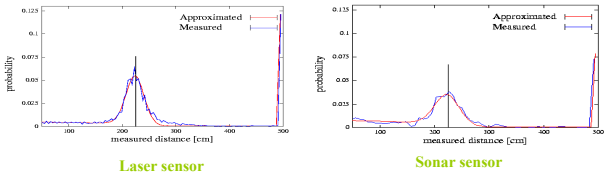
Piecewise Constant Belief



Piecewise Constant Representation



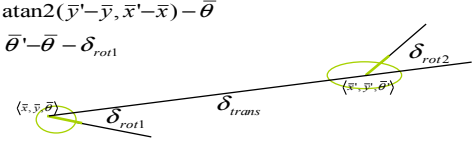
Proximity Sensor Model



Probabilistic Kinematics

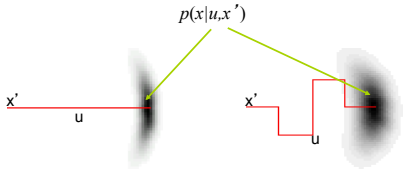
- Robot moves from $\langle \bar{x}, \bar{y}, \bar{\theta} \rangle$ to $\langle \bar{x}', \bar{y}', \bar{\theta}' \rangle$
- Odometry information $u = \langle \delta_{rot1}, \delta_{rot2}, \delta_{trans} \rangle$

$$\delta_{trans} = \sqrt{(\bar{x}'-\bar{x})^2 + (\bar{y}'-\bar{y})^2}$$
$$\delta_{rot1} = \text{atan2}(\bar{y}'-\bar{y}, \bar{x}'-\bar{x}) - \bar{\theta}$$
$$\delta_{rot2} = \bar{\theta}'-\bar{\theta} - \delta_{rot1}$$

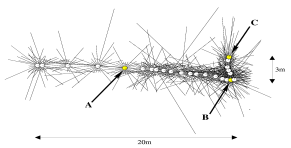


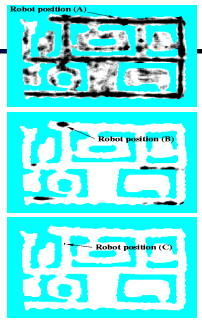
Probabilistic Kinematics

- Odometry information is inherently noisy.

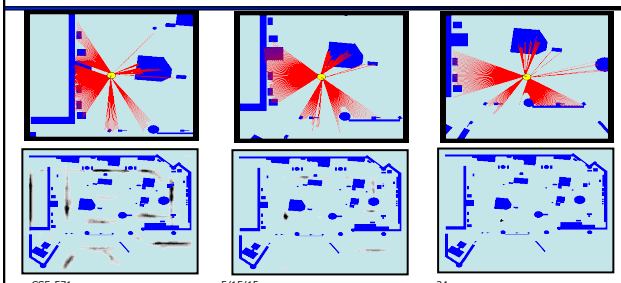


Sonars and Occupancy Grid Map



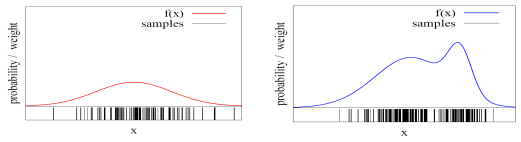


Laser-based Localization



Sample-Based Density Approximation

- Particle sets can be used to approximate densities

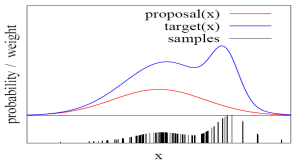


- The more particles fall into an interval, the higher the probability of that interval
- How to draw samples form a function/distribution?

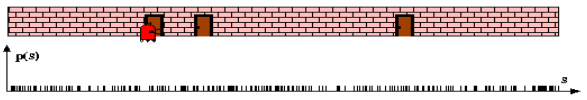
25

Importance Sampling Principle

- We can 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



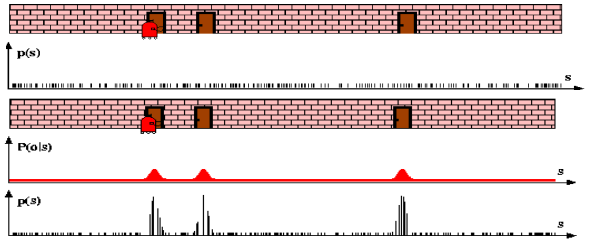
Particle Filters



Sensor Information: Importance Sampling

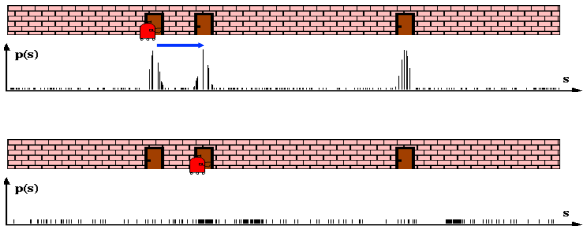
$$Bel(x) \leftarrow \alpha\, p(z \mid x)\, Bel^{-}(x)$$

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



Robot Motion

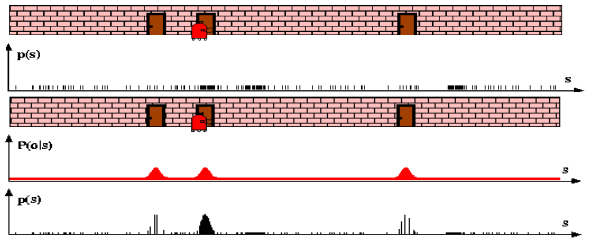
$$Bel^{-}(x) \leftarrow \int p(x \mid u, x')\, Bel(x')\, dx'$$



Sensor Information: Importance Sampling

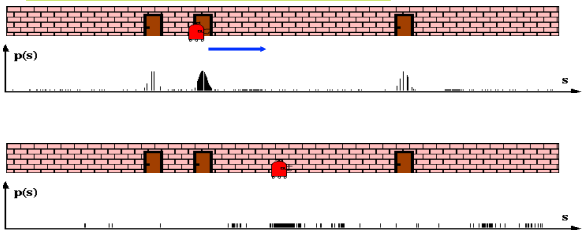
$$Bel(x) \leftarrow \alpha\, p(z \mid x)\, Bel^{-}(x)$$

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



Robot Motion

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



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

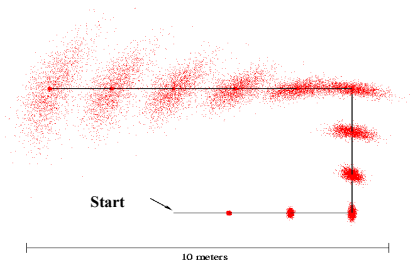
Importance factor for x^i_t:

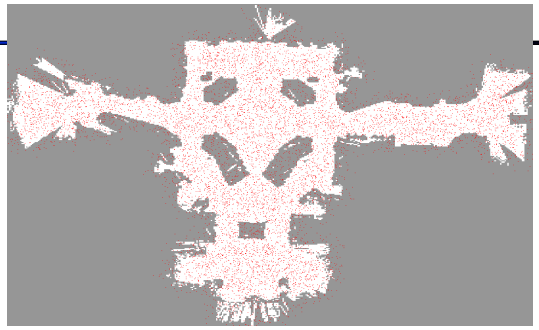
$$w^i_t = \frac{\text{target distribution}}{\text{proposal distribution}}$$

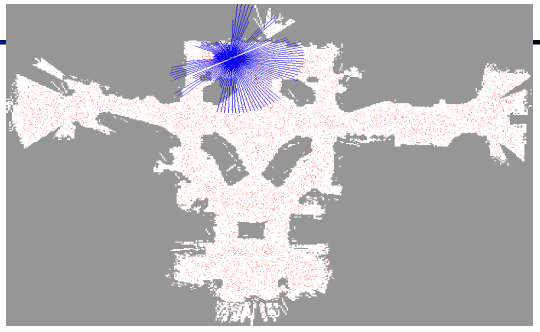
$$= \frac{\eta \, p(z_t \mid x_t) \, p(x_t \mid x_{t-1}, u_{t-1}) \, Bel(x_{t-1})}{p(x_t \mid x_{t-1}, u_{t-1}) \, Bel(x_{t-1})}$$

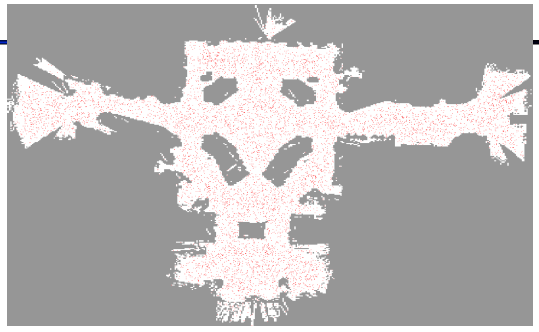
$$\propto p(z_t \mid x_t)$$

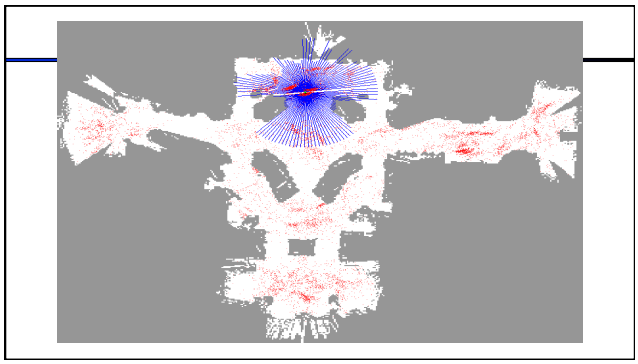
Sampled Motion Model

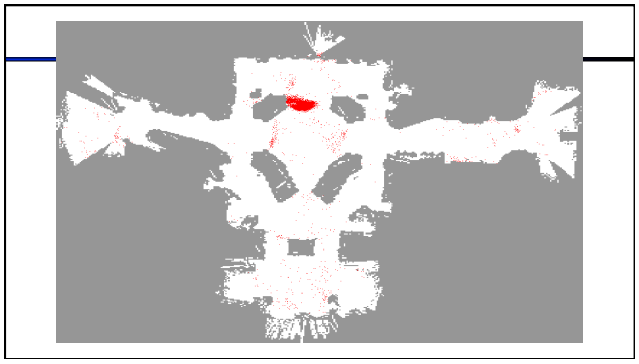


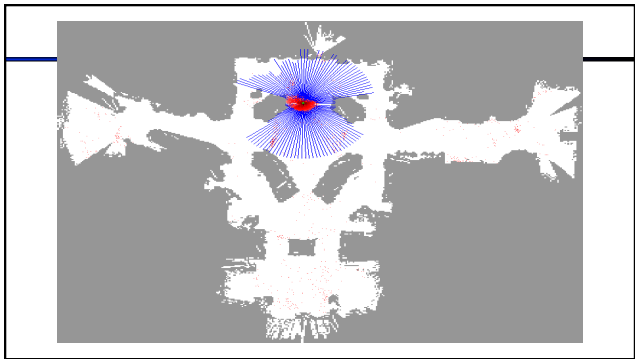


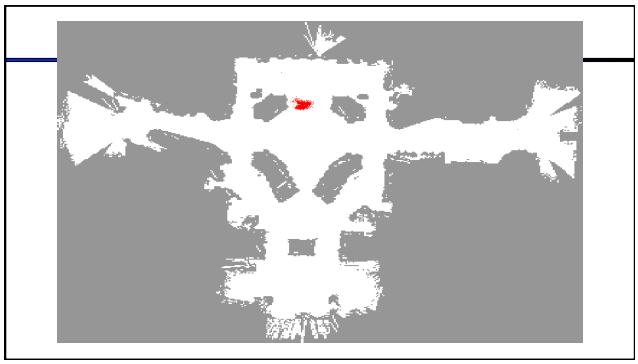


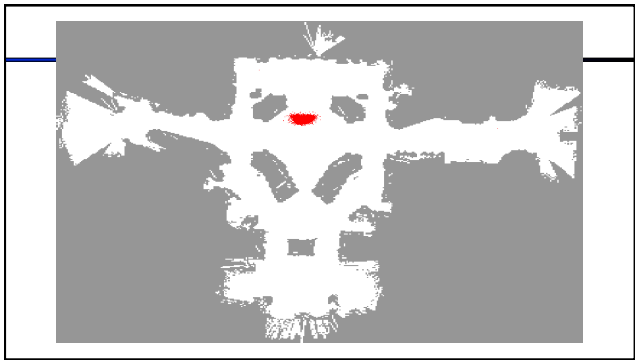


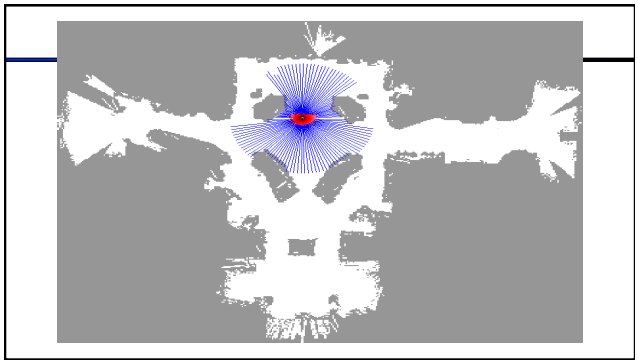


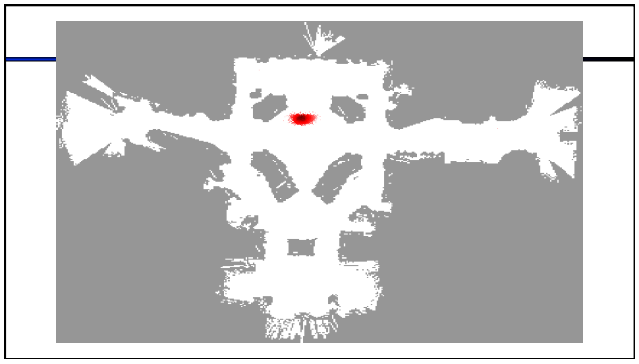


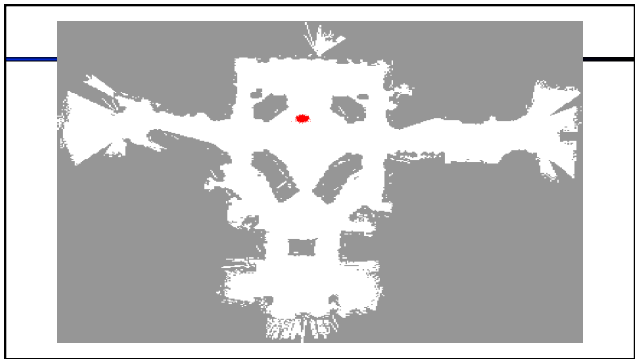


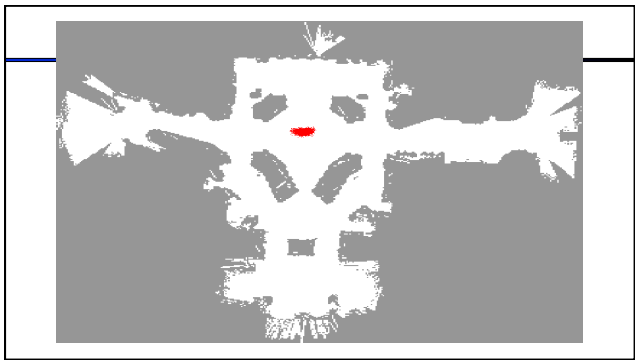


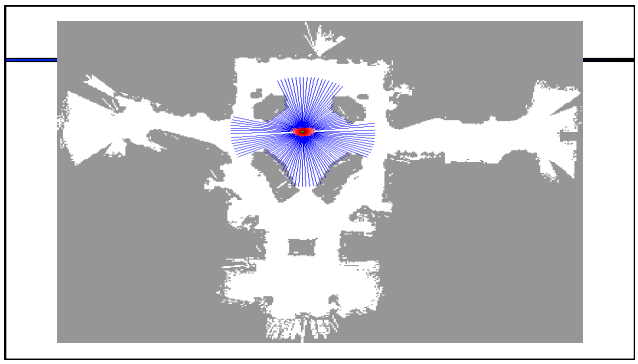


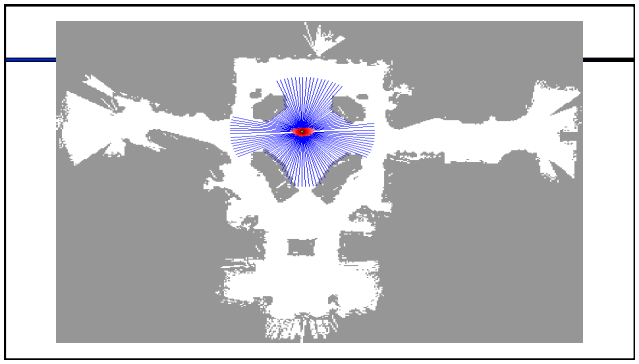




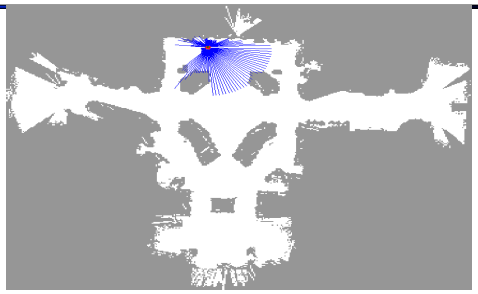




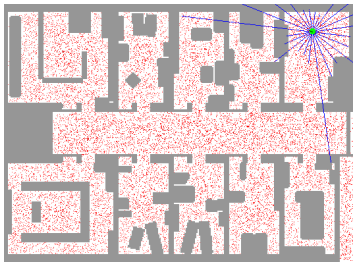




Recovery from Failure

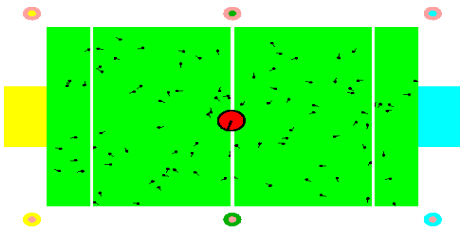


Particle Filter Localization (Sonar)

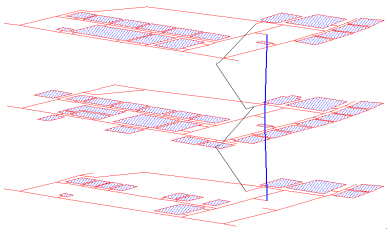


[Video: global-floor.gif]

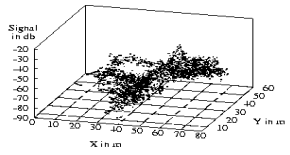
Localization for AIBO robots

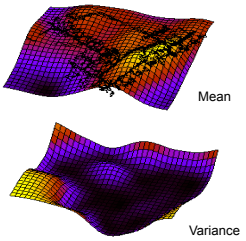


Hybrid Model for People Tracking

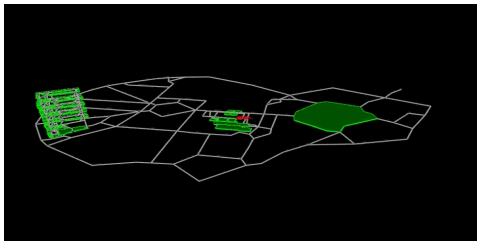


WiFi Sensor Model



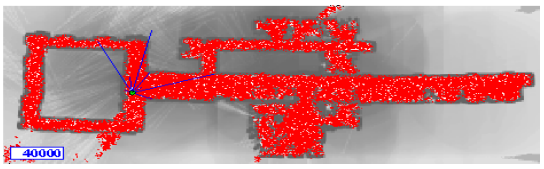


Tracking Example



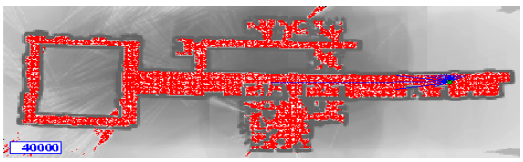
Adaptive Sampling

KLD-Sampling Sonar



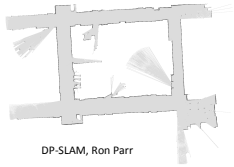
Adapt number of particles on the fly based on statistical approximation measure

KLD-Sampling Laser

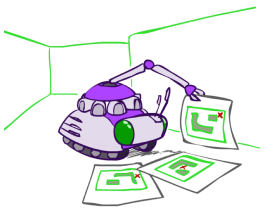


Robot Mapping

- SLAM: Simultaneous Localization And Mapping
  - We do not know the map or our location
  - State consists of position AND map!
  - Main techniques: Kalman filtering (Gaussian HMMs) and particle methods

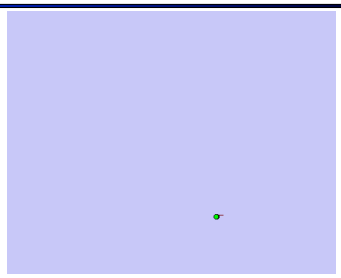


DP-SLAM, Ron Parr



[Demo: PARTICLES-SLAM-mapping1-new.avi]

Particle Filter SLAM – Video 2



[Demo: PARTICLES-SLAM-fastslam.avi]