Bayesian Filtering
for Robot Localization

CSE-473

Probabilistic Robotics

Key idea: Explicit representation of uncertainty

(using the calculus of probability theory)

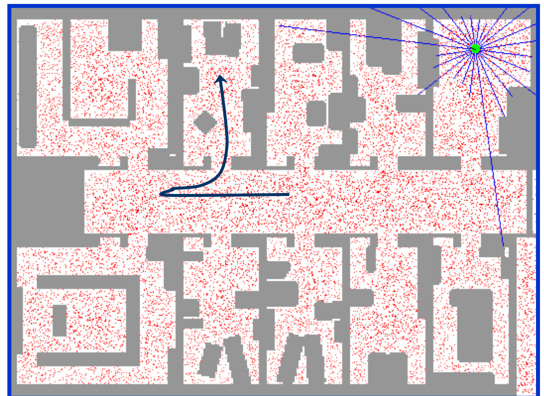
- Perception = state estimation
- Control = utility optimization

Localization

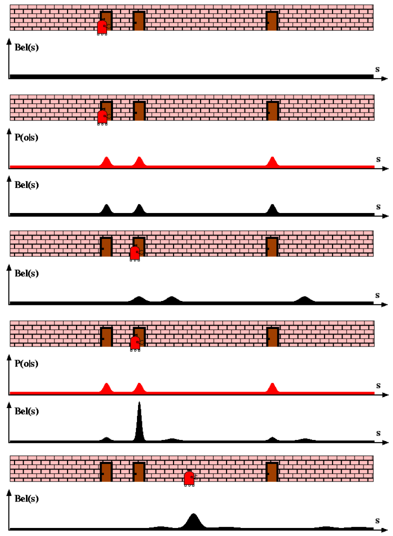
"Using sensory information to locate the robot in its environment is the most fundamental problem to providing a mobile robot with autonomous capabilities." [Cox '91]

- **Given**
  - Map of the environment.
  - Sequence of sensor measurements.
- **Wanted**
  - Estimate of the robot's position.
- **Problem classes**
  - Position tracking
  - Global localization
  - Kidnapped robot problem (recovery)

Sample-based Localization (sonar)



Bayes Filters for Robot Localization



Bayes Filters: Framework

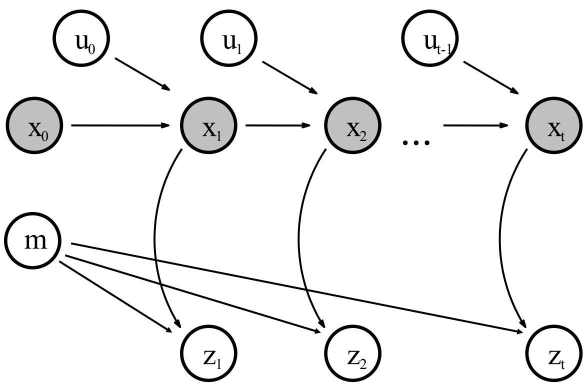
- **Given:**
  - Stream of observations *z* and action data *u:*

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

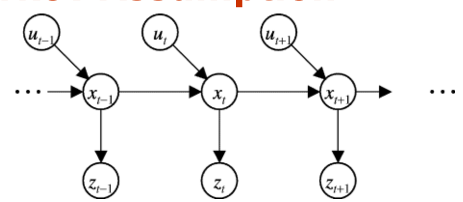
  - Sensor model *P(z|x).*
  - Action model *P(x|u,x').*
  - Prior probability of the system state *P(x).*
- **Wanted:**
  - Estimate of the state *X* of a dynamical system.
  - The posterior of the state is also called **Belief**:

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

Graphical Model of Localization



Markov Assumption



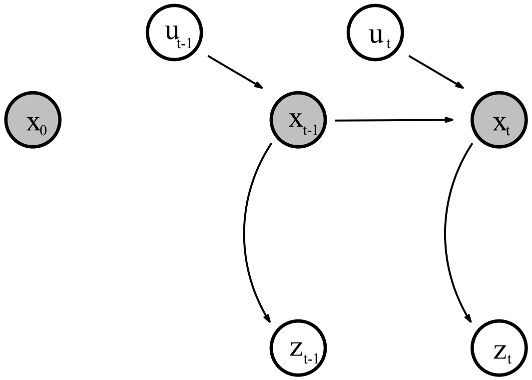
$$p(z_t \mid x_{0:t}, z_{1:t-1}, u_{1:t}) = p(z_t \mid x_t)$$

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

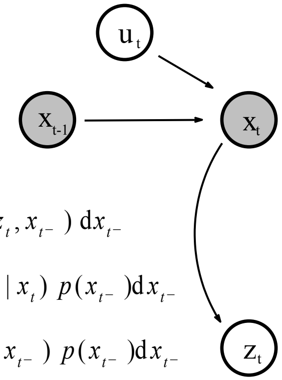
Underlying Assumptions

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

Two Time Slice Representation of Dynamic Bayes Net



Belief Update



$$p(x_t \mid u_t, z_t) = \alpha \int p(x_t, u_t, z_t, x_{t-1})\, dx_{t-1}$$

$$= \alpha \int p(x_t \mid u_t, x_{t-1})\, p(z_t \mid x_t)\; p(x_{t-1})\, dx_{t-1}$$

$$= \alpha \; p(z_t \mid x_t) \int p(x_t \mid u_t, x_{t-1})\; p(x_{t-1})\, dx_{t-1}$$

Bayes Filters

z = observation
u = action
x = state

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

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

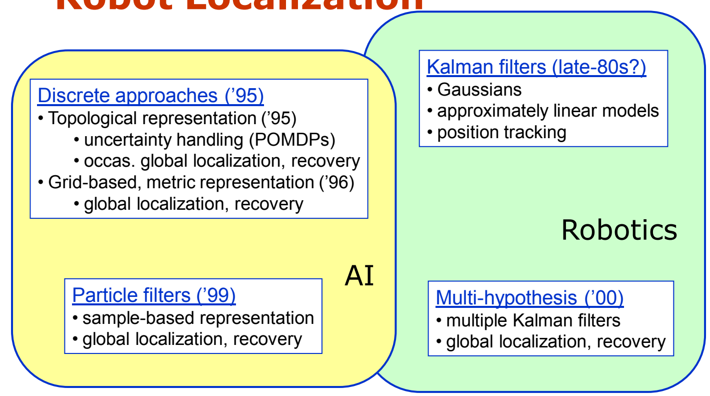
Markov $= \eta \; P(z_t \mid x_t)\; P(x_t \mid u_1, z_2, \ldots, u_{t-1})$

Total prob. $= \eta \; P(z_t \mid x_t) \int P(x_t \mid u_1, z_2, \ldots, u_{t-1}, x_{t-1})\; P(x_{t-1} \mid u_1, z_2, \ldots, u_{t-1})\, dx_{t-1}$

Markov $= \eta \; P(z_t \mid x_t) \int P(x_t \mid u_{t-1}, x_{t-1})\; P(x_{t-1} \mid u_1, z_2, \ldots, u_{t-1})\, dx_{t-1}$

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

Representations for Bayesian Robot Localization



Particle Filters

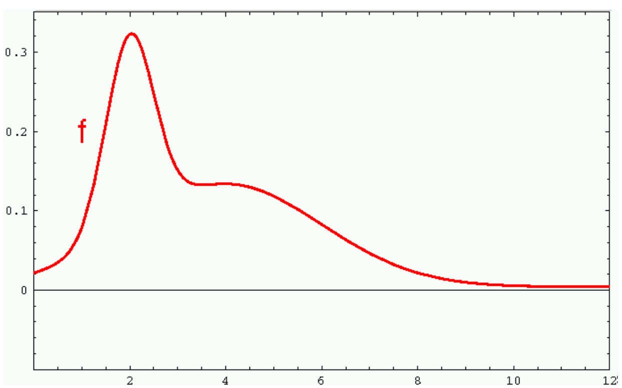
Particle Filters

- Represent belief by random samples
- Estimation of non-Gaussian, nonlinear processes

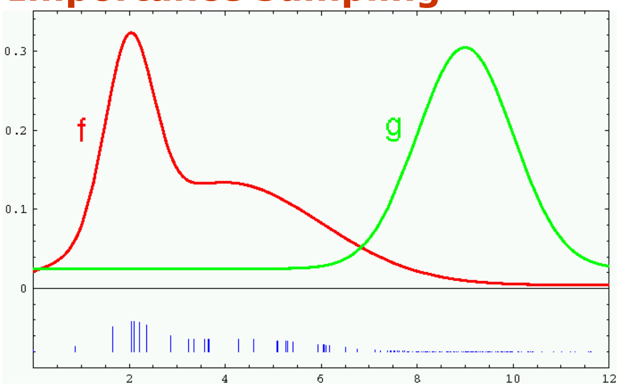
- Monte Carlo filter, Survival of the fittest, Condensation, Bootstrap filter, Particle filter

- Filtering: [Rubin, 88], [Gordon et al., 93], [Kitagawa 96]
- Computer vision: [Isard and Blake 96, 98]
- Dynamic Bayesian Networks: [Kanazawa et al., 95]d

Sample-based Density Representation

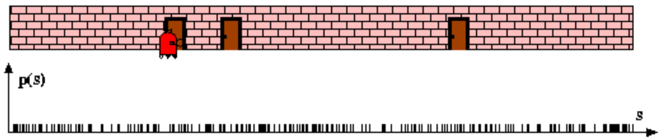


Importance Sampling



Weight samples: $w = f / g$

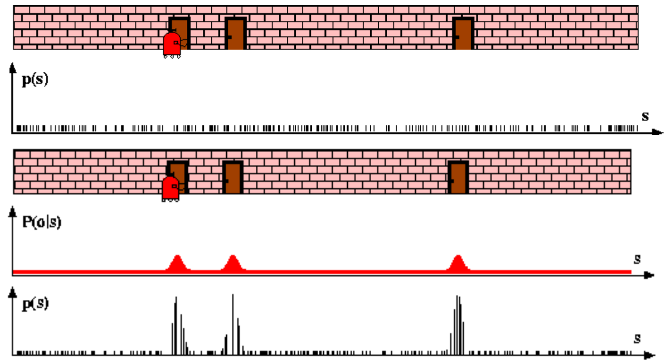
Particle Filters



Sensor Information: Importance Sampling

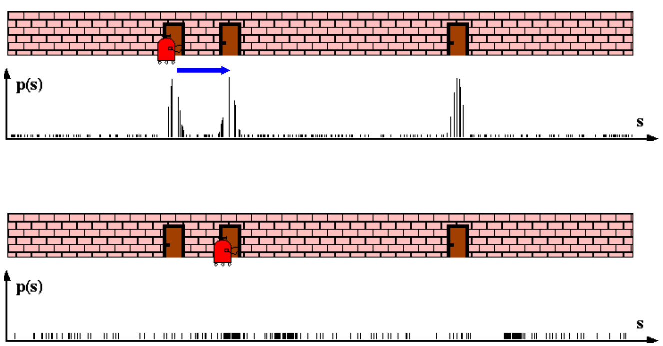
$$Bel(x) \leftarrow \alpha\, p(z \mid x)\, Bel^{-}(x)$$

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



Robot Motion

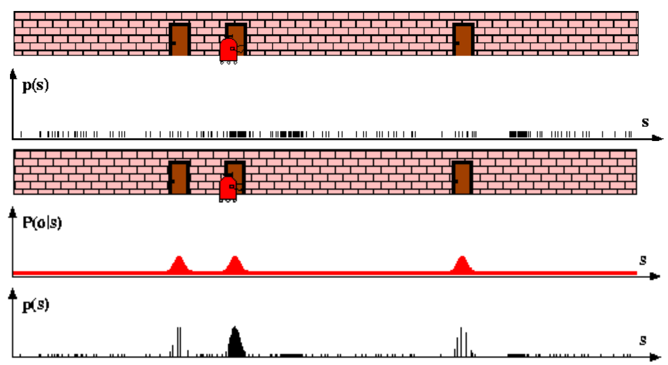
$$Bel^{-}(x) \leftarrow \int p(x \mid u, x')\, Bel(x')\, \mathrm{d}x'$$



Sensor Information: Importance Sampling

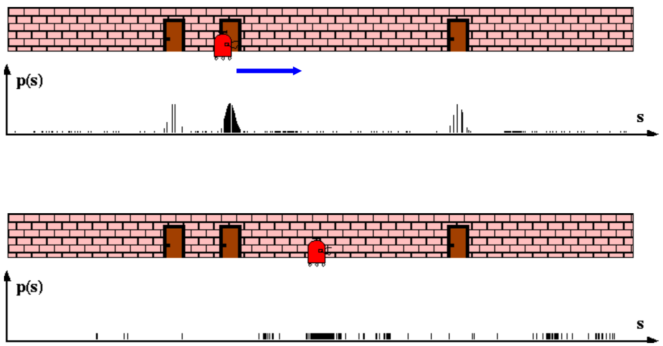
$$Bel(x) \leftarrow \alpha\, p(z \mid x)\, Bel^{-}(x)$$

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



Robot Motion

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



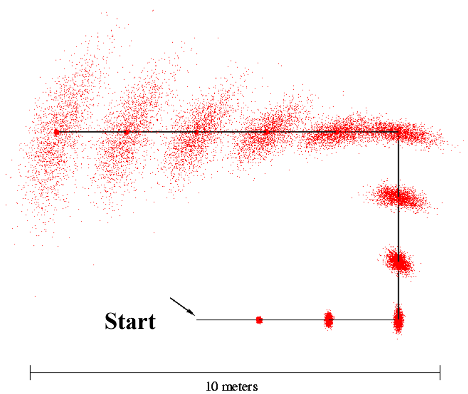
Particle Filter Algorithm

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

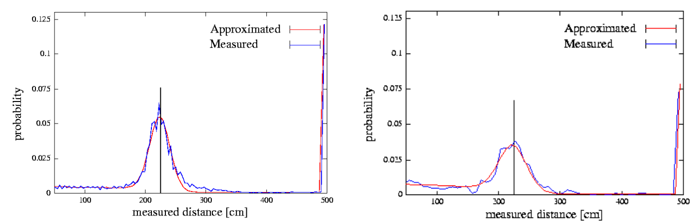
- draw x^i_{t-1} from $Bel(x_{t-1})$
- draw x^i_t from $p(x_t \mid x^i_{t-1}, u_{t-1})$
- Importance factor for x^i_t:

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

Motion Model Reminder

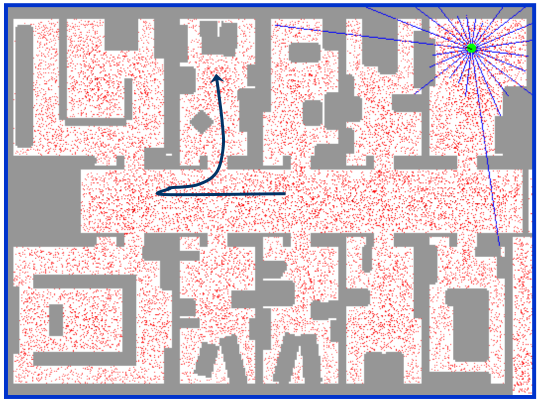


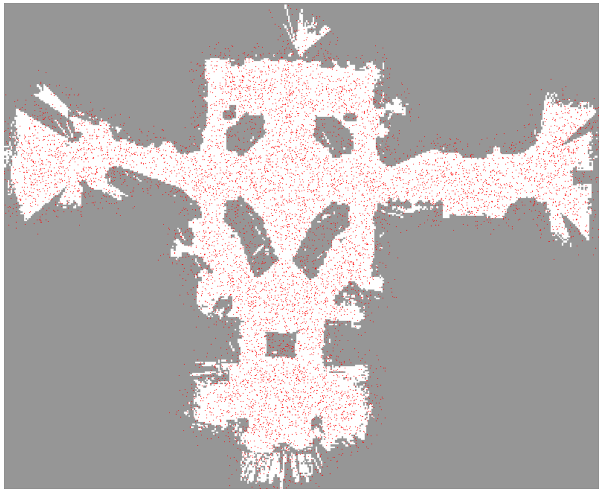
Proximity Sensor Model Reminder

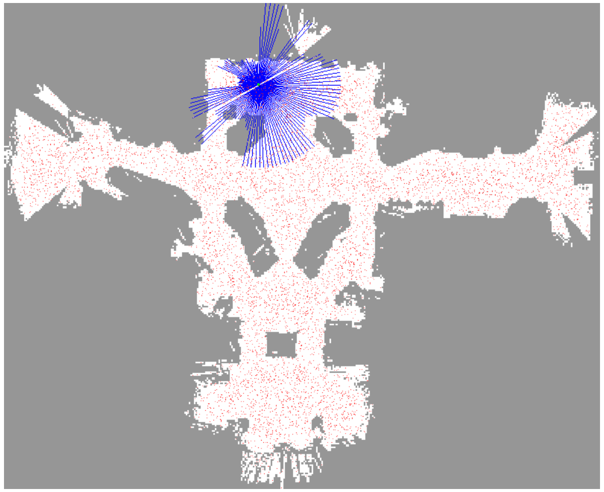


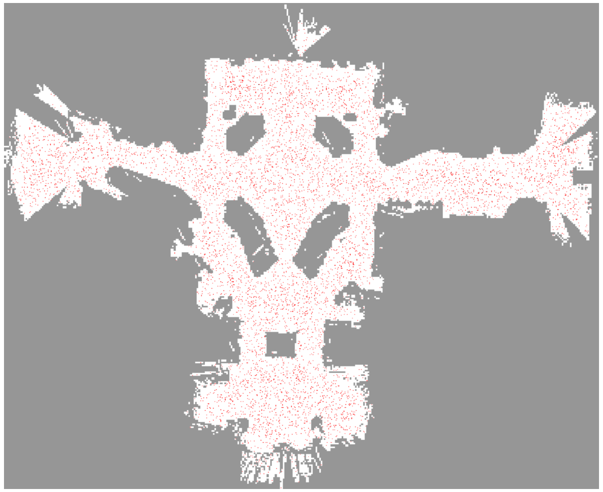
Laser sensor **Sonar sensor**

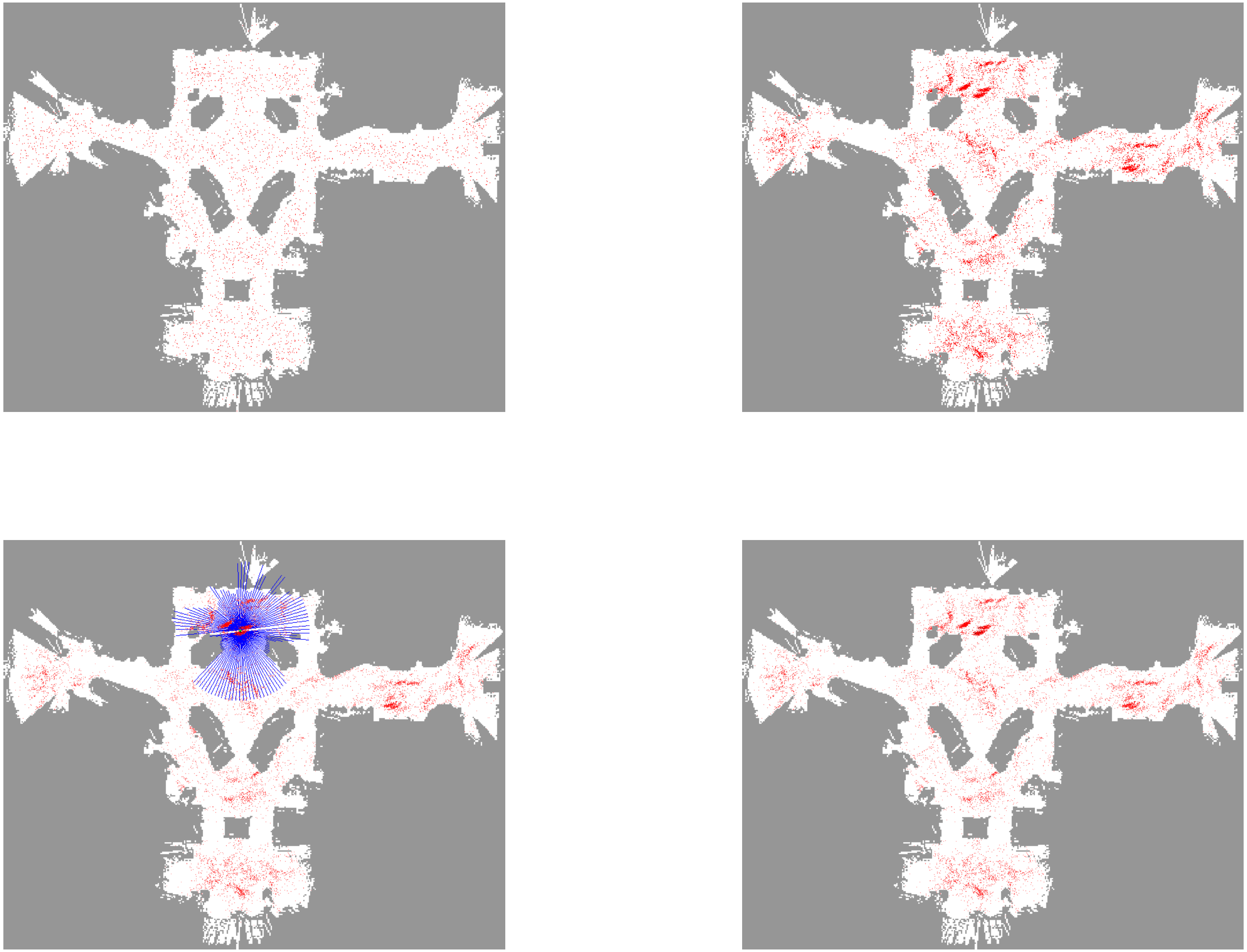
Sample-based Localization (sonar)



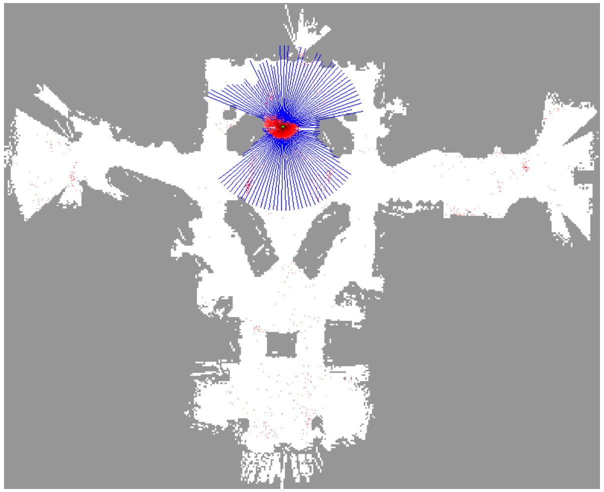


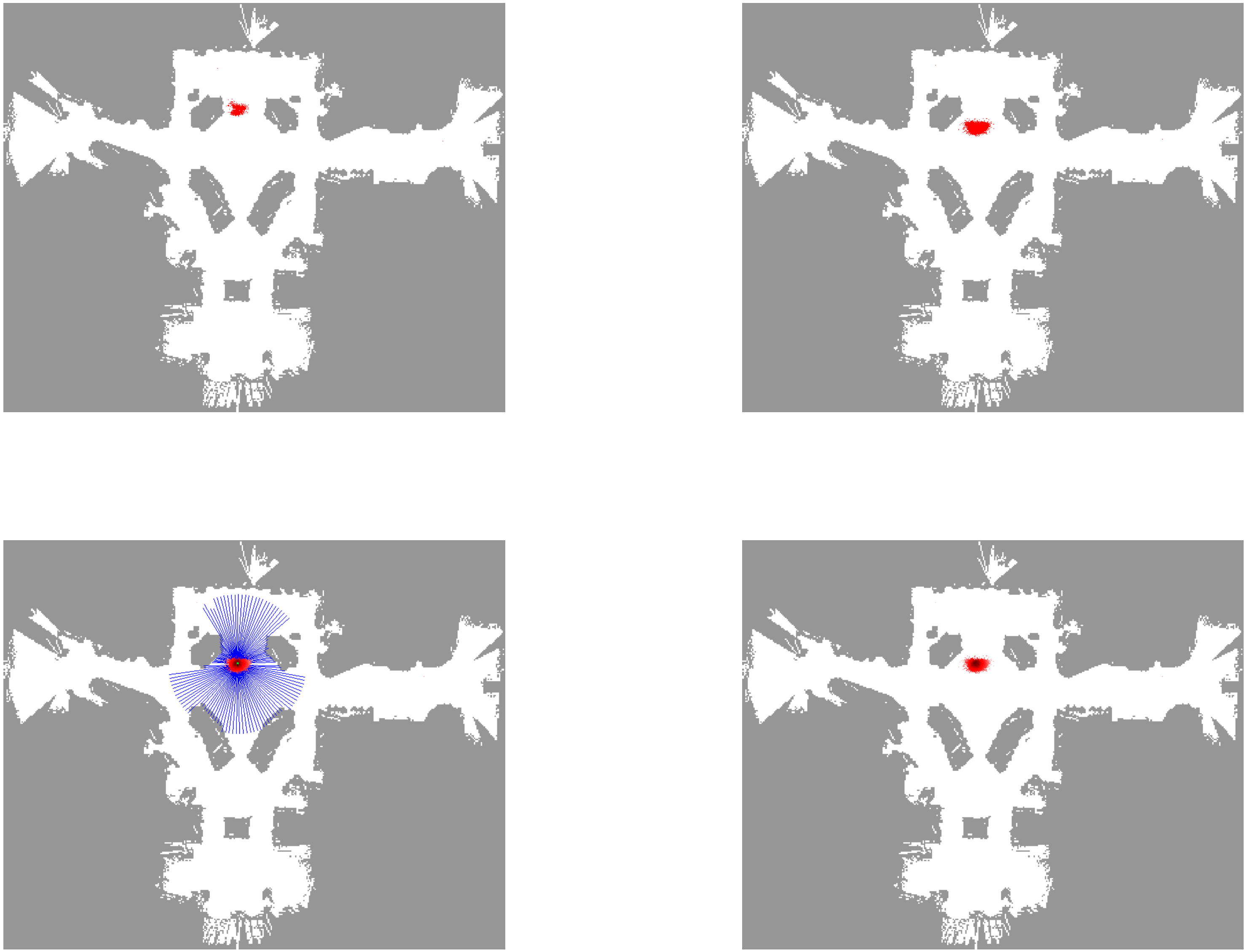


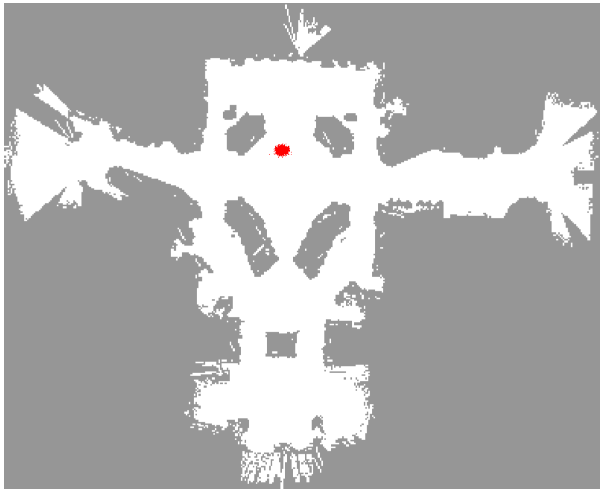
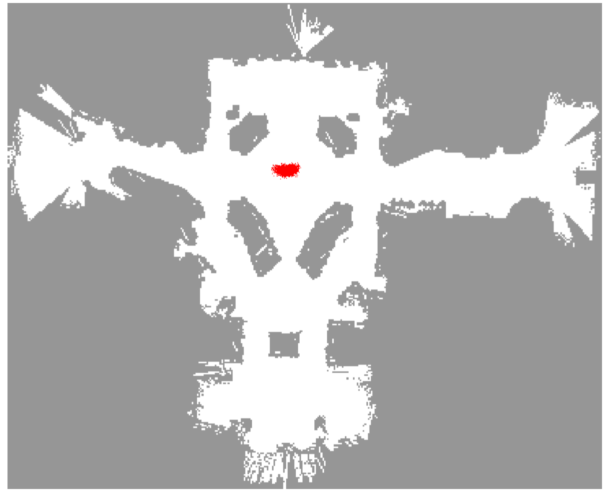
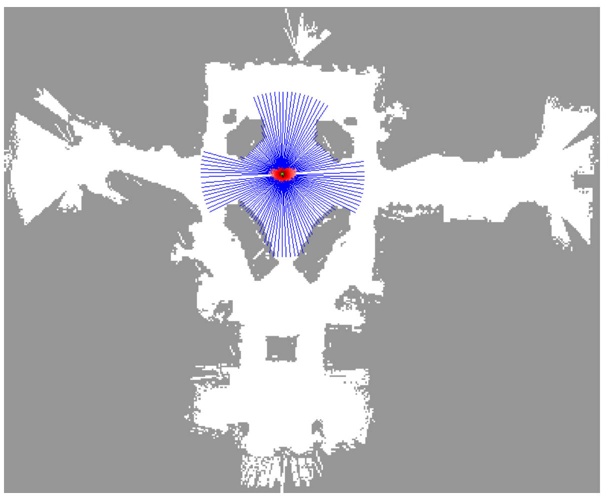
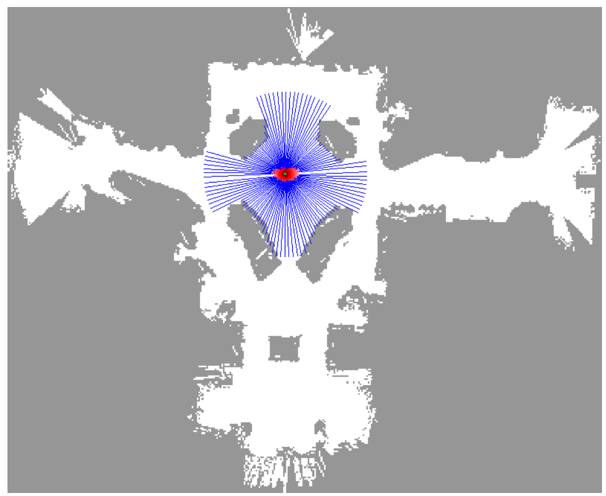




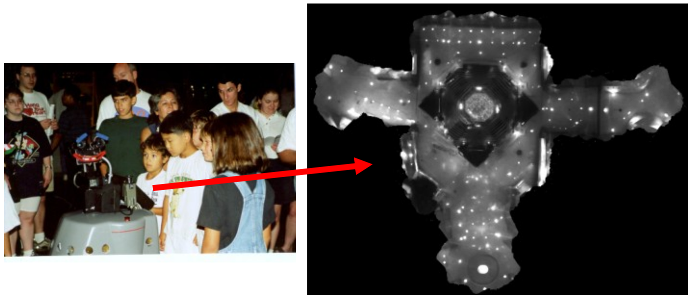




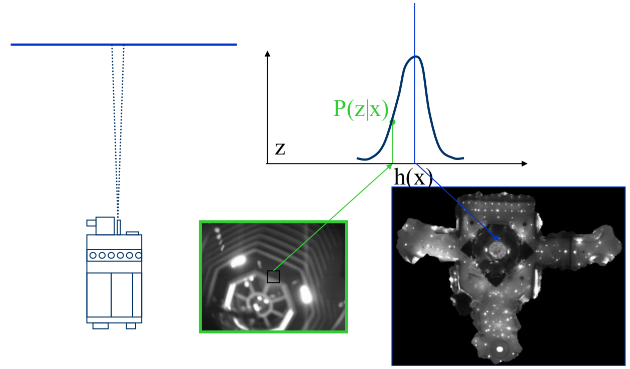





Using Ceiling Maps for Localization



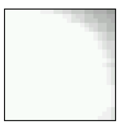
[Dellaert et al. 99]

Vision-based Localization

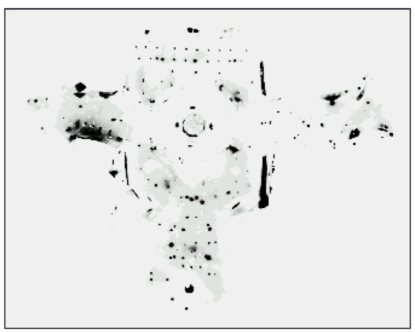


Under a Light

Measurement z:

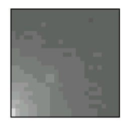


***P(z|x)*:**

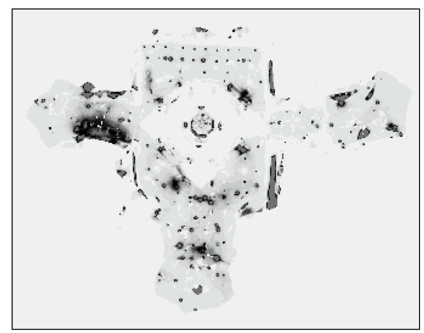


Next to a Light

Measurement z:

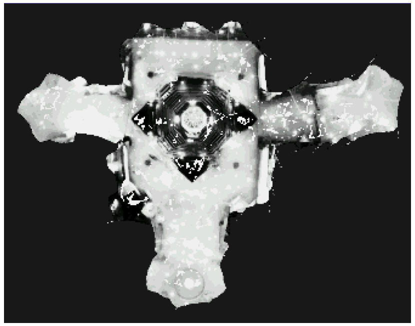


***P(z|x)*:**

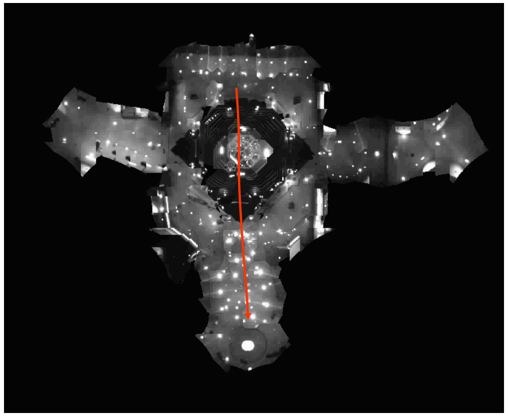


Elsewhere

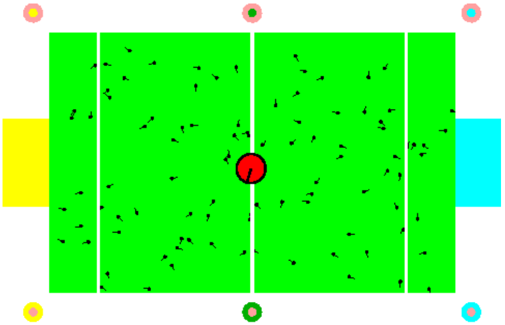
Measurement z: *P(z|x)*:



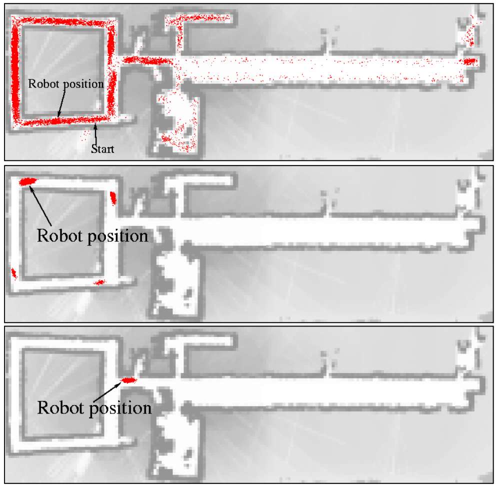
Global Localization Using Vision



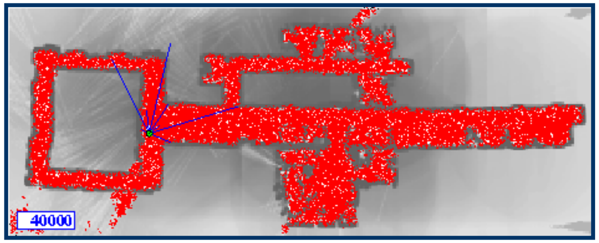
Localization for AIBO robots



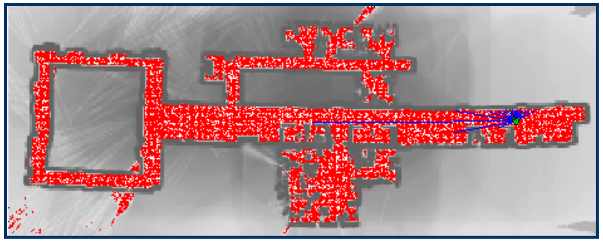
Adaptive Sampling



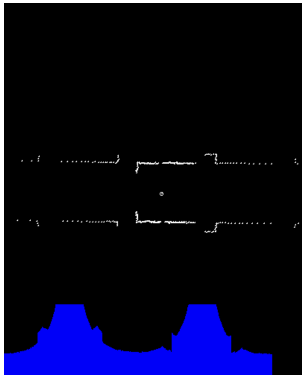
Example Run Sonar



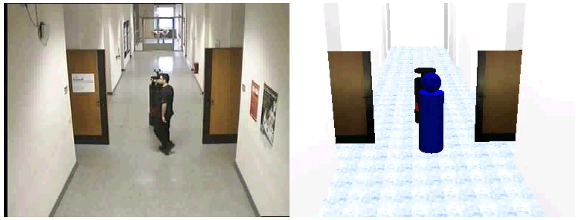
Example Run Laser



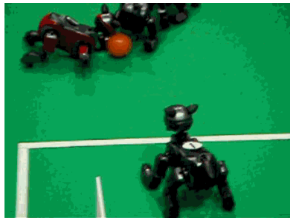
Example Run



Tracking with a Moving Robot



Ball Tracking in RoboCup

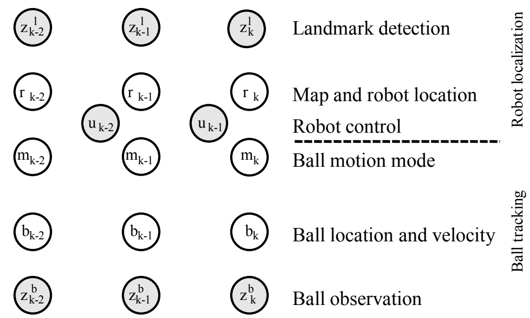


- Extremely noisy (nonlinear) motion of observer
- Inaccurate sensing, limited processing power
- Interactions between target and

Goal: Unified framework for modeling the ball and its interactions.

t

Dynamic Bayes Network for Ball Tracking



Rao-Blackwellised PF for Inference

- Represent posterior by random samples
- Each sample

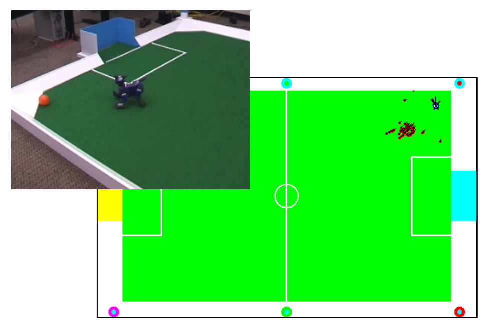
$$s_i = \langle r_i, m_i, b_i \rangle = \langle \langle x, y, \theta \rangle_i, m_i, \langle \mu, \Sigma \rangle_i \rangle$$

contains robot location, ball mode, ball Kalman filter

- Generate individual components of a particle stepwise using the factorization

$$p(b_k, m_{1:k}, r_{1:k} \mid z_{1:k}, u_{1:k-1}) =$$
$$p(b_k \mid m_{1:k}, r_{1:k}, z_{1:k}, u_{1:k-1})\, p(m_{1:k} \mid r_{1:k}, z_{1:k}, u_{1:k-1}) \cdot p(r_{1:k} \mid z_{1:k}, u_{1:k-1})$$

Ball-Environment Interaction

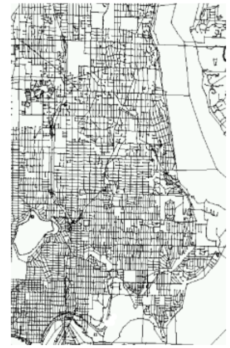


GPS Receivers We Used

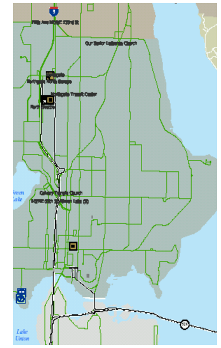
GeoStats wearable
GPS logger

Nokia 6600 Java Cell
Phone with Bluetooth
GPS unit

Geographic Information Systems

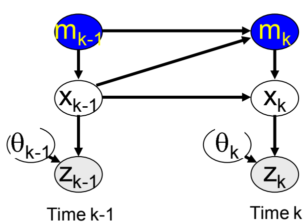


Street map
Data source: Census 2000
Tiger/line data



Bus routes and bus stops
Data source: Metro GIS

Adding Mode of Transportation



Transportation mode

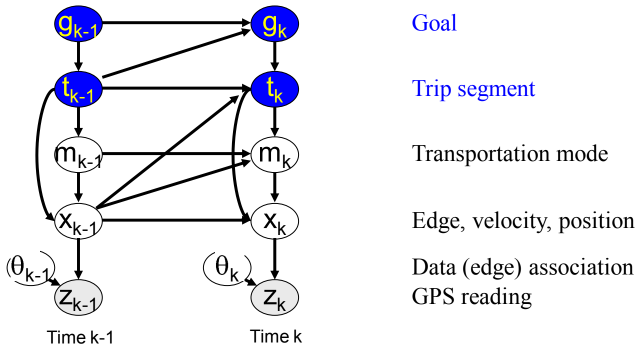
Edge, velocity, position

Data (edge) association
GPS reading

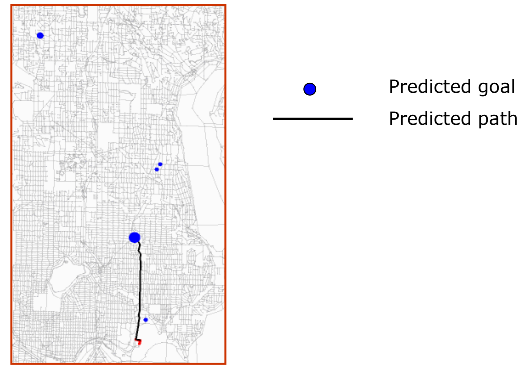
Infer Location and Transportation

■	Measurements
●	Projections
Green	Bus mode
Red	Car mode
Blue	Foot mode

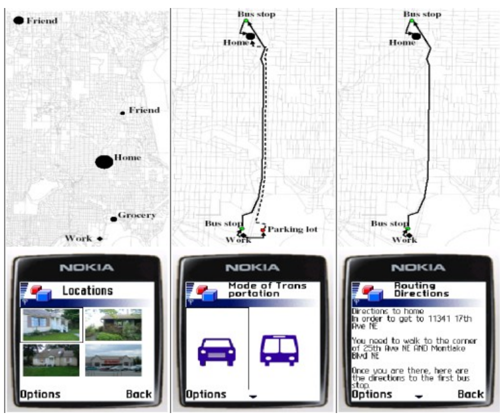
Hierarchical Model



Predict Goal and Path



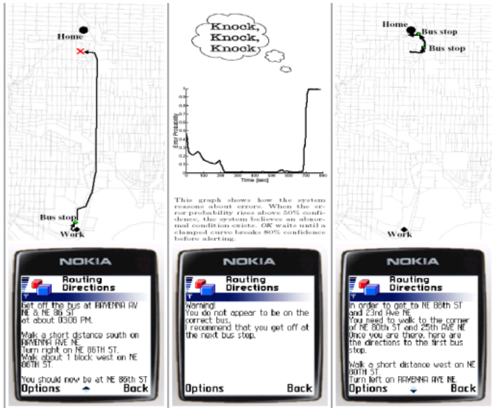
Application: Opportunity Knocks



Detect User Errors

Untrained Trained Instantiated

Application: Opportunity Knocks



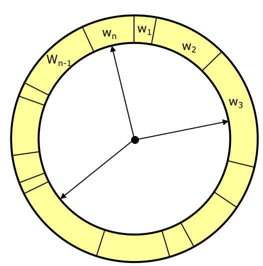
Particle Filter Algorithm

Algorithm Particle_filter($\mathcal{X}_{t-1}, u_t, z_t$):
- $\bar{\mathcal{X}}_t = \mathcal{X}_t = \emptyset$
- for $m = 1$ to M do
  - sample $x_t^{[m]} \sim p(x_t \mid u_t, x_{t-1}^{[m]})$
  - $w_t^{[m]} = p(z_t \mid x_t^{[m]})$
  - $\bar{\mathcal{X}}_t = \bar{\mathcal{X}}_t + \langle x_t^{[m]}, w_t^{[m]} \rangle$
- endfor
- for $m = 1$ to M do
  - draw i with probability $\propto w_t^{[i]}$
  - add $x_t^{[i]}$ to $\mathcal{X}_t$
- endfor
- return $\mathcal{X}_t$

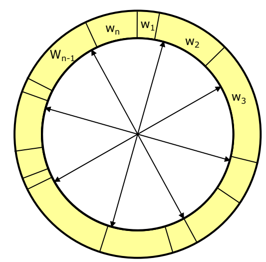
Resampling

- **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, log n



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

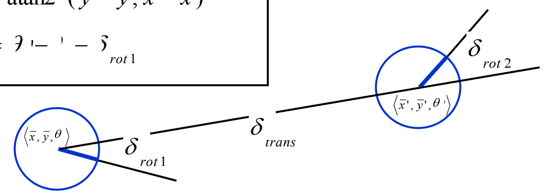
Resampling Algorithm

```
Algorithm Low_variance_sampler(X_t, W_t):
    X̄_t = ∅
    r = rand(0; M^{-1})
    c = w_t^{[1]}
    i = 1
    for m = 1 to M do
        U = r + (m − 1) · M^{-1}
        while U > c
            i = i + 1
            c = c + w_t^{[i]}
        endwhile
        add x_t^{[i]} to X̄_t
    endfor
    return X̄_t
```

Probabilistic Kinematics

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

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



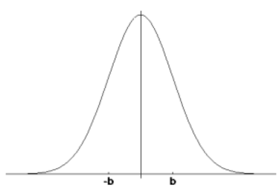
Noise Model for Motion

- The measured motion is given by the true motion corrupted with noise.
- To predict sample position, just sample from "noisy version" of measured motion.

$$\delta_{rot1} = \delta_{rot1} + \varepsilon_{\alpha_1 |\delta_{rot1}| + \alpha_2 |\delta_{trans}|}$$
$$\delta_{trans} = \delta_{trans} + \varepsilon_{\alpha_3 |\delta_{trans}| + \alpha_4 |\delta_{rot1} + \delta_{rot2}|}$$
$$\delta_{rot2} = \delta_{rot2} + \varepsilon_{\alpha_1 |\delta_{rot2}| + \alpha_2 |\delta_{trans}|}$$

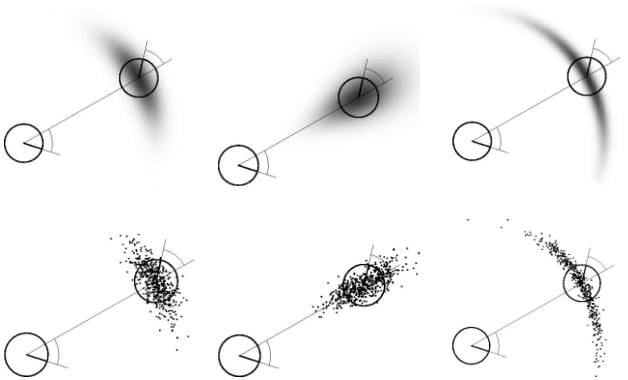
Gaussian Noise Model

Normal distribution

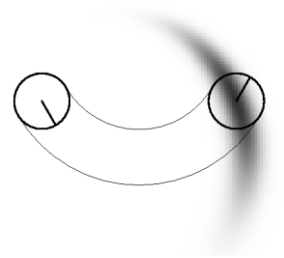


$$\varepsilon_{\sigma^2}(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2}\frac{x^2}{\sigma^2}}$$

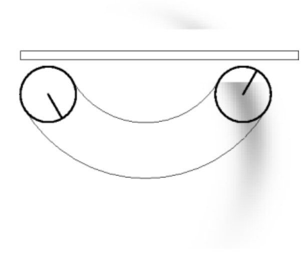
Examples (odometry based)



Motion Model with Map



$P(x \mid u, x')$



$P(x \mid u, x', m) = \eta\, P(x \mid m)\, P(x \mid u, x')$

- When does this approximation fail?