CS 4495 Computer Vision
Tracking 3: Follow the pixels…

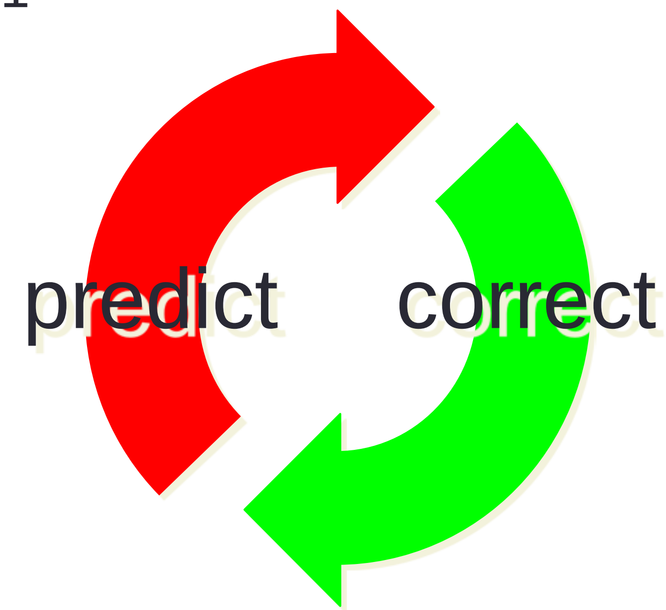
Aaron Bobick
School of Interactive Computing

Administrivia

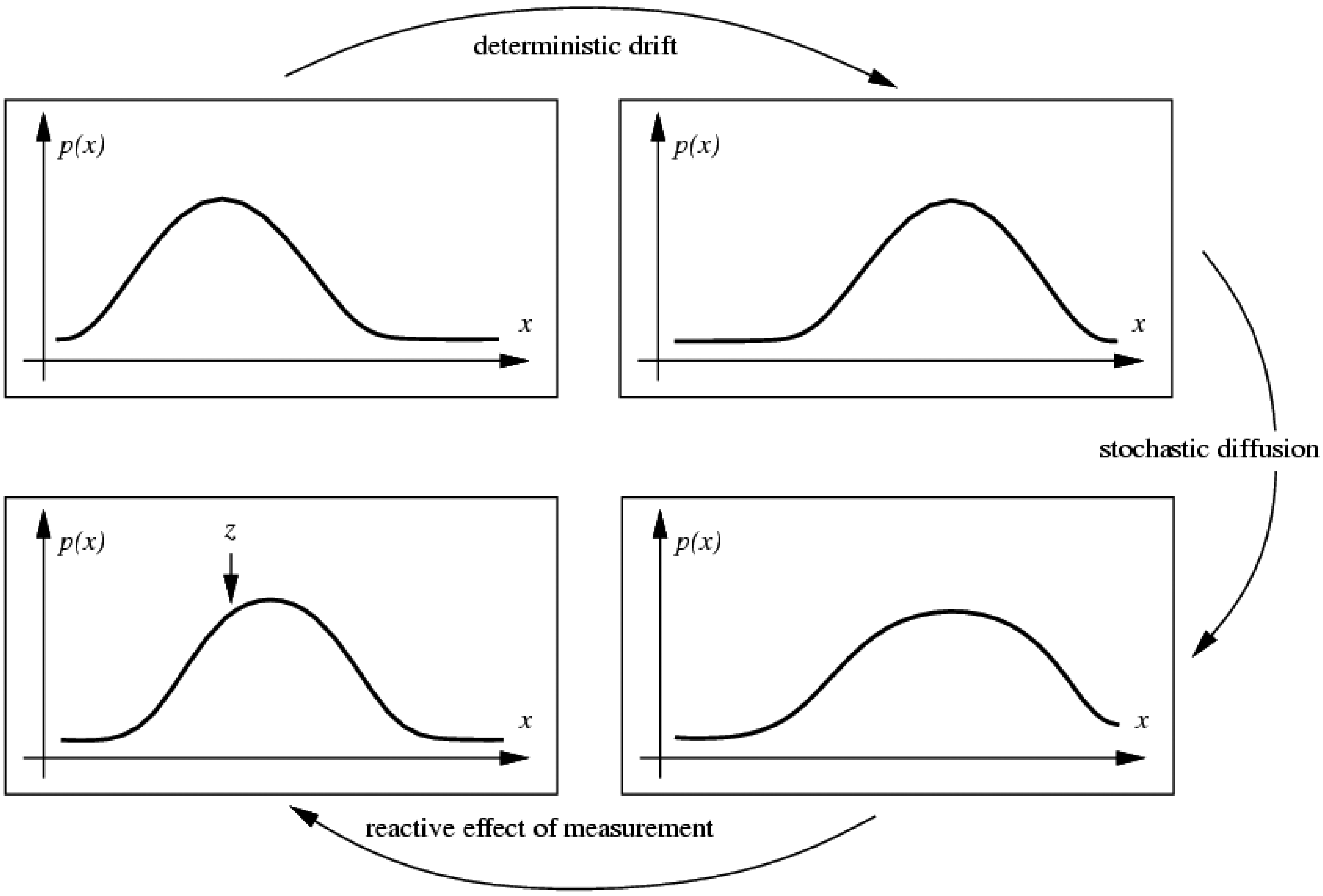
- PS6 out Thurs Nov 14, due Nov 24th
- EXAM: Tues before Thanksgiving, Nov 26th (not 21st !!)
  - Covers concepts and basics
  - Example questions will be posted on calendar by November 19th
- PS7 out Tue Nov 26th, due technically Friday, December 5.
  - Extension available until Sunday, Dec 7, a day that will live in infamy, 11:55pm.
- Problem set resubmission policy:
  - Full questions only
  - You get 50% credit to replace whatever you got last time on that question.
  - Must be submitted by: **SUNDAY DECEMBER 7, 11:55pm**
  - No prior notice needed.
- Grades posted as available.

Tracking as induction

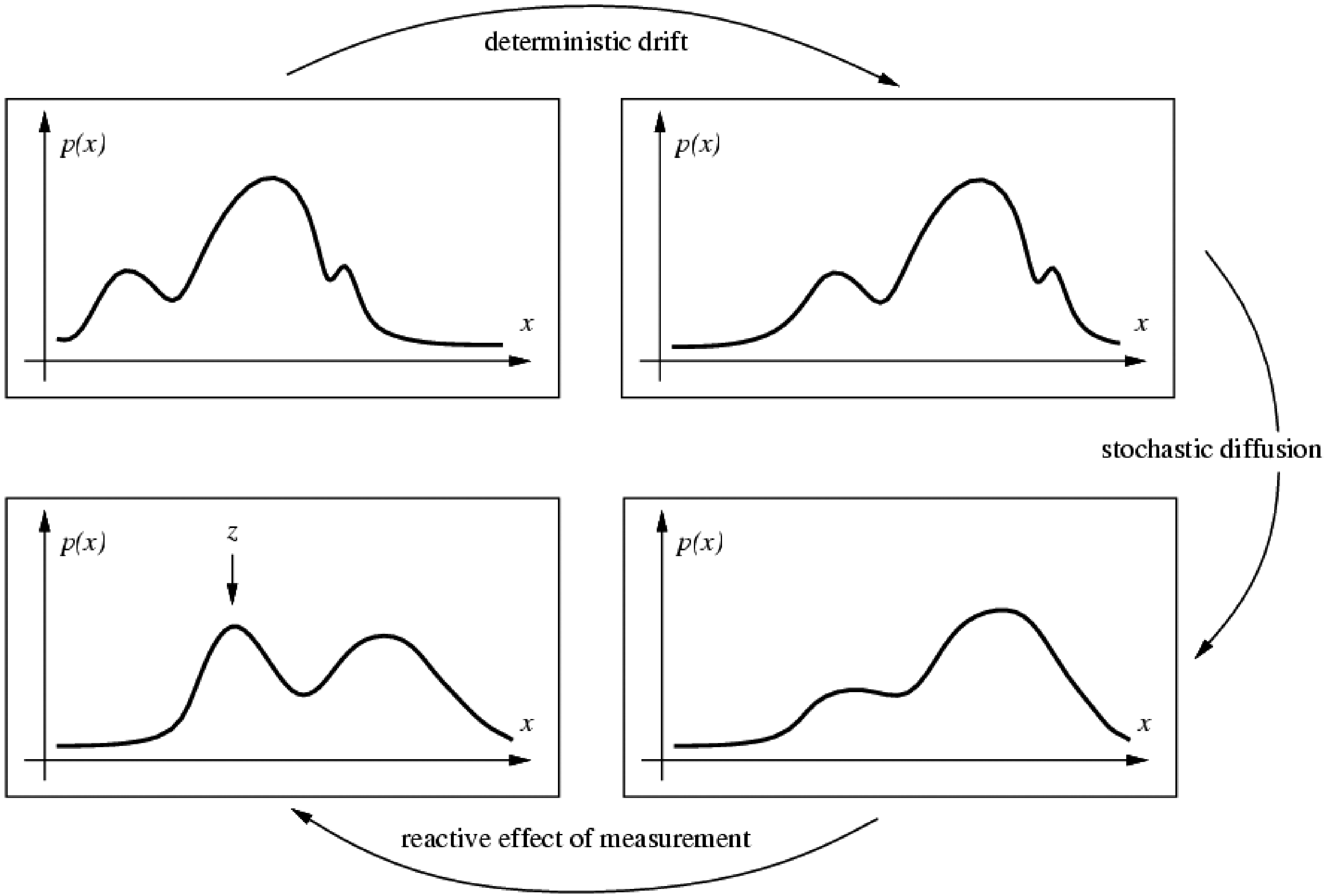
- Base case:
  - Assume we have initial prior that predicts state in absence of any evidence: $P(X_0)$
  - At the first frame, *correct* this given the value of $Y_0=y_0$
- Given corrected estimate for frame t:
  - Predict for frame t+1
  - Correct for frame t+1



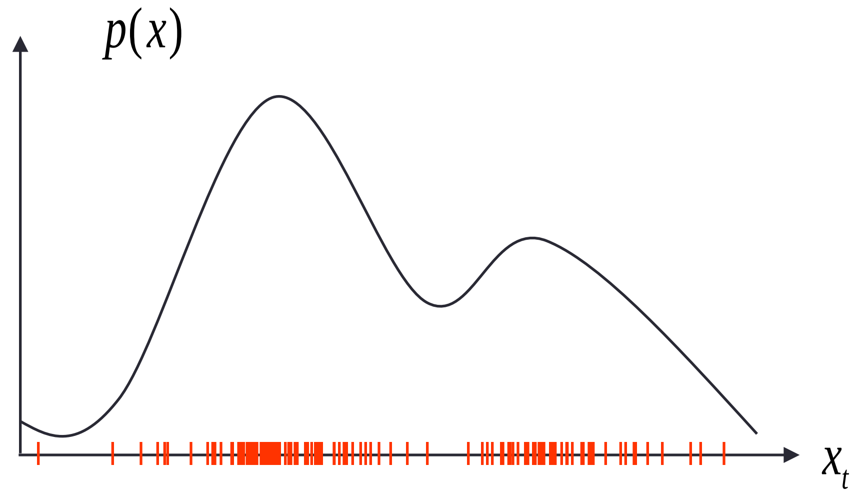
Kalman: Propagation of Gaussian densities



Propagation of general densities



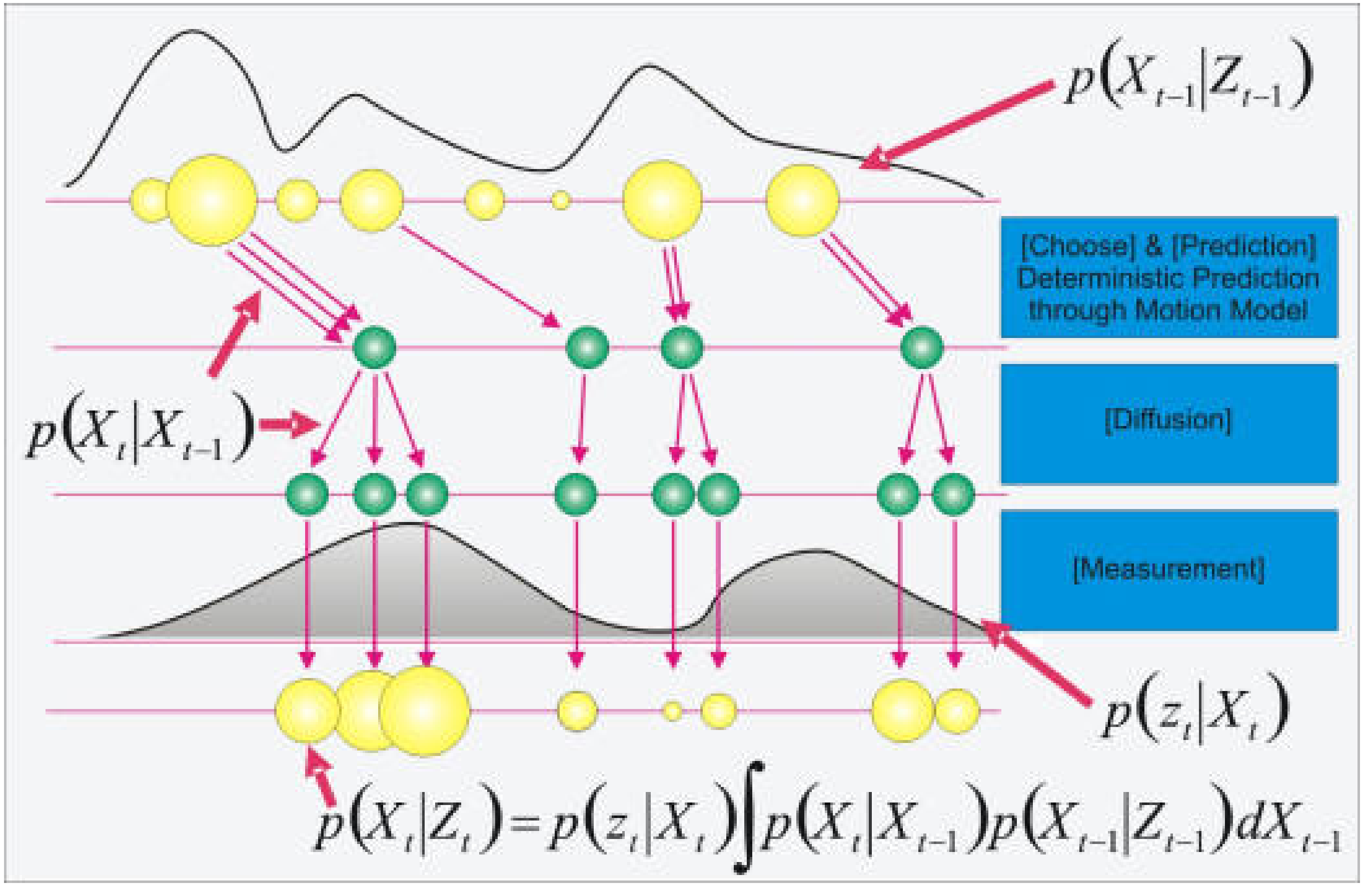
Particle Filters: Basic Idea



set of n (weighted) particles X_t

Density is represented by both **where** the particles are and their **weight**.

Graphical steps particle filtering



Bayes Filters: Framework

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

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

  - Sensor model $P(z_t|x_t)$.
  - Action model $P(x_{t+1}|u_t,x_t)$.
  - 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**:

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

Bayes Rule reminder

Prior before measurement

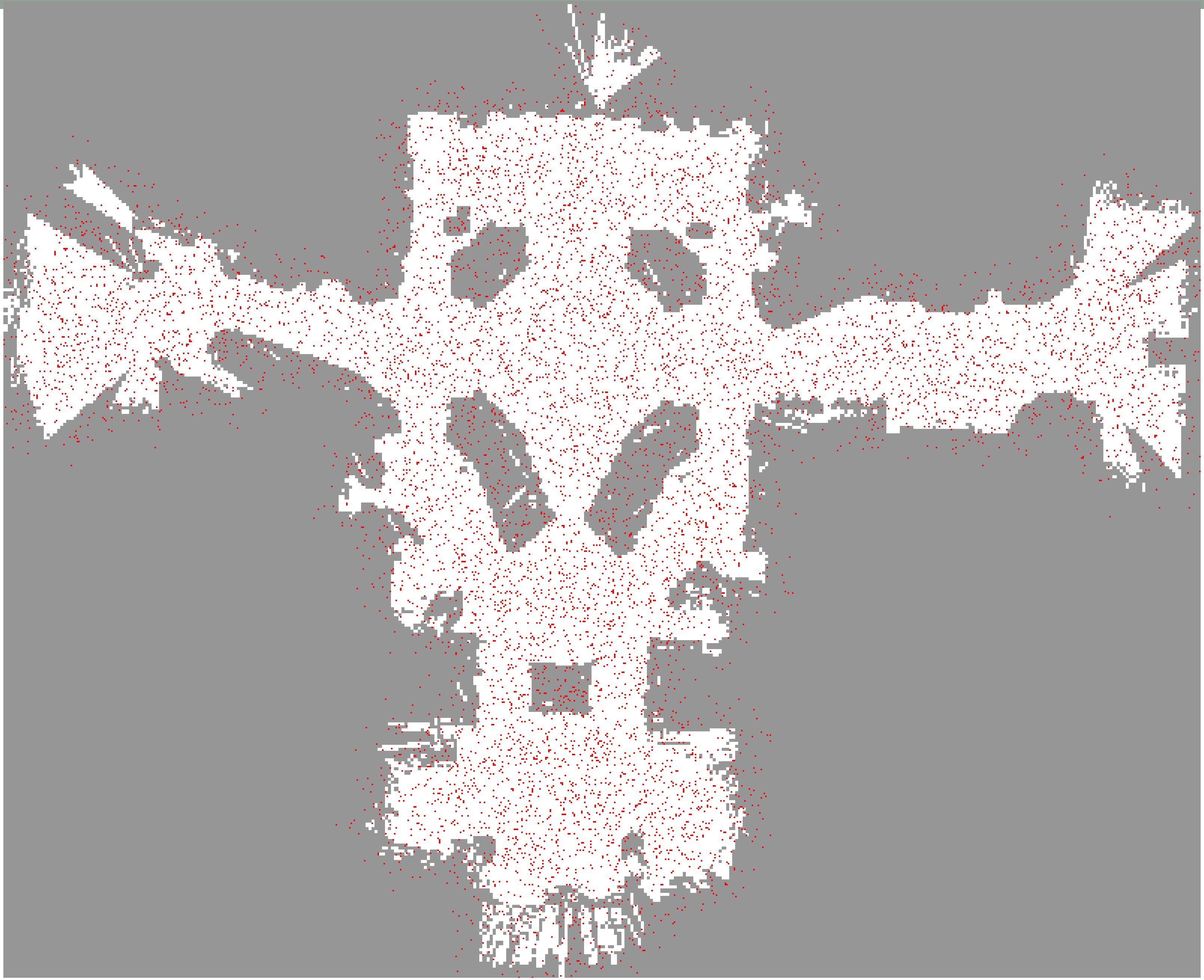
$$p(x \mid z) = \frac{p(z \mid x)\,\boxed{p(x)}}{p(z)}$$

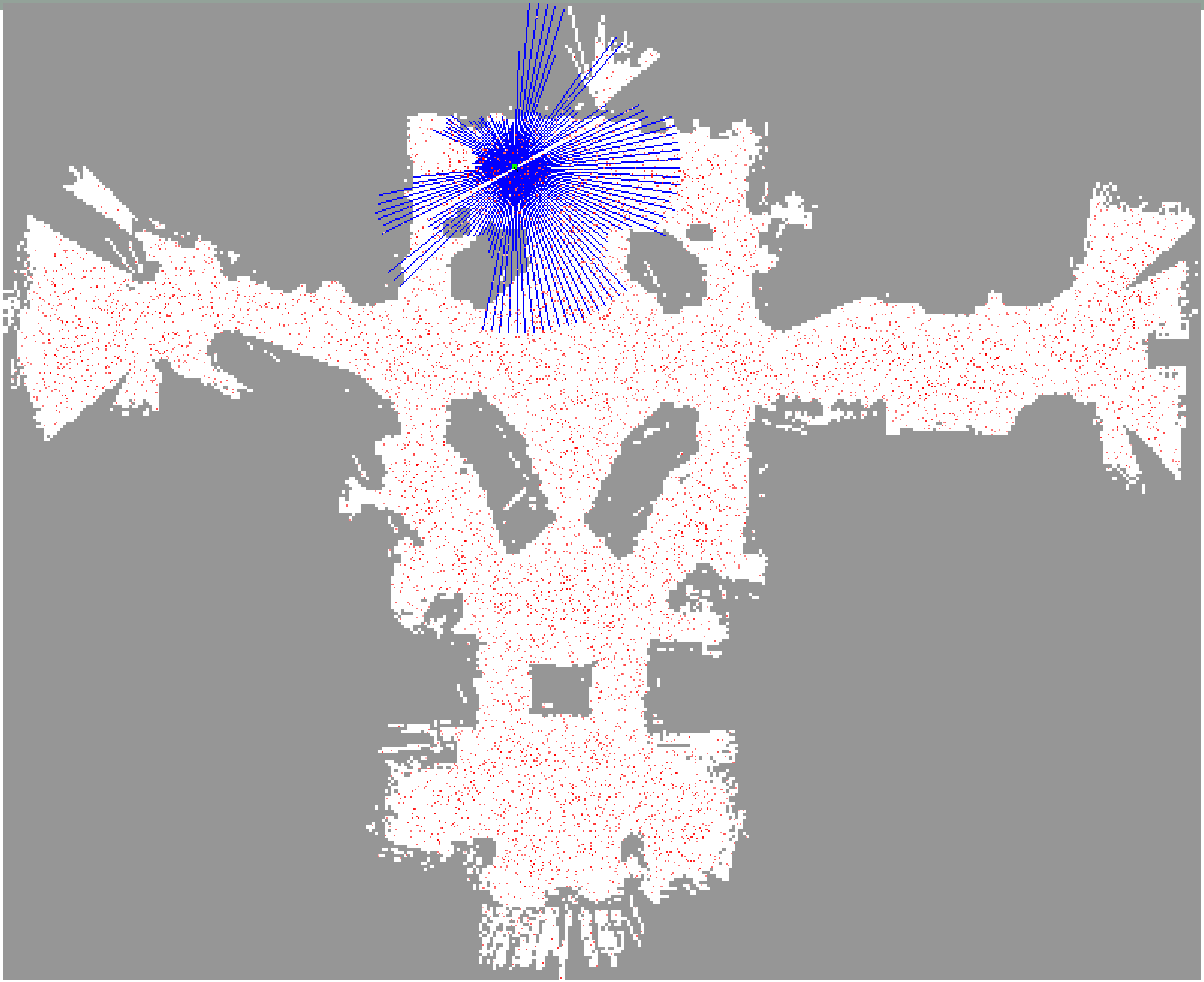
$$= \eta\, p(z \mid x)\, p(x)$$

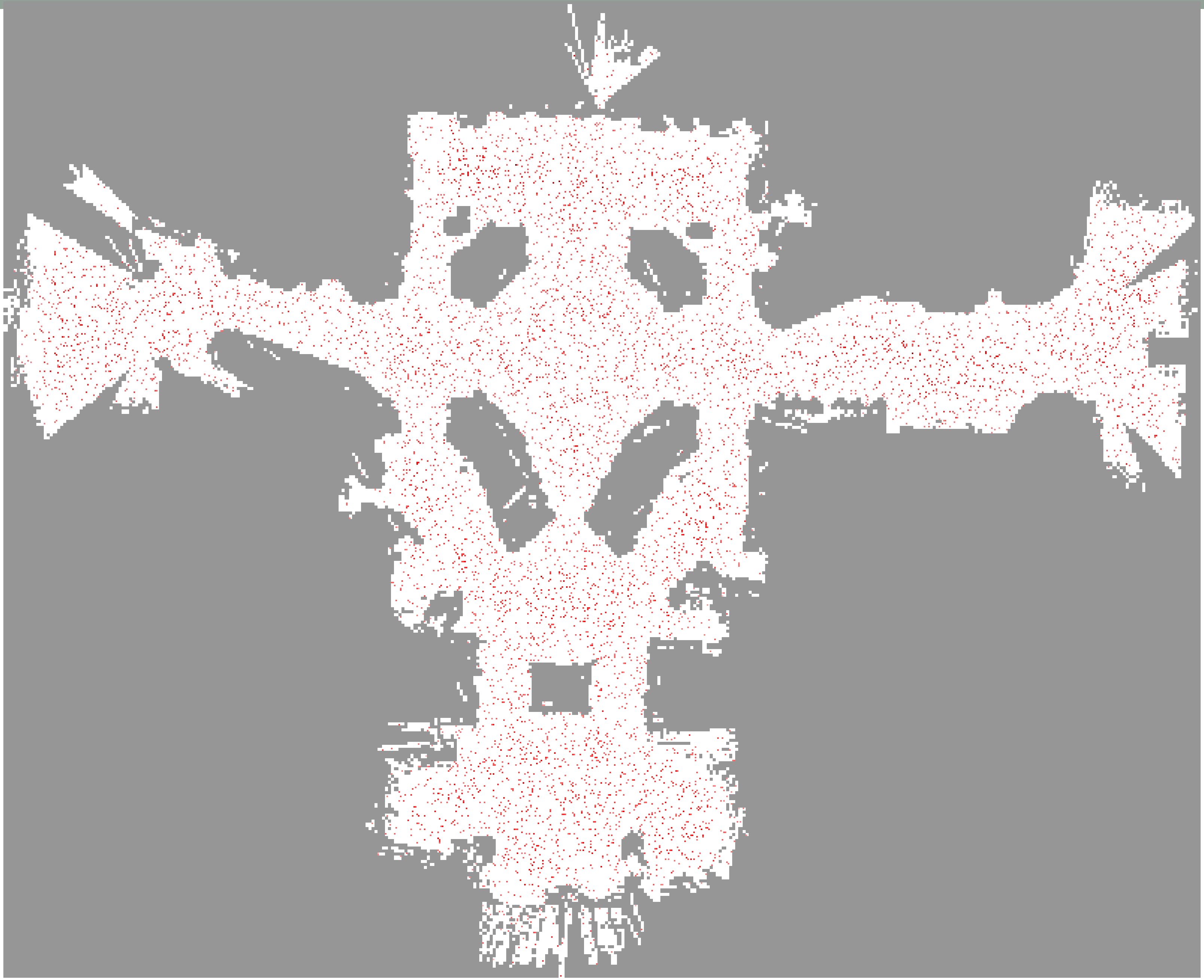
Particle Filter Algorithm (Sequential Importance Resampling)

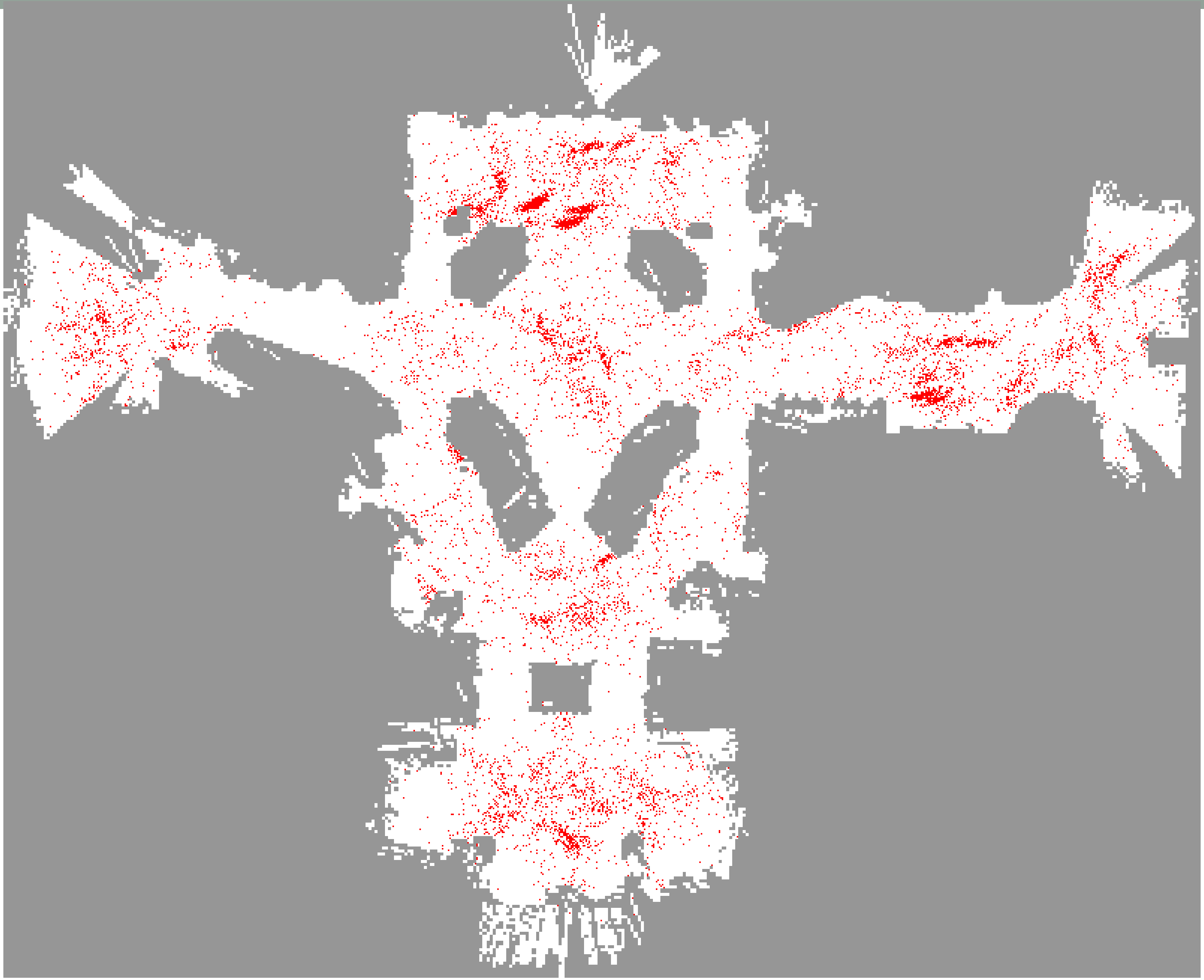
1. Algorithm **particle_filter** $\{S_{t-1} = <x_{t-1}^j, w_{t-1}^j>, u_t, z_t\}$
2. $S_t = \emptyset, \quad \eta = 0$
3. **For** $i = 1 \ldots n$ ***Resample (generate i new samples)***
4. Sample index *j(i)* from the discrete distribution given by w_{t-1}
5. Sample x_t^i from $p(x_t \mid x_{t-1}, u_t)$ using $x_{t-1}^{j(i)}$ and u_t ***Control***
6. $w_t^i = p(z_t \mid x_t^i)$ ***Compute importance weight (or reweight)***
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***

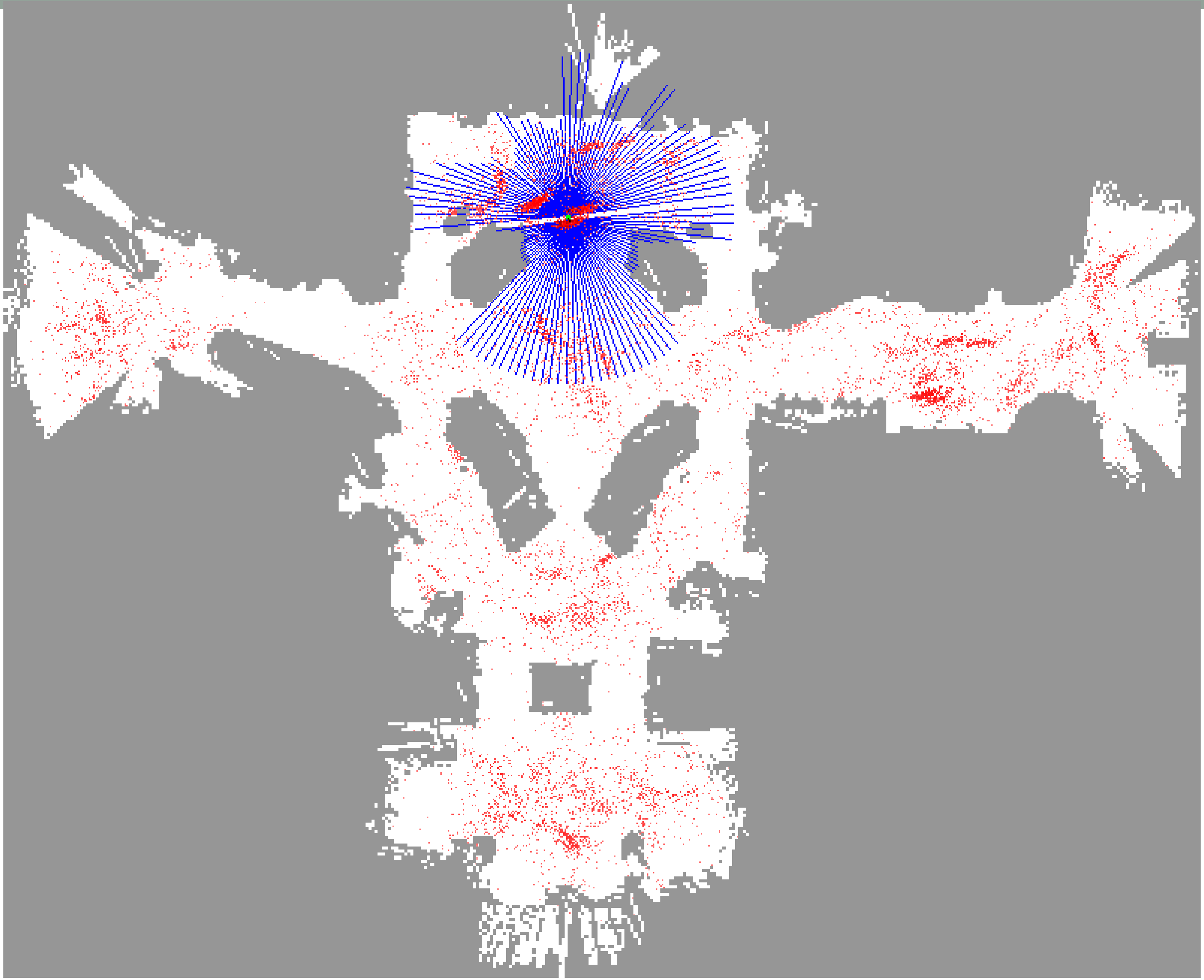
Smithsonian Museum of American History…

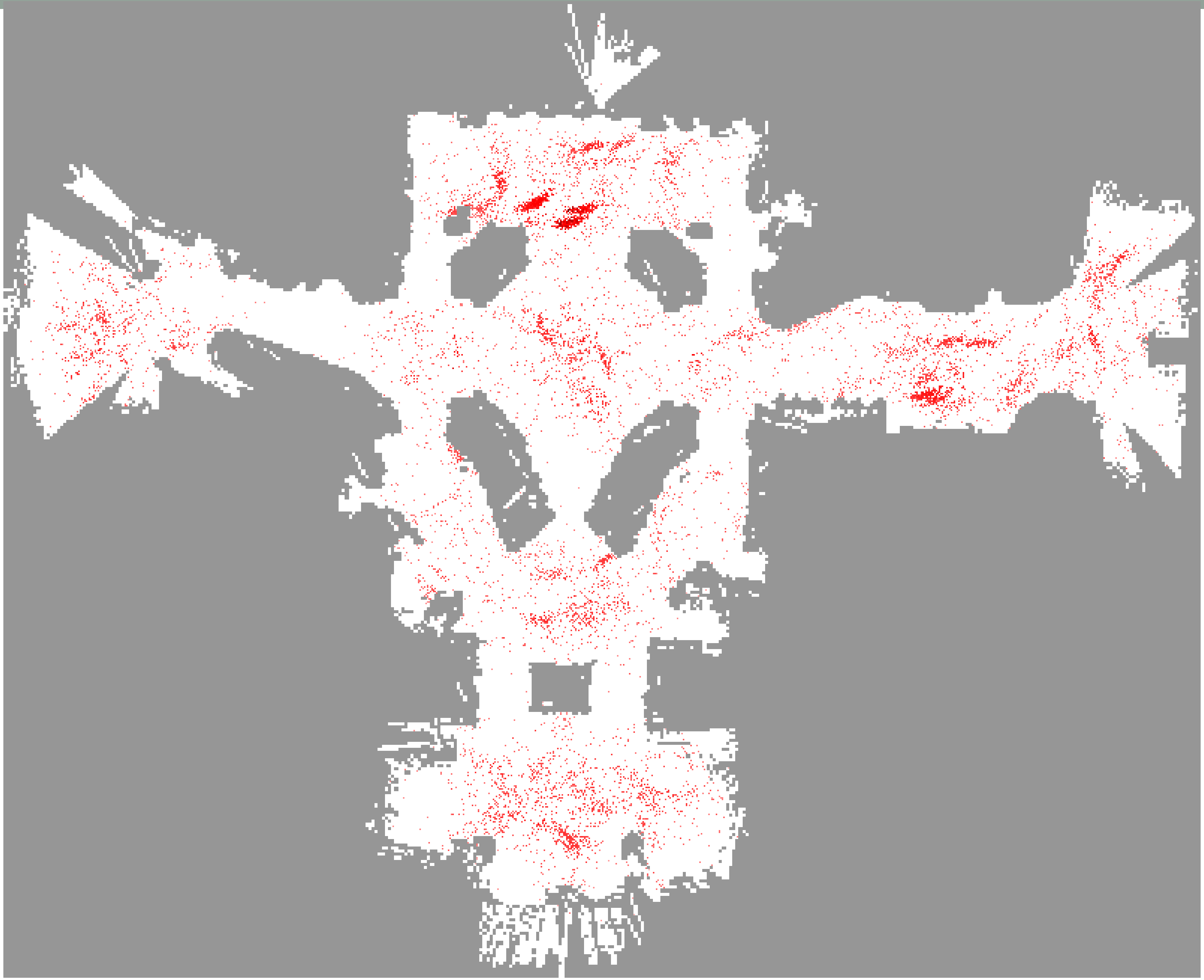


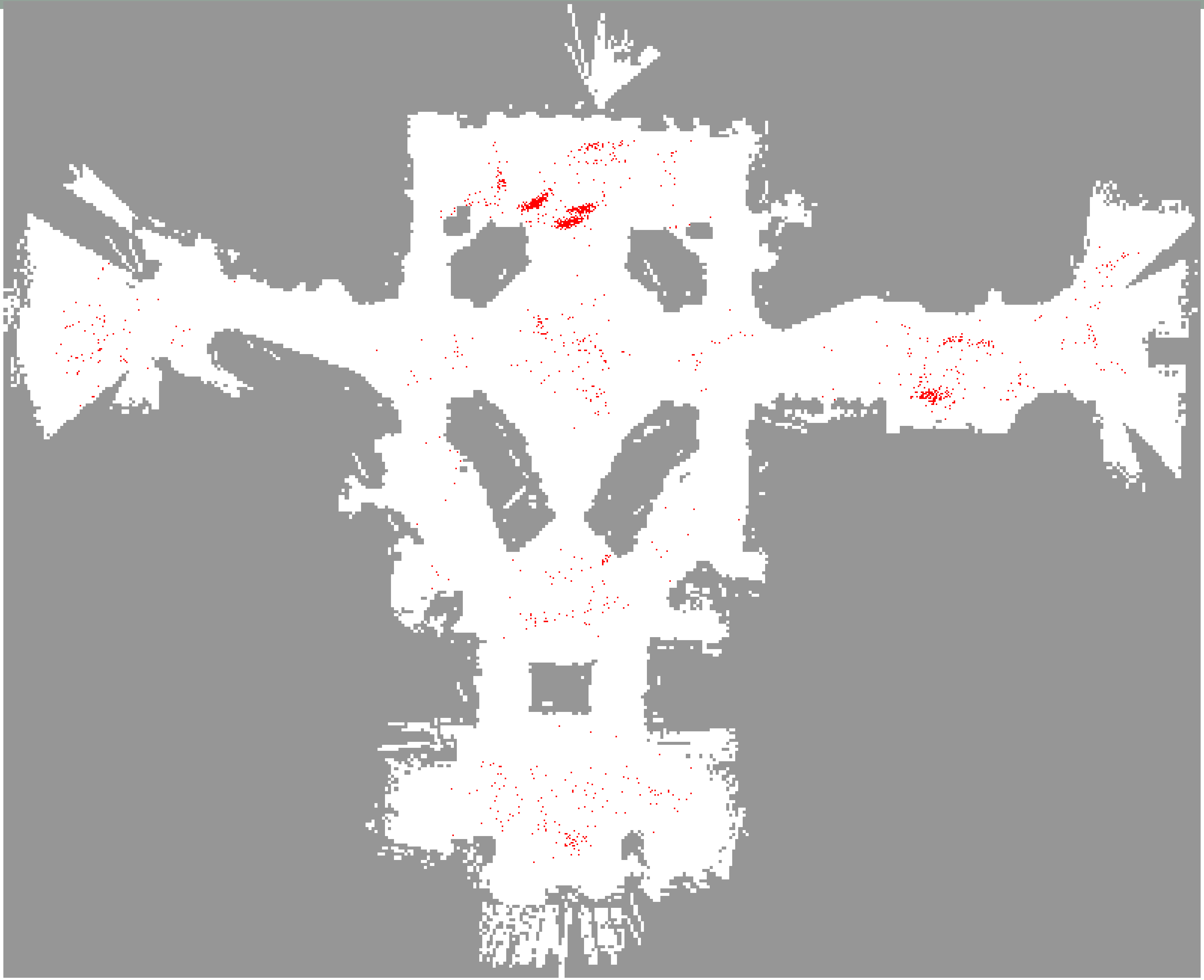


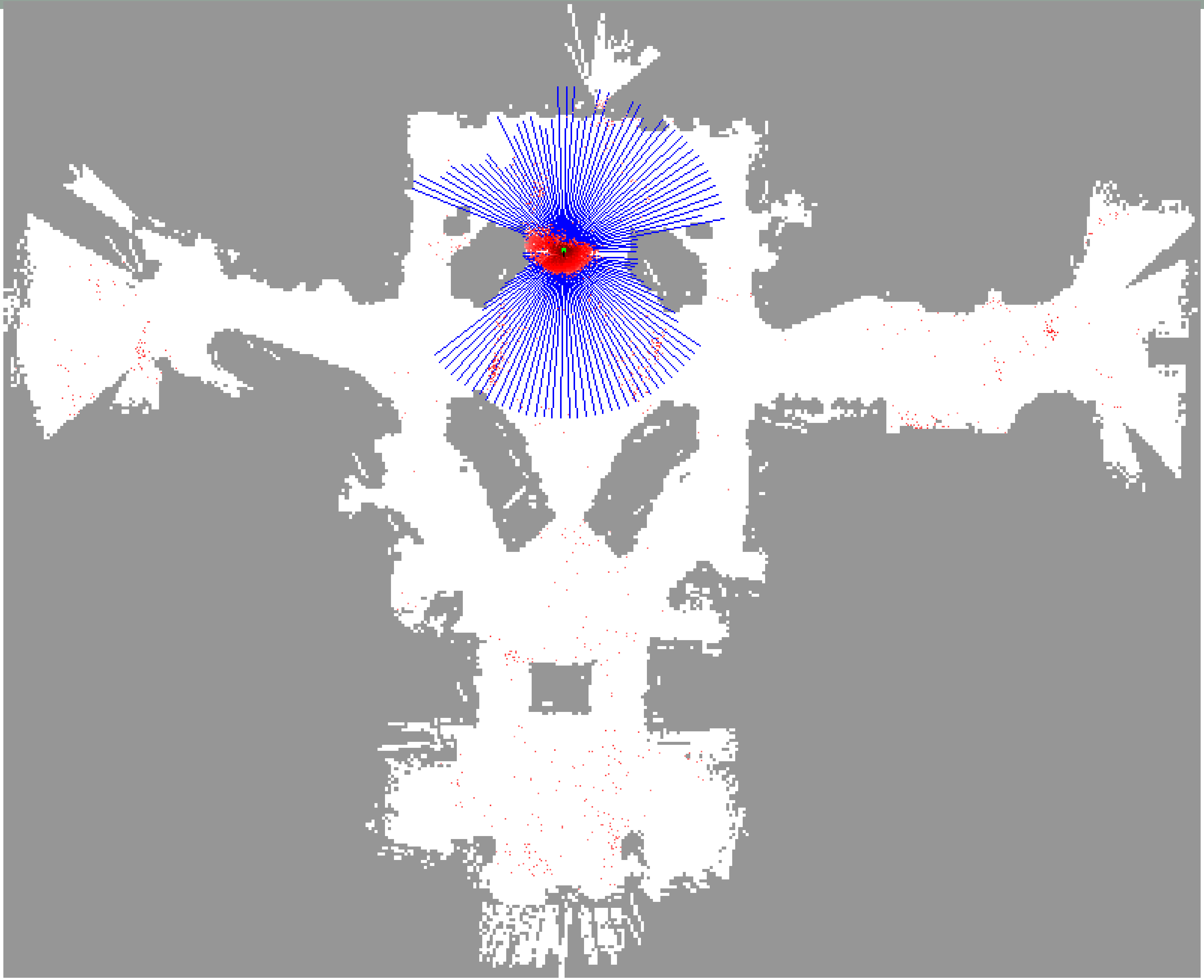


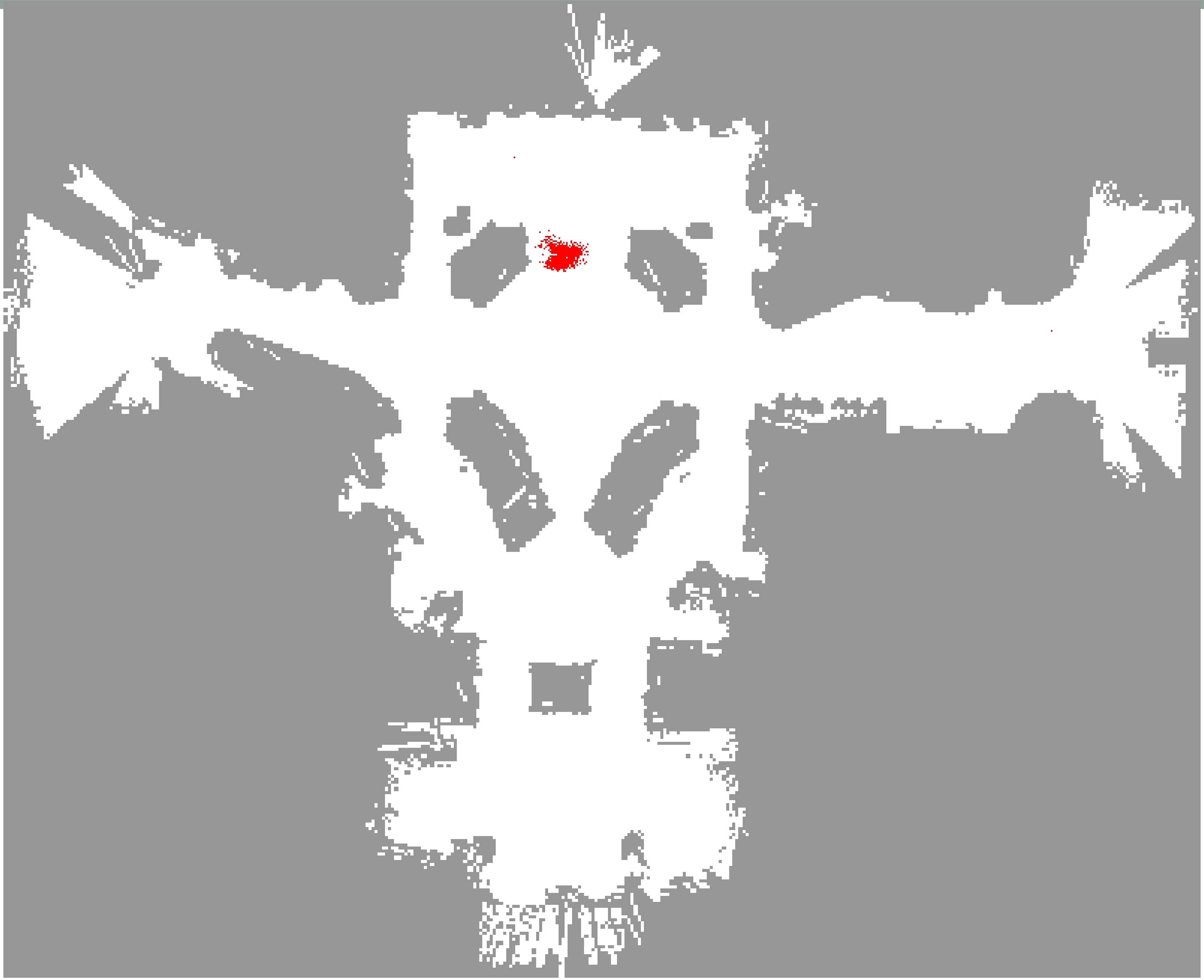


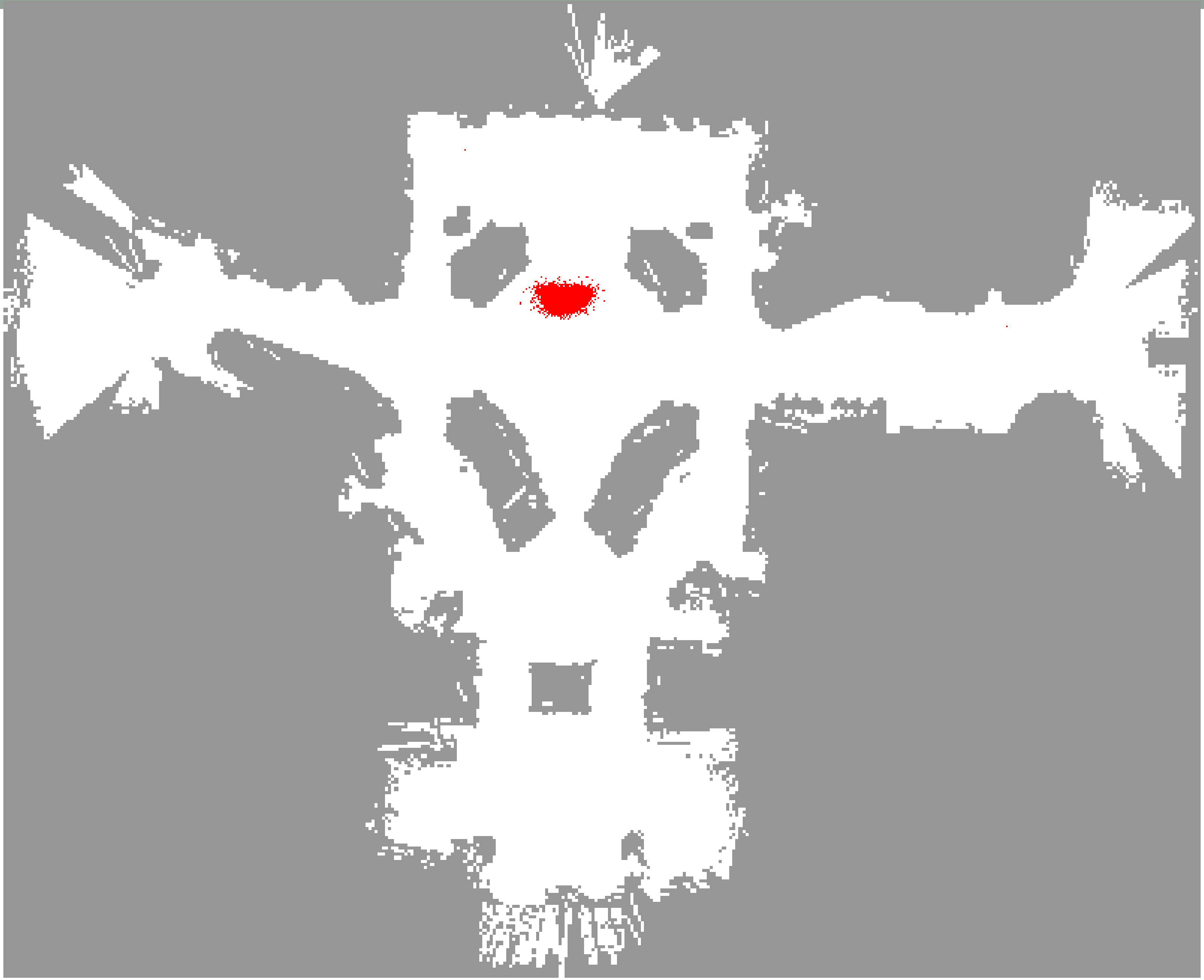


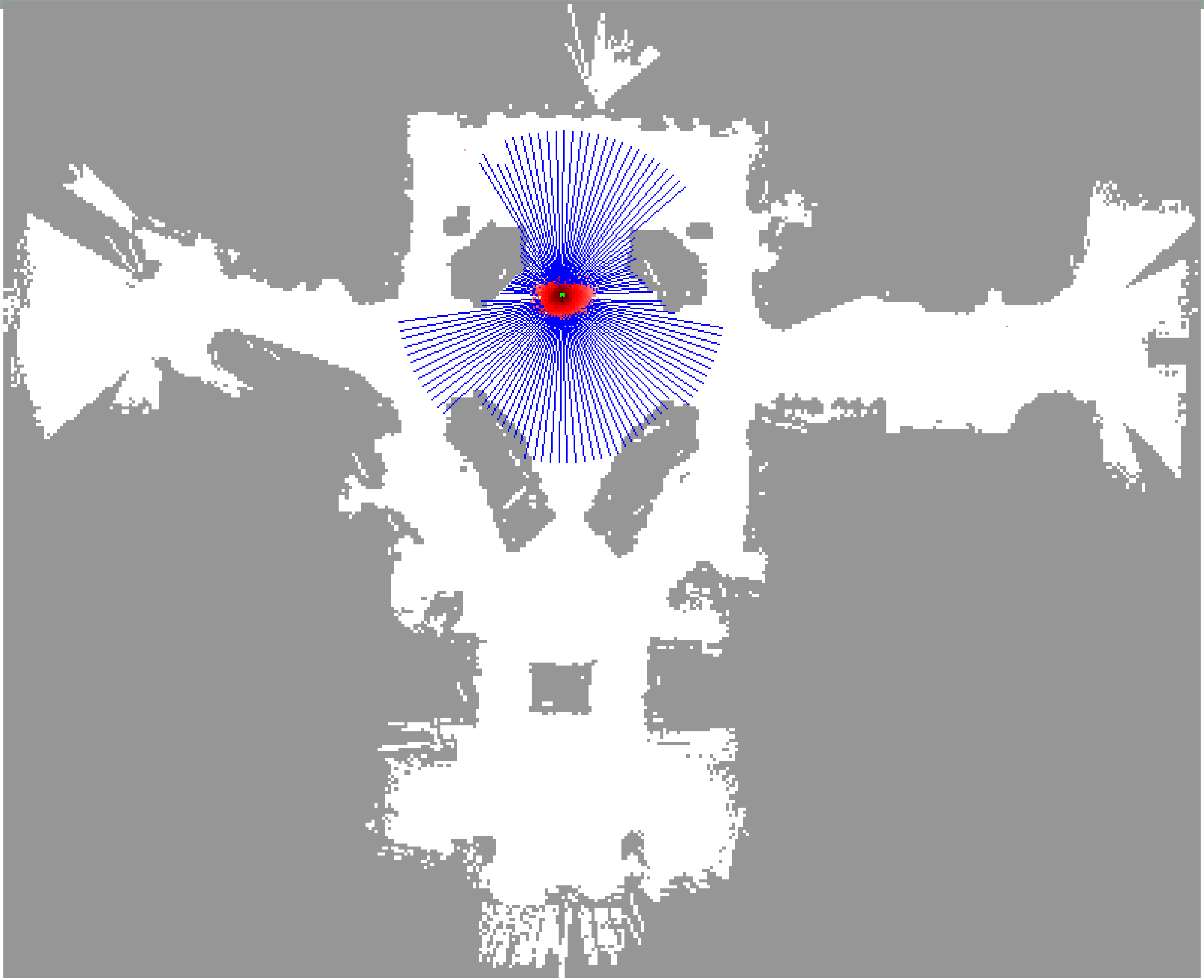


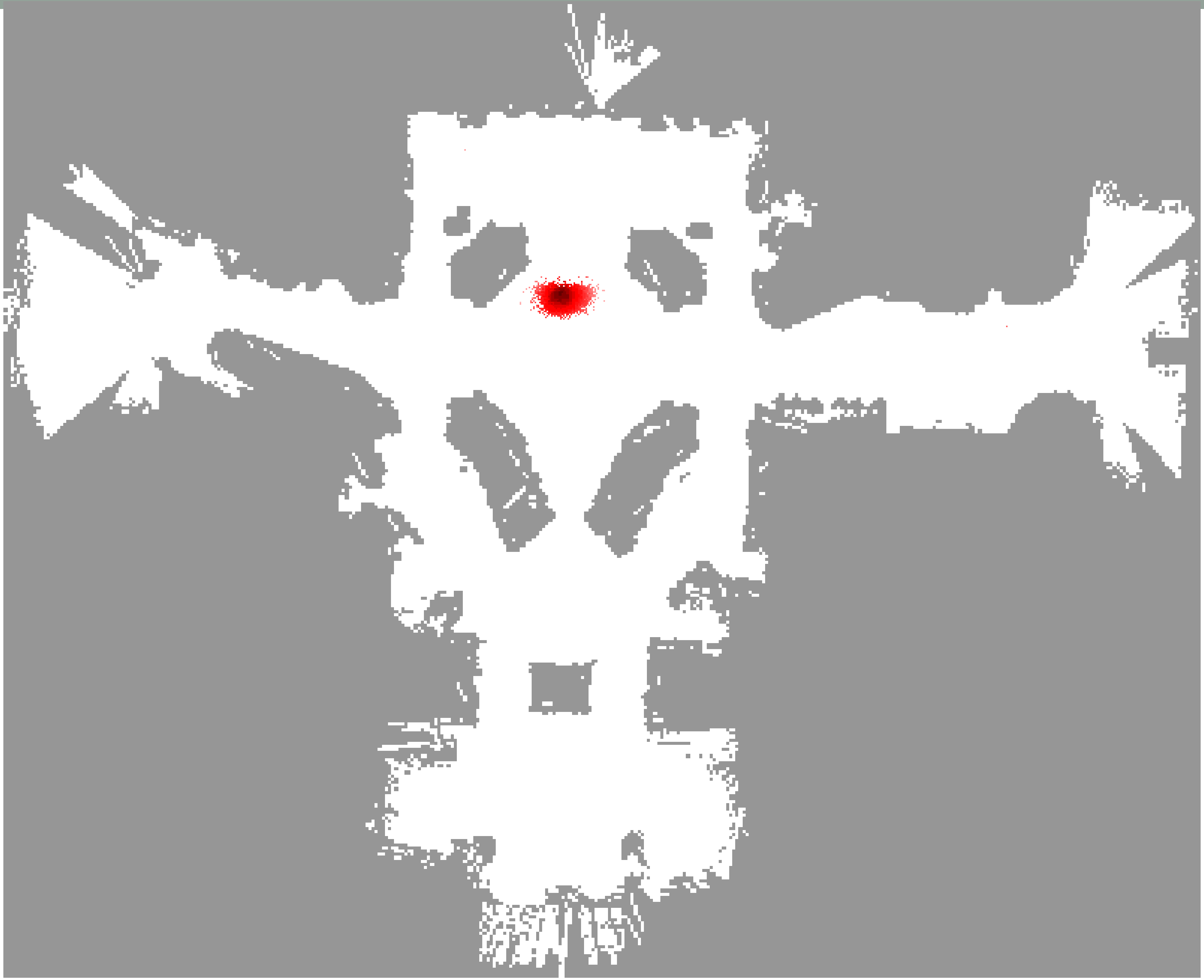


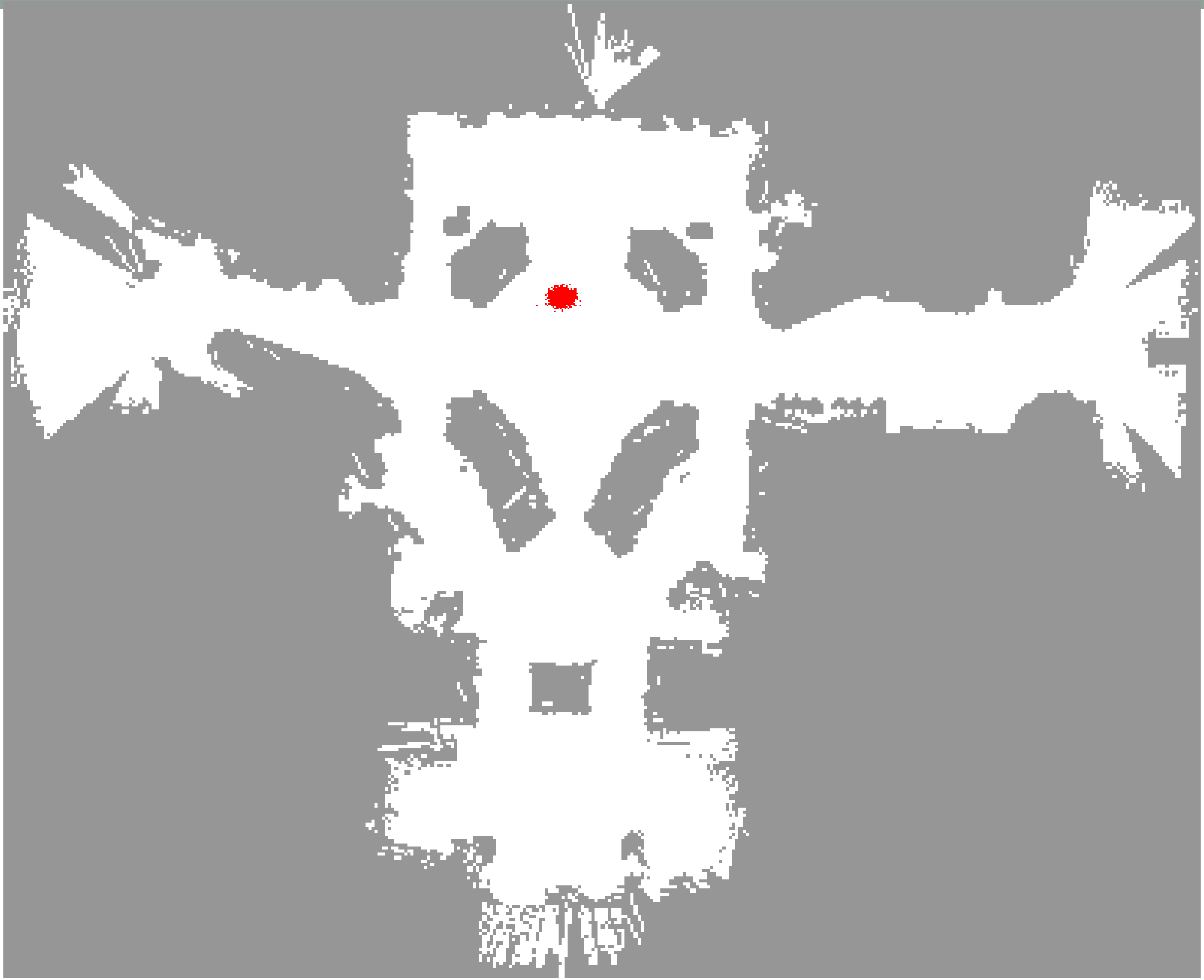


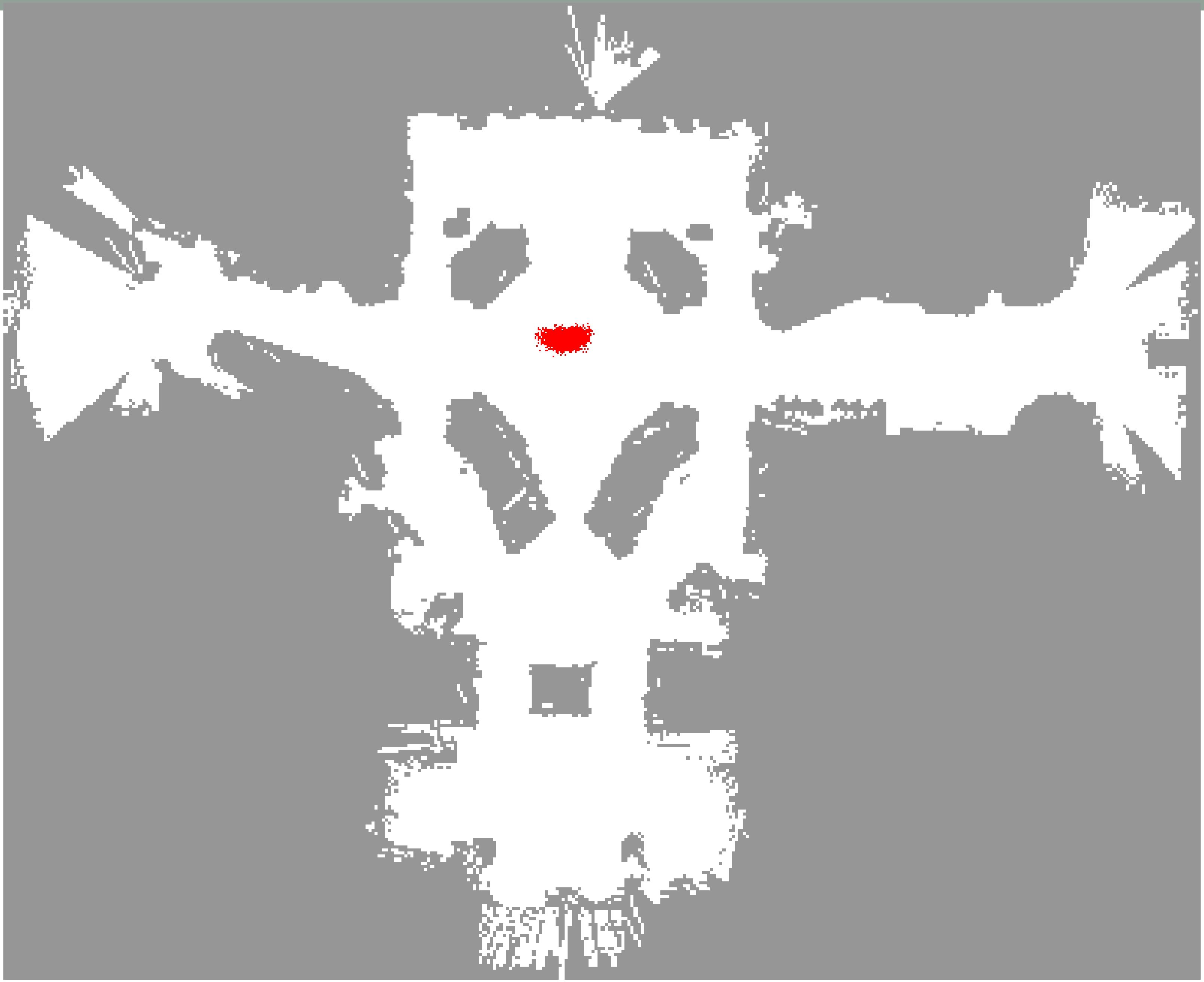


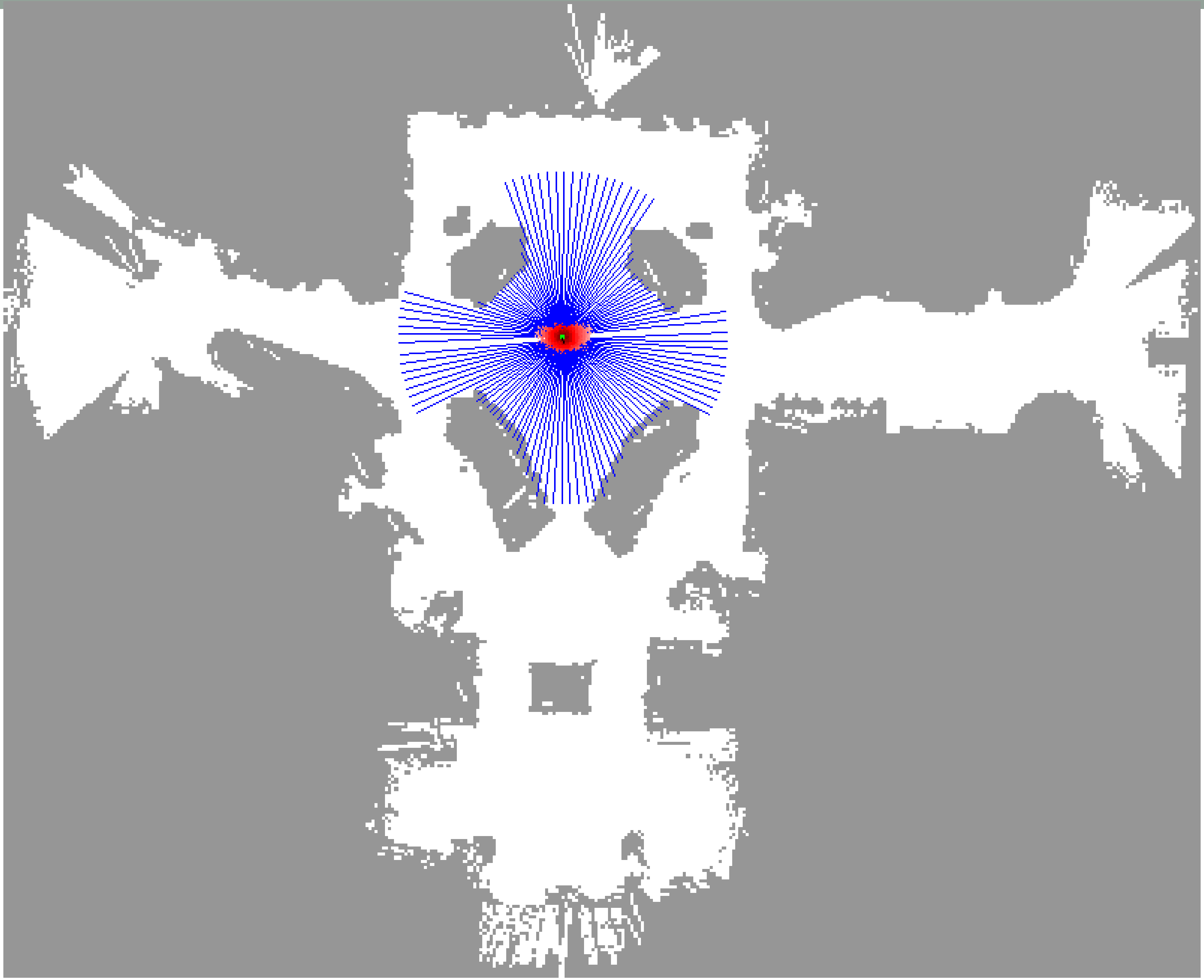






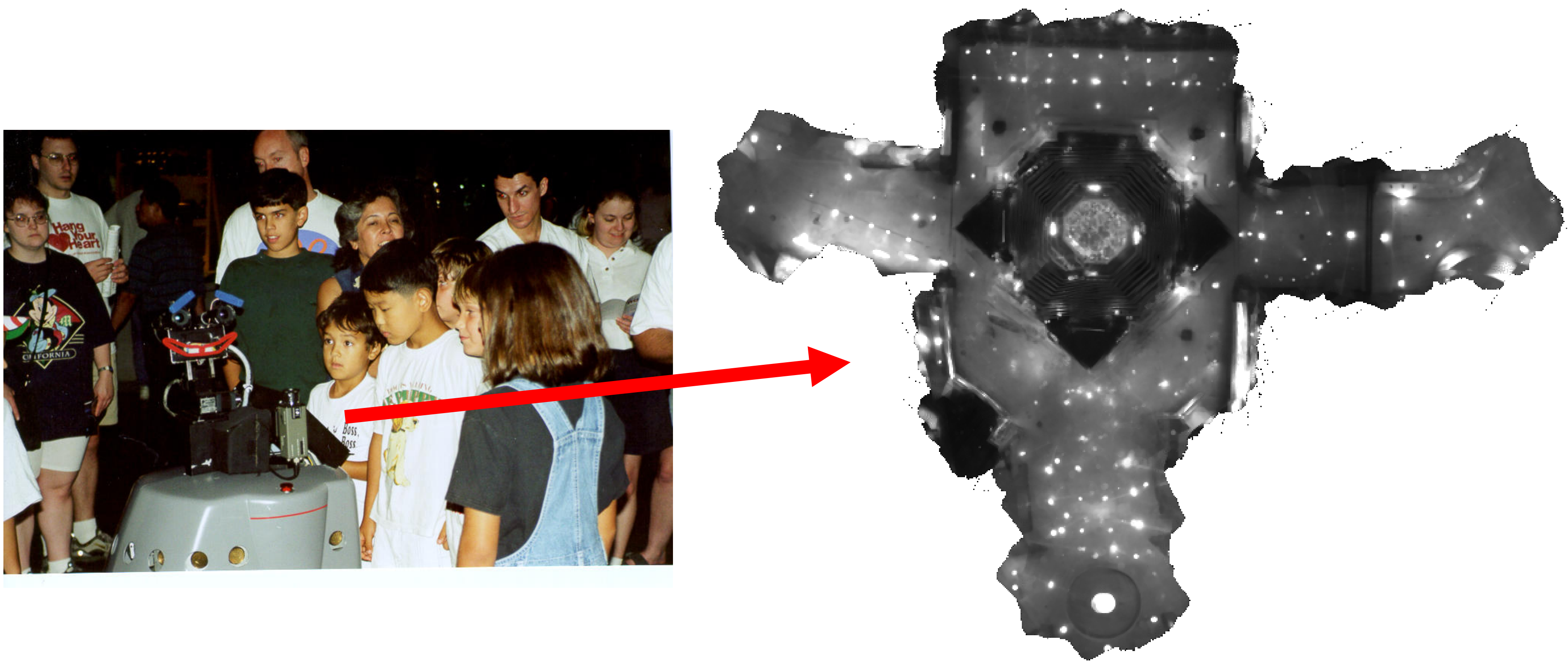






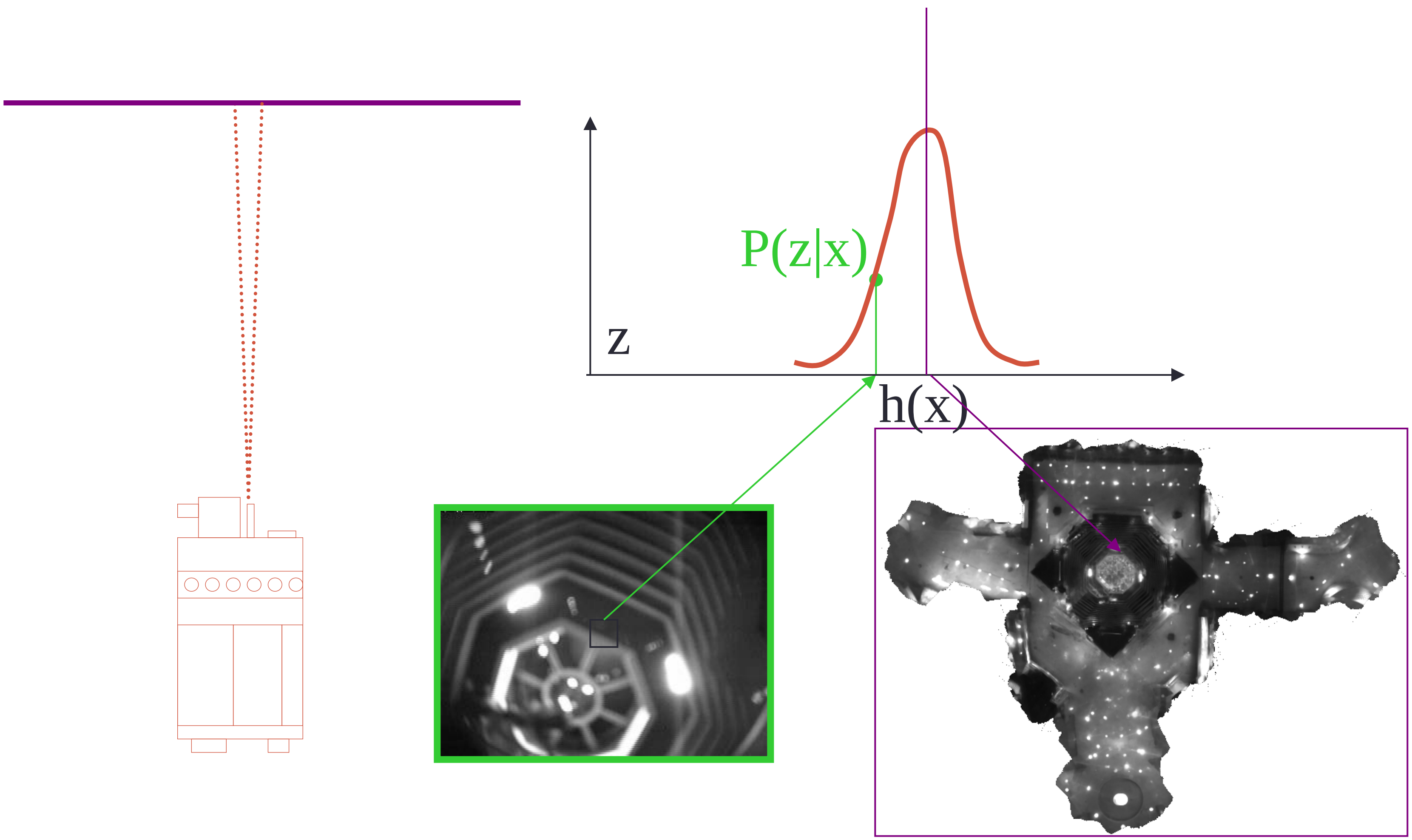
How about simple vision….

Using Ceiling Maps for Localization



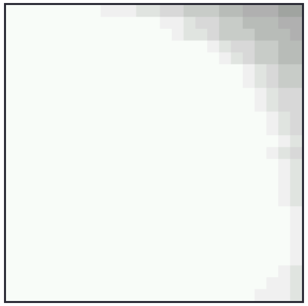
Dellaert, et al. 1997

Vision-based Localization

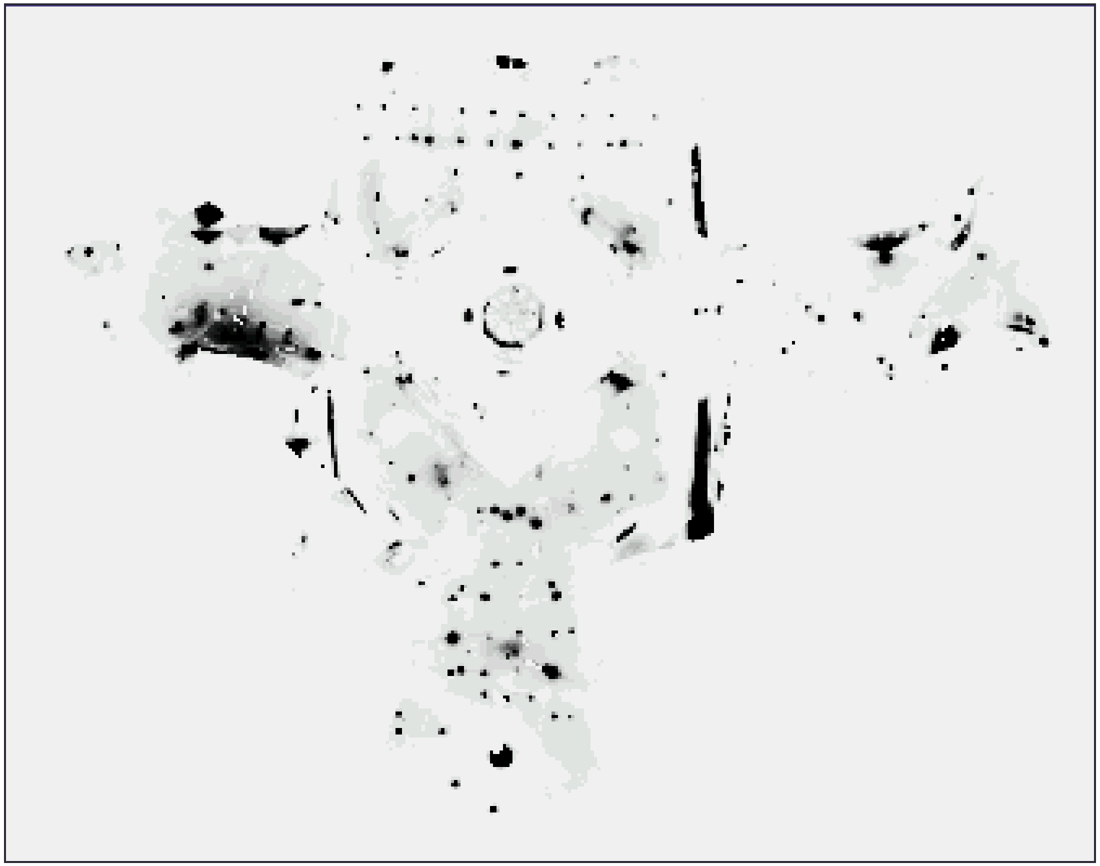


Under a Light

Measurement z:

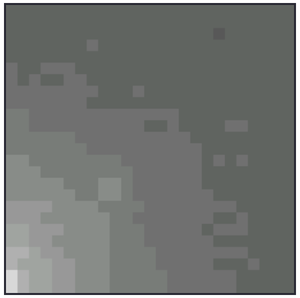


***P(z|x)*:**

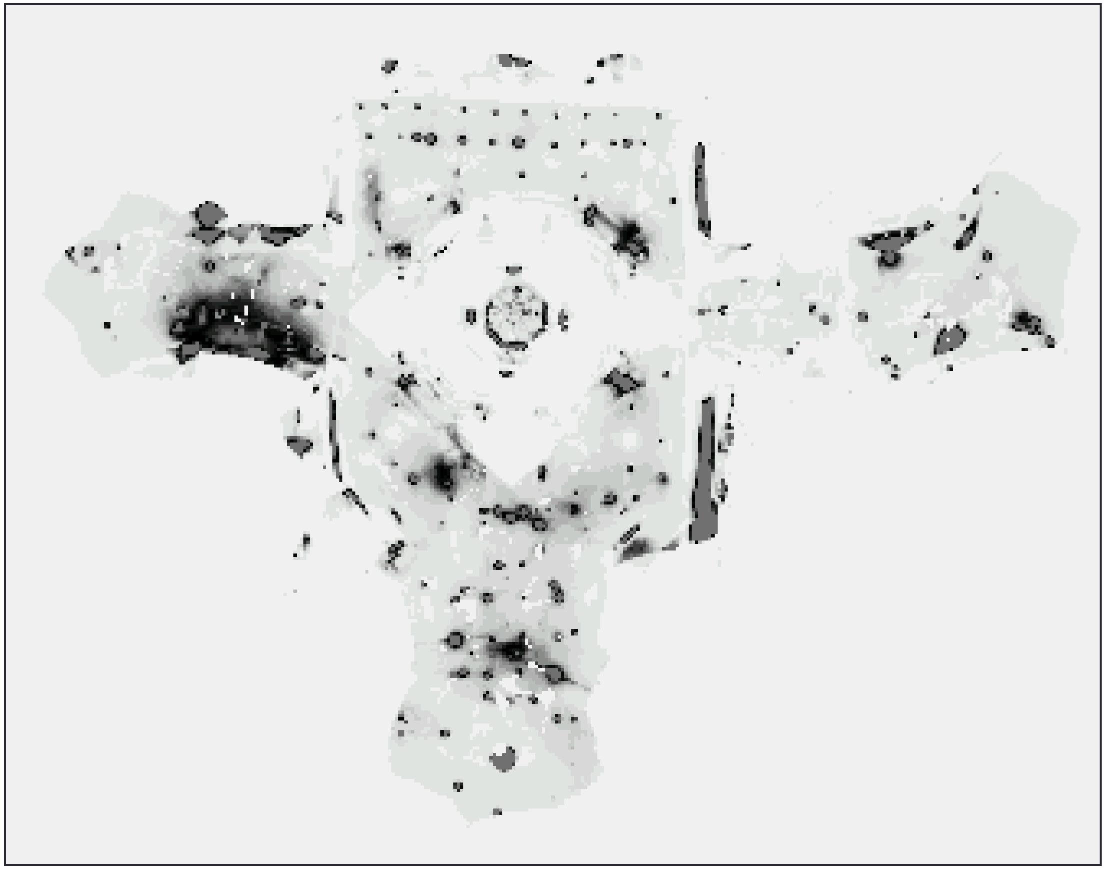


Next to a Light

Measurement z:



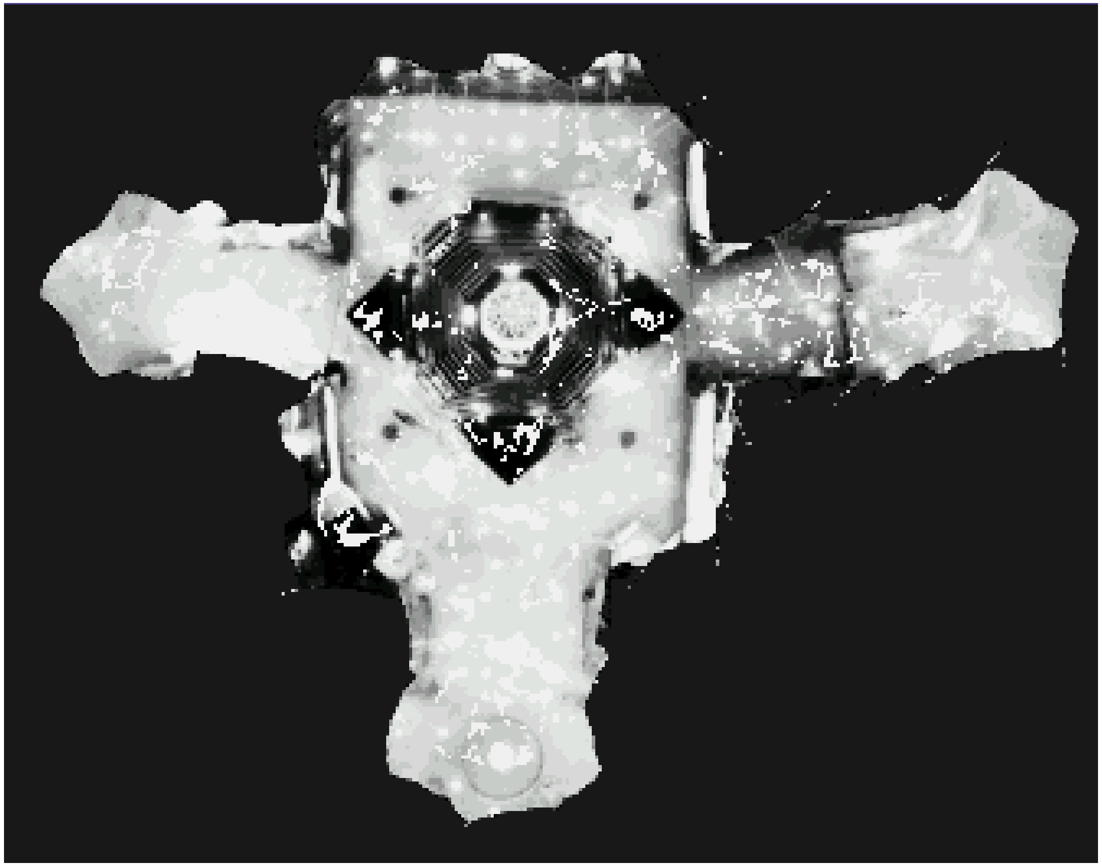
***P(z|x)*:**



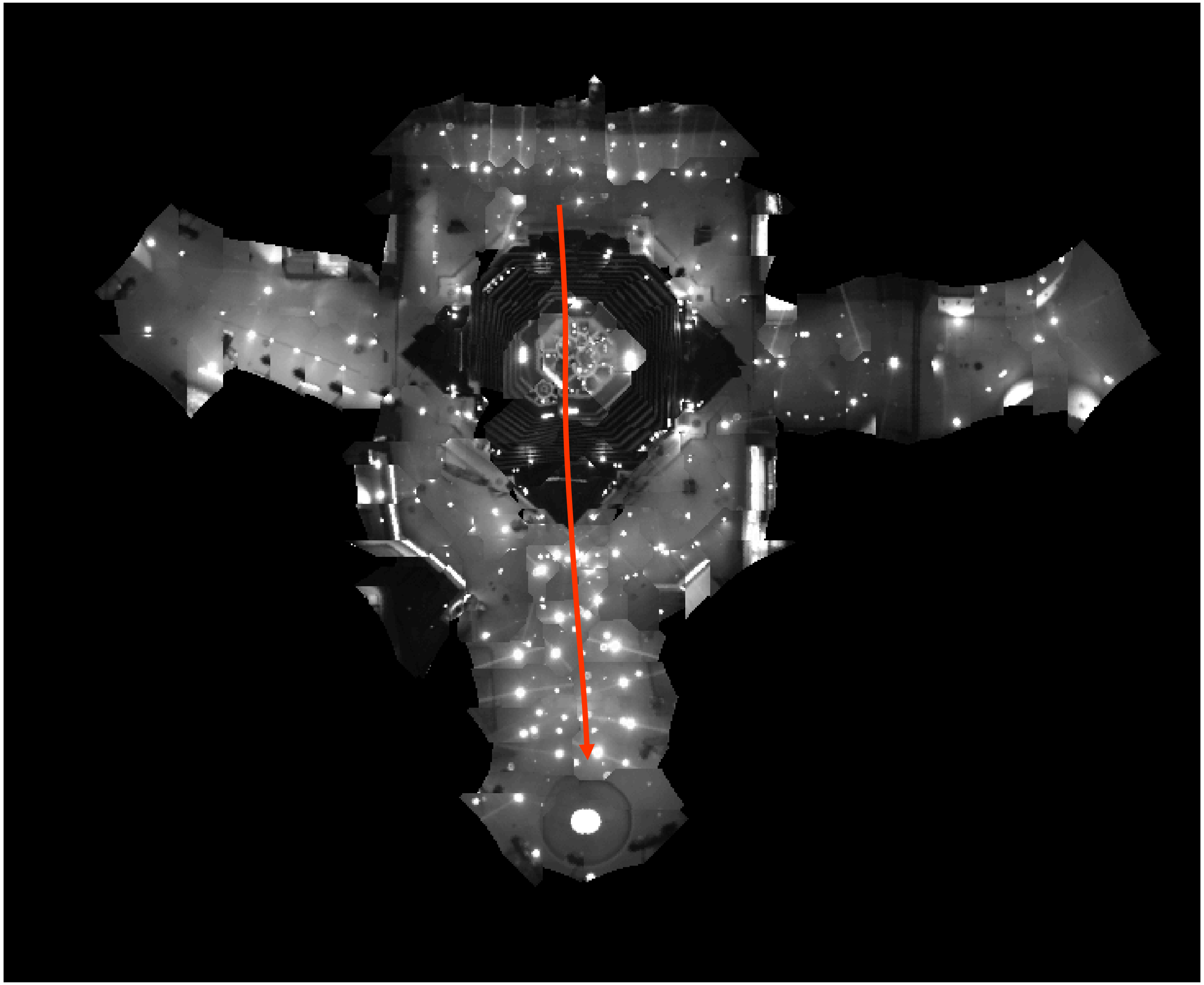
Elsewhere

Measurement z:

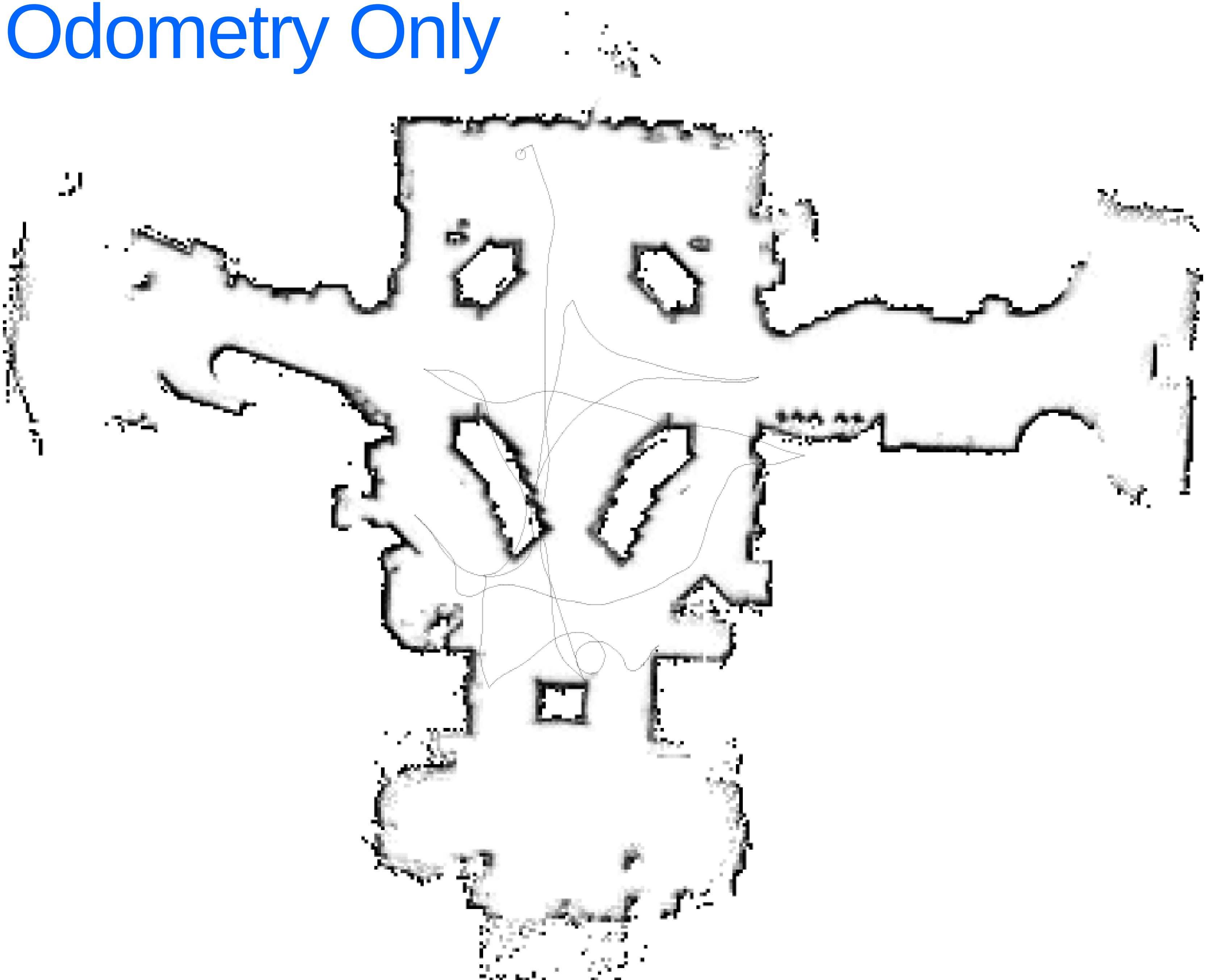
***P(z|x)*:**



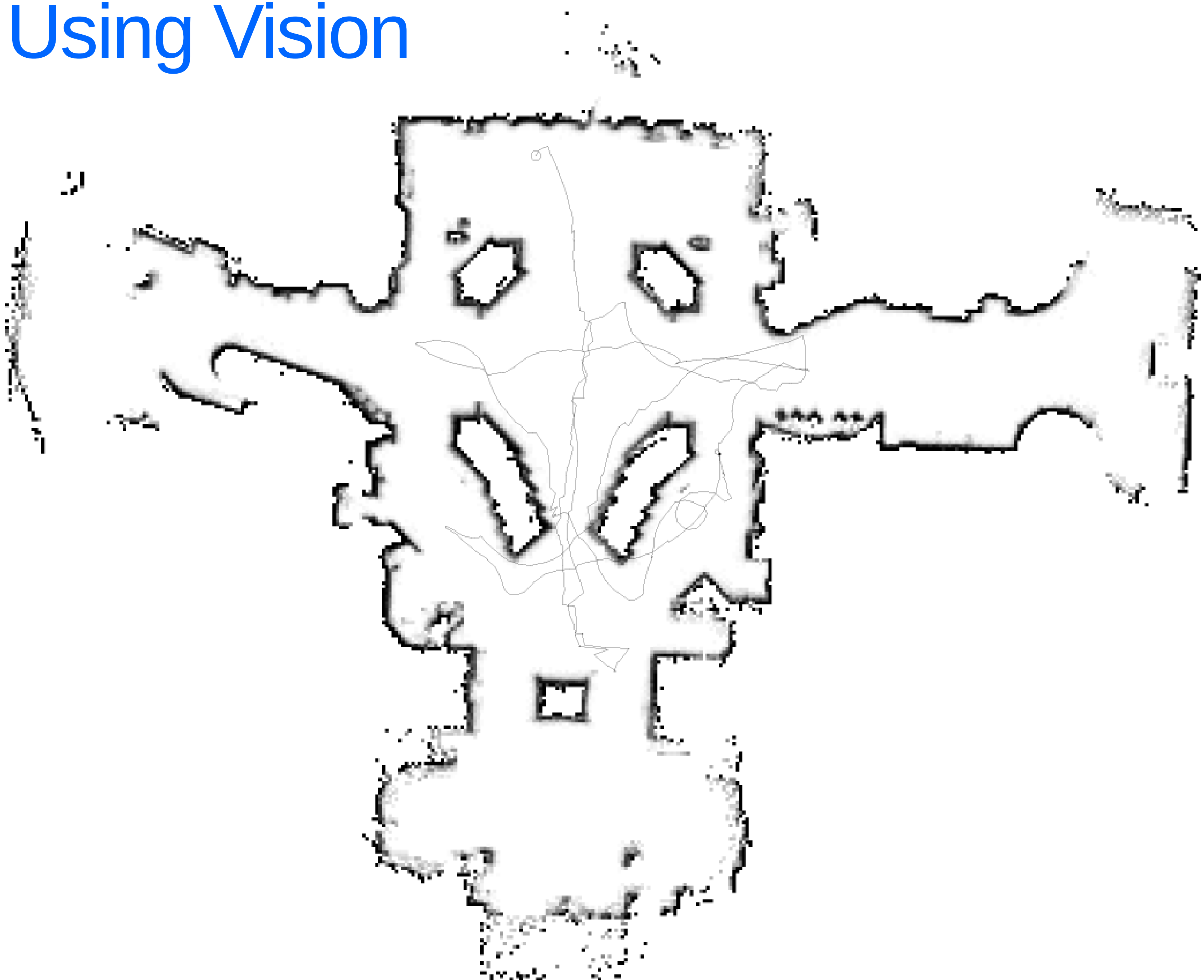
Global Localization Using Vision



Odometry Only



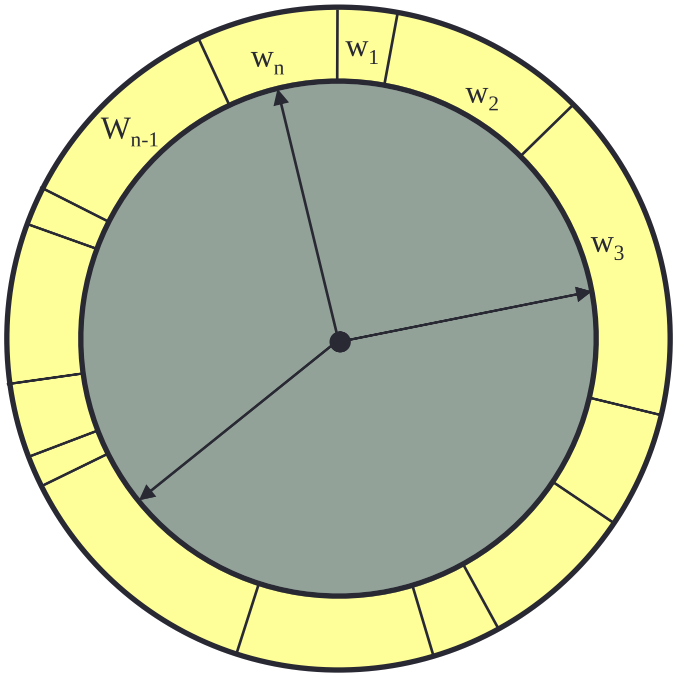
Using Vision



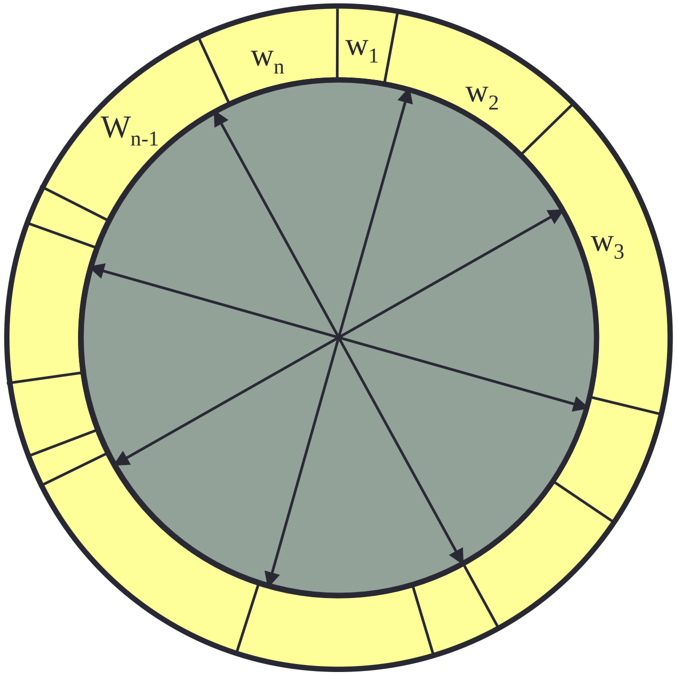
A detail: Resampling method can matter

- **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'.
  - *Or even not done except when needed…too many low weight particles.*

Resampling



- Roulette wheel
- Binary search, n log n



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

Resampling Algorithm

1. Algorithm **systematic_resampling**(S,n):

2. $S' = \emptyset, c_1 = w^1$
3. **For** $i = 2 \ldots n$ — ***Generate cdf***
4. $c_i = c_{i-1} + w^i$
5. $u_1 \sim U[0, n^{-1}], i = 1$ — ***Initialize threshold***

6. **For** $j = 1 \ldots n$ — ***Draw samples …***
7. **While** ($u_j > c_i$) — ***Skip until next threshold reached***
8. $i = i + 1$
9. $S' = S' \cup \left\{ < x^i, n^{-1} > \right\}$ — ***Insert***
10. $u_{j+1} = u_j + n^{-1}$ — ***Increment threshold***

11. **Return** S' *(Also called **stochastic universal sampling**)*

PF: Practical Considerations

- If dealing with 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
- Recover from failure by selectively adding samples from observations
- Recover from failure by uniformly adding some samples
- Can Resample only when necessary (efficiency of representation measured by variance of weights)

To do real tracking…

State

$$p(\underbrace{x_t \mid x_{t-1}, u_t}_{\text{State dynamics}})$$

$$p(x \mid z) = \frac{p(z \mid x)\, p(x)}{p(z)}$$

$$= \eta \underbrace{p(z \mid x)}_{\text{Sensor model}} p(x)$$

- ***x*** is the "state". But of what? The object? Some representation of the object?
- ***z*** is the "measurement". But what measurement? And how does it relate to the state?
- Where do you get your dynamics from?

The source…

International Journal of Computer Vision 29(1), 5–28 (1998)

CONDENSATION—Conditional Density Propagation for Visual Tracking

MICHAEL ISARD AND ANDREW BLAKE
Department of Engineering Science, University of Oxford, Oxford OX1 3PJ, UK

Received July 16, 1996; Accepted March 3, 1997

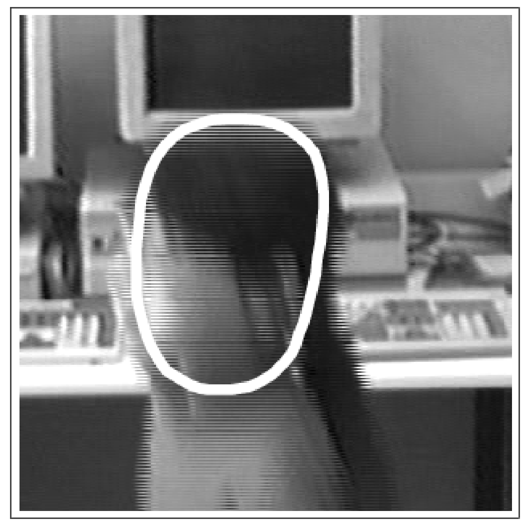
Abstract. The problem of tracking curves in dense visual clutter is challenging. Kalman filtering is inadequate because it is based on Gaussian densities which, being unimodal, cannot represent simultaneous alternative hypotheses. The CONDENSATION algorithm uses "factored sampling", previously applied to the interpretation of static images, in which the probability distribution of possible interpretations is represented by a randomly generated set. CONDENSATION uses learned dynamical models, together with visual observations, to propagate the random set over time. The result is highly robust tracking of agile motion. Notwithstanding the use of stochastic methods, the algorithm runs in near real-time.

Particle filter tracking - state

- The “object” to be tracked here is a hand initialized **contour**. Could have been the image pixels. Which is better?
- Its state is its affine deformation. How many parameters?
- Each particle represents those six parameters.

(a)

Picture of the states represented by the top weighted particles



(b)

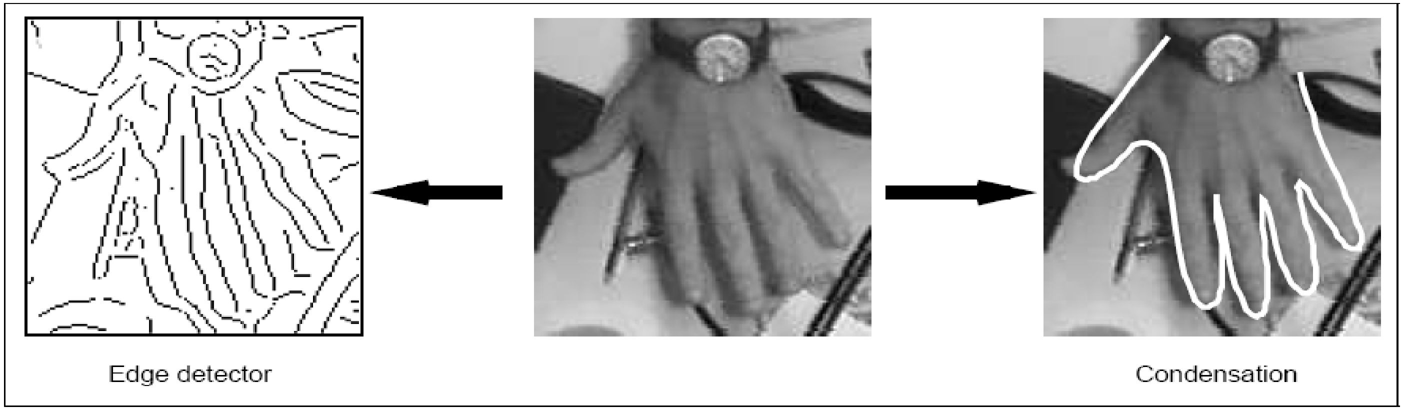
The mean state

[Isard 1998]

More complex state

- Tracking of a hand movement using an edge detector

- State is translation and rotation of hand plus angle of each finger; 12 DOF



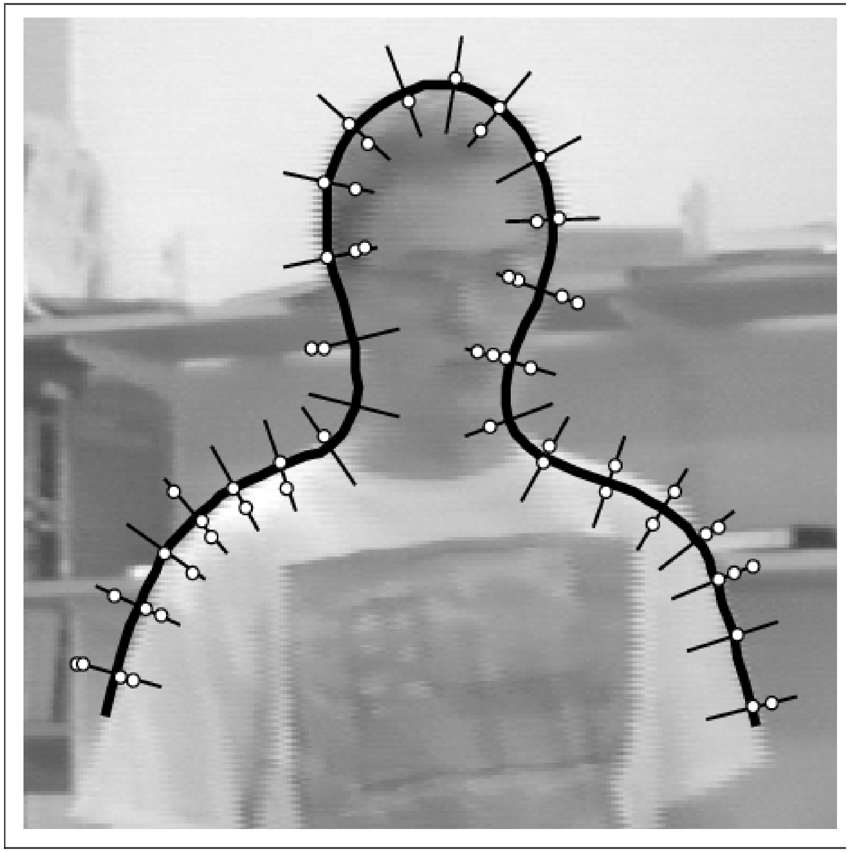
Particle filter tracking - measurement

- Suppose $\boldsymbol{x}$ is a hand initialized contour.
- What is $\boldsymbol{z}$?

$$p(\mathbf{z} \mid x) \propto \exp\left(-\frac{1}{2\sigma^2} f(\nu_1; \mu)\right)$$

$$\text{where } f(\nu; \mu) = \min(\nu^2, \mu^2), \qquad (14)$$

$$p(\mathbf{z} \mid \mathbf{x}) \propto \exp\left(-\sum_{m=1}^{M} \frac{1}{2rM} f(\mathbf{z}_1(s_m) - \mathbf{r}(s_m); \mu)\right), \qquad (16)$$

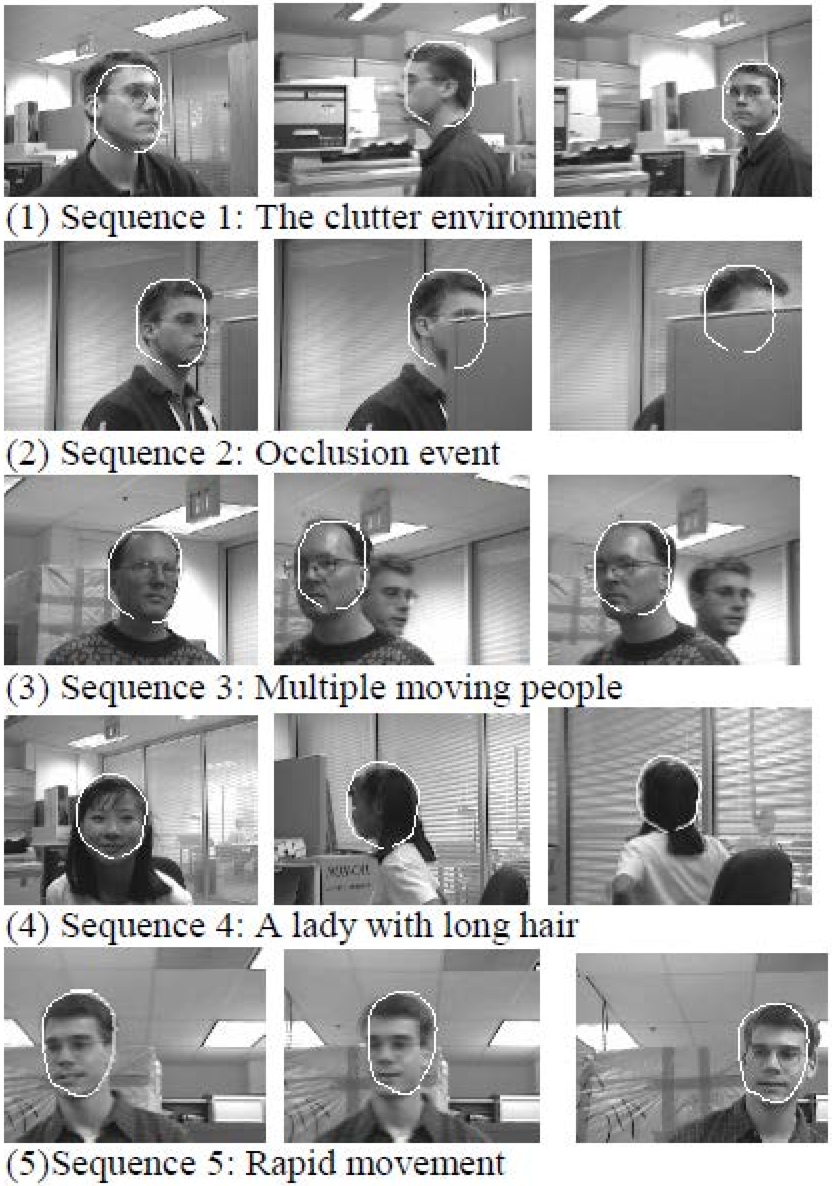


- Gaussian in Distance to nearest high-contrast feature summed over the contour.

[Isard 1998]

More tracking contours

- Head tracking with contour models (Zhihong et al. 2002)
- How did it do occlusion?
  - With velocity?
  - Without velocity?
- How did you get "dynamics"?



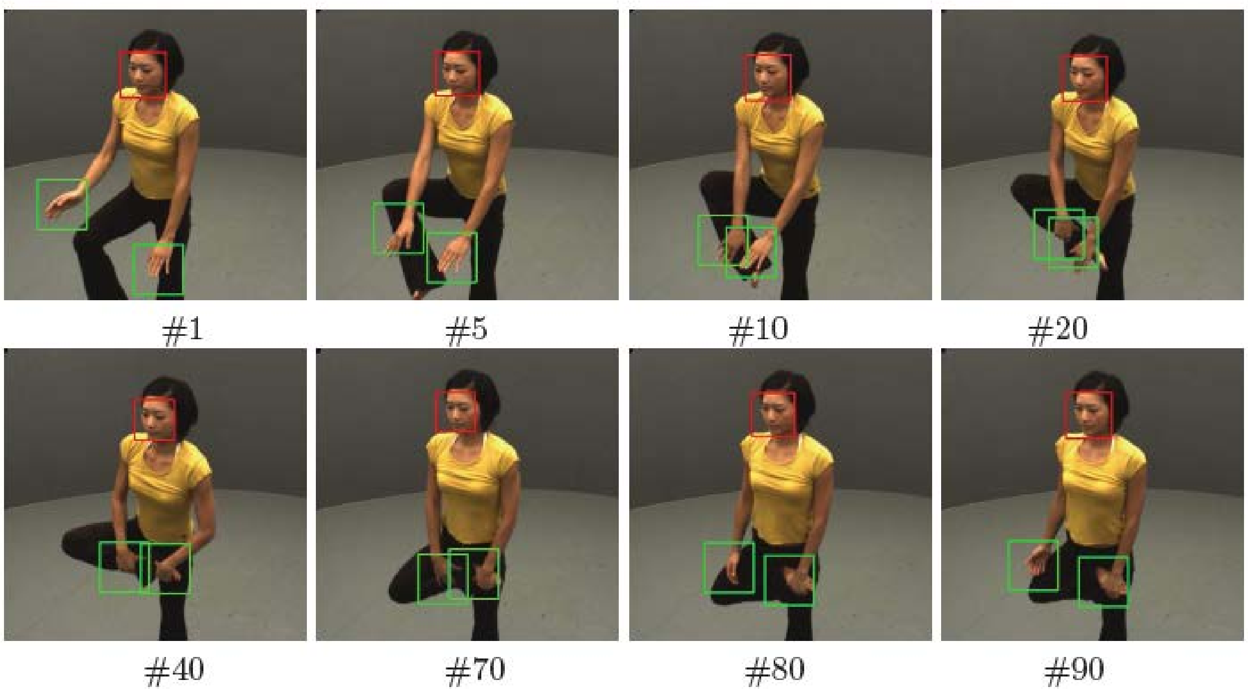
(1) Sequence 1: The clutter environment
(2) Sequence 2: Occlusion event
(3) Sequence 3: Multiple moving people
(4) Sequence 4: A lady with long hair
(5)Sequence 5: Rapid movement

Getting the dynamics

- Why are we doing all this work? 'Cuz tracking is hard.
- If it weren't hard we could just detect the contour.
- If we could just detect the contour we could get out correct trajectories,
- If we can get correct trajectories we can….?

A different model

- Hands and head movement tracking using color models and optical flow (Tung et al. 2008)
- State: location of colored blob (x,y)
- Prediction based upon flow.
- Sensor model: color match



How about a really, really simple model?

- State is just location of an image patch: x,y
- Dynamics: just random noise
- Sensor model: avg squared difference of pixel intensities.
  - Really a similarity model: more similar is more likely.

- Oh, you need a patch…

An even better model

- Suppose you want to track a region of colors.
- What would be a good model/State:
  - Location
  - Region size?
  - *Distribution of colors*
- What would be a good sensor model?
  - *Similarity of distributions*

Recapping: Tracking issues

- Initialization
  - Manual
  - Background subtraction
  - Detection

Recapping: Tracking issues

- Initialization
- Obtaining observation and dynamics model
  - Dynamics model: learn (difficult) or specify using domain knowledge
    - Can cheat if you have "easy" tracking case
  - Generative observation model: "render" the state on top of the image and compare. E.g. put down the contour and evaluate.

Recapping: Tracking issues

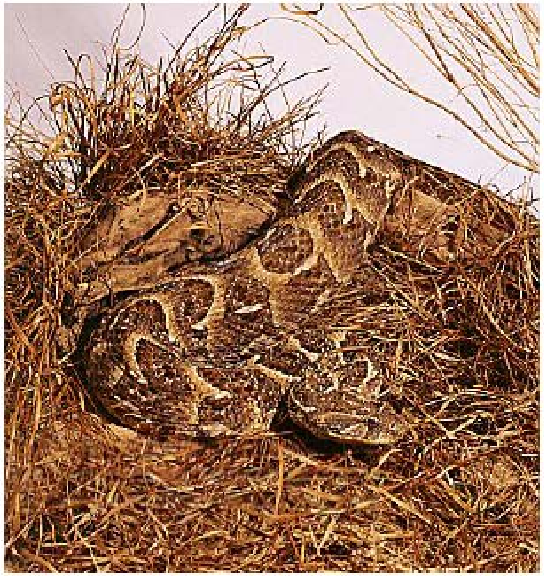
- Initialization
- Obtaining observation and dynamics model
- Prediction vs. correction
  - If the dynamics model noise is too low, will end up ignoring the data
  - If the observation noise model is too low, tracking is reduced to repeated detection (Kalman). If too peaked, only a few particles survive (PF).

Recapping: Tracking issues

- Initialization
- Obtaining observation and dynamics model
- Prediction vs. correction
- Data association
    - What if we don't know which measurements to associate with which tracks?

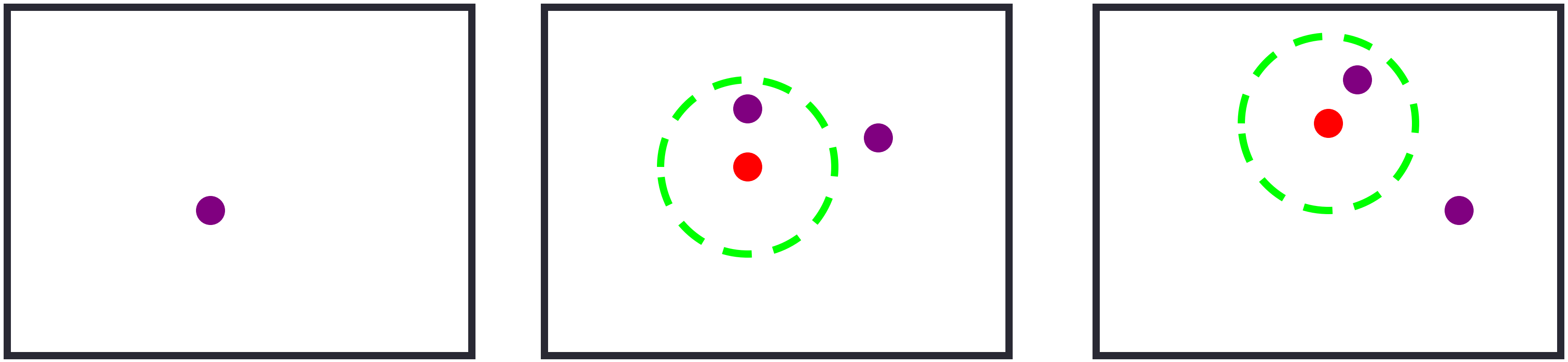
Data association

- So far, we've assumed the entire measurement to be relevant to determining the state
- In reality, there may be uninformative measurements (clutter) or measurements may belong to different tracked objects
- **Data association**: task of determining which measurements go with which tracks



Data association

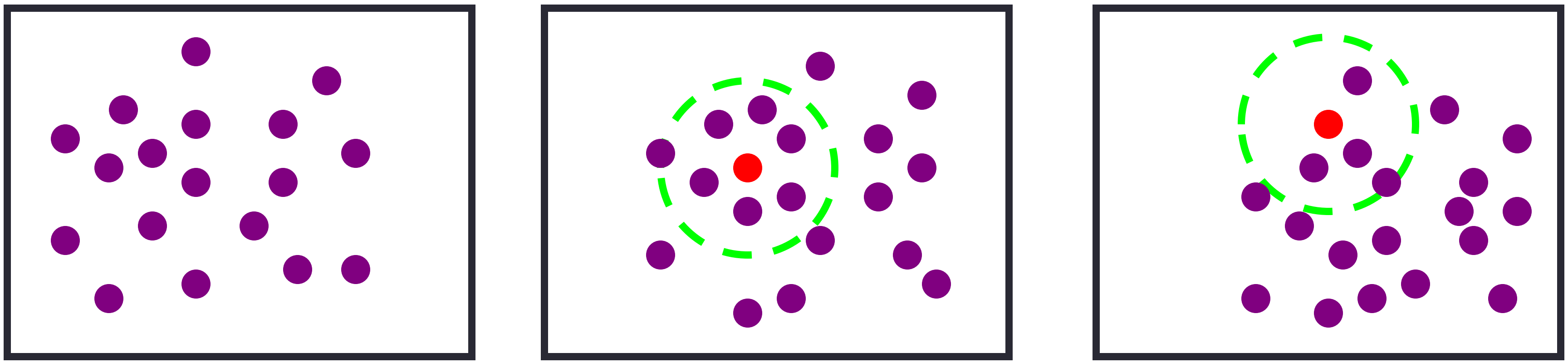
- Simple strategy: only pay attention to the measurement that is “closest” to the prediction



Source: Lana Lazebnik

Data association

- Simple strategy: only pay attention to the measurement that is “closest” to the prediction



Doesn’t always work…

Alternative: keep track of **multiple hypotheses** at once…

Source: Lana Lazebnik

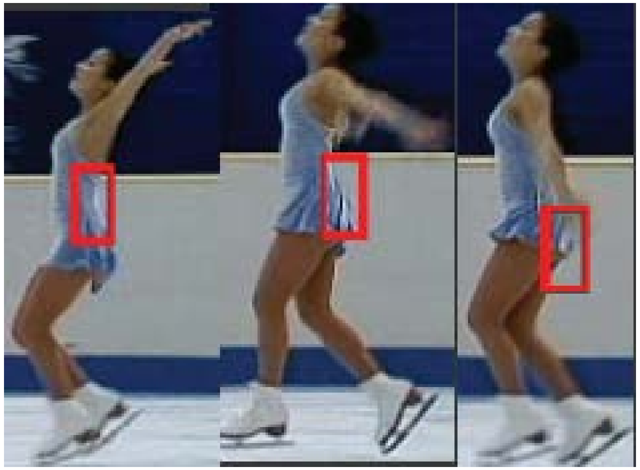
Data association

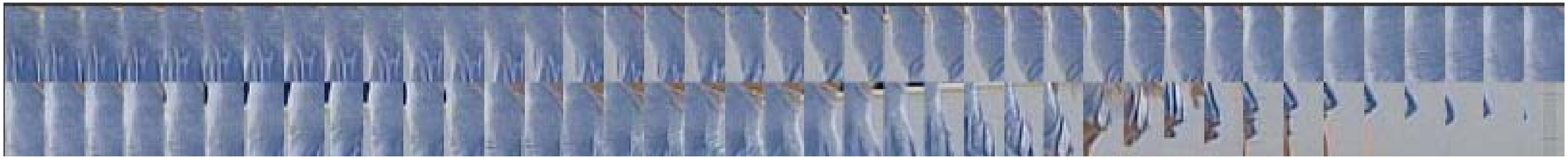
- Simple strategy: only pay attention to the measurement that is “closest” to the prediction
- More sophisticated strategy: keep track of multiple state/observation hypotheses
  - Can be done with particle filtering
- This is a general problem in computer vision, there is no easy solution

Tracking issues

- Initialization
- Obtaining observation and dynamics model
- Prediction vs. correction
- Data association
- Drift
  - Errors caused by dynamical model, observation model, and data association tend to accumulate over time

Drift





D. Ramanan, D. Forsyth, and A. Zisserman. [Tracking People by Learning their Appearance](). PAMI 2007.