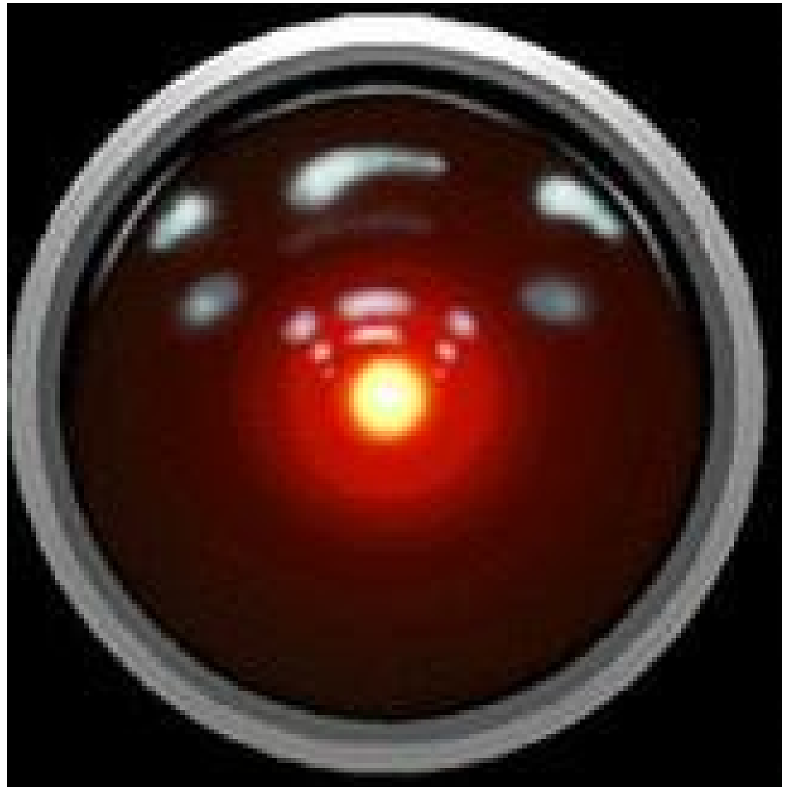
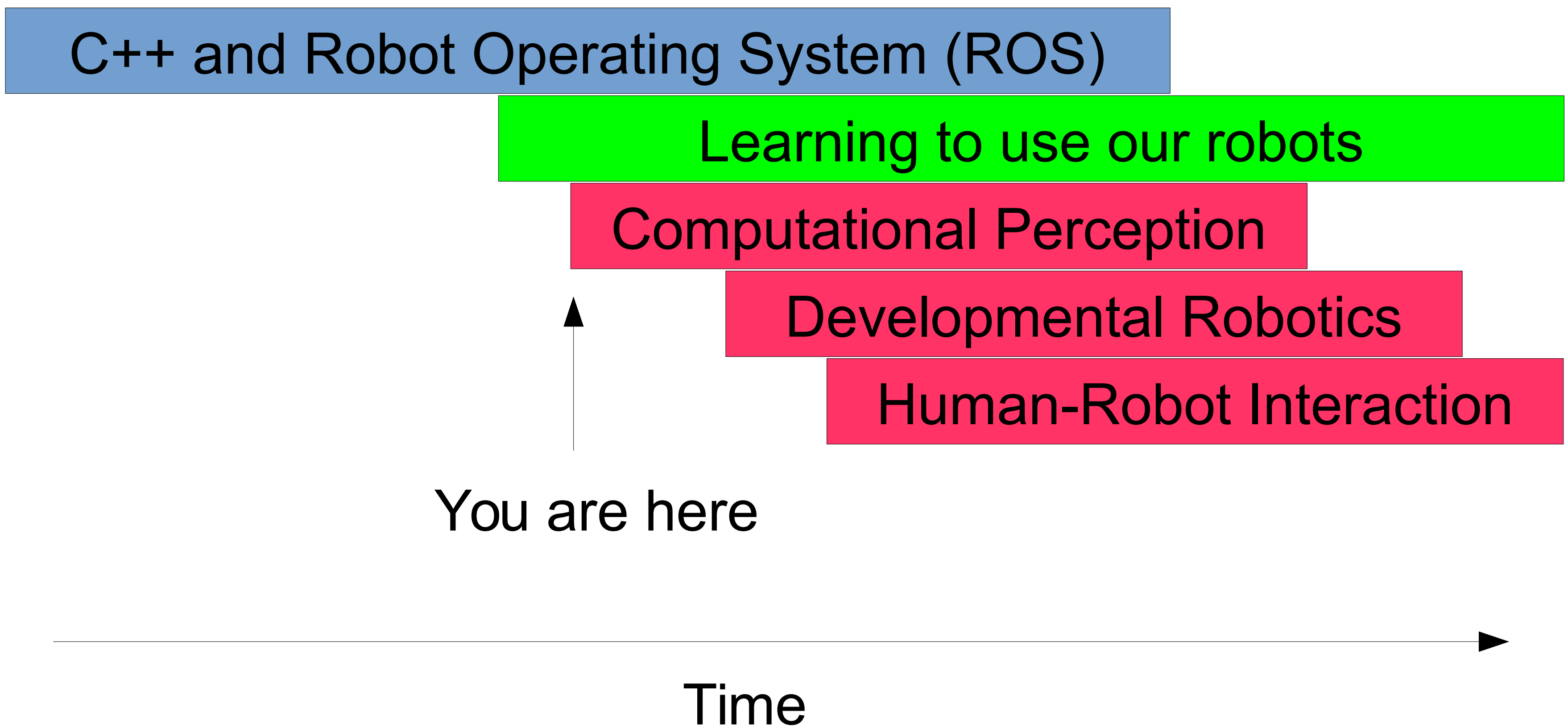


CS 309: Autonomous Intelligent Robotics

Instructor: Jivko Sinapov

http://www.cs.utexas.edu/~jsinapov/teaching/cs309_spring2017/

Semester Schedule



Announcements

A few volunteers are needed for ExploreUT this Saturday, March 4

Setup starts at 9:45 am and event runs from 11:00 to 4 pm

If you can make it, send me an email!

Final Project Timeline

- Project Proposal due: Apr. 3rd
- Project Presentations / Demos: Finals Period assigned for this class
- Final Report due: May 11th

Project Proposal Guidelines

- Work in groups of 2-3
- Preferably, team up with people with different skills than yours
- Purpose of the proposal is to give you an outline / roadmap

Project Proposal Guidelines

- Each proposal should be about 3-4 pages
- Each proposal should include:
  - What is the application / task / problem?
  - Any previous experience you may have in that area
  - What do you expect to achieve by the end of the semester?
  - How do you plan to evaluate whether it works or not?
  - Related work in robotics
  - A timeline / schedule of progress and milestones

Project Proposal Guidelines

- Organization: your proposal should have sections and headings (don't just submit one long essay)
- For example:
  - Introduction / problem formulation
  - Related Work in Robotics
  - Proposed approach / software
  - Proposed evaluation
  - Summary of anticipated end result

Project Ideas

Help the robot “see” something it currently cannot

Help the robot “hear” something (e.g., the elevator sound)

Help the robot “do” something (e.g., follow a person)

Final Project Timeline

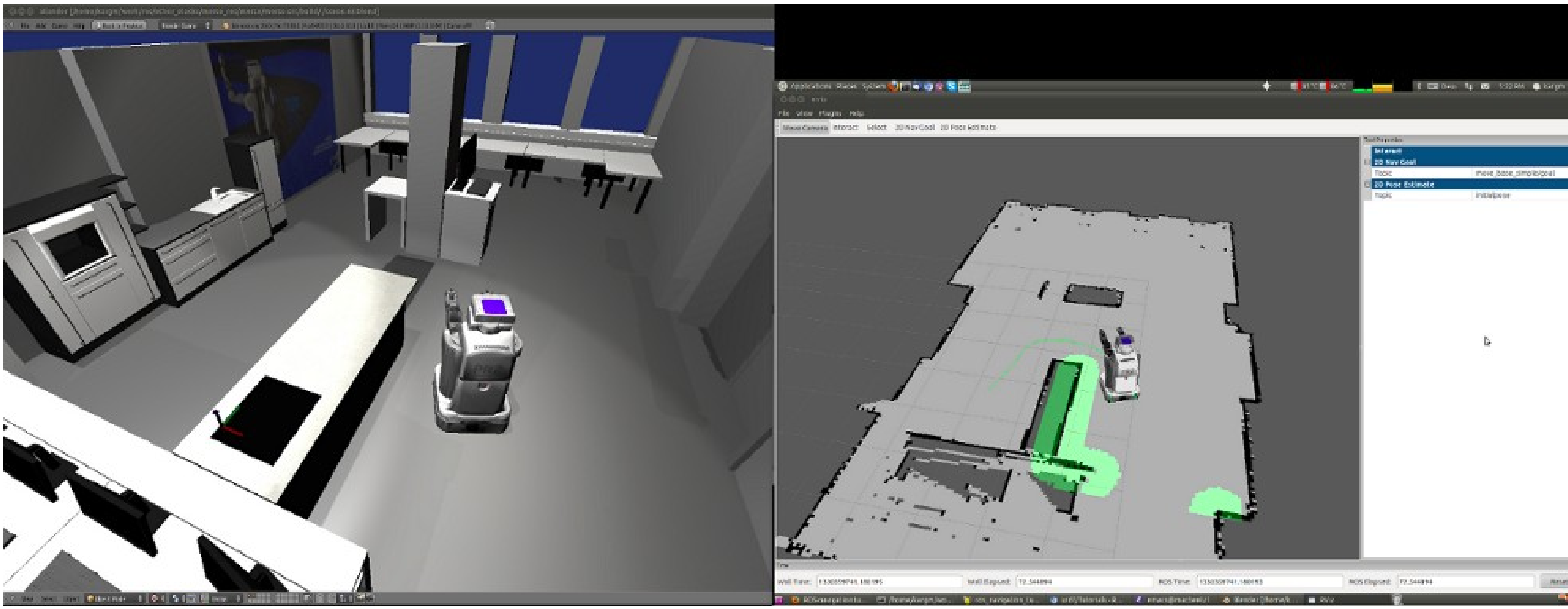
The most important thing is to start early, and discuss your ideas with the TAs, mentors and myself. We'll point you to a starting point, describe functionality that already exists, and help refine your ideas.

Where in the world is the robot?
(aka localization and mapping)

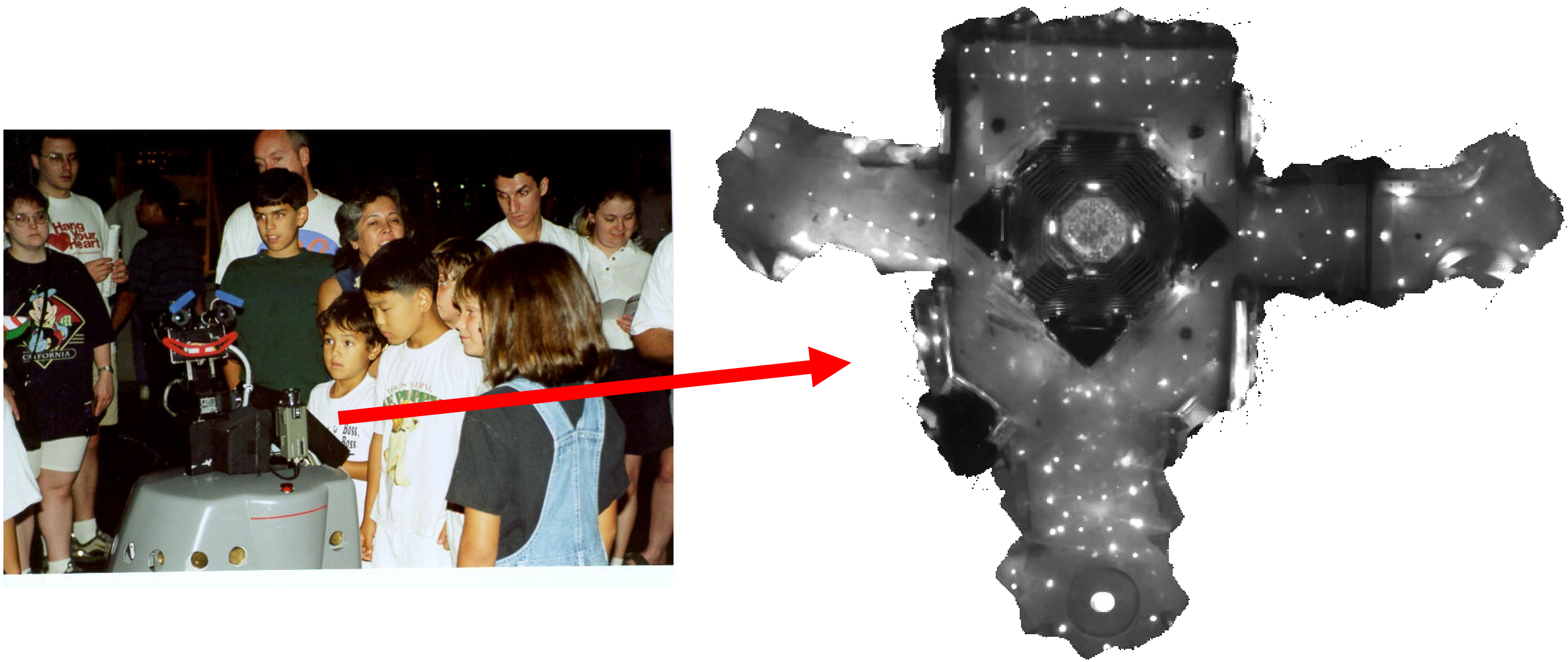
Robot Localization

- Main problems:
  - How should the robot represent the map of the world?
  - How should the robot use existing sensory data, combined with knowledge of its own movements, to figure out where it is in the map?

2D Laser Scan for Localization



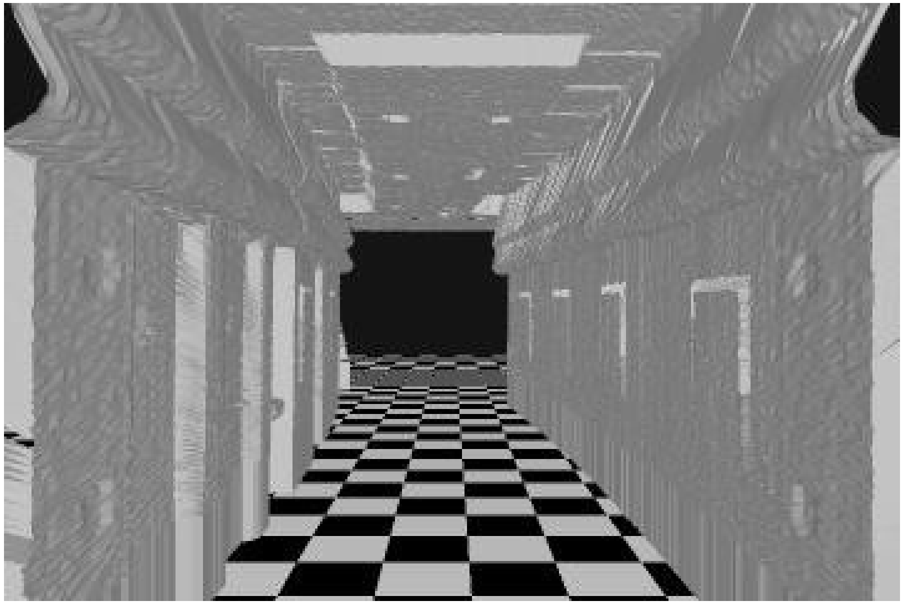
Using Ceiling Maps for Localization



3D Laser Mapping

http://www.cc.gatech.edu/aimosaic/robot-lab/research/3d/

3D mapping



[Michael Kaess, Georgia Tech]

3D mapping

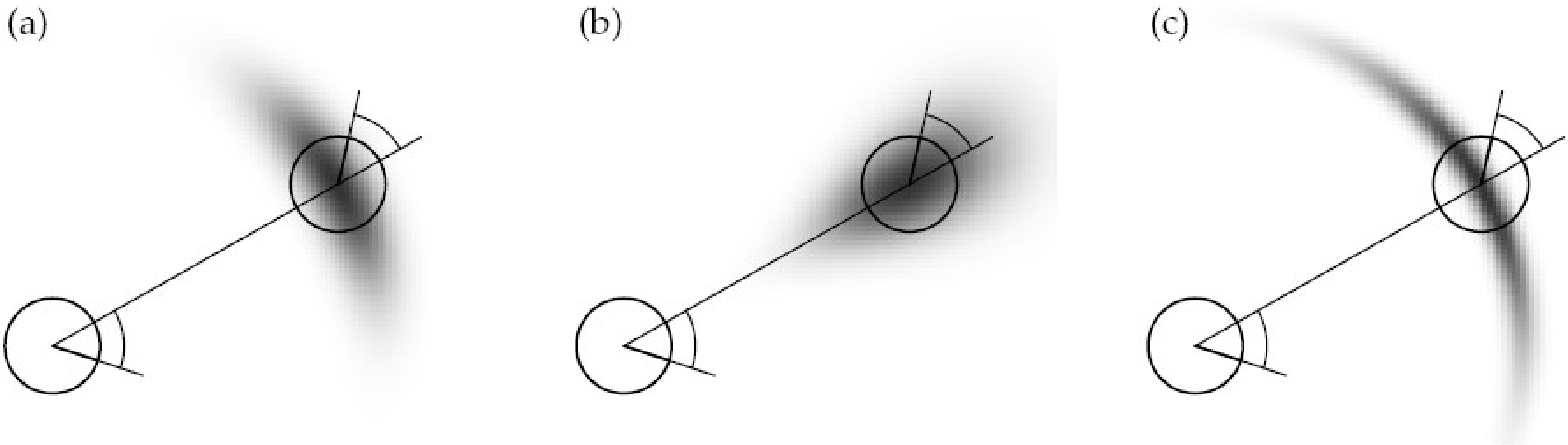
 

[Michael Kaess, Georgia Tech]

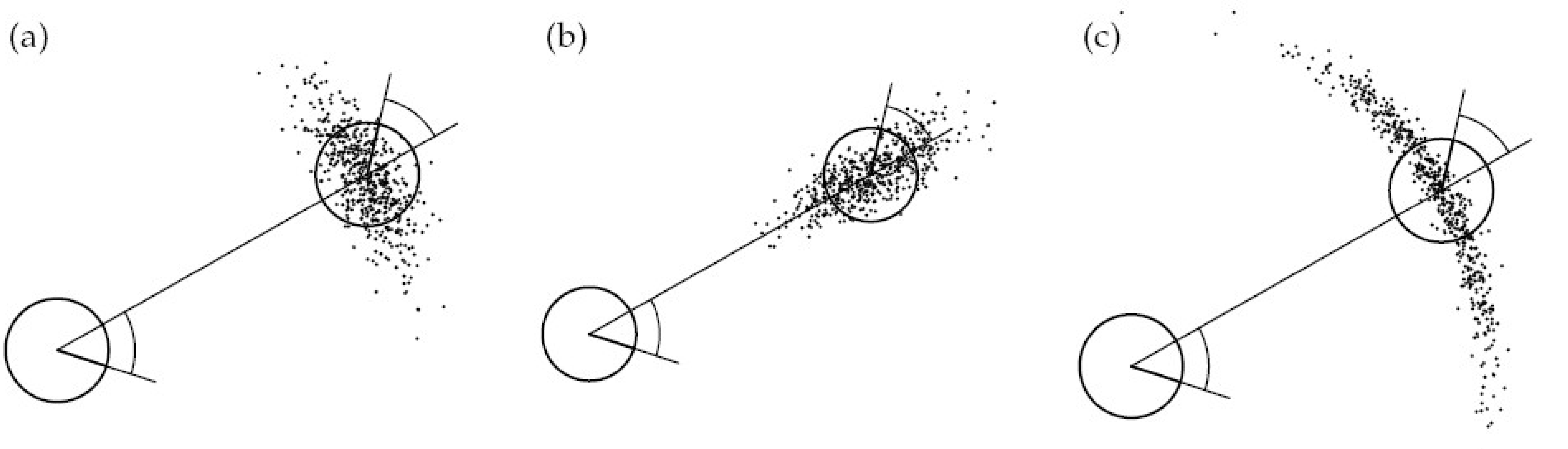
Robot Localization

Why is it not enough to simply keep track of the robot's movements relative to the starting point in the map?

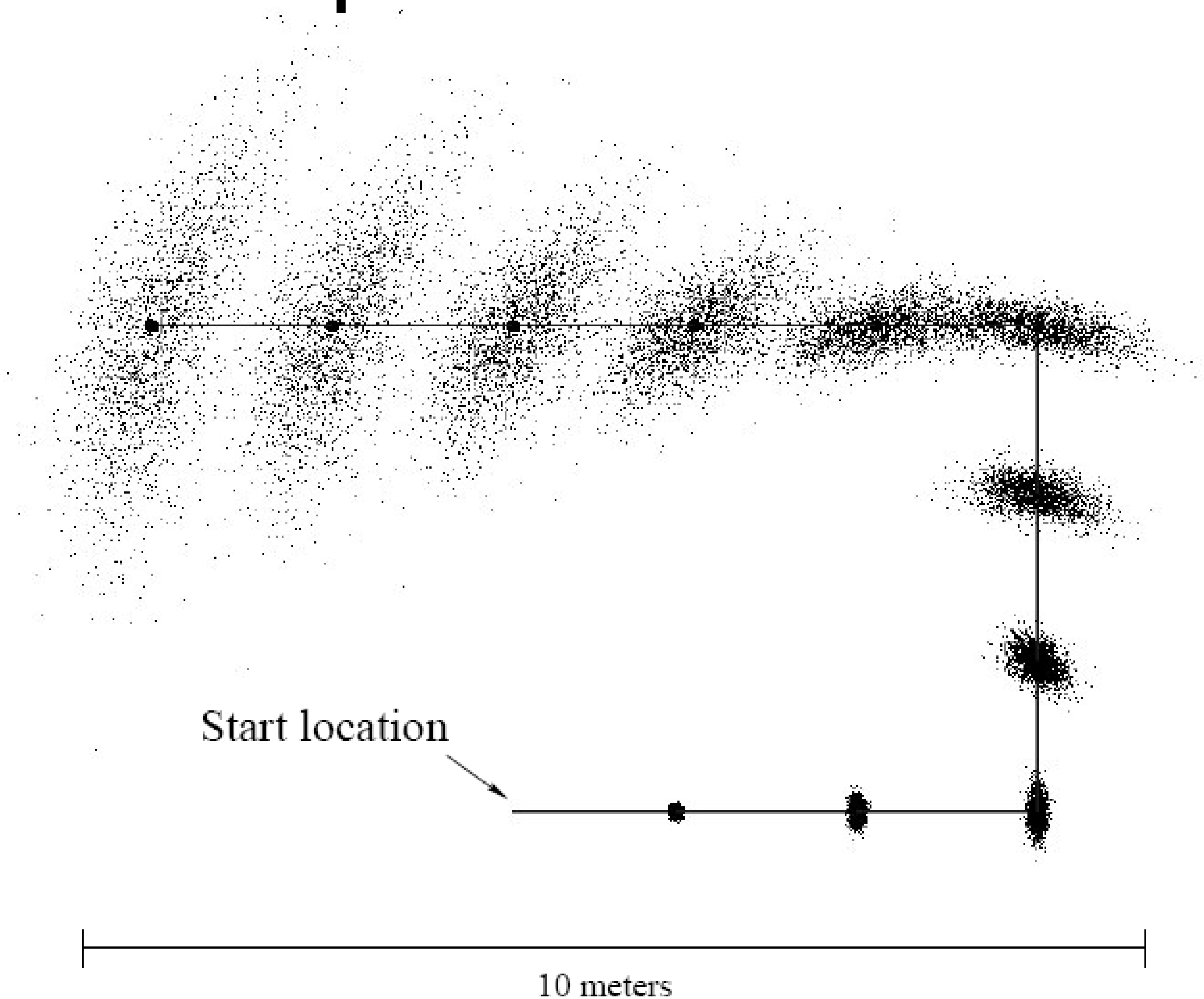
Odometry Motion Model



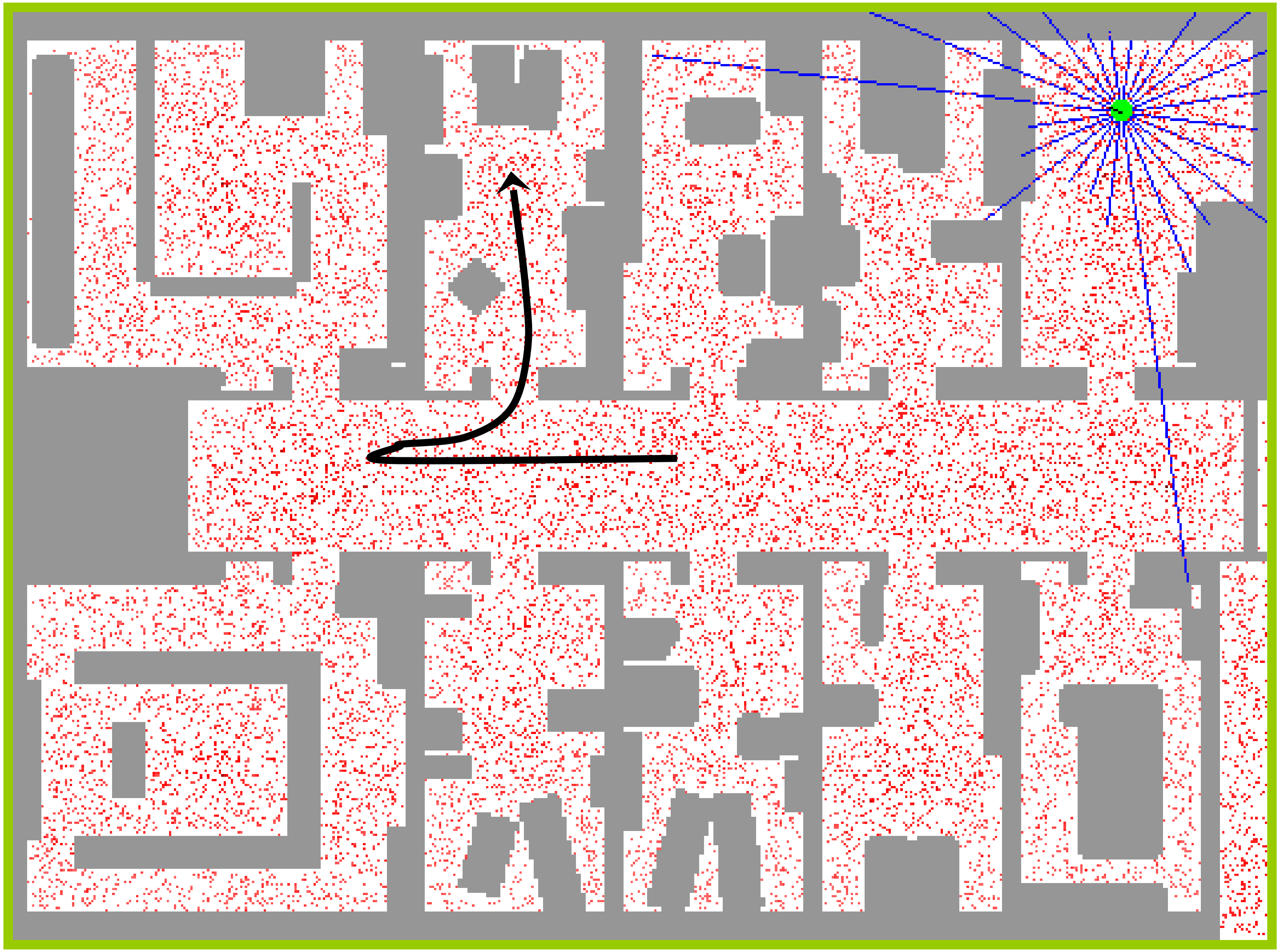
Sampling From the Odometry Model

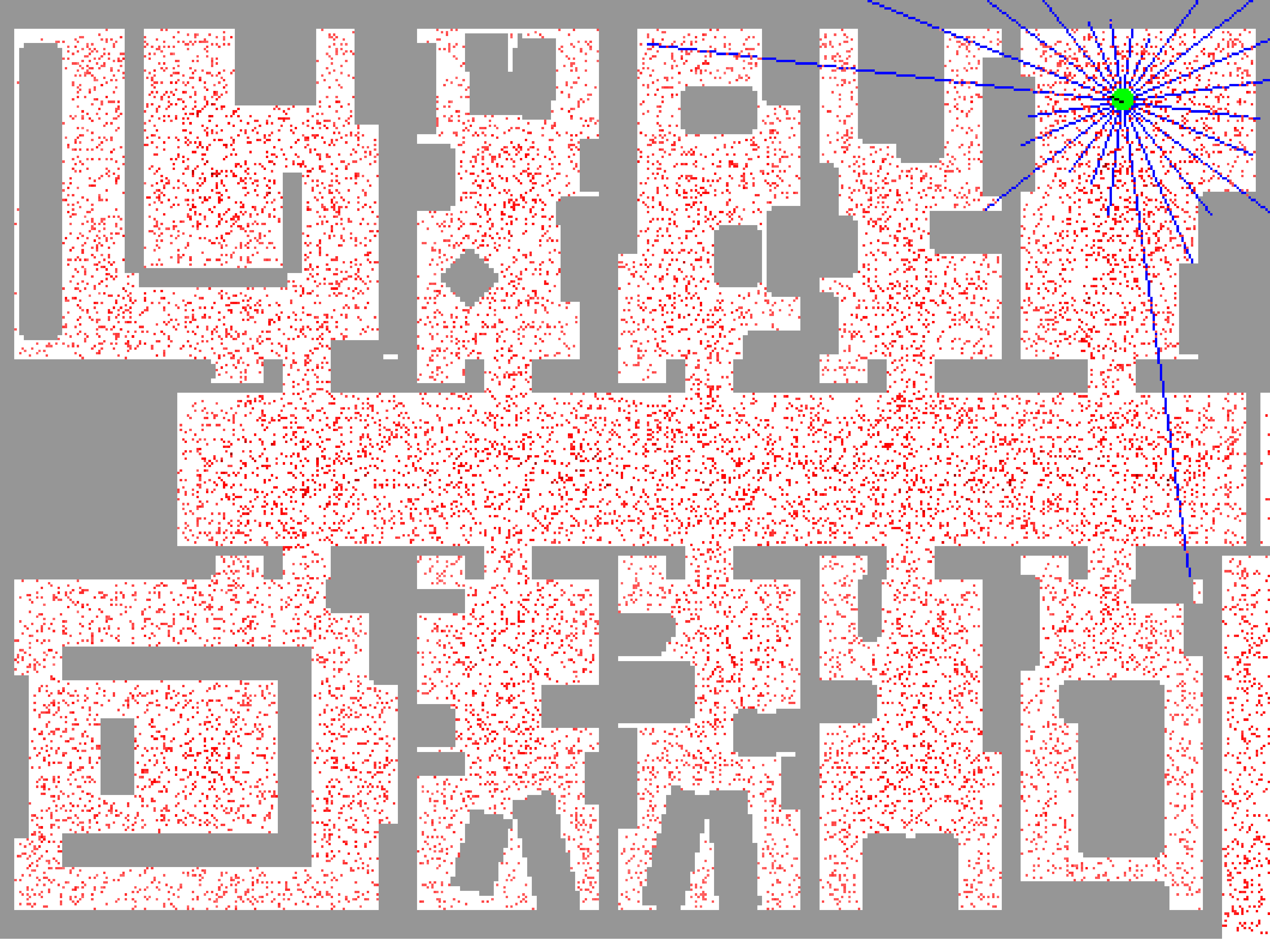


Uncertainty accumulates after multiple movements



Localization using Sonar





Example

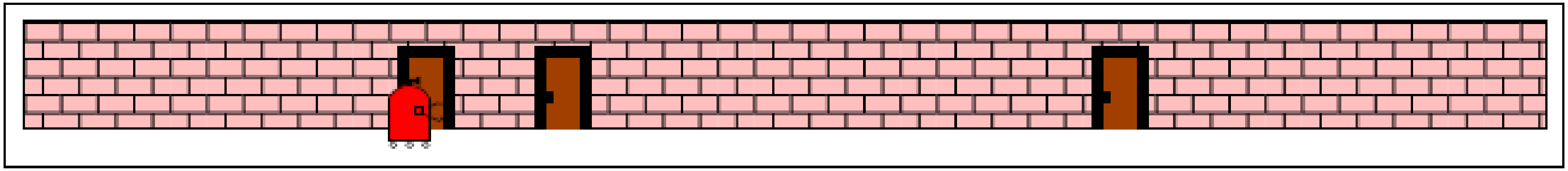
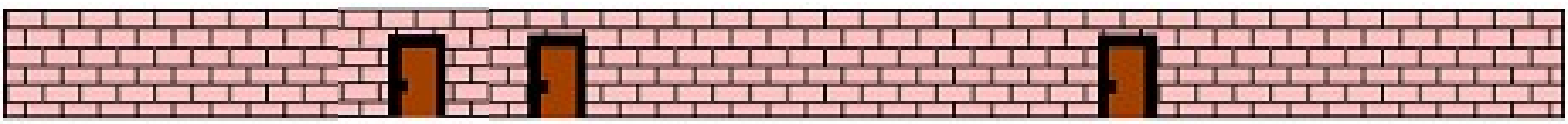
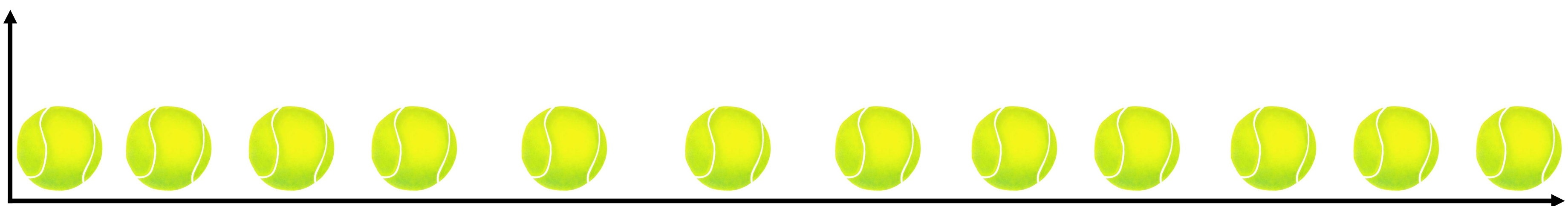


Figure 7.4 Example environment used to illustrate mobile robot localization: One-dimensional hallway environment with three indistinguishable doors. Initially the robot does not know its location except for its heading direction. Its goal is to find out where it is.

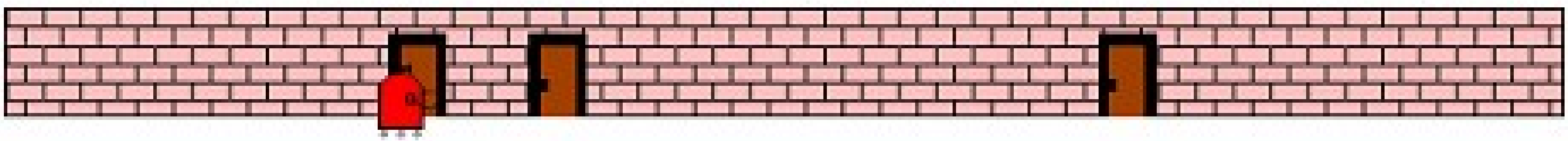
[Thrun, Burgard & Fox (2005)]

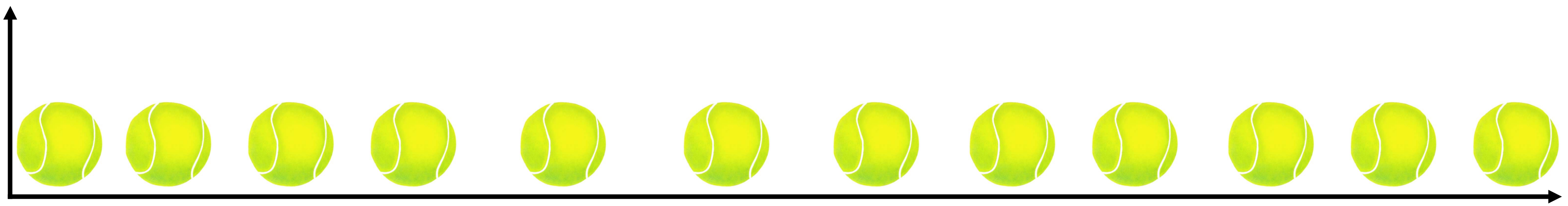
Initially we don't know the location of the robot so we have particles everywhere



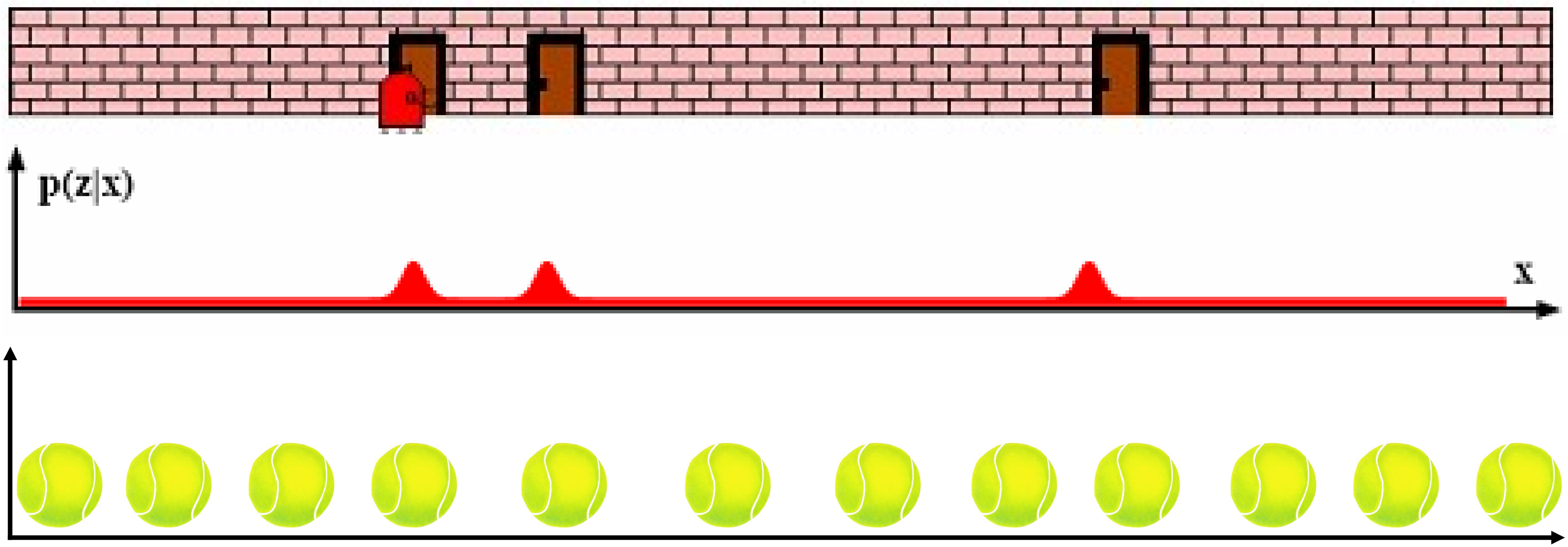


Next, the robot senses that it is near a door

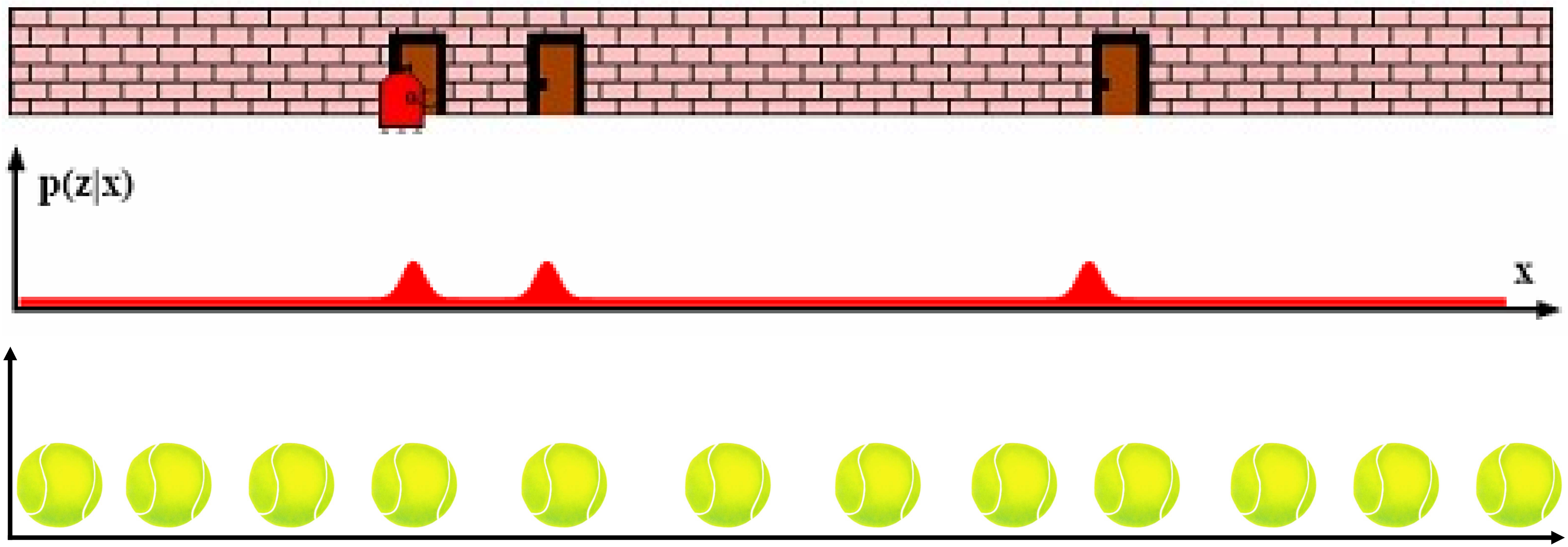




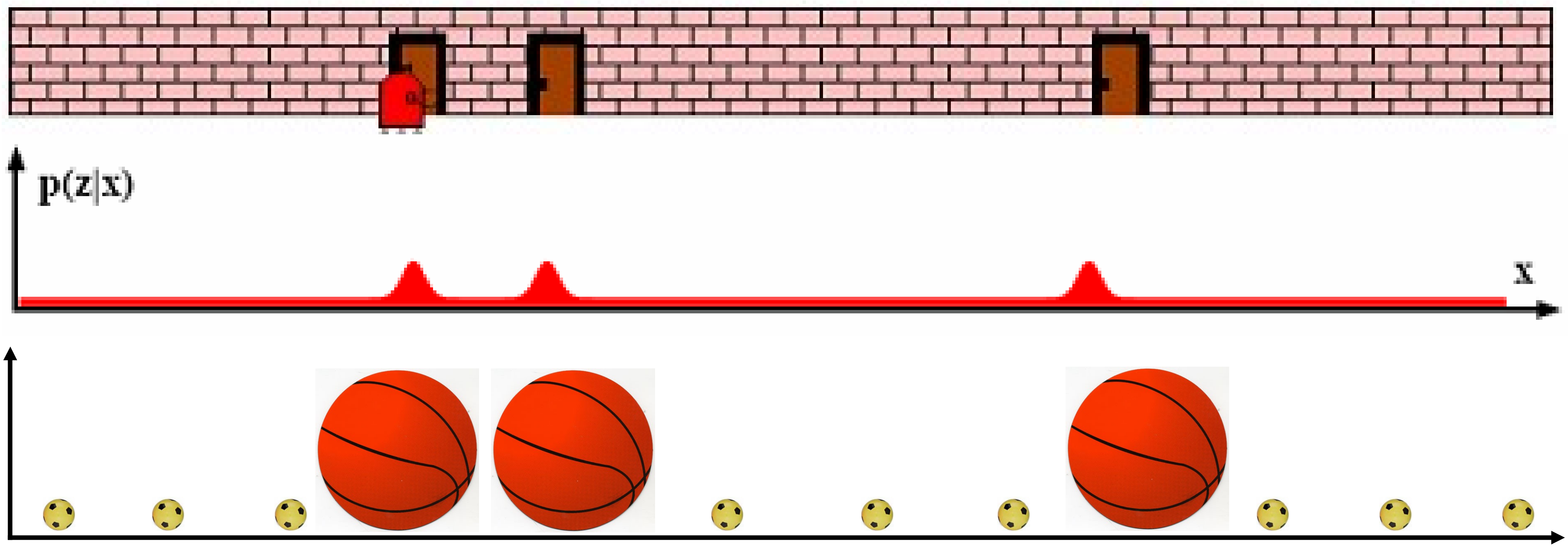
Since there are 3 identical doors the robot can be next to any one of them



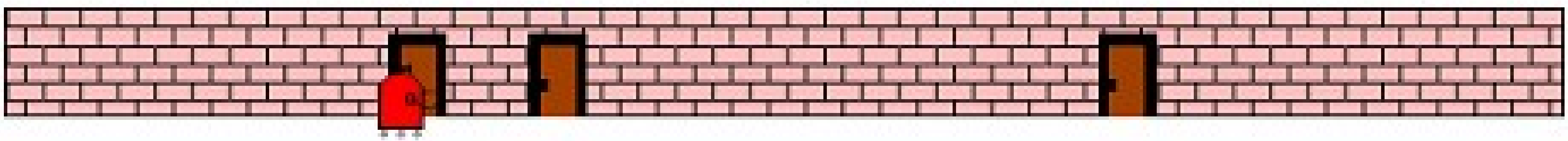
Therefore, we inflate balls (particles) that are next to doors and shrink all others

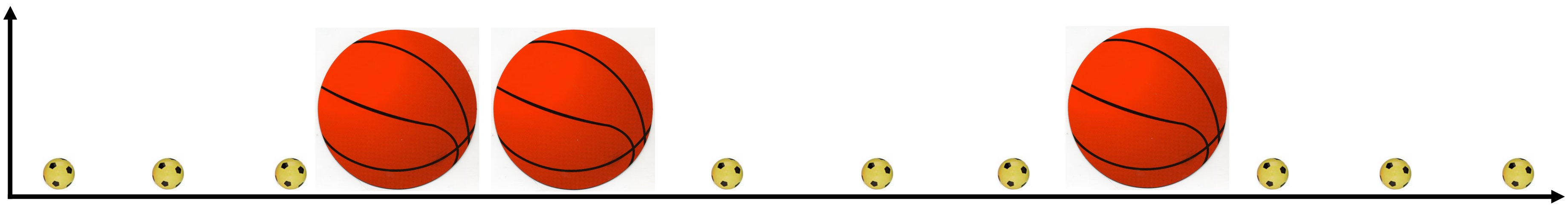


Therefore, we grow balls (particles) that are next to doors and shrink all others

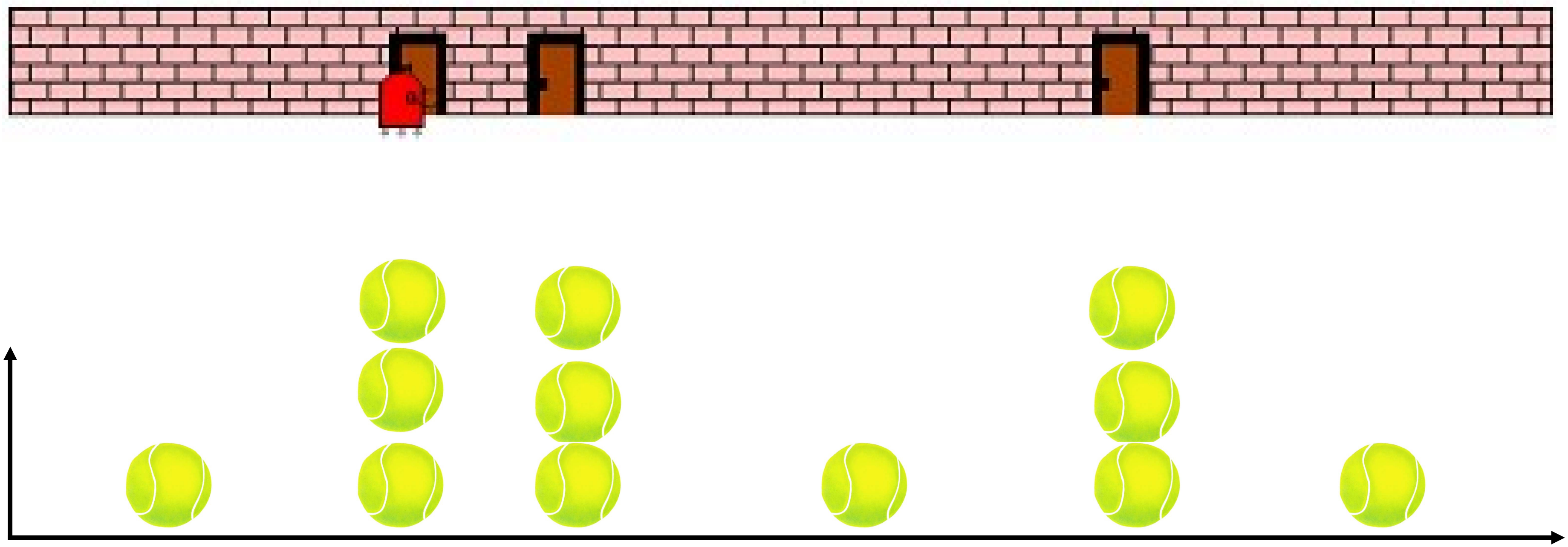


Before we continue we have to make all balls to be of equal size. We need to resample.

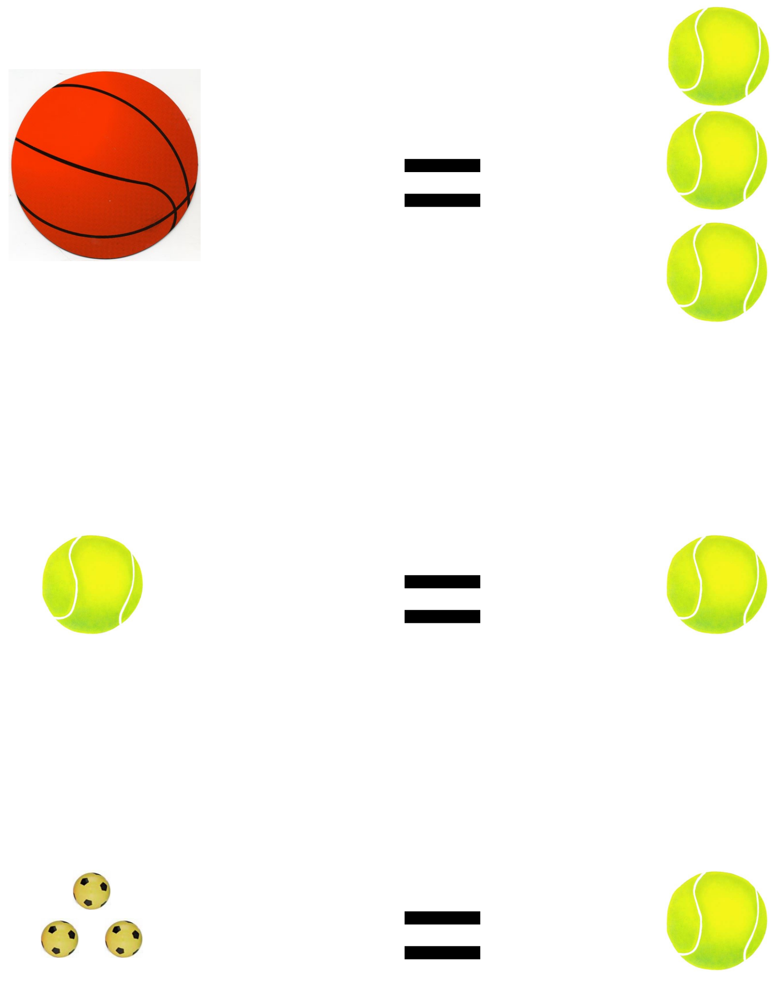




Before we continue we have to make all balls to be of equal size. We need to resample.



Resampling Rules

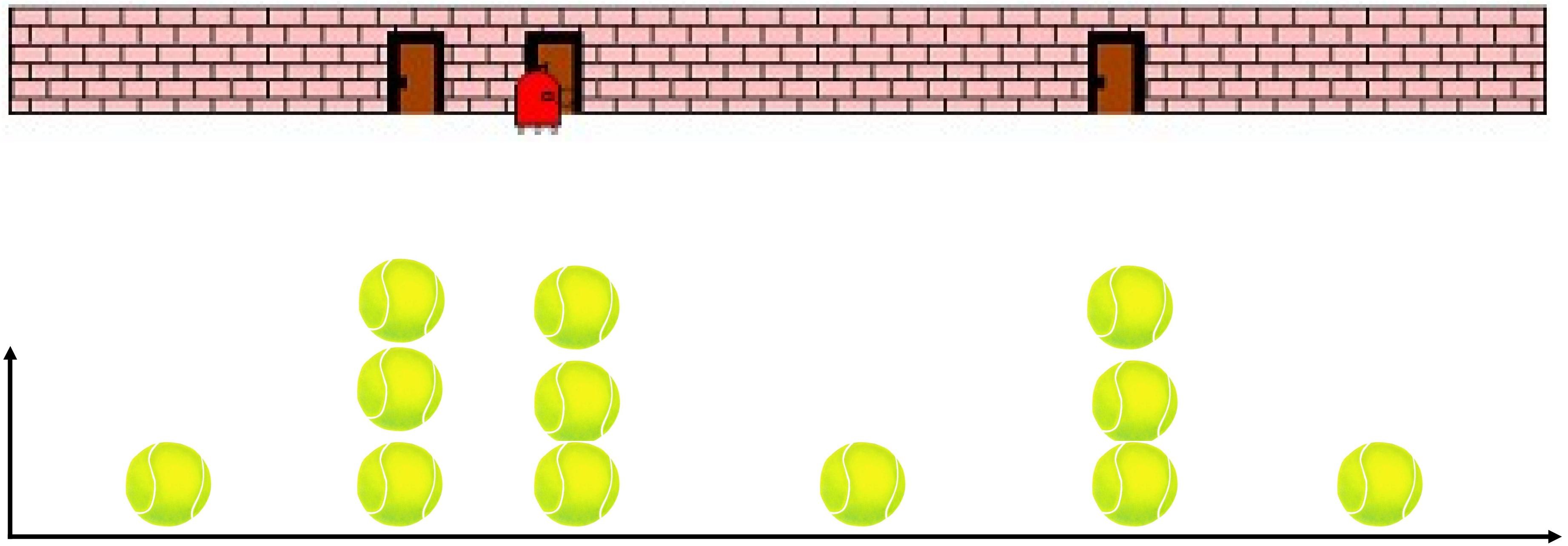


Resampling

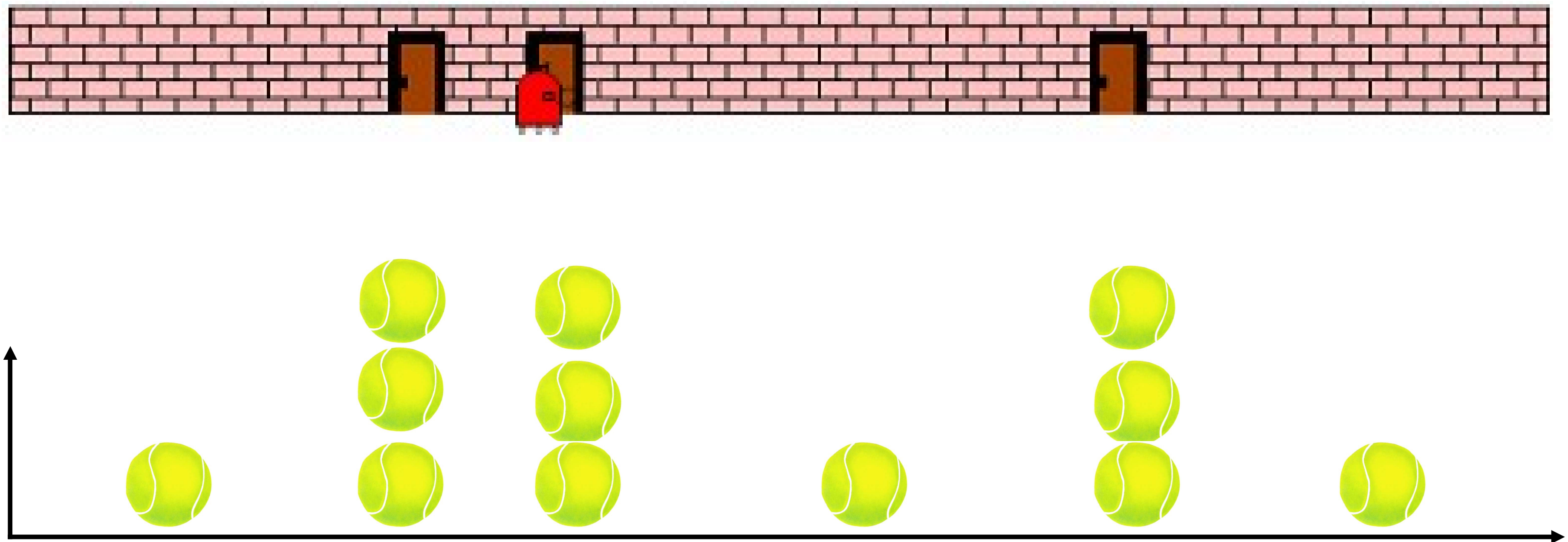
- **Given**: Set S of weighted samples.
- **Wanted** : Random sample, where the probability of drawing x_i is given by w_i.
- Typically done n times with replacement to generate new sample set S'.

[From Thrun's book "Probabilistic Robotics"]

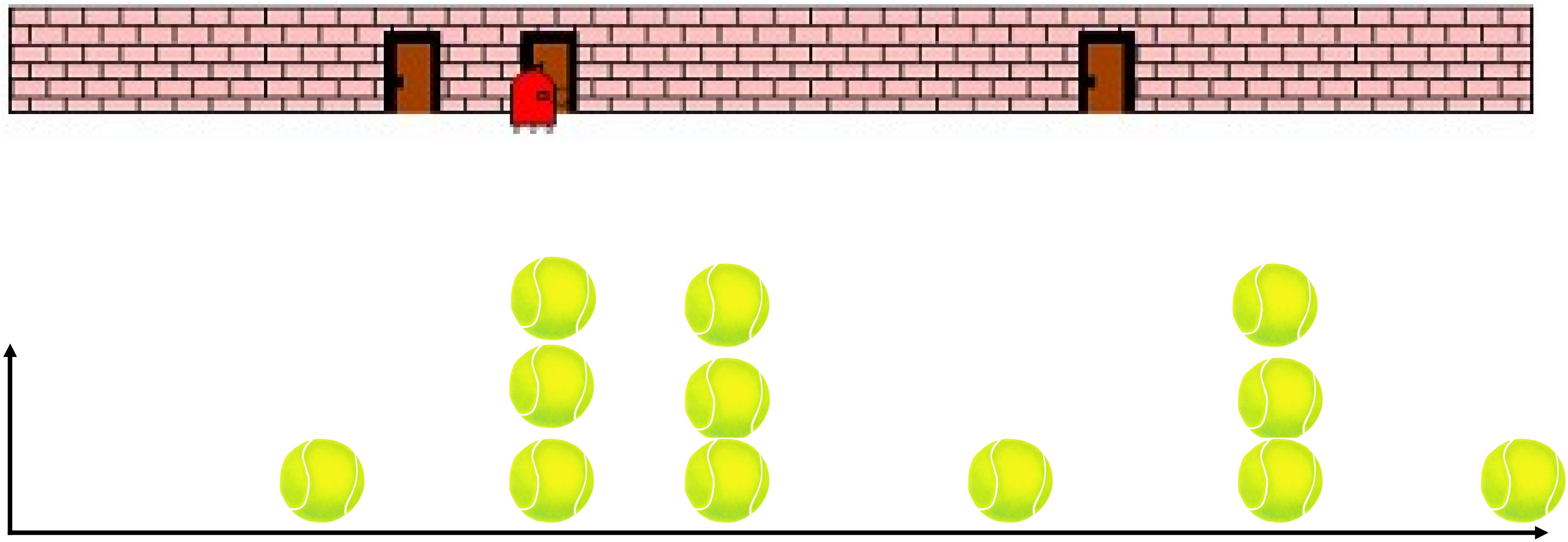
Next, The robot moves to the right



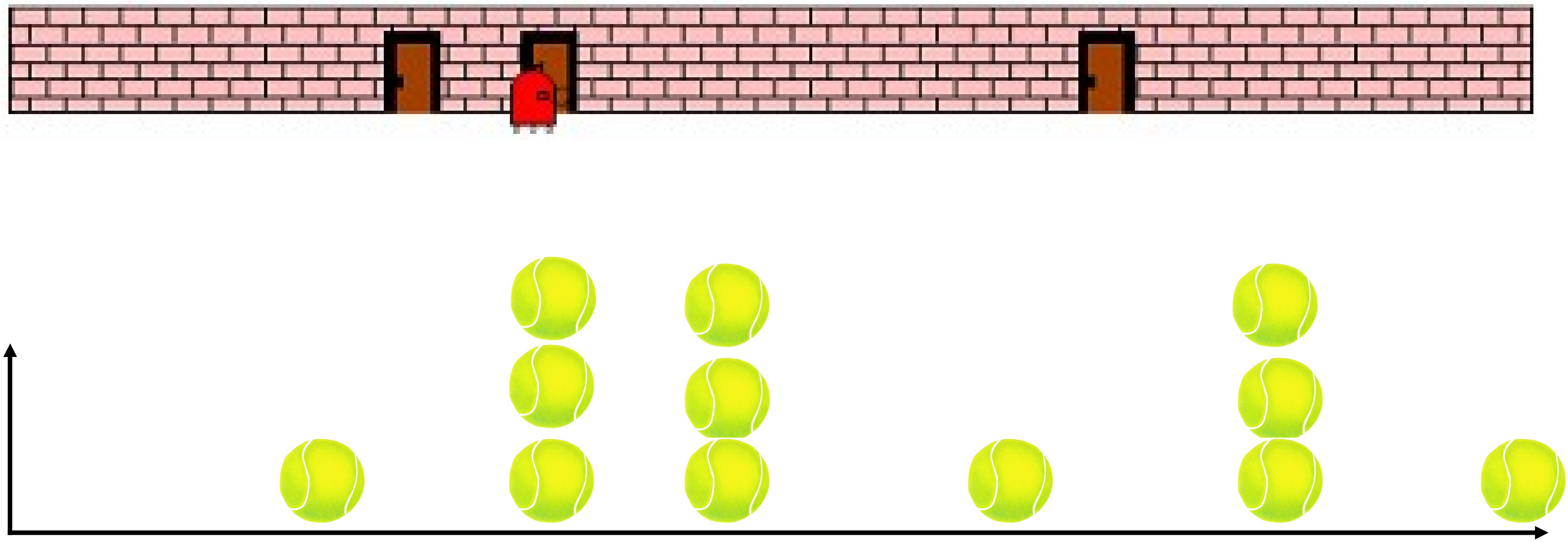
… thus, we have to shift all balls (particles) to the right



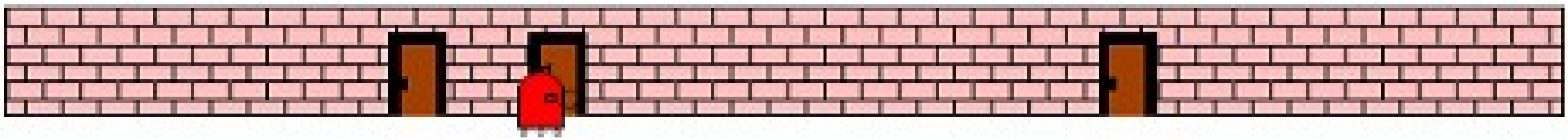
… thus, we have to shift all balls (particles) to the right

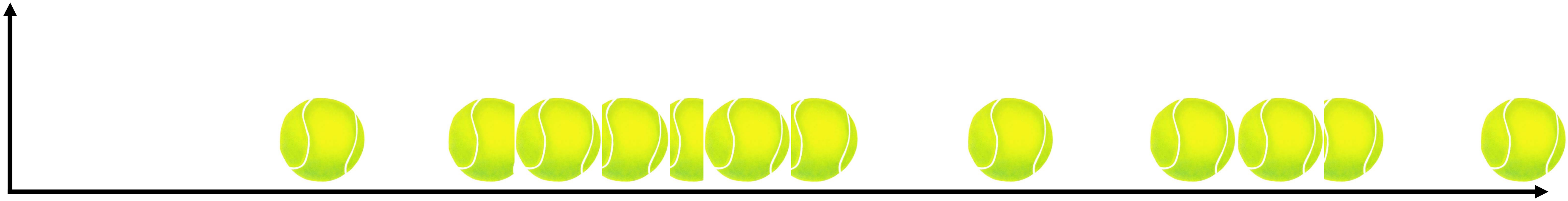


… and add some position noise

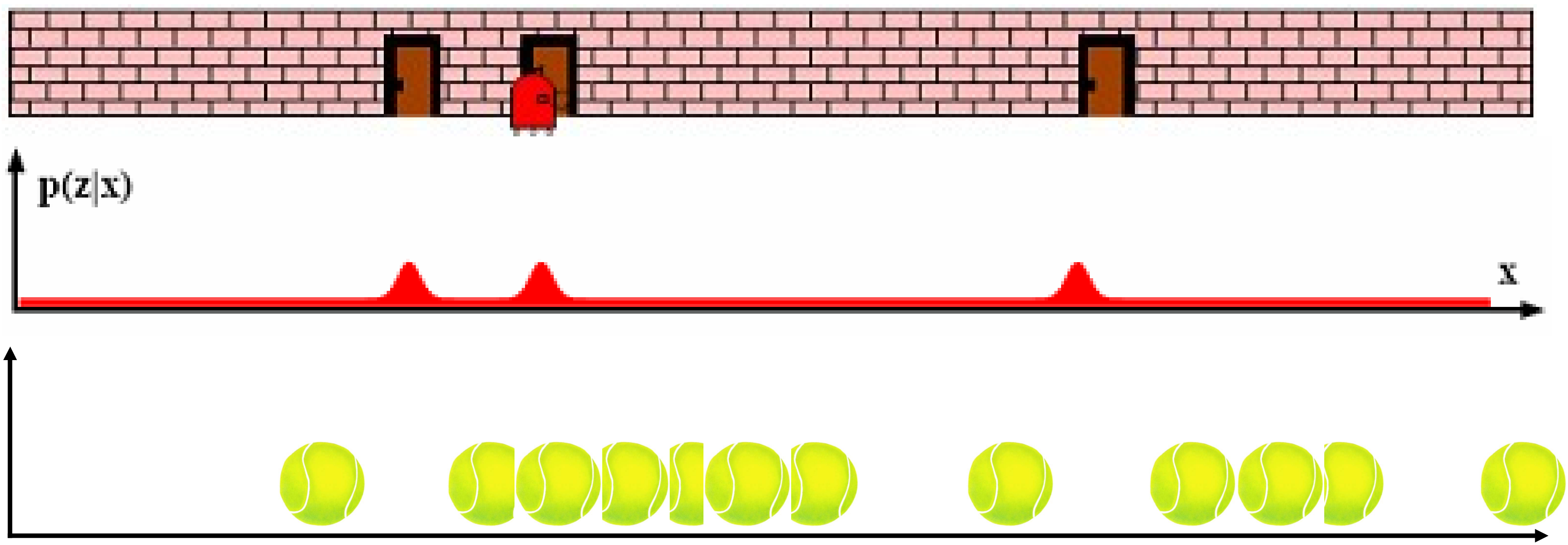


… and add some position noise

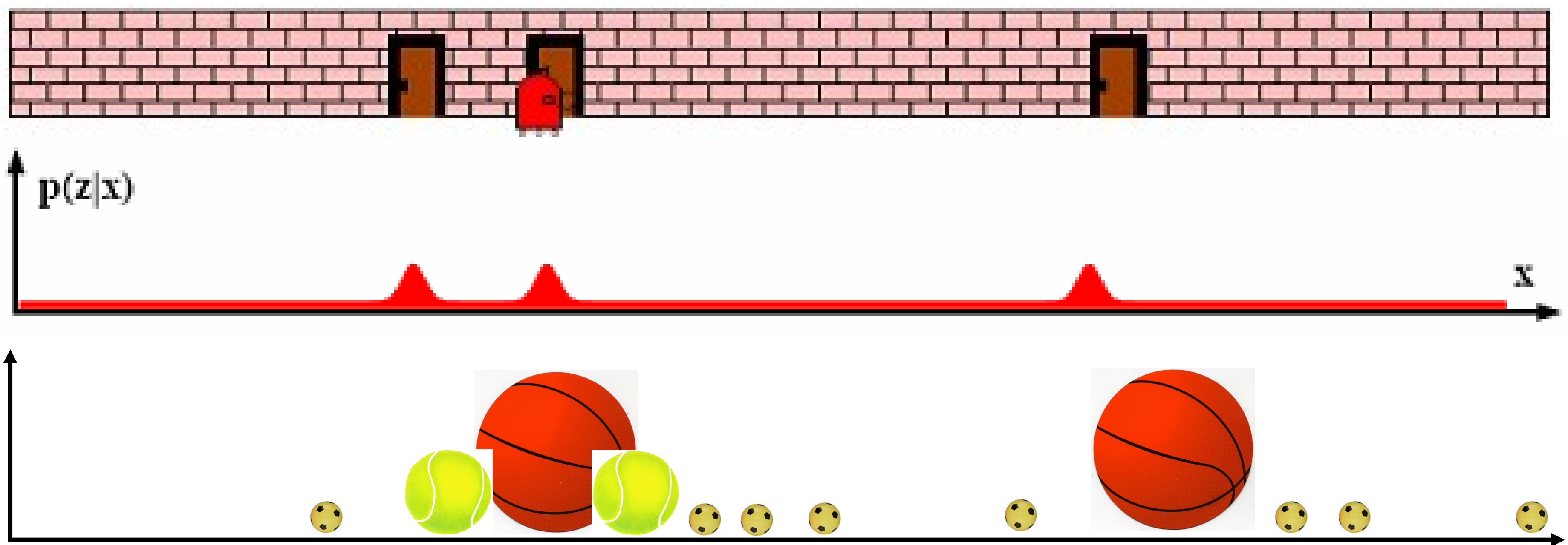




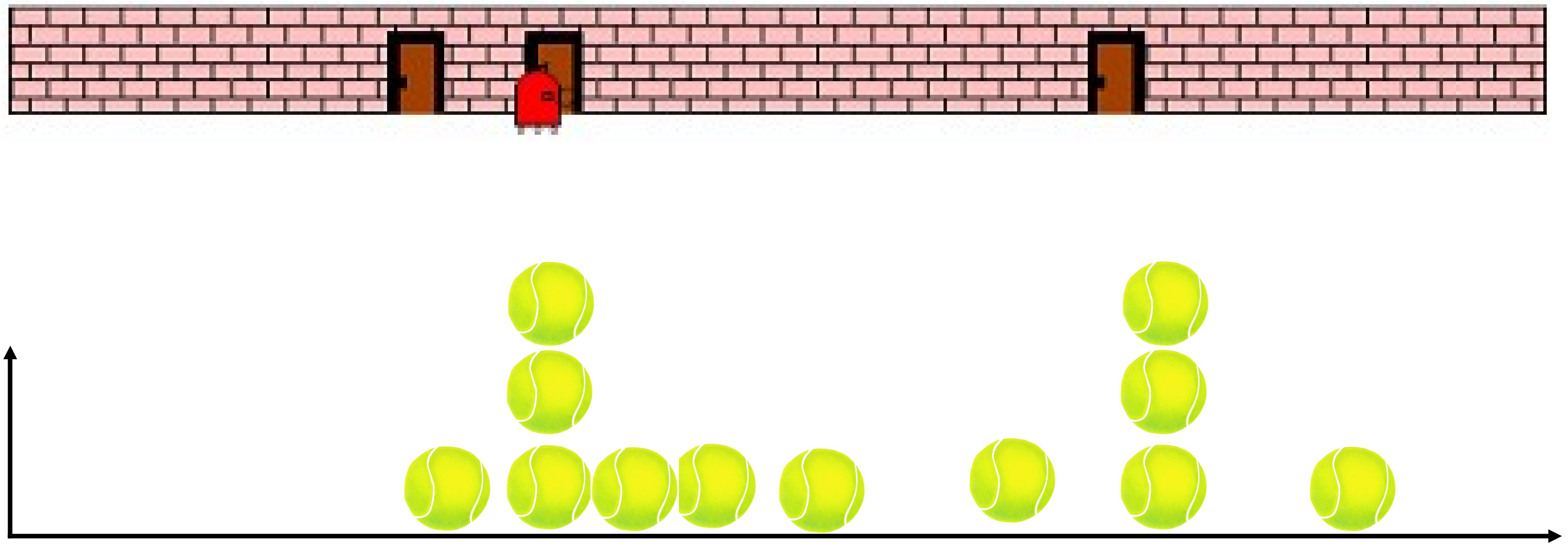
Next, the robot senses that it is next to one of the three doors



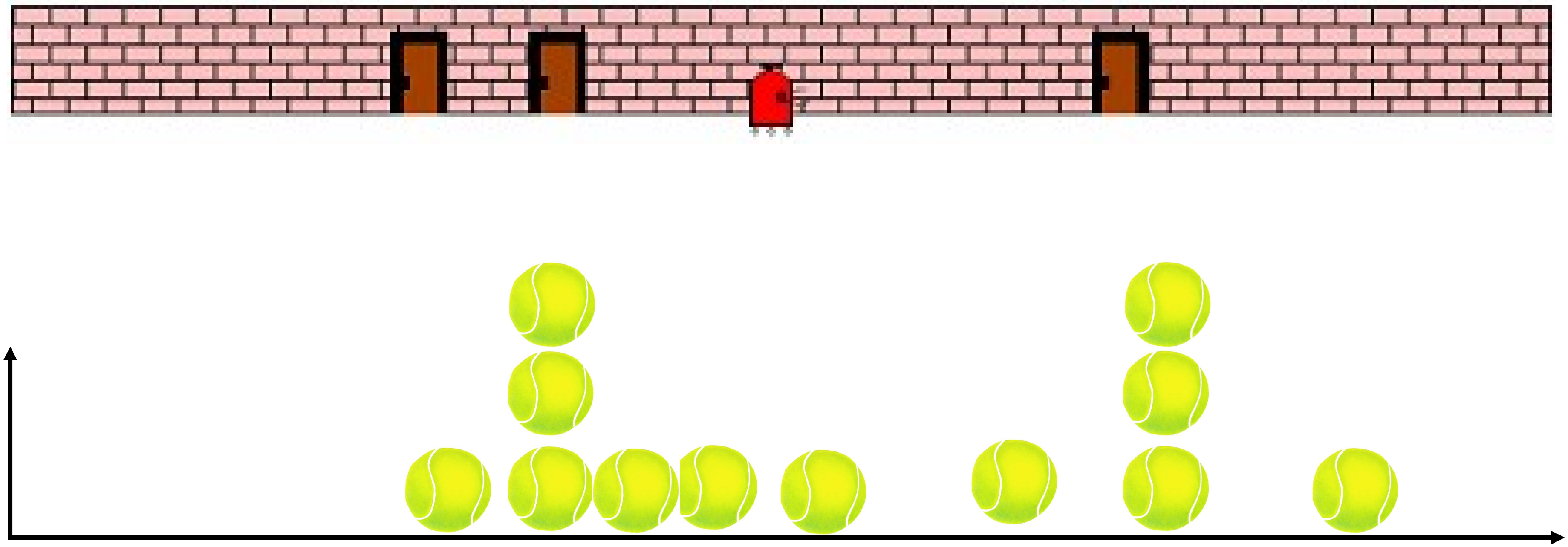
Next, the robot senses that it is next to one of the three doors



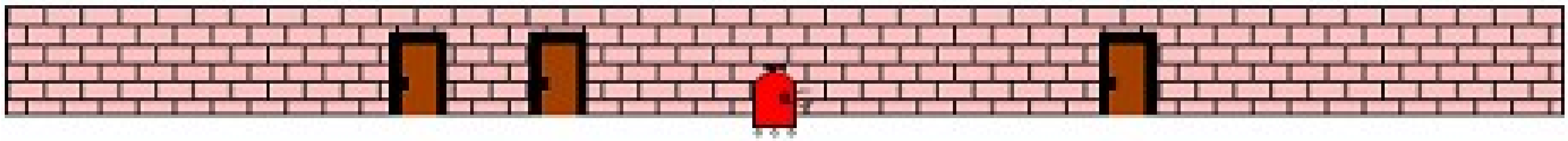
Now we have to resample again

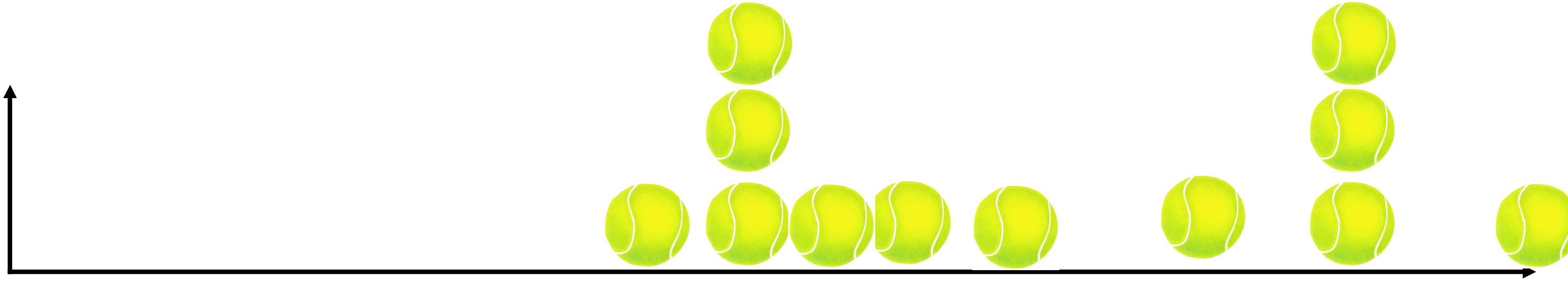


The robot moves again

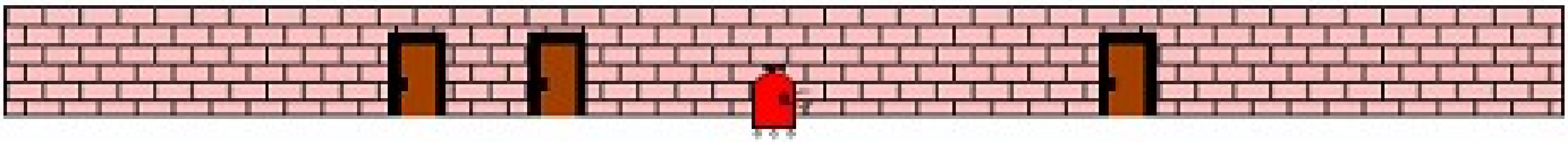


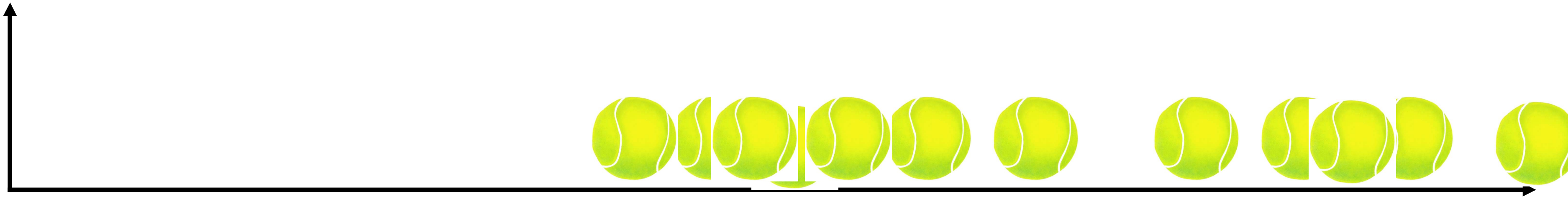
… so we must move all balls (particles) to the right again





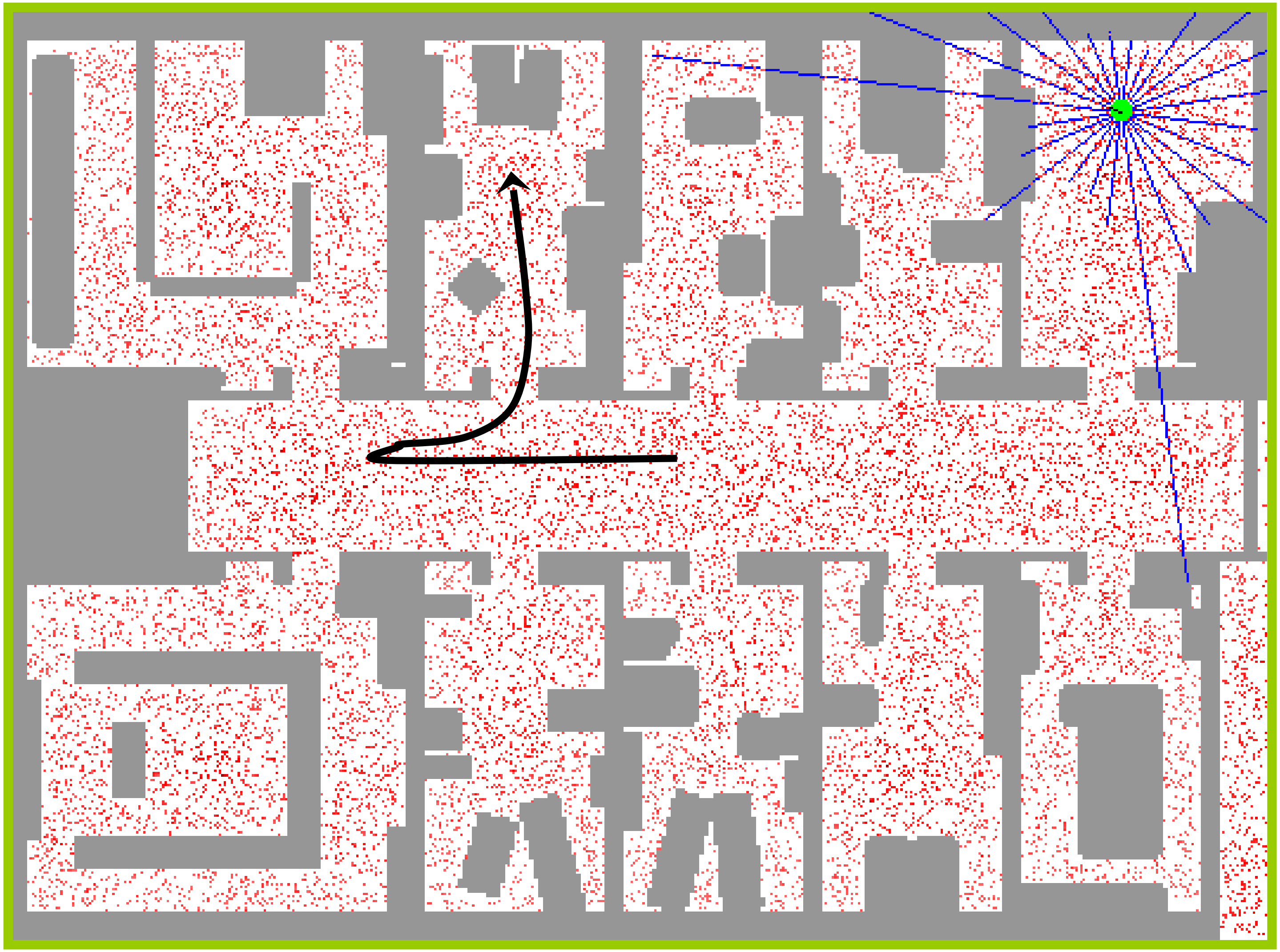
… and add some position noise

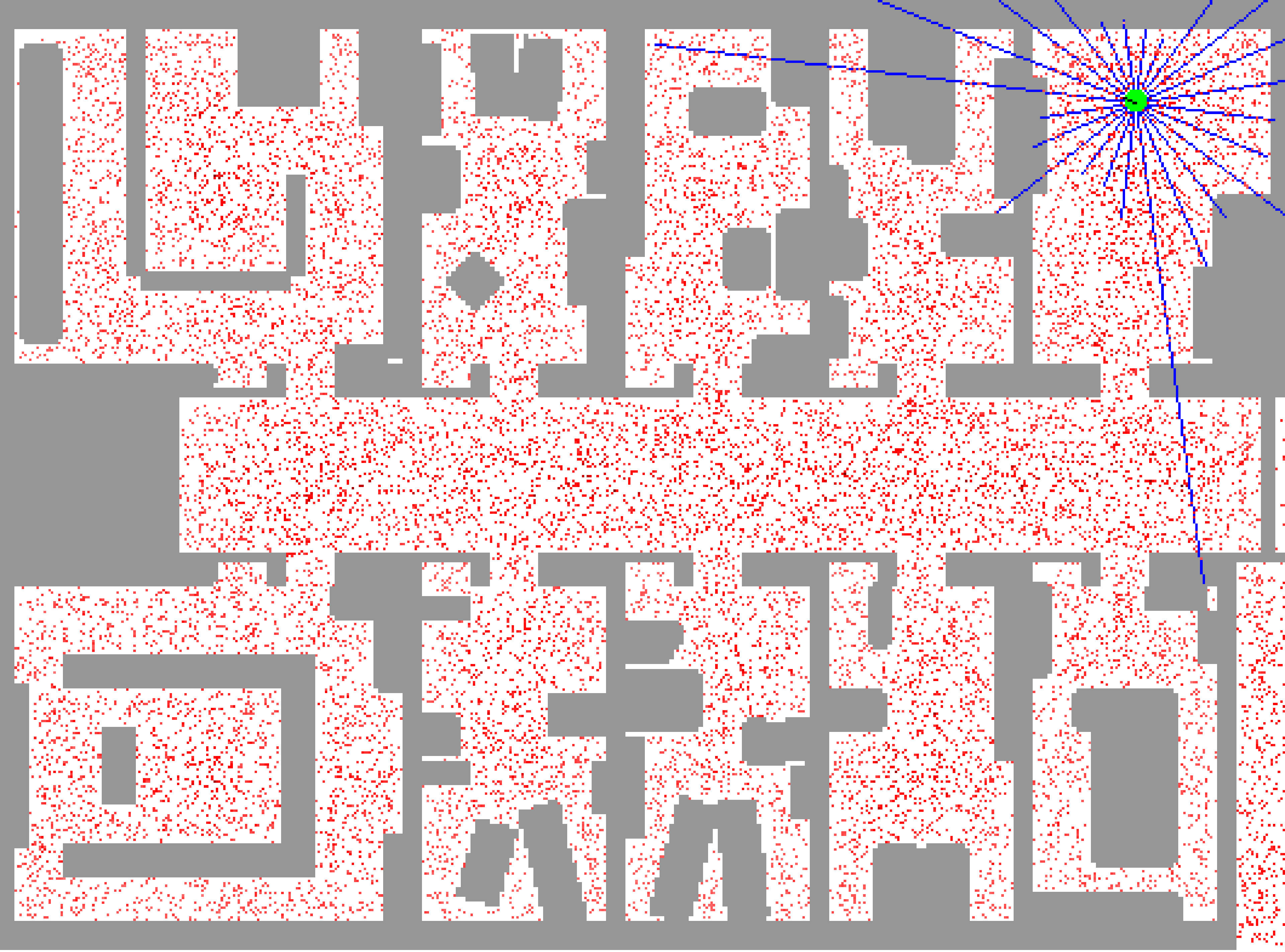




And so on …

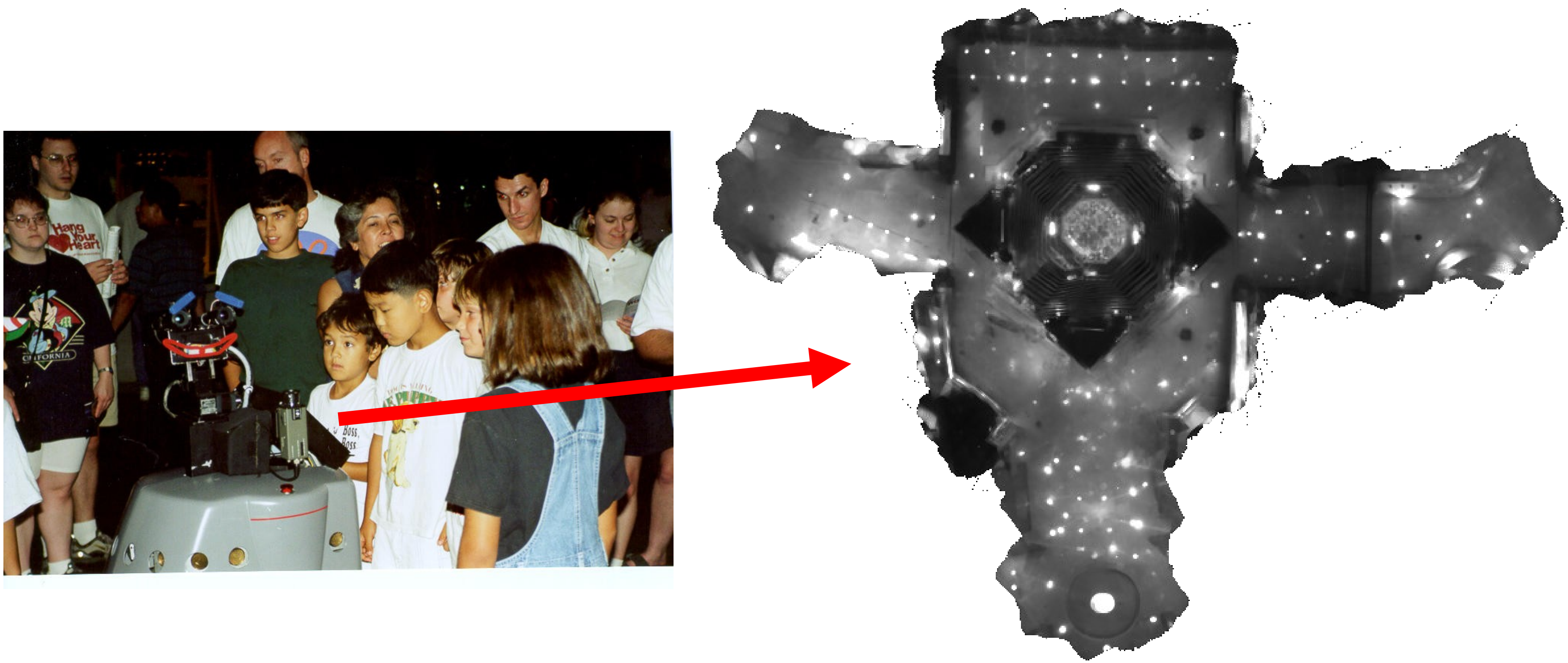
Localization using Sonar



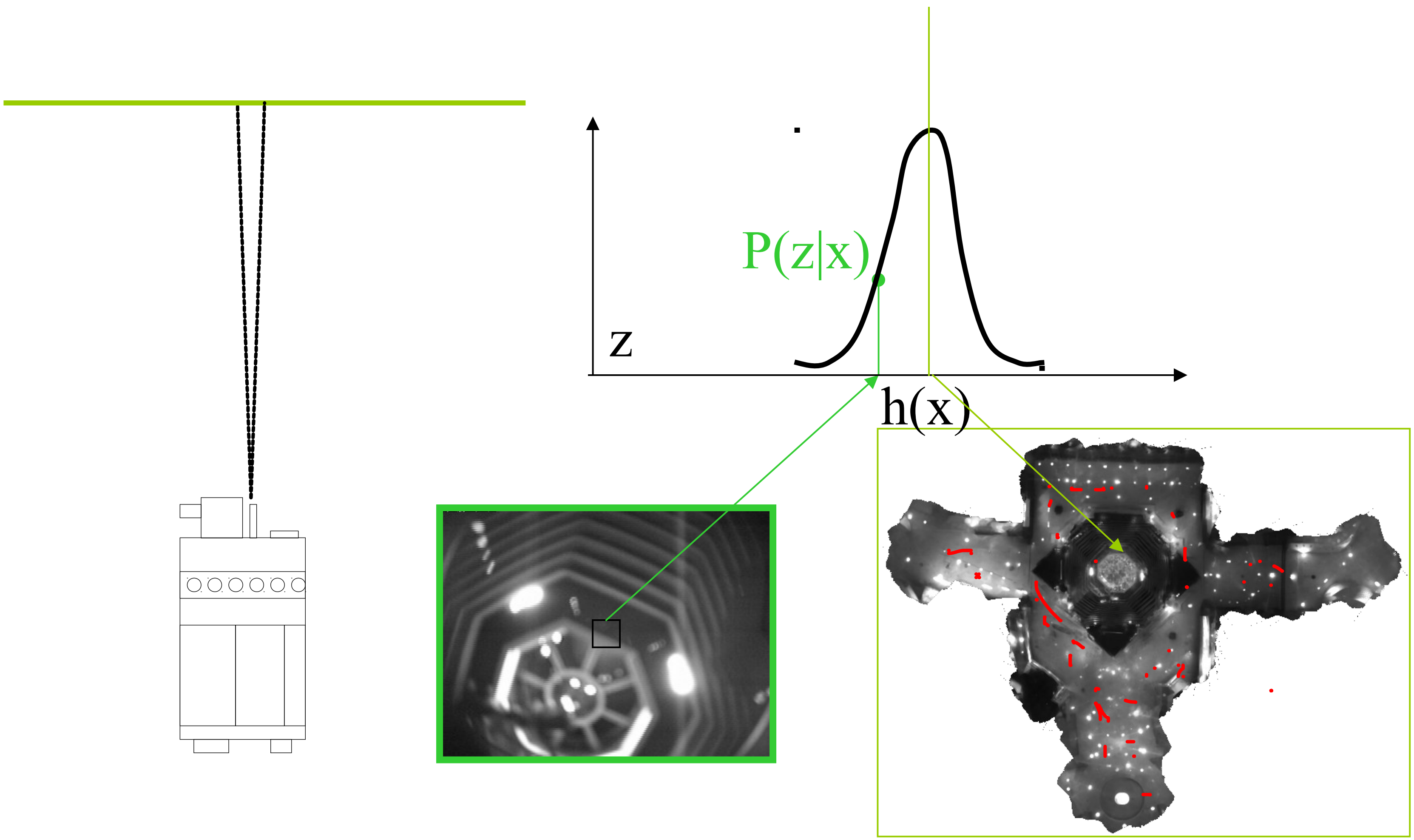


What does this look like on a real robot?

Using Ceiling Maps for Localization

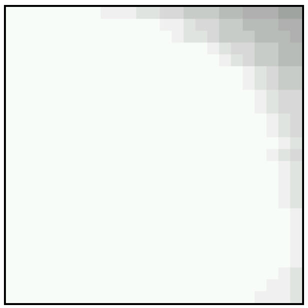


Vision-based Localization

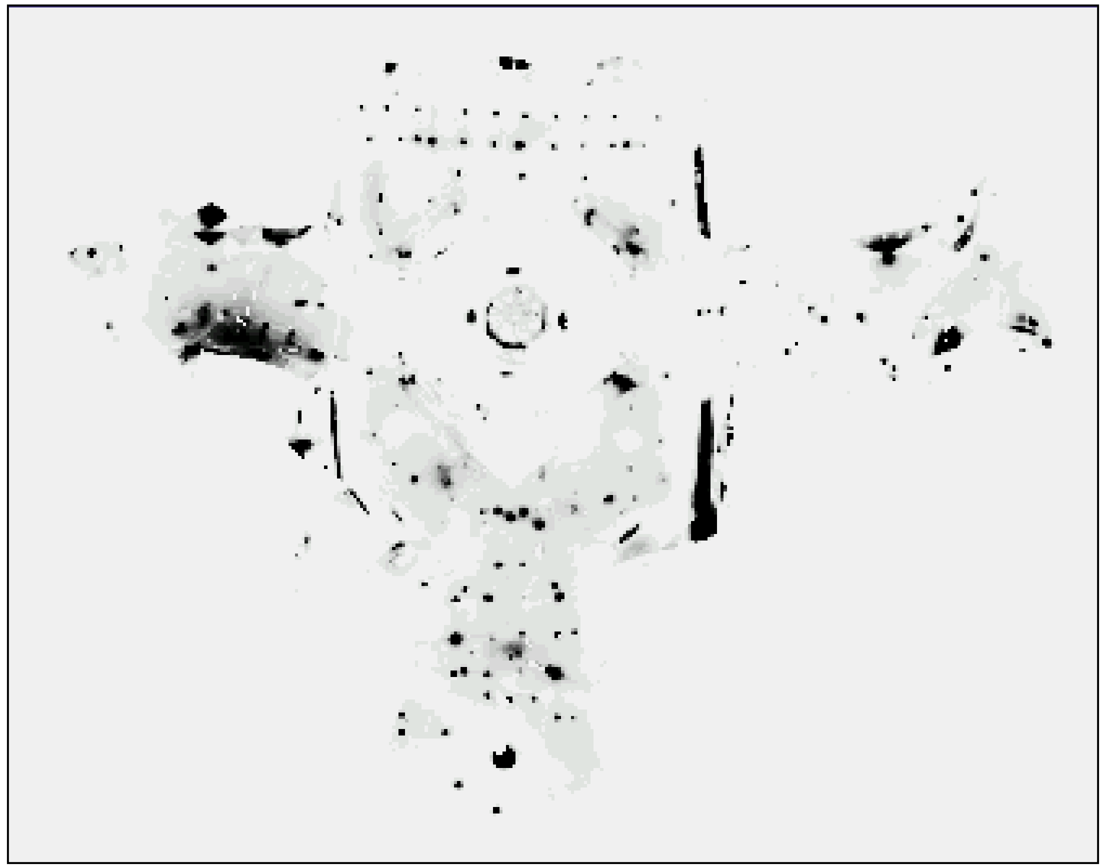


Under a Light

Measurement z:

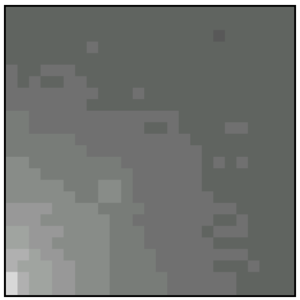


P(z|x):

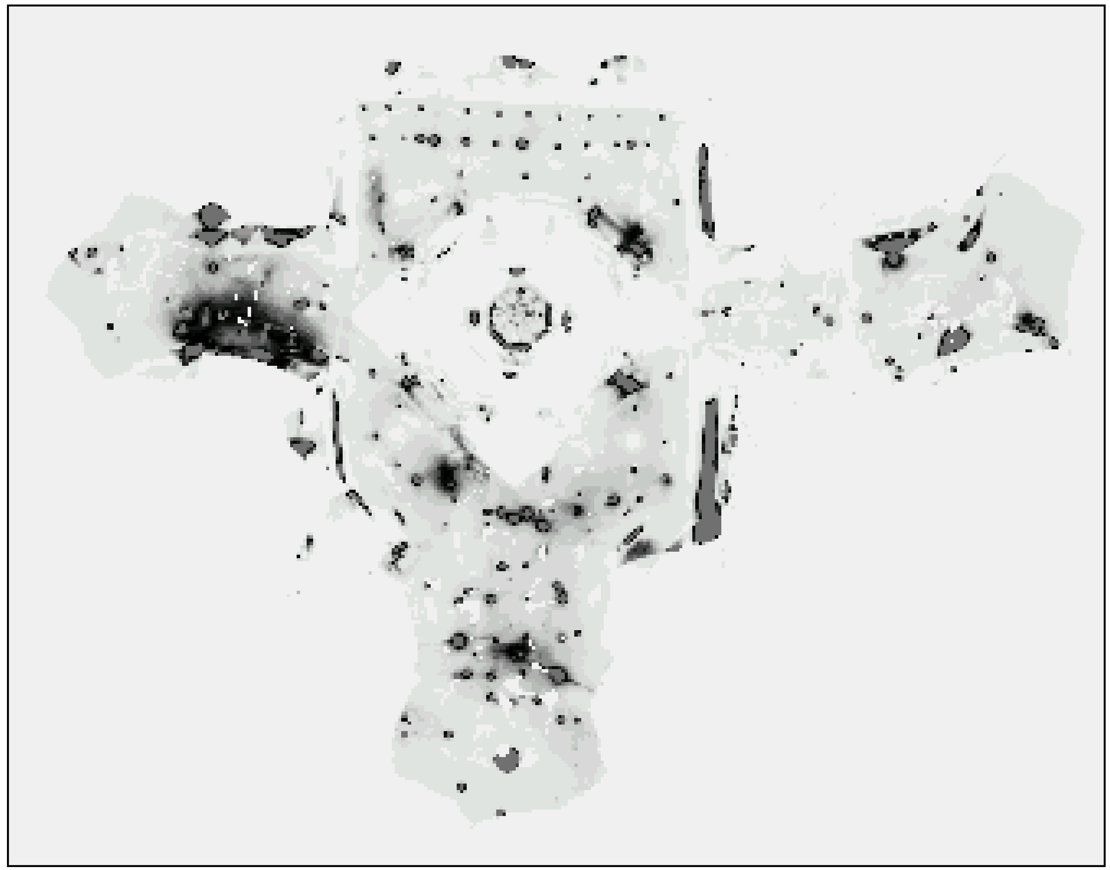


Next to a Light

Measurement z:



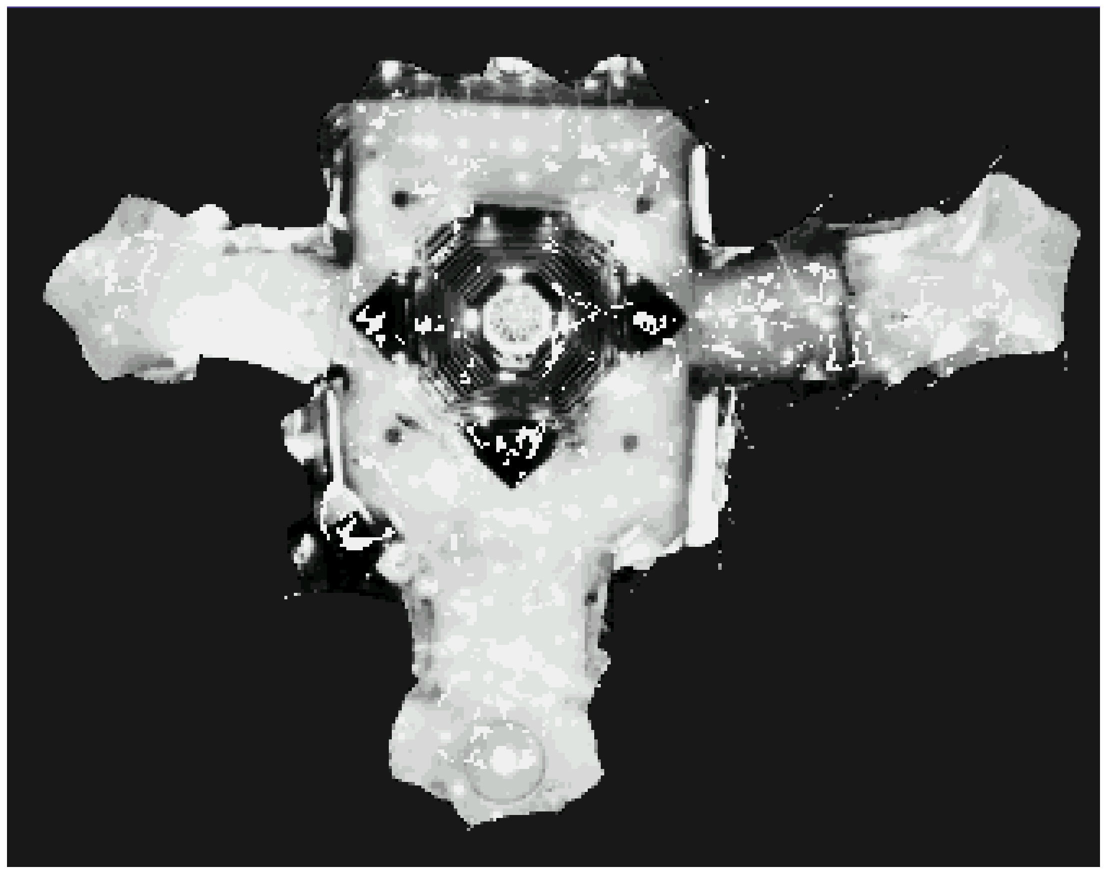
P(z|x):



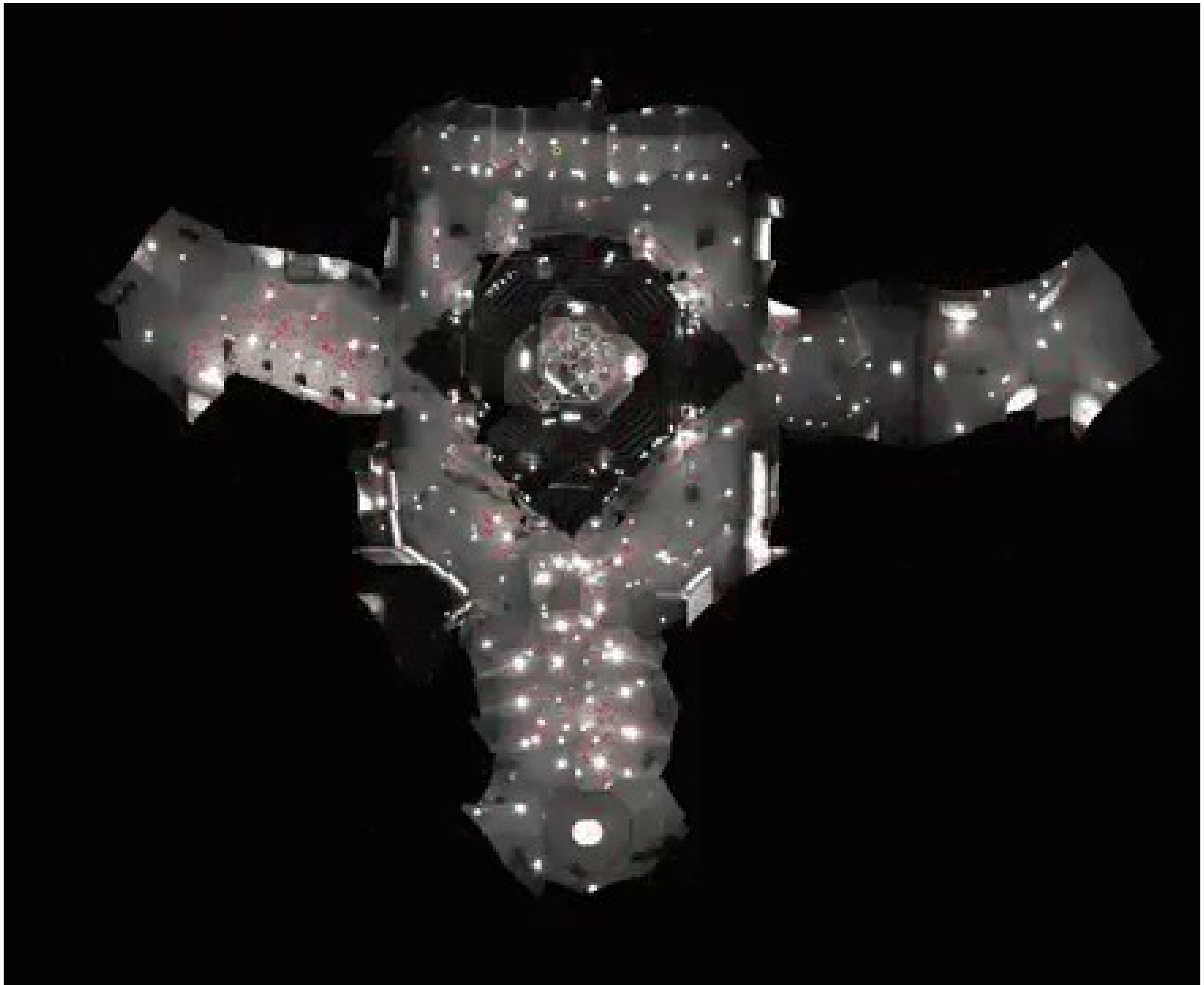
Elsewhere

Measurement z:

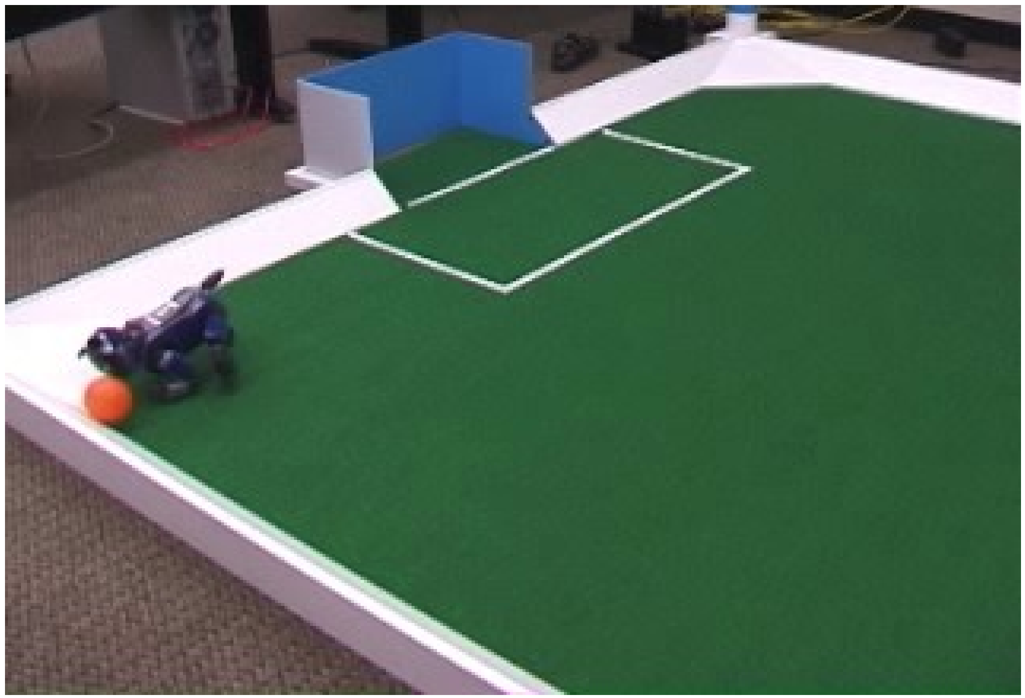
P(z|x):



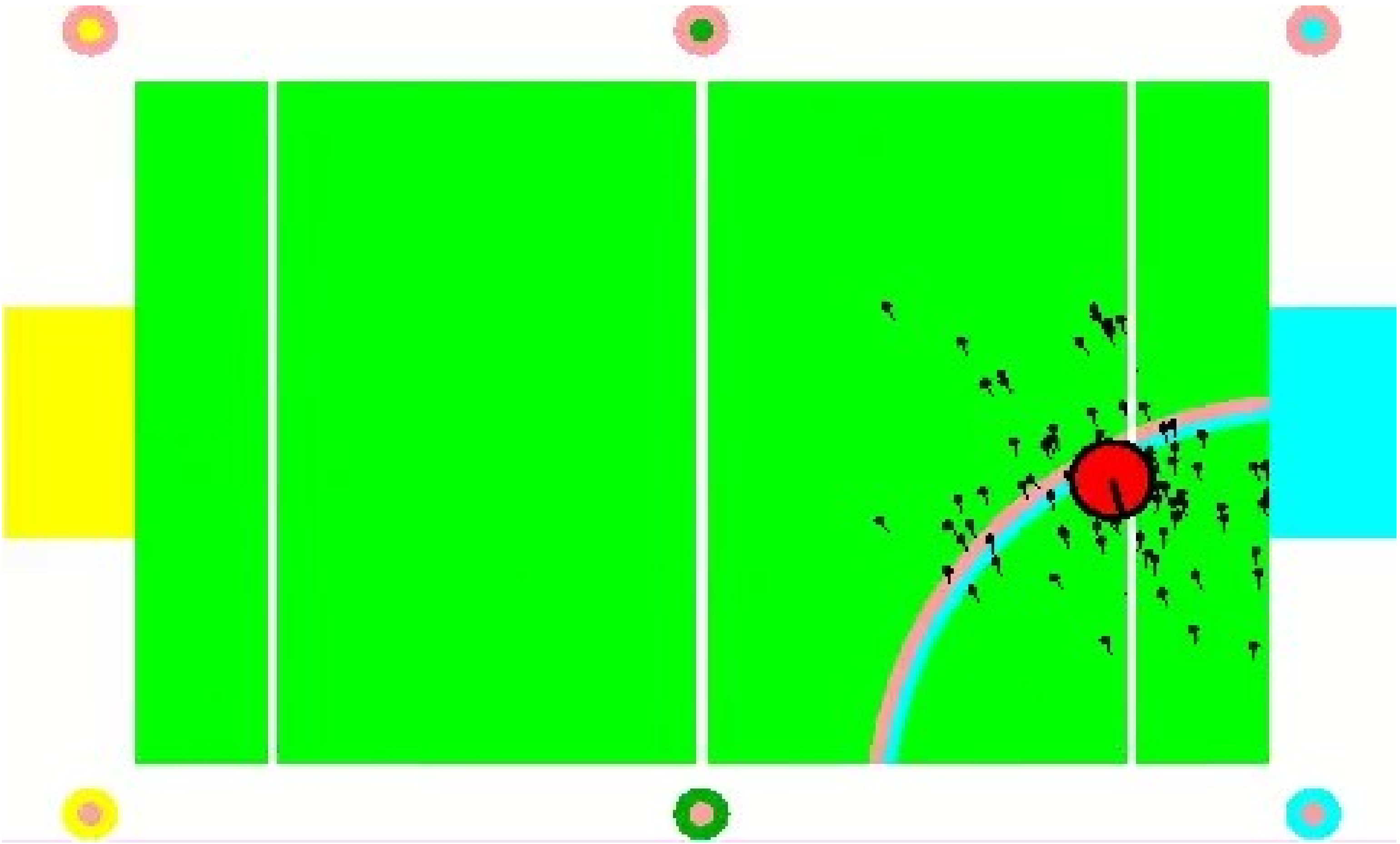
Global Localization Using Vision



Example in RoboCup



Example in RoboCup



Example in Simulation

How does all of this work in ROS?

- ROS package for "Adaptive Monte-Carlo Localization" with 2D laser readings: ***amcl***
- ROS package for building 2D maps: ***gmapping***

THE END