Particle Filters

Stefan Contiu

- UTCN -

April 2015

Particle Filters

- Basic Algorithm
- Importance Sampling
- Mathematical Derivation of the PF
- Low Variance Resampling
- Robot Localization Example
- Conclusion

Particle Fitlers

- Alternative non-parametric implementation of Bayes filter.
- Key Idea : represent the posterior $bel(x_t)$ by a set of random state samples drawn from the posterior.

Call the samples : **Particles**, denote by:

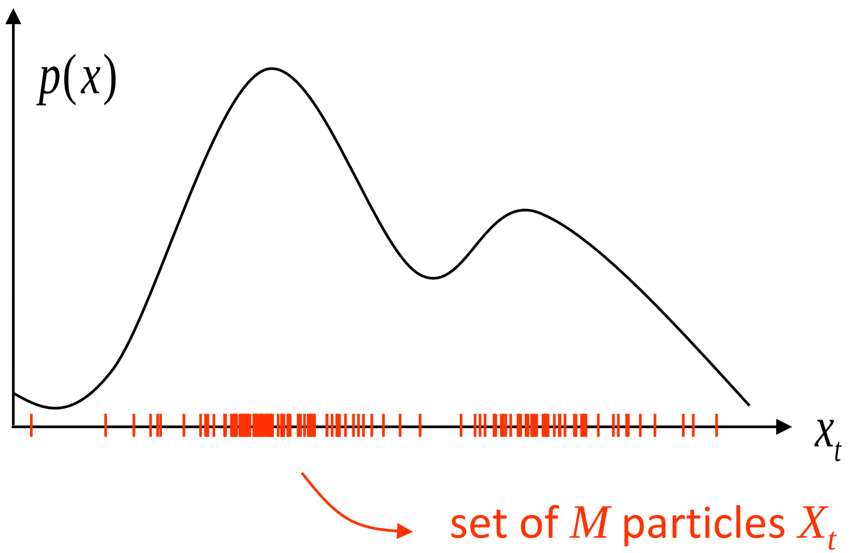
$$X_t = {x_t}^{[1]}, {x_t}^{[2]}, \dots, {x_t}^{[M]}$$

- each particle is a hypothesis as to what the true state may be at time t.

The Particles

- Ideally, the likelihood for a state hypothesis x_t to be included in the particle set X_t shall be proportional to its Bayes filter posterior $bel(x_t)$:

$$x_t^{[m]} \sim p(x_t \mid z_{1:t}, u_{1:t})$$



$$p(x_t \in X_t) \approx p(x_t \mid z_{1 \ldots t}) \quad \text{(equality for)} \quad M \uparrow \infty$$

Basic Algorithm

```
Algorithm Particle_fitler(X_{t-1}, u_t, z_t):
    \bar{X_t} = X_t = ∅
    for m = 1 to M do
        sample x_t^[m] ~ p(x_t | u_t, x_{t-1}^[m])
        w_t^[m] = p(z_t | x_t^[m])
        \bar{X_t} = \bar{X_t} + ⟨x_t^[m], w_t^[m]⟩
    endfor
    for m = 1 to M do
        draw i with probability ∝ w_t^[i]
        add x_t^[i] to X_t
    endfor
    return X_t
```

Algorithm Particle_fitler(X_{t-1}, u_t, z_t):

$\overline{X_t} = X_t = \emptyset$

for $m = 1$ to M do

  sample $x_t{}^{[m]} \sim p(x_t \mid u_t, x_{t-1}{}^{[m]})$

  $w_t{}^{[m]} = p(z_t \mid x_t{}^{[m]})$

  $\overline{X_t} = \overline{X_t} + \langle x_t{}^{[m]}, w_t{}^{[m]} \rangle$

endfor

for $m = 1$ to M do

  draw i with probability $\propto w_t{}^{[i]}$

  add $x_t{}^{[i]}$ to X_t

endfor

return X_t

Basic Algorithm

Algorithm Particle_fitler(X_{t-1}, u_t, z_t):

$\overline{X_t} = X_t = \emptyset$

for $m = 1$ to M do

sample $x_t^{[m]} \sim p(x_t \mid u_t, x_{t-1}^{[m]})$

$w_t^{[m]} = p(z_t \mid x_t^{[m]})$

$\overline{X_t} = \overline{X_t} + \langle x_t^{[m]}, w_t^{[m]} \rangle$

endfor

for $m = 1$ to M do

draw i with probability $\propto w_t^{[i]}$

add $x_t^{[i]}$ to X_t

endfor

return X_t

Sequential importance sampling

Construct a temporary particle set $\overline{X_t}$ reminiscent to the belief $\overline{bel}(x_t)$

Basic Algorithm

Algorithm Particle_fitler(X_{t-1}, u_t, z_t):

$\overline{X_t} = X_t = \emptyset$

for $m = 1$ to M do

sample $x_t^{[m]} \sim p(x_t \mid u_t, x_{t-1}^{[m]})$

$w_t^{[m]} = p(z_t \mid x_t^{[m]})$

$\overline{X_t} = \overline{X_t} + \langle x_t^{[m]}, w_t^{[m]} \rangle$

endfor

for $m = 1$ to M do

draw i with probability $\propto w_t^{[i]}$

add $x_t^{[i]}$ to X_t

endfor

return X_t

- Generates a hypothetical state x_t for time t based on particle x_{t-1} and u_t
- We need to sample from $p(x_t \mid u_t, x_{t-1})$. The ability of sample from the state transition probability is not given for arbitrary distributions. However many major distributions posses efficient algorithms for generating samples.
- Set of sampled particles is the representation of $\overline{bel}(x_t)$

Basic Algorithm

: **Algorithm Particle_fitler**(X_{t-1}, u_t, z_t):

    $\overline{X_t} = X_t = \emptyset$

    for $m = 1$ to M do

        sample $x_t{}^{[m]} \sim p(x_t \mid u_t, x_{t-1}{}^{[m]})$

        $w_t{}^{[m]} = p(z_t \mid x_t{}^{[m]})$

        $\overline{X_t} = \overline{X_t} + \langle x_t{}^{[m]}, w_t{}^{[m]} \rangle$

    endfor

    for $m = 1$ to M do

        draw i with probability $\propto w_t{}^{[i]}$

        add $x_t{}^{[i]}$ to X_t

    endfor

    return X_t

- Calculates for each particle $x_t{}^{[m]}$ the importance factor.
- Incorporates measurment z_t into the particle set.
- Set of weighted particles represents (in aprox) $bel(x_t)$

Basic Algorithm

Algorithm Particle_fitler(X_{t-1}, u_t, z_t):

$\overline{X_t} = X_t = \emptyset$

for $m = 1$ to M do

sample $x_t^{[m]} \sim p(x_t \mid u_t, x_{t-1}^{[m]})$

$w_t^{[m]} = p(z_t \mid x_t^{[m]})$

$\overline{X_t} = \overline{X_t} + \langle x_t^{[m]}, w_t^{[m]} \rangle$

endfor

for $m = 1$ to M do

draw i with probability $\propto w_t^{[i]}$

add $x_t^{[i]}$ to X_t

endfor

return X_t

Add the particle and its weight to the temporary particle set.

Basic Algorithm

Algorithm Particle_fitler(X_{t-1}, u_t, z_t):

$\overline{X_t} = X_t = \emptyset$

for $m = 1$ to M do

  sample $x_t^{[m]} \sim p(x_t \mid u_t, x_{t-1}^{[m]})$

  $w_t^{[m]} = p(z_t \mid x_t^{[m]})$

  $\overline{X_t} = \overline{X_t} + \langle x_t^{[m]}, w_t^{[m]} \rangle$

endfor

for $m = 1$ to M do

  draw i with probability $\propto w_t^{[i]}$

  add $x_t^{[i]}$ to X_t

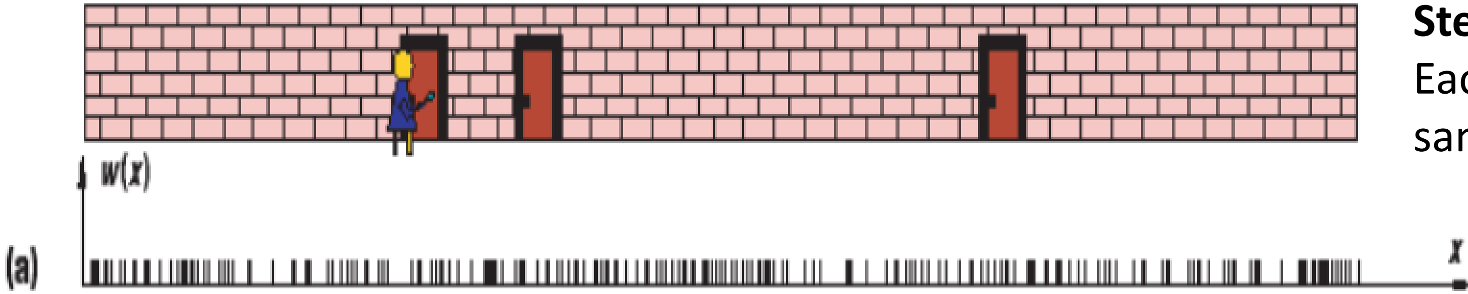
endfor

return X_t

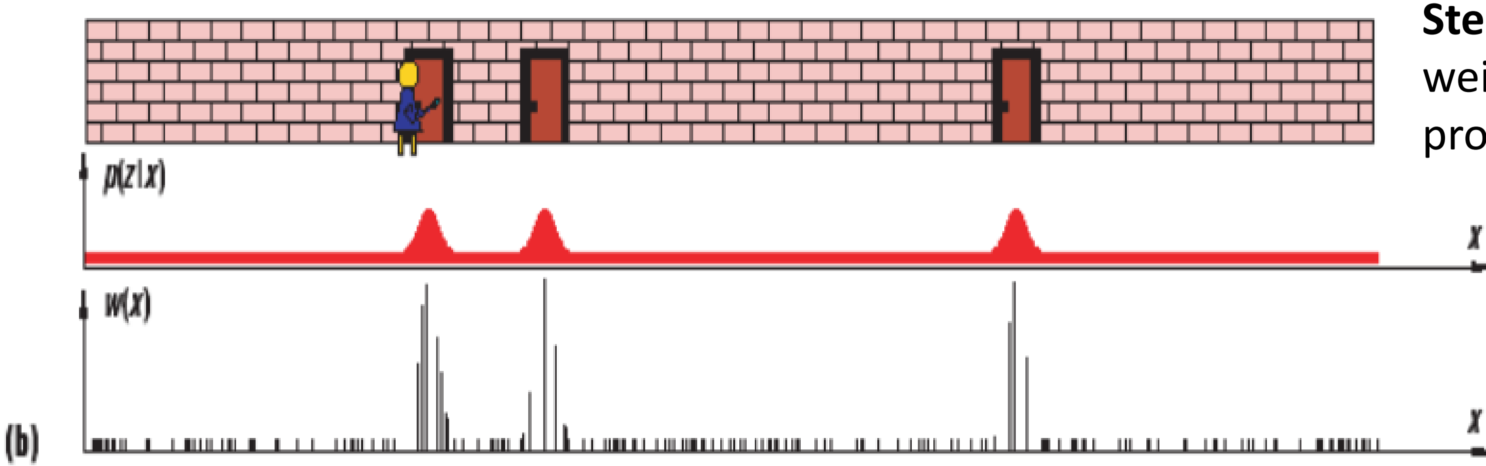
Resampling:

- Eliminate particles with small importance weights
- Concentrate on particles with large weights

PF Algorithm Example

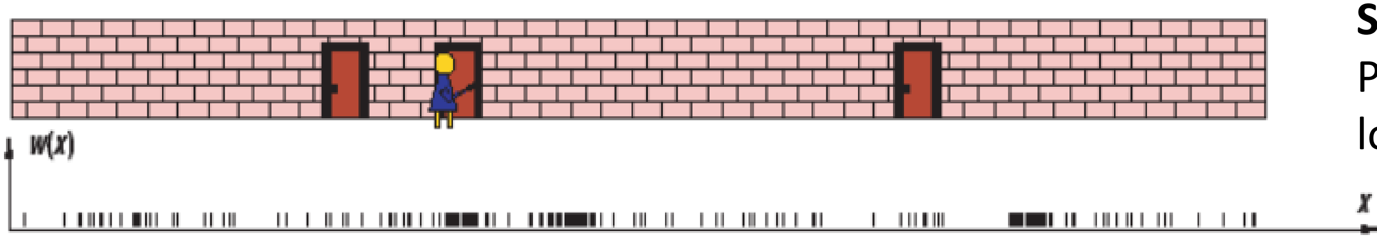


Step 0 : initialization
Each particle has the same weight



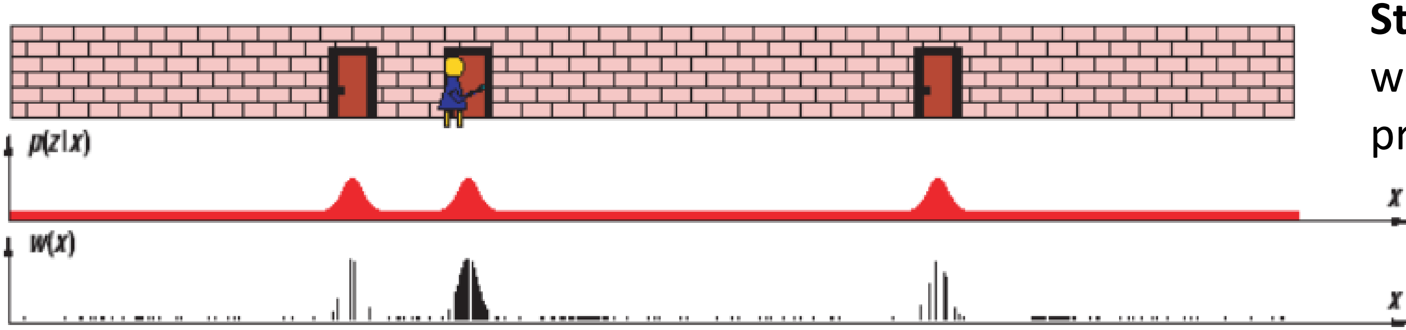
Step 1 : updating weights. Weights are proportional to p(z|x)

PF Algorithm Example



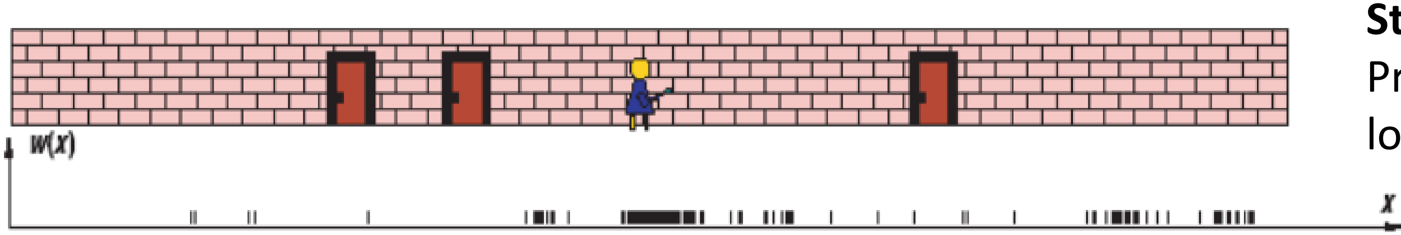
(c)

Step 2 : predicting
Predict the new locations of particles.



(d)

Step 3 : updating weights. Weights are proportional to p(z|x)



(e)

Step 4 : predicting
Predict the new locations of particles.

Importance Sampling

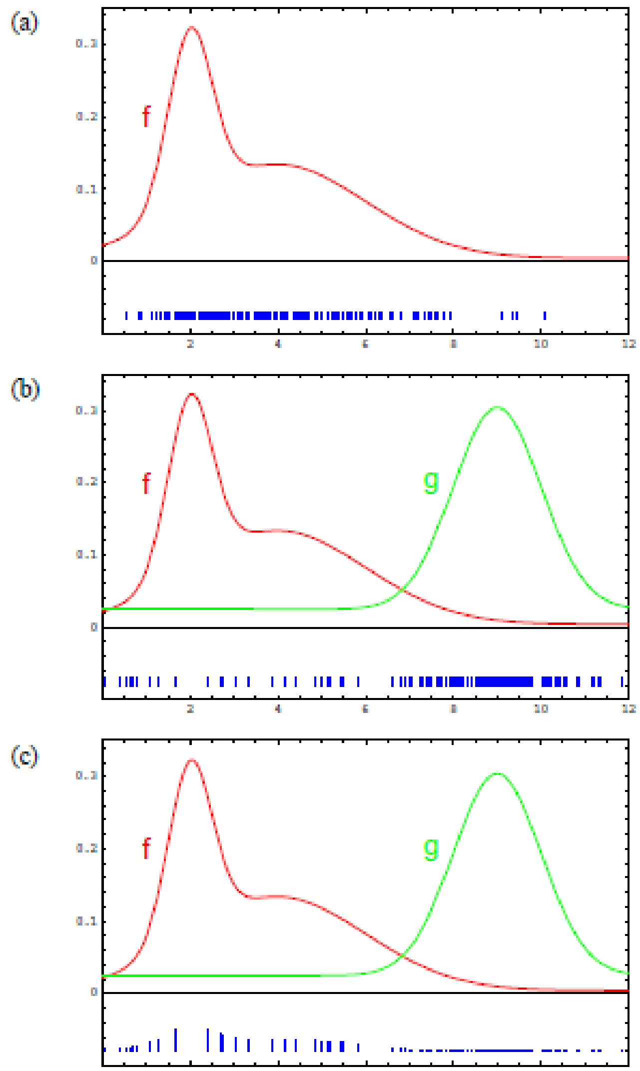
- Unfortunately it is often not possible to sample directly from the posterior distribution (f).
- We can use a different distribution g to generate samples from f.
- By introducing the importance weight w, we can account for the "differences between g and f".
- $w = f/g$
- f is often called target
- g is often called proposal

Importance Sampling

Samples cannot be drawn conveniently from the target distribution f.

Instead, the importance sampler draws samples from the proposal distribution g, which has a simpler form.

A sample of f is obtained by attaching the weight f/g to each sample x.



In particle filters:

$$f = bel(x_t)$$
$$g = \overline{bel}(x_t)$$

Mathematical Derivation of PF (1)

- Think of particles as samples of state sequences:

$$x_{0:t}^{[m]} = x_0^{[m]}, x_1^{[m]}, \ldots, x_t^{[m]}$$

- Modify the algorithm:
  - Append to the particle ${x_t}^{[m]}$ the sequence of state samples from which it was generated ${x_{0:t-1}}^{[m]}$
  - This algorithm computes the posterior over all state sequences:

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

instead of the belief

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

Mathematical Derivation of PF (2)

- The posterior $bel(x_{0:\,t})$ is obtained in same way as the derivation of $bel(x_t)$ of Bayes Filters.
- First, we apply Bayes rule to the target posterior:

$$p(x_{0:t} \mid z_{1:t}, u_{1:t}) \overset{\text{Bayes}}{=} \frac{p(z_t \mid x_{0:t}, z_{1:t-1}, u_{1:t})\, p(x_{0:t} \mid z_{1:t-1}, u_{1:t})}{p(z_t \mid z_{1:t-1}, u_{1:t})}$$

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

- Due to Conditional Independence:

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

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

$$\overset{\text{Markov}}{=} \quad \eta\, \underbrace{p(z_t \mid x_t)}_{\text{prediction}}\, \underbrace{p(x_t \mid x_{t-1}, u_t)\, p(x_{0:t-1} \mid z_{1:t-1}, u_{1:t-1})}_{\text{correction}}$$

Mathematical Derivation of PF (3)

- Carry out derivation by induction.
- Initial Condition verification.
- For the mth particle:

$$\begin{aligned} & p(x_t \mid x_{t-1}, u_t)\, bel(x_{0:t-1}) \\ = \quad & p(x_t \mid x_{t-1}, u_t)\, p(x_{0:t-1} \mid z_{0:t-1}, u_{0:t-1}) \end{aligned}$$

With

$$\begin{aligned} w_t^{[m]} &= \frac{\text{target distribution}}{\text{proposal distribution}} \\ &= \frac{\eta\, p(z_t \mid x_t)\, p(x_t \mid x_{t-1}, u_t)\, p(x_{0:t-1} \mid z_{1:t-1}, u_{1:t-1})}{p(x_t \mid x_{t-1}, u_t)\, p(x_{0:t-1} \mid z_{0:t-1}, u_{0:t-1})} \\ &= \eta\, p(z_t \mid x_t) \end{aligned}$$

Mathematical Derivation of PF (4)

- By resampling particles with probability proportional to wtm:

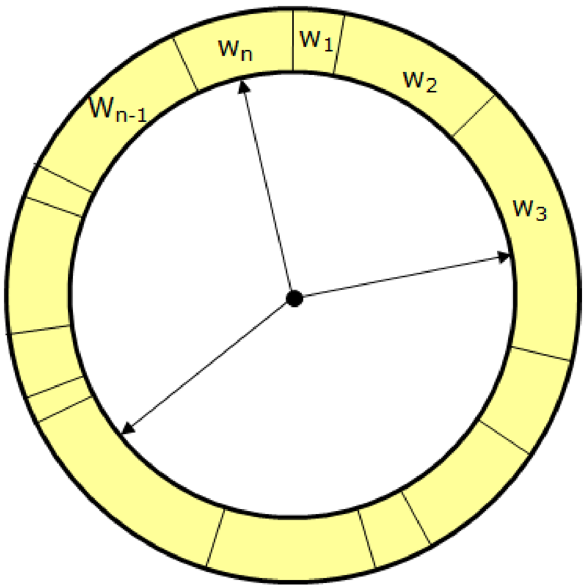
$$\eta \, w_t^{[m]} \, p(x_t \mid x_{t-1}, u_t) \, p(x_{0:t-1} \mid z_{0:t-1}, u_{0:t-1}) \quad = \quad bel(x_{0:t})$$

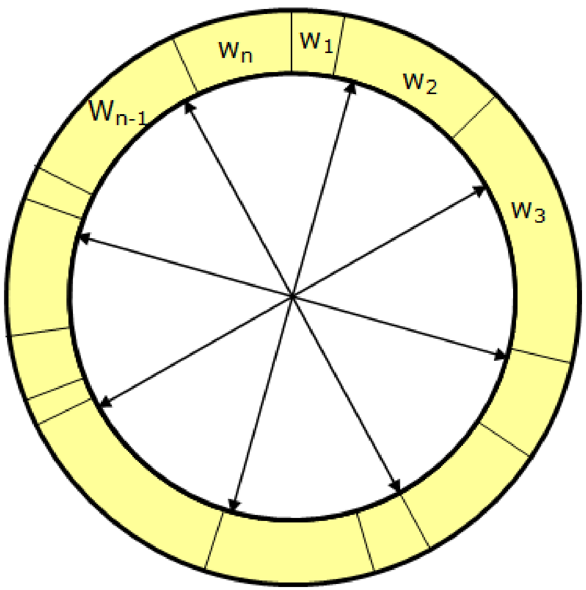
- The resulting particles are distributed according to the product of the proposal and the importance weight.
- This derivation is correct only for M -> inf. However, even for finite M it explains the intuition behind PF.

Resampling

Given : Set X of wieghted samples.

Wanted : Random sample, where the probability of drawing x_i is given by w_i.





Roulette wheel
Generating n random values
O(n lg n)

Stochastic universal sampling
Generating one random value
O(n)
Easy to implement, low variance

Low Variance Resampling

Algorithm Low_variance_sampler($\mathcal{X}_t, \mathcal{W}_t$):

$\bar{\mathcal{X}}_t = \emptyset$

$r = \text{rand}(0; M^{-1})$

$c = w_t^{[1]}$

$i = 1$

for $m = 1$ to M do

  $u = r + (m-1) \cdot M^{-1}$

  while $u > c$

    $i = i + 1$

    $c = c + w_t^{[i]}$

  endwhile

  add $x_t^{[i]}$ to $\bar{\mathcal{X}}_t$

endfor

return $\bar{\mathcal{X}}_t$

Draw only once one random number *r* in (0, M^{-1})

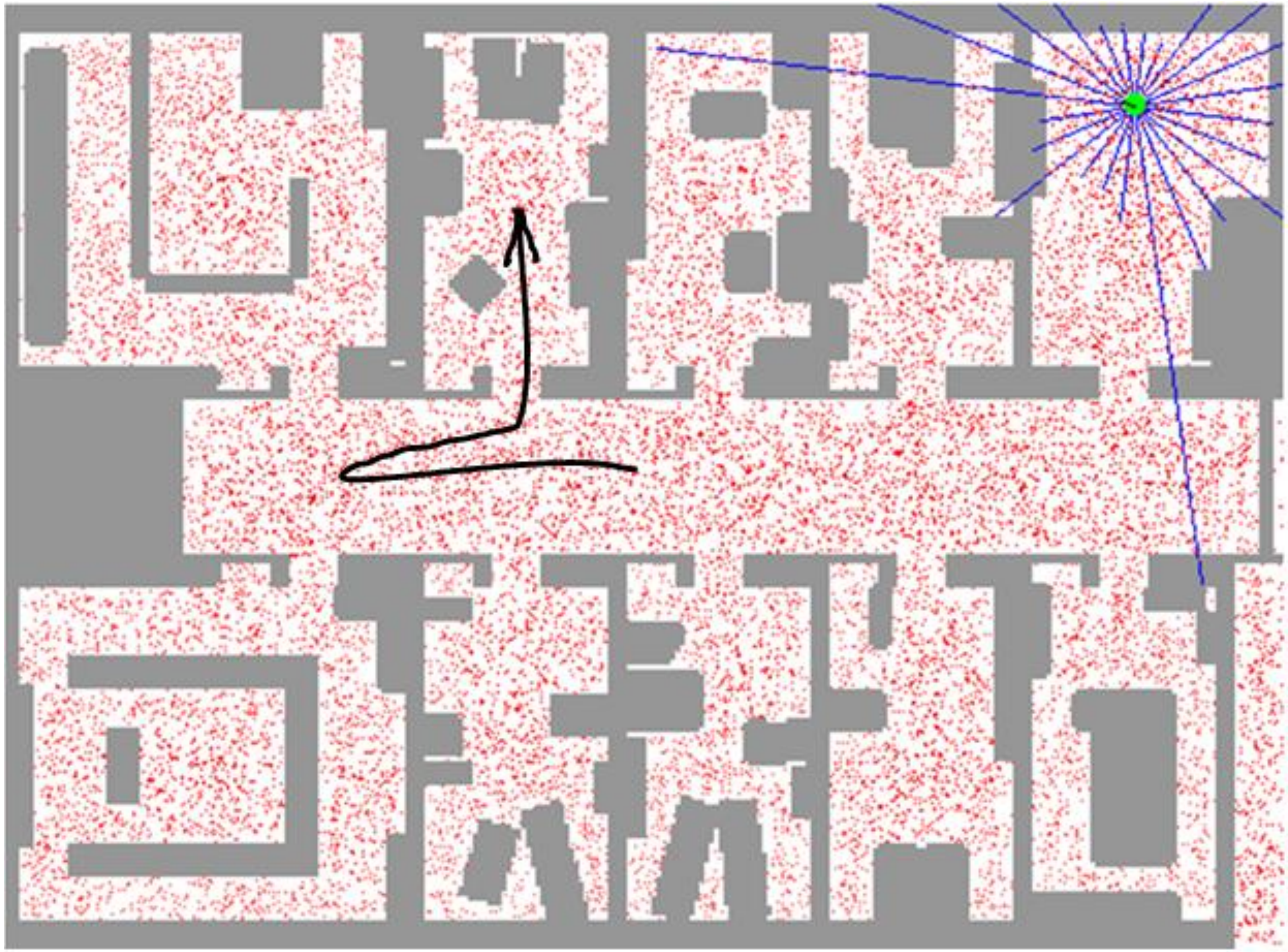
Add the fixed amount of M^{-1} to r

Choose the corresponding particle

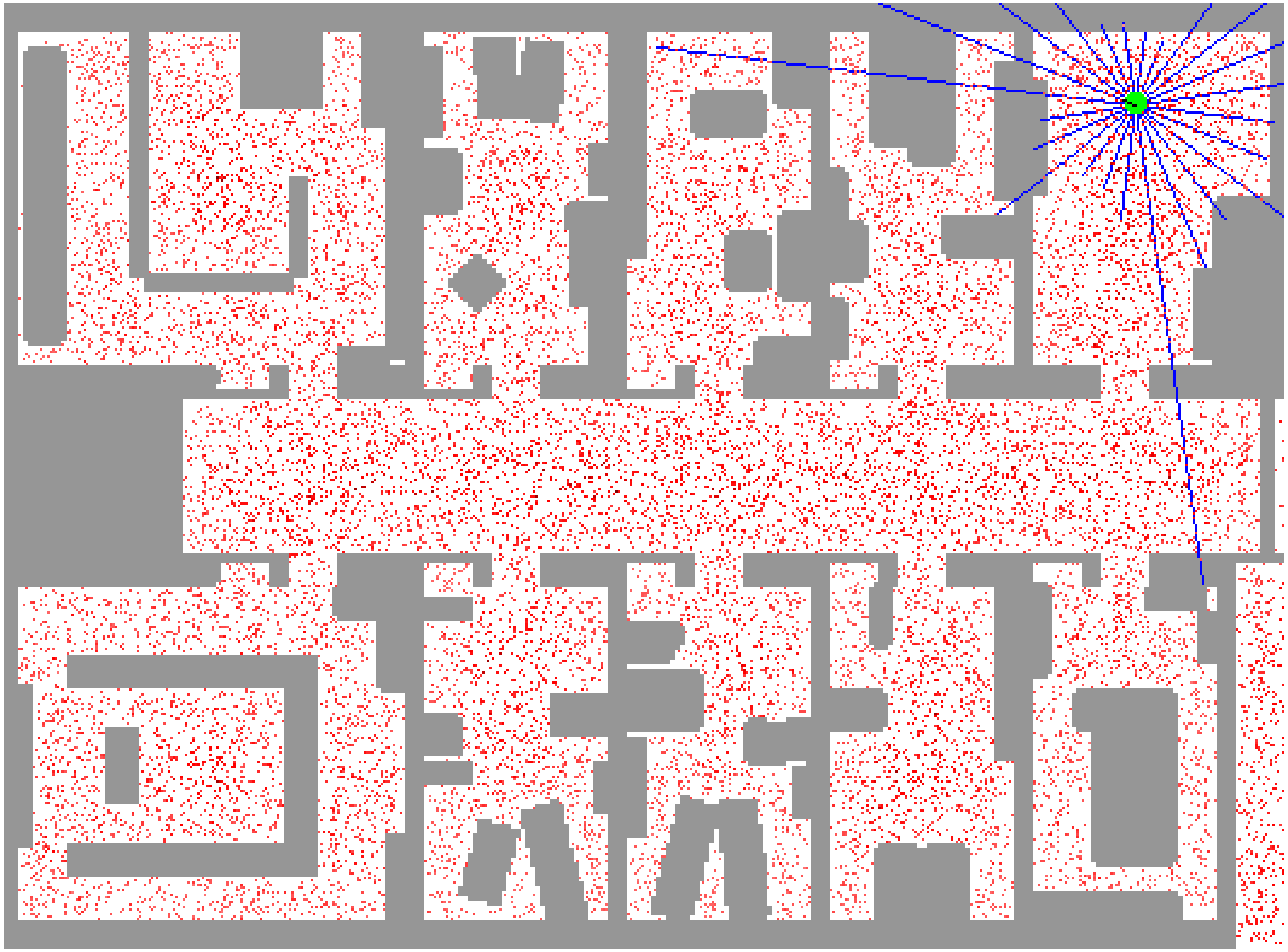
Advantages of Particle Filters

- can deal with non-liearities
- can deal with non-Gaussian noise
- mostly parallelizable
- easy to implement
- PFs focus adaptively on probable regions of state-space

Robot Localization using PF



Robot Localization using PF



References

- S. Thrun, W. Burgard, D. Fox: Probabilistic Robotics
- Stanford CS223B Computer Vision, Winter 2005, Lecture 12 Filters/Motion Tracking
- J. Xiao: Particle Filters, City College of New York.
- L. J. Latecki: Tutorial on Particle Filters, Temple Univeristy.