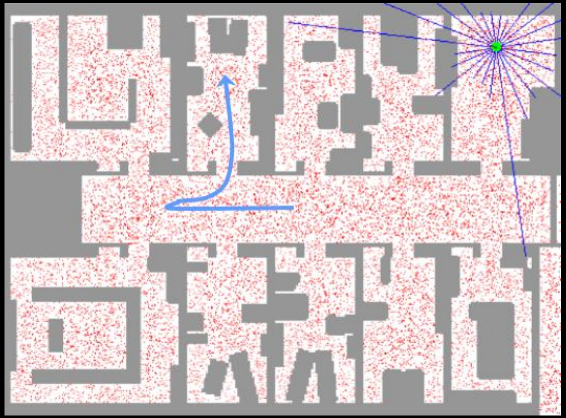
CS4495/6495
Introduction to Computer Vision

7C-L3 *Particle filters for localization*



Localization: A robot sensing problem

- Assume a robot **knows** a 3D map of its world.
- It has noisy depth sensors but whose sensing uncertainty is ***known***.
- It moves from frame to frame.
- How well can it know where it is in (x, y, θ) ?

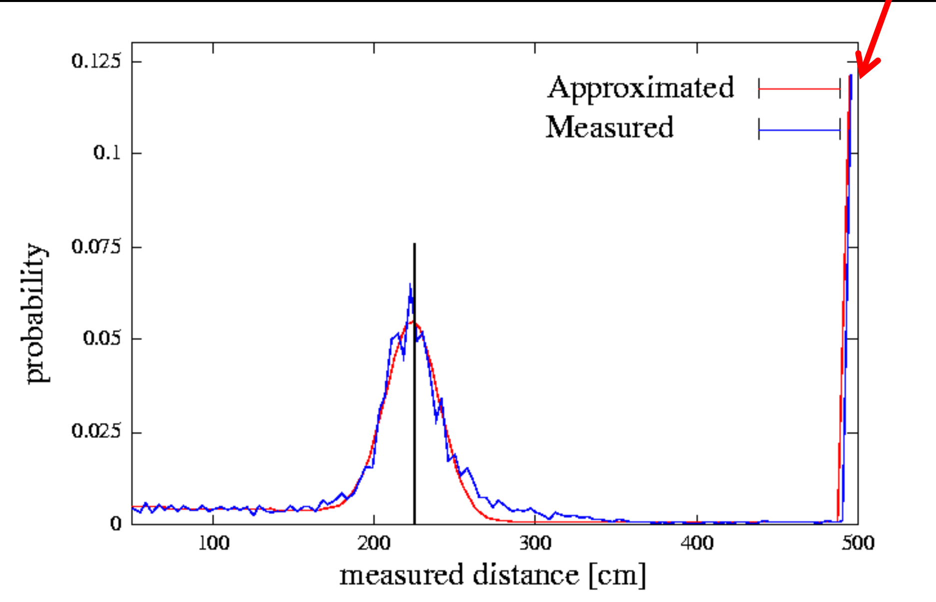
Bayes Filters: Framework

Given

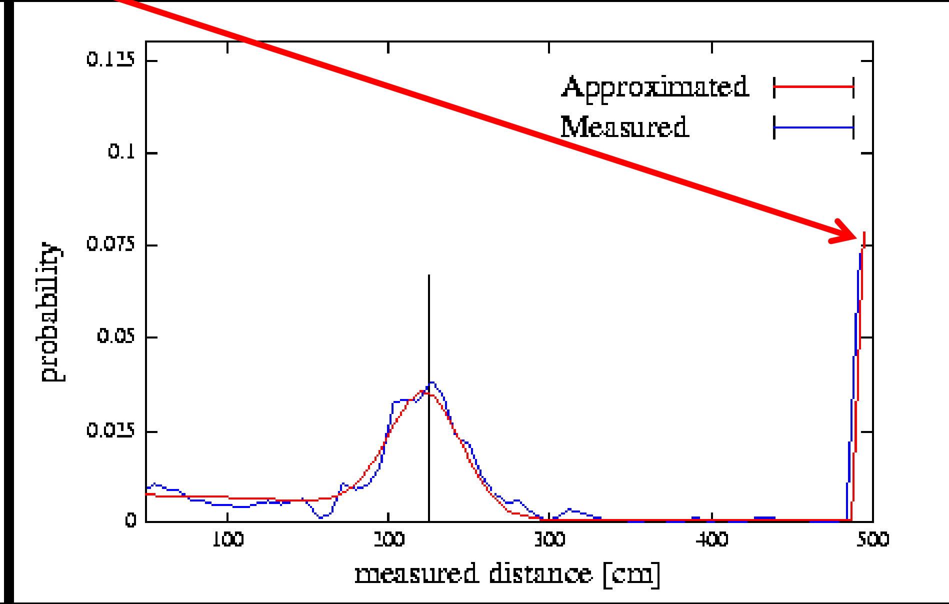
1. Prior probability of the system state $p(x)$
2. Action (dynamical system) model:
$$p(x_t|u_{t-1}, x_{t-1})$$
3. Sensor model (likelihood) $p(z|x)$
4. Stream of observations z and action data u:
$$data_t = \{u_1, z_2 \ldots, u_{t-1}, z_t\}$$

Proximity (depth) Sensor Model

No return!

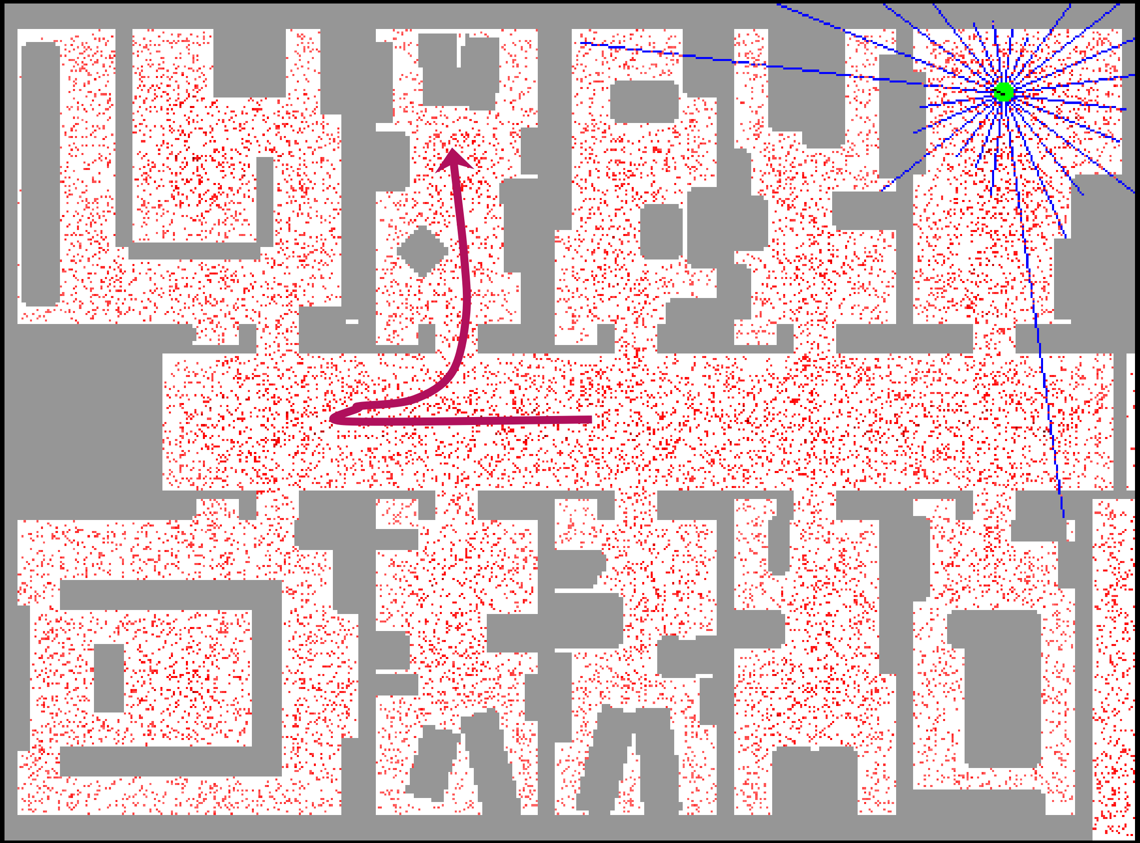


Laser sensor



Sonar sensor

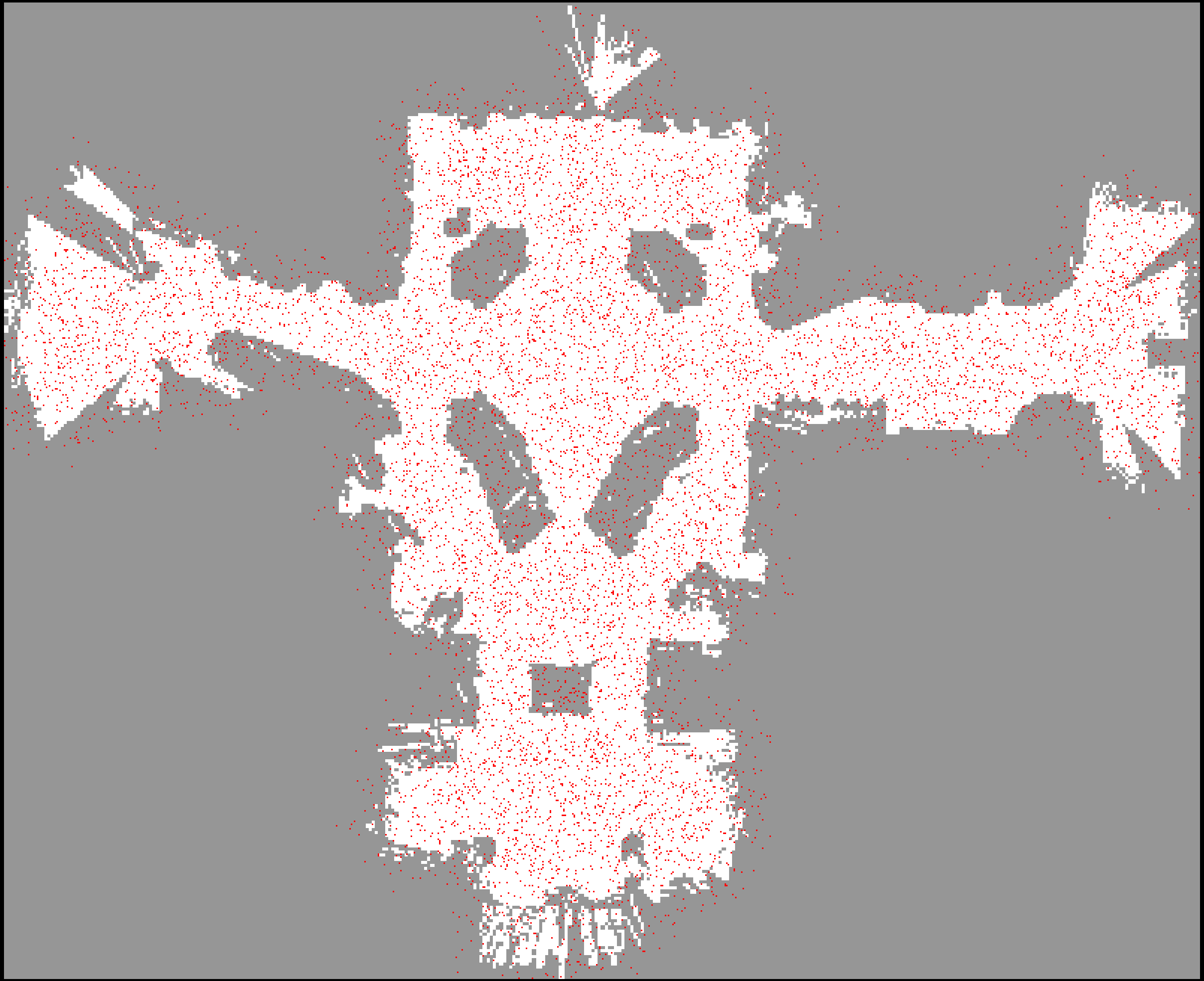
Sample-based Localization (sonar)



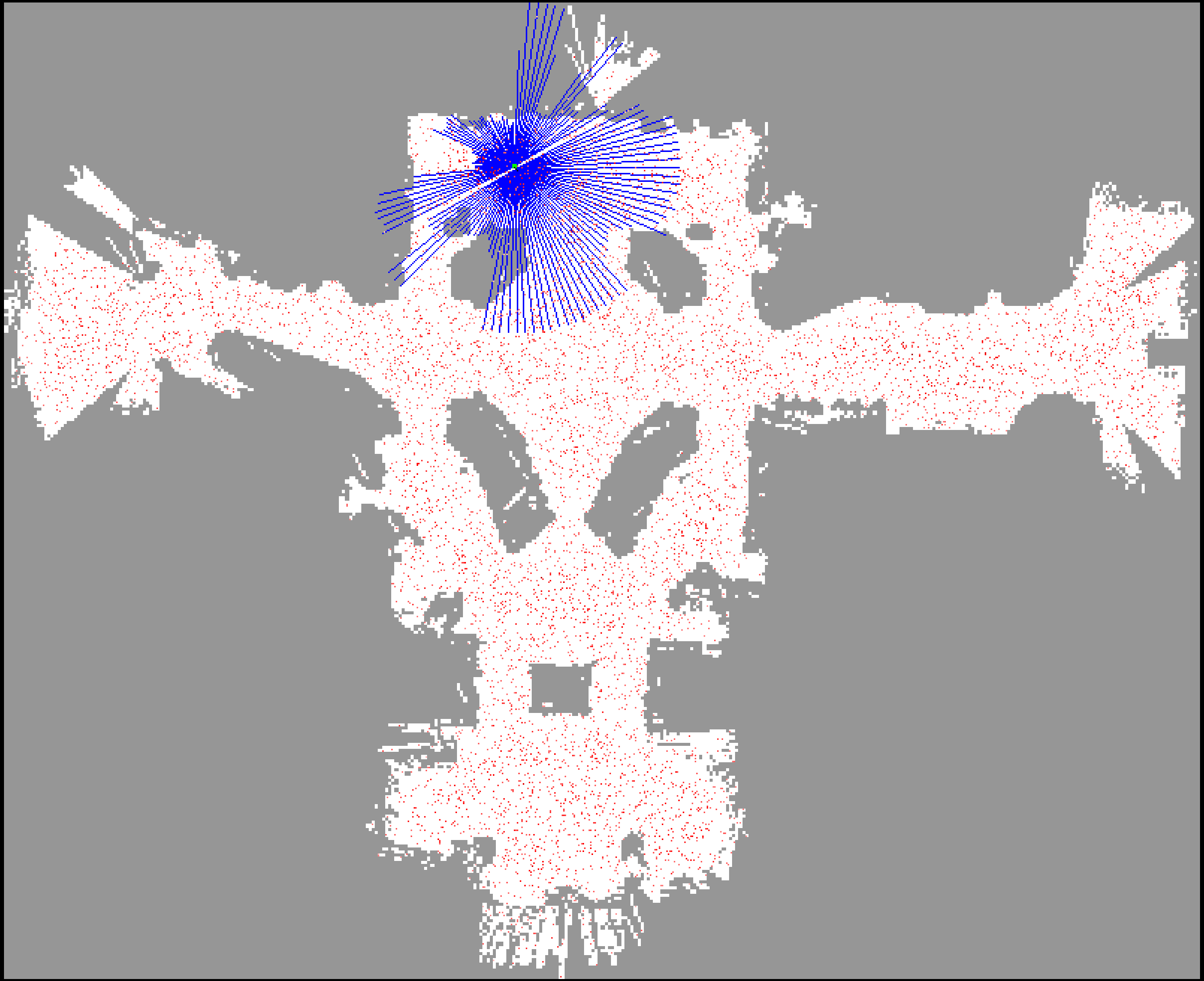
Sonar-based Localization Example

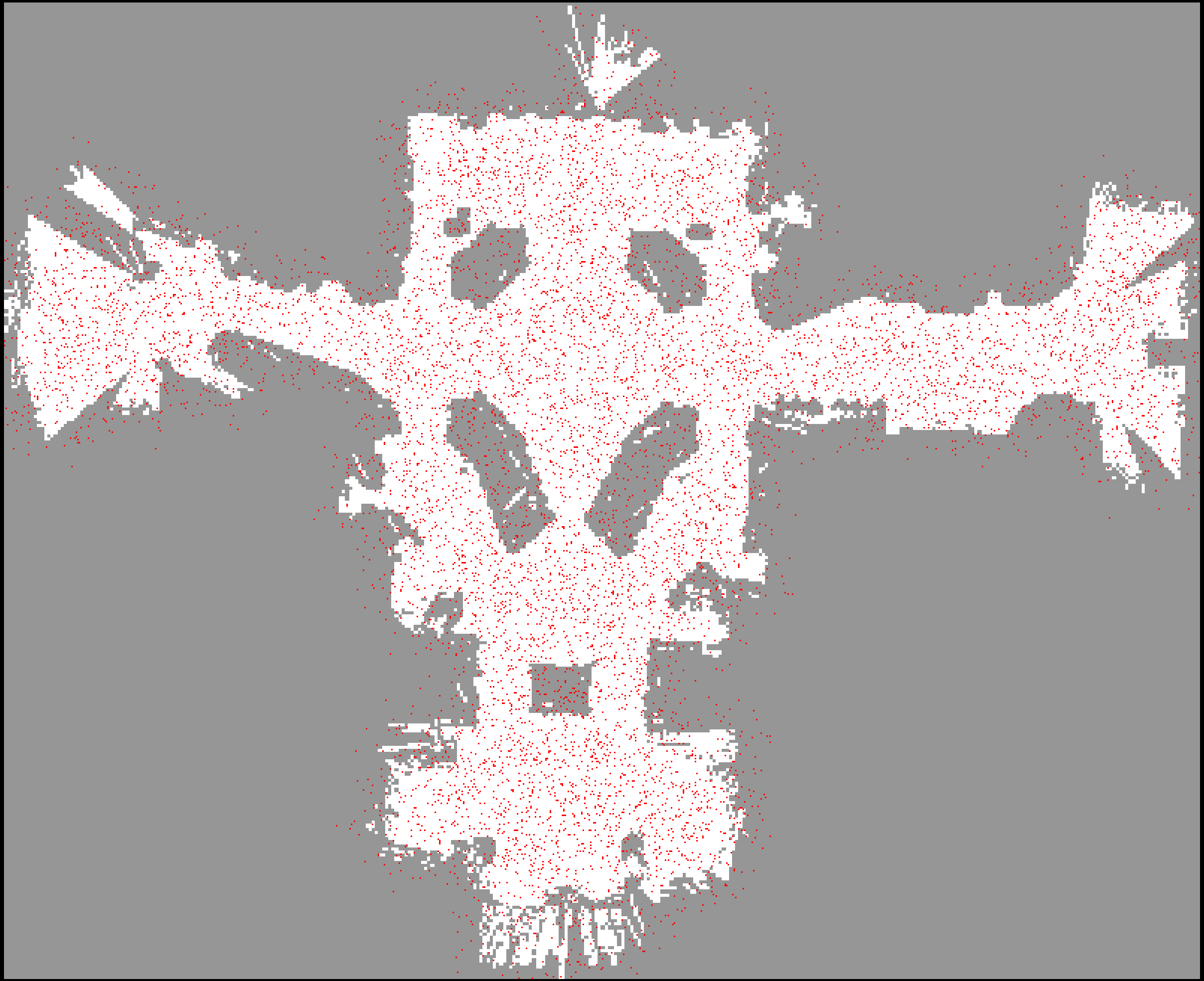
At the *Smithsonian Museum of American History*…

Initial prediction

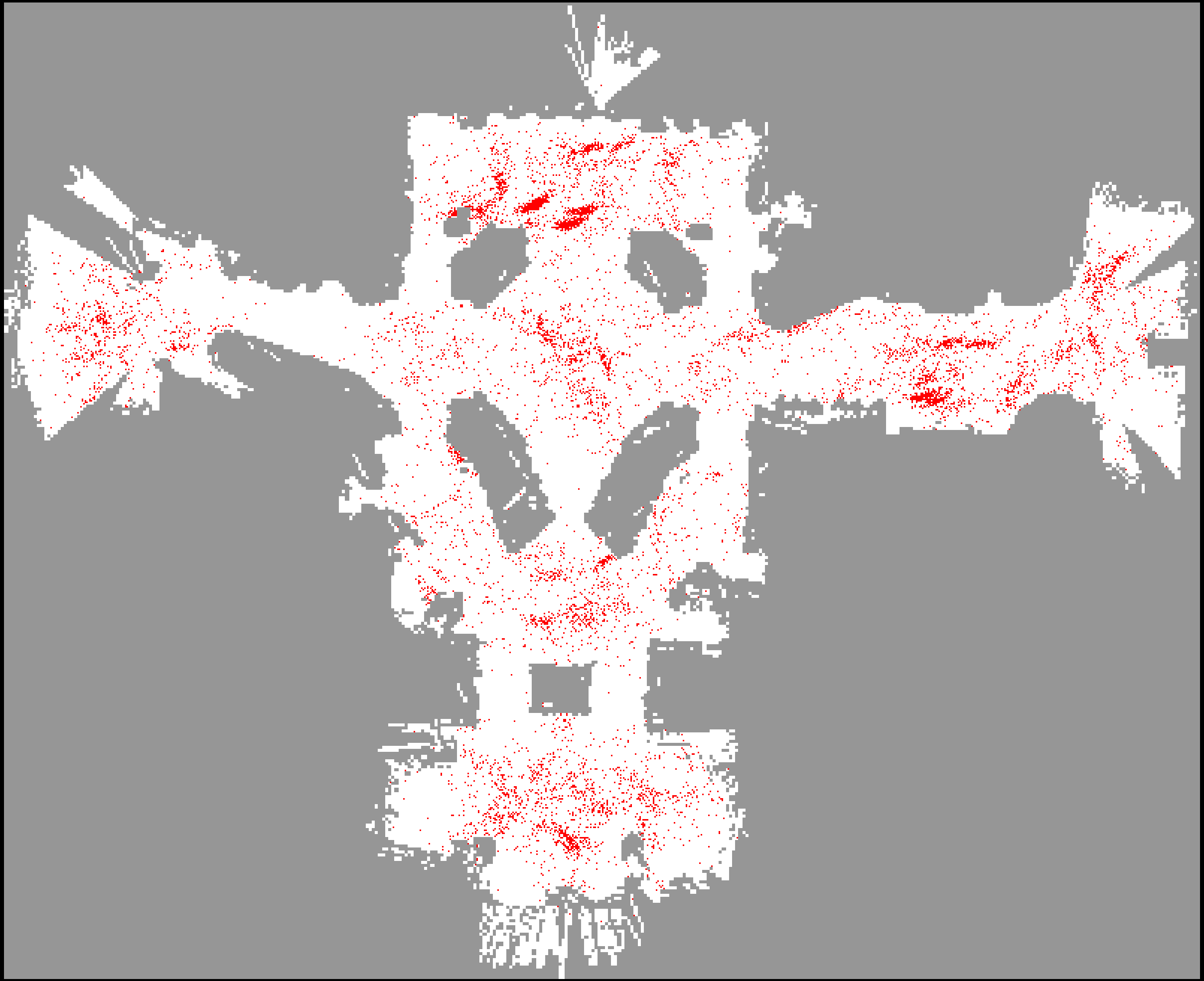


First observation

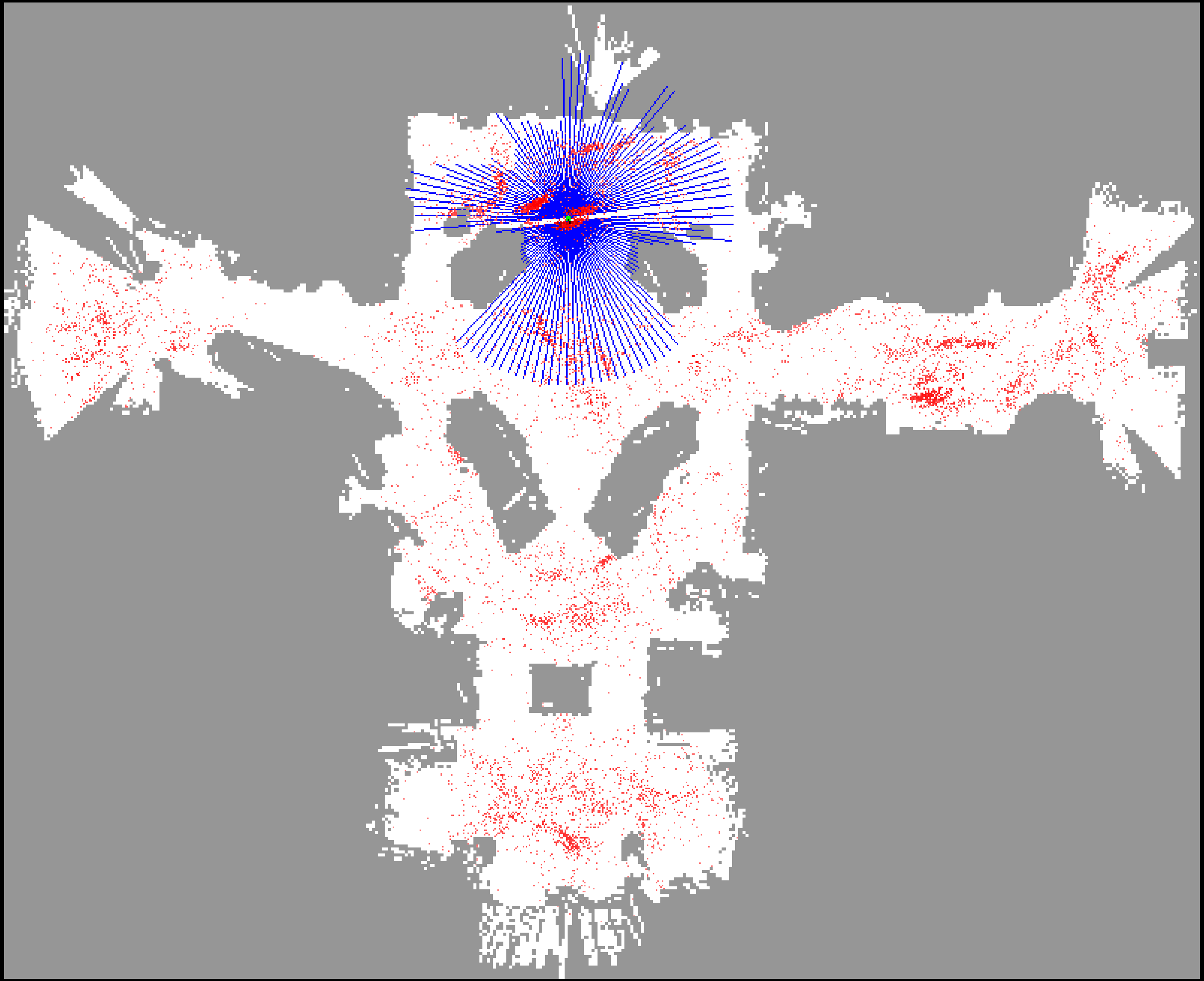


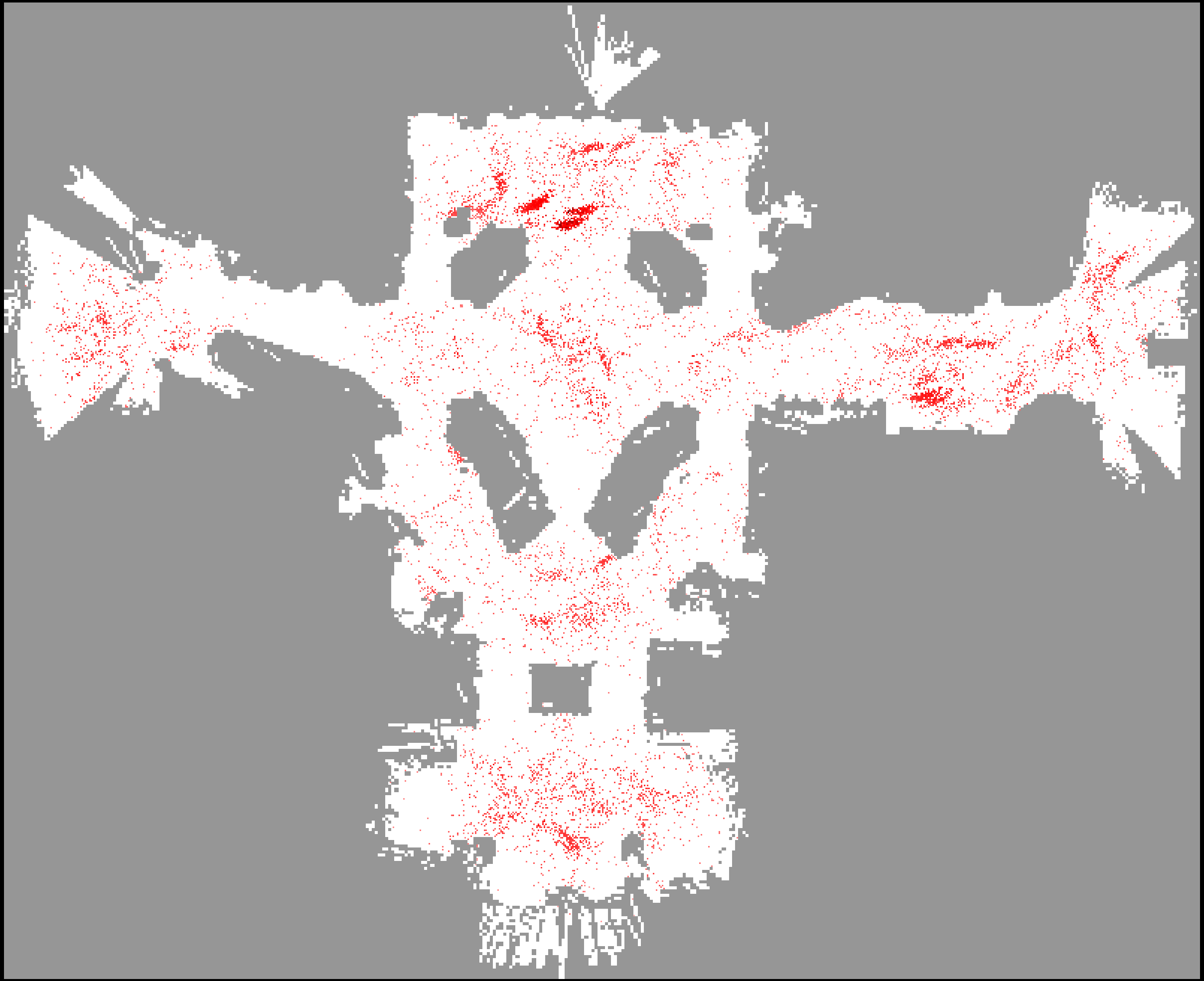


Corrected prediction

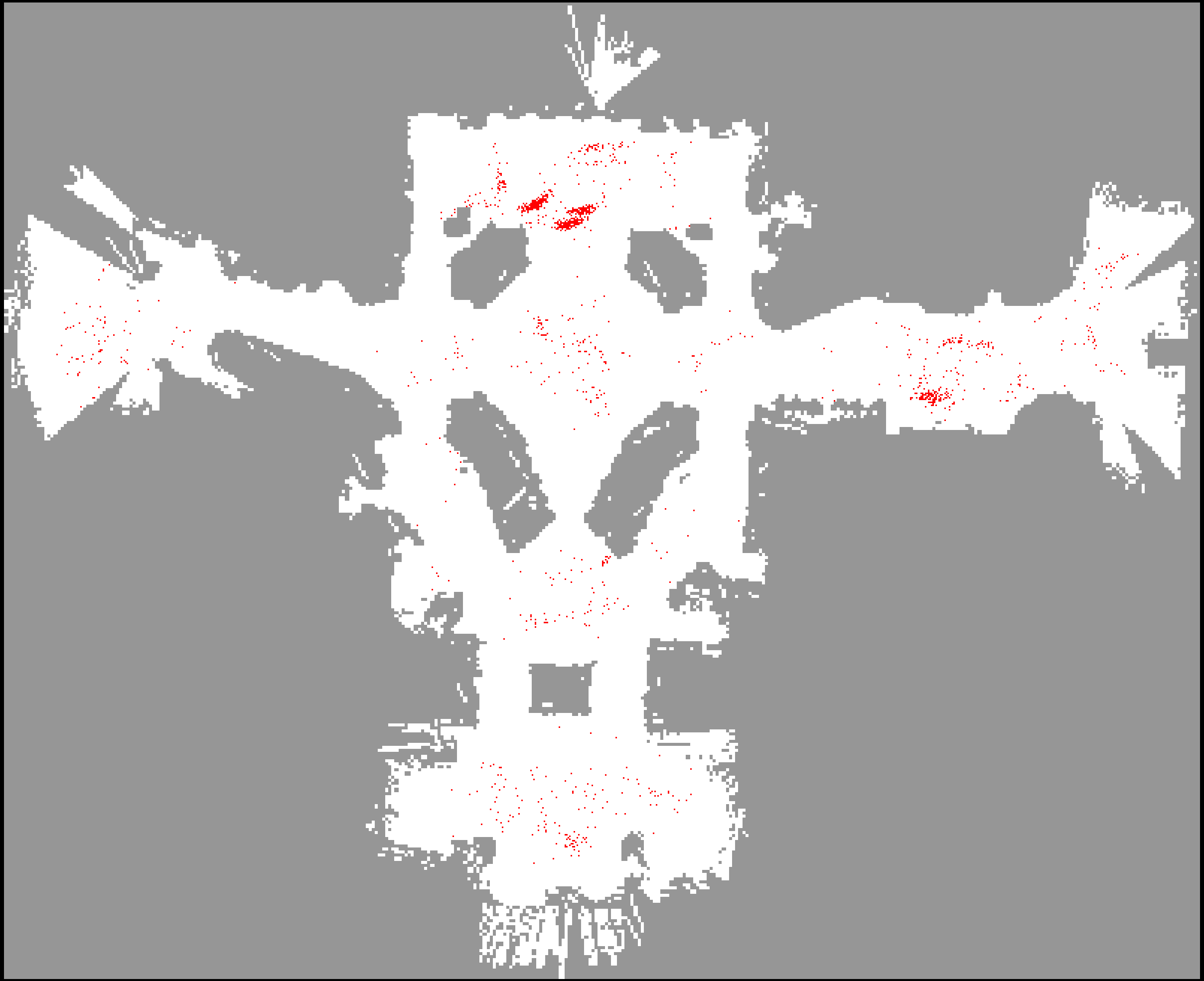


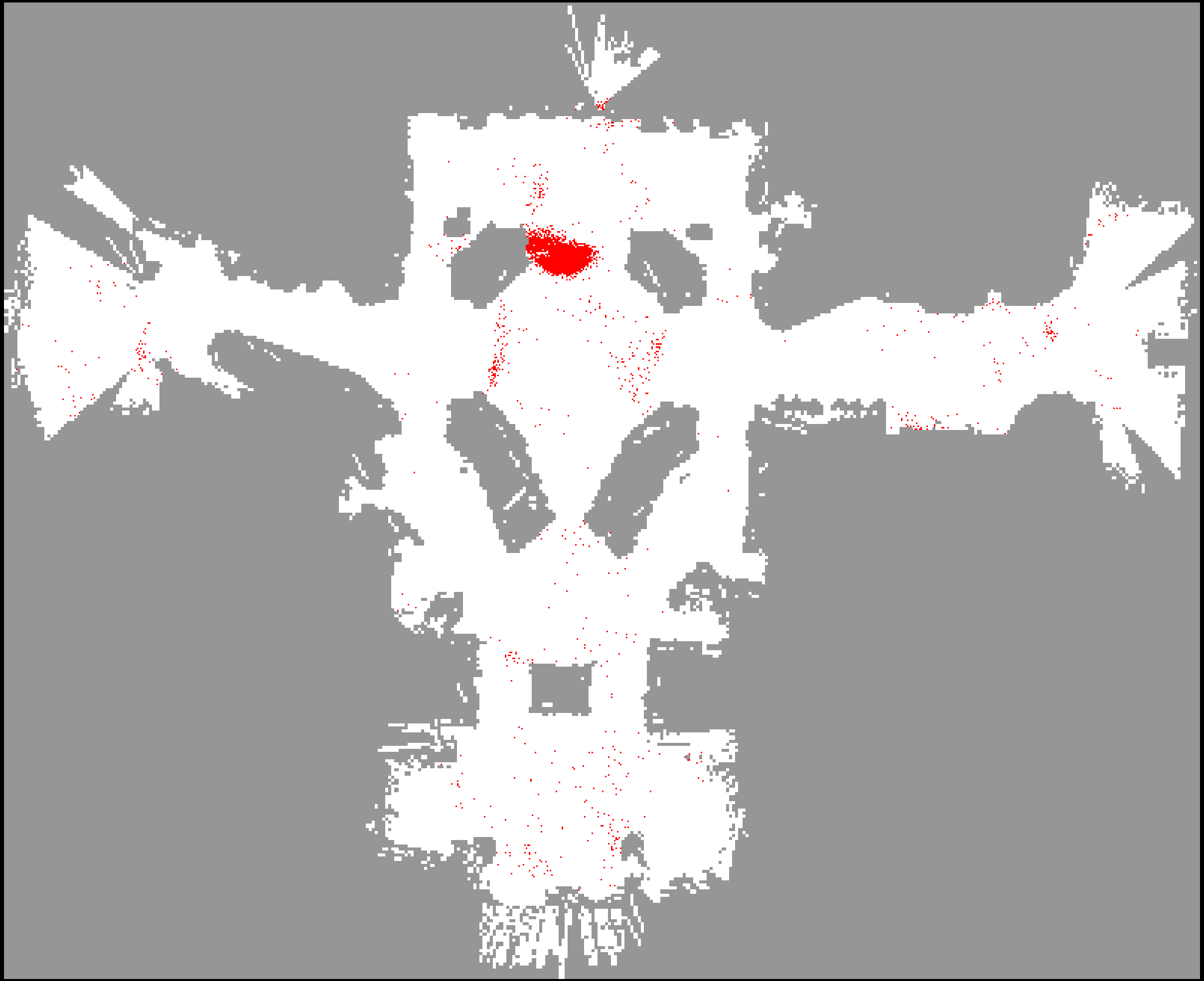
Next observation



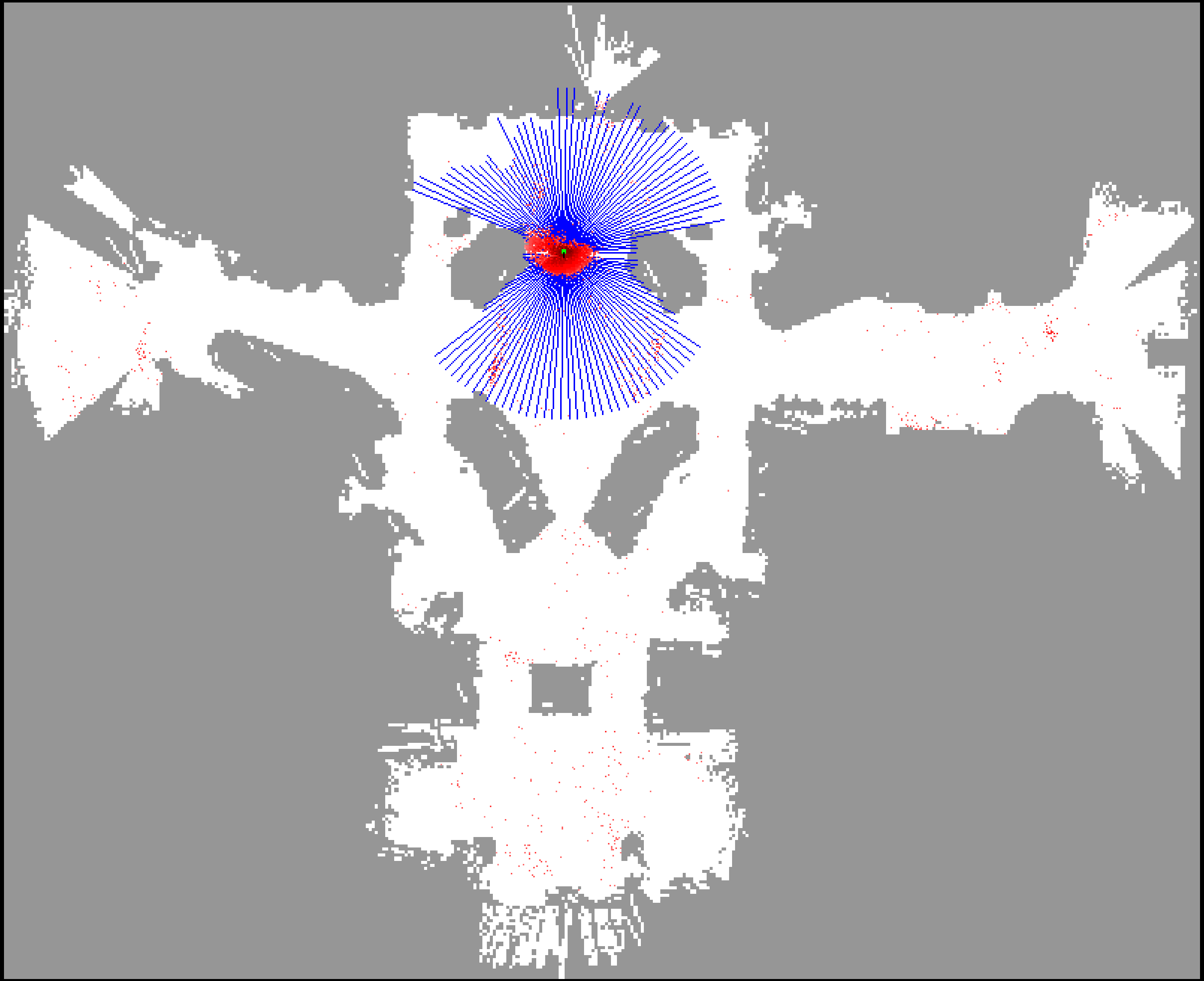


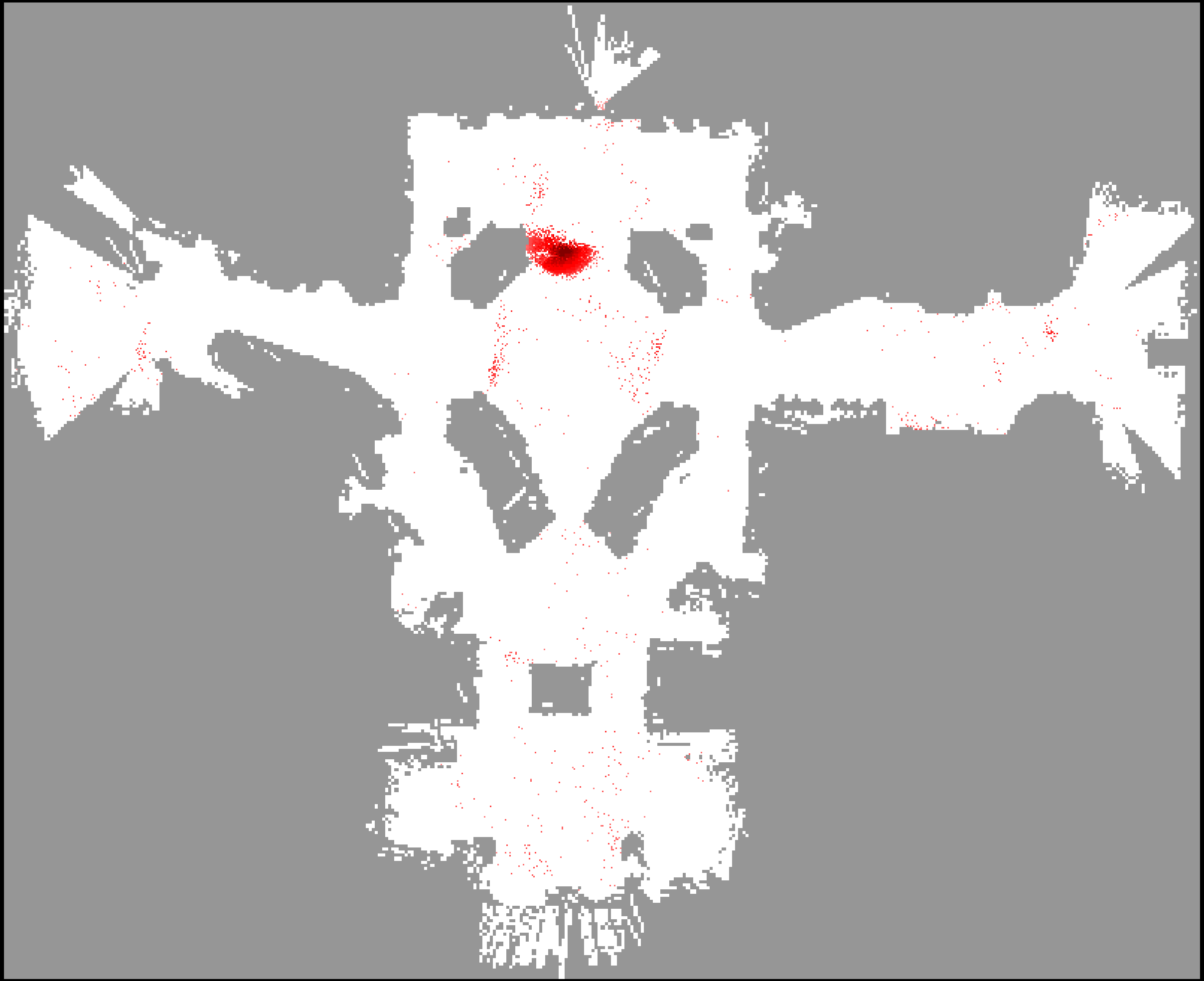
Next correction

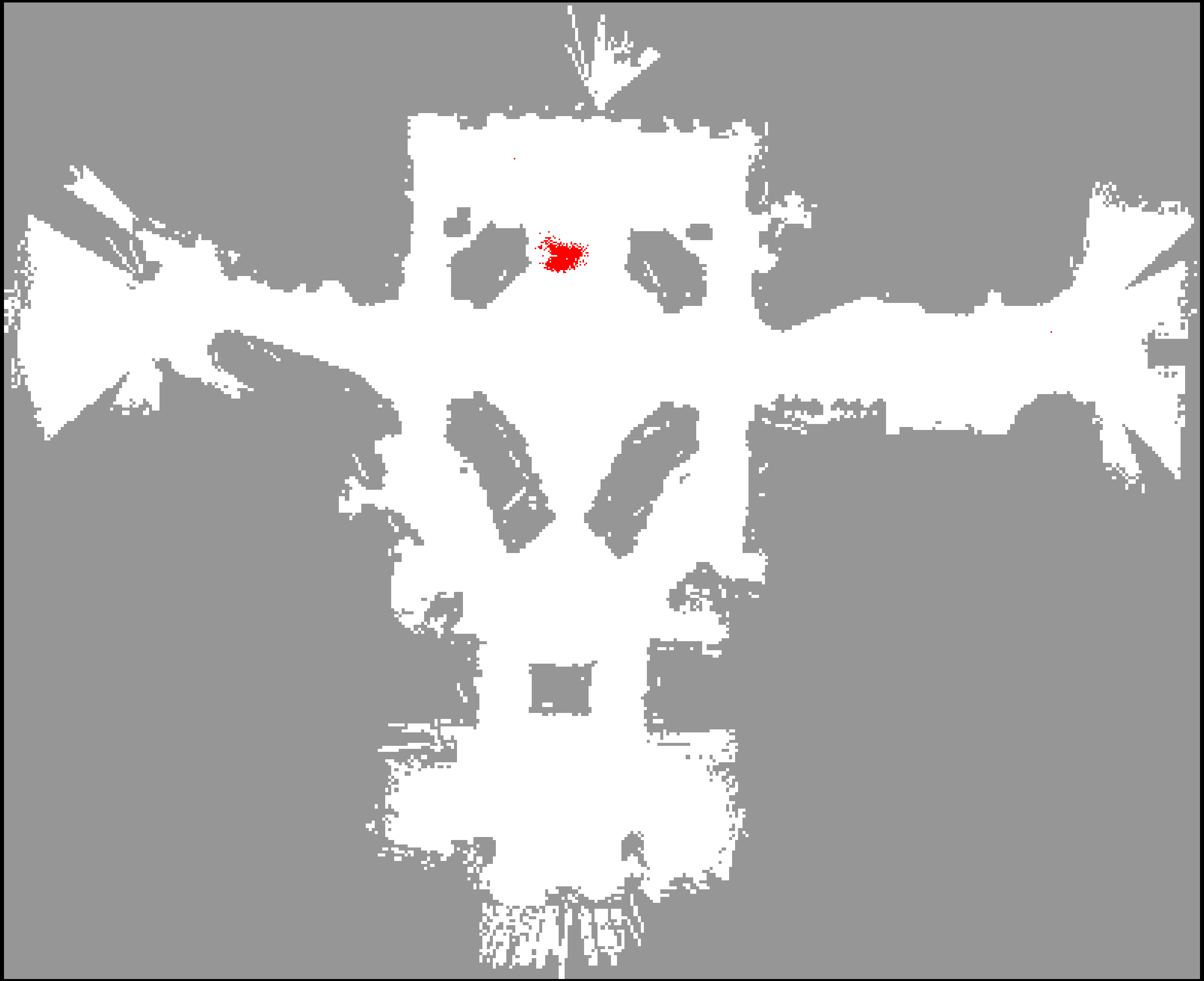


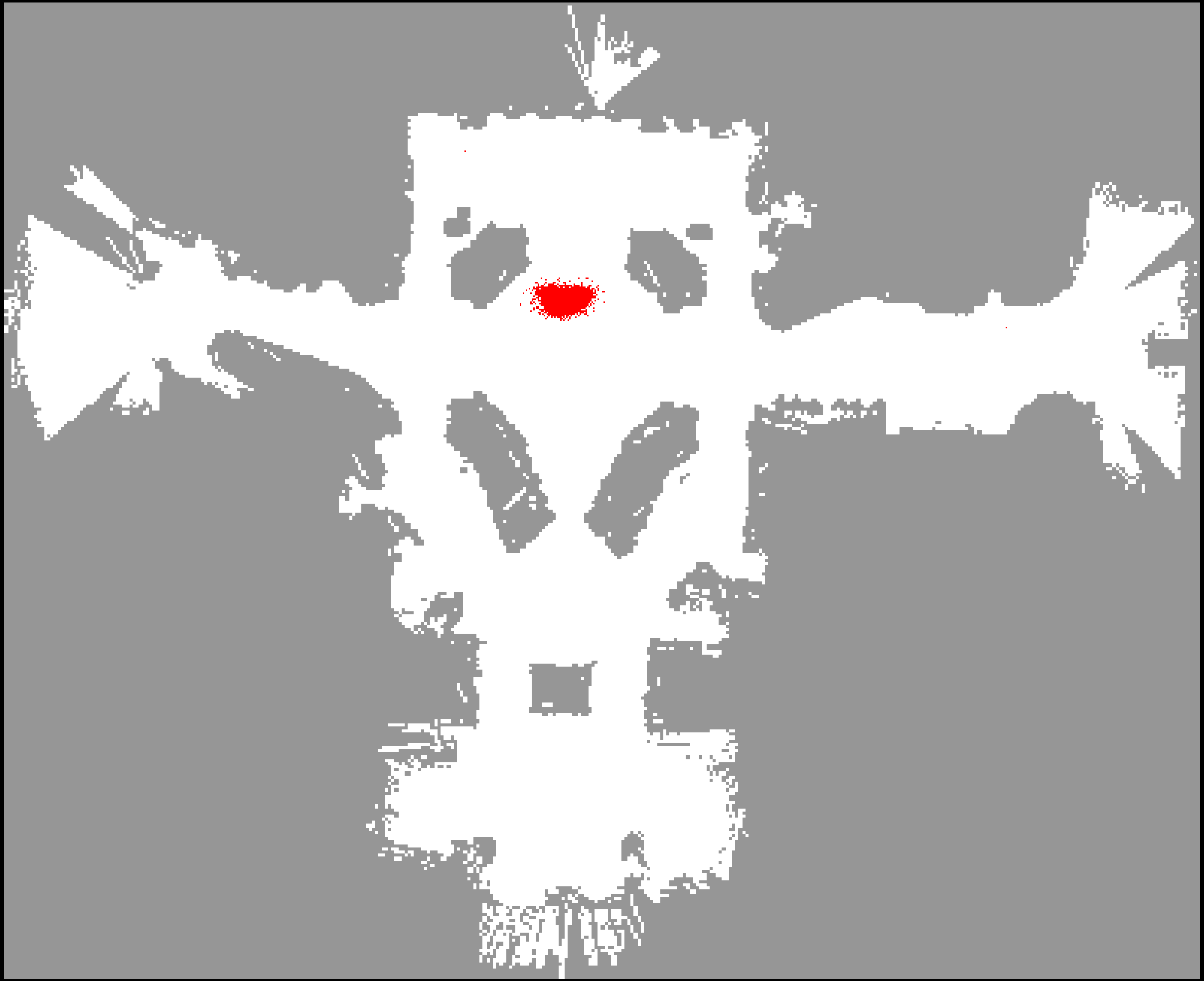


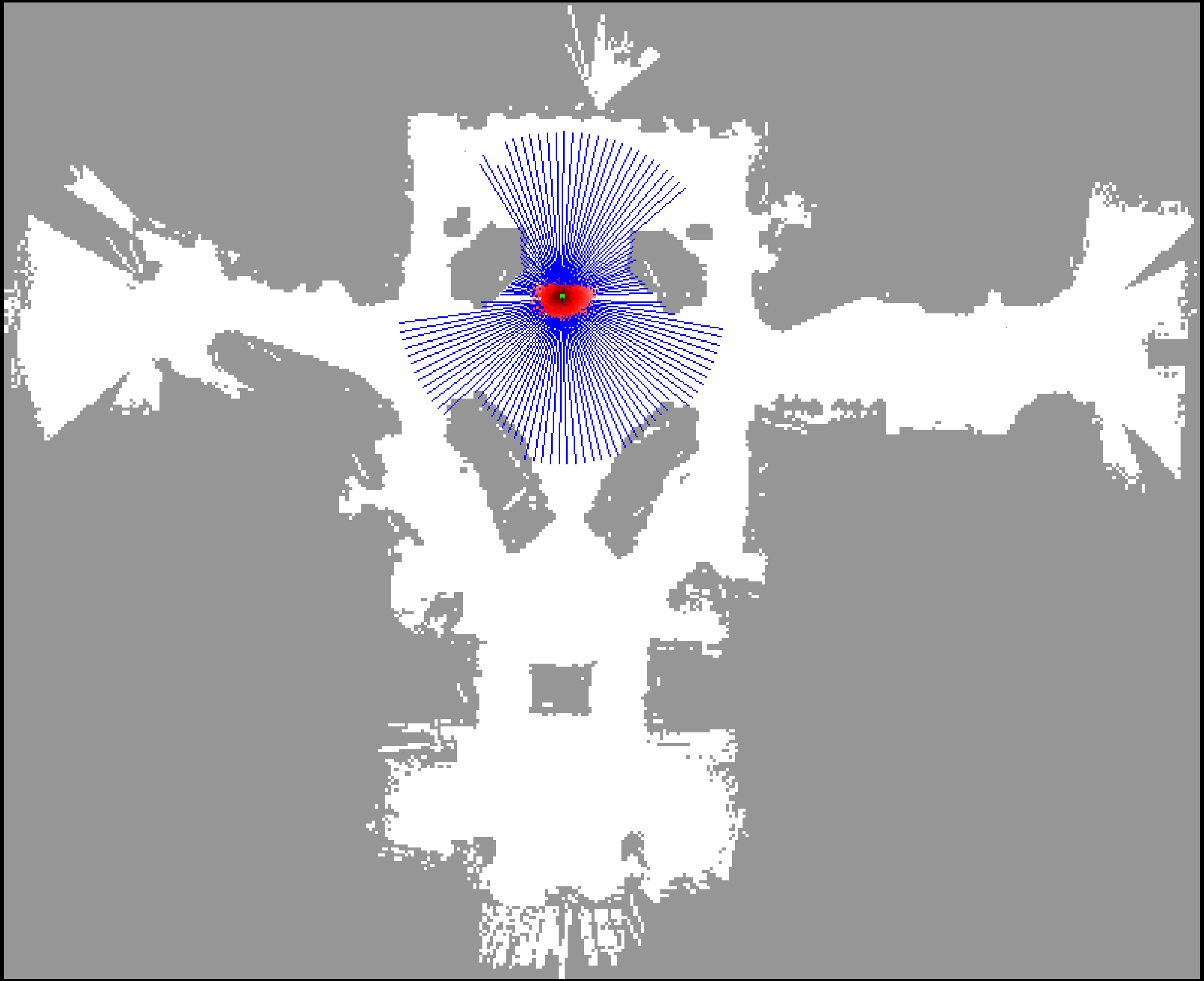
And so on...

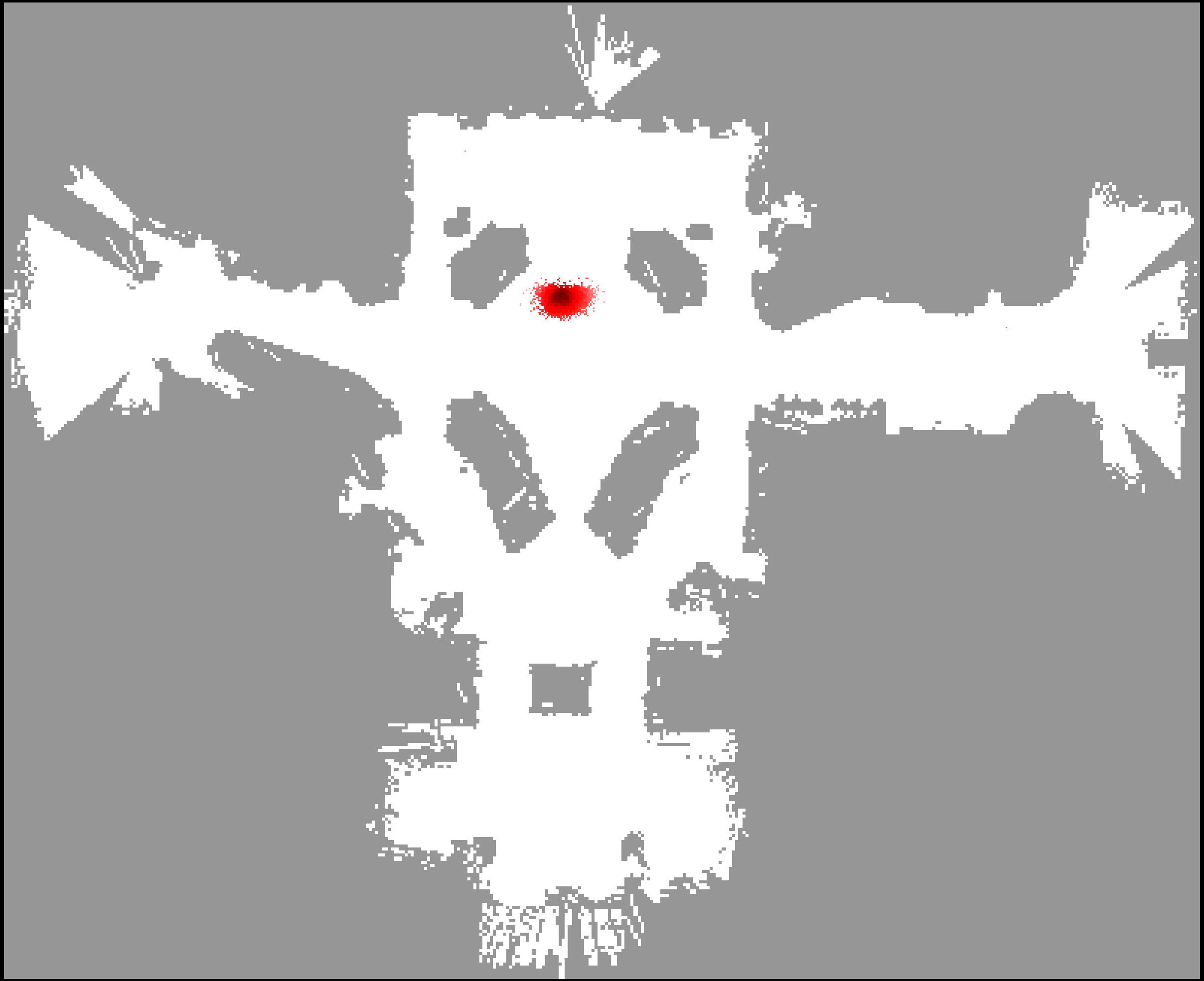


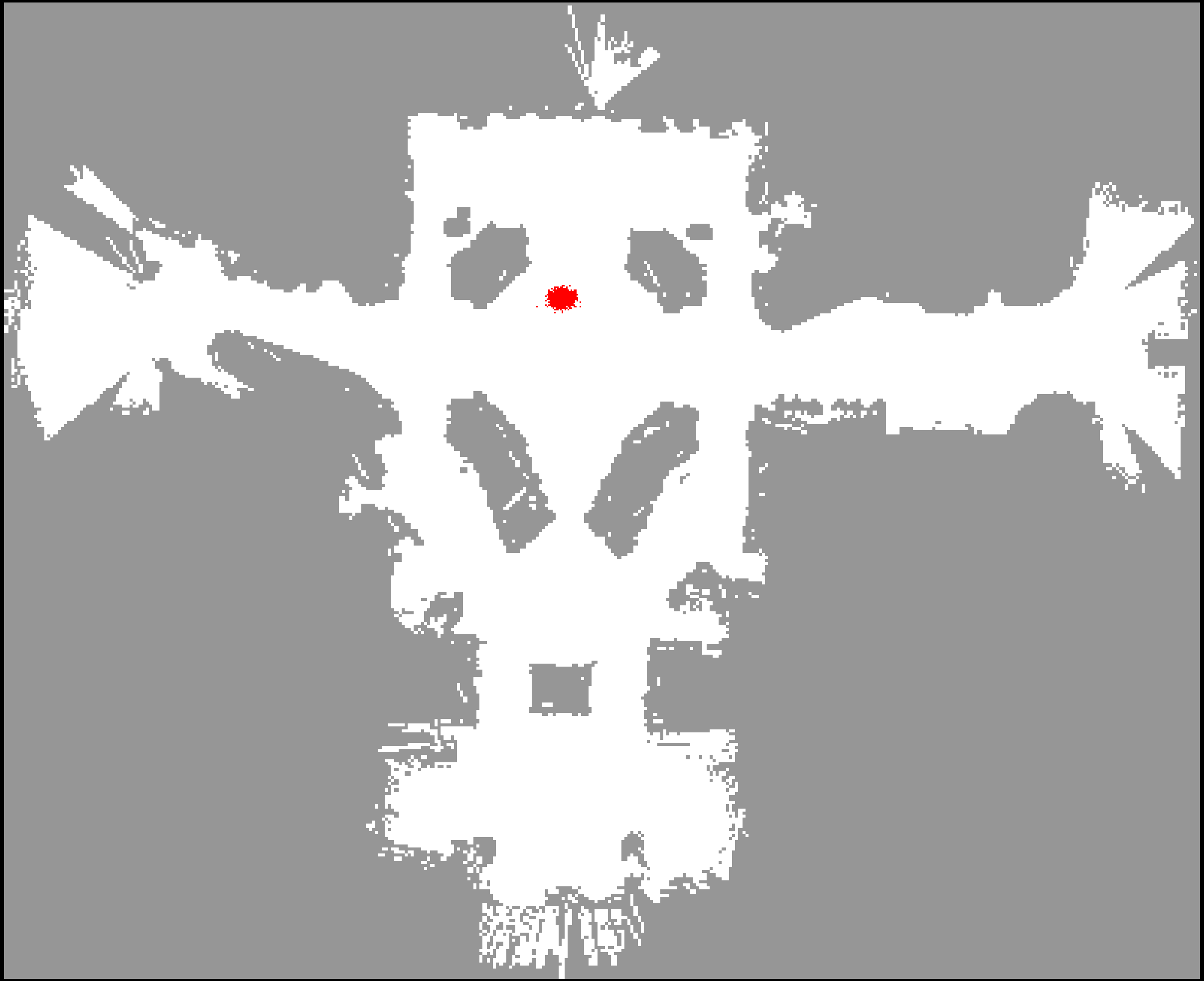




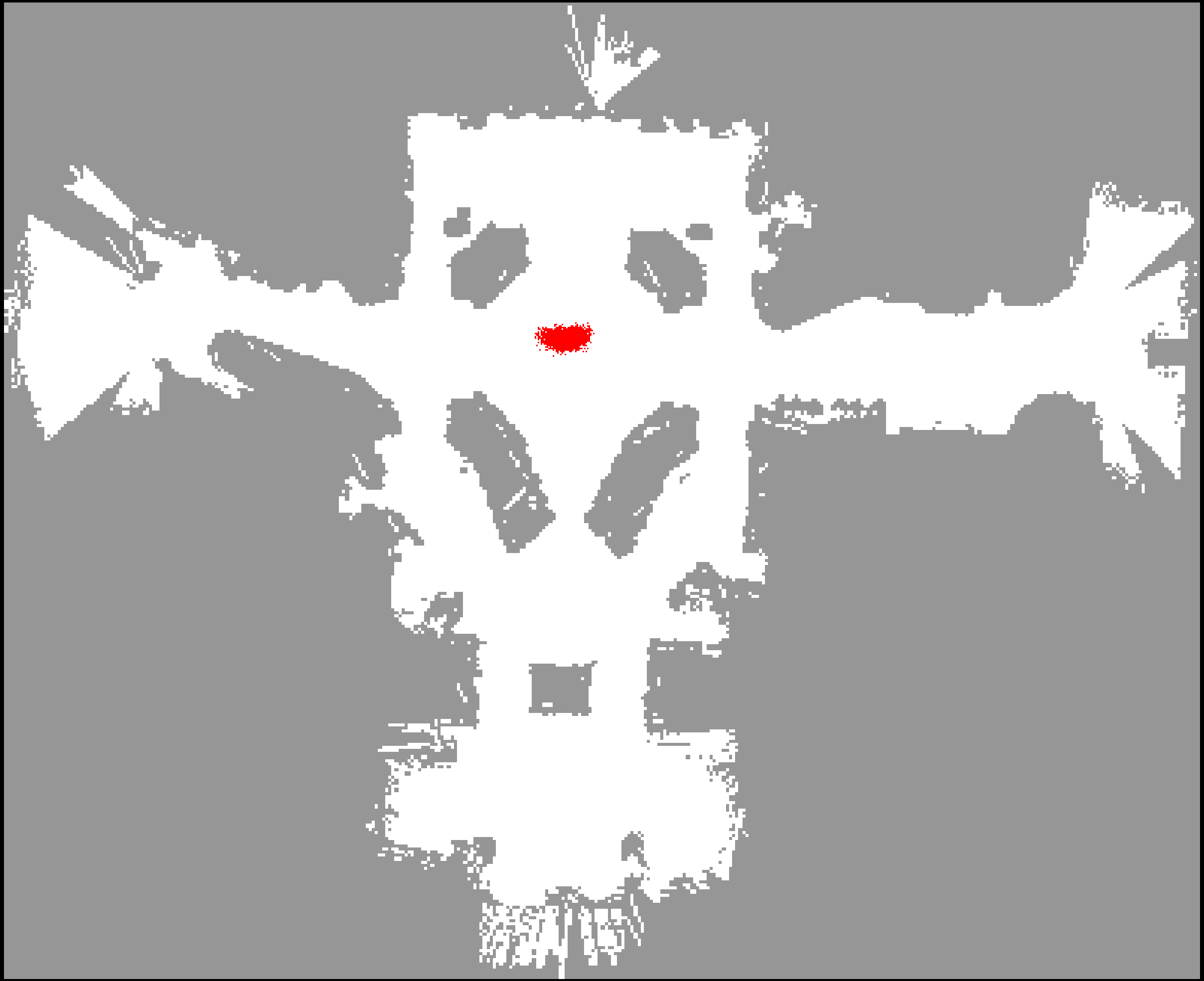


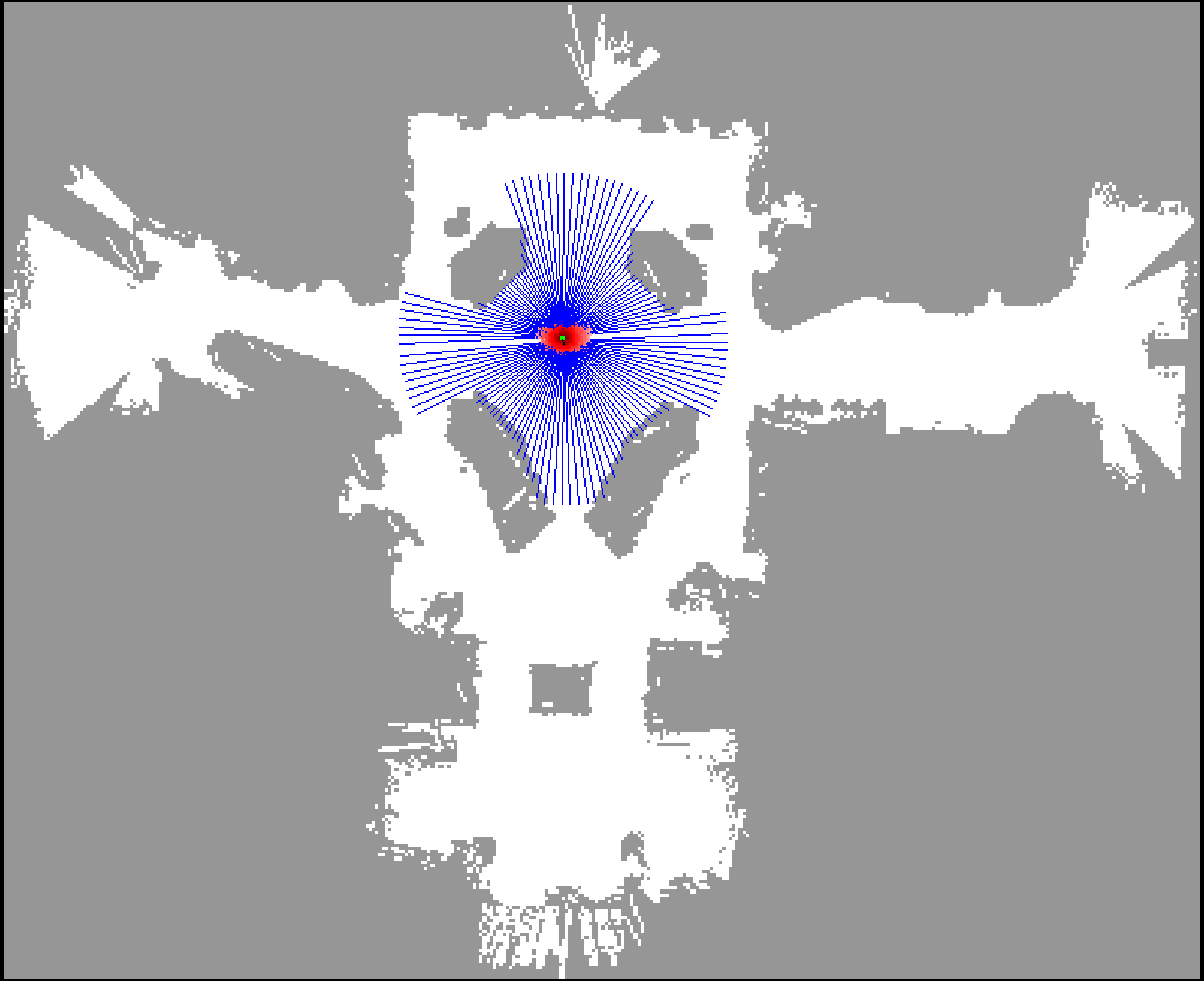






Converging to one cloud

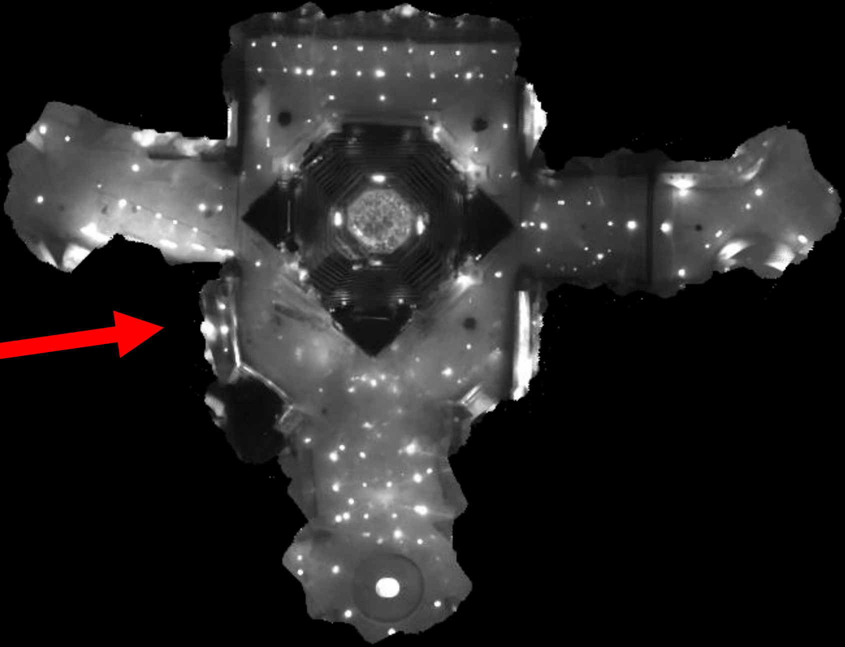




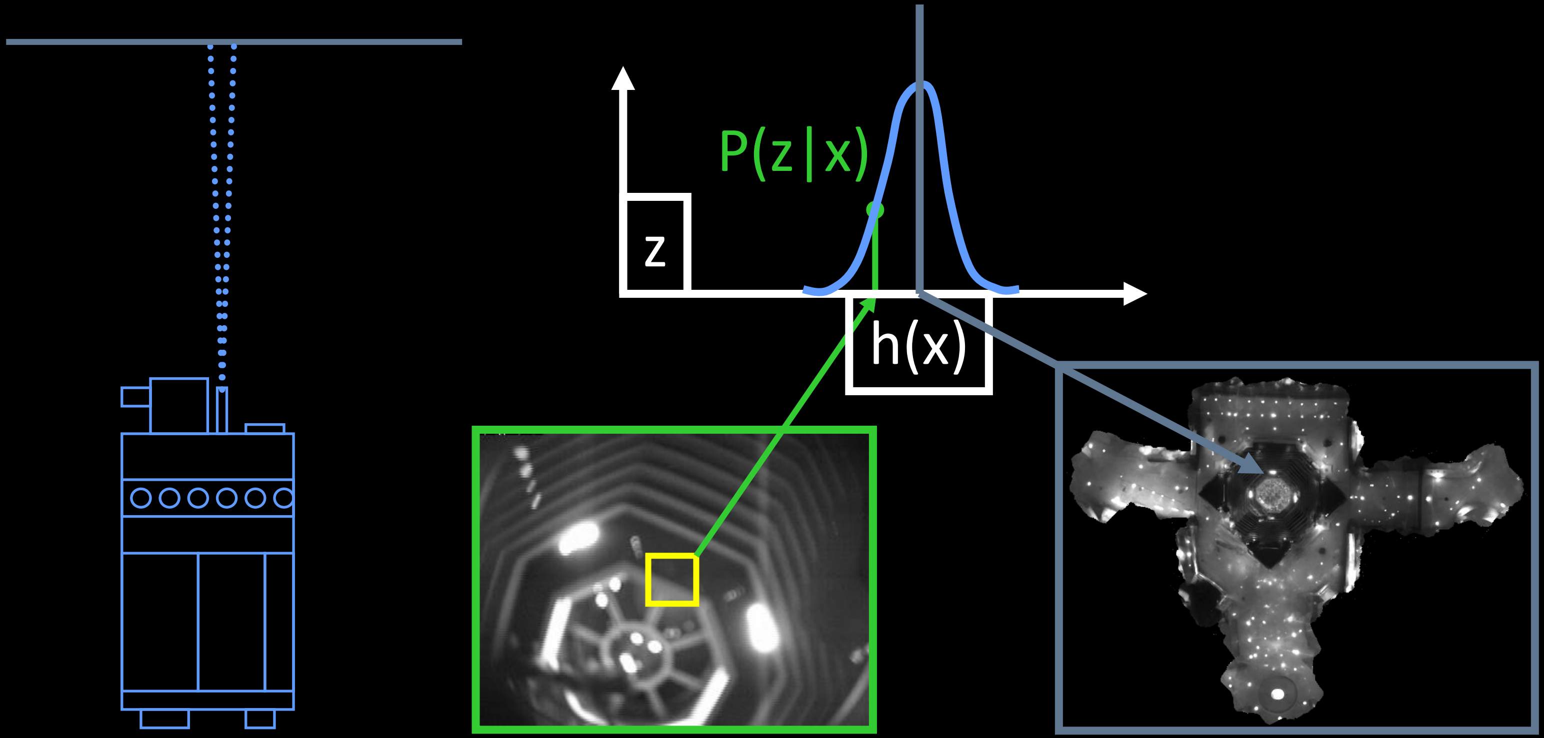
How about simple vision?

Using Ceiling Maps for Localization

Dellaert, et al. 1997

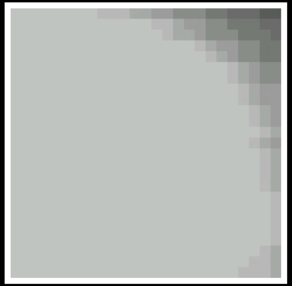


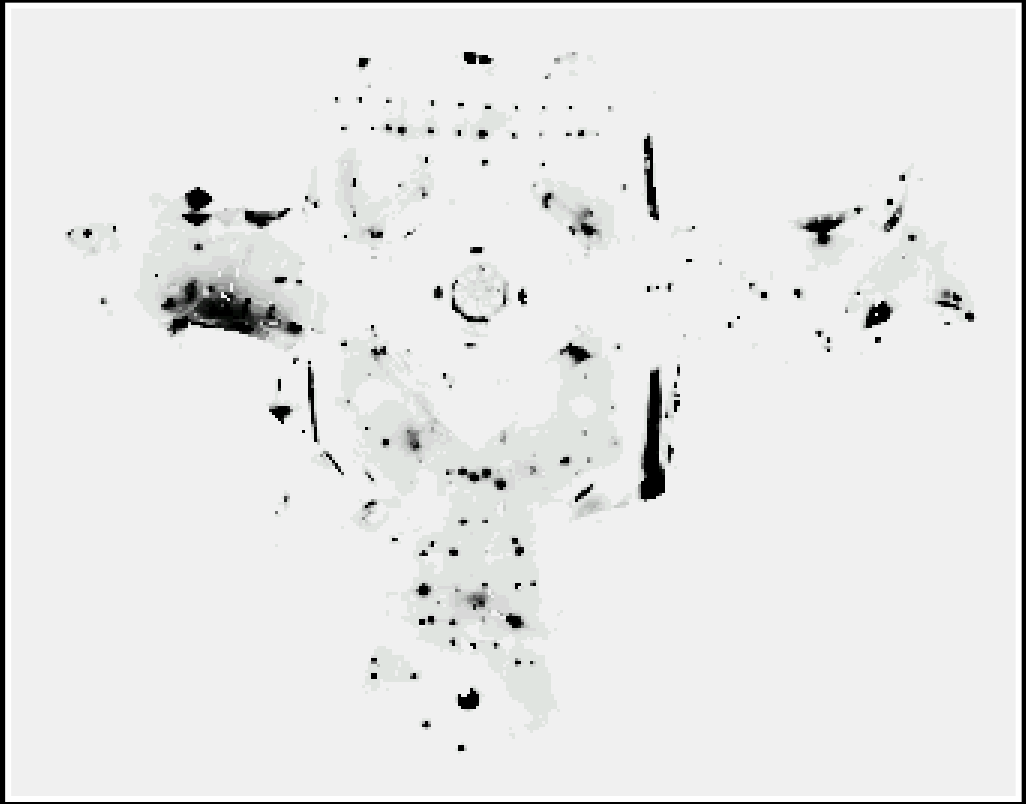
Vision-based Localization



Under a Light

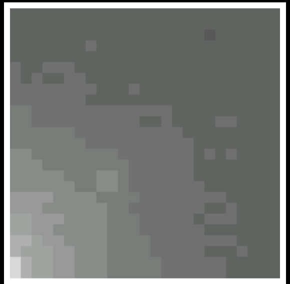
Measurement z: $P(z|x)$:

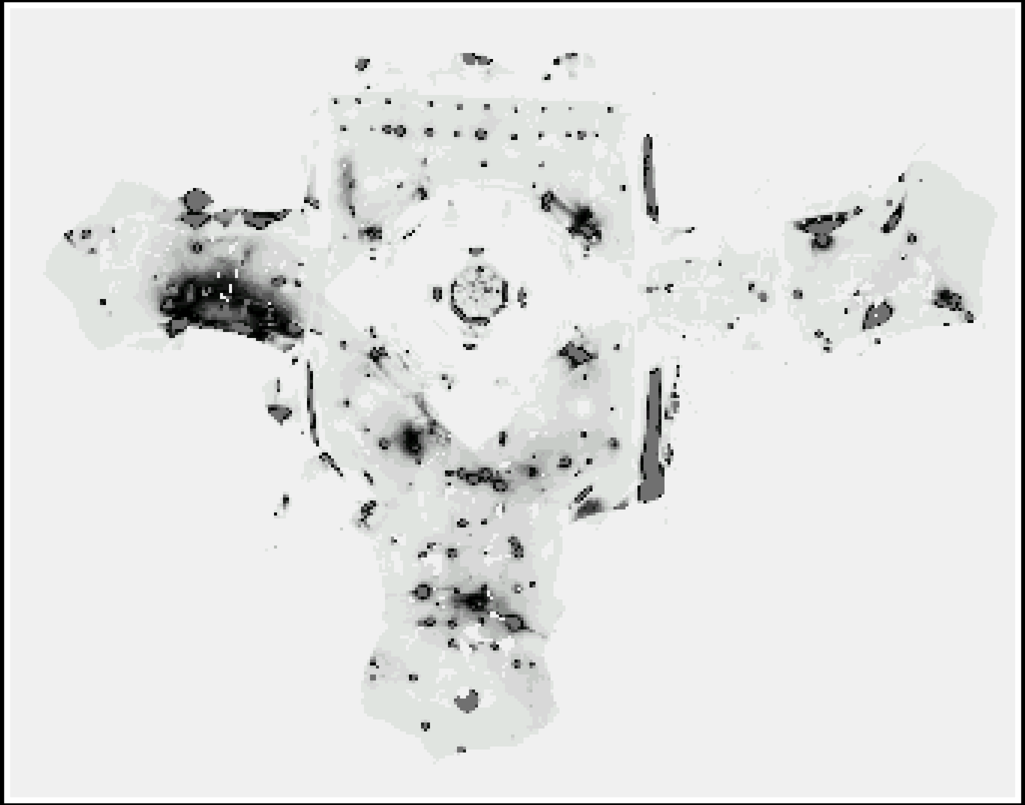




Next to a Light

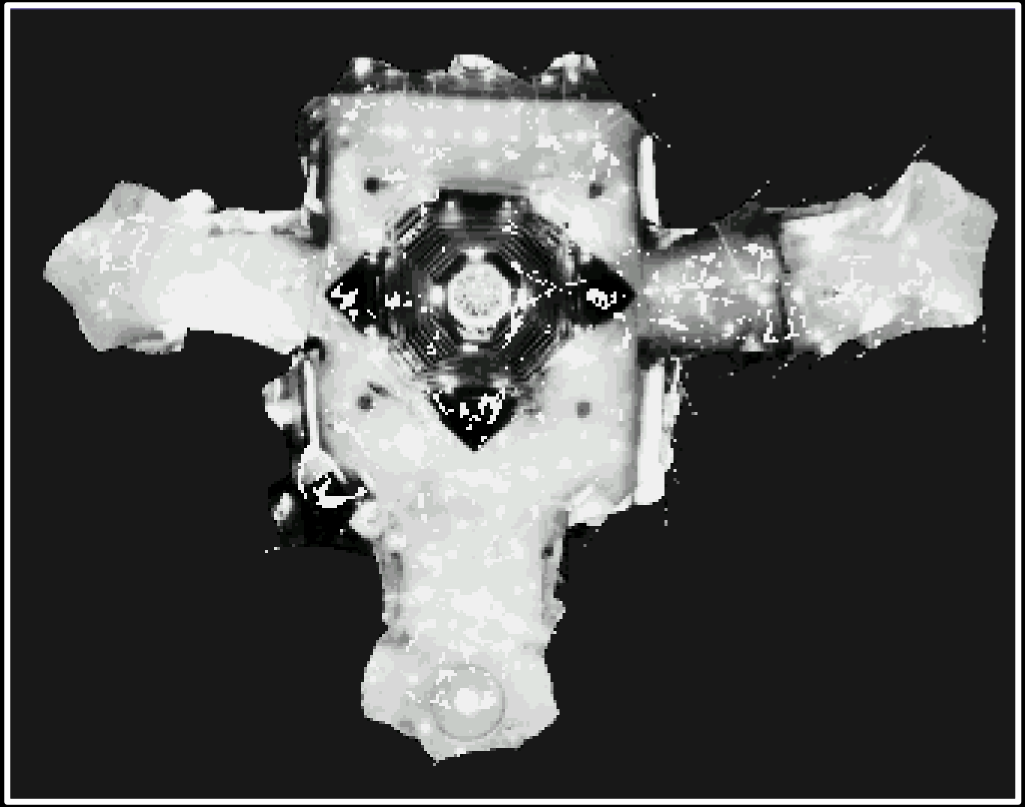
Measurement z: $P(z|x)$:



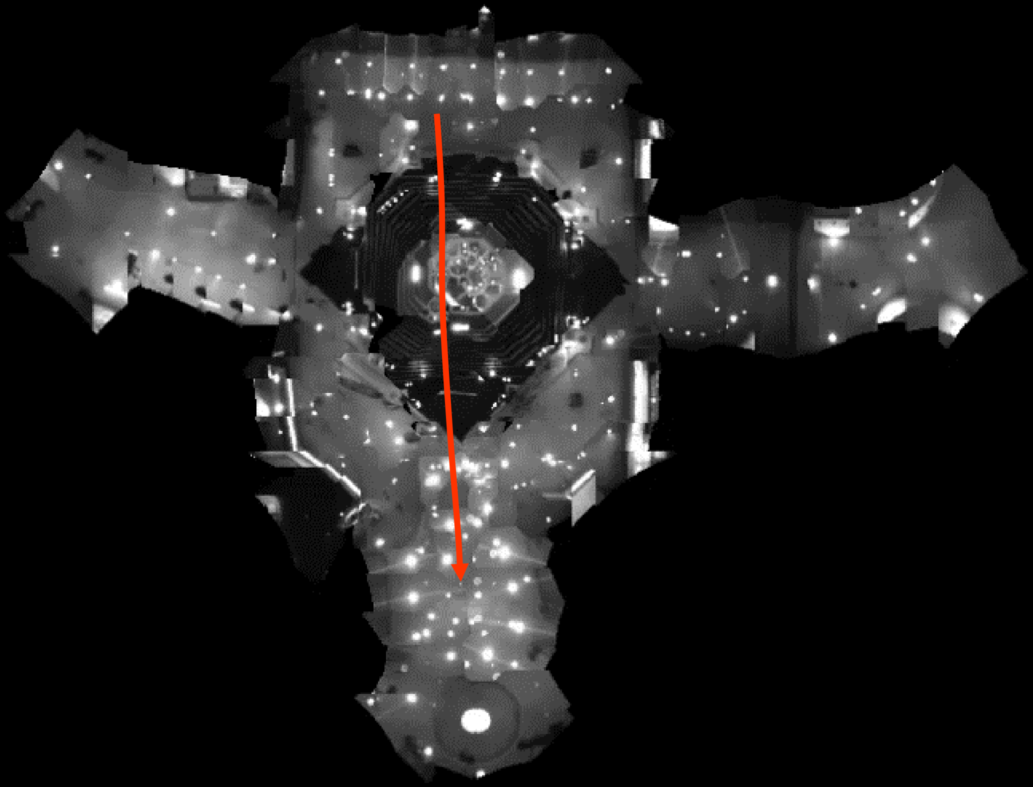


Elsewhere

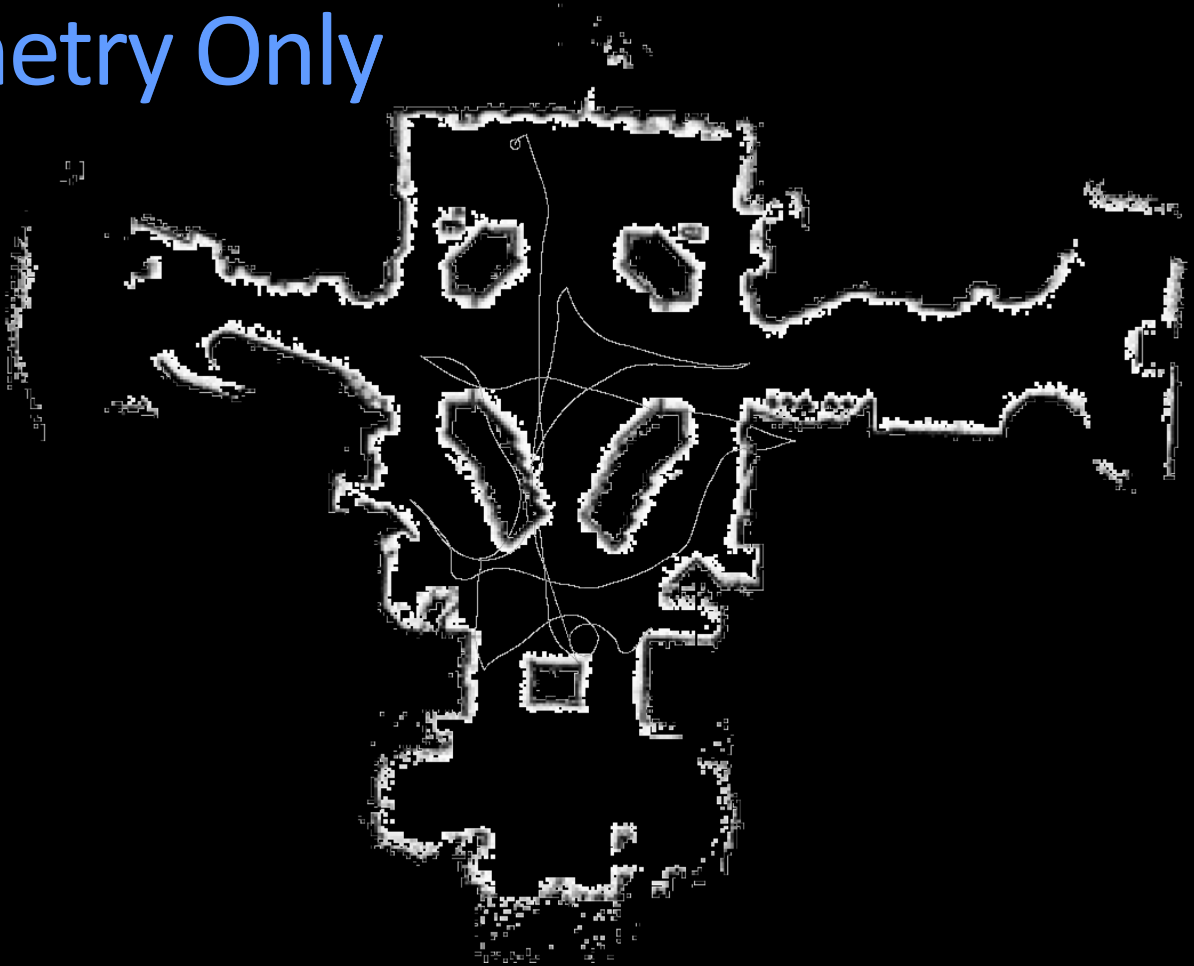
Measurement z: $P(z|x)$:



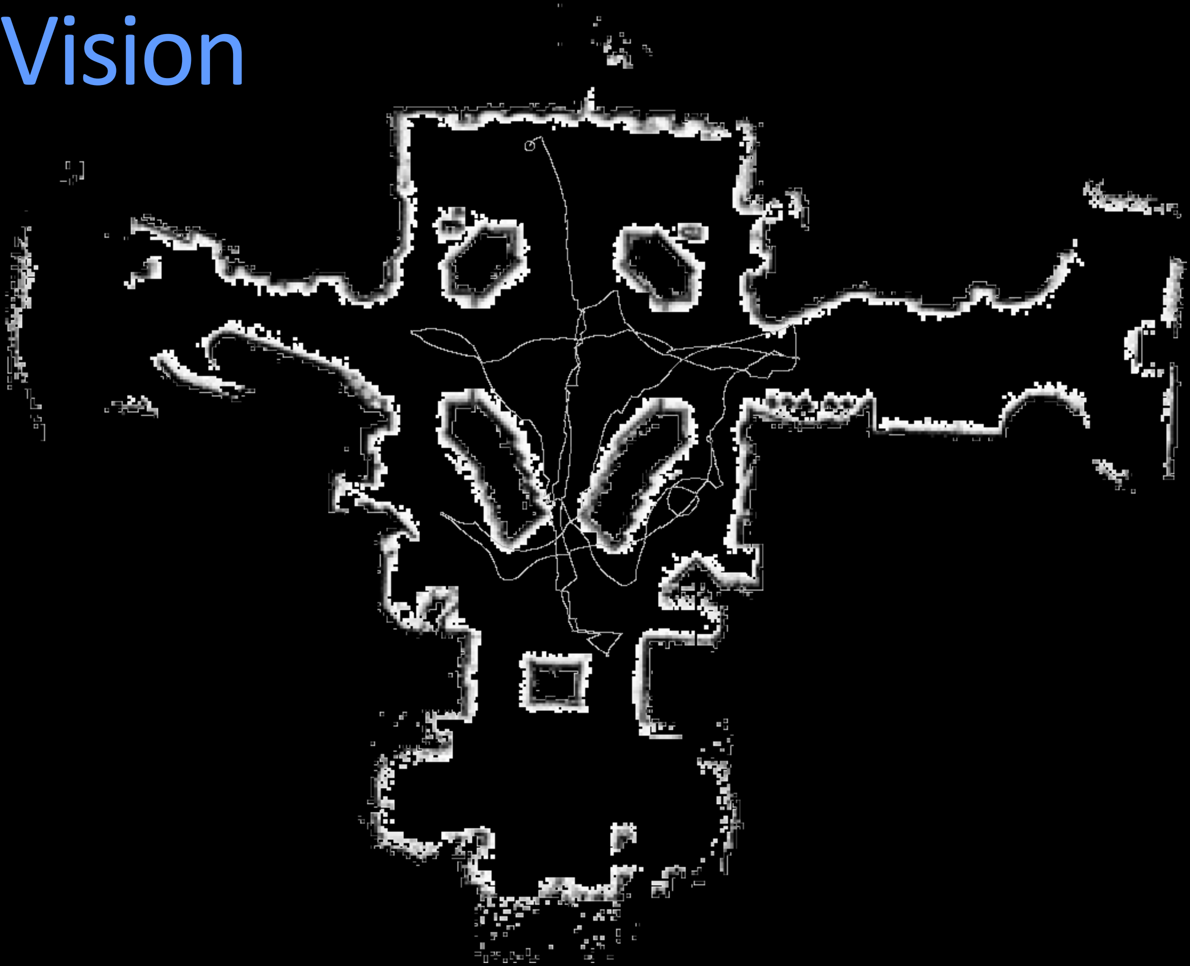
Global Localization Using Vision



Odometry Only



Using Vision



Particle Filters: Practical Considerations

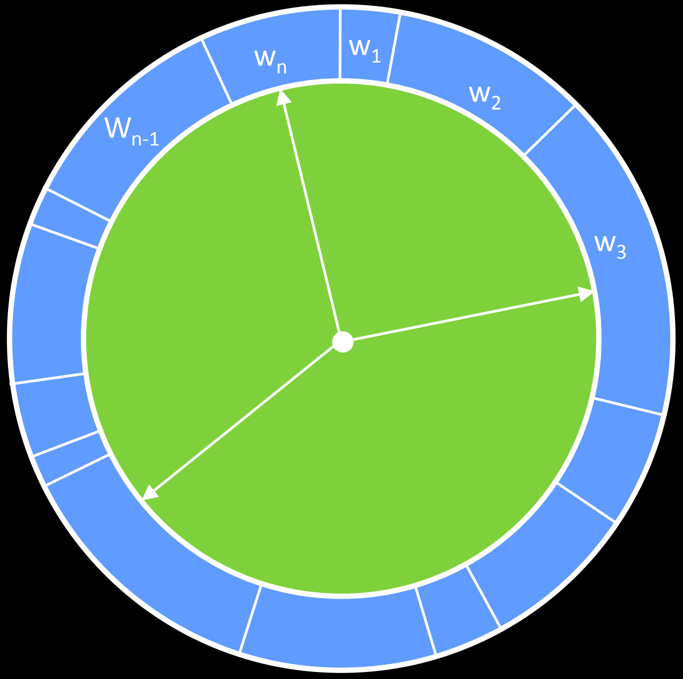
1. Sampling….

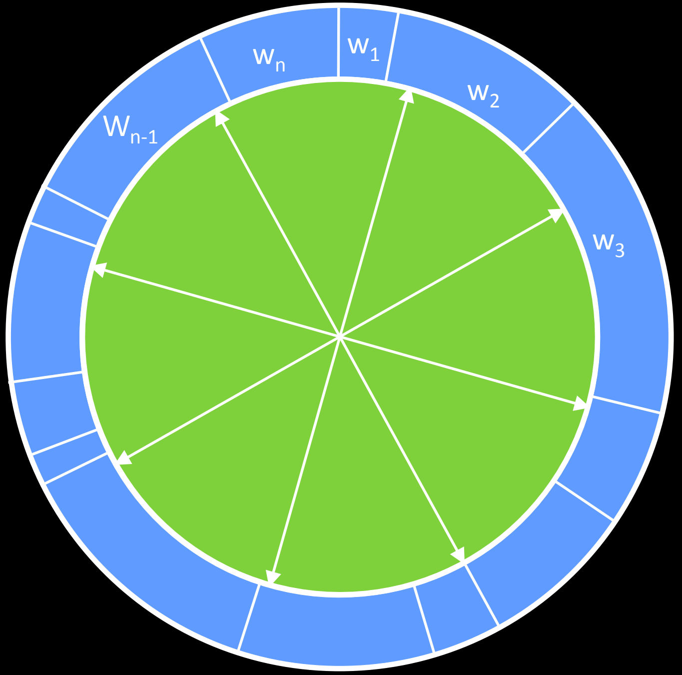
A detail: Resampling method can matter

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'





- Roulette wheel
- Binary search, n log n

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

Algorithm systematic resampling (S, n):

1. $S' = \emptyset,\ c_1 = w^1$
2. **For** $i = 2 \ldots n$
3. $\quad c_i = c_{i-1} + w^i$ — *Generate cdf (outer ring)*
4. $u_1 \sim U[0, n^{-1}],\ i = 1$ — *Initialize offset and first cdf bin*
5. **For** $j = 1 \ldots n$ — *Draw samples …*
6. $\quad$ **While** $(u_j > c_i)$ — *Skip until next cdf threshold reached*
7. $\quad\quad i = i + 1$
8. $\quad S' = S' \cup \{< x^i, n^{-1} >\}$ — *Insert sample from cdf ring*
9. $\quad u_{j+1} = u_j + n^{-1}$
10. **Return** S'

*(Also called **stochastic universal sampling**)*

Particle Filters: Practical Considerations

1. Sampling….

Resample only when necessary

- Efficiency of representation can be measured by variance of weights – want them “uniform.”

Particle Filters: Practical Considerations

2. Highly peaked observations
    - Add noise to observation and prediction models
    - Better proposal distributions – e.g., perform Kalman filter step to determine proposal

Overestimating noise often reduces number of required samples – always better to slightly over estimate than under.

Particle Filters: Practical Considerations

3. Recovery from failure - *resample*

- Selectively add samples from observations
- Uniformly add some samples