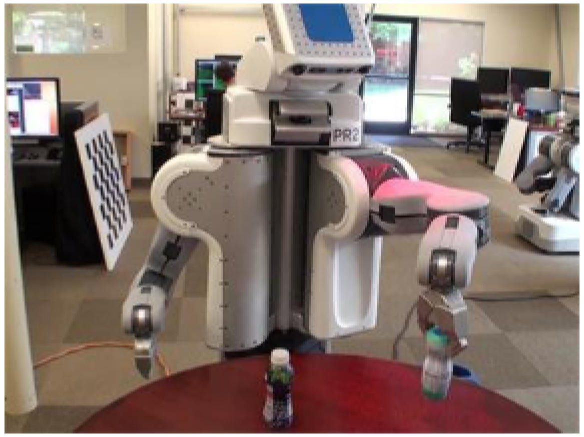


COMP 150: Probabilistic Robotics for Human-Robot Interaction

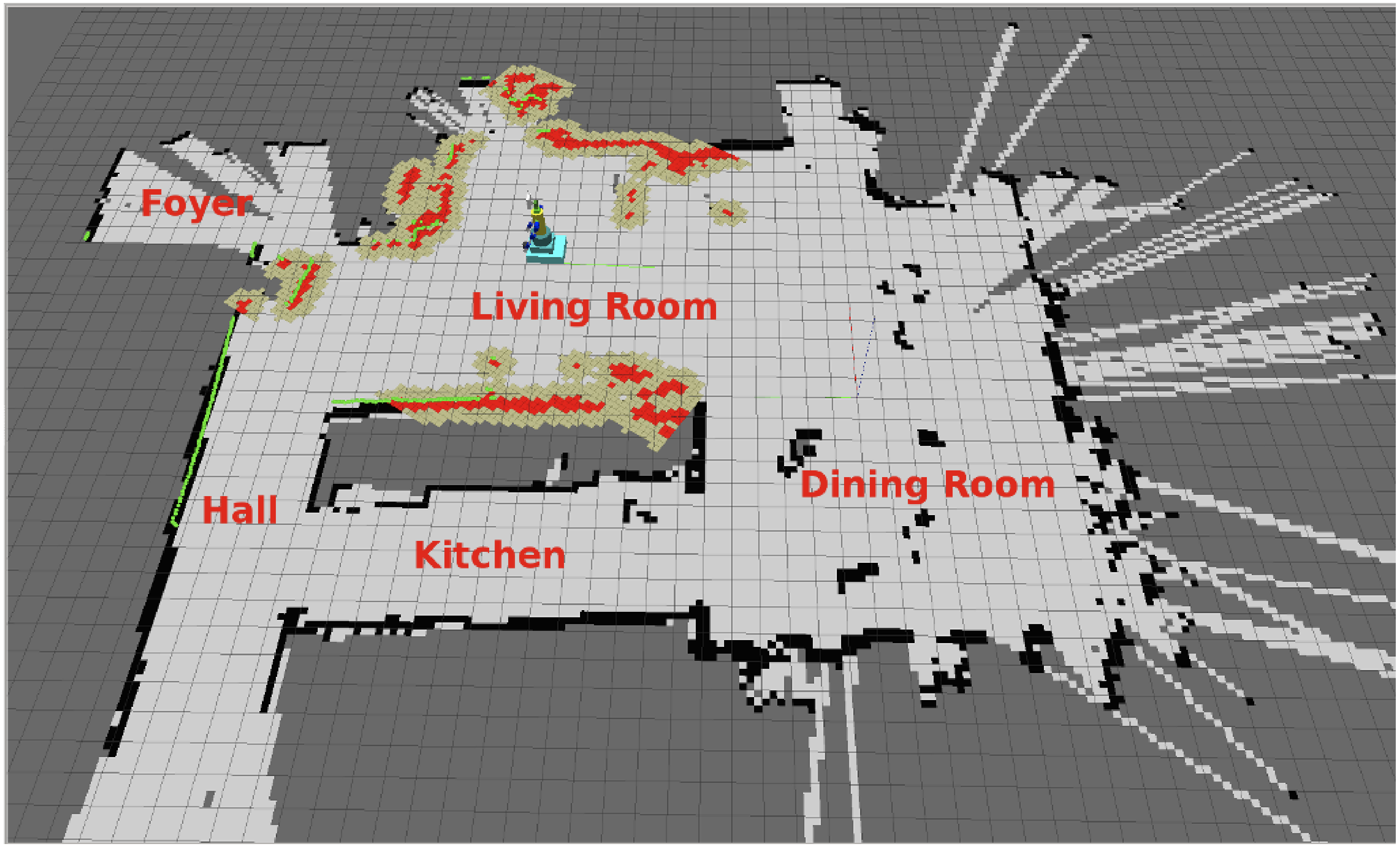
Instructor: Jivko Sinapov
www.cs.tufts.edu/~jsinapov

Today

- Homework 1 hints
- Introduction for Particle Filters for localization

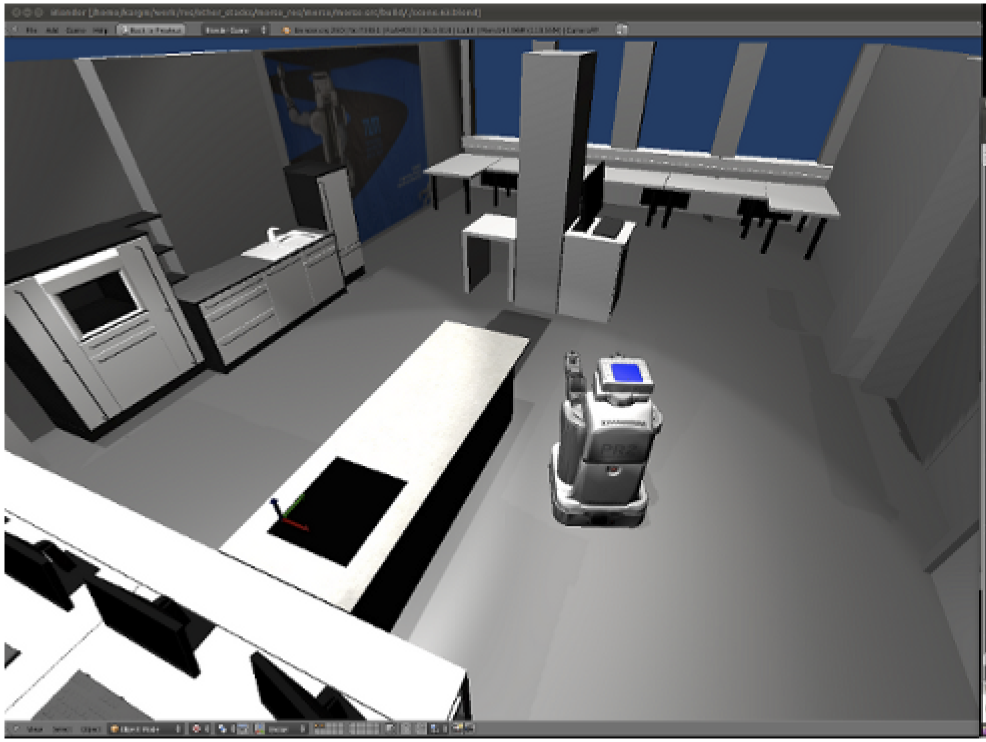
Localization and Mapping

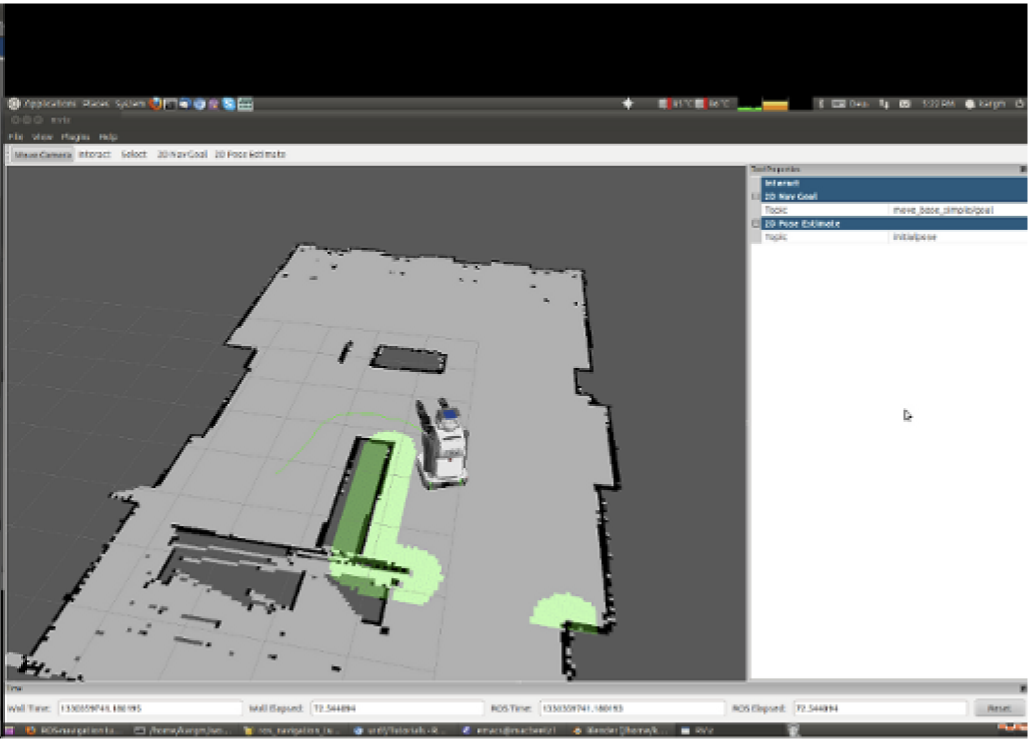
Robot Maps



[https://www.pirobot.org/blog/0015/map-1b.png]

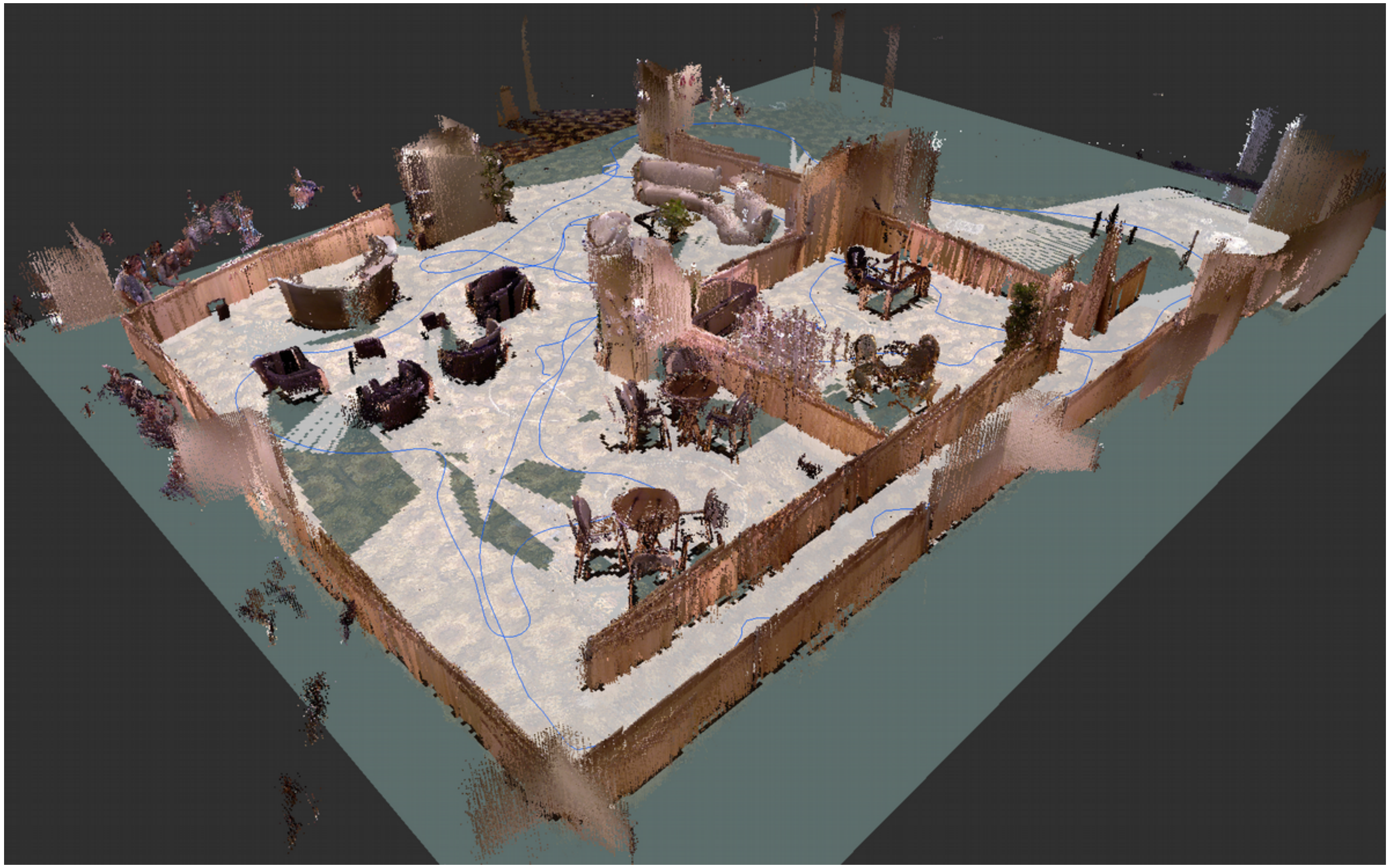
Robot Maps





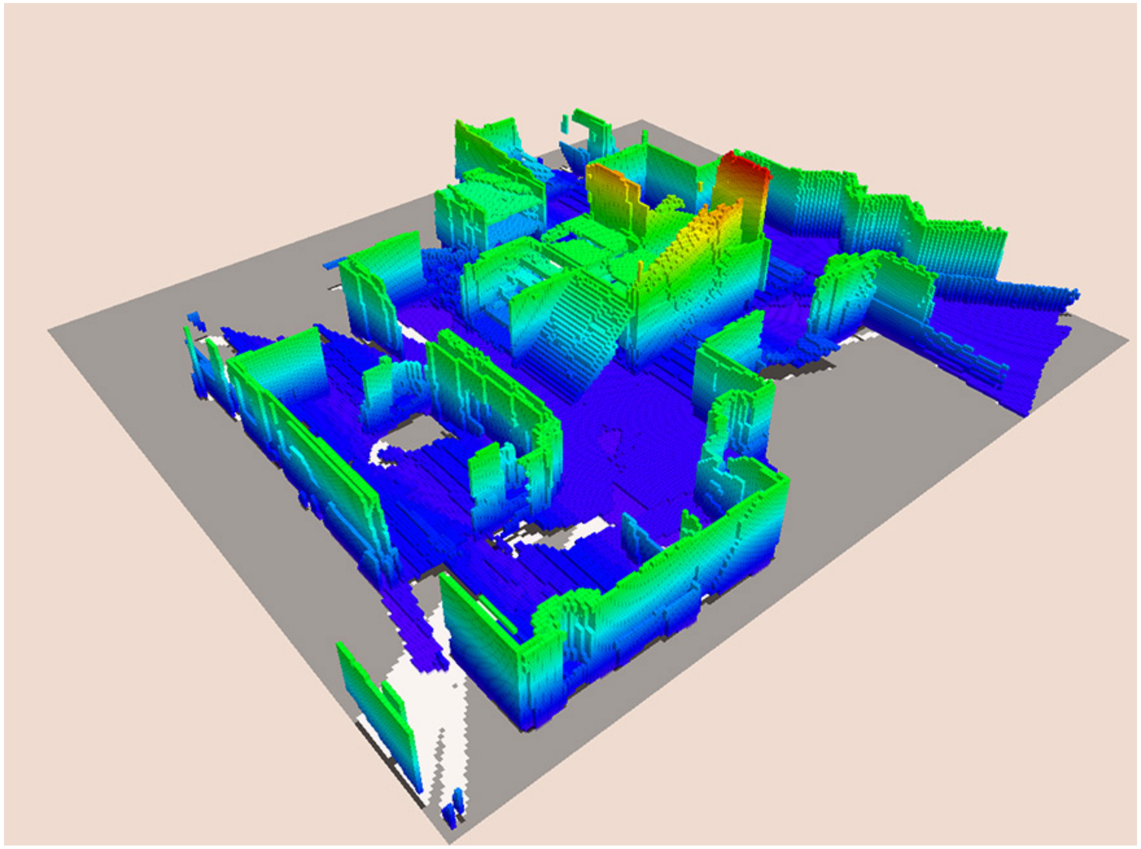
[http://www.openrobots.org/morse/doc/1.2/user/advanced_tutorials/ros_nav_tutorial.html]

Robot Maps



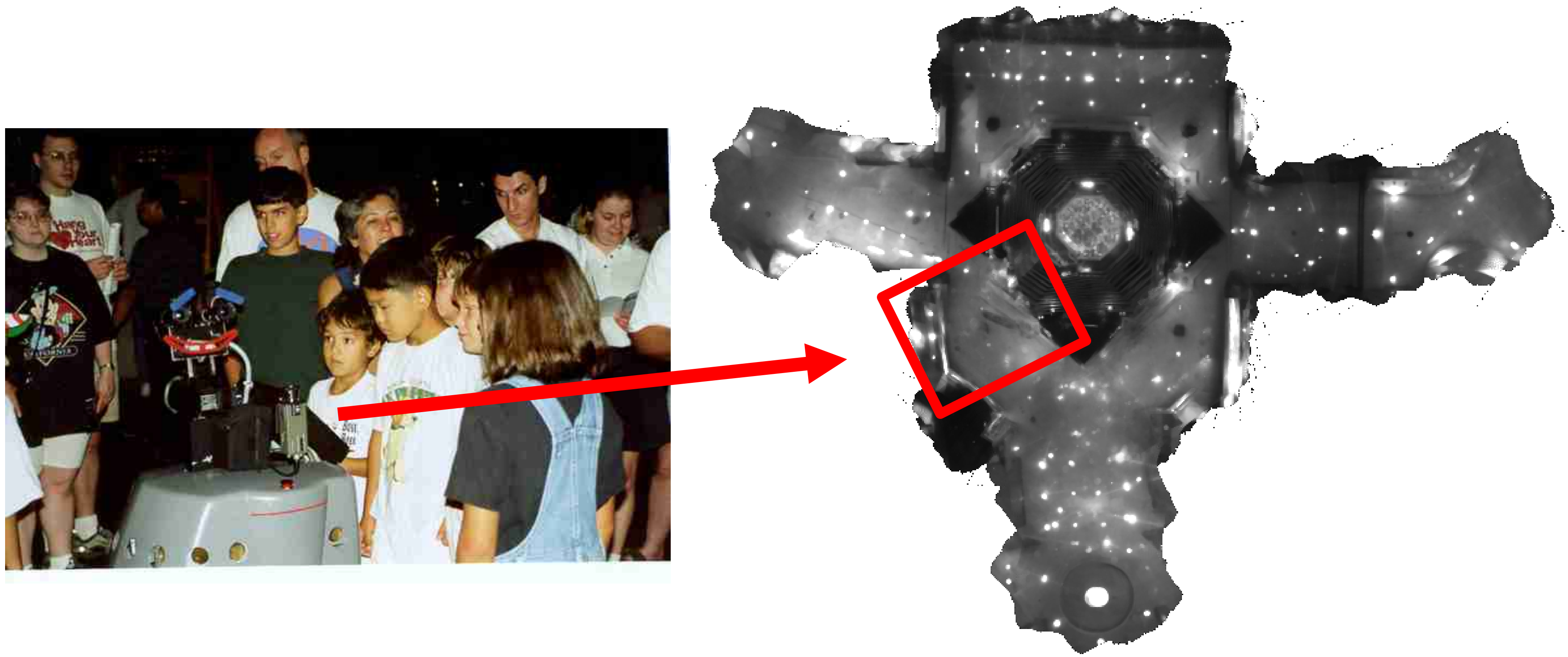
[https://raw.githubusercontent.com/wiki/introlab/rtabmap/doc/IROS-Kinect-Challenge/combined.png]

Robot Maps

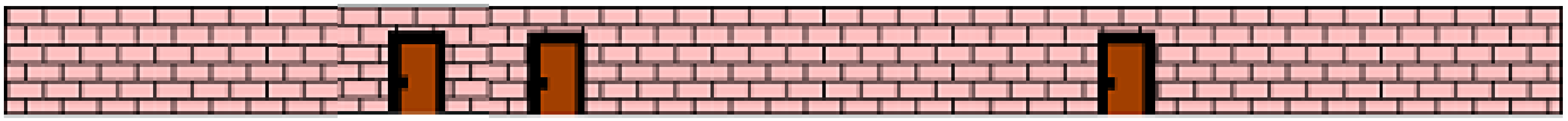


[http://rrt.fh-wels.at/images/sites/fancybox/3d_mapping.jpg]

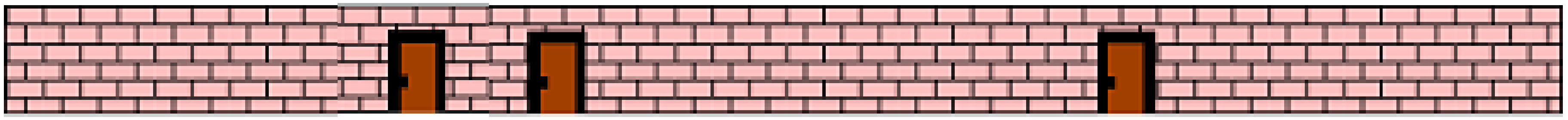
Robot Maps



A Simple 1-D Map

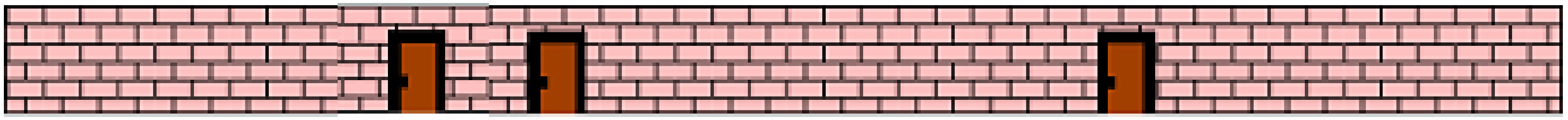


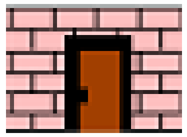
A Simple 1-D Map



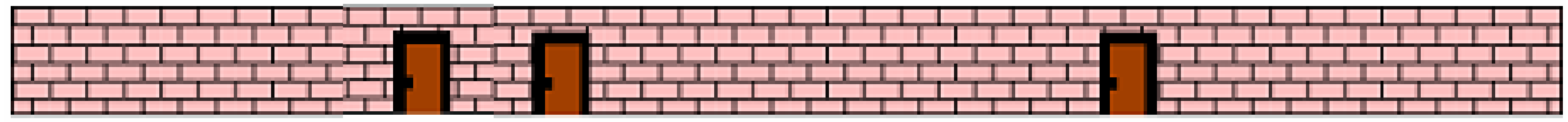
How would we represent this map using math?

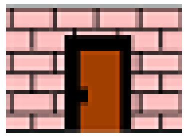
A Simple 1-D Map



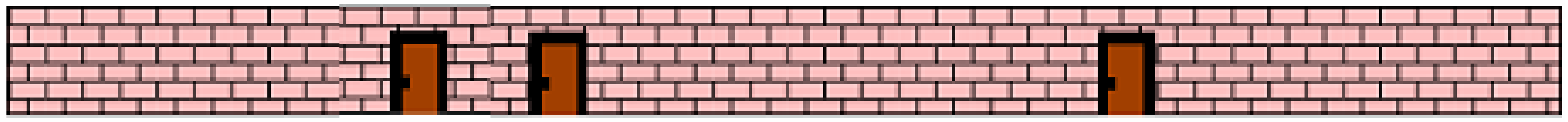
At t = 1, our robot receives an observation: 

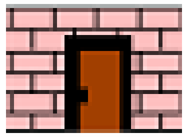
A Simple 1-D Map

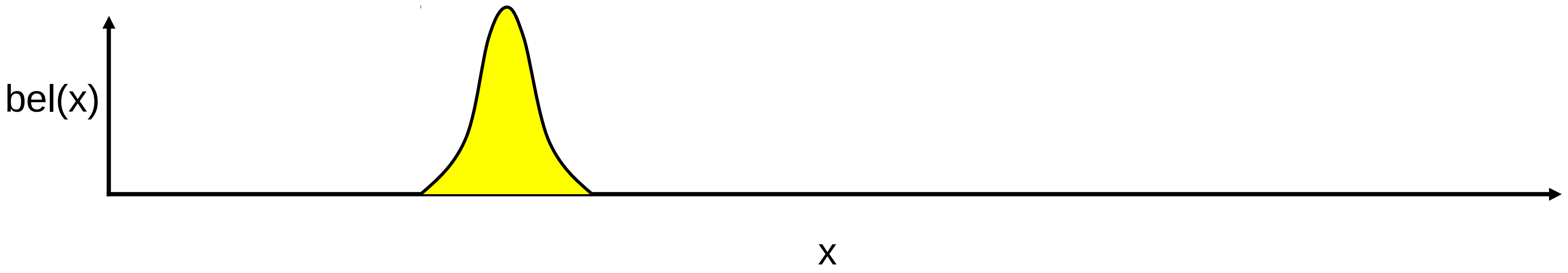


At t = 1, our robot receives an observation: 

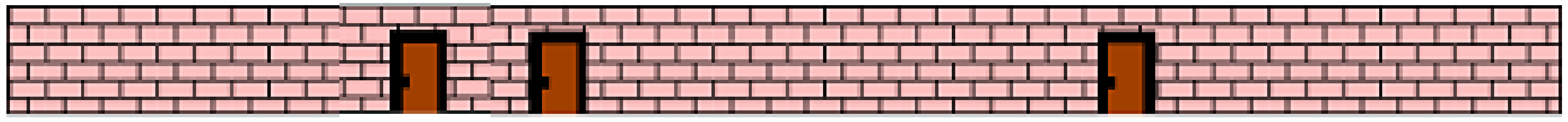
A Simple 1-D Map

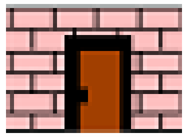


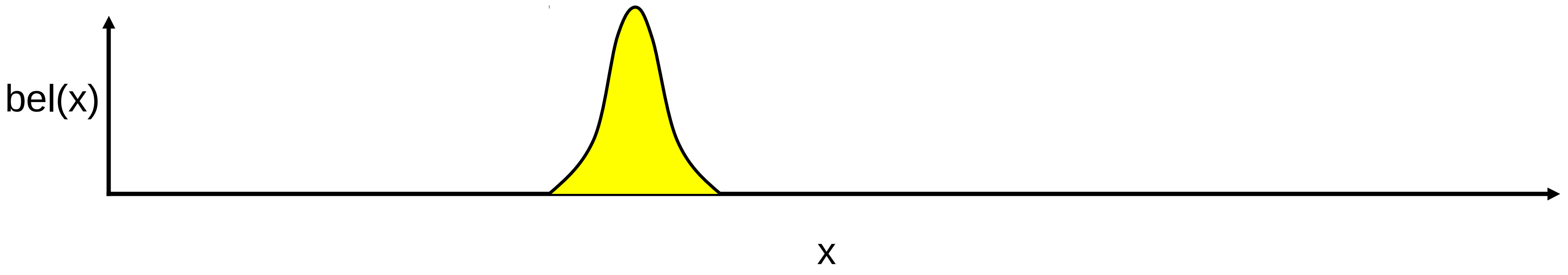
At t = 1, our robot receives an observation: 



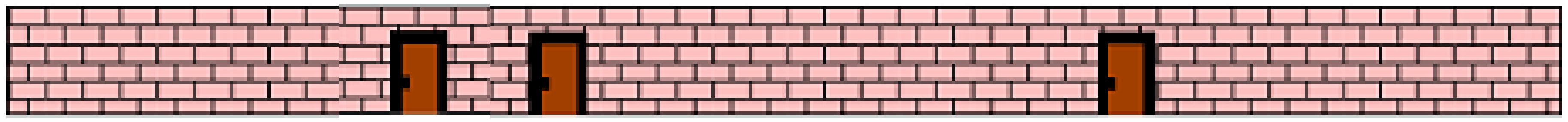
A Simple 1-D Map

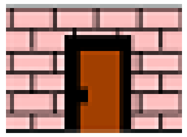


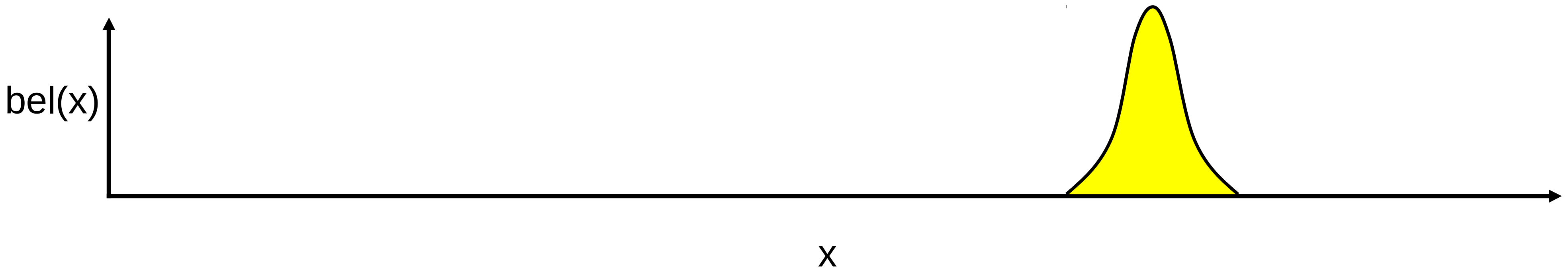
At t = 1, our robot receives an observation: 



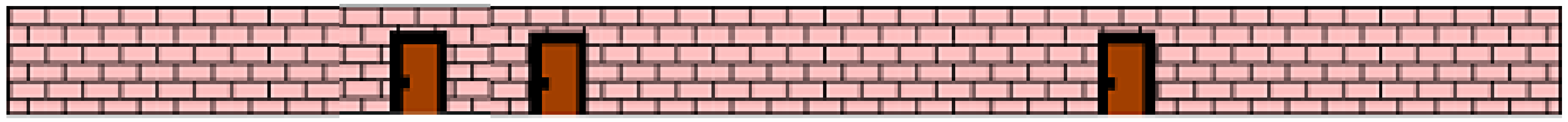
A Simple 1-D Map

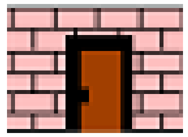


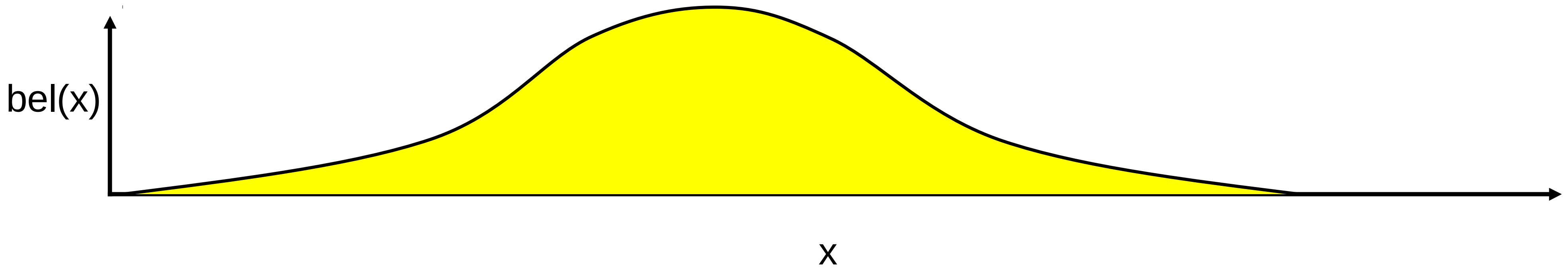
At t = 1, our robot receives an observation: 



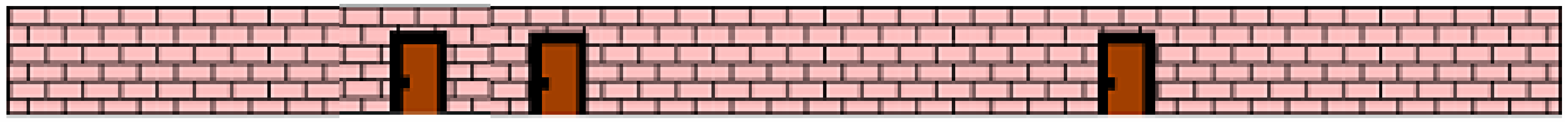
A Simple 1-D Map

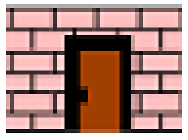


At t = 1, our robot receives an observation: 

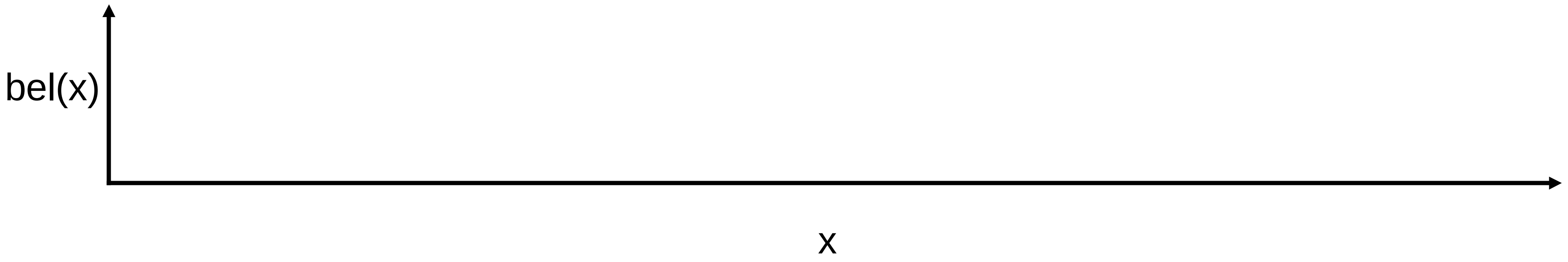


A Simple 1-D Map

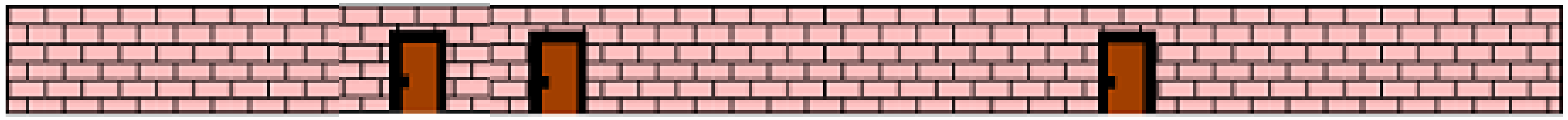


At t = 1, our robot receives an observation: 

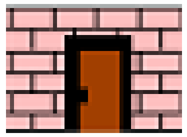
Clearly, a single Gaussian is insufficient to represent our belief
that we may be at either of the 3 doors with equal probability



A Simple 1-D Map

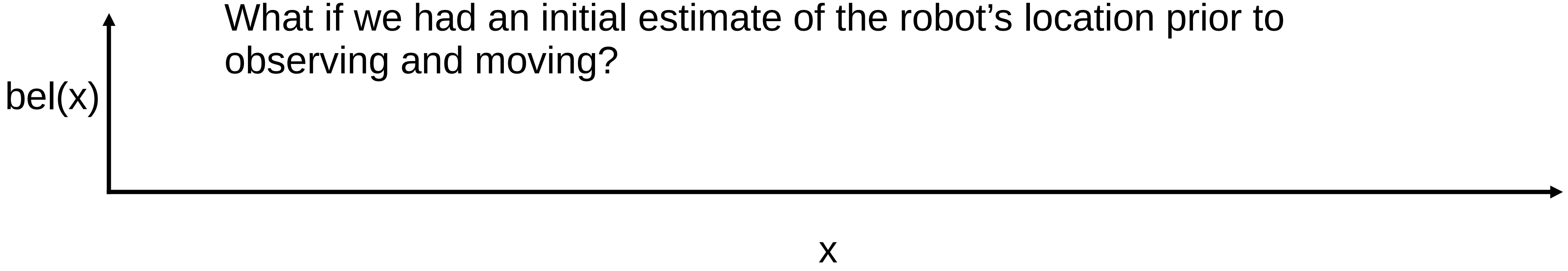


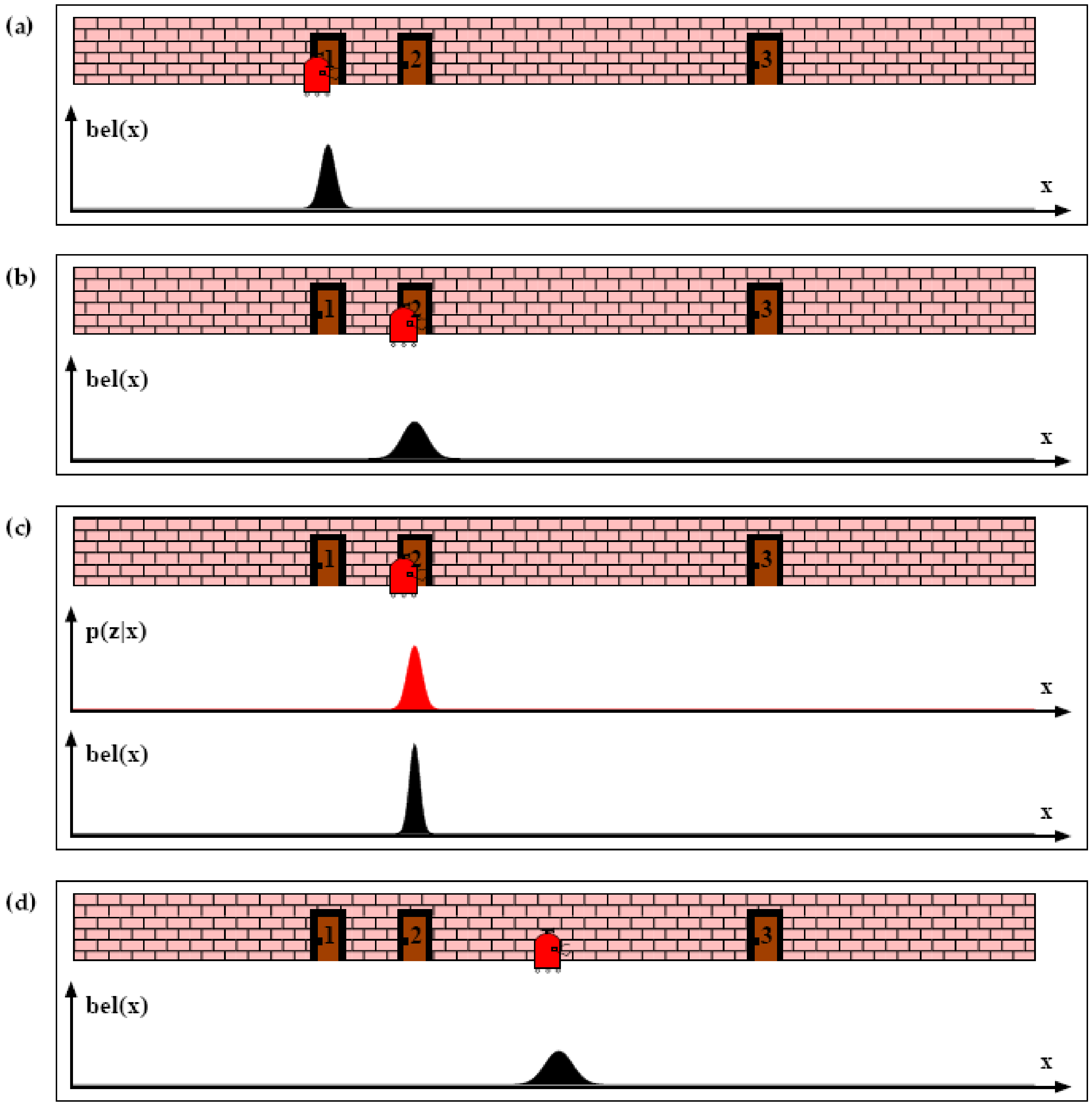
At t = 1, our robot receives an observation:



Clearly, a single Gaussian is insufficient to represent our belief that we may be at either of the 3 doors with equal probability

What if we had an initial estimate of the robot's location prior to observing and moving?




(a)
1
2
3
bel(x)
x
(b)
1
2
3
bel(x)
x
(c)
1
2
3
p(z|x)
x
bel(x)
x
(d)
1
2
3
bel(x)
x

But what if we don’t know where we are at the start?

Or, what if somebody moves the robot manually after it started its operation?

Odometry

“Odometry is the use of data from motion sensors to estimate change in position over time. It is used in robotics by some legged or wheeled robots to estimate their position relative to a starting location. This method is sensitive to errors due to the integration of velocity measurements over time to give position estimates. “

Odometry Errors

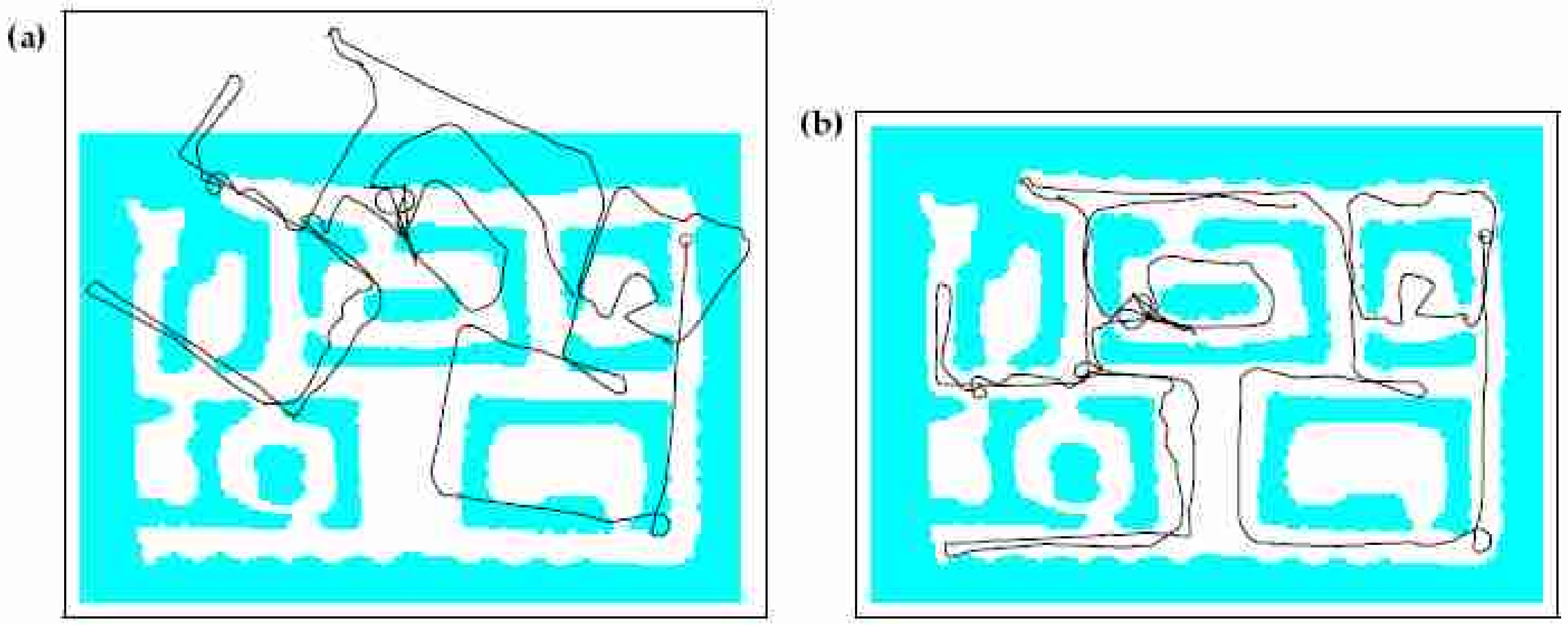
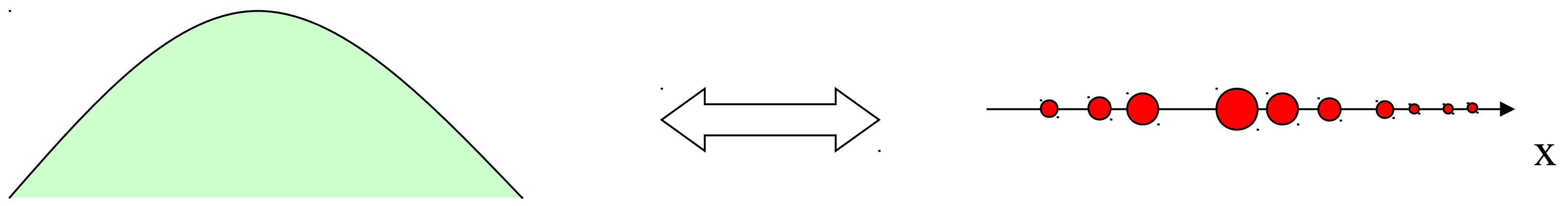
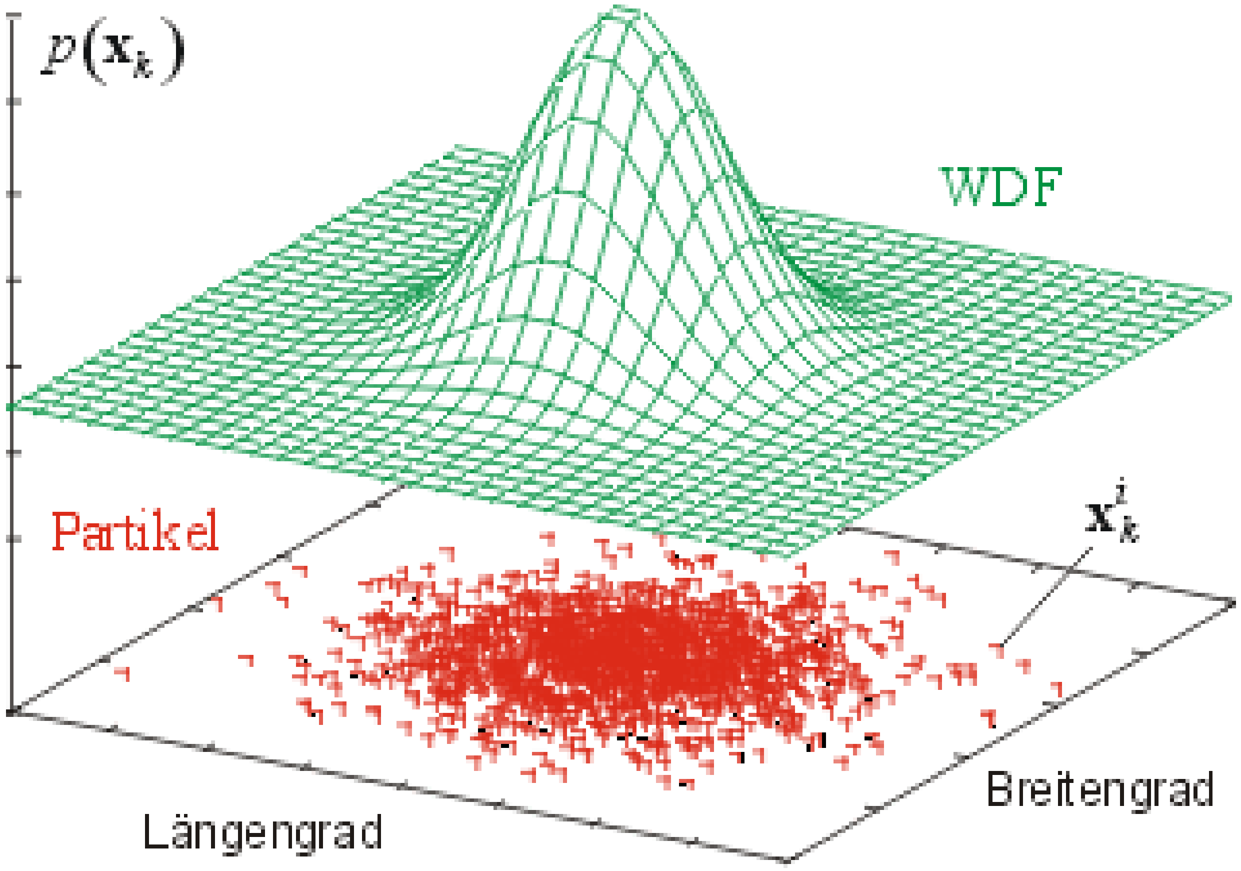


Figure 8.10 (a) Odometry information and (b) corrected path of the robot.

Basic idea behind Particle Filters

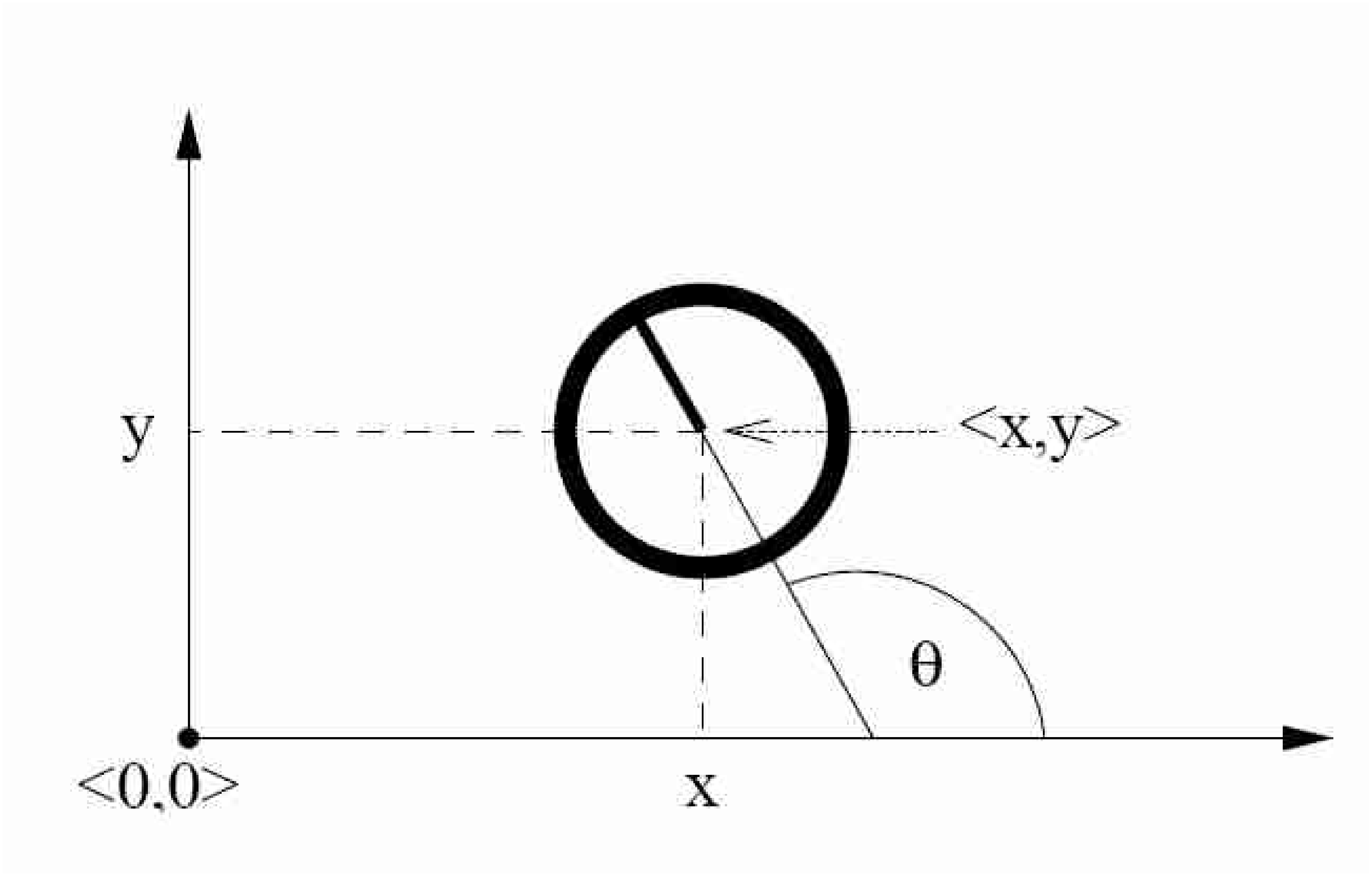


Now, in 2-D

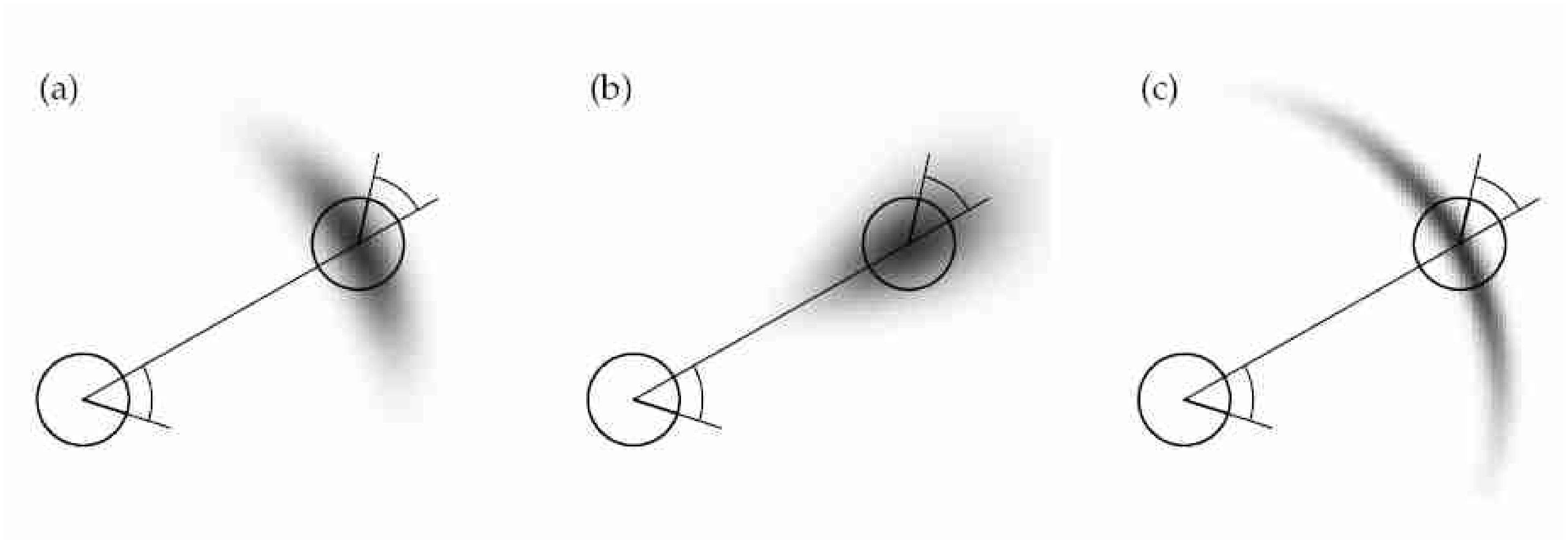


[http://www.ite.uni-karlsruhe.de/METZGER/DIPLOMARBEITEN/dipl2.html]

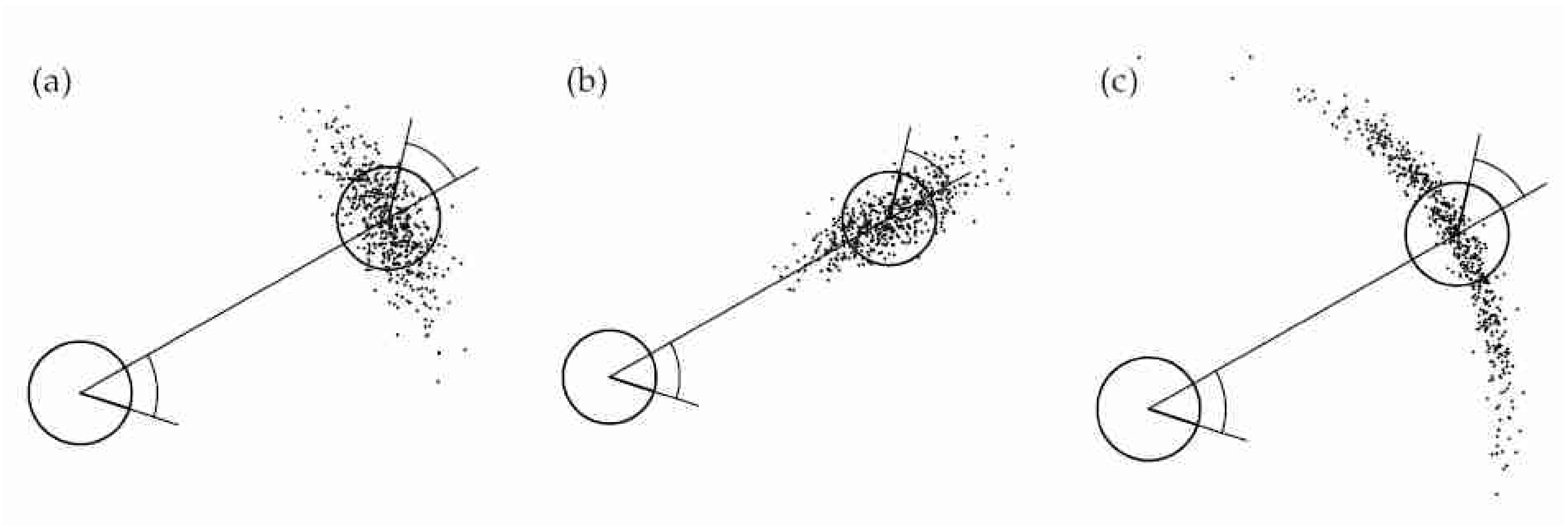
Robot Pose



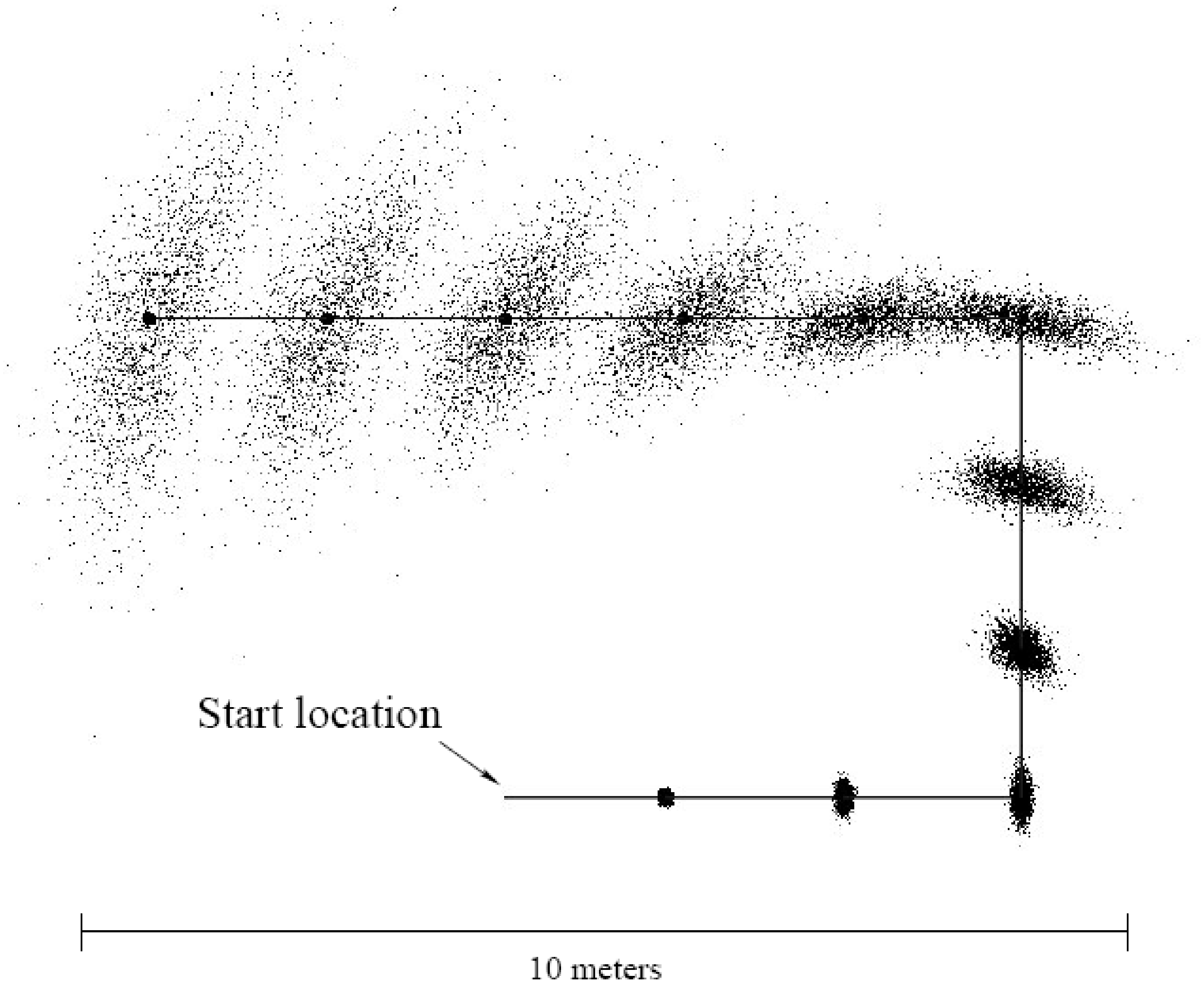
Odometry Motion Model



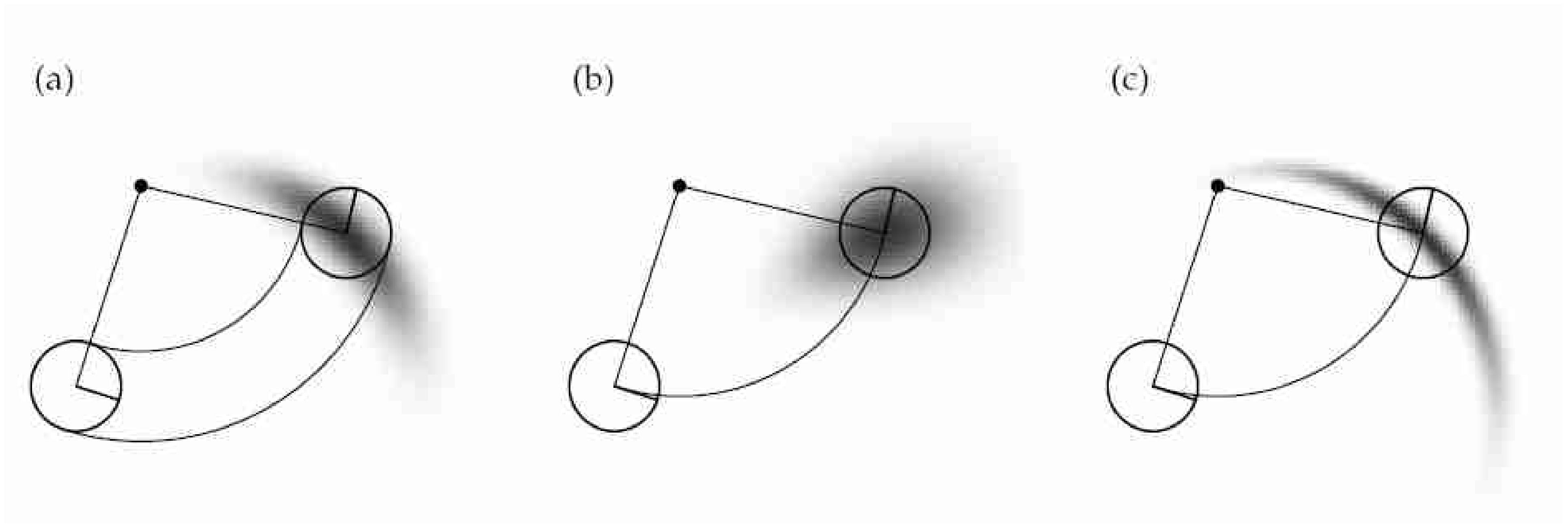
Sampling from the Model



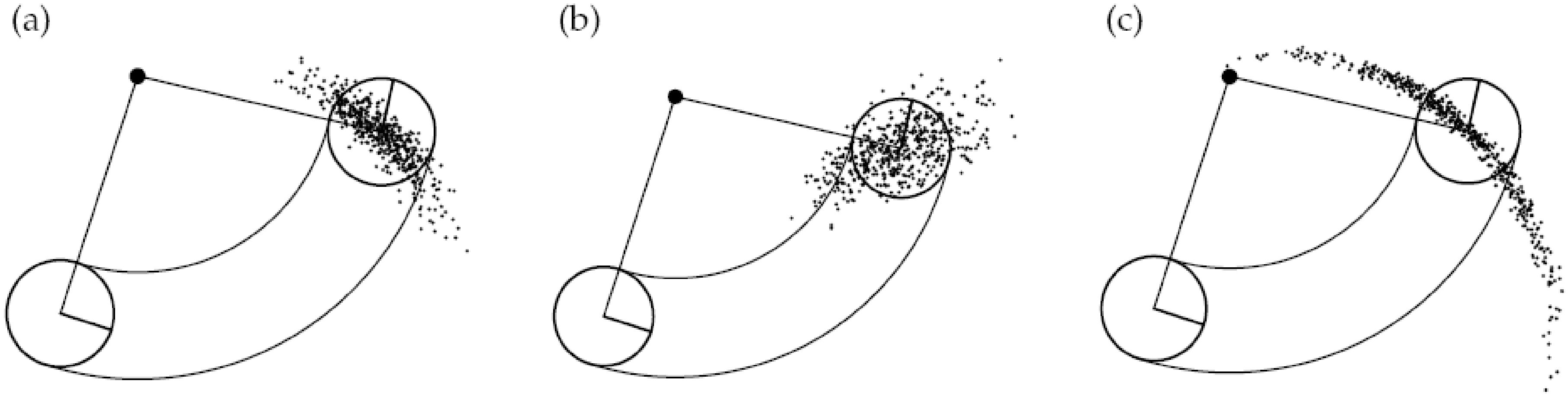
Motion Model



Velocity Models with Different Parameters



Velocity Models with Different Parameters



Example

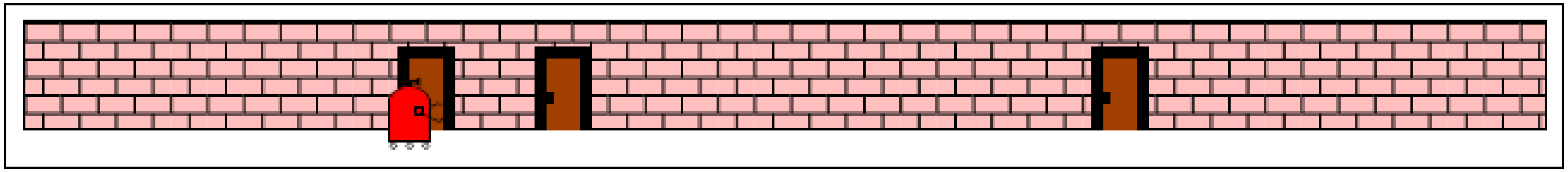
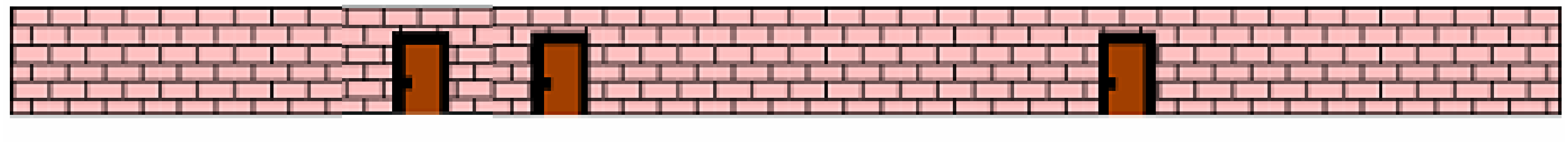
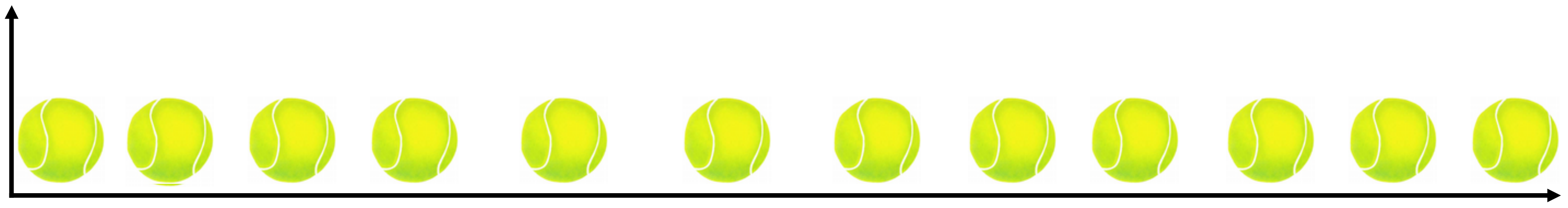


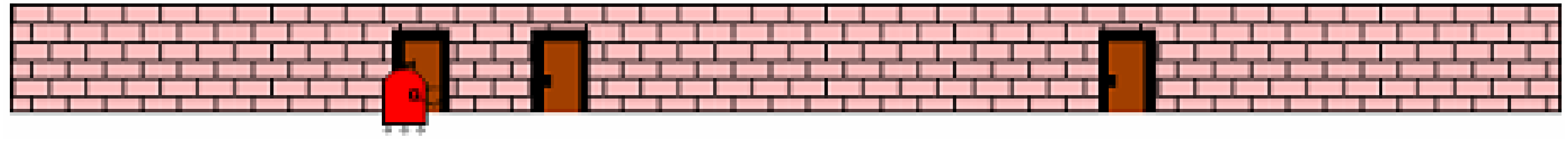
Figure 7.4 Example environment used to illustrate mobile robot localization: One-dimensional hallway environment with three indistinguishable doors. Initially the robot does not know its location except for its heading direction. Its goal is to find out where it is.

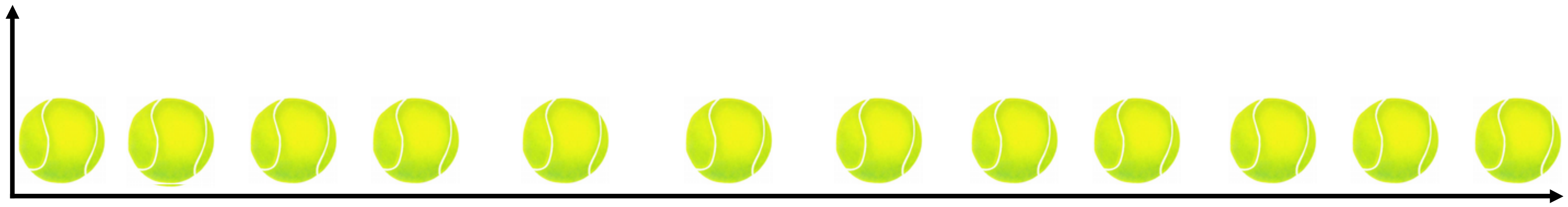
Initially, we do not know the location of the robot, so the particles are everywhere



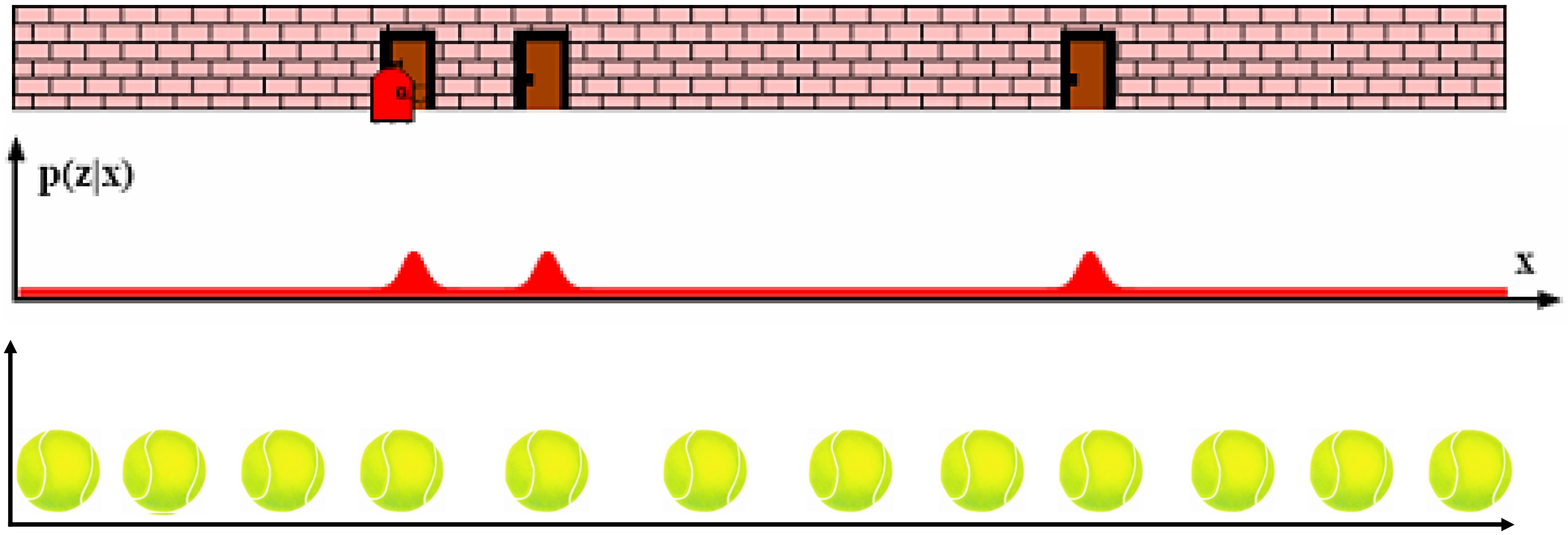


Next, the robot sees a door

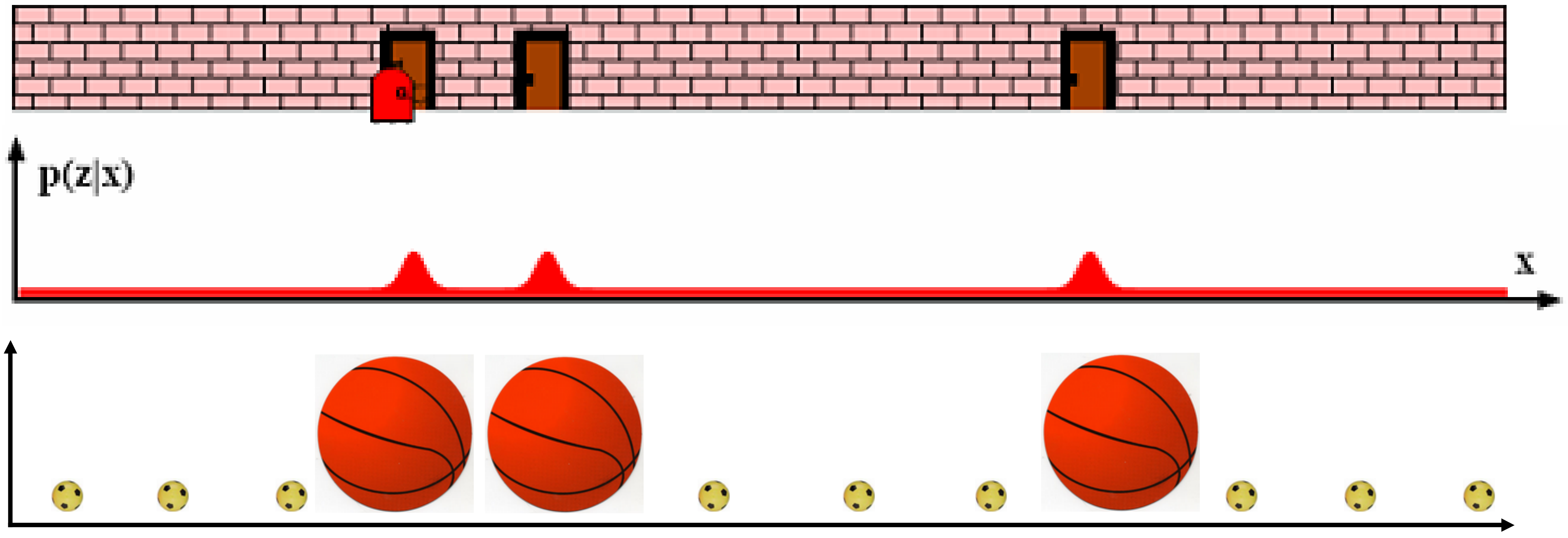




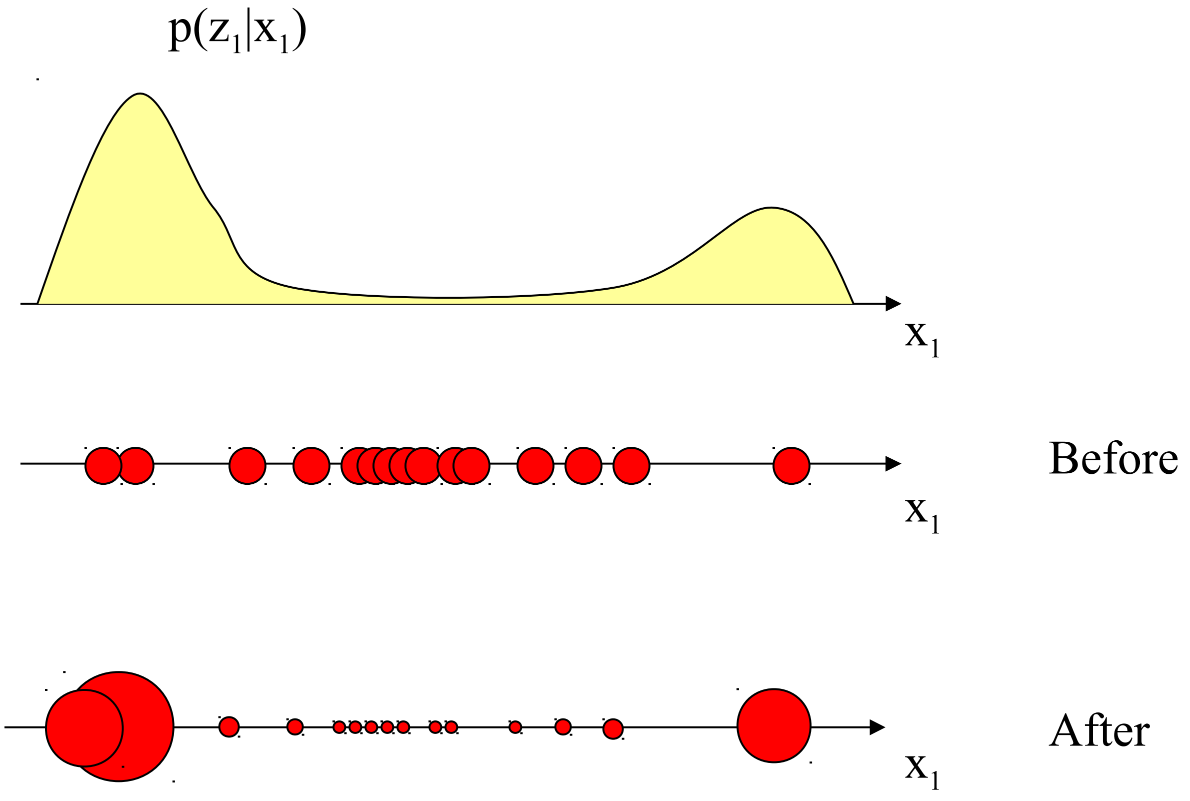
Therefore, we inflate particles next to a door and shrink the rest



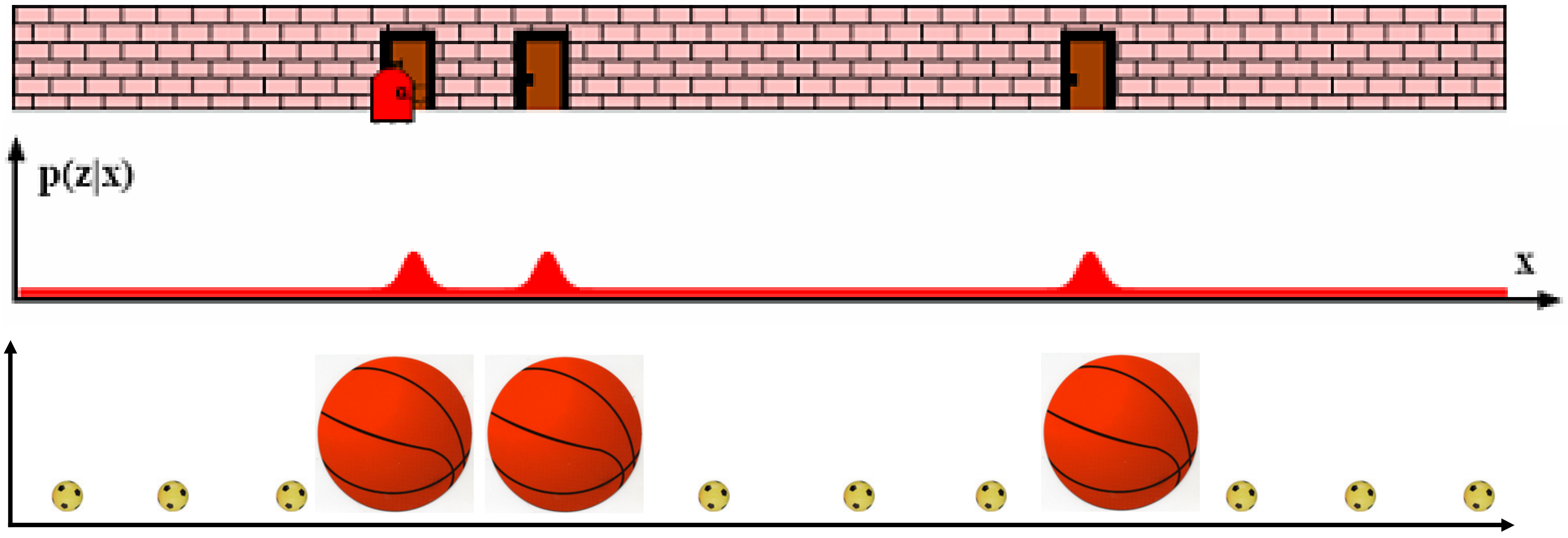
Therefore, we inflate particles next to a door and shrink the rest



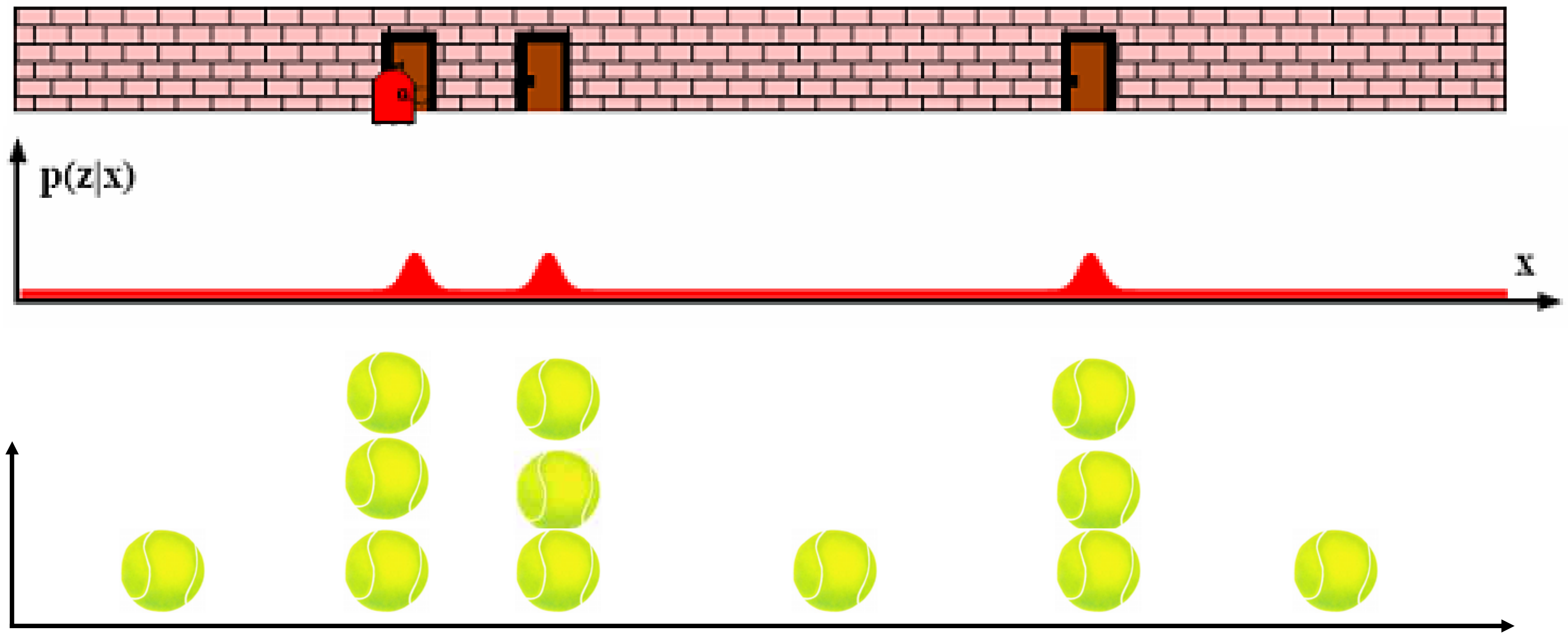
Computing the weights



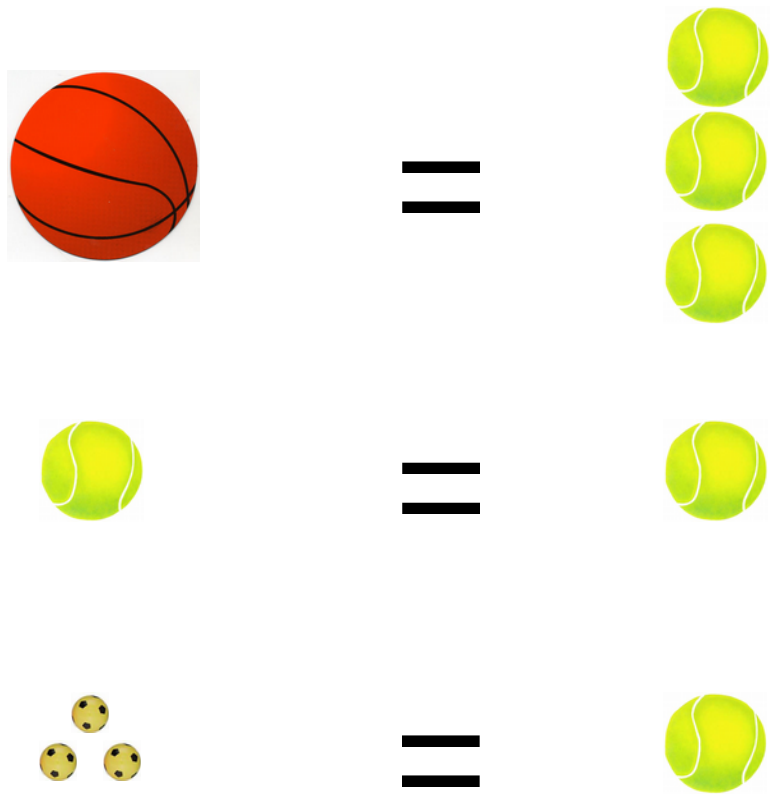
Before, we continue we re-sample our particles to make them all the same size



Before, we continue we re-sample our particles to make them all the same size



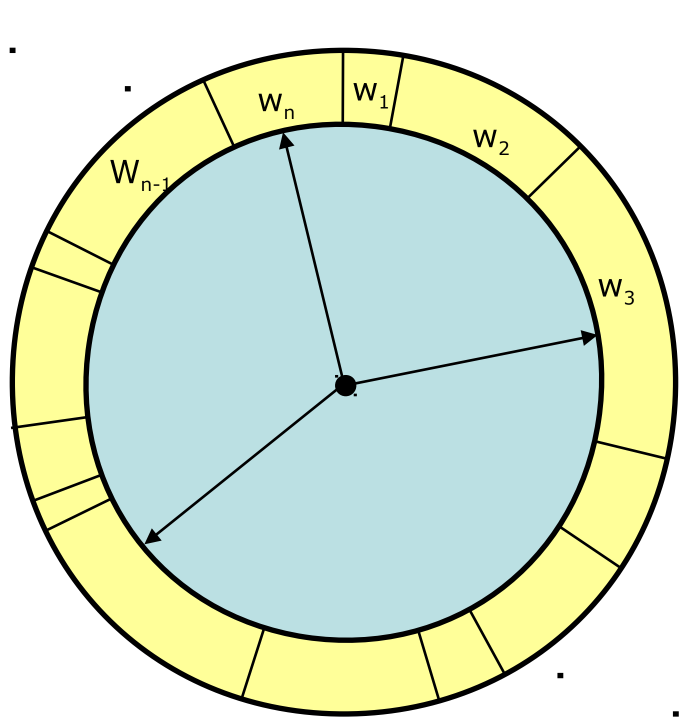
Resampling Rules

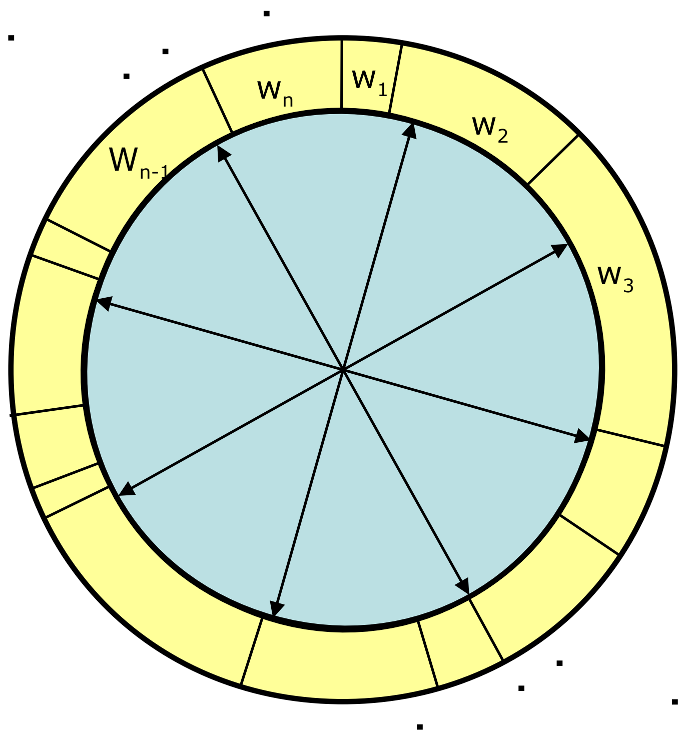


Resampling

- **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 Resampling



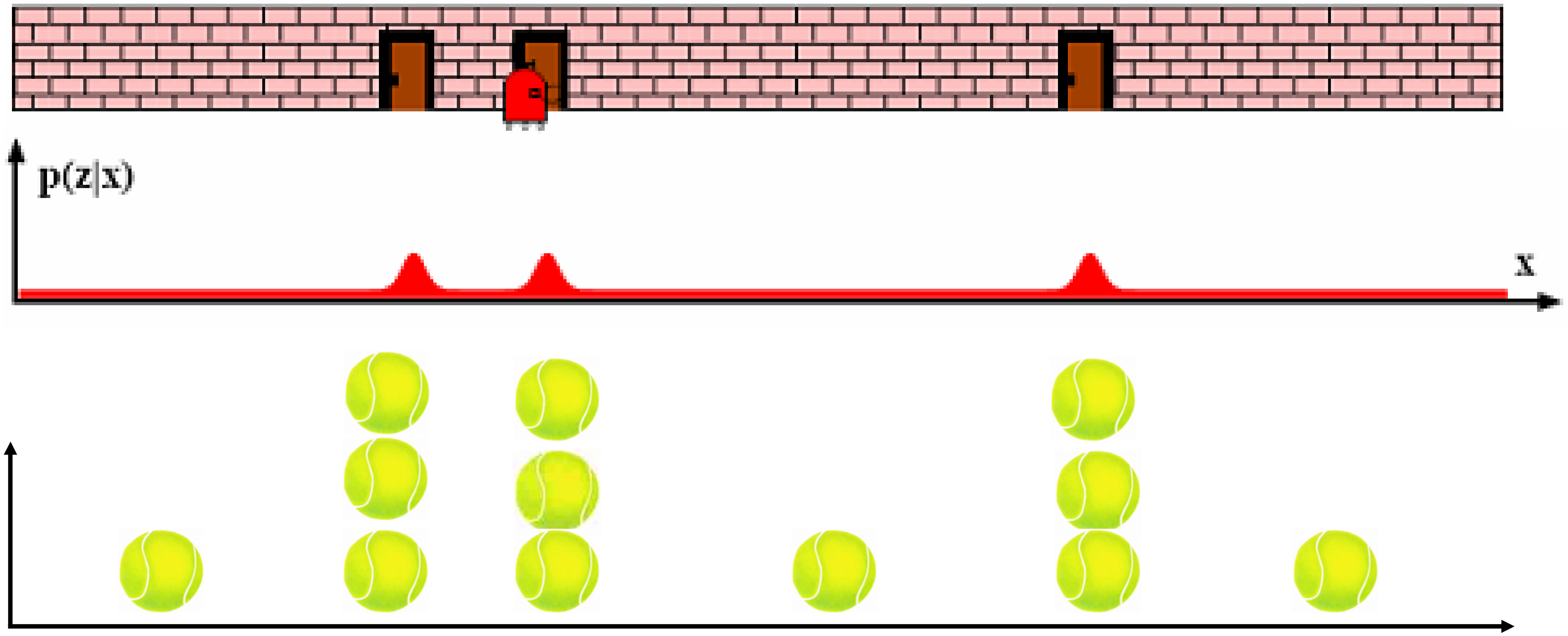


- Roulette wheel
- Binary search, n log n

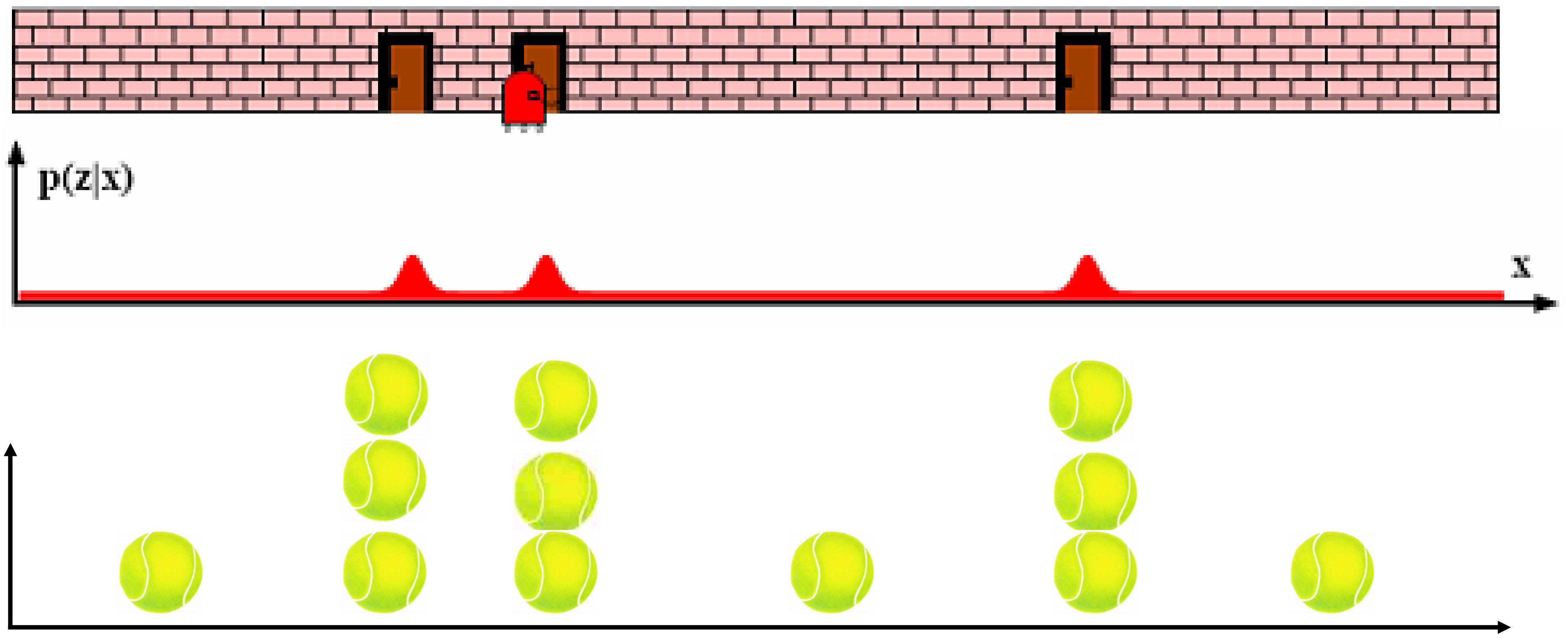
- Stochastic universal sampling
- Systematic resampling
- Linear time complexity
- Easy to implement, low variance

[From Thrun's book "Probabilistic Robotics"]

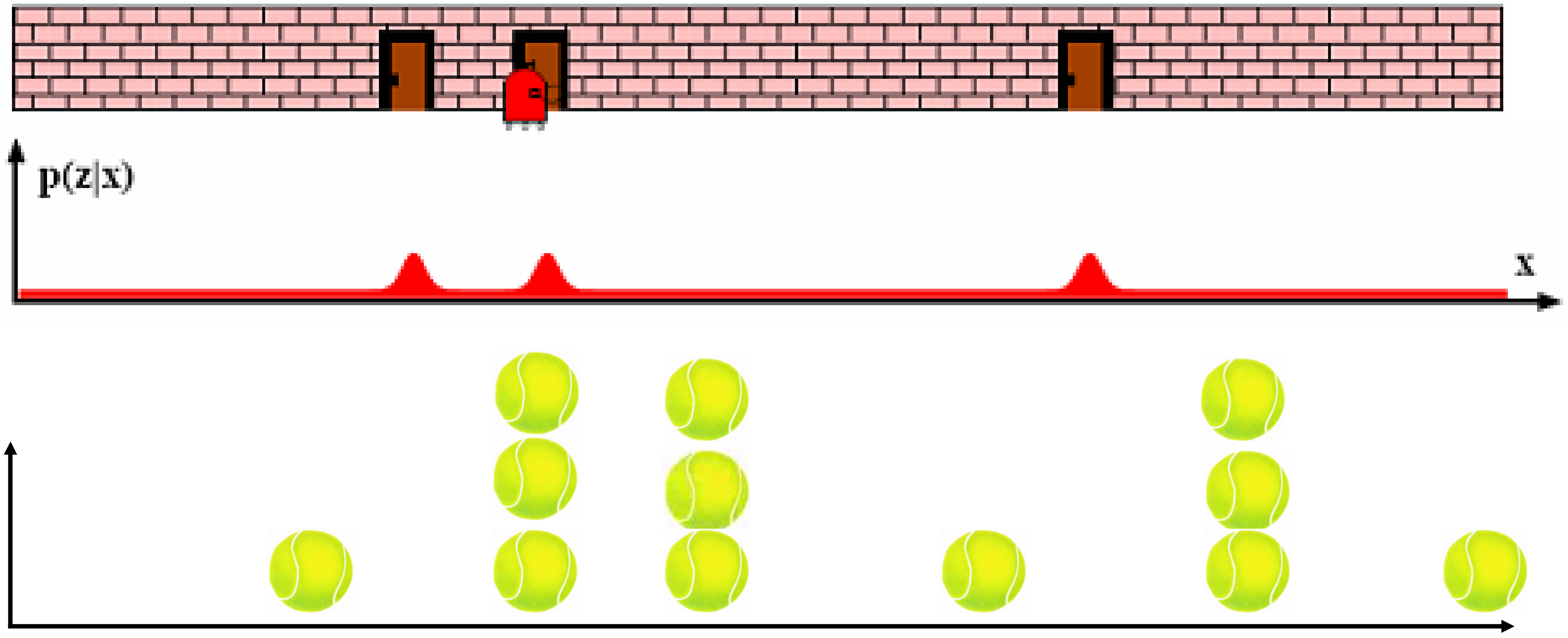
Next, the robot moves to the right...



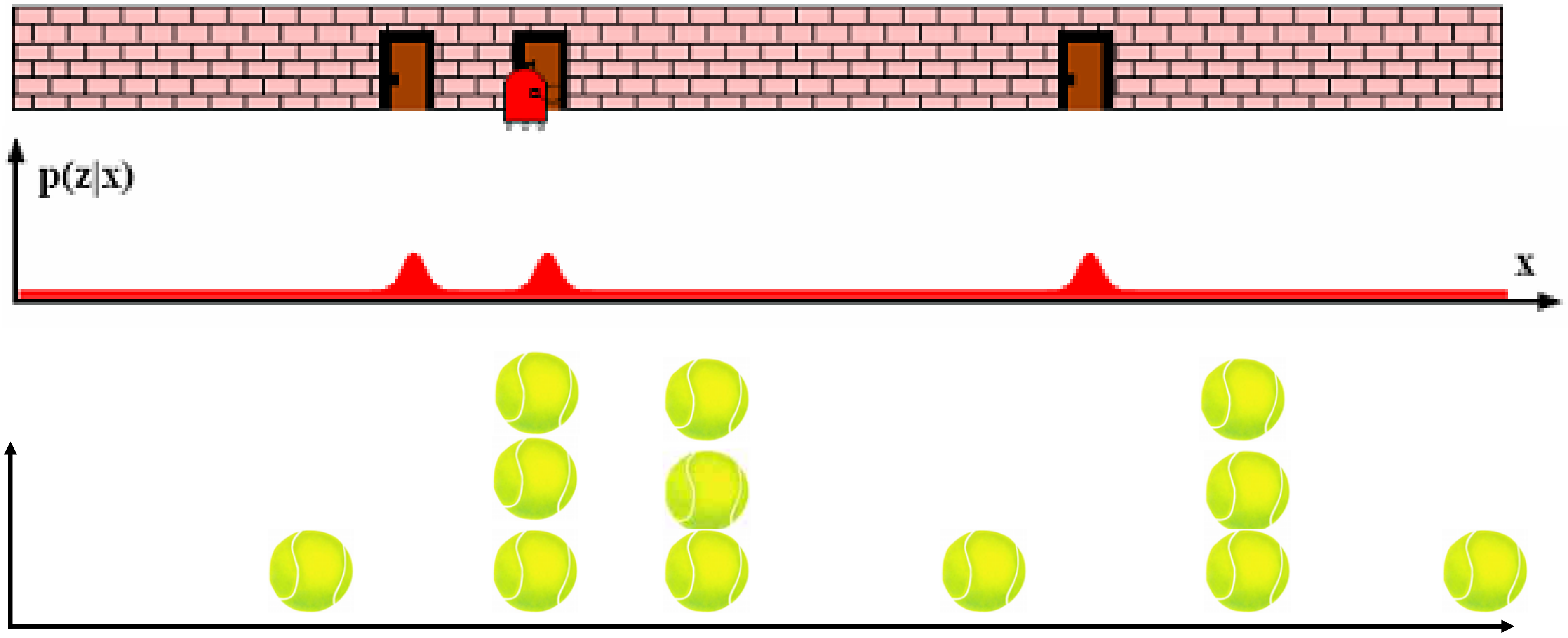
Therefore, we have to shift the particles to the right as well



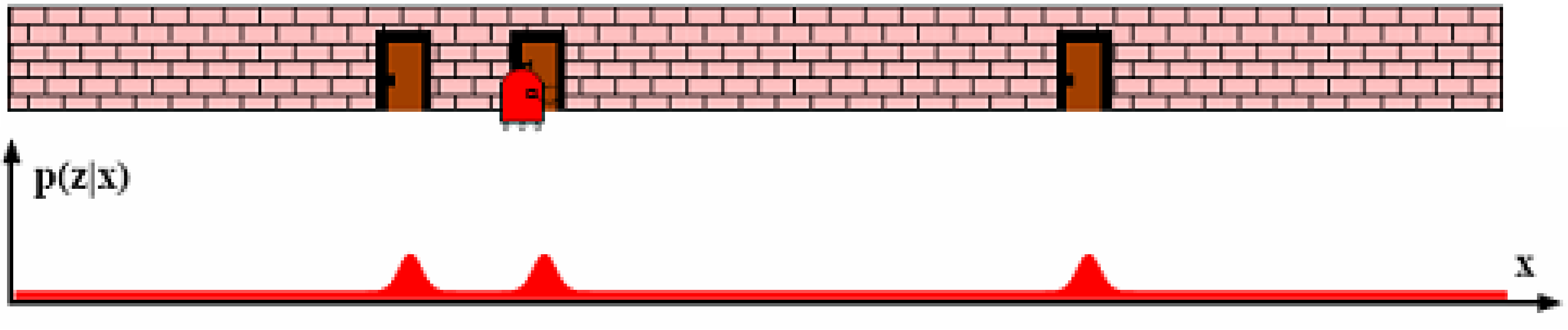
Therefore, we have to shift the particles to the right as well

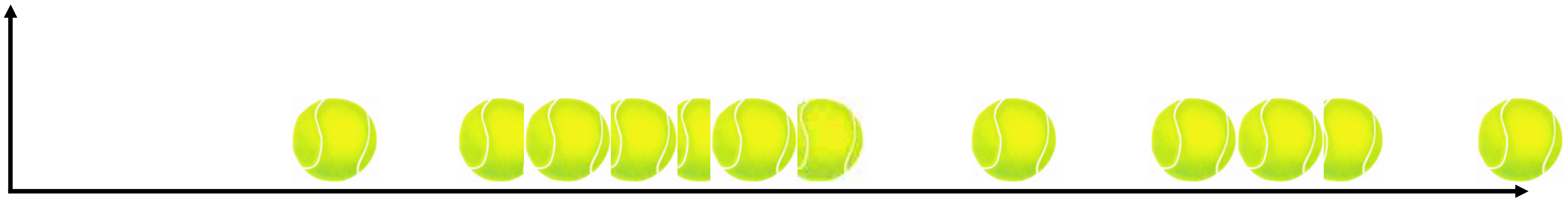


...and add some position noise

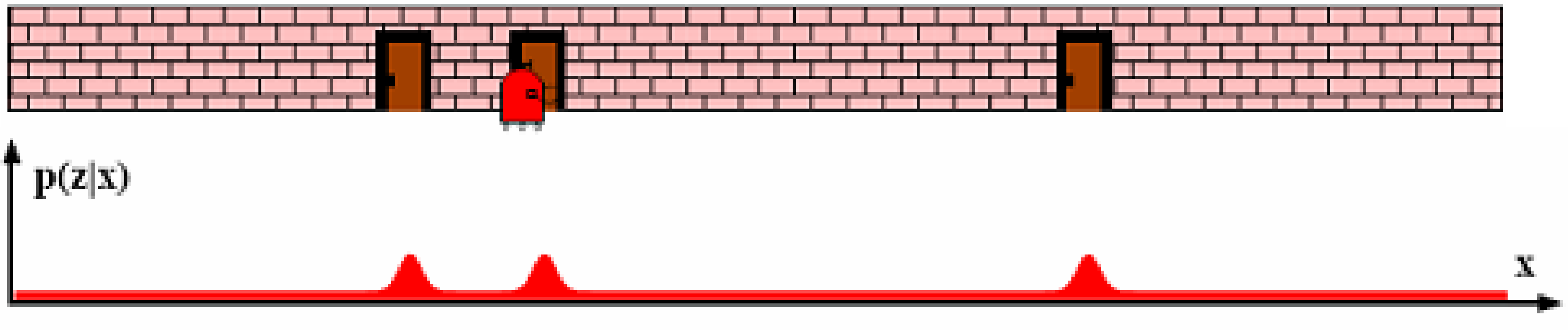


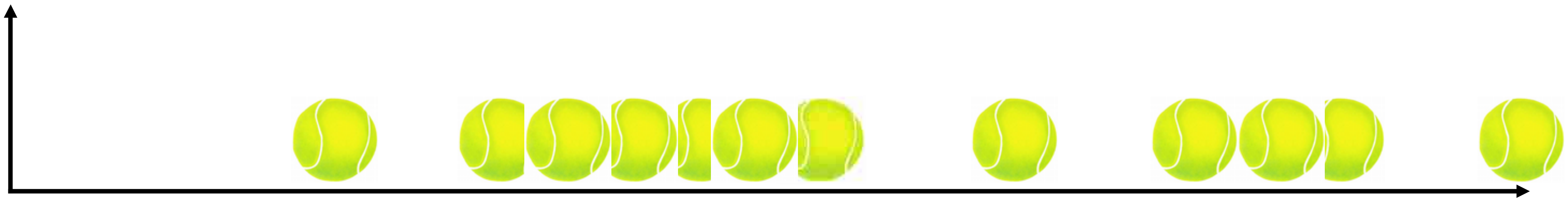
...and add some position noise



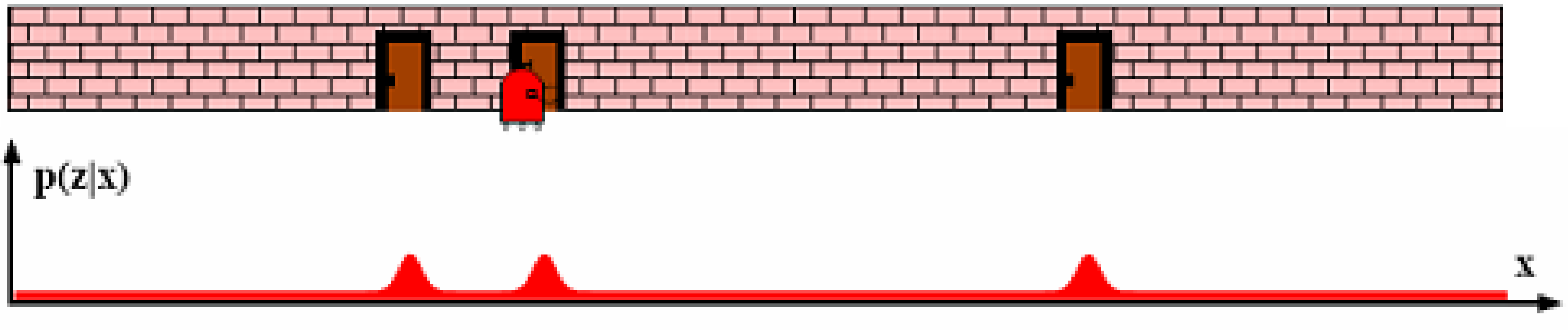


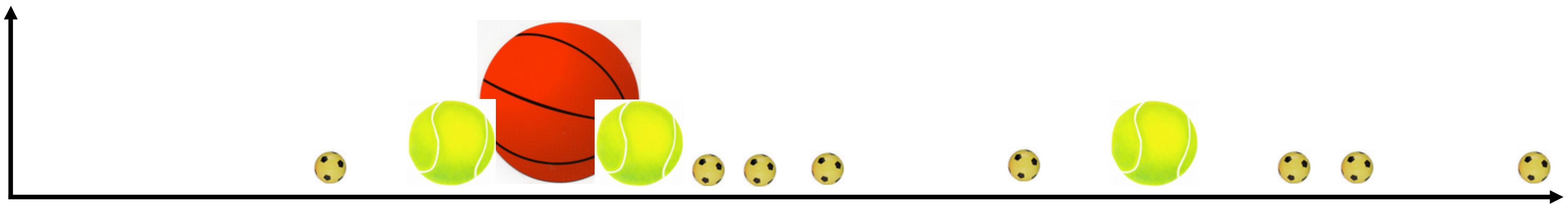
Next, the robot senses that is next to a door



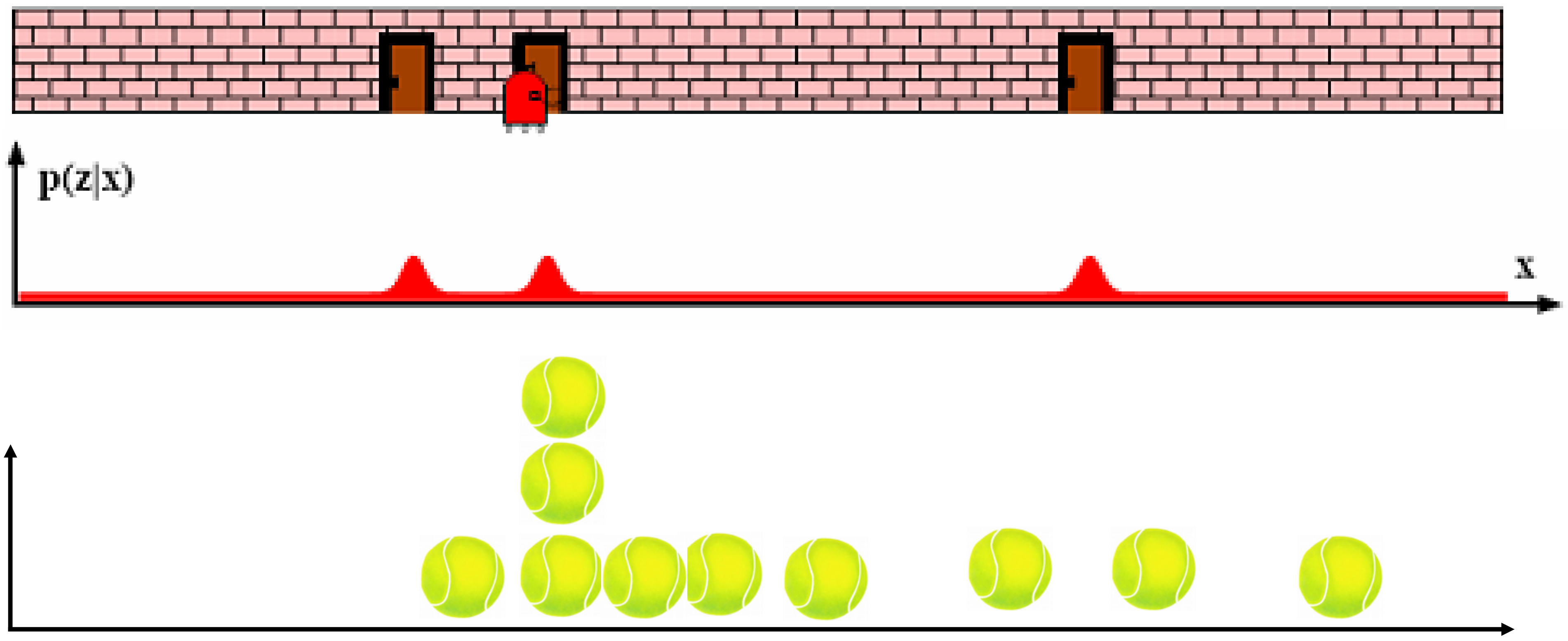


Next, the robot senses that is next to a door

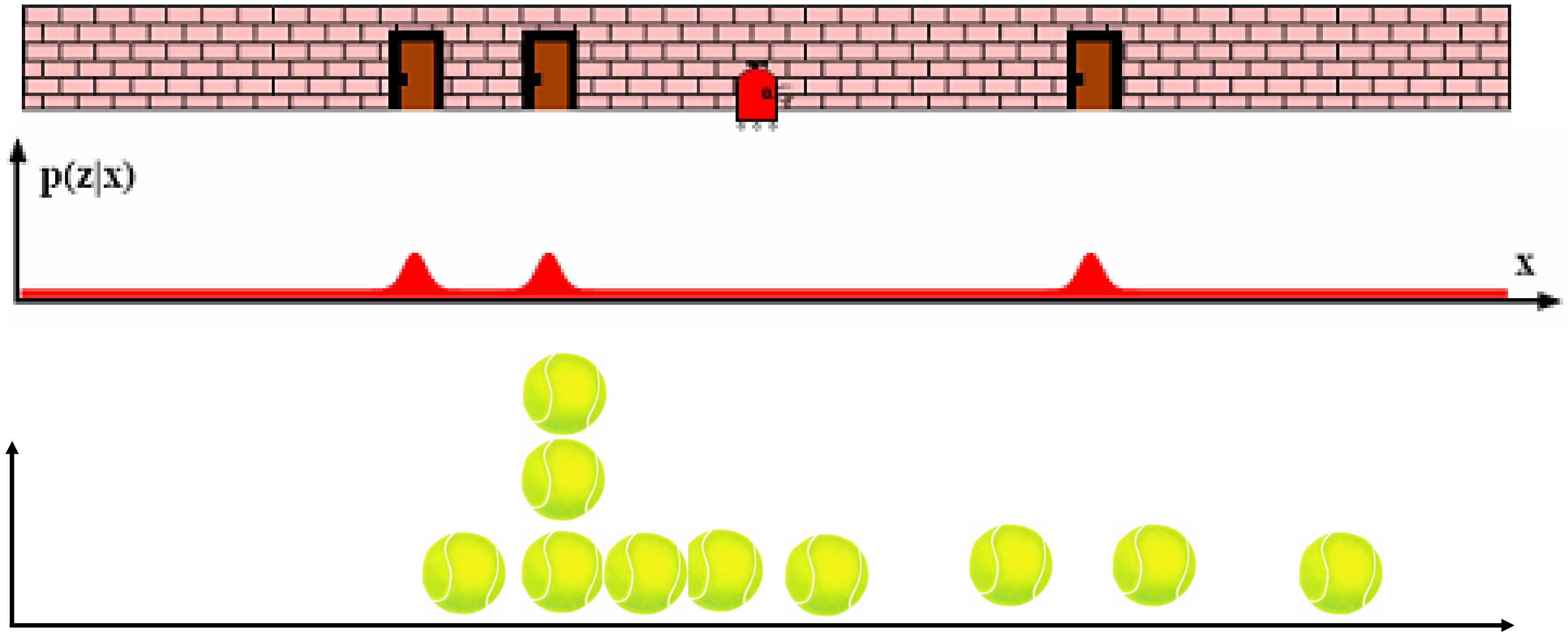




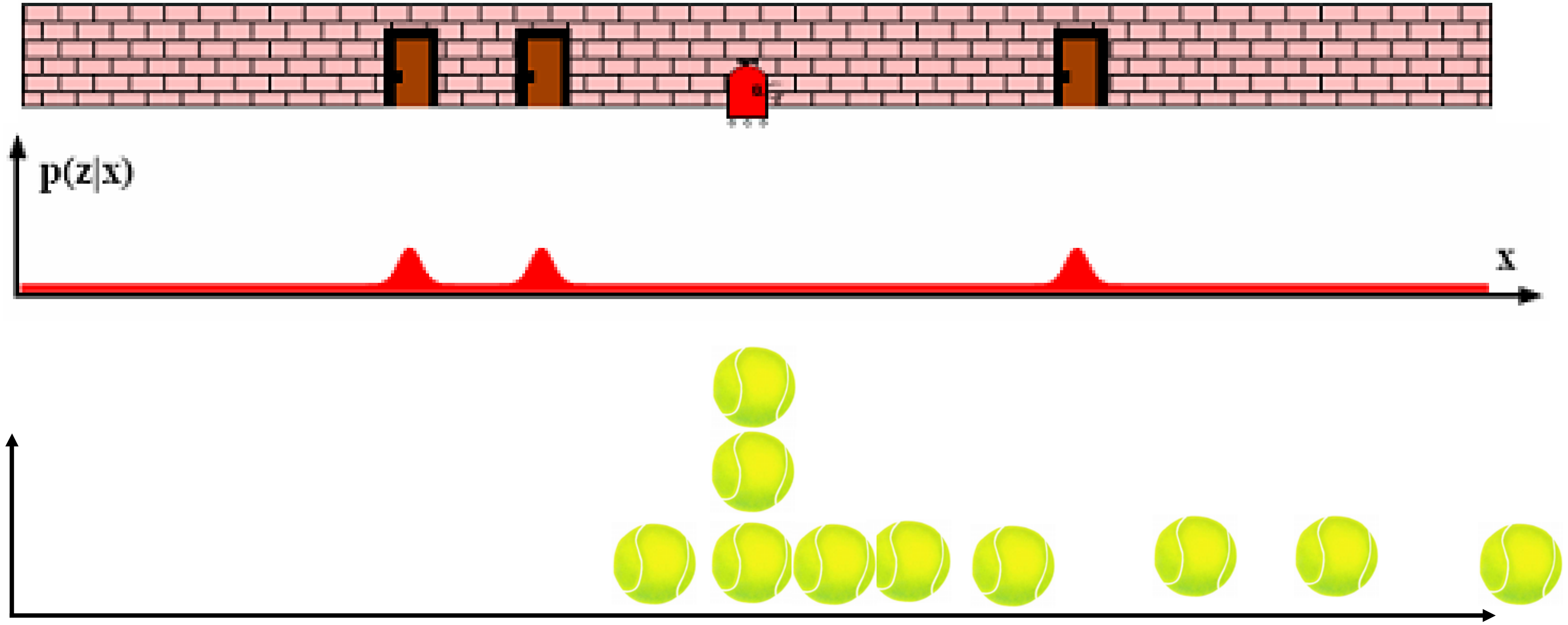
…we resample again



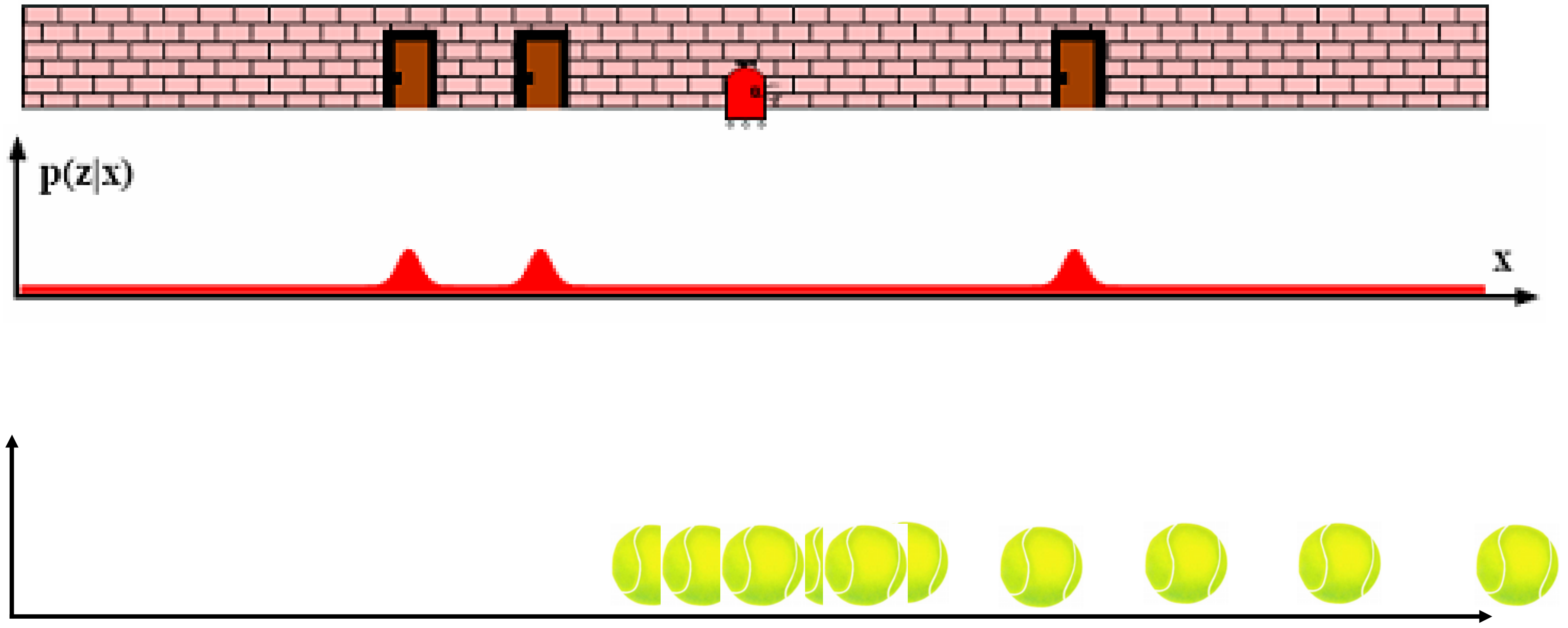
The robot keeps going to the right...



...we shift the particles again



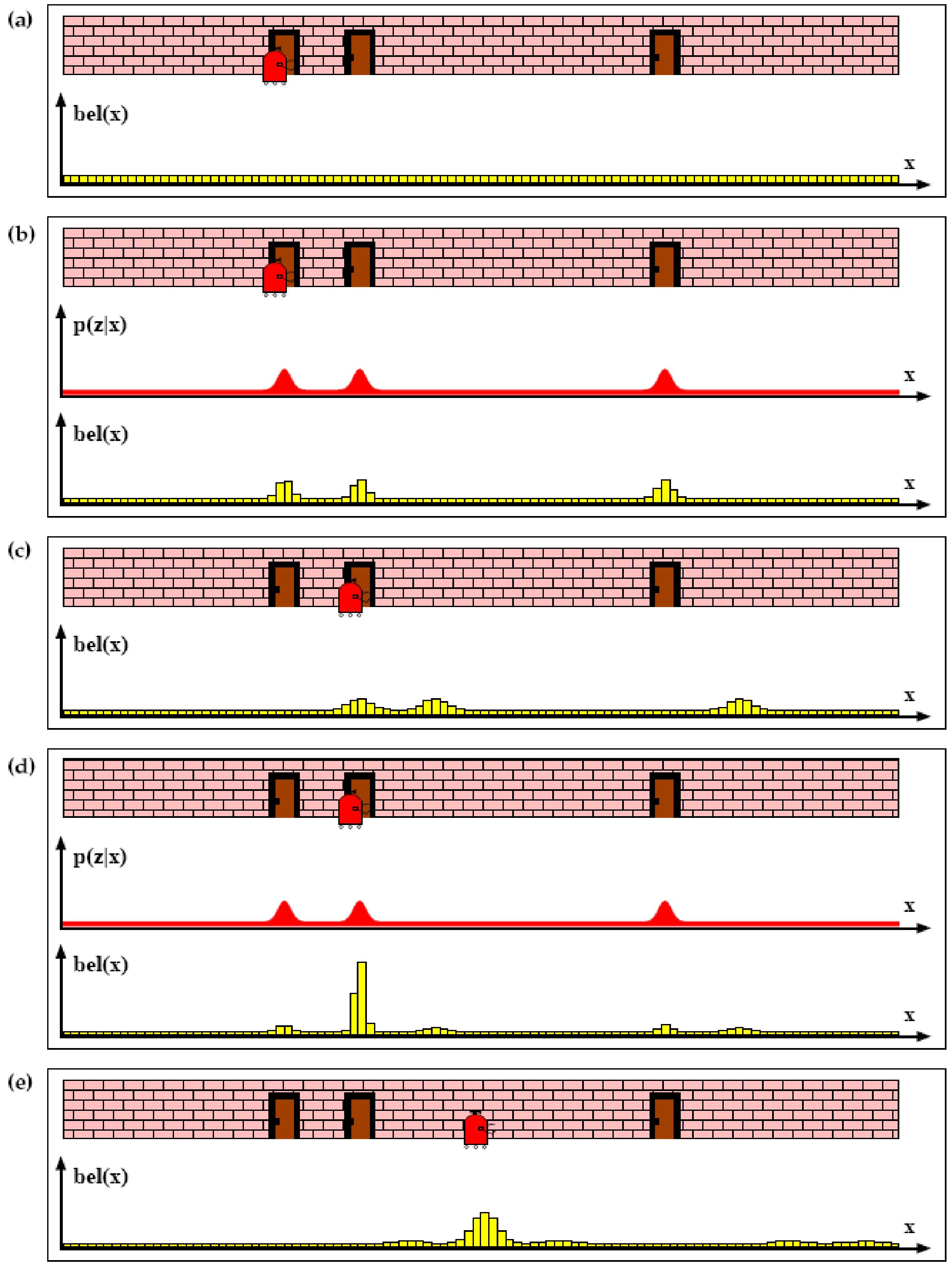
...and we add noise.



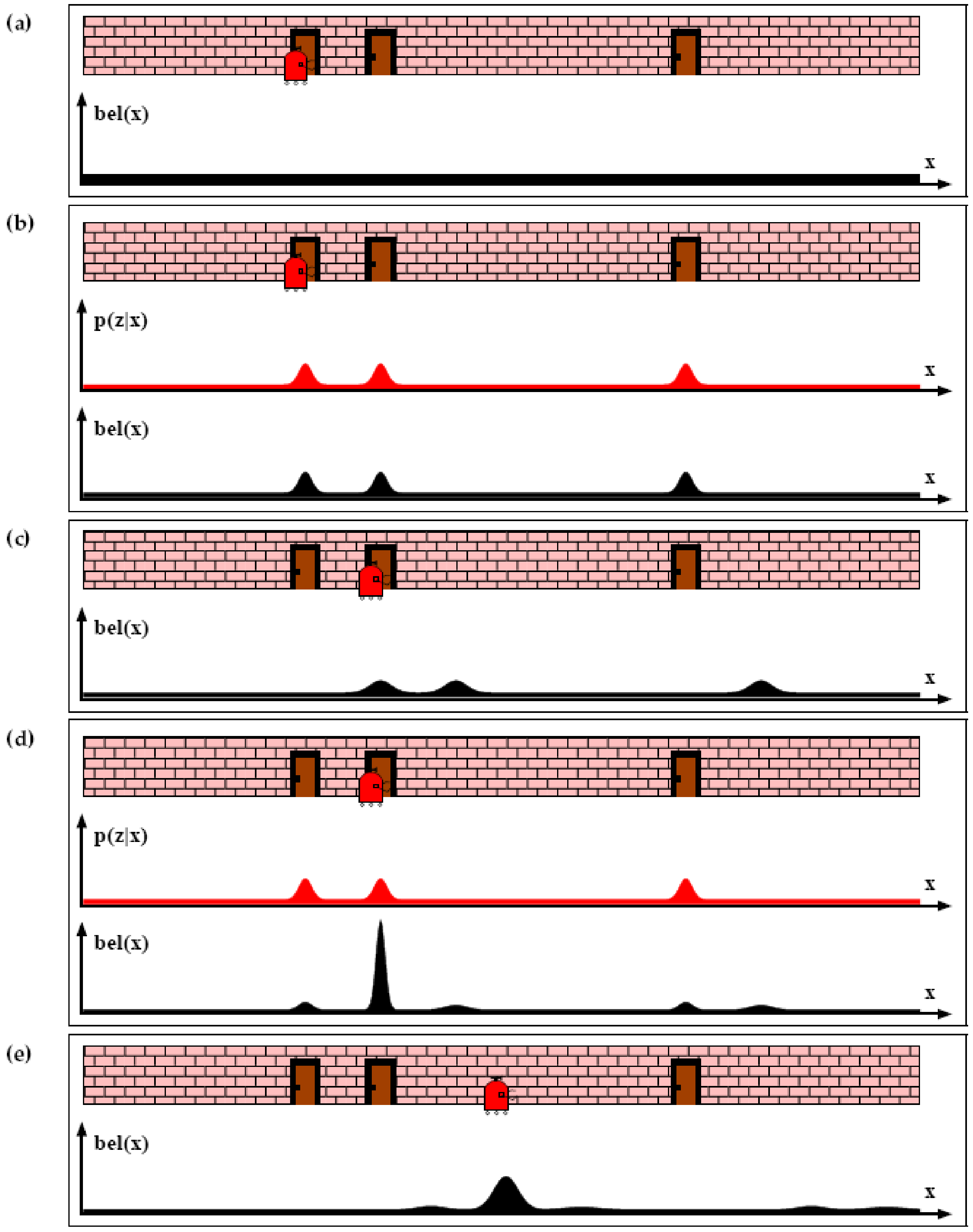
And so on...

Now, let’s compare that to some of the other methods

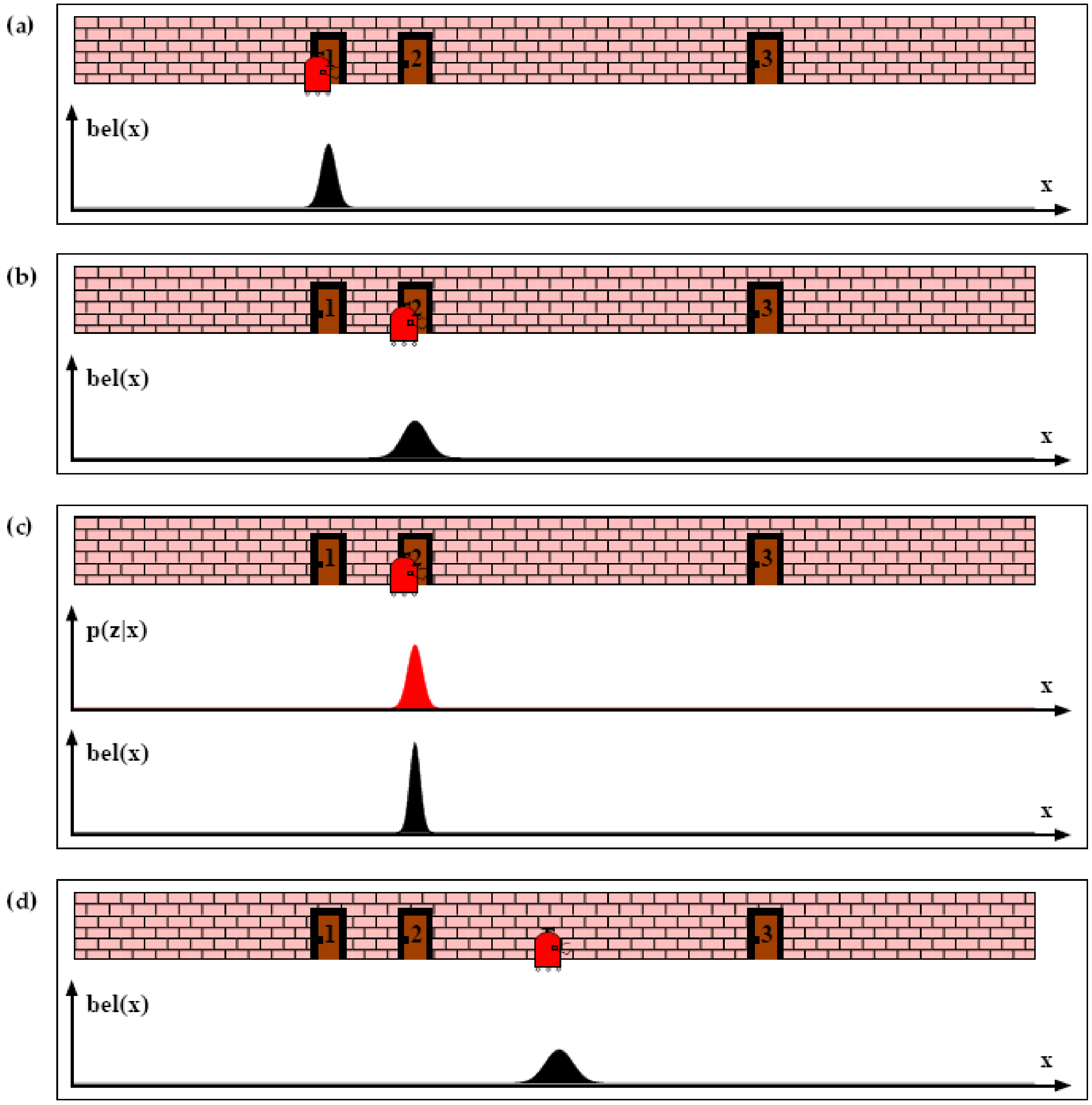
Grid Localization



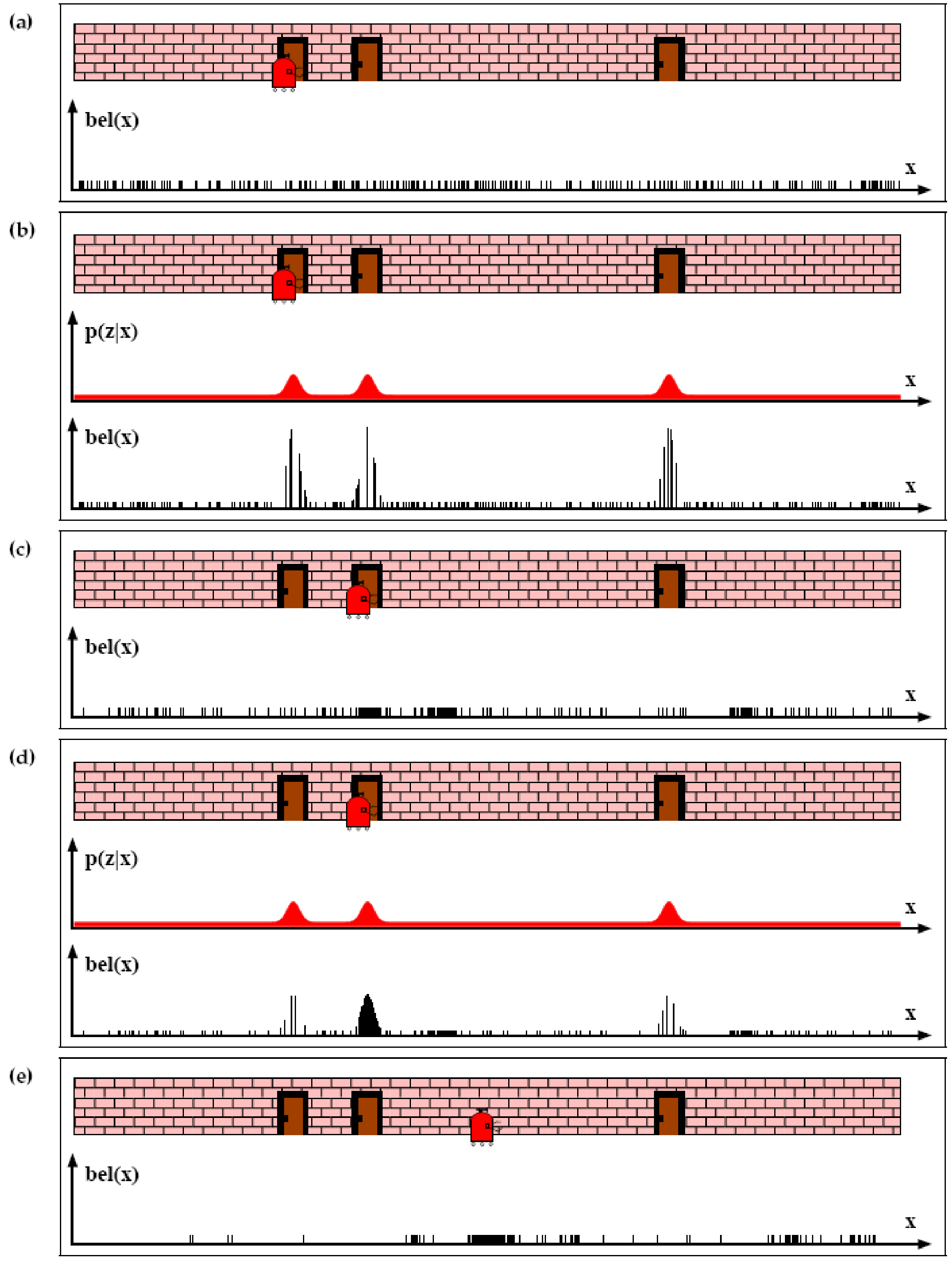
Markov Localization



KF

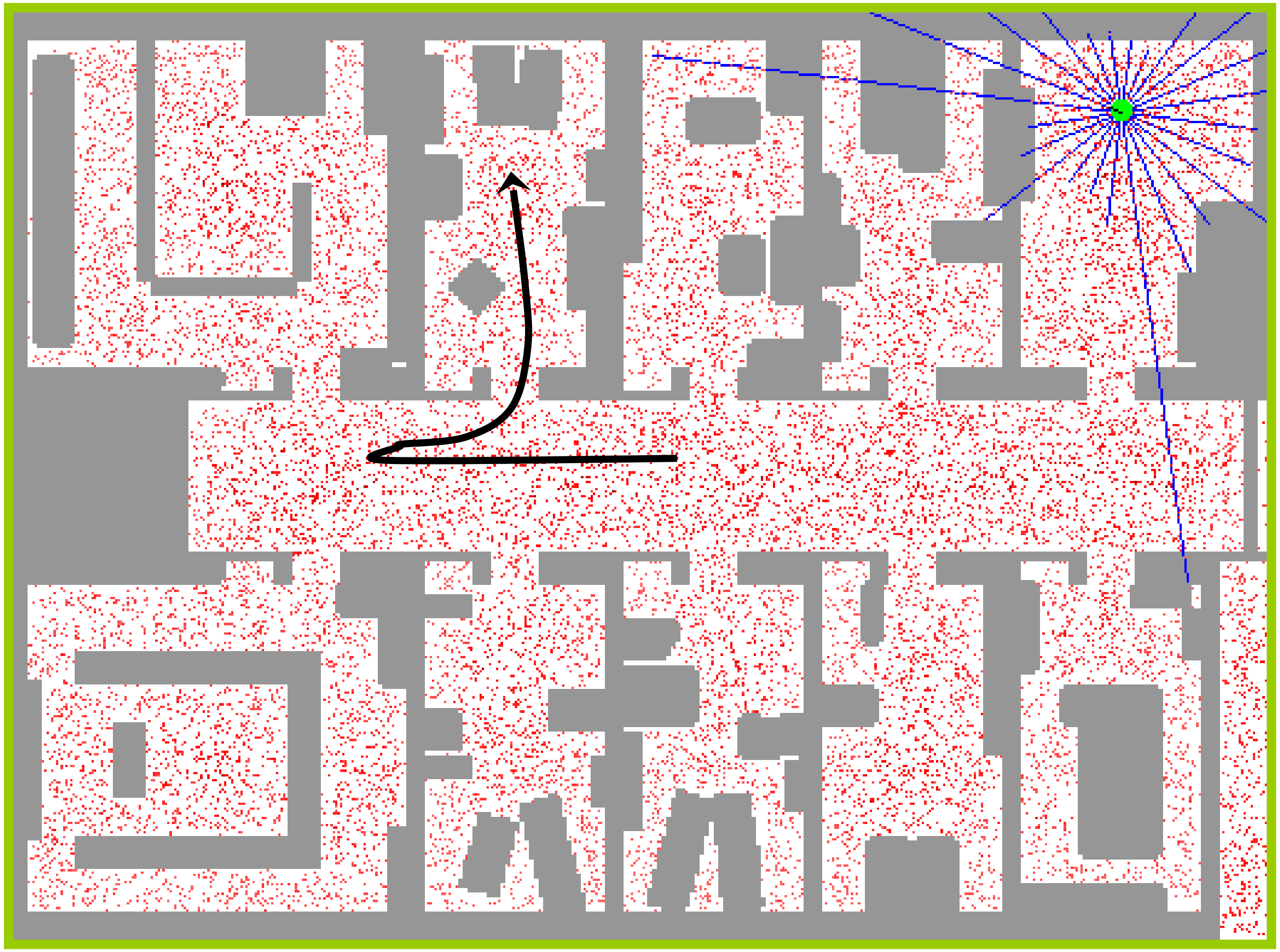


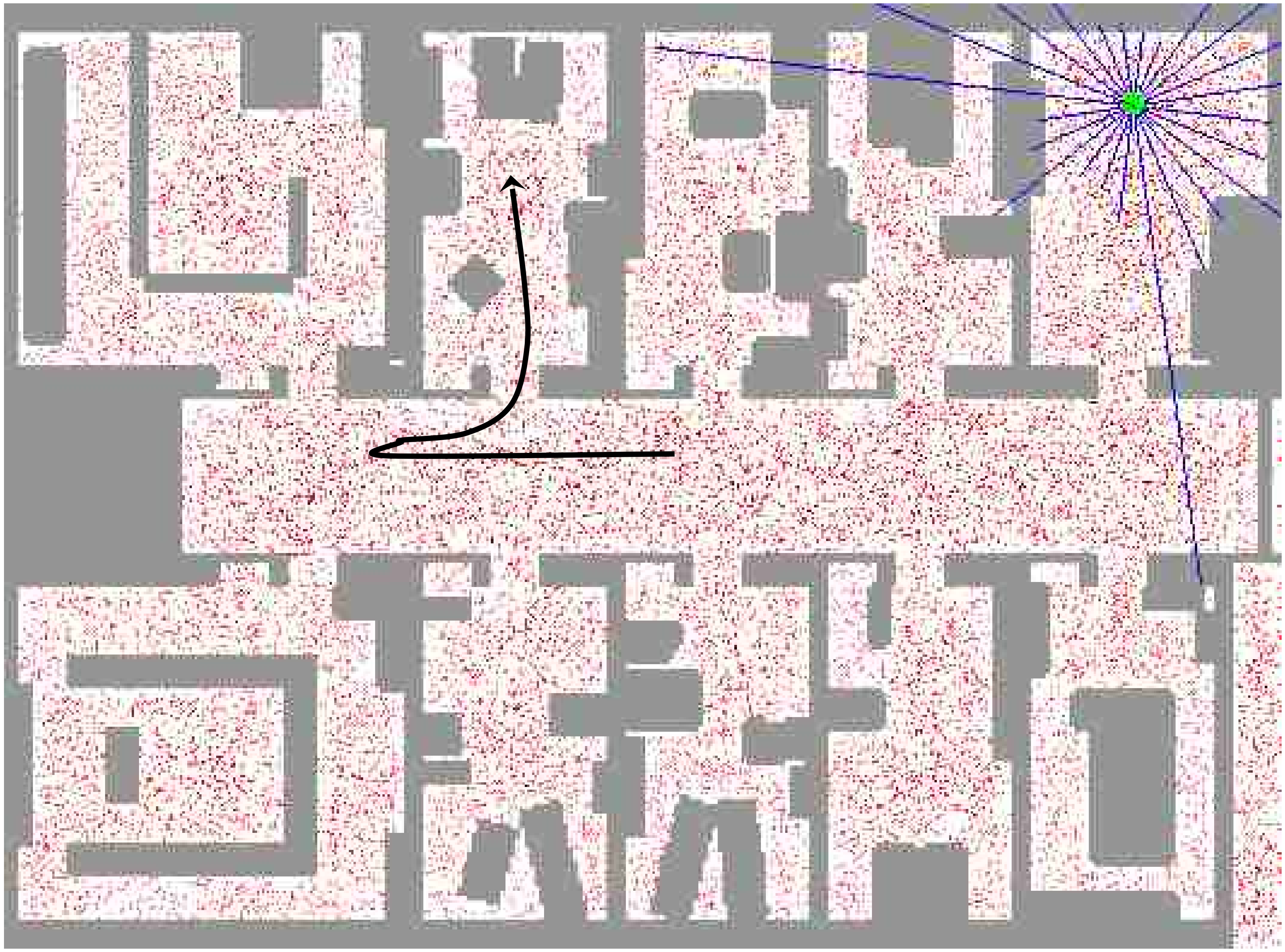
Particle Filter

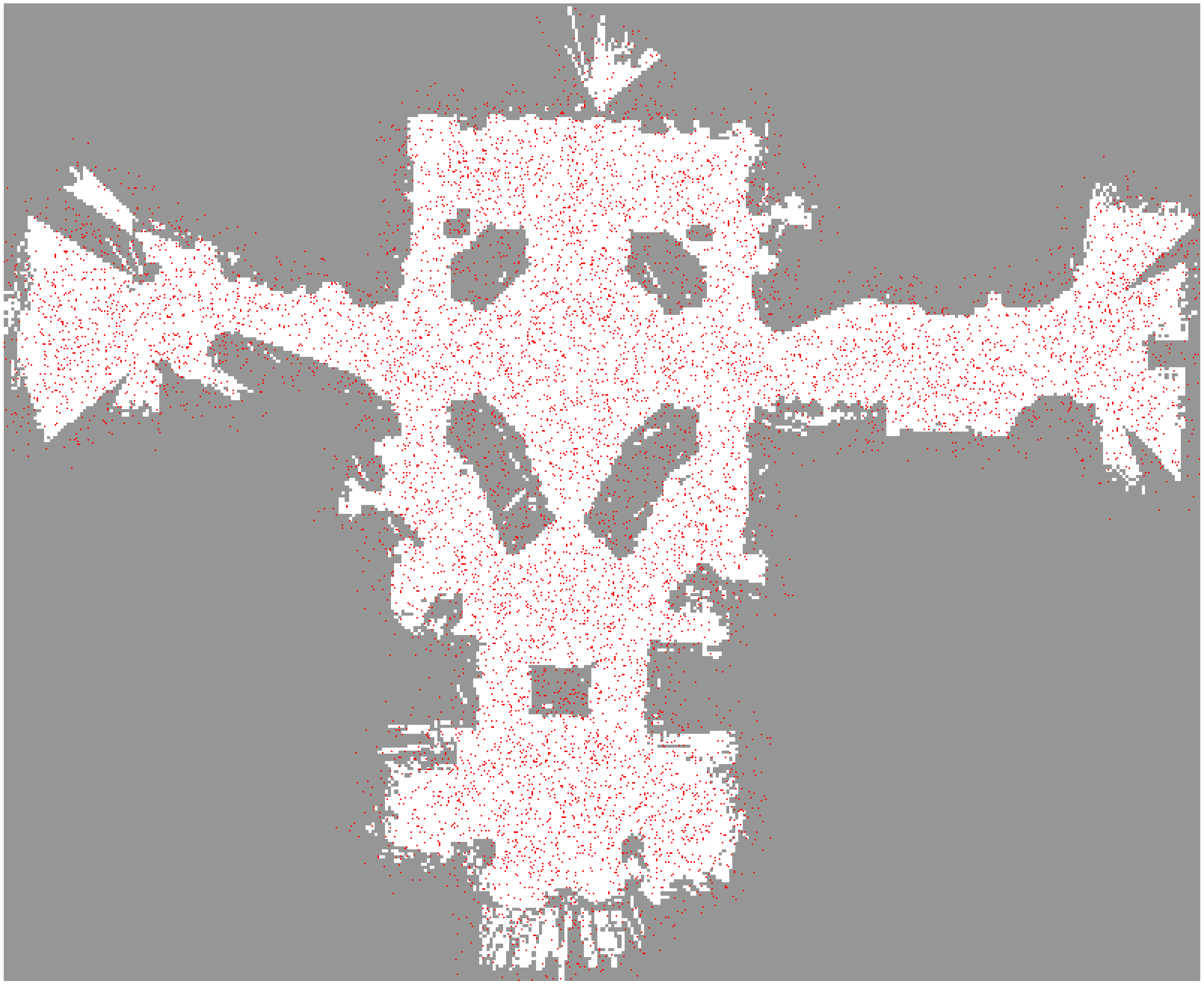


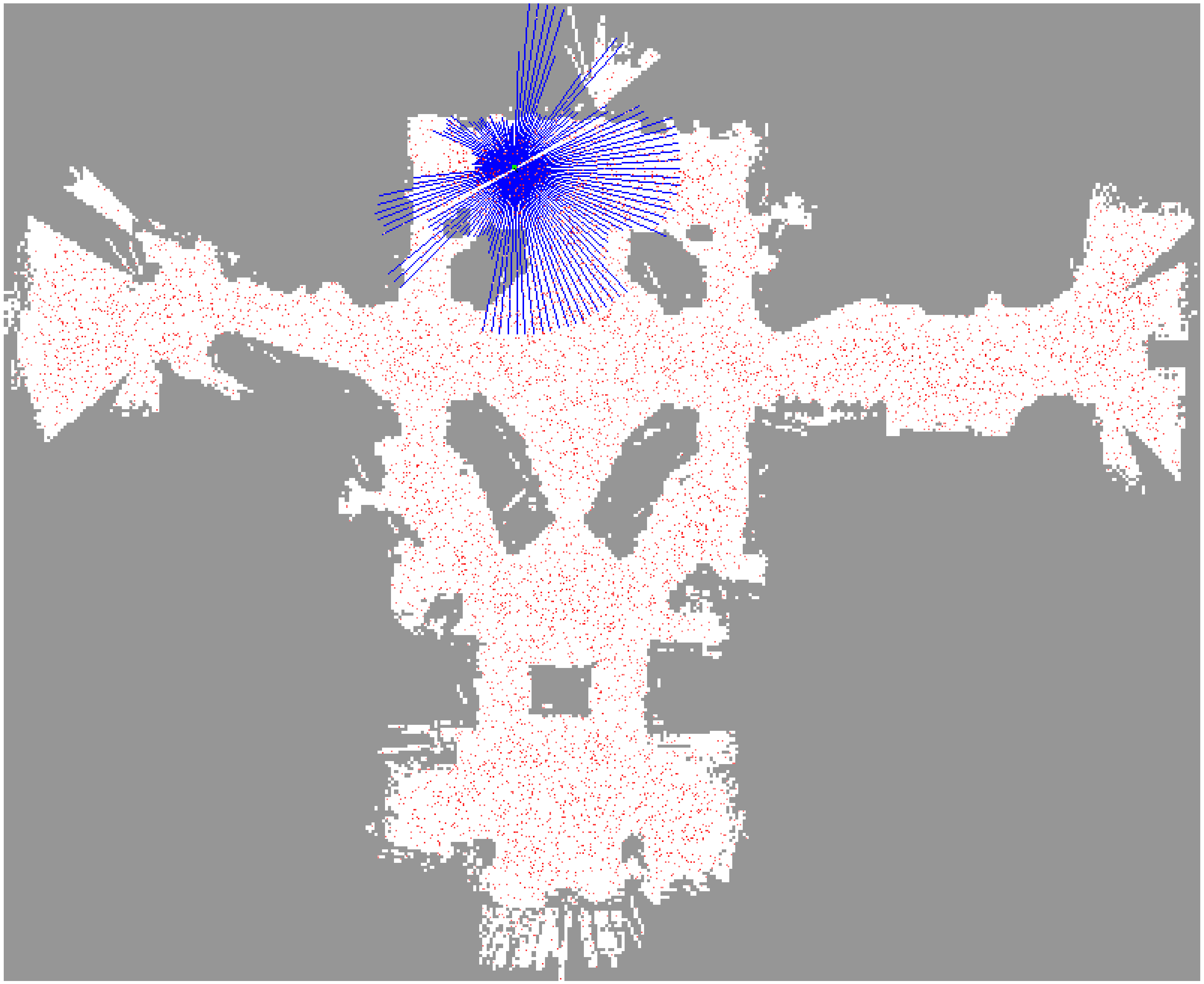
Examples

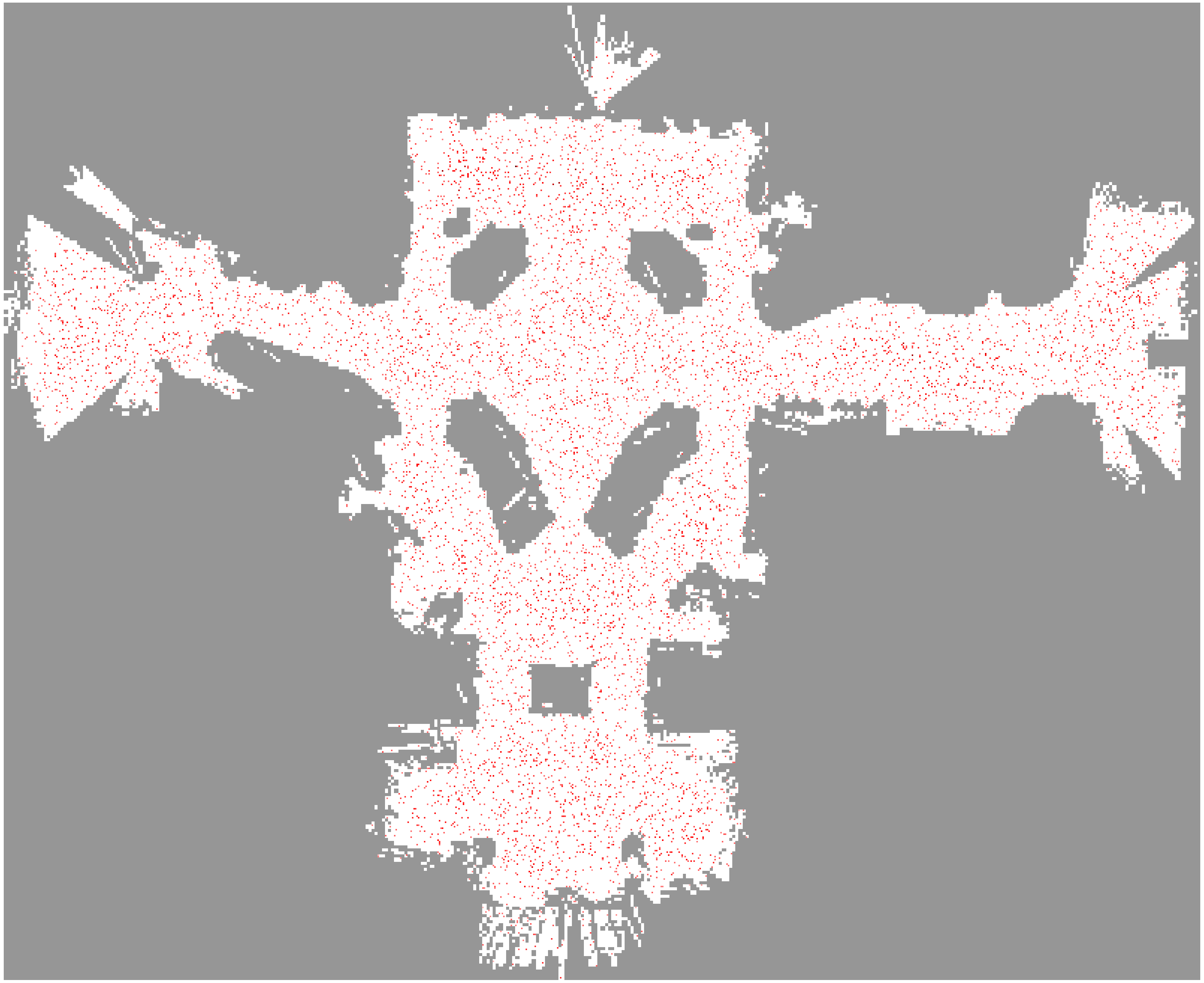
Localizing using Sonar

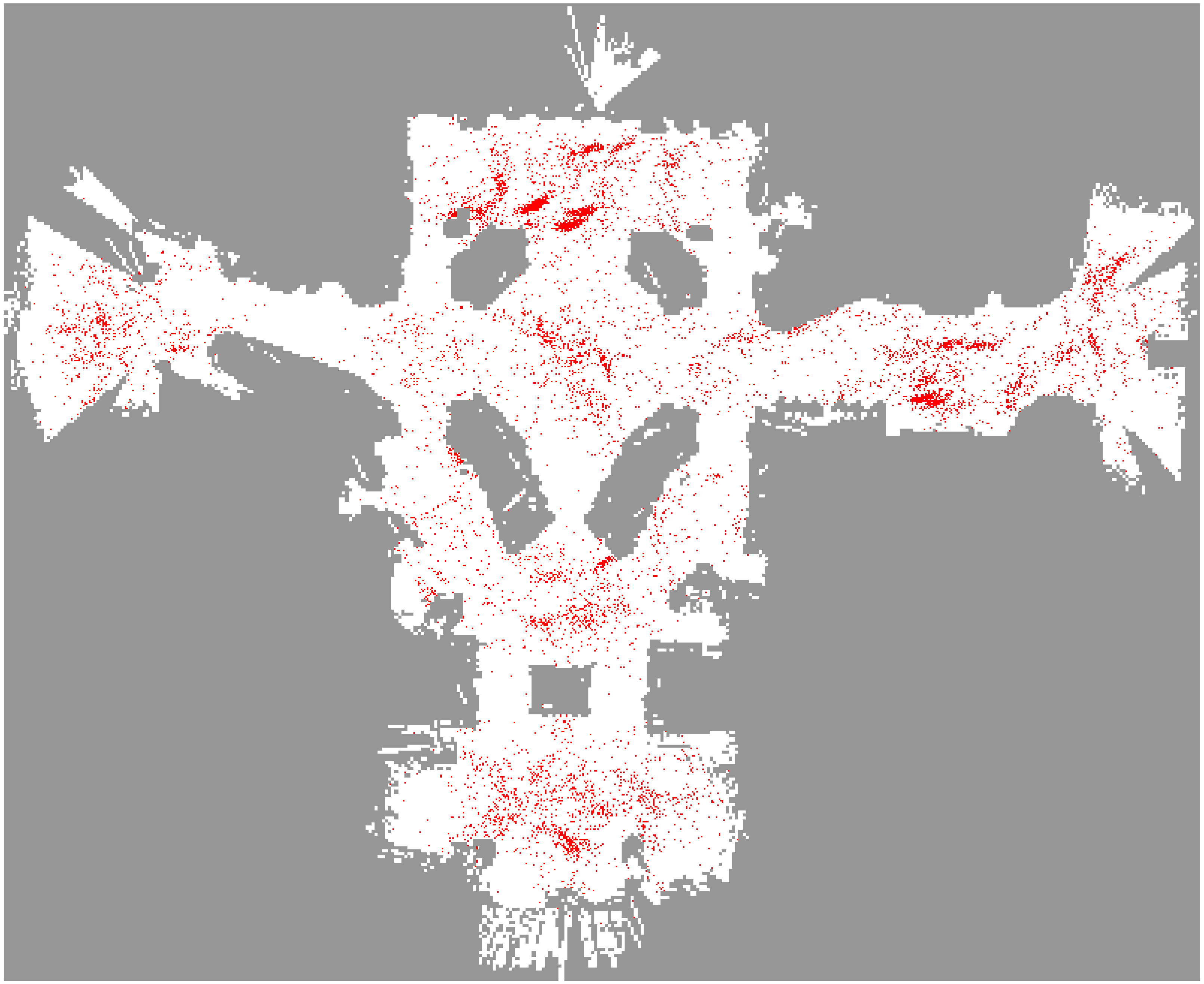


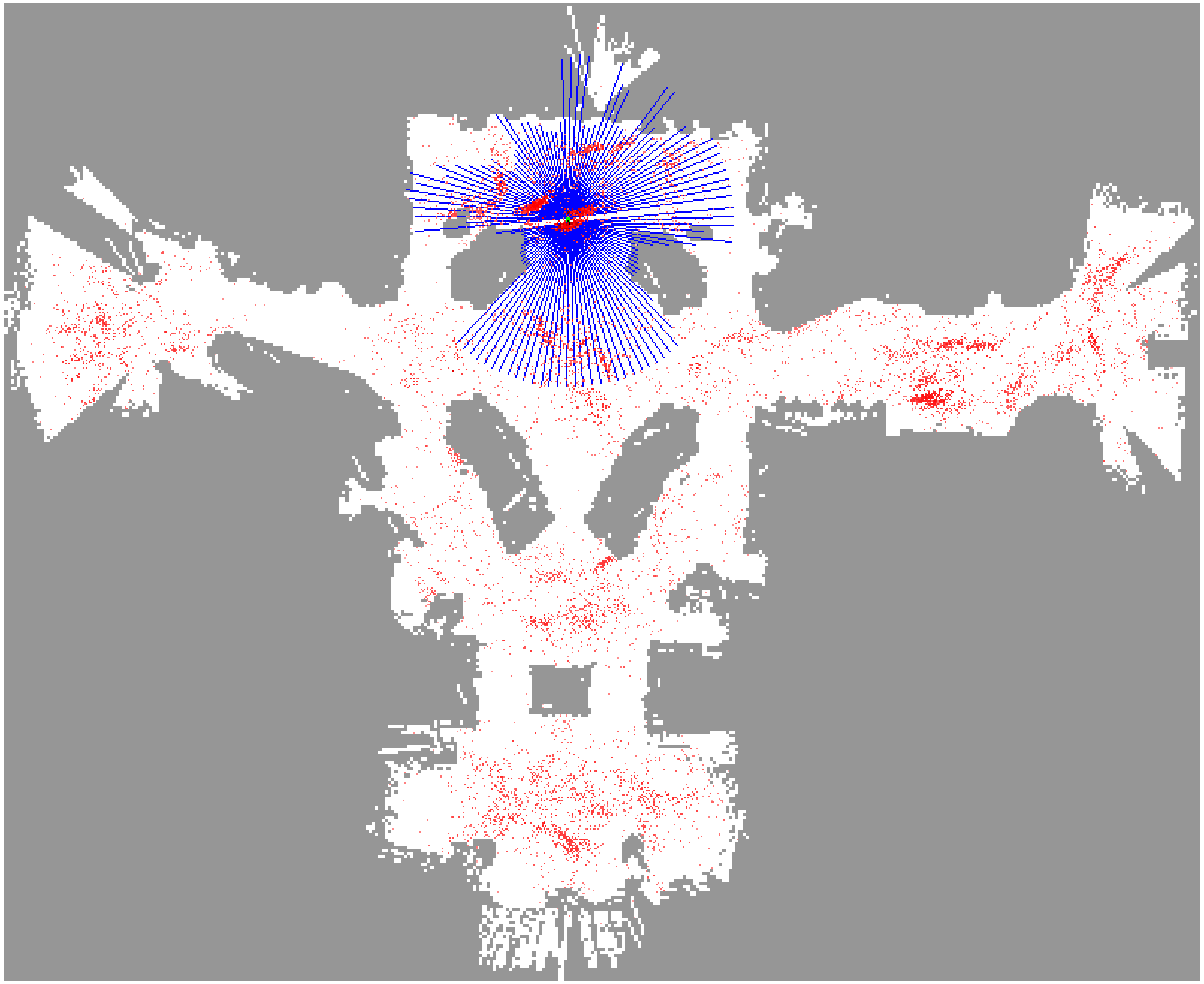


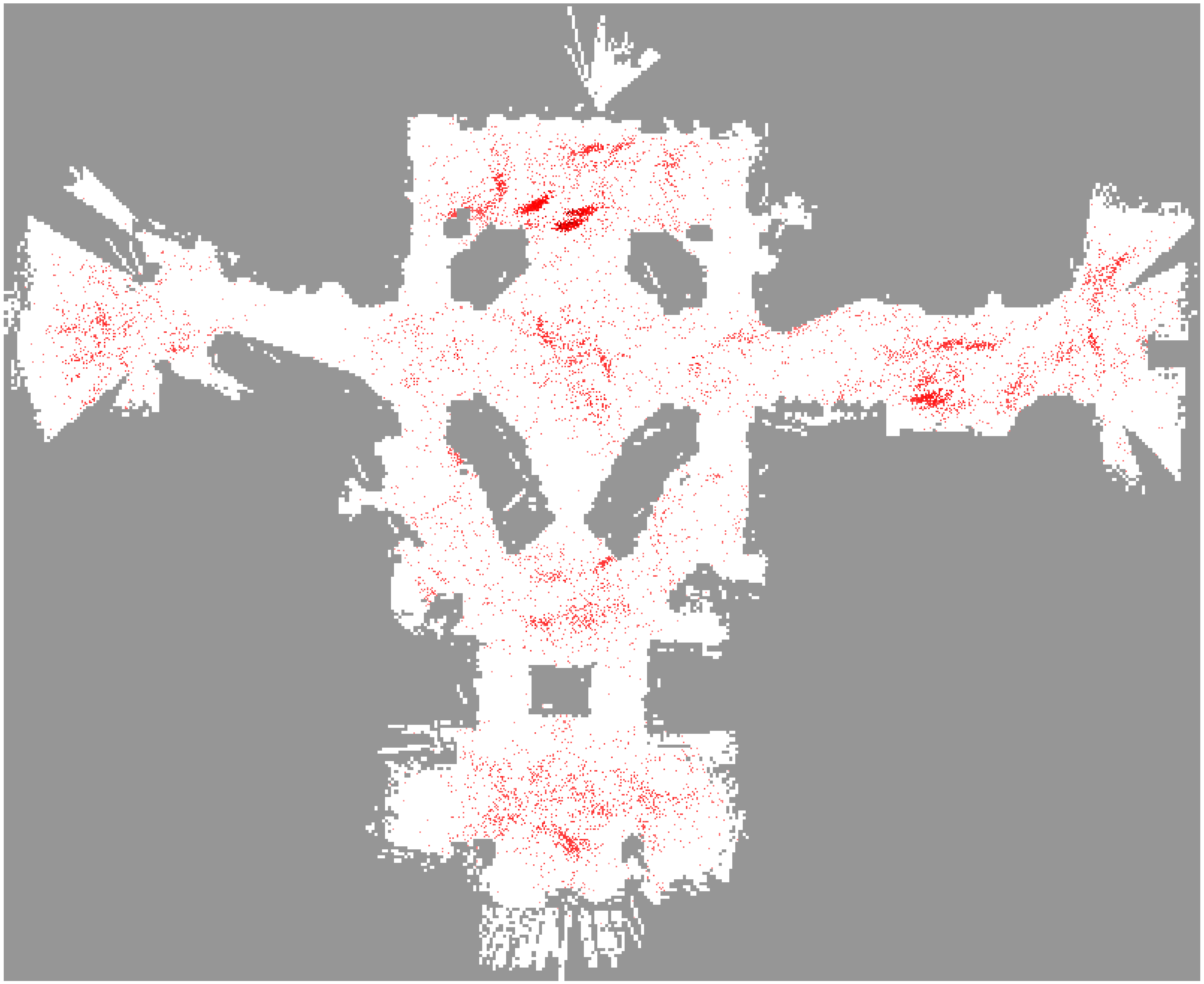


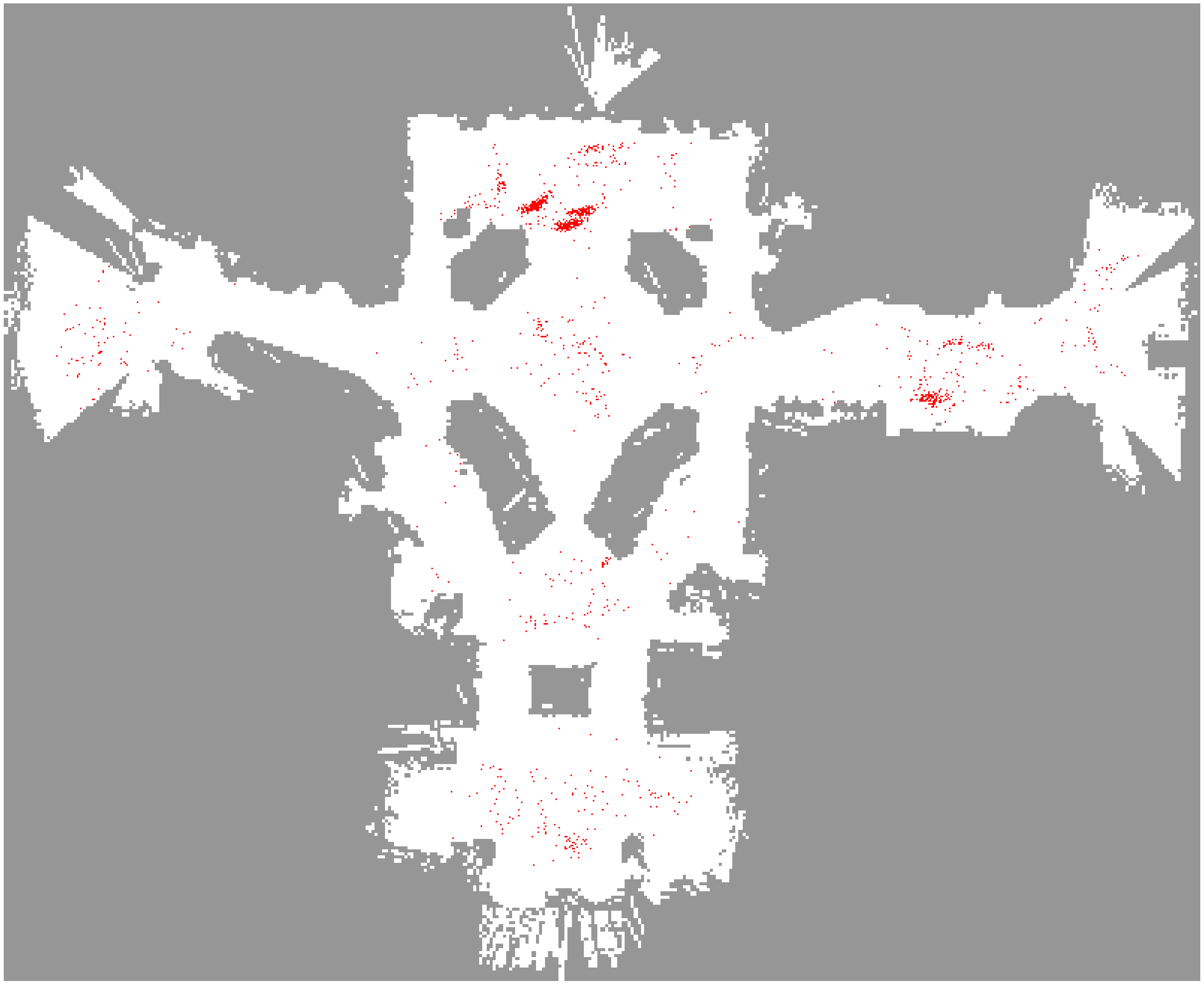


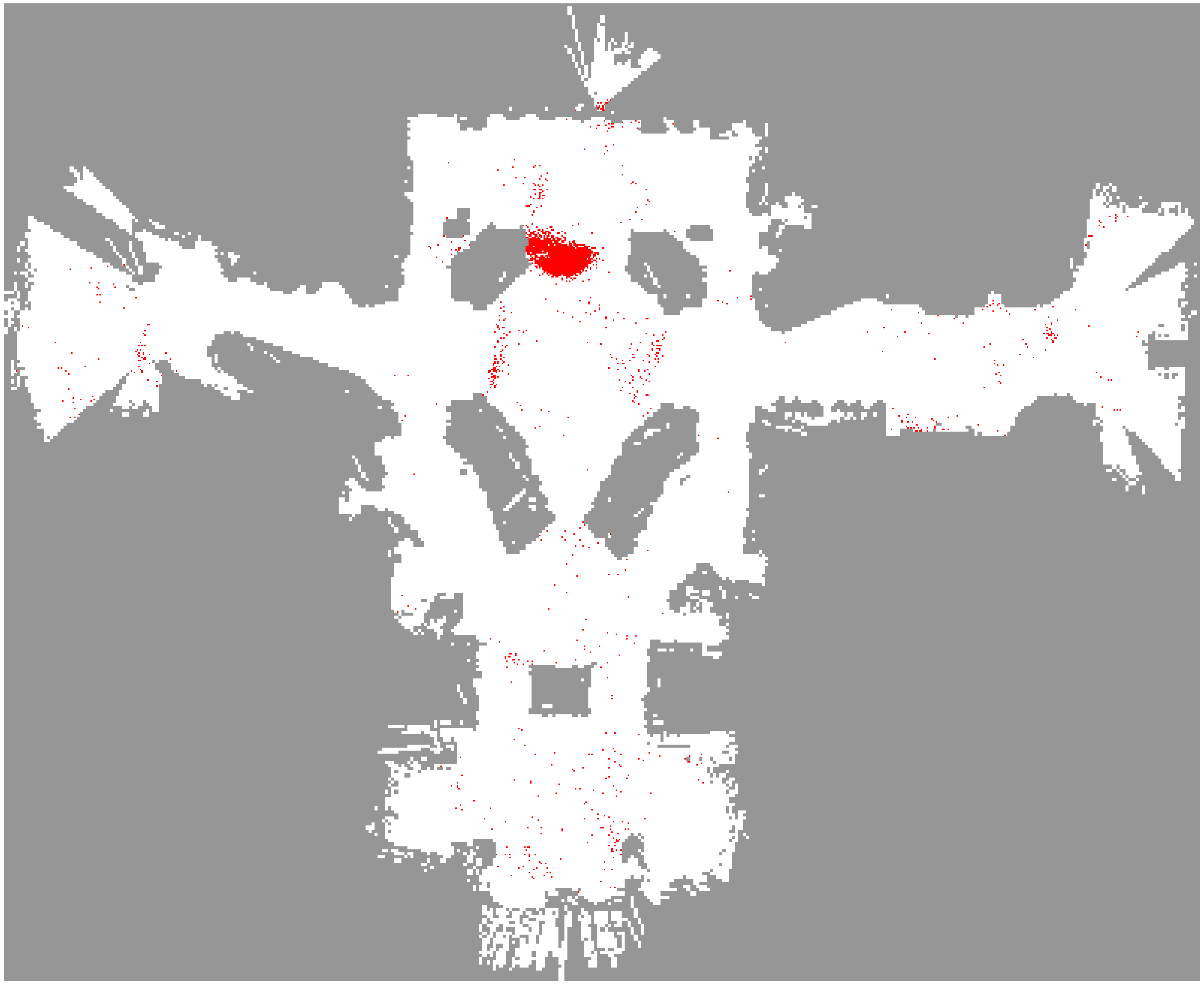


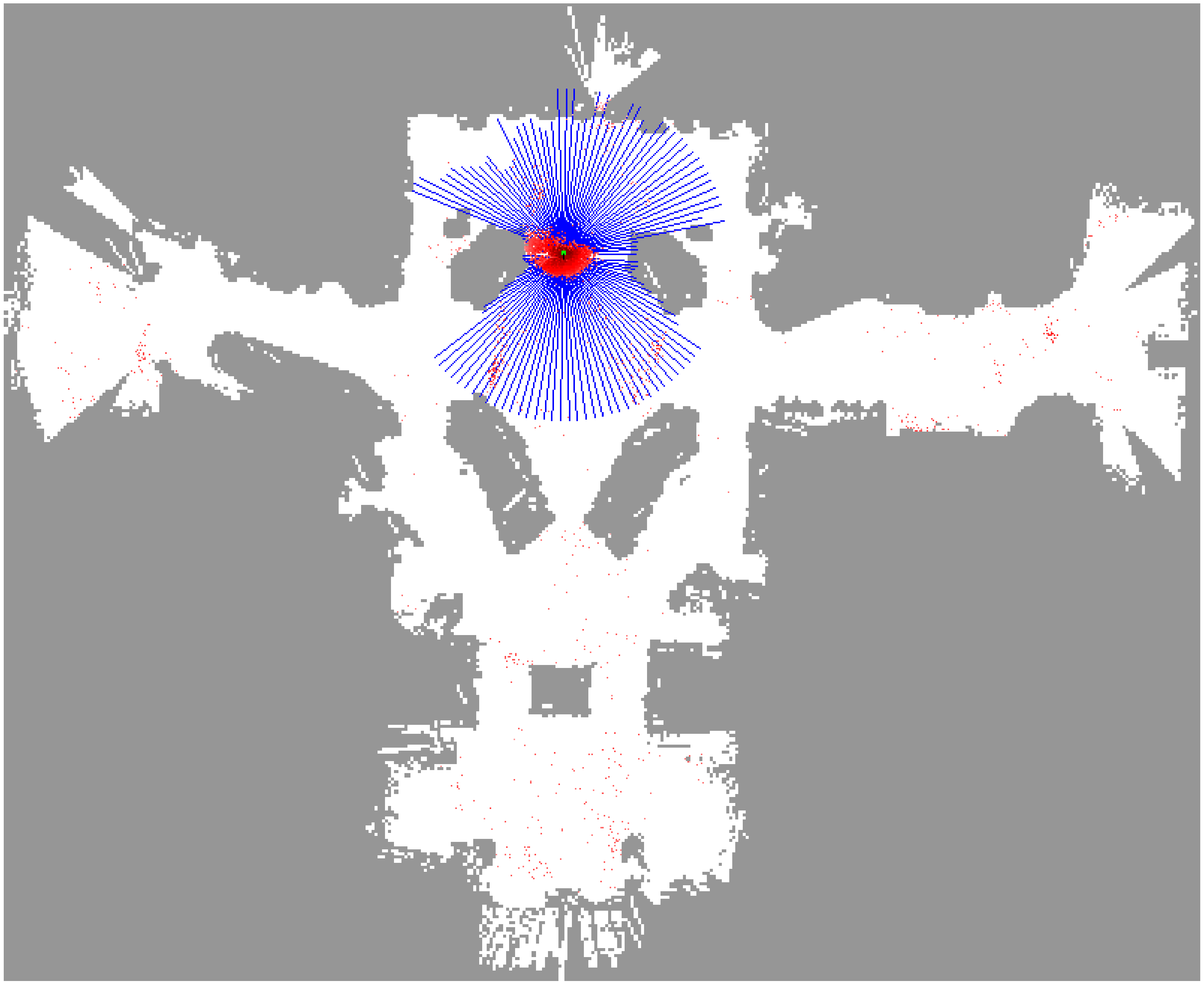


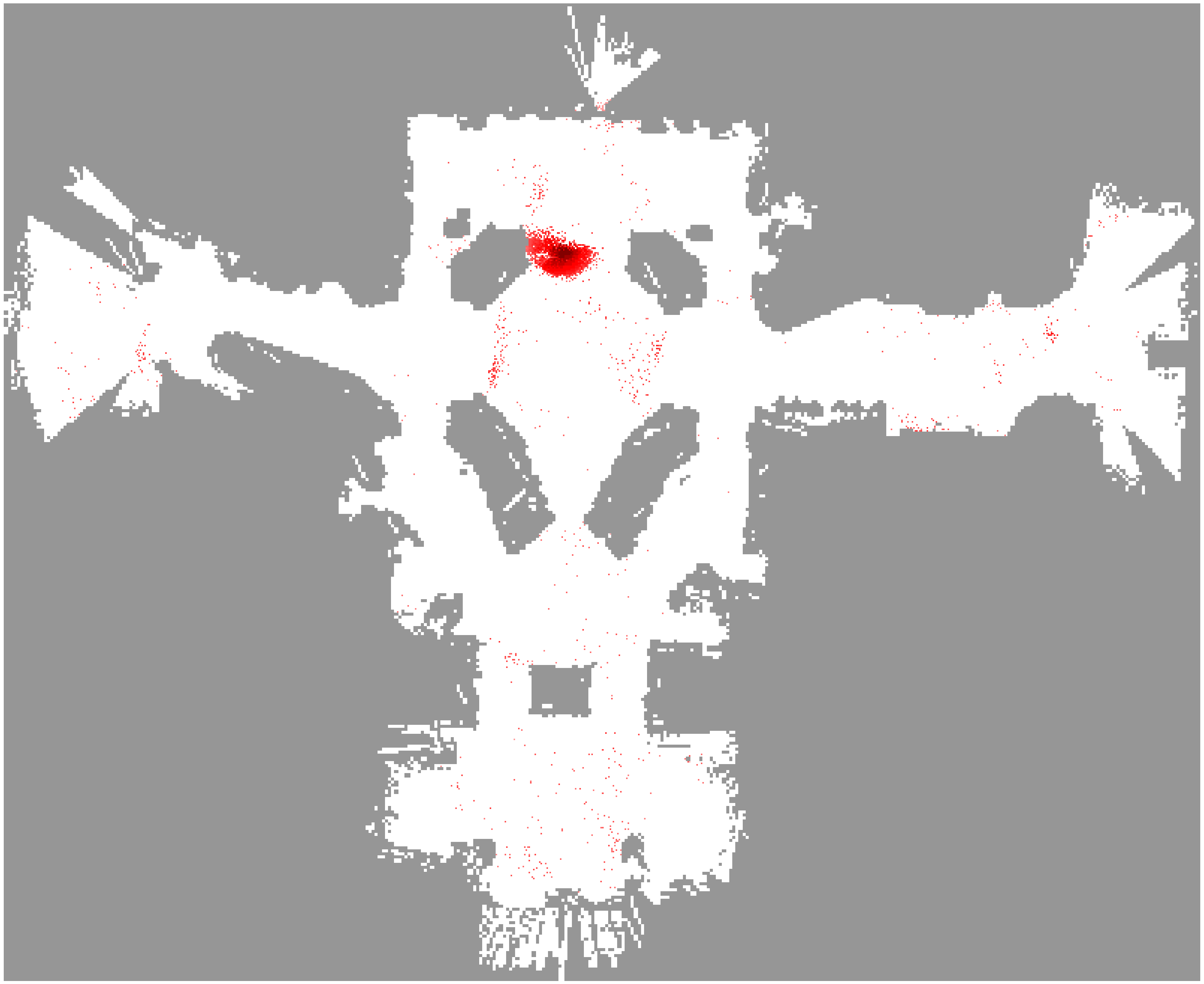


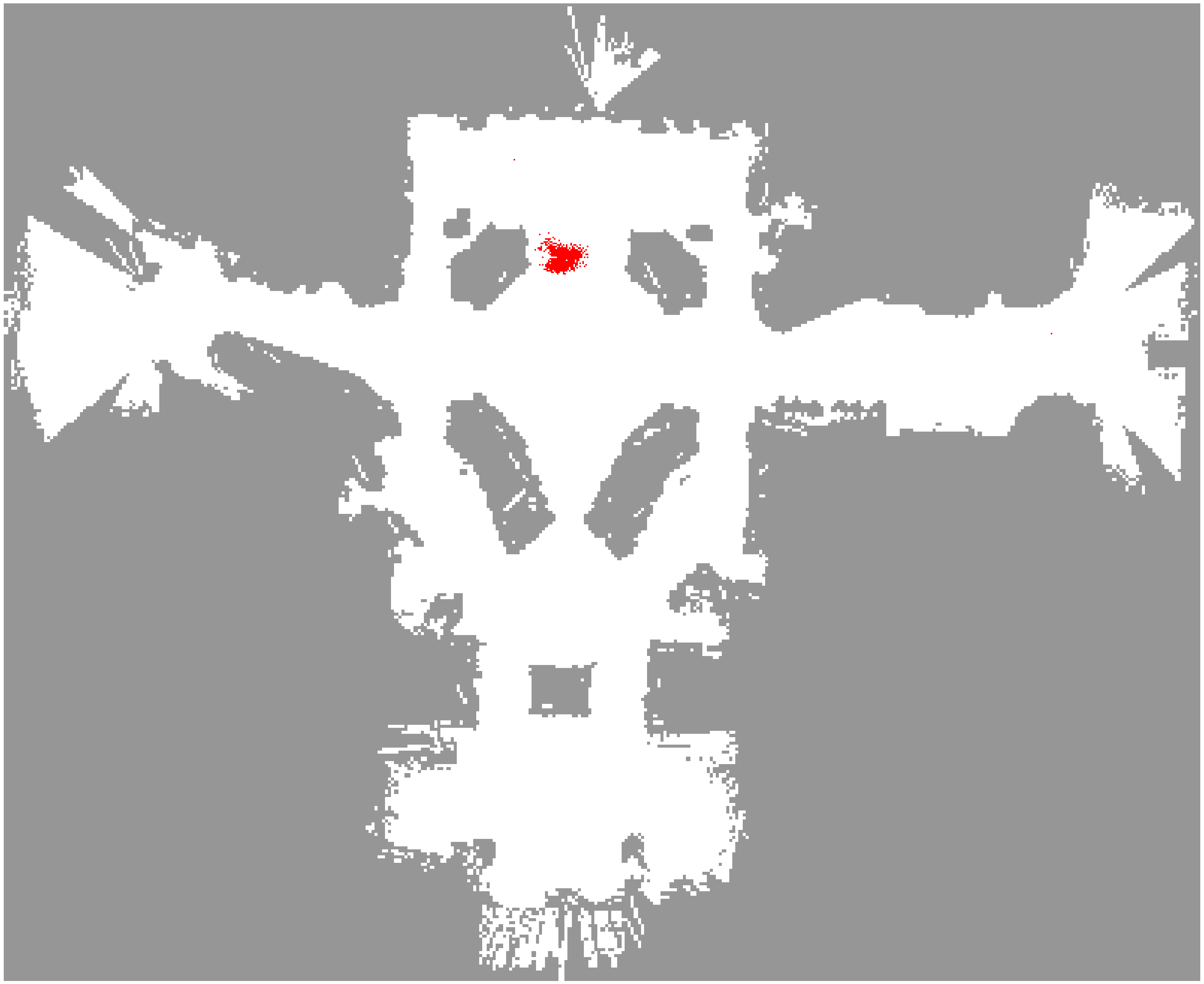


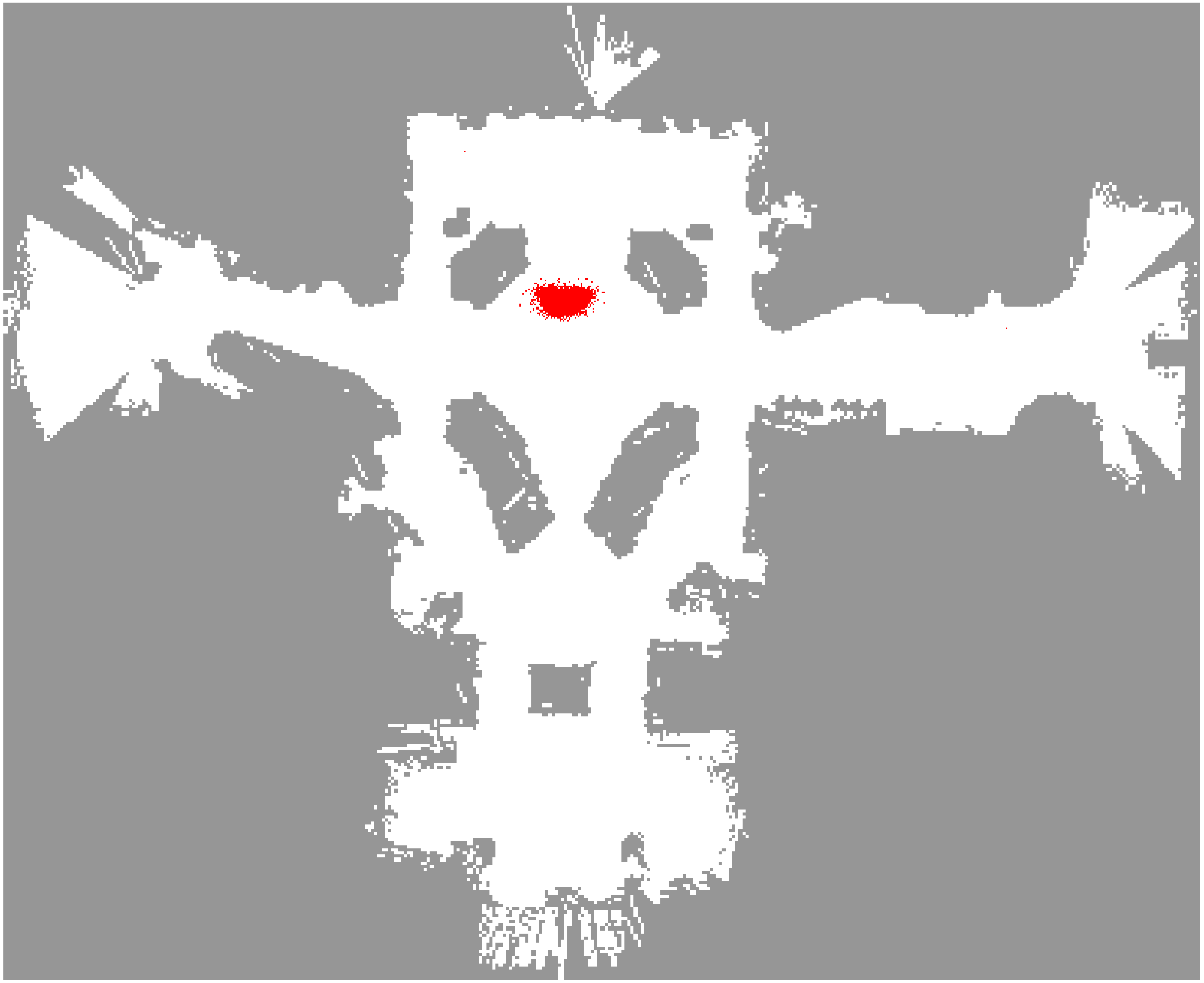


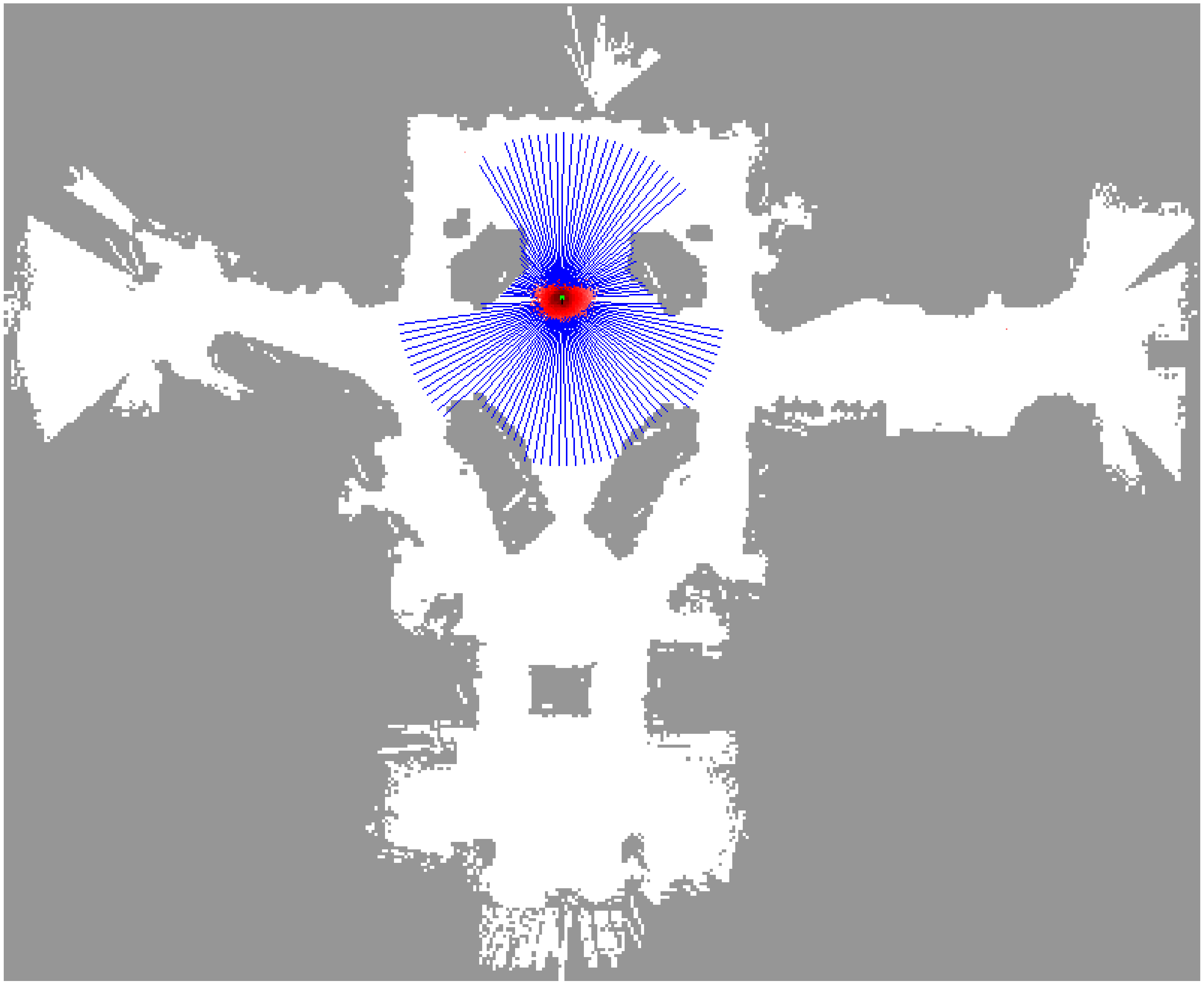


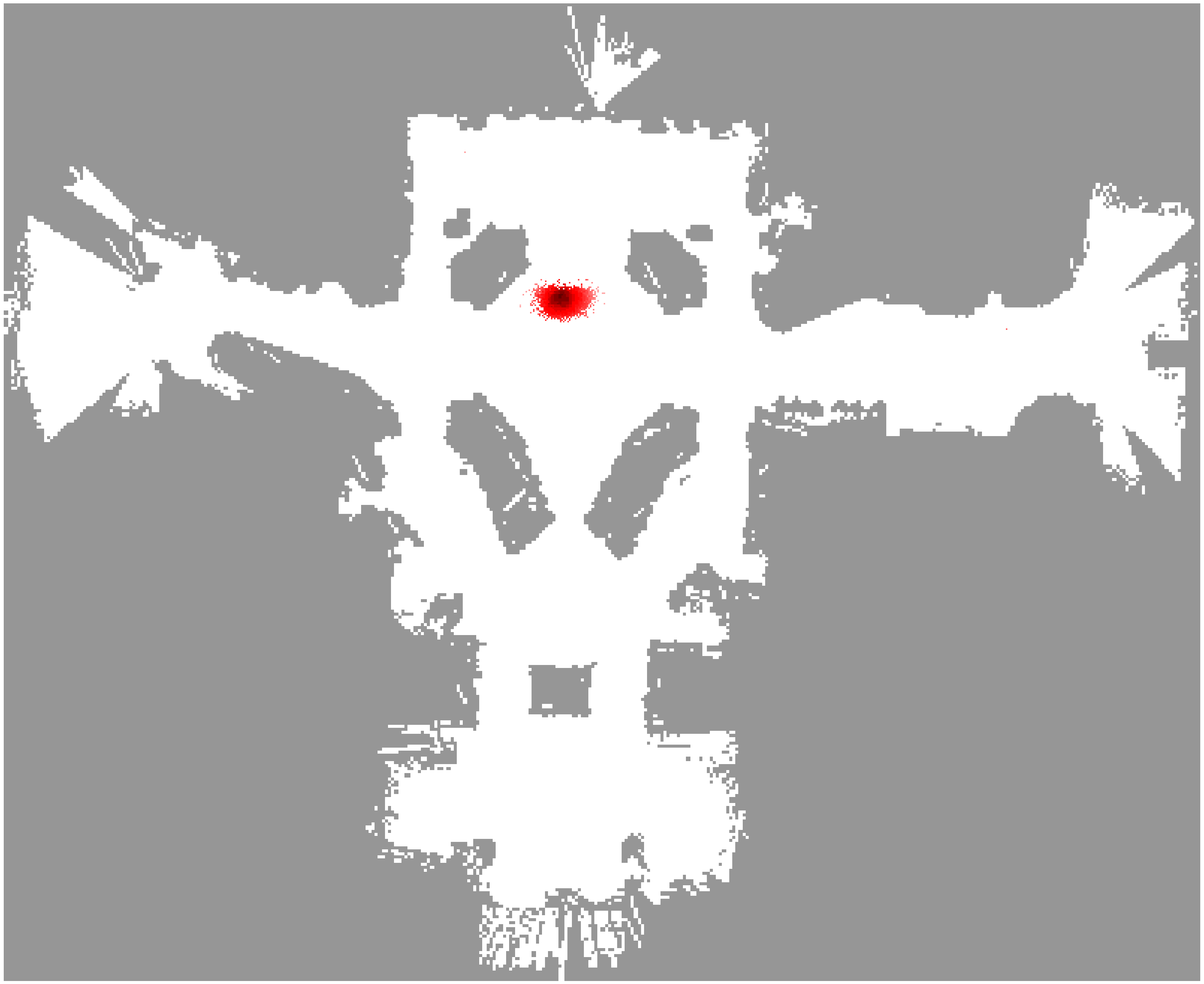


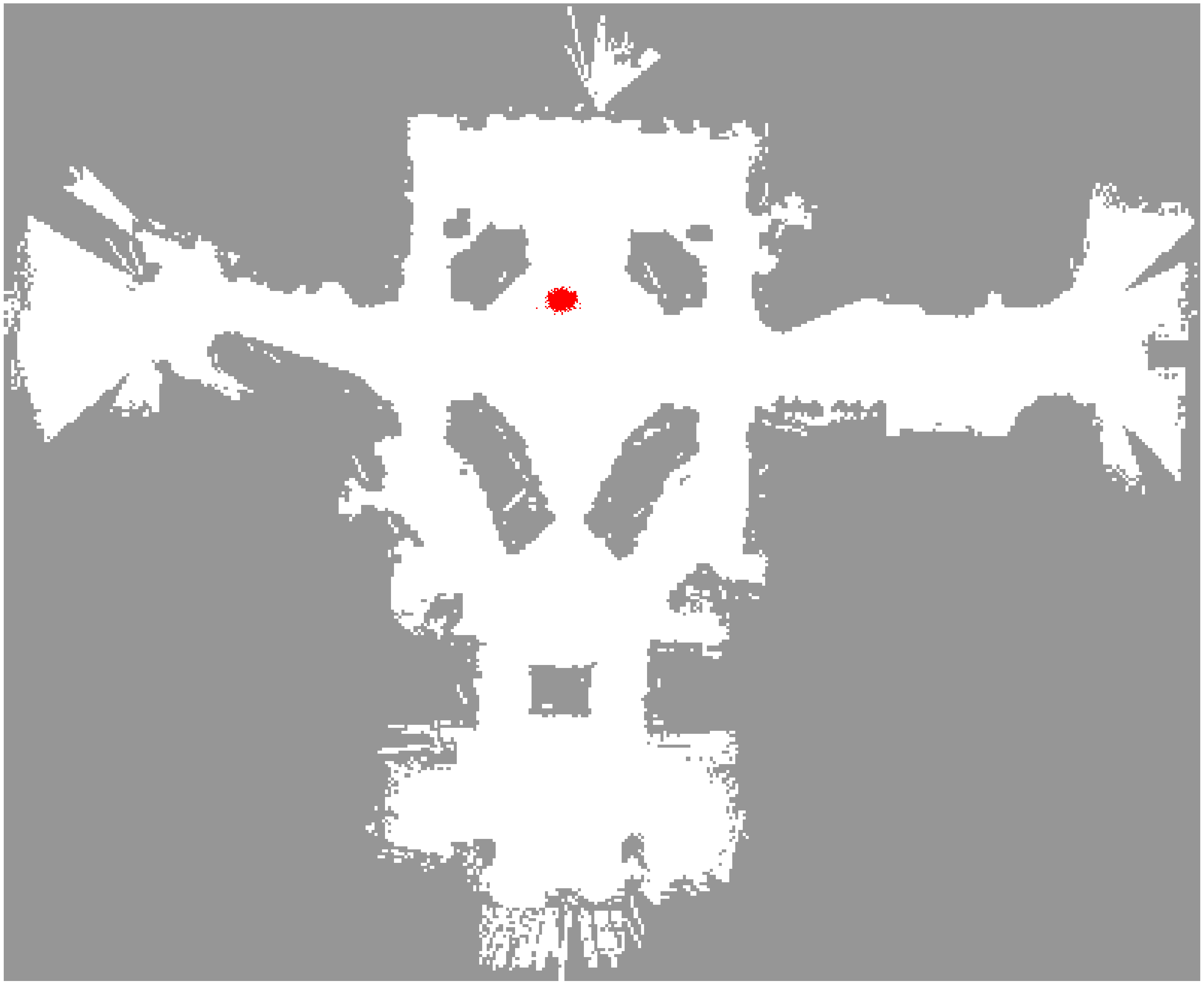




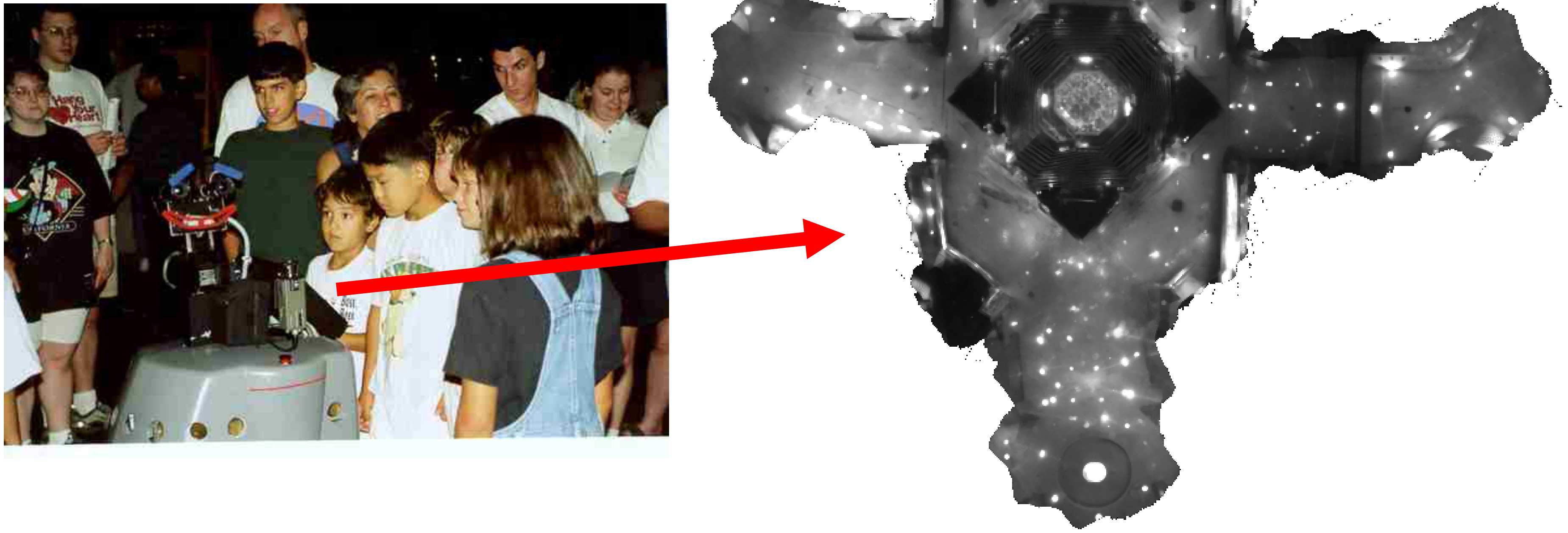




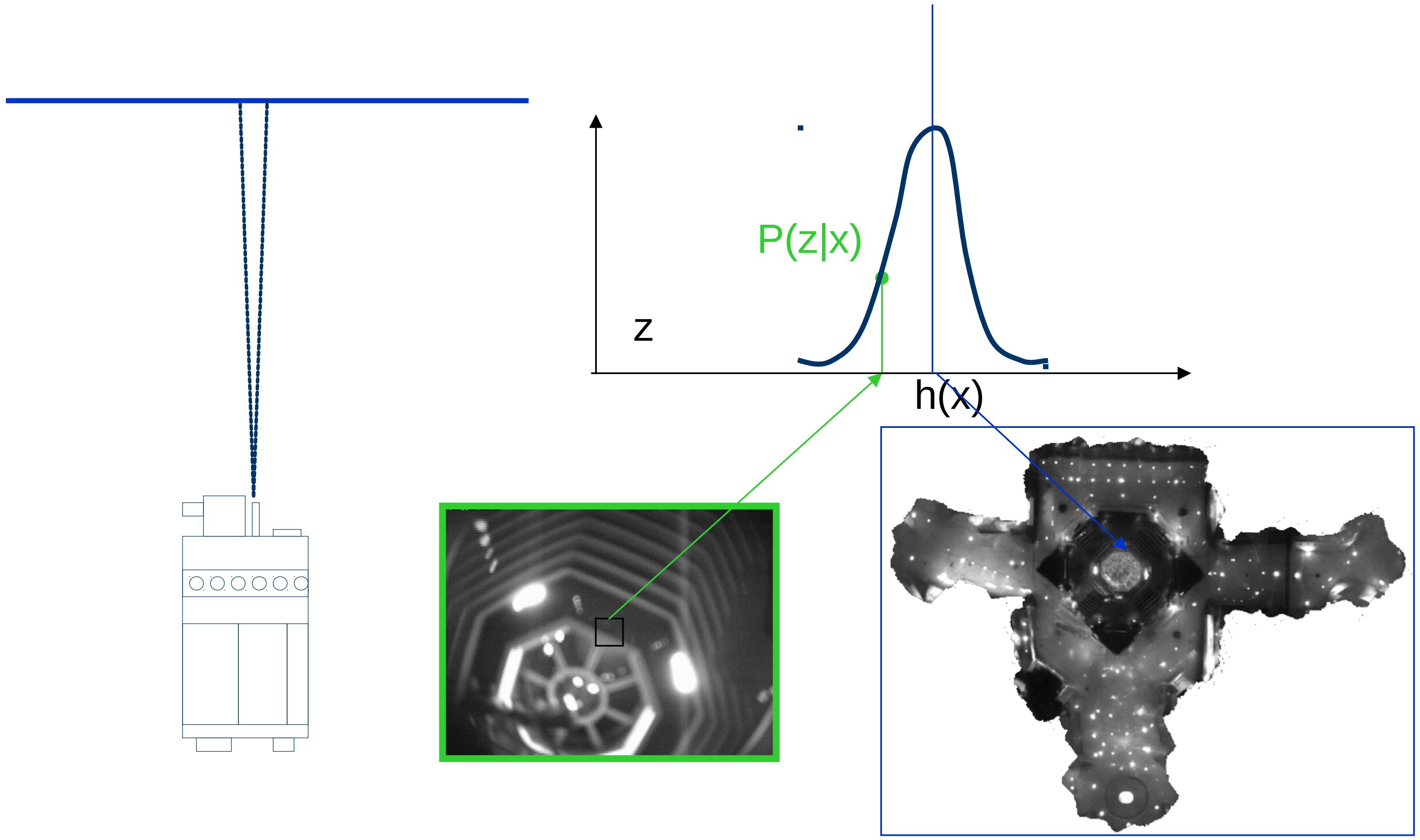




Using Ceiling Maps

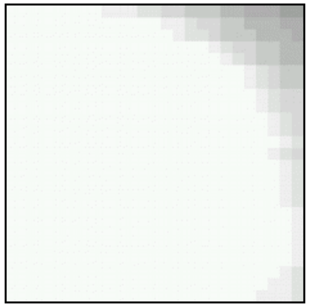


Vision-Based Localization

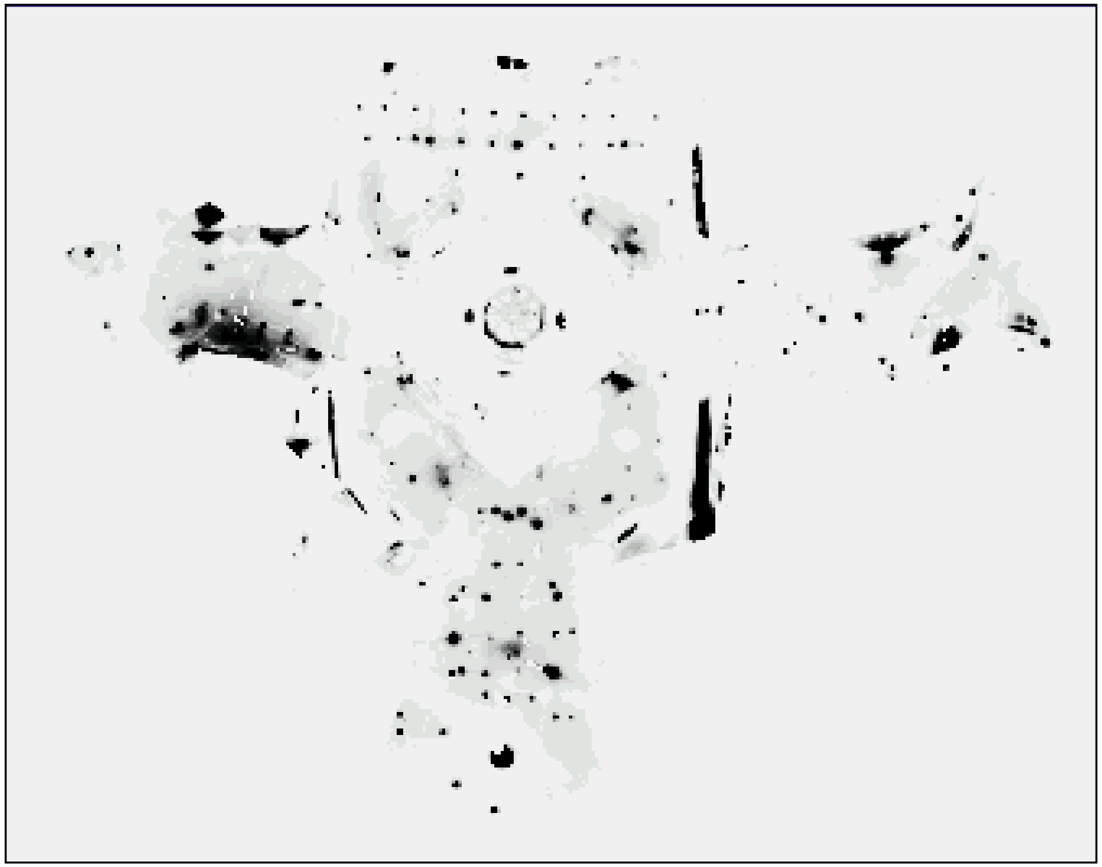


Under a light

Measurement z:

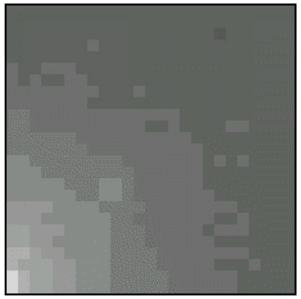


***P(z|x)*:**

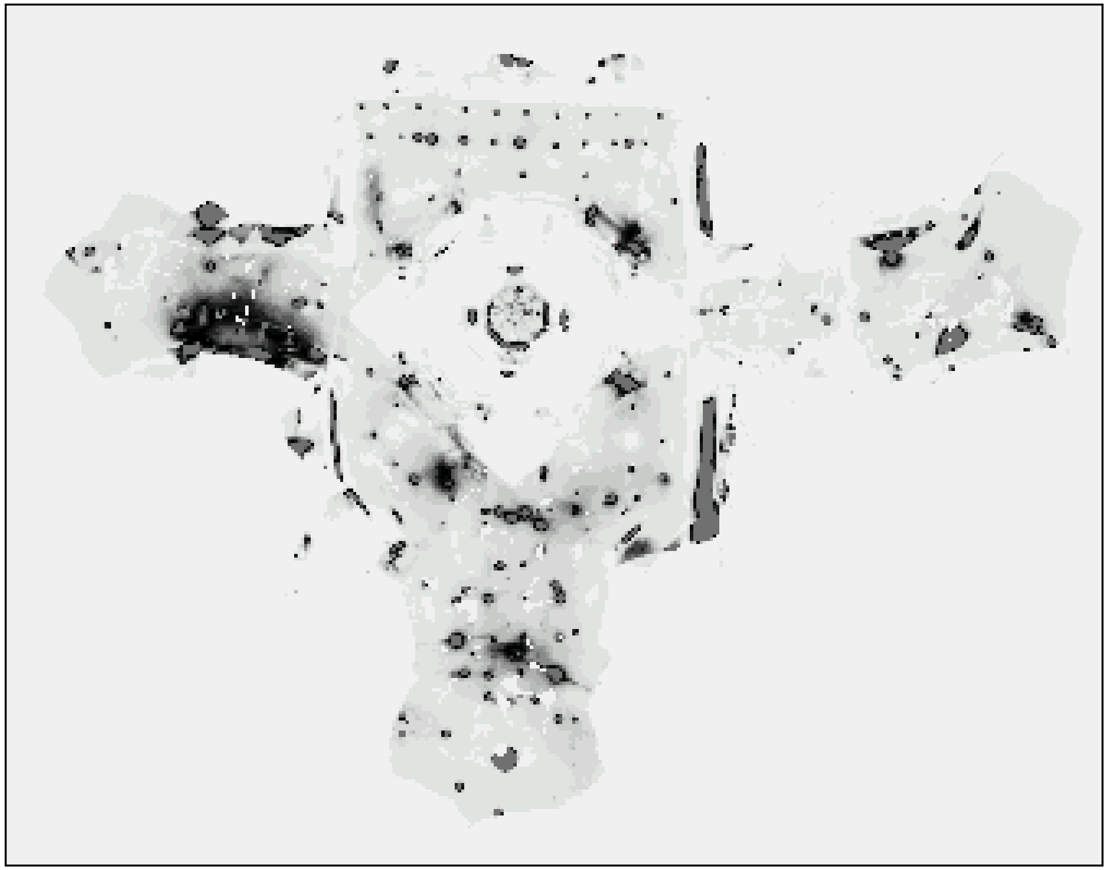


Close to a light

Measurement z:



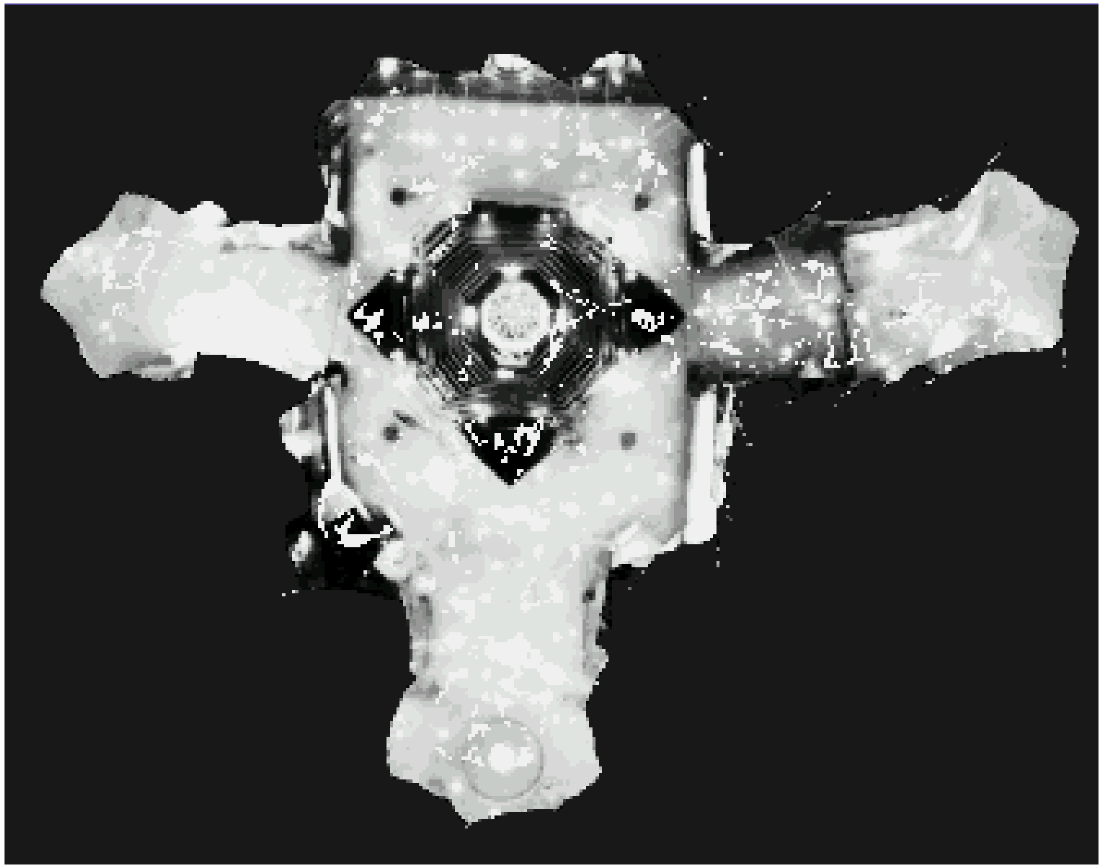
***P(z|x)*:**



Far from a light

Measurement z:

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



Could the robot use both vision and sonar to localize? How?

Summary

Particle filters are an implementation of recursive Bayesian filtering

They represent the posterior by a set of weighted samples.

In the context of localization, the particles are propagated according to the motion model.

They are then weighted according to the likelihood of the observations.

In a re-sampling step, new particles are drawn with a probability proportional to the likelihood of the observation.

Localization and Mapping Project Ideas

- Build maps and localize using vision
- 2D and/or 3D vision
- Ceiling Maps
- Incorporate multiple sources of observations for computing $p(z1,z2,\ldots,zk|x)$
- Integrate existing mapping and localization algorithms into the turtlebot2 code-base

Credits

- Some slides adapted / borrowed from:

Alexander Stoytchev

Sebastian Thrun

THE END