Freie Universität Berlin

Introduction to Robotics

Probabilistic State Estimation II, Monte-Carlo Particle Filters, SLAM

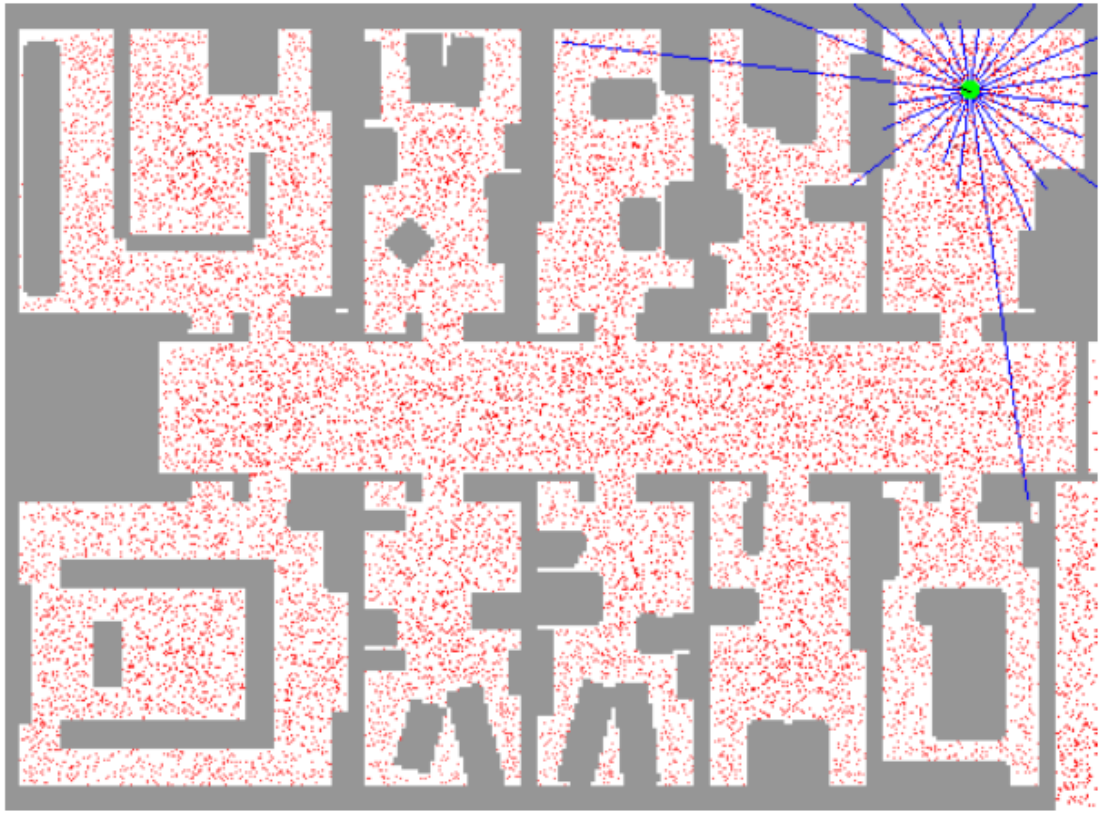
WS 2017 / 18

Prof. Dr. Daniel Göhring
Intelligent Systems and Robotics
Department of Computer Science
Freie Universität Berlin

Freie Universität Berlin

Particle Filters

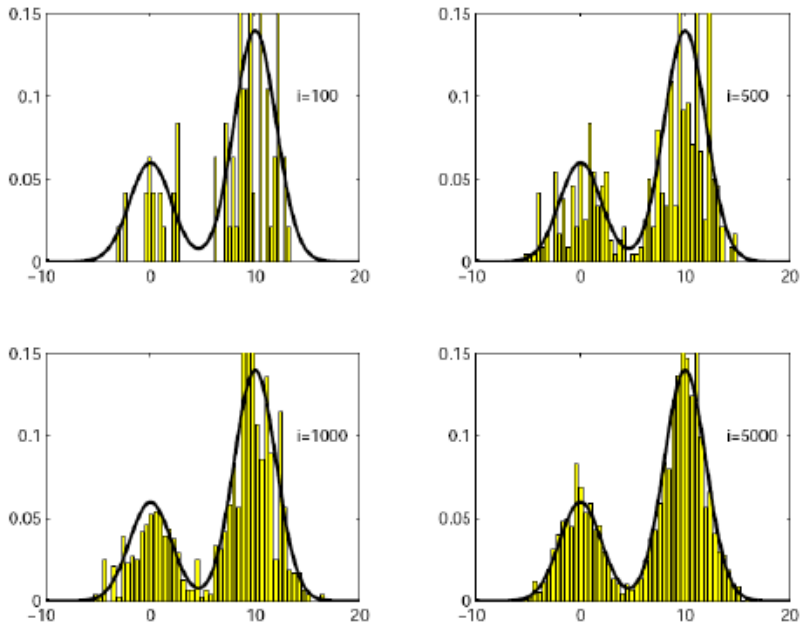
Sample-based Localization (sonar)



Particle Representation of a Distribution

- Weighed set of N particles $\{(x^i, w^i)\}_{i=1}^N$

$$p(x) \approx q(x) := \sum_{i=1}^N w^i \delta(x, x^i)$$



Freie Universität Berlin

Problem to be Solved

- Given a sample-based representation $S_t = \{x_t^1, x_t^2, \ldots, x_t^N\}$

  of $Bel(x_t) = P(x_t \mid z_1, \ldots, z_t, u_1, \ldots, u_t)$

Find a sample-based representation $S_{t+1} = \{x_{t+1}^1, x_{t+1}^2, \ldots, x_{t+1}^N\}$

of $Bel(x_{t+1}) = P(x_{t+1} \mid z_1, \ldots, z_t, \mathbf{z_{t+1}}, u_1, \ldots, \mathbf{u_{t+1}})$

Freie Universität Berlin

Dynamics Update

- Given a sample-based representation $S_t = \{x_t^1, x_t^2, \ldots, x_t^N\}$

  of $Bel(x_t) = P(x_t \mid z_1, \ldots, z_t, u_1, \ldots, u_t)$

  Find a sample-based representation

  of $P(x_{t+1} \mid z_1, \ldots, z_t, u_1, \ldots, \mathbf{u_{t+1}})$

- Solution:
  - For i=1, 2, …, N
    - Sample x^i_{t+1} from $P(X_{t+1} \mid X_t = x^i_t, u_{t+1})$

Freie Universität Berlin

Observation update

- Given a sample-based representation of $\{x_{t+1}^1, x_{t+1}^2, \ldots, x_{t+1}^N\}$

  $P(x_{t+1} \mid z_1, \ldots, z_t)$

  Find a sample-based representation of

  $P(x_{t+1} \mid z_1, \ldots, z_t, z_{t+1}) = C * P(x_{t+1} \mid z_1, \ldots, z_t) * P(z_{t+1} \mid x_{t+1})$

- Solution:
  - For i=1, 2, …, N
    - $w^{(i)}_{t+1} = w^{(i)}_t * P(z_{t+1} \mid X_{t+1} = x^{(i)}_{t+1})$
  - the distribution is represented by the weighted set of samples

$$\{< x_{t+1}^1, w_{t+1}^1 >, < x_{t+1}^2, w_{t+1}^2 >, \ldots, < x_{t+1}^N, w_{t+1}^N >\}$$

Sequential Importance Sampling (SIS) Particle Filter

- Sample $x^1_1, x^2_1, \ldots, x^N_1$ from $P(X_1)$
- Set $w^i_1 = 1$ for all $i=1,\ldots,N$
- For $t=1, 2, \ldots$
  - Dynamics update:
    - For $i=1, 2, \ldots, N$
      - Sample x^i_{t+1} from $P(X_{t+1} \mid X_t = x^i_t, u_{t+1})$
  - Observation update:
    - For $i=1, 2, \ldots, N$
      - $w^i_{t+1} = w^i_t * P(z_{t+1} \mid X_{t+1} = x^i_{t+1})$
- At any time t, the distribution is represented by the weighted set of samples

  $\{ \langle x^i_t, w^i_t \rangle ; i=1,\ldots,N\}$

Freie Universität Berlin

SIS particle filter major issue

- The resulting samples are only weighted by the evidence
- The samples themselves are never affected by the evidence

→ Fails to concentrate particles/computation in the high probability areas of the distribution $P(x_t \mid z_1, \ldots, z_t)$

Sequential Importance Resampling (SIR)

- At any time t, the distribution is represented by the weighted set of samples

  $\{ \langle x_t^i, w_t^i \rangle ; i=1,\ldots,N \}$

→ Sample N times from the set of particles

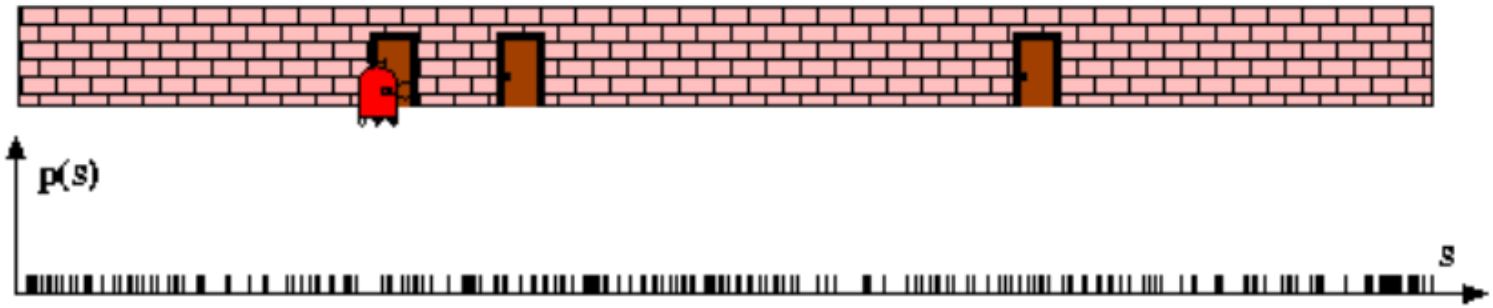
→ The probability of drawing each particle is given by its importance weight

→ More particles/computation focused on the parts of the state space with high probability mass

Sequential Importance Resampling (SIR) Particle Filter

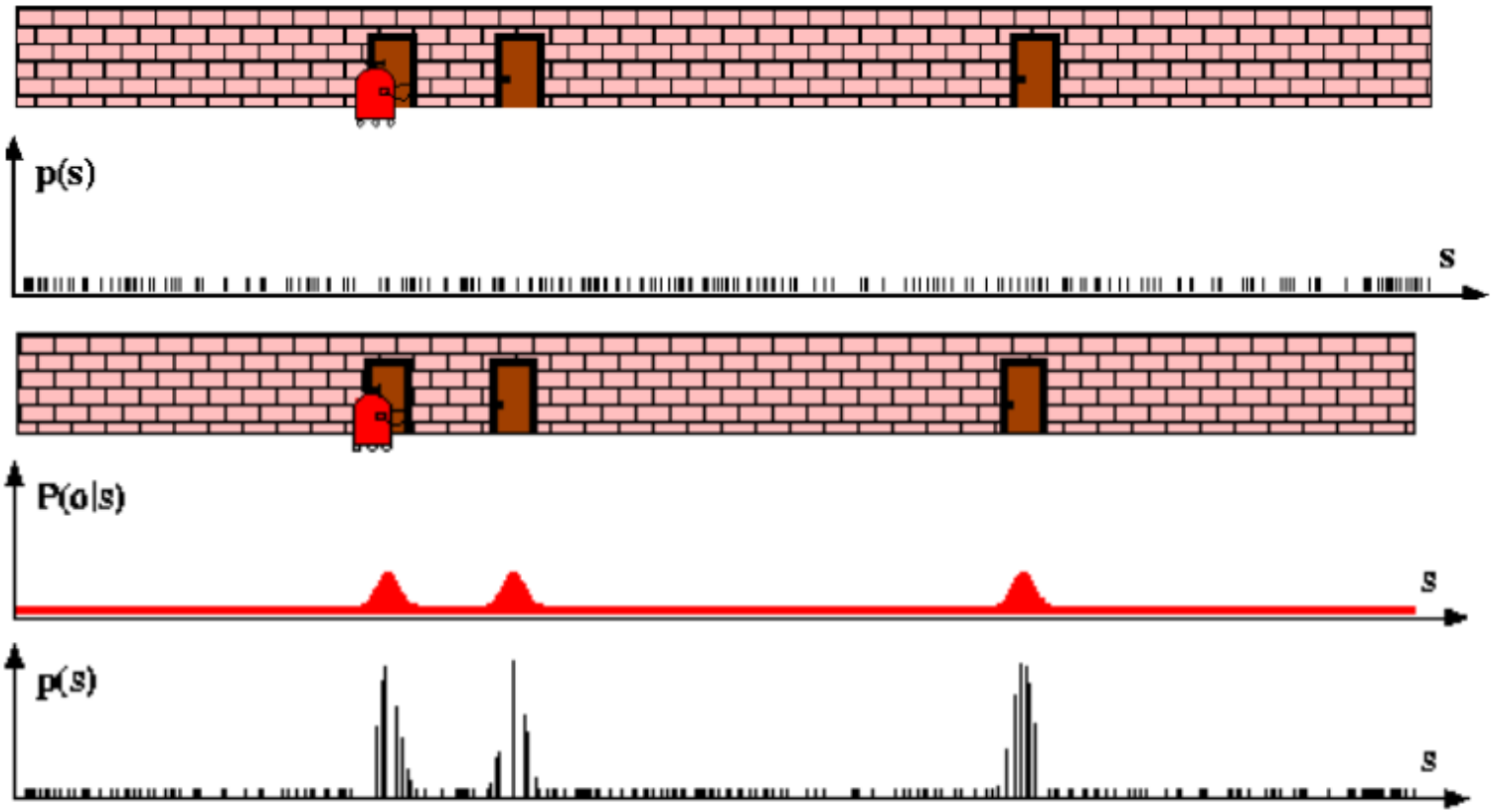
1. Algorithm **particle_filter**(S_{t-1}, u_t , z_t):
2. $S_t = \emptyset, \quad \eta = 0$
3. **For** $i = 1 \ldots n$ *Generate new samples*
4. Sample index $j(i)$ from the discrete distribution given by w_{t-1}
5. Sample x_t^i from $p(x_t \mid x_{t-1}, u_t)$ using $x_{t-1}^{j(i)}$ and u_t
6. $w_t^i = p(z_t \mid x_t^i)$ *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*
11. Return S_t

Particle Filter



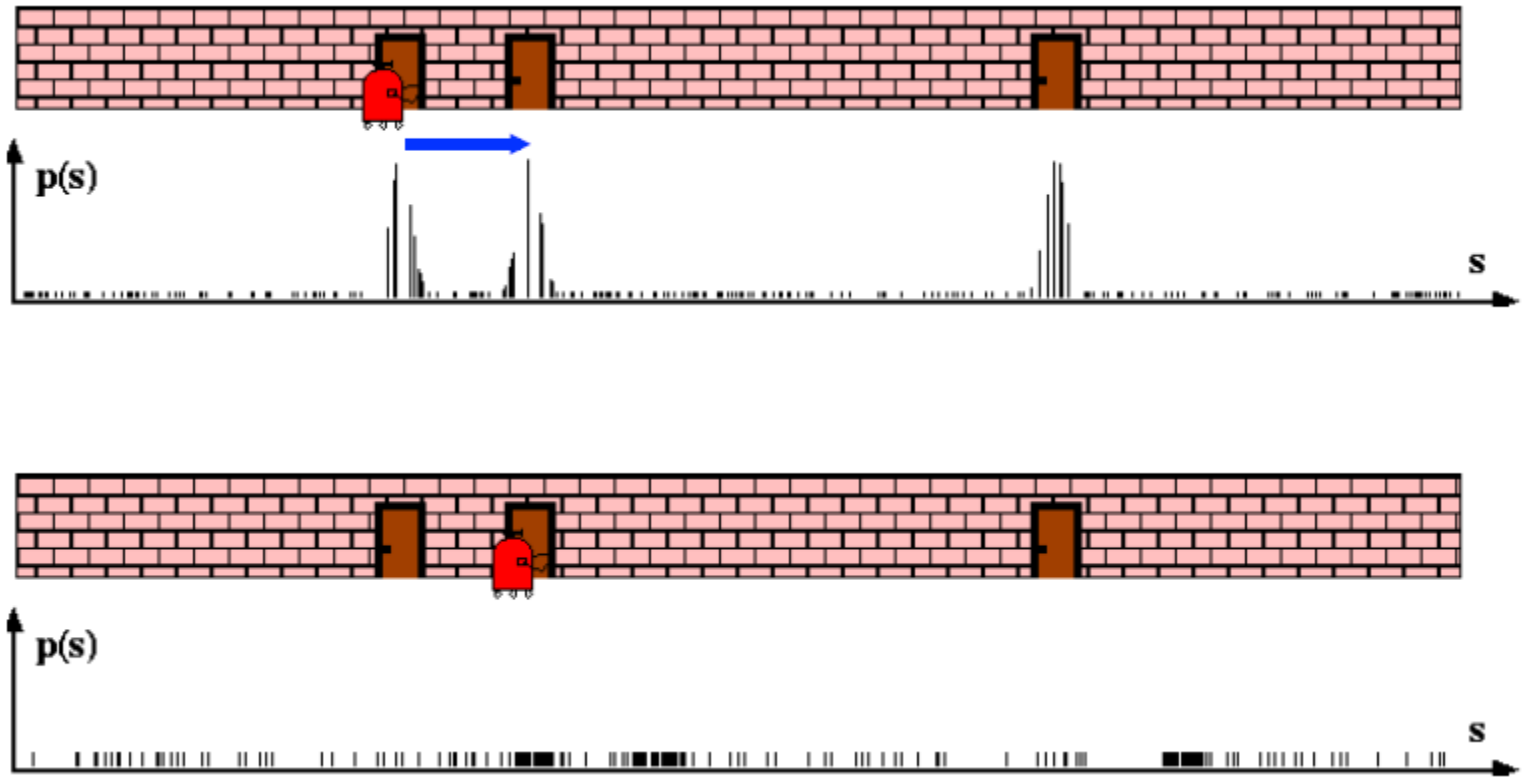
Freie Universität Berlin

Update Step and Weighting



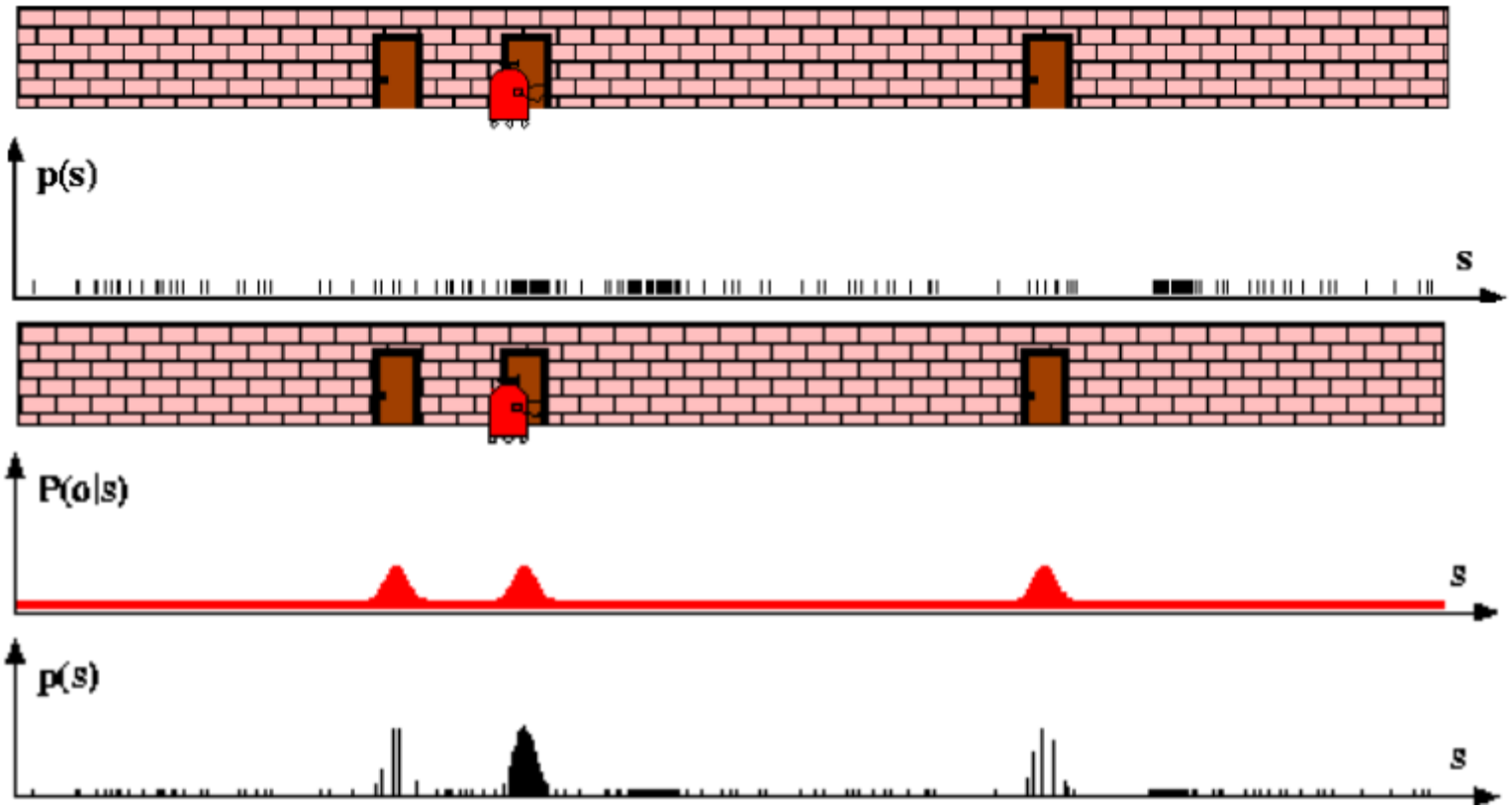
Freie Universität Berlin

Resampling and Robot Motion



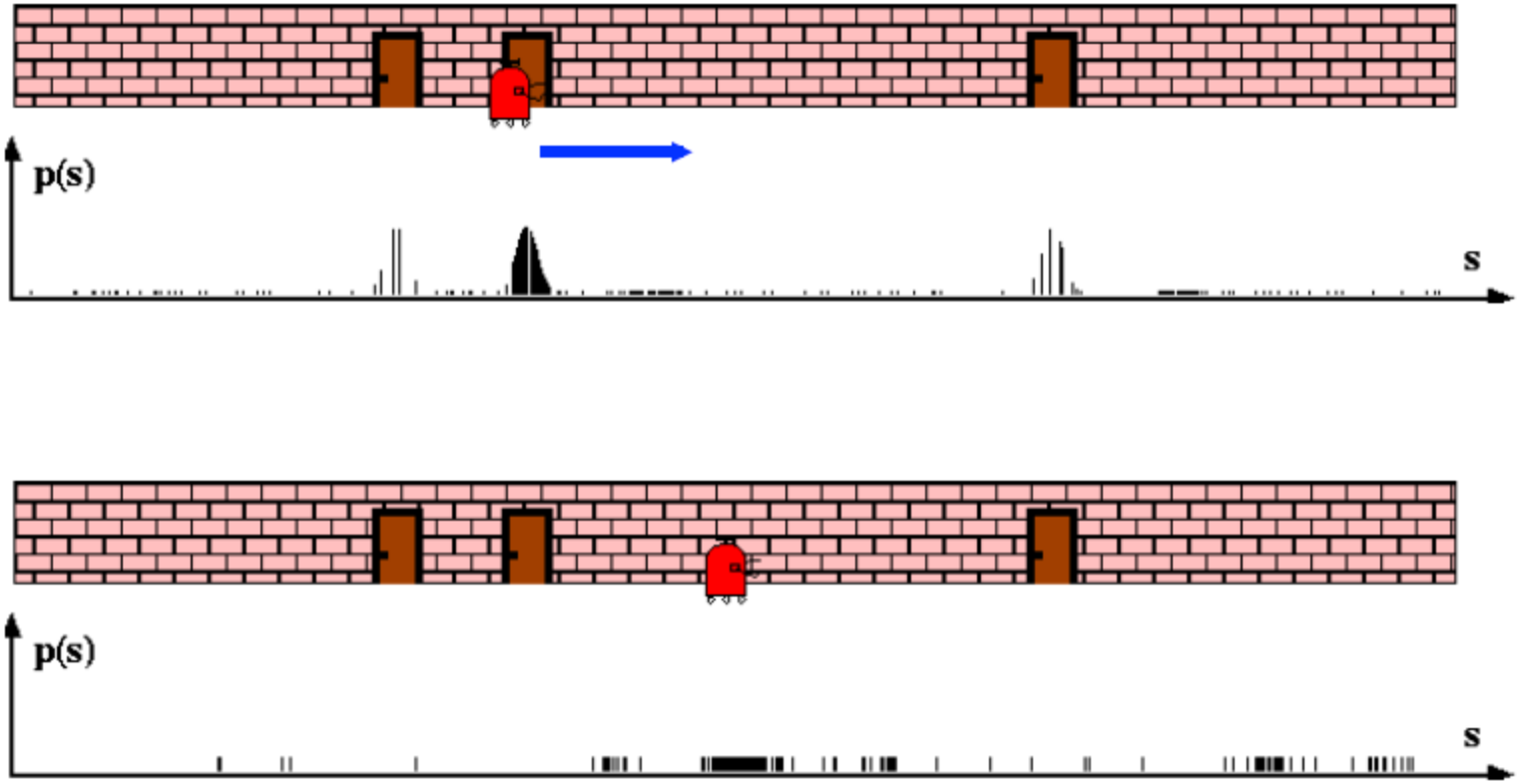
Freie Universität Berlin

Update Step and Weighting



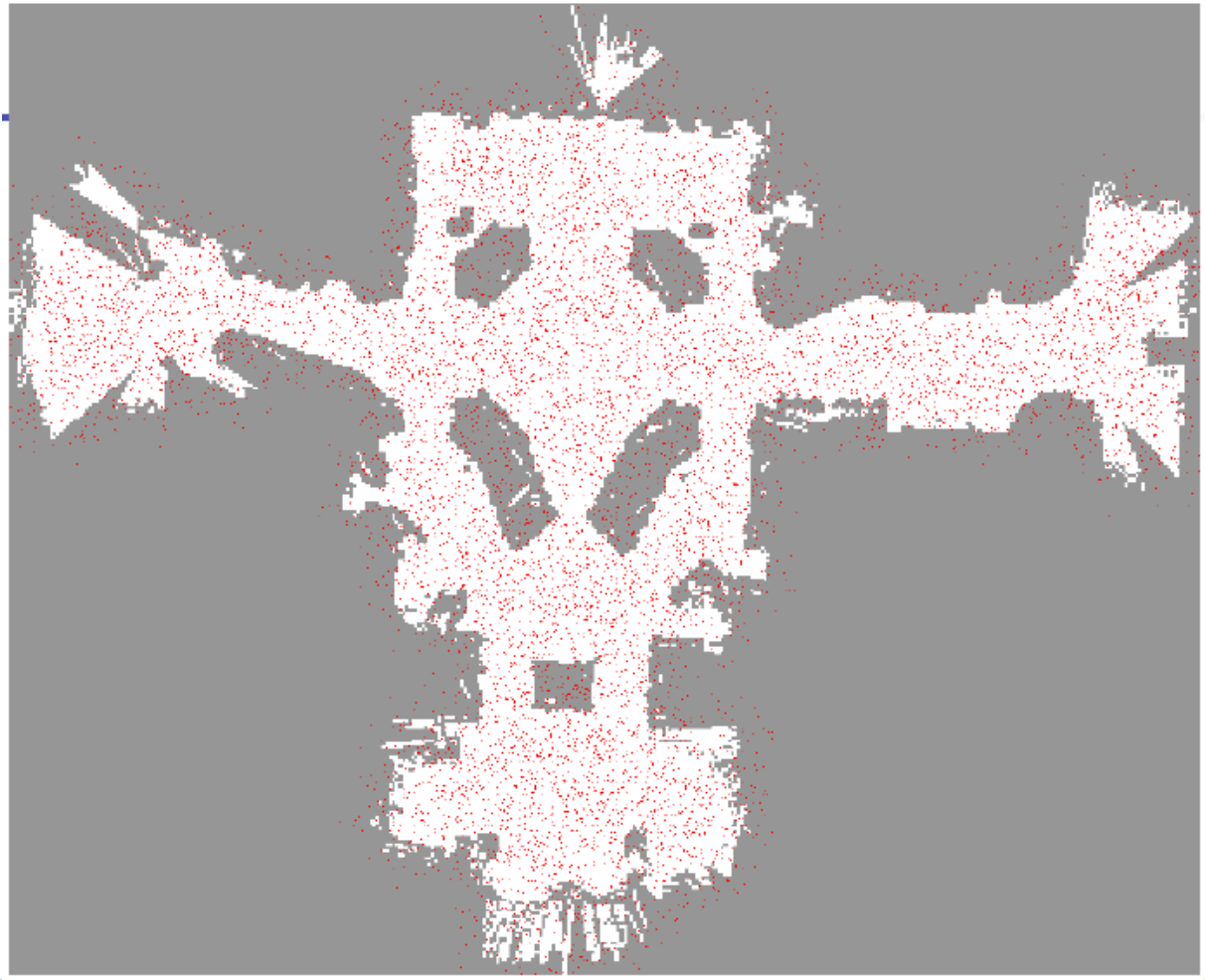
Freie Universität Berlin

Resampling and Robot Motion



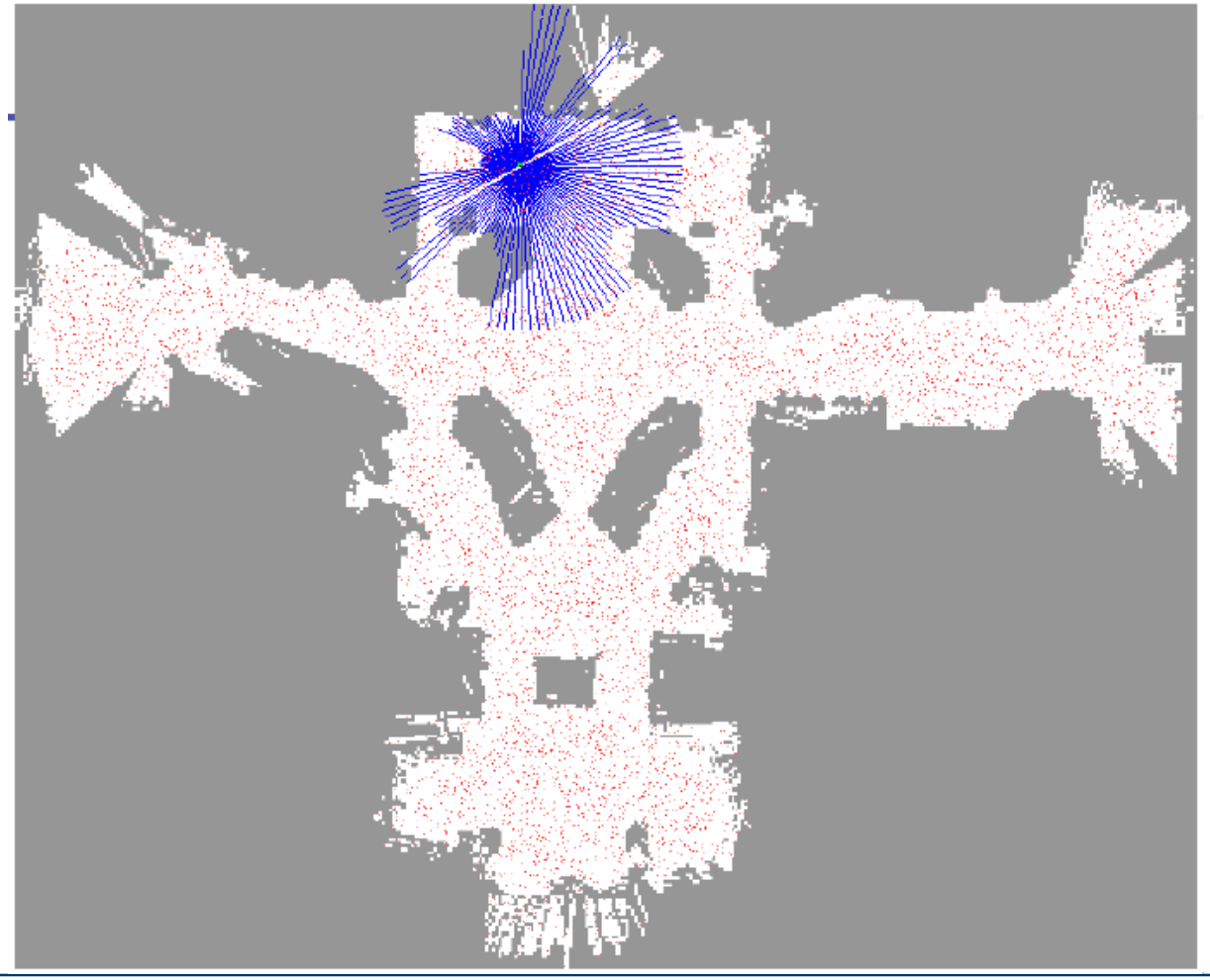
Freie Universität Berlin

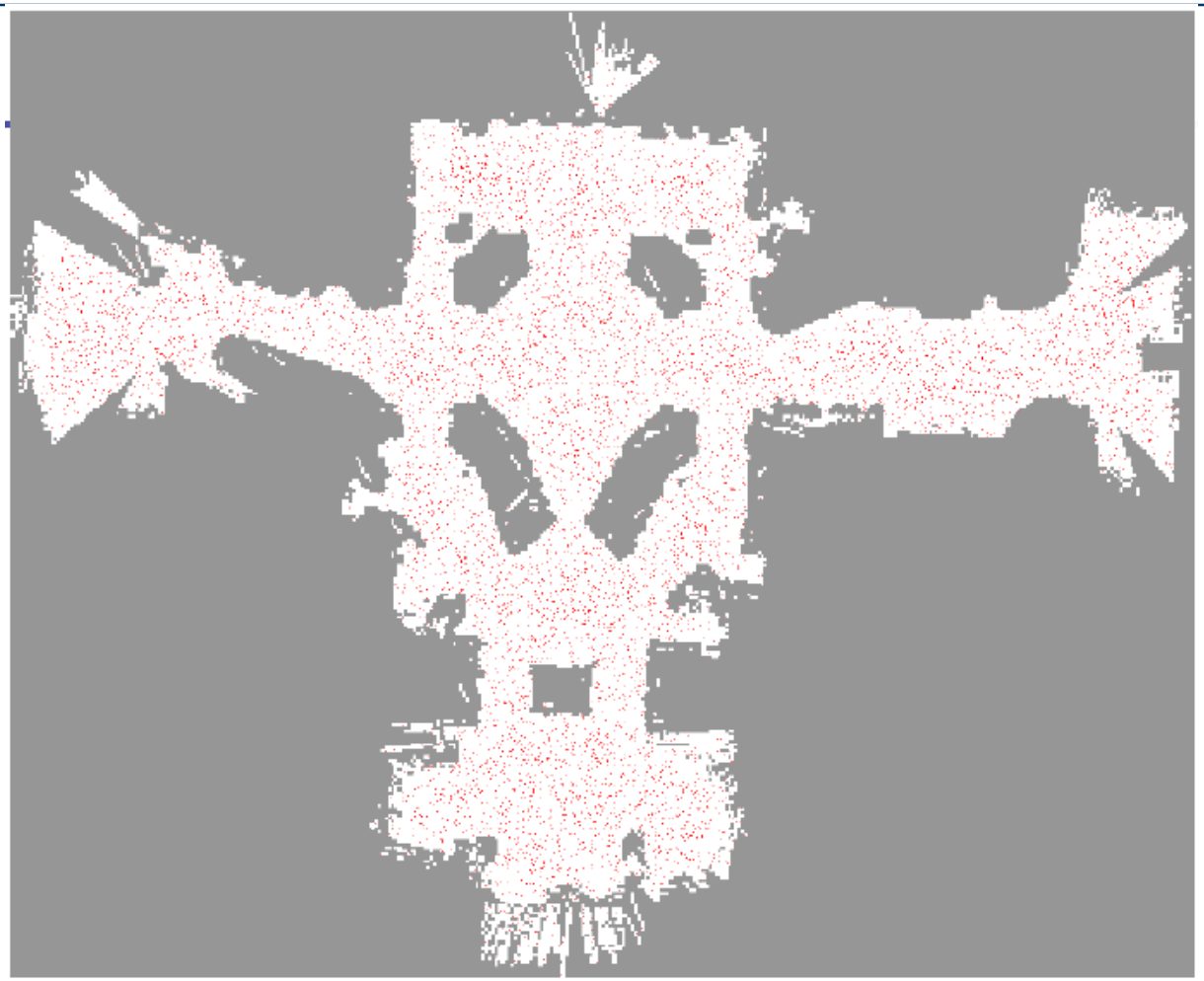
Example

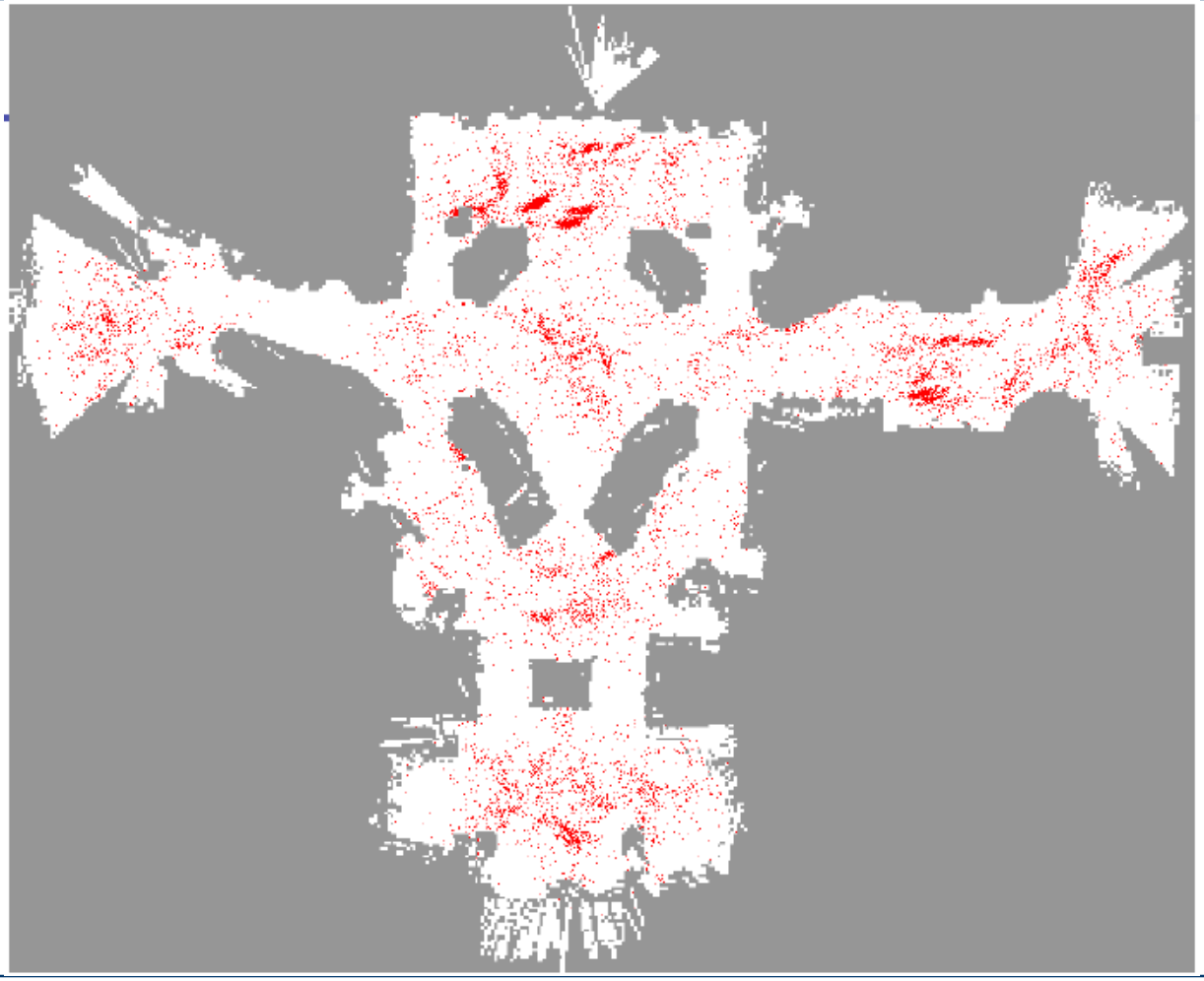


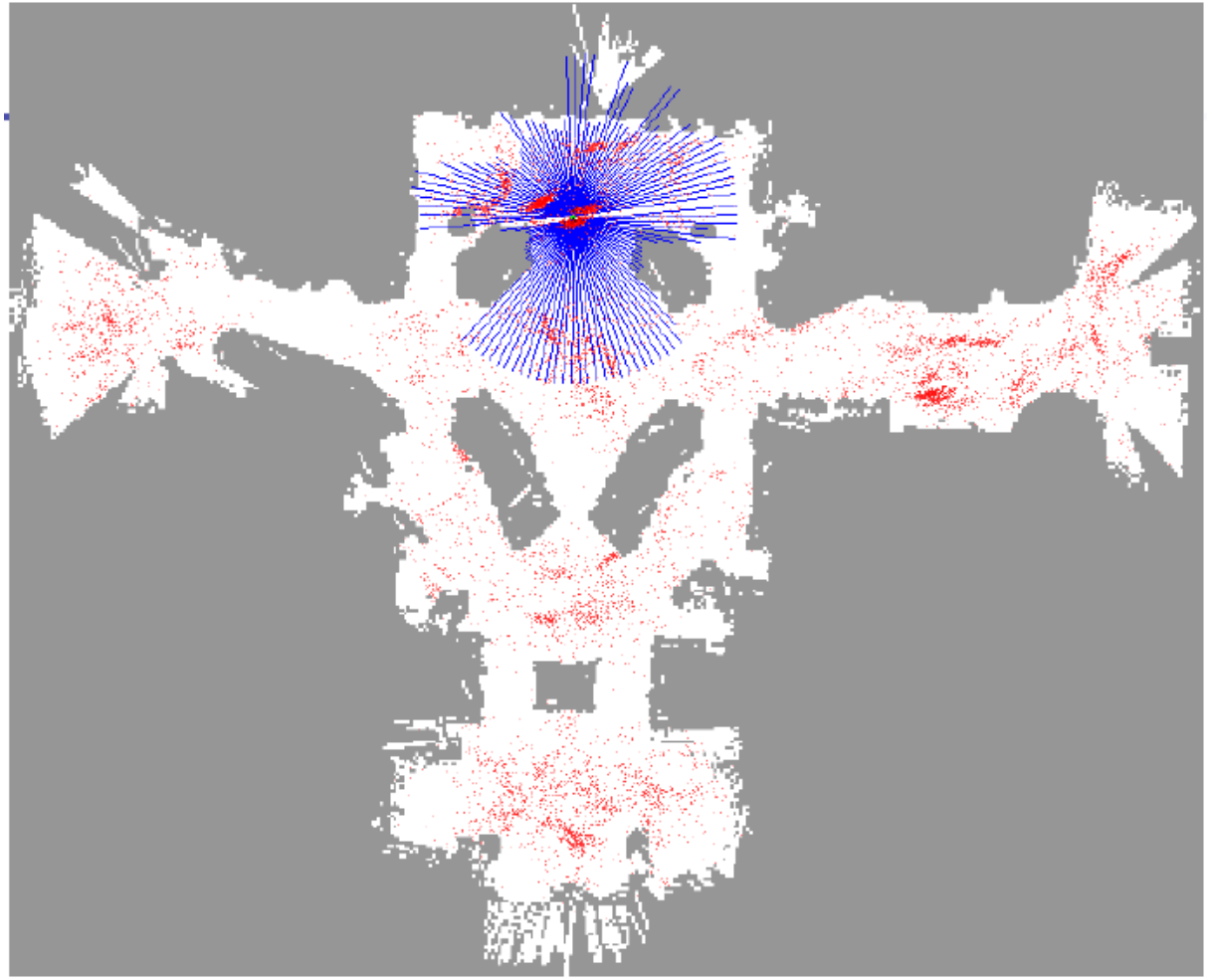
Freie Universität Berlin

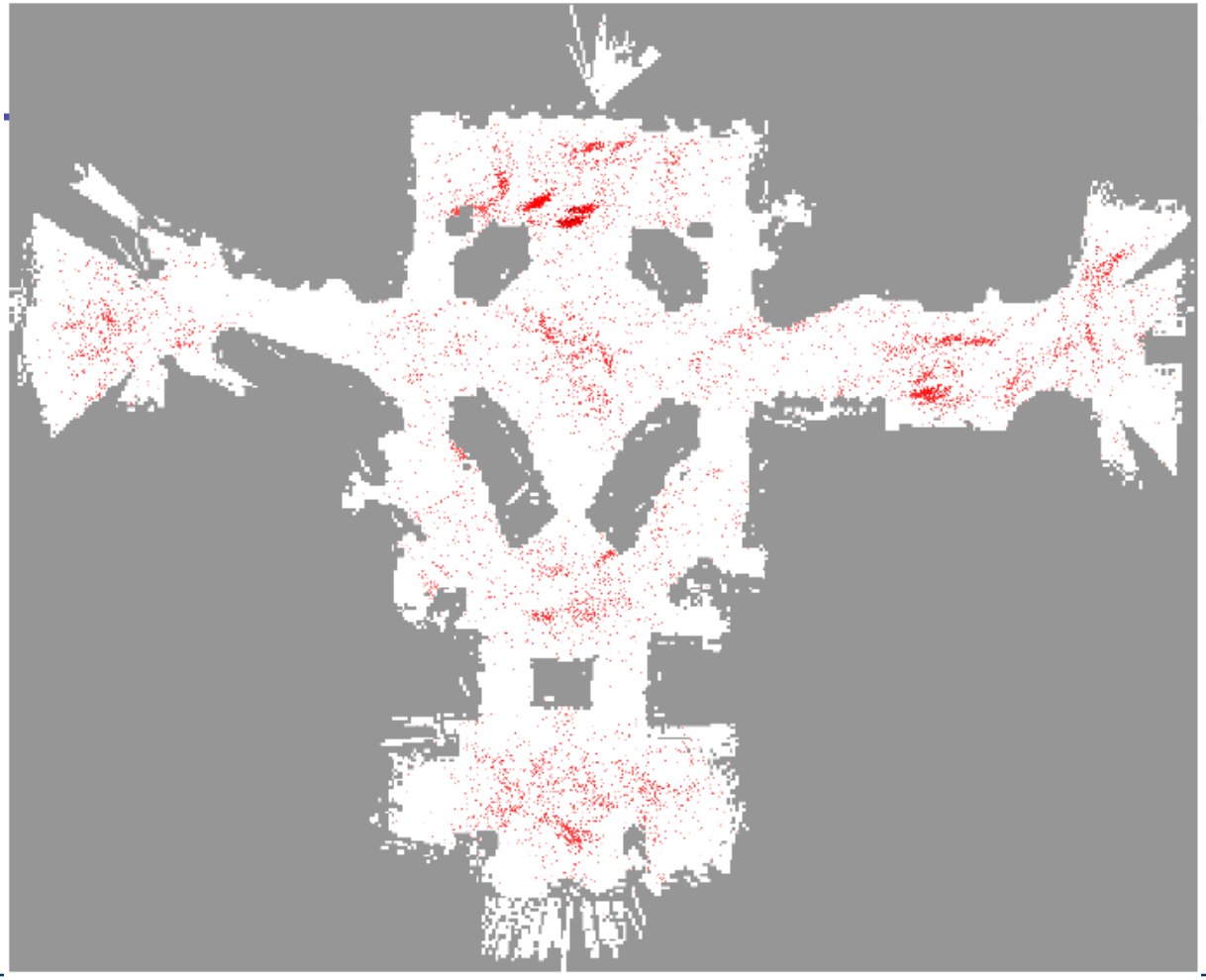
Example

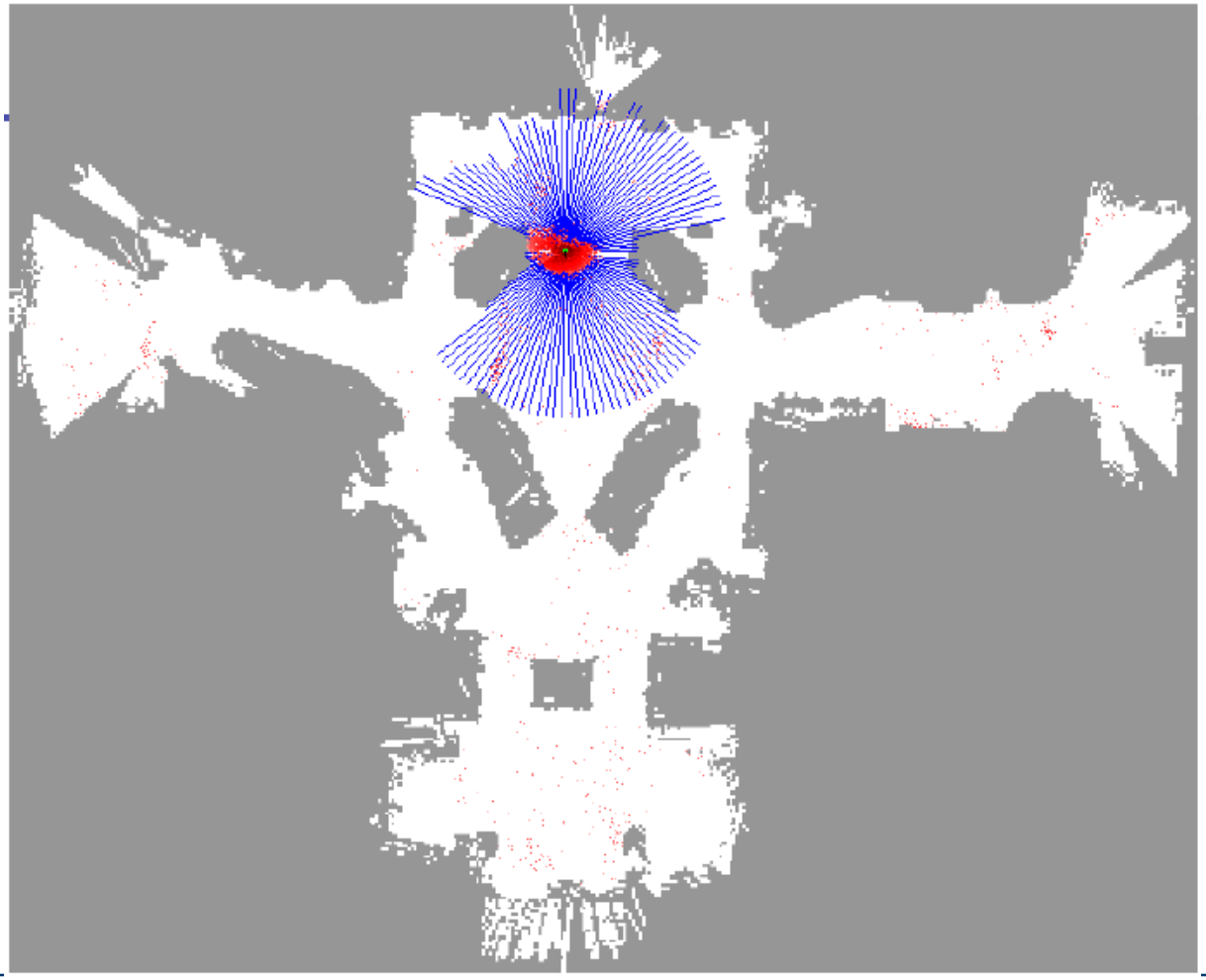


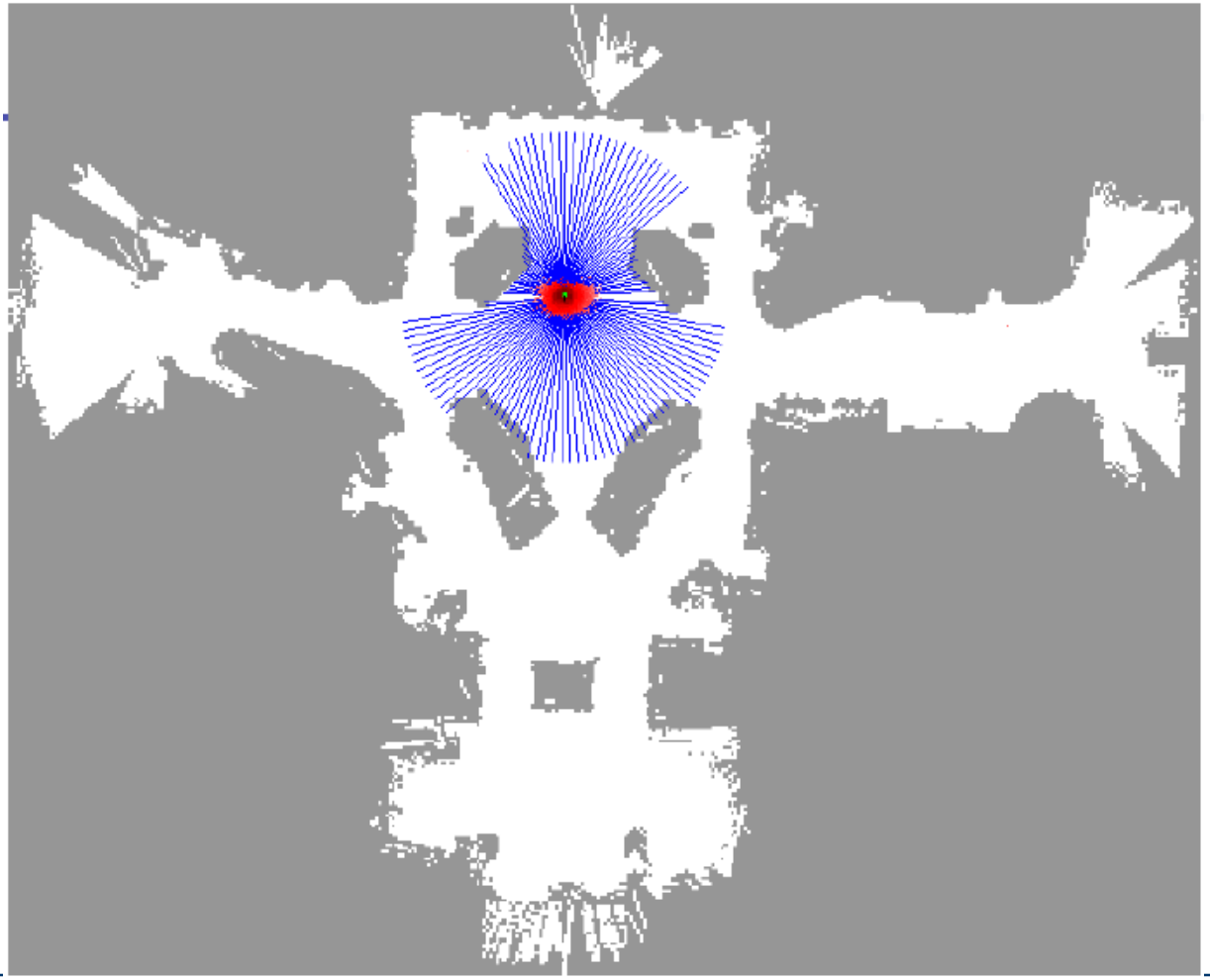


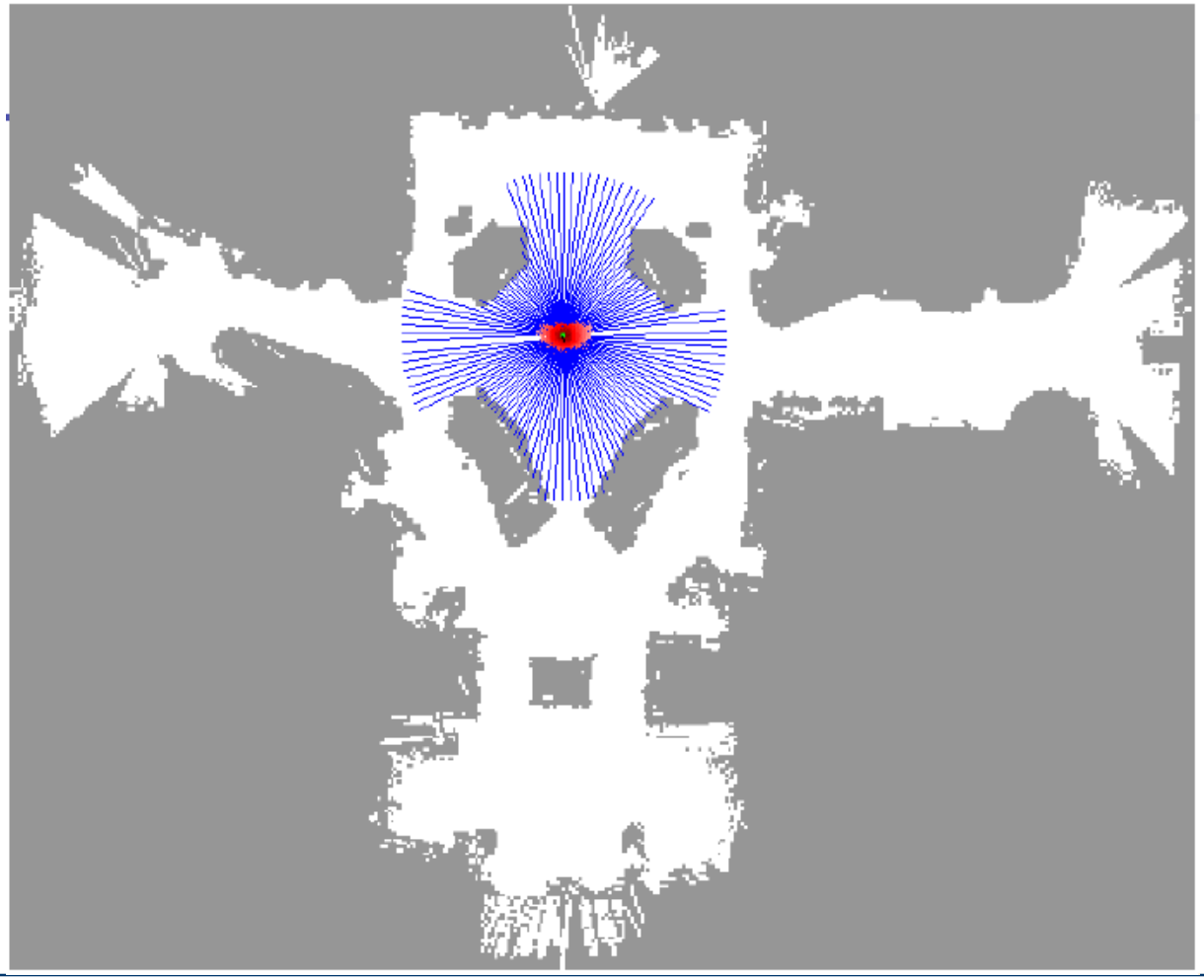












Noise Dominated by Motion Model

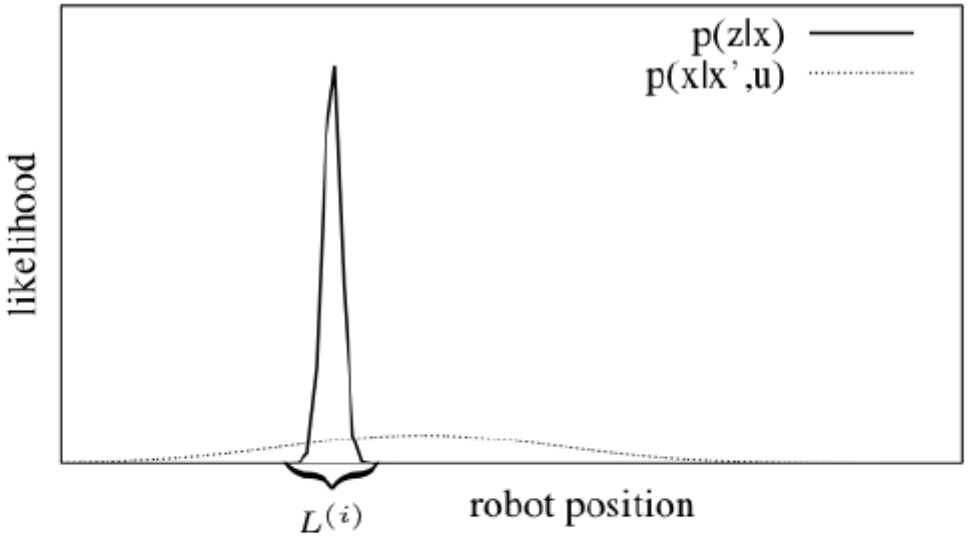


Fig. 1. Within the interval $L^{(i)}$ the product of both functions is dominated by the observation likelihood in case an accurate sensor is used. [Grisetti, Stachniss, Burgard, T-RO2006]

→ Most particles get (near) zero weights and are lost.

Resampling

- Consider running a particle filter for a system with deterministic dynamics and no sensors
- **Problem:**
  - While no information is obtained that favors one particle over another, due to resampling some particles will disappear and after running sufficiently long with very high probability all particles will have become identical.
  - On the surface it might look like the particle filter has uniquely determined the state.
- Resampling induces loss of diversity. The variance of the particles decreases, the variance of the particle set as an estimator of the true belief increases.

Resampling Solution I

- Effective sample size:

$$\hat{N}_{\text{eff}} = \frac{1}{\sum_{i=1}^{N} (\tilde{w}_k^{(i)})^2}$$

Normalized weights

- Example:
  - All weights = 1/N → Effective sample size = N
  - All weights = 0, except for one weight = 1 → Effective sample size = 1

- Idea: resample only when effective sampling size is low

Resampling Solution II: Low Variance Sampling

- M = number of particles

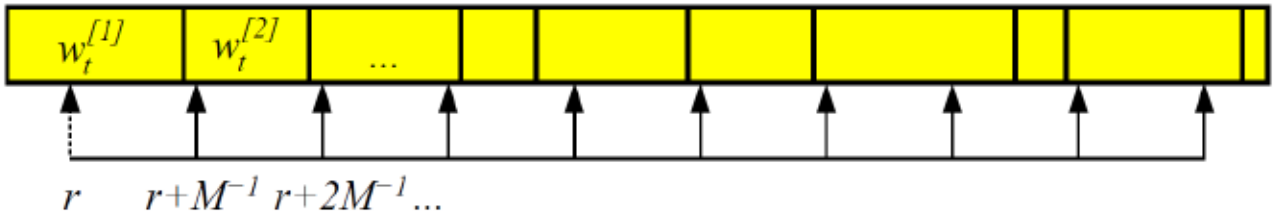


Figure 4.7 Principle of the low variance resampling procedure. We choose a random number r and then select those particles that correspond to $u = r + (m-1) \cdot M^{-1}$ where $m = 1, \ldots, M$.

  - $r \in [0, 1/M]$

- Advantages:
  - More systematic coverage of space of samples
  - If all samples have same importance weight, no samples are lost
  - Lower computational complexity

Resampling Solution III

- Loss of diversity caused by resampling from a discrete distribution

- Solution: "regularization"
  - Consider the particles to represent a continuous density
  - Sample from the continuous density
  - E.g., given (1-D) particles $\{x^{(1)}, x^{(2)}, \ldots, x^{(K)}\}$

    sample from the density: $p(x) = \sum_{k=1}^{K} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-x^{(k)})^2}{\sigma^2}}$

Particle Deprivation

- = when there are no particles in the vicinity of the correct state
- Occurs as the result of the variance in random sampling. An unlucky series of random numbers can wipe out all particles near the true state. This has non-zero probability to happen at each time → will happen eventually.
- Popular solution: add a small number of randomly generated particles when resampling.
  - Advantages: reduces particle deprivation, simplicity.
  - Con: incorrect posterior estimate even in the limit of infinitely many particles.
  - Other benefit: initialization at time 0 might not have gotten anything near the true state, and not even near a state that over time could have evolved to be close to true state now; adding random samples will cut out particles that were not very consistent with past evidence anyway, and instead gives a new chance at getting close the true state.

Particle Deprivation: How Many Particles to Add?

- Simplest: Fixed number.
- Better way:
  - Monitor the probability of sensor measurements

$$p(z_t|z_{1:t-1}, u_{1:t}, m)$$

which can be approximated by:

$$\frac{1}{N}\sum_{i=1}^{N} w_t^{(i)}$$

  - Average estimate over multiple time-steps and compare to typical values when having reasonable state estimates. If low, inject random particles.

Noise-free Sensors

- Consider a measurement obtained with a noise-free sensor, e.g., a noise-free laser-range finder---issue?
  - All particles would end up with weight zero, as it is very unlikely to have had a particle matching the measurement exactly.
- Solutions:
  - Artificially inflate amount of noise in sensors
  - Better proposal distribution (e.g., optimal sequential proposal)

Freie Universität Berlin

Literature

- Sebastian Thrun, Wolfram Burgard, Dieter Fox, *Probabilistic Robotics*, MIT Press, 2005
- Pieter Abbeel, Lecture Slides, UC Berkeley, 2012
- Marc Toussaint, Lecture Slides, FU Berlin, 2011