3 – Particle Filter

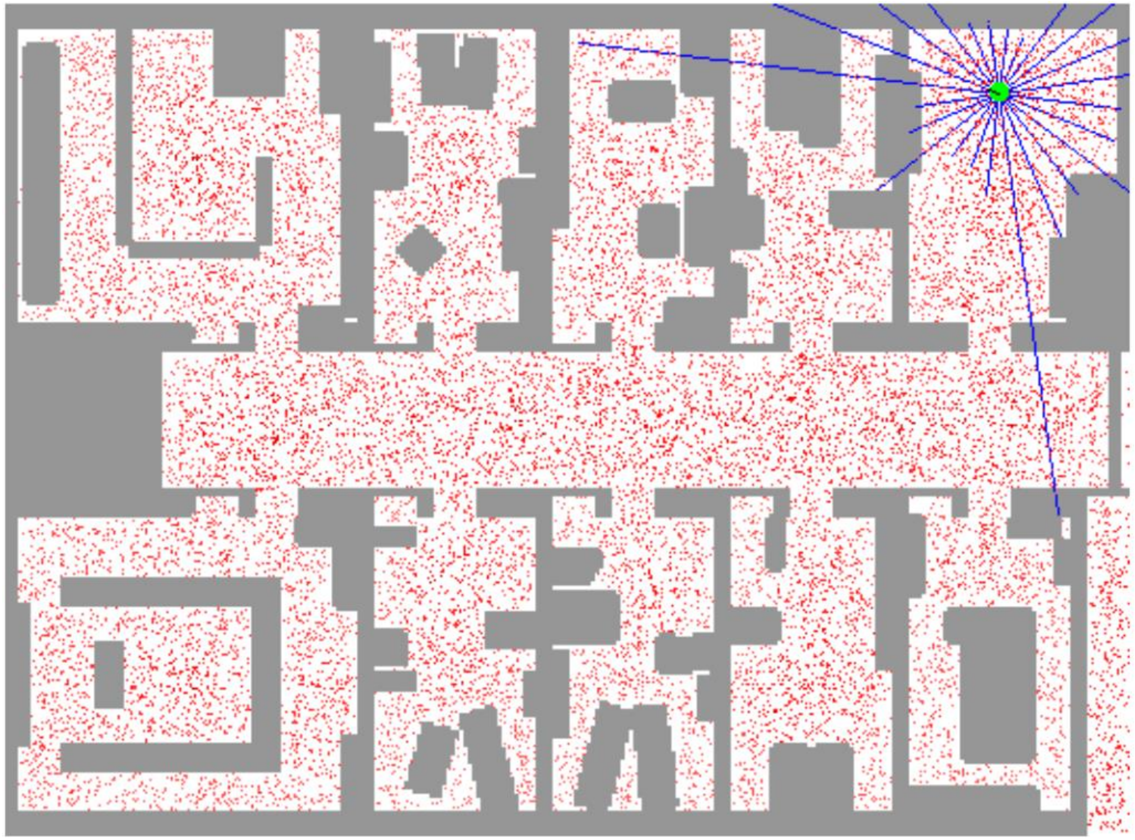
Advanced Methods for Mapping and Self-localization in Robotics (MPC-MAP)

Tomas Lazna

Brno University of Technology
2023

What can a convenient localization algorithm offer?

- Multimodality
- Continuity
- Intuitivness
- Efficiency
- Scalability



[1]

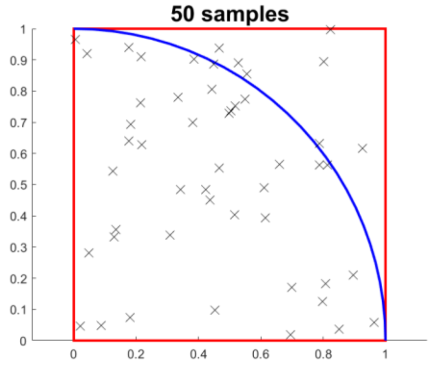
PDF = Probability Density Function

[1] TRIEBEL, Rudolph. The Particle Filter. In: *Machine Learning for Computer Vision* [online]. Technische Universität München, 2017 [cit. 2021-02-19]. Available at:https://vision.in.tum.de/_media/teaching/ss2017/ml4cv/variationalinference.pdf

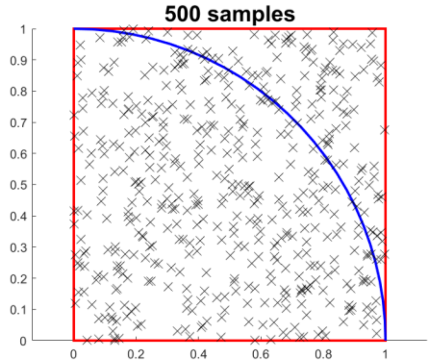
Monte Carlo methods

- Numerical methods based on random sampling
- Optimization, numerical intergration, drawing from PDFs, modeling
- Law of large numbers
- Estimating the π value (Buffon‘s needle)
- Particle filter = Sequential MC method

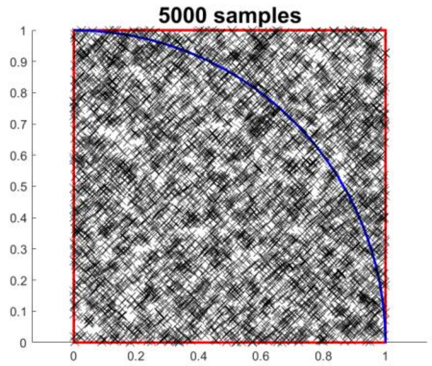
[1]



$\pi \approx 3.36$



$\pi \approx 3.10$



$\pi \approx 3.13$

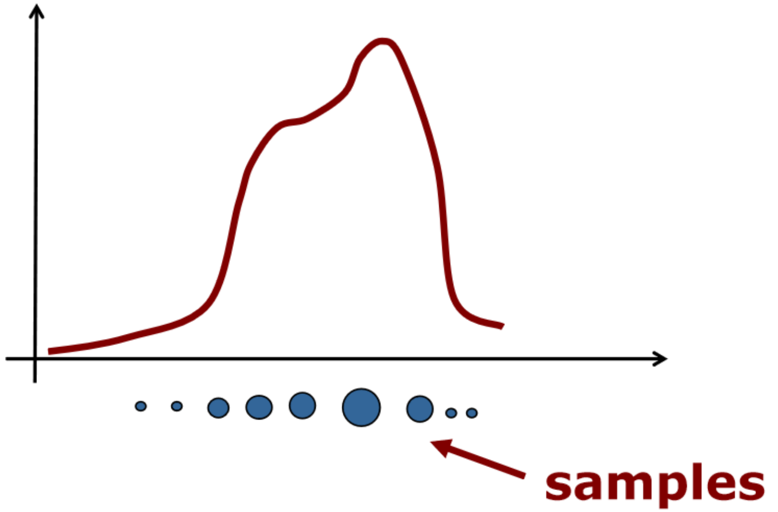
[1] Ceasars Slots. Casino Dice Games: The Complete List. Note: Illustrative photo

Introduction

- Particle representation of a PDF

$$\chi = \{[\boldsymbol{x}^{(i)}, w^{(i)}]\}_{i=1,\dots,N} \qquad p(\boldsymbol{x}) = \sum_{i=1}^{N} w^{(i)} \delta_{\boldsymbol{x}^{(i)}}(\boldsymbol{x})$$

State hypothesis   Belief (weight)   Dirac delta function



[1]

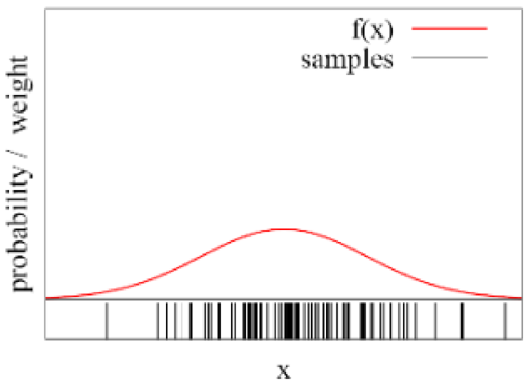
- Bayes‘ theorem

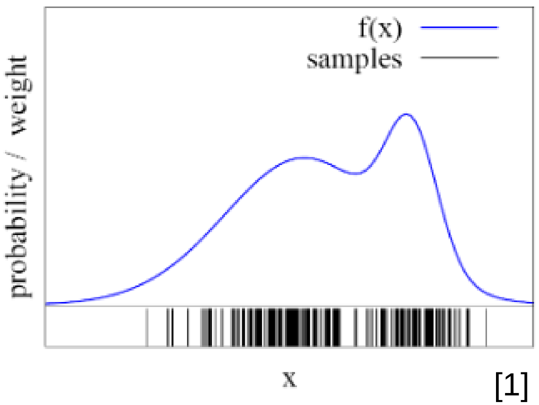
$$P(H|E) = \frac{P(E|H) \cdot P(H)}{P(E)}$$

$H \dots$ hypothesis

$E \dots$ evidence

- Particle filter applications:
  - Robot localization
  - Object tracking, computer vision
  - General estimation in nonlinear systems
- Original article (referred to as ‚bootstrap filter‘) [2]





[1]

[1] STACHNISS, Cyrill. Short Introduction to Particle Filters and Monte Carlo Localization [online]. Uni Freiburg, 2013 [cit. 2021-02-18]. Available at: http://ais.informatik.uni-freiburg.de/teaching/ws13/mapping/pdf/slam10-particle-filter-4.pdf
[2] GORDON, N.J. et al. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F Radar and Signal Processing* [online]. 1993, **140**(2) [cit. 2021-02-21]. DOI: 10.1049/ip-f-2.1993.0015

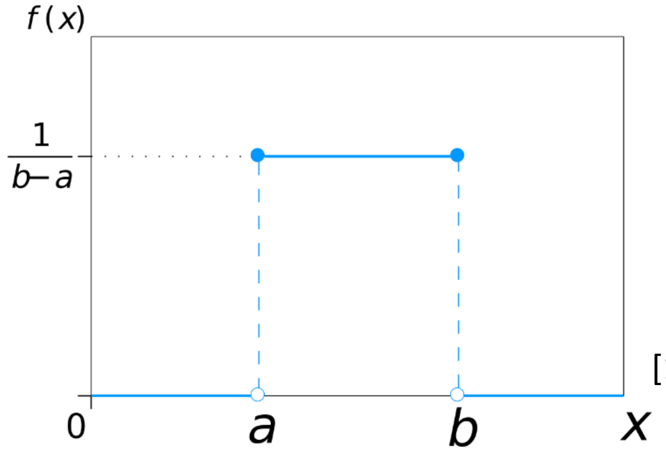
Drawing random samples

Uniform distribution
- Any random nuber generator
- Usually pseudorandom series
- Can be utilized for other PDFs

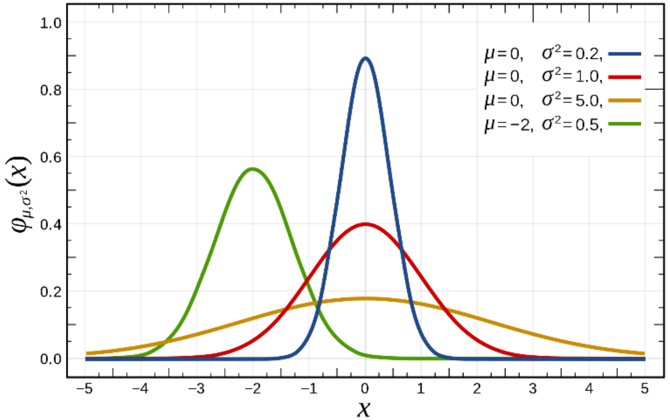
Normal distribution (Gaussian)
- Parametric function (mean μ, variance σ^2)
- Approximation by UD:

$$x \leftarrow \sum_{i=1}^{12} \mathrm{rand}(-0.5\sigma, 0.5\sigma)$$

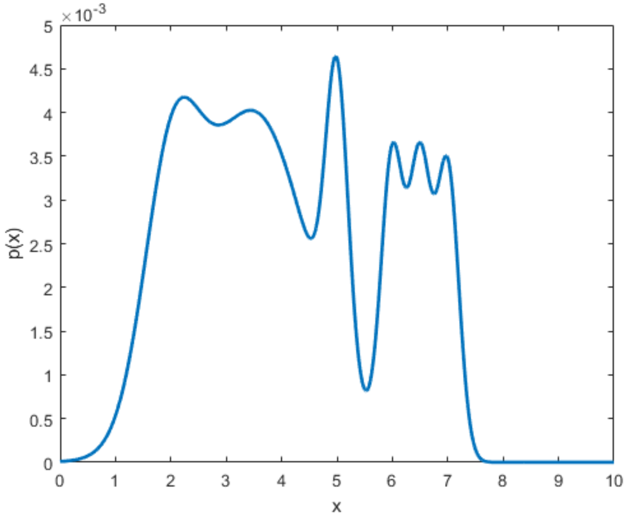
Other distributions
- Parametric
- Non-parametric

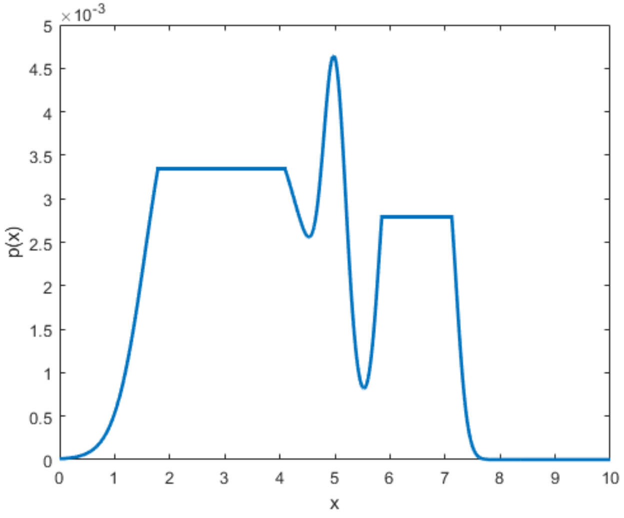


[1]



[2]

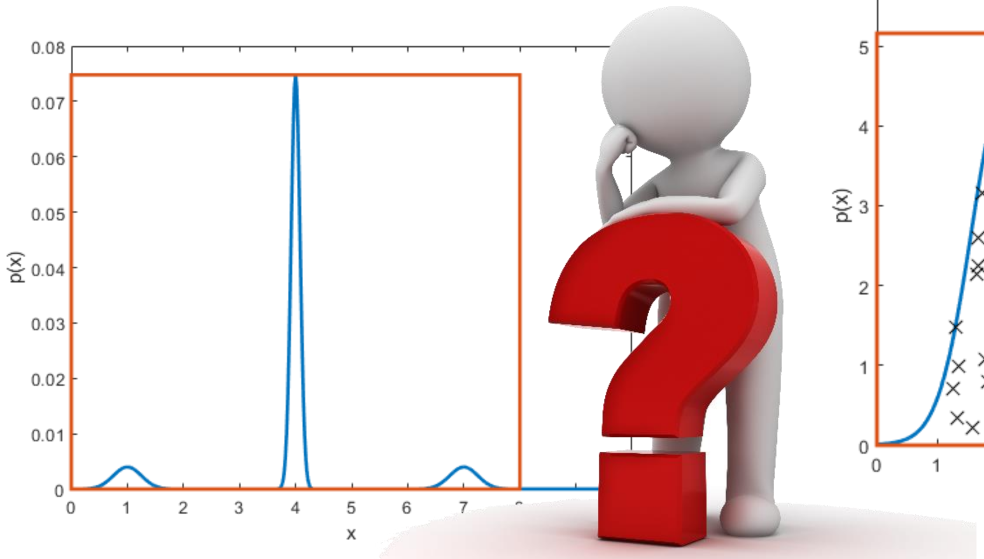


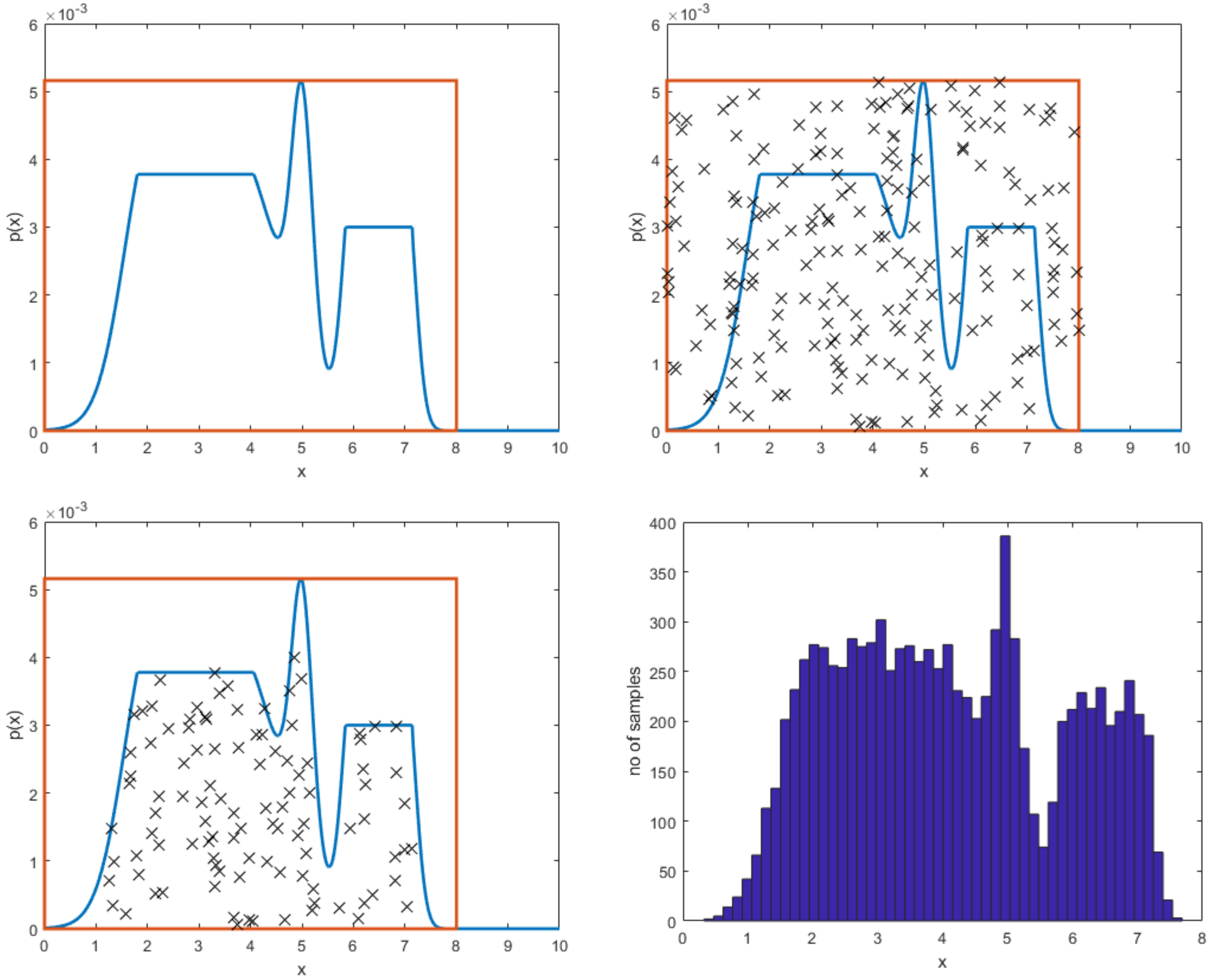


[1] Continuous uniform distribution. Wikipedia, Wikimedia Foundation, 20 Dec 2020. Available at: https://en.wikipedia.org/wiki/Continuous_uniform_distribution
[2] Normal distribution. Wikipedia, Wikimedia Foundation, 13 Feb 2021. Available at: https://en.wikipedia.org/wiki/Normal_distribution

Rejection sampling

- Generate random samples $[y,\ f(y)]$ in range:
  - $y \in \langle x_{\min};\ x_{\max} \rangle$
  - $f(y) \in (0; \max(p(x))\rangle$
- Reject sample if:
  - $f(y) > p(y)$
- The random variable Y is now distributed according to $p(x)$
- Tends to be inefficient

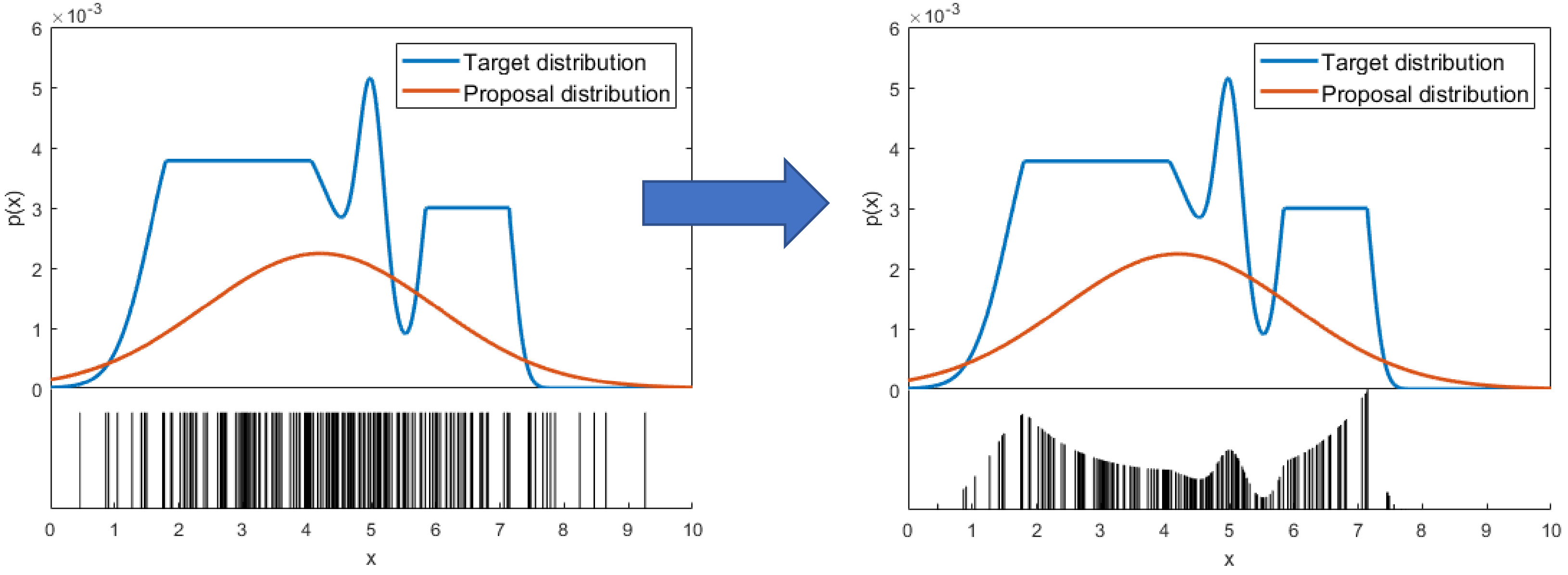




Importance sampling

- Use other distribution π that is simple to draw from
- Correct the difference between the target distribution f and proposal π by assigning ‚weights' to random samples:

$$w(x) = \frac{f(x)}{\pi(x)}$$



- To ensure samples are drawn from the whole target distribution, following condition needs to be met:

$$\forall x \in \mathbb{R}: f(x) > 0 \Rightarrow \pi(x) > 0$$

Other methods for drawing random samples

- Adaptive rejection sampling [1]
- Markov chain Monte Carlo (MCMC), e.g. Metropolis-Hastings algorithm [2]

[1] MARTINO, Luca and Joaquín MÍGUEZ. A generalization of the adaptive rejection sampling algorithm. *Statistics and Computing* [online]. 2011, **21**(4). [cit. 2021-02-19]. DOI: 10.1007/s11222-010-9197-9
[2] CHIB, Siddhartha and Edward GREENBERG. Understanding the Metropolis-Hastings Algorithm. *The American Statistician* [online]. 1995, **49**(4). [cit. 2021-02-19]. DOI: 10.1080/00031305.1995.10476177

Prediction step

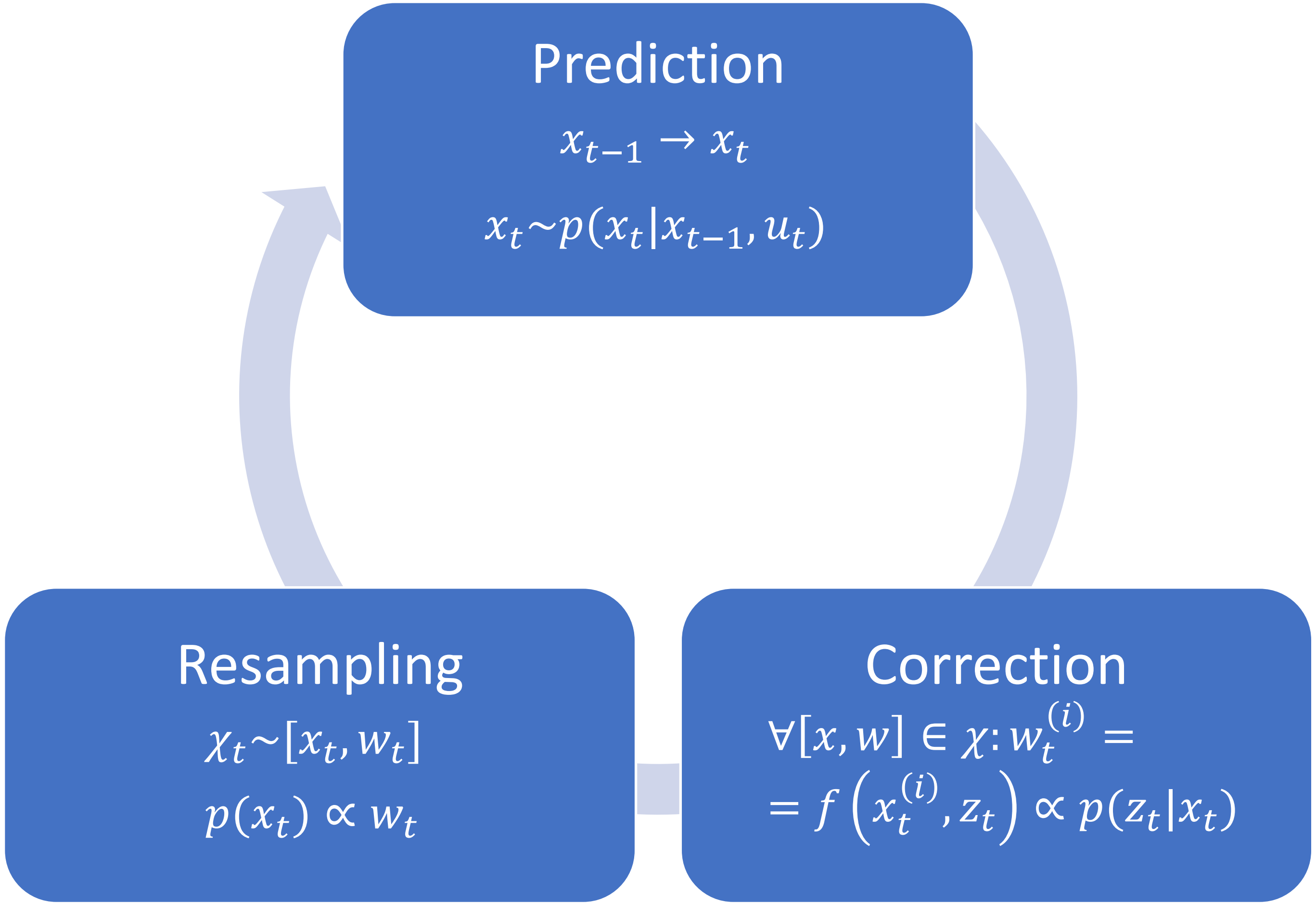
- State of all particles is updated in accordance with starting state and the input vector
- Goal: Estimate the state transition

Correction step

- Particles‘ hypotheses are compared with actual measurement
- Goal: Find out which particles are the fittest

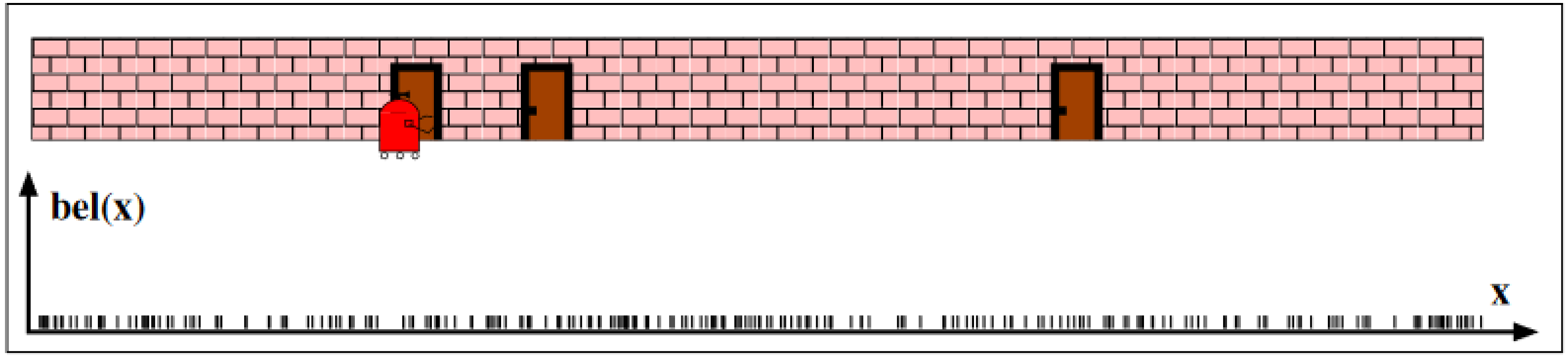
Resampling step

- Draw random samples from the previous particle set with the probability given by weights
- Goal: Increase the particle density in more probable parts of state space

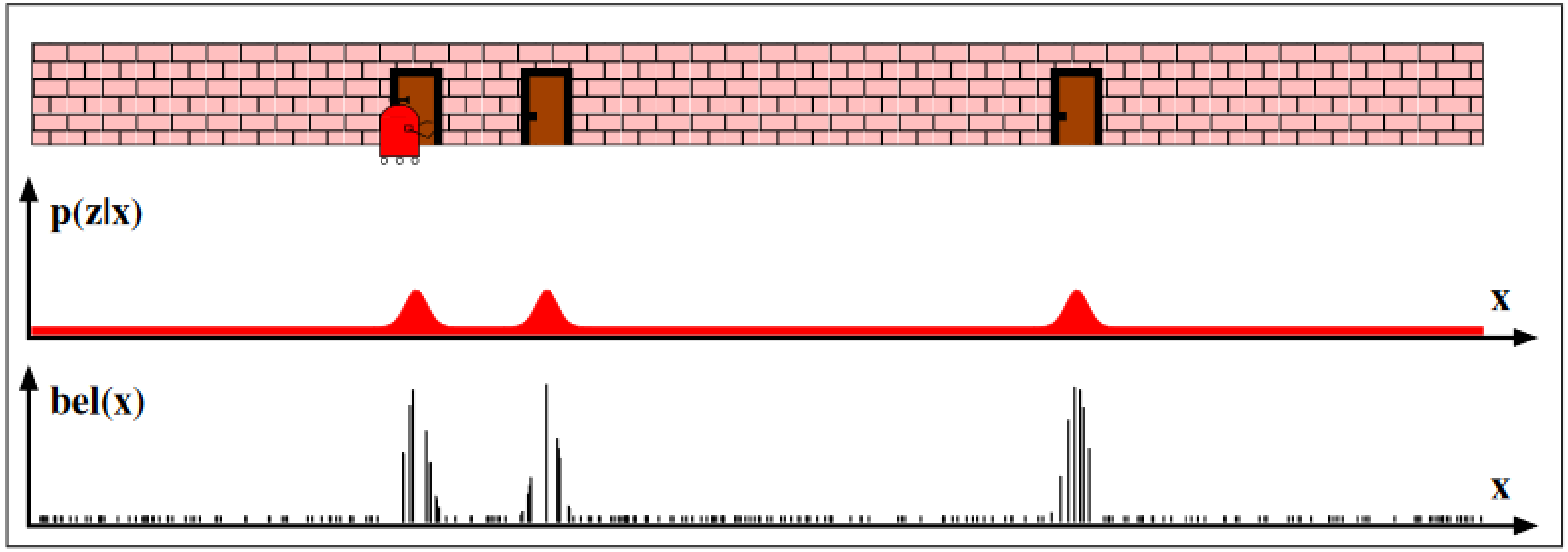


Particle filter example in 1D

1. Initialization



2. Correction

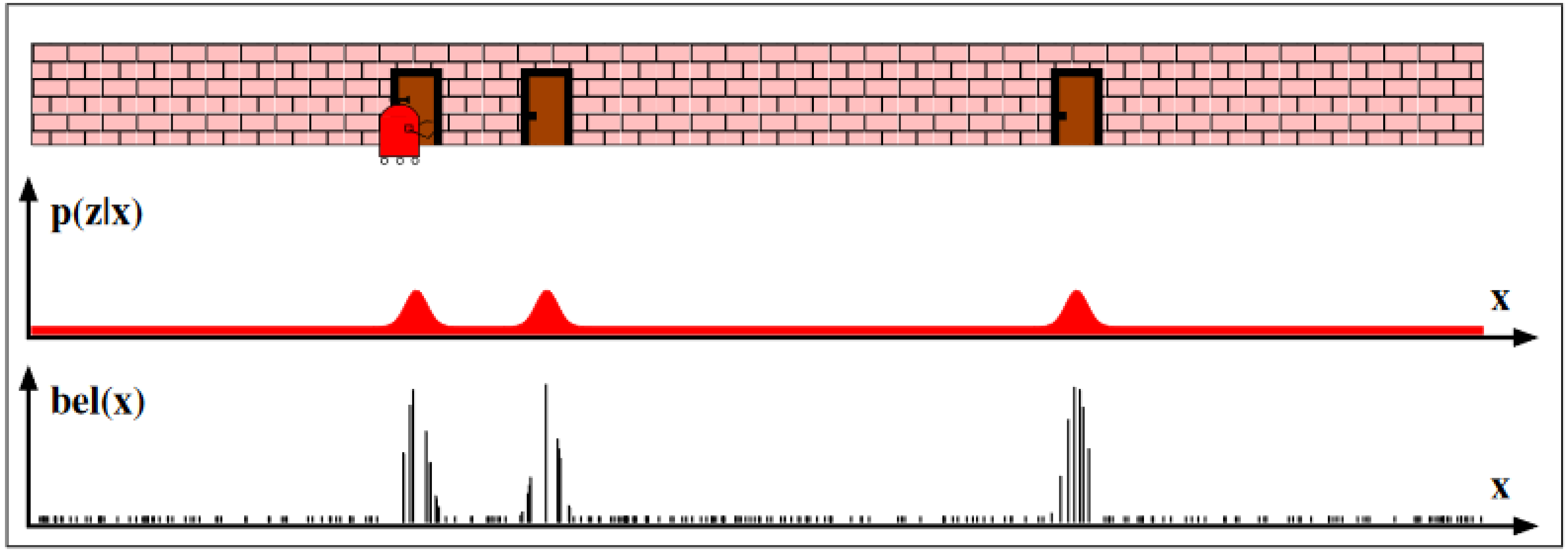


[1]

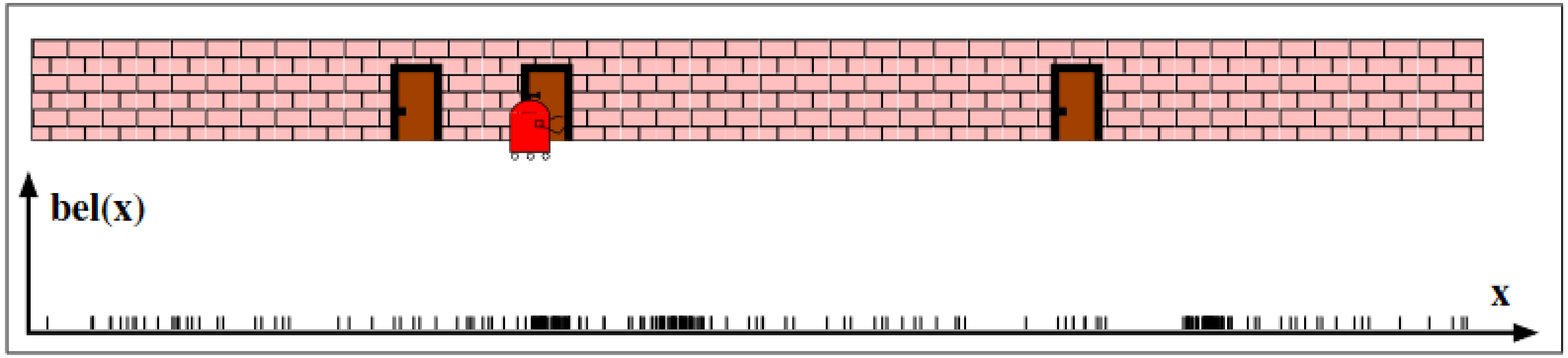
[1] THRUN, Sebastian, Wolfram BURGARD and Dieter FOX. Probabilistic robotics. Massachusetts: MIT Press, 2006. ISBN 978-0-262-20162-9.

Particle filter example in 1D

2. Correction



3. Resampling
& prediction

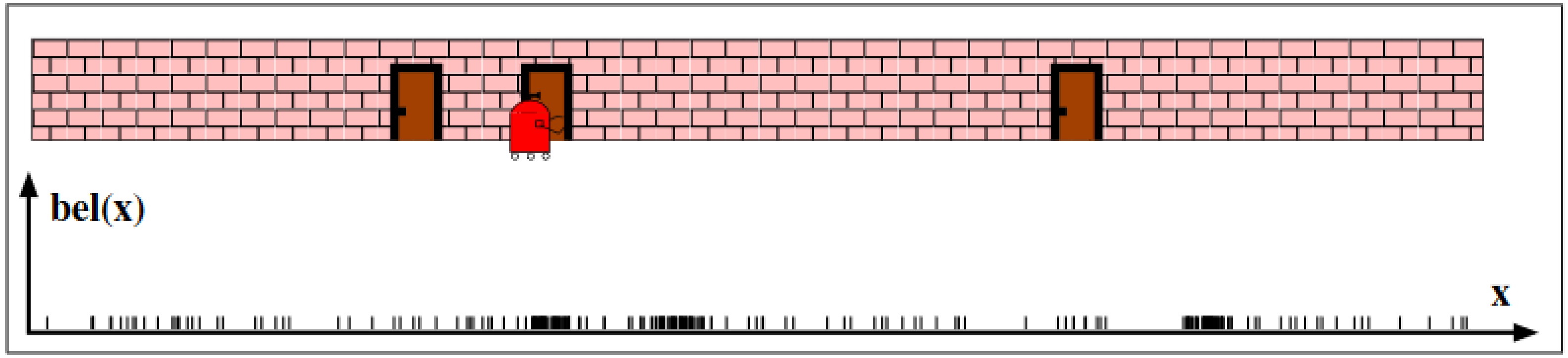


[1]

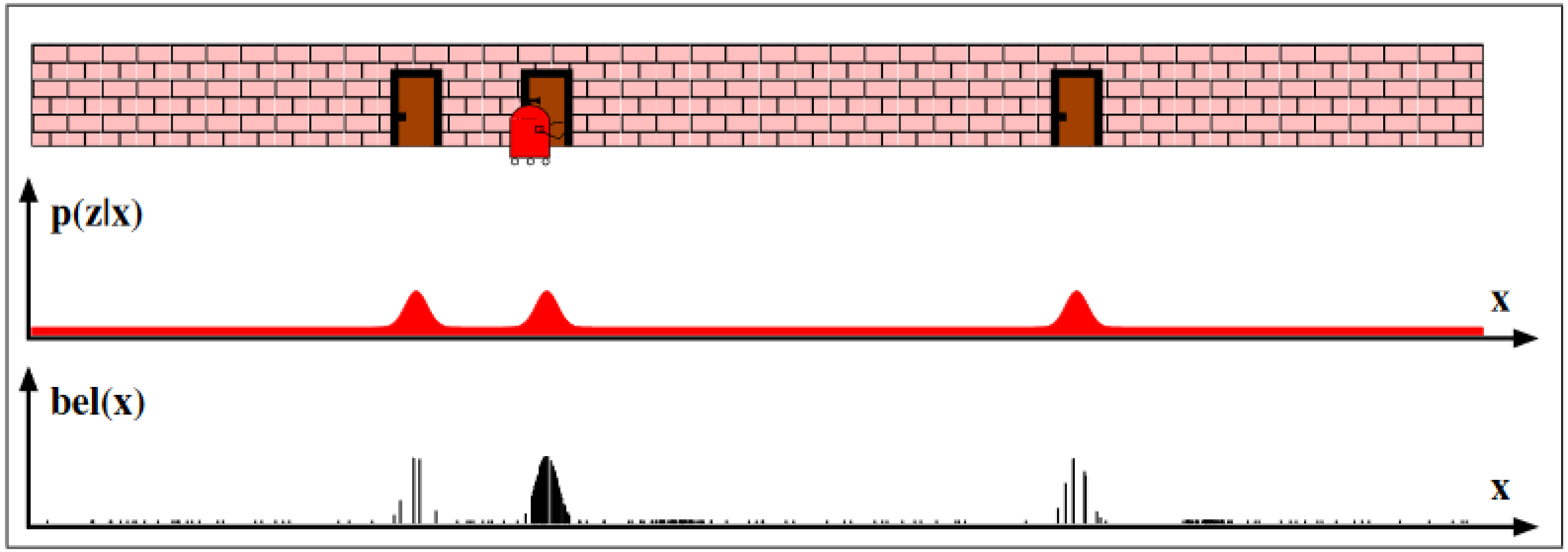
[1] THRUN, Sebastian, Wolfram BURGARD and Dieter FOX. Probabilistic robotics. Massachusetts: MIT Press, 2006. ISBN 978-0-262-20162-9.

Particle filter example in 1D

3. Resampling & prediction



4. Correction

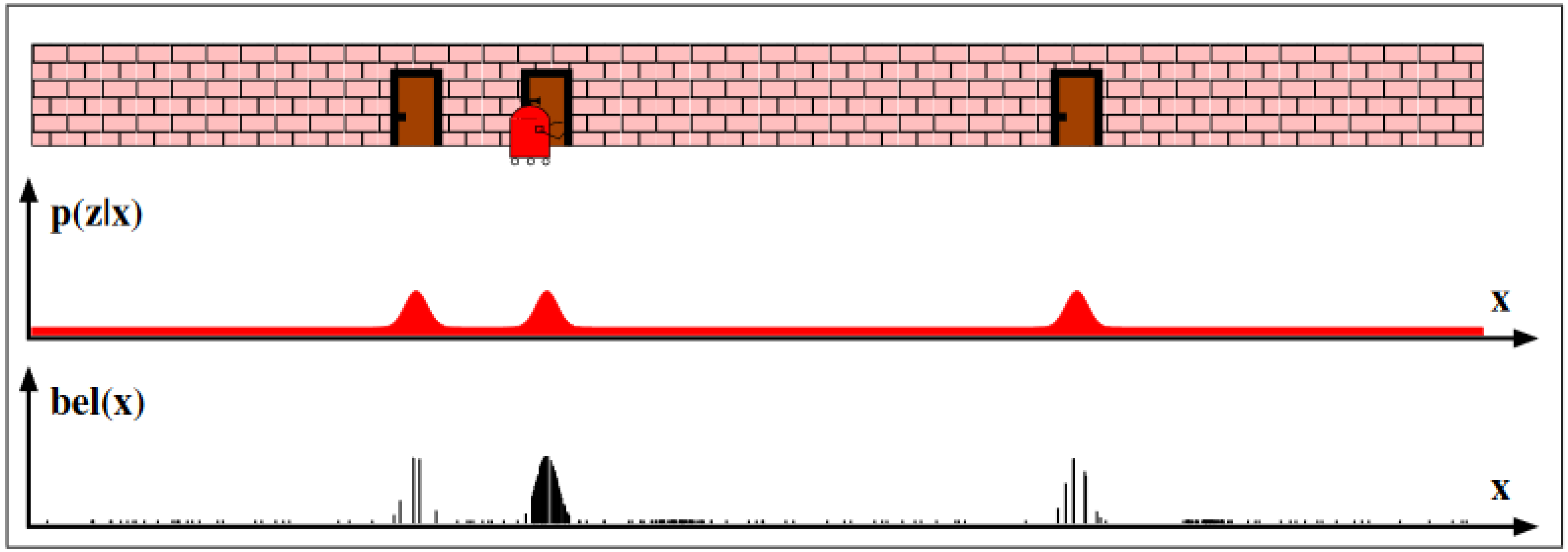


[1]

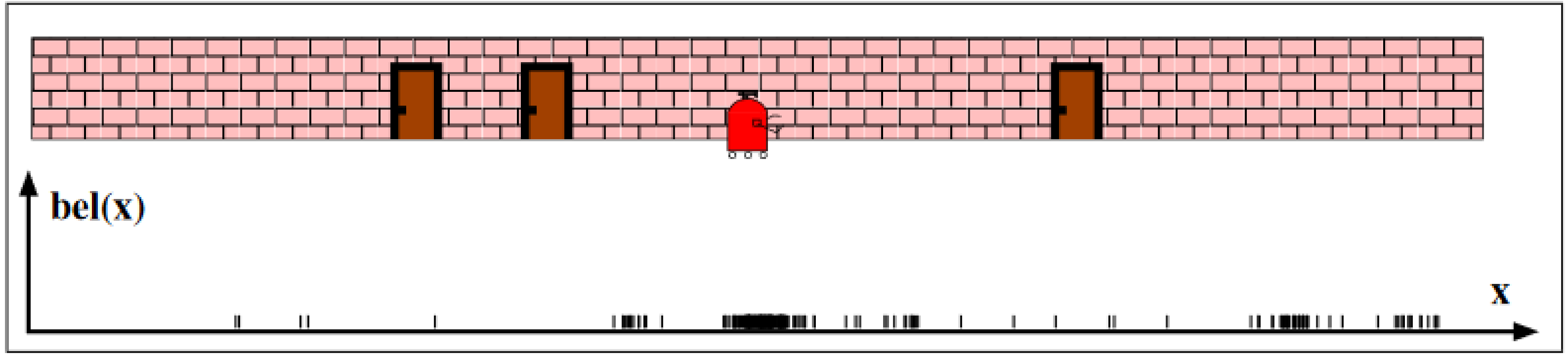
[1] THRUN, Sebastian, Wolfram BURGARD and Dieter FOX. Probabilistic robotics. Massachusetts: MIT Press, 2006. ISBN 978-0-262-20162-9.

Particle filter example in 1D

4. Correction



5. Resampling
& prediction

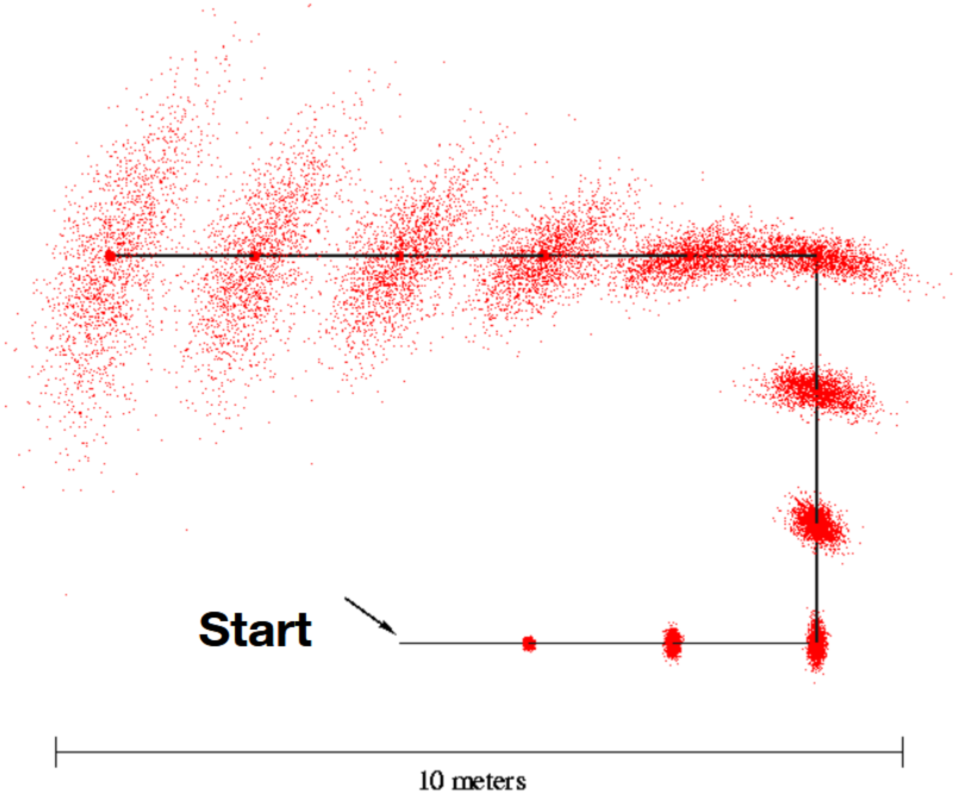


[1]

[1] THRUN, Sebastian, Wolfram BURGARD and Dieter FOX. Probabilistic robotics. Massachusetts: MIT Press, 2006. ISBN 978-0-262-20162-9.

Prediction

- Predict state of the system after a control vector is applied
- Increase variance of particles
- Let us assume that the state transition depends on the previous state only (first order Markov Process)
- The noise can have arbitrary distribution



[1]

Omnidirectional drive

$$\begin{pmatrix} x_{k+1} \\ y_{k+1} \end{pmatrix} = \begin{pmatrix} x_k \\ y_k \end{pmatrix} + \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} T + \boldsymbol{Q}$$

Noise

Differential drive

$$\begin{pmatrix} x_{k+1} \\ y_{k+1} \\ \theta_{k+1} \end{pmatrix} = \begin{pmatrix} x_k - R\sin\theta \\ y_k + R\cos\theta \\ \dot{\theta}T \end{pmatrix} + \begin{pmatrix} \cos\dot{\theta}T & -\sin\dot{\theta}T & 0 \\ \sin\dot{\theta}T & \cos\dot{\theta}T & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} R\sin\theta \\ -R\cos\theta \\ \theta_k \end{pmatrix} + \boldsymbol{Q}$$

Prediction

$$x_{t-1} \rightarrow x_t$$

$$x_t \sim p(x_t | x_{t-1}, u_t)$$

[1] TRIEBEL, Rudolph. The Particle Filter. In: *Machine Learning for Computer Vision* [online]. Technische Universität München, 2017 [cit. 2021-02-19]. Available at:https://vision.in.tum.de/_media/teaching/ss2017/ml4cv/variationalinference.pdf

- Application of Bayes‘ rule
  - Prior: all particles have the same weight 1 / N
  - Posterior: proportional to the measurement model
- Weights should be normalized

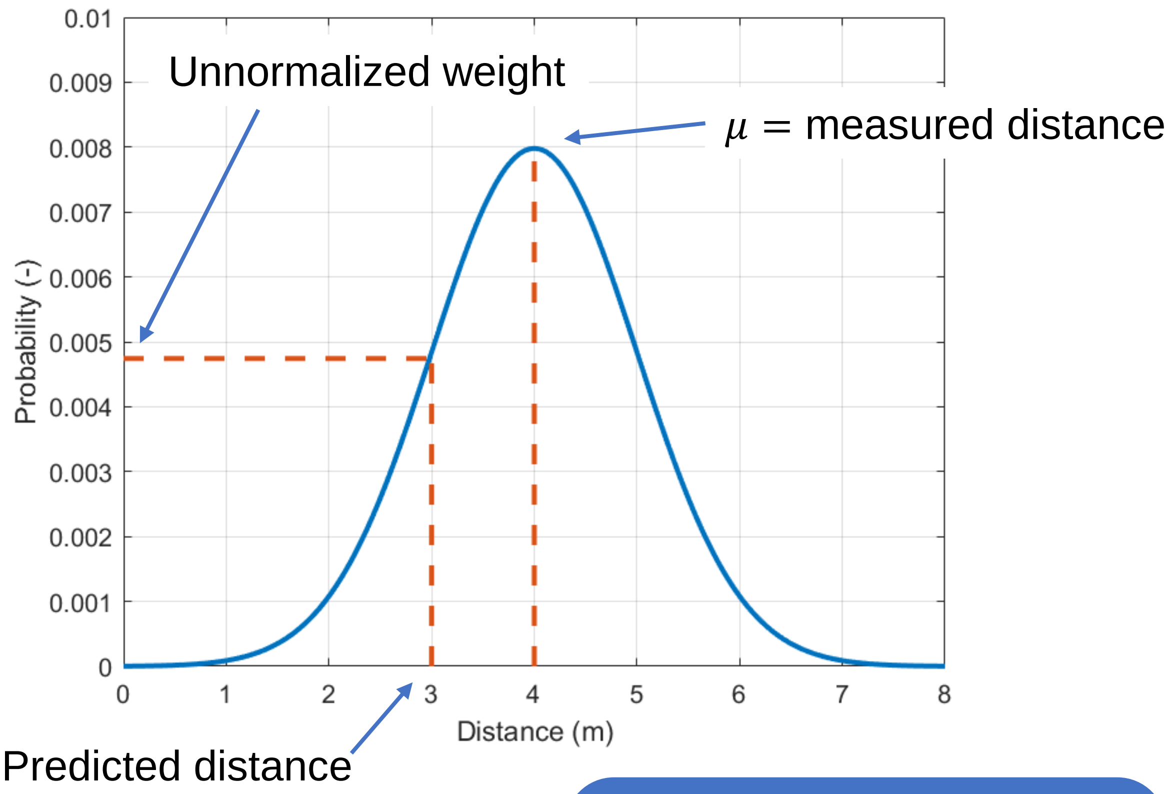
Using Normal distribution

$$w \propto \prod_{m=1}^{M} e^{-\frac{1}{2}\left(\frac{d_m - p_m}{\sigma}\right)^2}$$

Example for rangefinders

Using Euclidean distance

$$w \propto \frac{1}{\sqrt{\sum_{m=1}^{M}(d_m - p_m)^2}}$$



d … measured distance
p … predicted distance
M … number of measurements

Correction

$$\forall [x, w] \in \chi : w_t^{(i)} = f\left(x_t^{(i)}, z_t\right) \propto p(z_t | x_t)$$

Resampling

- Increase density of particles in regions of high posterior probability and vice versa
- Needed in case of limited number of samples
- Draw N particles with the probability given by weights of original set

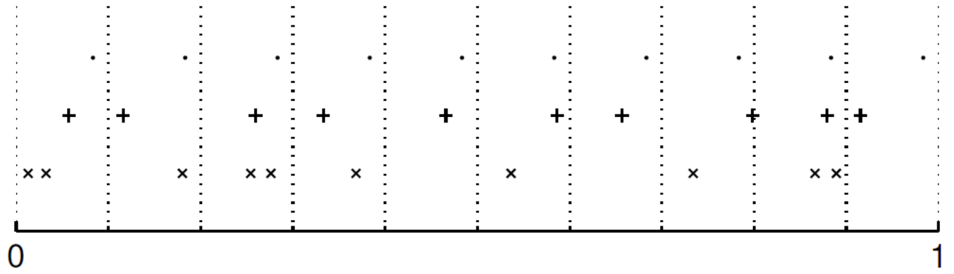
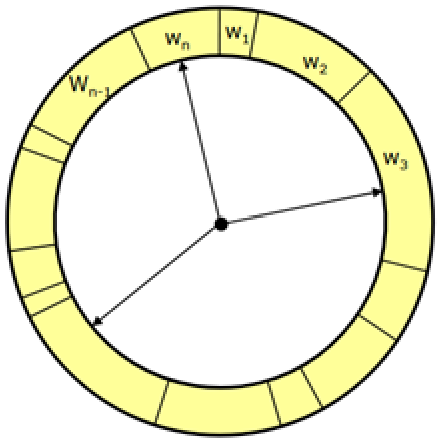


Fig. 1. Ten standard uniform samples generated using multinomial resampling (x), stratified resampling (+) and systematic resampling (·). [1]

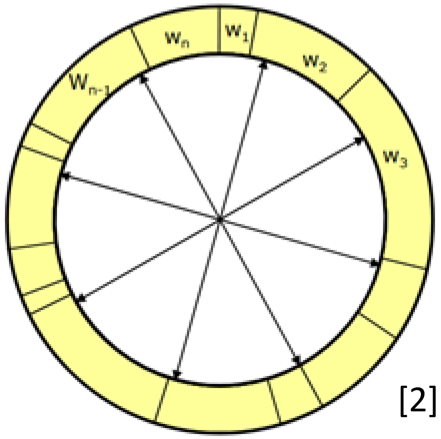
General algorithm

1. Generate sorted set of N random numbers u_k in range $\langle 0, 1)$
2. Compute cummulative sum of weights
3. For each u_k pick particle x_i according to condition:

$$u_k \in \left\langle \sum_{s=1}^{i-1} w_s, \sum_{s=1}^{i} w_s \right)$$



Roulette wheel



[2]

Systematic

Resampling

$\chi_t \sim [x_t, w_t]$

$p(x_t) \propto w_t$

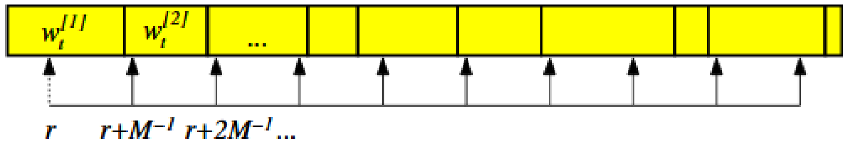
[1] HOL, Jeroen D. et al. On Resampling Algorithms for Particle Filters. In: *2006 IEEE Nonlinear Statistical Signal Processing Workshop* [online]. IEEE, 2006 [cit. 2021-02-20]. DOI: 10.1109/NSSPW.2006.4378824
[2] STACHNISS, Cyrill. Short Introduction to Particle Filters and Monte Carlo Localization [online]. Uni Freiburg, 2013 [cit. 2021-02-18]. Available at: http://ais.informatik.uni-freiburg.de/teaching/ws13/mapping/pdf/slam10-particle-filter-4.pdf

Resampling – algorithms

Low variance systematic resampling

- Only one iteration through the set of weights
- Keeps particles of even weights alive
- Generate random numer $\tilde{u}$ in range $\left(0, \frac{1}{N}\right)$

$$u_k = \frac{(k-1) + \tilde{u}}{N}$$



[1]

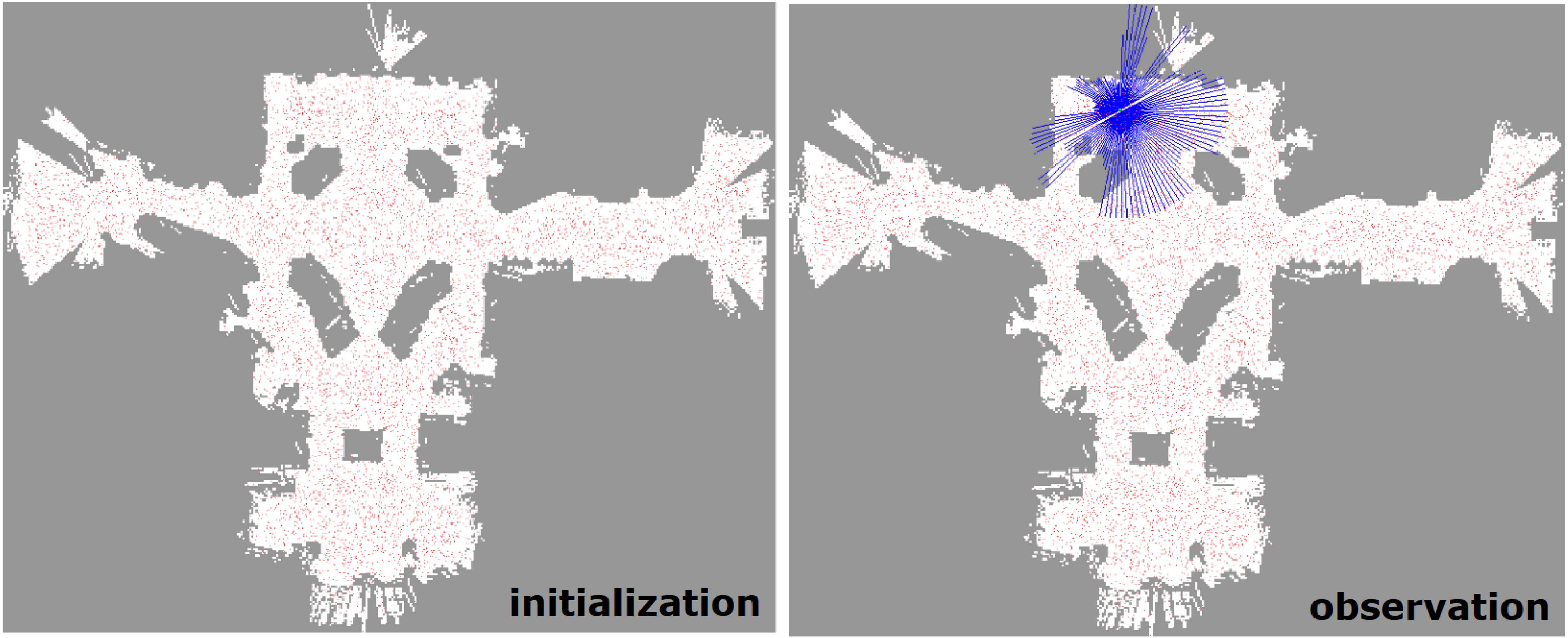
Thrun's heuristic algorithm

- Higher degree of randomness
- Easy implementation

```
index = rand(0, N-1)
for i = 1 to N do
    beta = rand(0, 2 * w_max)
    while w[index] < beta
        beta = beta – w[index]
        index++
        if index > N
            index = 1
    new_particles[i] = particles[index]
```

[1] THRUN, Sebastian, Wolfram BURGARD and Dieter FOX. Probabilistic robotics. Massachusetts: MIT Press, 2006. ISBN 978-0-262-20162-9.

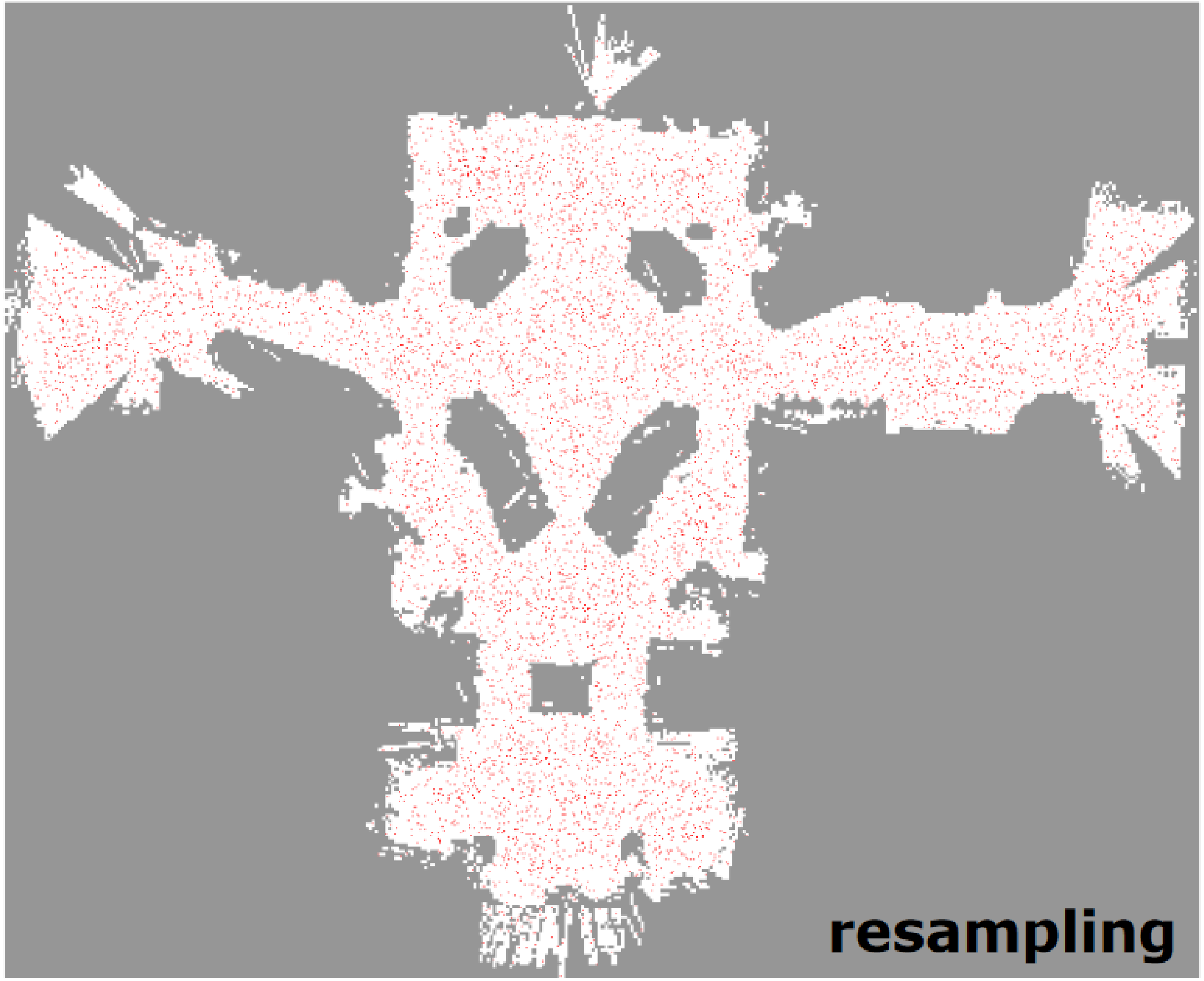
Particle filter example in 2D

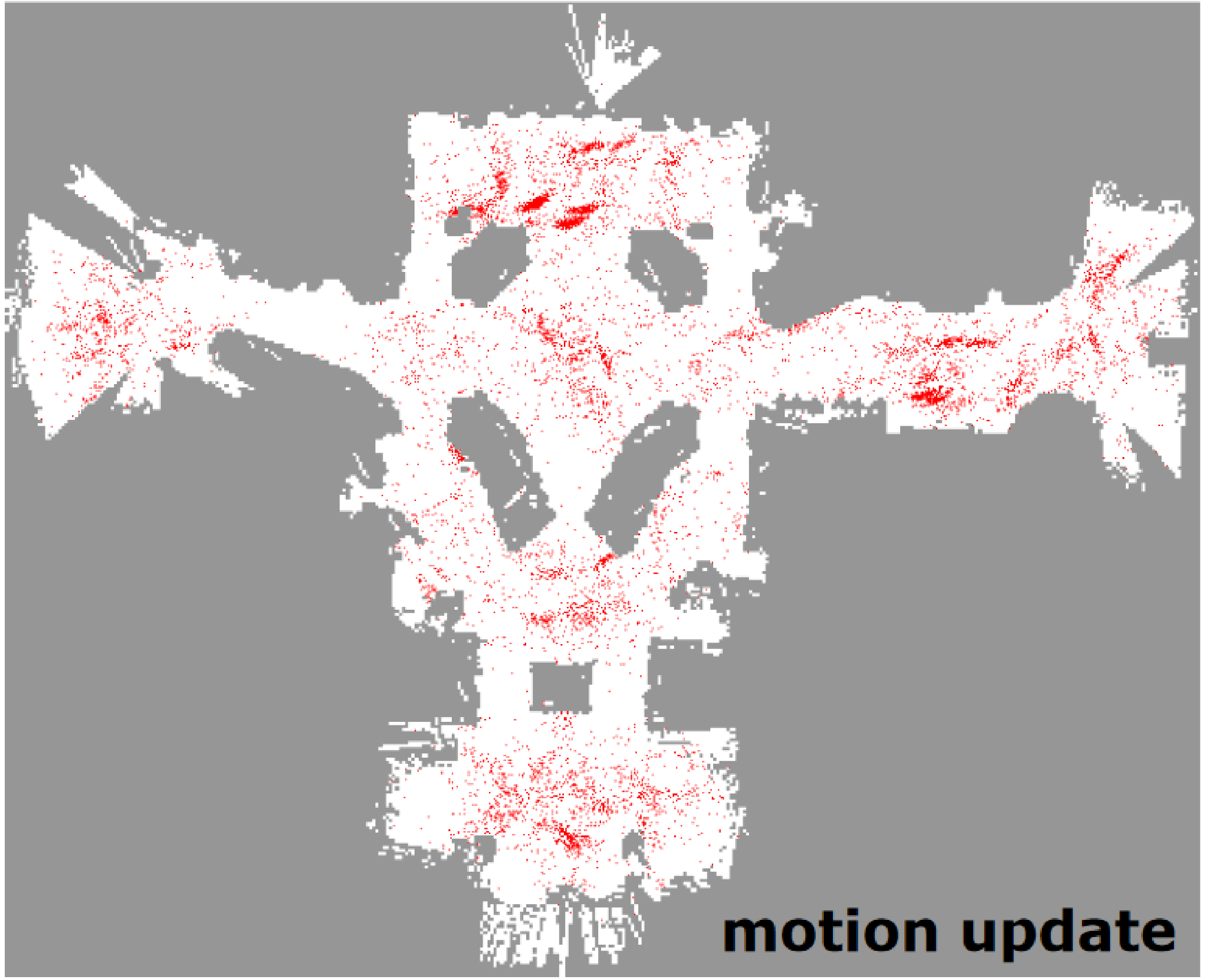


[1]

[1] STACHNISS, Cyrill. Short Introduction to Particle Filters and Monte Carlo Localization [online]. Uni Freiburg, 2013 [cit. 2021-02-18]. Available at: http://ais.informatik.uni-freiburg.de/teaching/ws13/mapping/pdf/slam10-particle-filter-4.pdf

Particle filter example in 2D

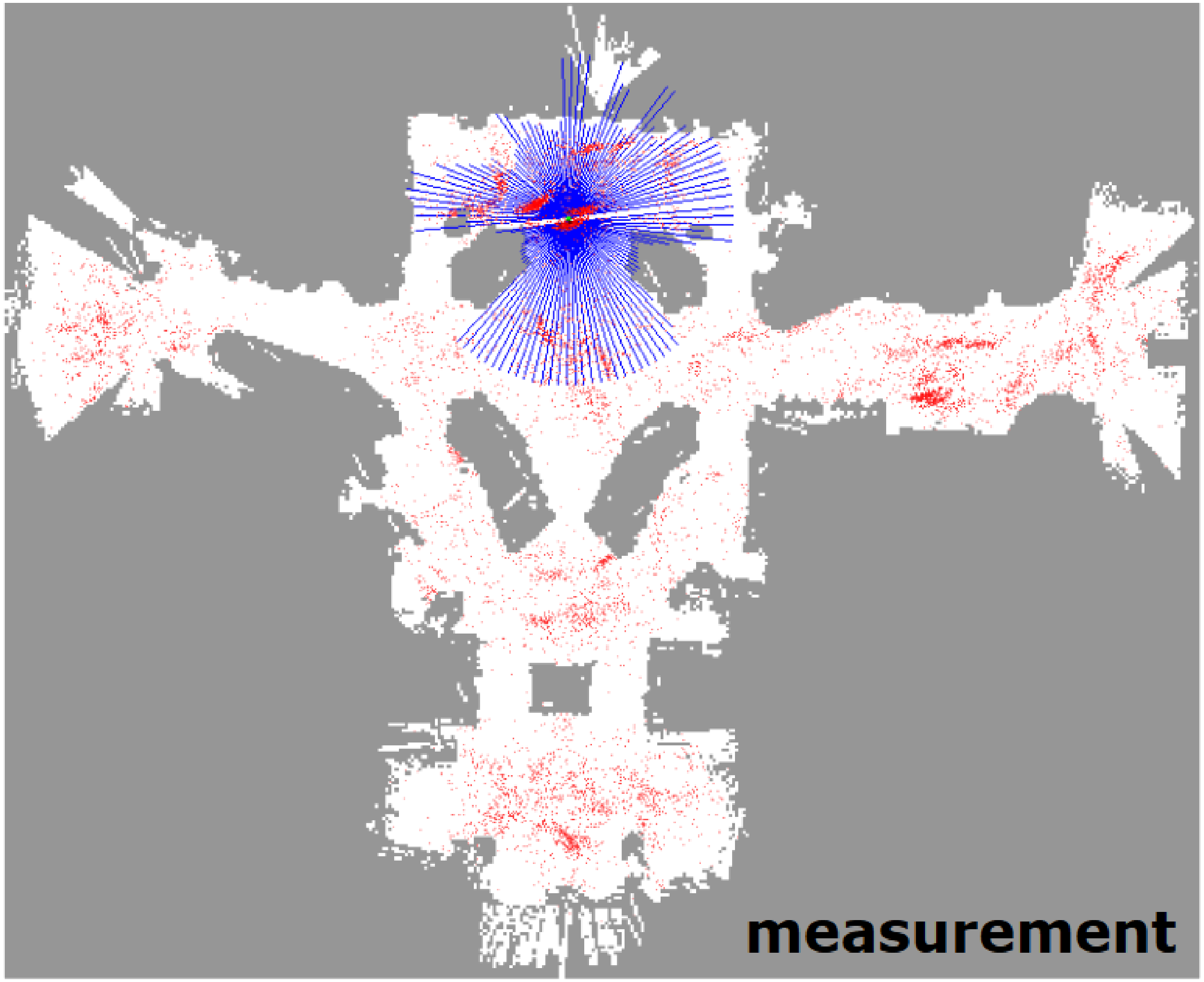




[1]

[1] STACHNISS, Cyrill. Short Introduction to Particle Filters and Monte Carlo Localization [online]. Uni Freiburg, 2013 [cit. 2021-02-18]. Available at: http://ais.informatik.uni-freiburg.de/teaching/ws13/mapping/pdf/slam10-particle-filter-4.pdf

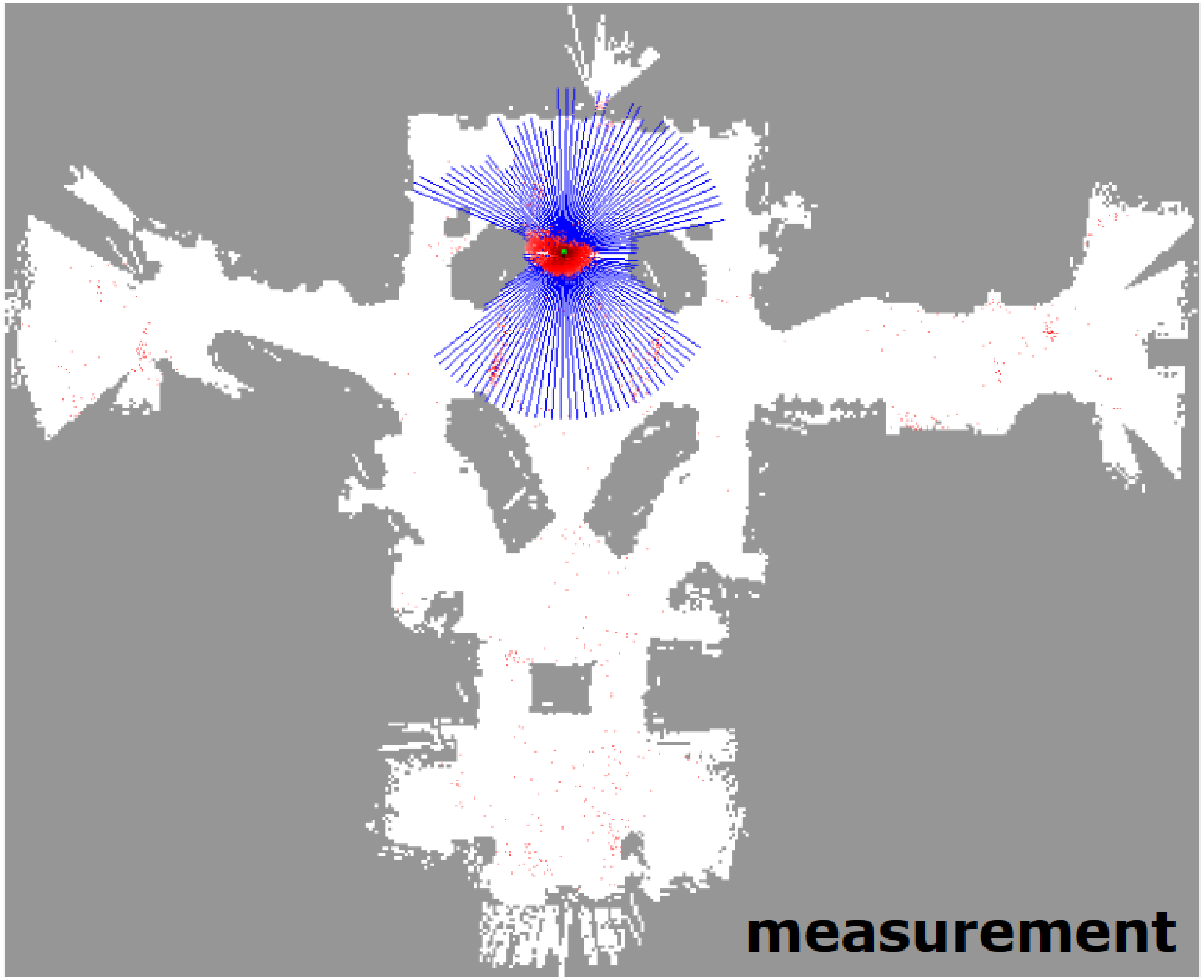
Particle filter example in 2D



[1]

[1] STACHNISS, Cyrill. Short Introduction to Particle Filters and Monte Carlo Localization [online]. Uni Freiburg, 2013 [cit. 2021-02-18]. Available at: http://ais.informatik.uni-freiburg.de/teaching/ws13/mapping/pdf/slam10-particle-filter-4.pdf

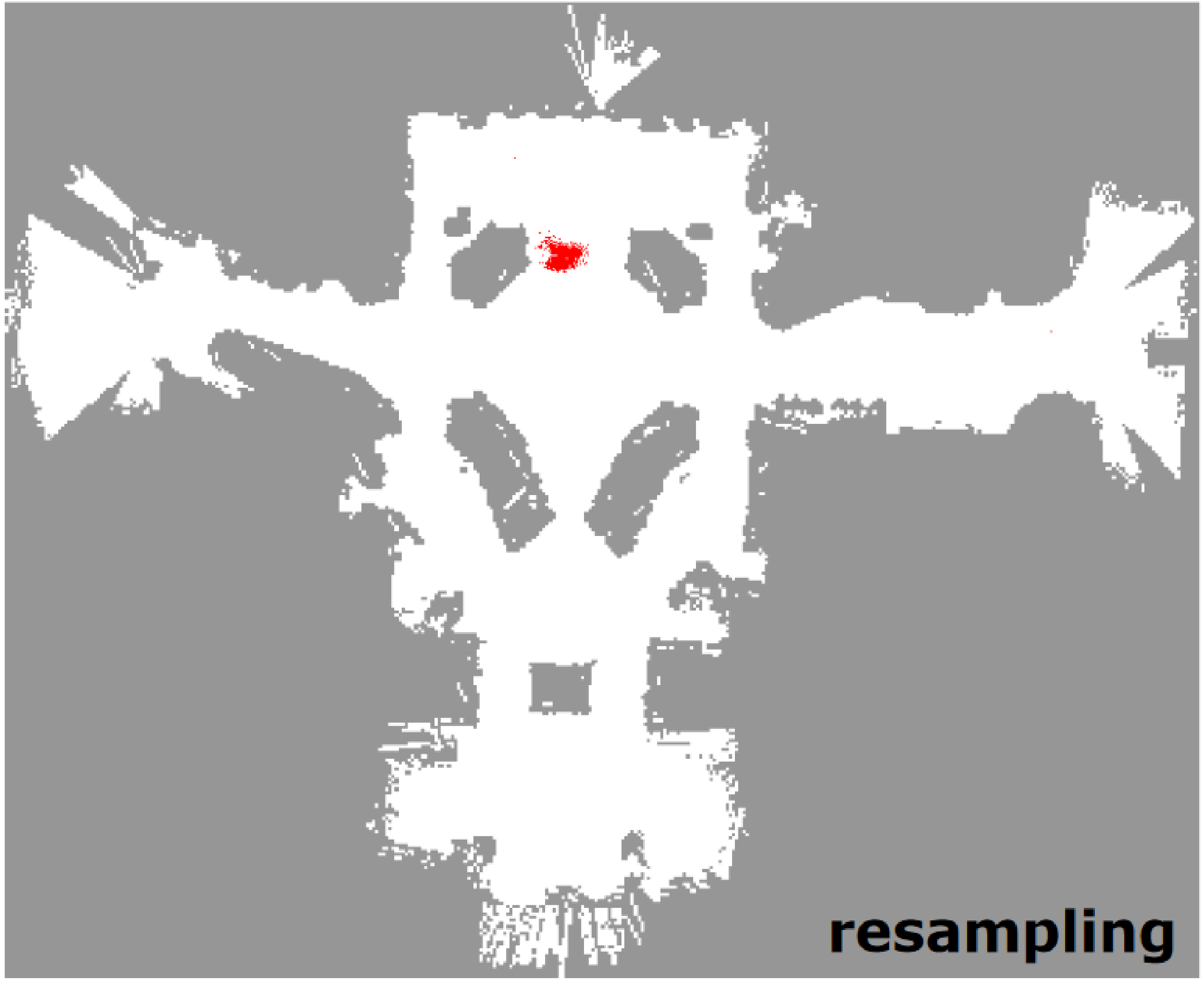
Particle filter example in 2D

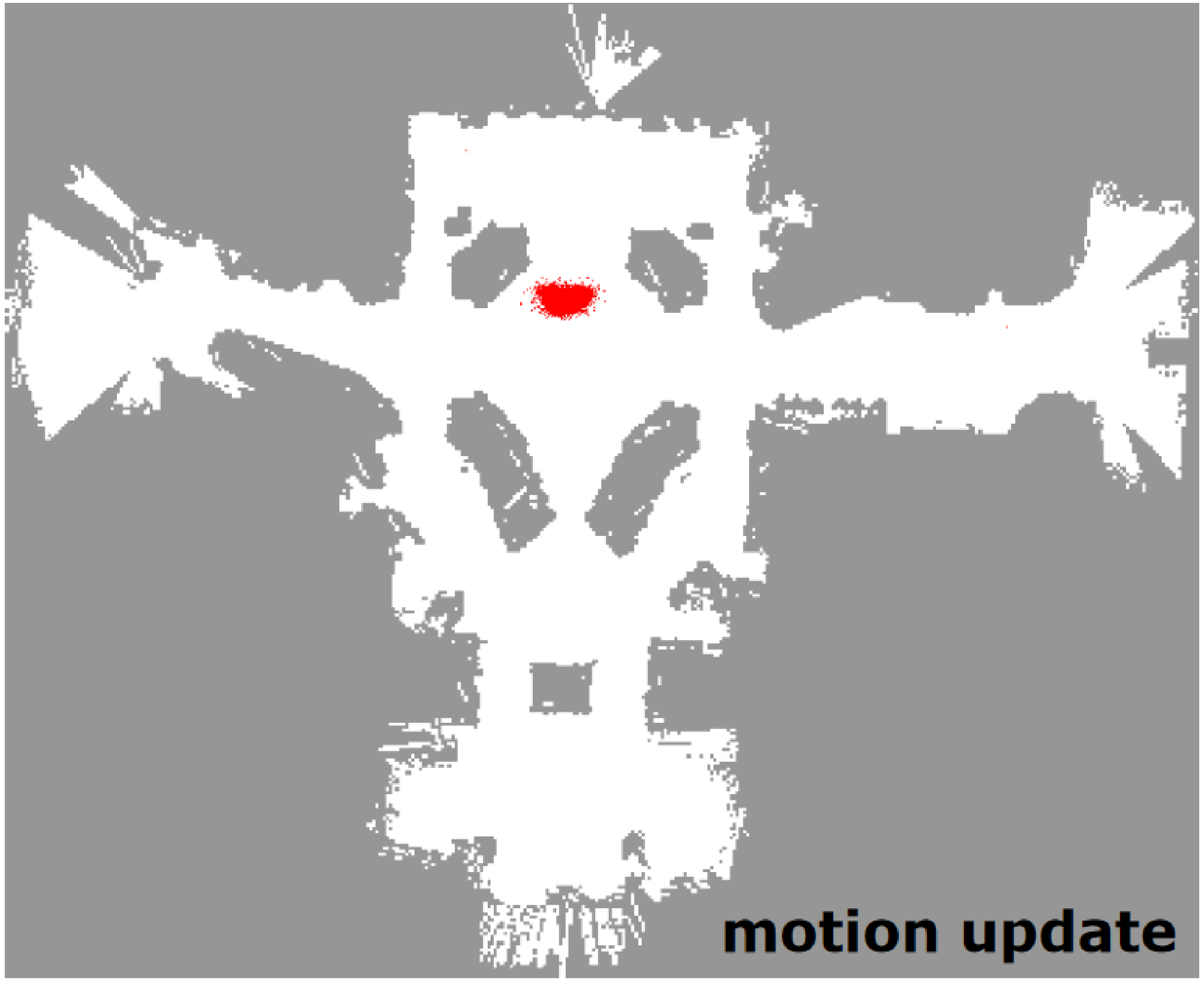


[1]

[1] STACHNISS, Cyrill. Short Introduction to Particle Filters and Monte Carlo Localization [online]. Uni Freiburg, 2013 [cit. 2021-02-18]. Available at: http://ais.informatik.uni-freiburg.de/teaching/ws13/mapping/pdf/slam10-particle-filter-4.pdf

Particle filter example in 2D

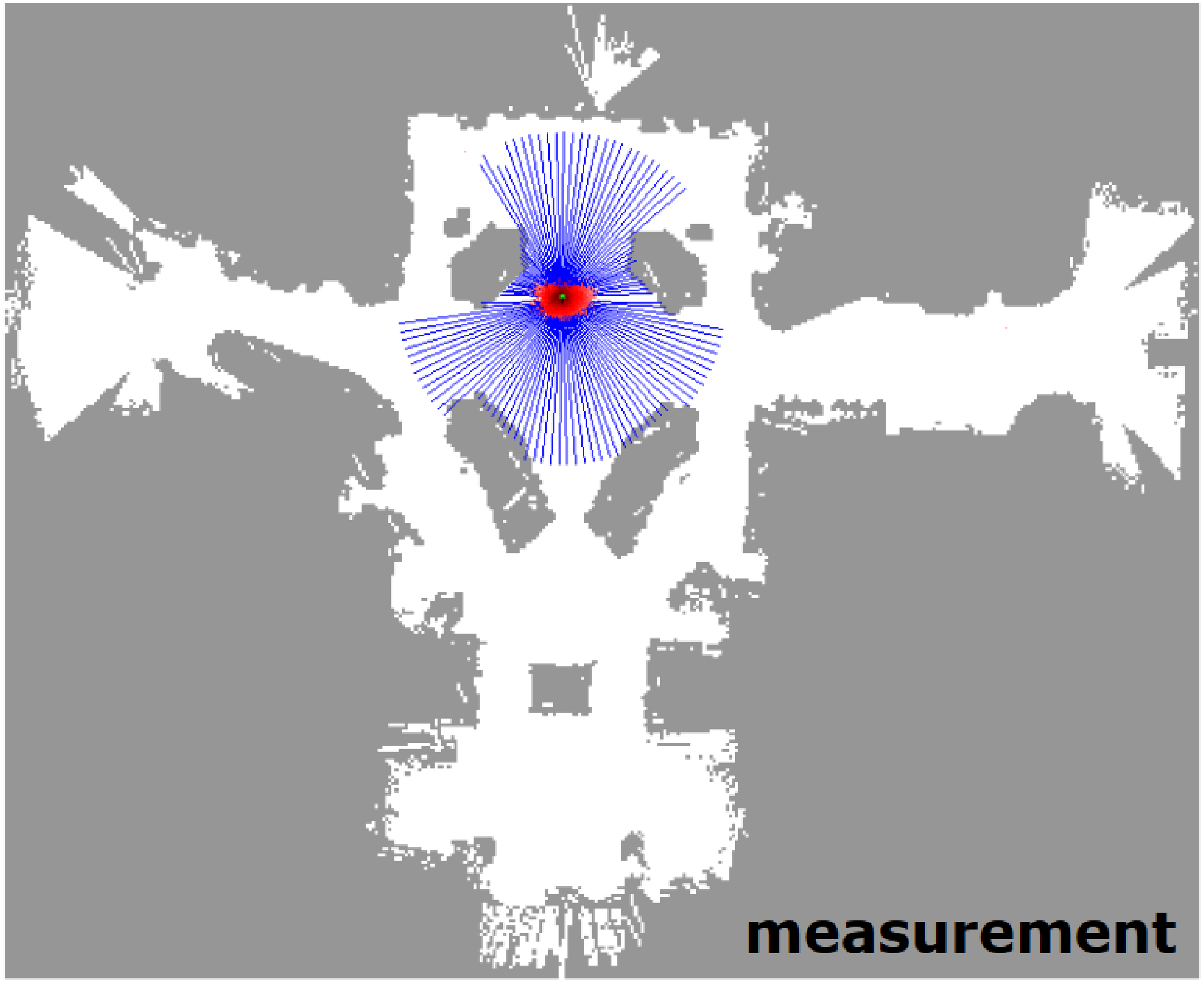




[1]

[1] STACHNISS, Cyrill. Short Introduction to Particle Filters and Monte Carlo Localization [online]. Uni Freiburg, 2013 [cit. 2021-02-18]. Available at: http://ais.informatik.uni-freiburg.de/teaching/ws13/mapping/pdf/slam10-particle-filter-4.pdf

Particle filter example in 2D



[1]

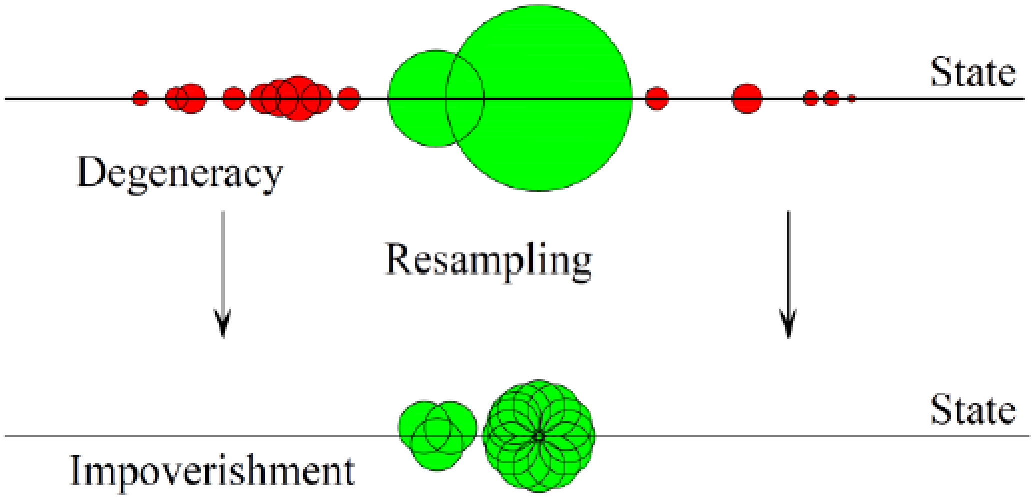
[1] STACHNISS, Cyrill. Short Introduction to Particle Filters and Monte Carlo Localization [online]. Uni Freiburg, 2013 [cit. 2021-02-18]. Available at: http://ais.informatik.uni-freiburg.de/teaching/ws13/mapping/pdf/slam10-particle-filter-4.pdf

Particle filter in pseudocode

Particle_filter(χ_{t-1}, u_t, z_t):	
$\bar{\chi}_t = \chi_t = \emptyset$	
for $n = 1$ **to** N **do**	
sample $x_t^{(n)} \sim p\left(x_t \mid u_t, x_{t-1}^{(n)}\right)$	prediction step
$w_t^{(m)} = p\left(z_t \mid x_t^{(n)}\right)$	compute weight (correction)
$\bar{\chi}_t \leftarrow \left[x_t^{(n)}, w_t^{(m)}\right]$	keep list of weighted original particles
r = rand(0, 1/N)	low variance resampling algorithm
$c = w_t^{(1)}, i = 1$	
for $n = 1$ **to** N **do**	
$u = r + (n-1)/N$	
while $u > c$	
$c = c + w_t^{(++i)}$	
$\chi_t \leftarrow x_t^{(i)}$	
return χ_t	

Particle degeneracy

- Each resampling step results in so-called particle degeneracy and impoverishment
- Particles with low probability are eliminated while particles with large weights are exist in too many copies



[1]

How to address the degeneracy issue?

- It is essential to increase the variance of the particle set
- Intensify noise in the prediction step
- Add Gaussin noise in the resampling step (= regularized particle filters [2])

$$x_t^{(i)} = x_t^{(i)} + h\Gamma_t\varepsilon$$

h … Bandwidth
Γ_t … Square root of empirical covariance matrix
ε … Random vector drawn from Gaussian kernel

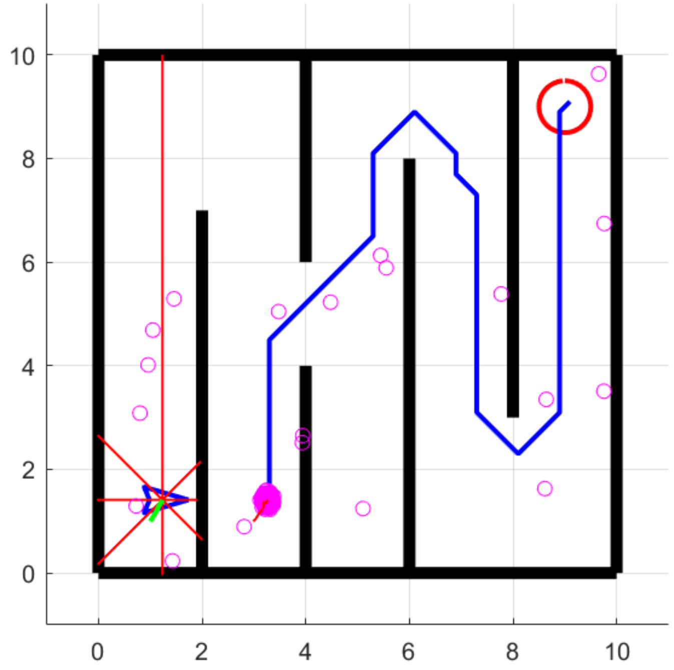
[1] LI, Tiancheng et al. Fight sample degeneracy and impoverishment in particle filters: A review of intelligent approaches. *Expert Systems with Applications* [online]. 2014, **41**(8) [cit. 2021-02-21]. DOI: 10.1016/j.eswa.2013.12.031
[2] MUSSO, Christian et al. Improving Regularised Particle Filters. *Sequential Monte Carlo Methods in Practice* [online]. New York, NY: Springer New York, 2001 [cit. 2021-02-21]. DOI: 10.1007/978-1-4757-3437-9_12

Kidnapped robot problem

- Robot is relocated by some unpredictable intervention
- The localization algorithm converges to a wrong location due to similarity of different parts of a map
- Tests the robustness of localization

Possible solutions for particle filter

- Assess the quality of localization
  - Sample covariance matrix
  - Error function (unnormalized weights)
- Reset the filter (i.e. initialize particles)
- Injection of random particles
  - Fixed rate
  - Adaptive rate



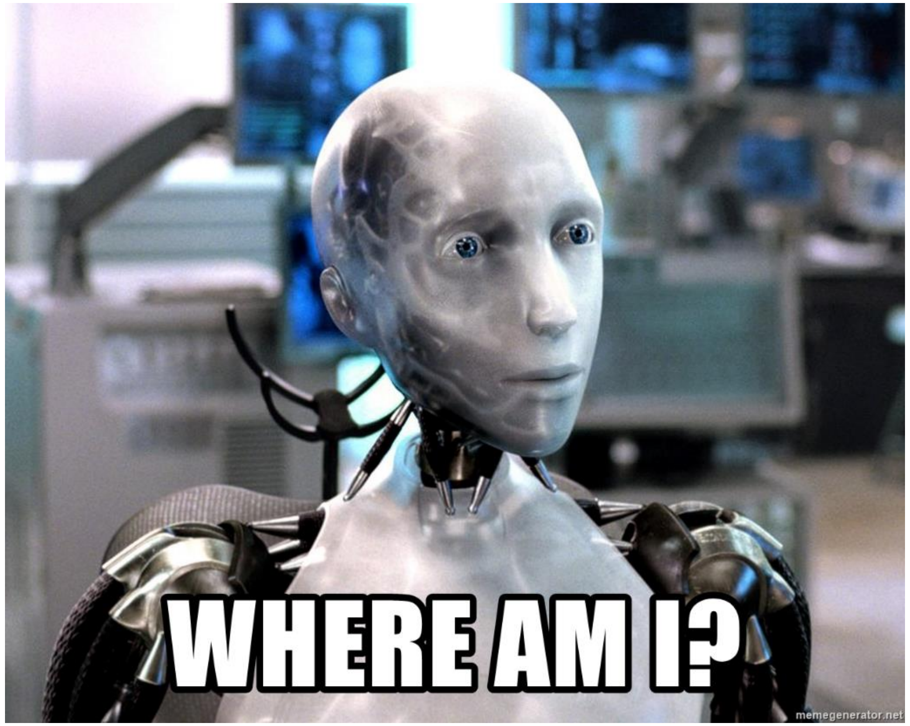
Summary

Particle filter

- Non-parametric recursive Bayes filter
- Approximates the posterior by weighted samples
- Can represent PDFs that are not Gaussian and model non-linear transitions
- Basic principles:
  Importance sampling and Survival-of-the-fittest

Monte Carlo localization (MCL)

- Based on the particle filter
- Prediction: Applying the motion model to particles
- Correction: Likelihood of observations
- Easy implementation
- Accuracy and robustness depends on the quality of motion and measurement models
- Standard for mobile robots localization



[1]

[1] I, robot [film]. Directed by PROYAS, Alex. USA: 20th Century Fox, 2004.

Profile

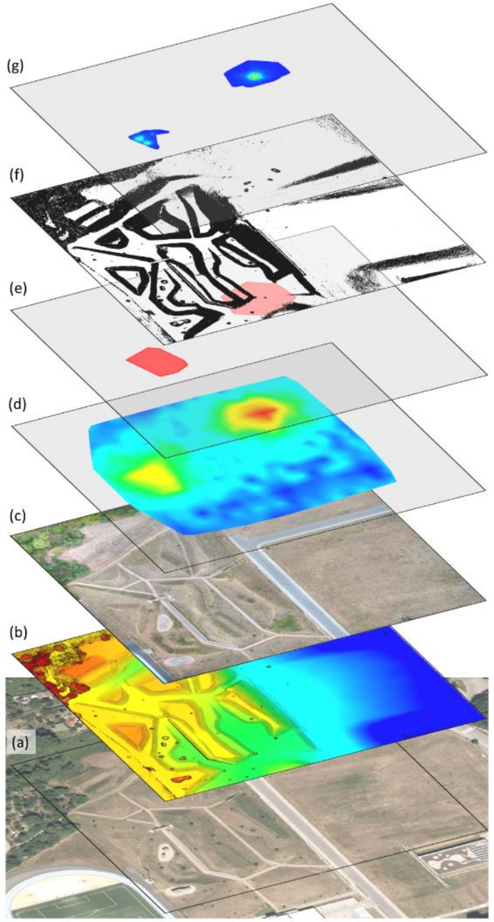
Tomas Lazna

Position: Ph.D. Student @ FEEC,
Junior Researcher @CEITEC

Research Topic: Radiation mapping via robotic platforms

Room: SE1.102

Contact: Chat @ MS Teams,
tomas.lazna@ceitec.vutbr.cz



Background:

- Motion planning in mobile robotics
- Radiation data processing
- Cooperation of UASs and UGVs
- Estimation problems

Hobbies and interests:

- Star Wars & science fiction
- LEGO
- Politics

BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

Tomas Lazna
tomas.lazna@ceitec.vutbr.cz

Brno University of Technology
Faculty of Electrical Engineering and Communication
Department of Control and Instrumentation

Robotics and AI Research Group