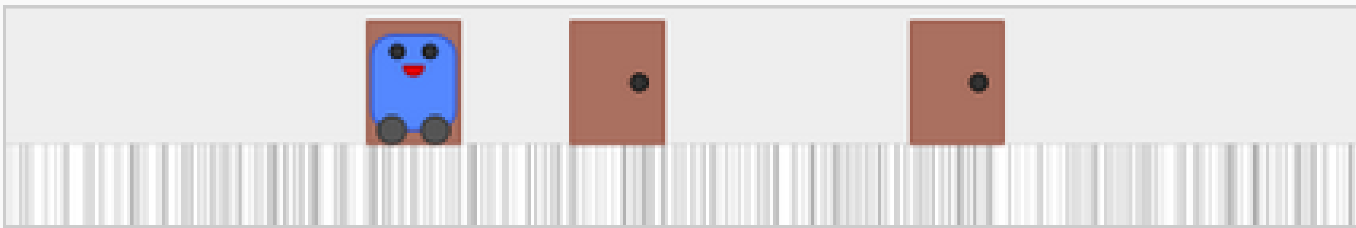
Particle Filtering
A Simple Example

Mohsen Afsharchi

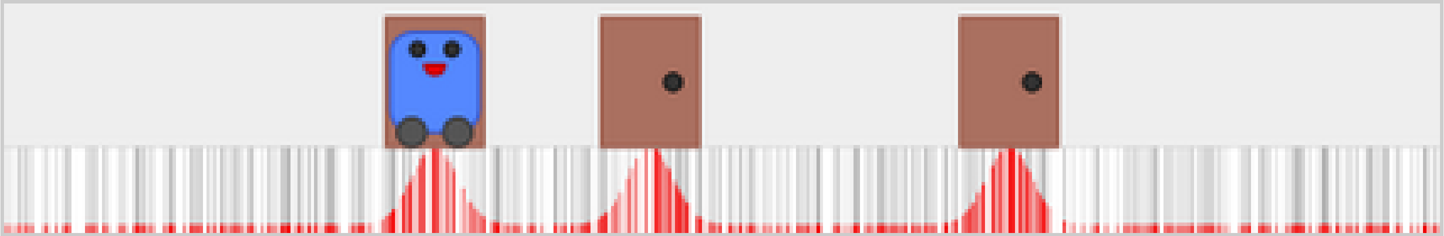
A Scenario



The algorithm initializes with a uniform distribution of particles. The robot considers itself equally likely to be at any point in space along the corridor, even though it is physically at the first door.

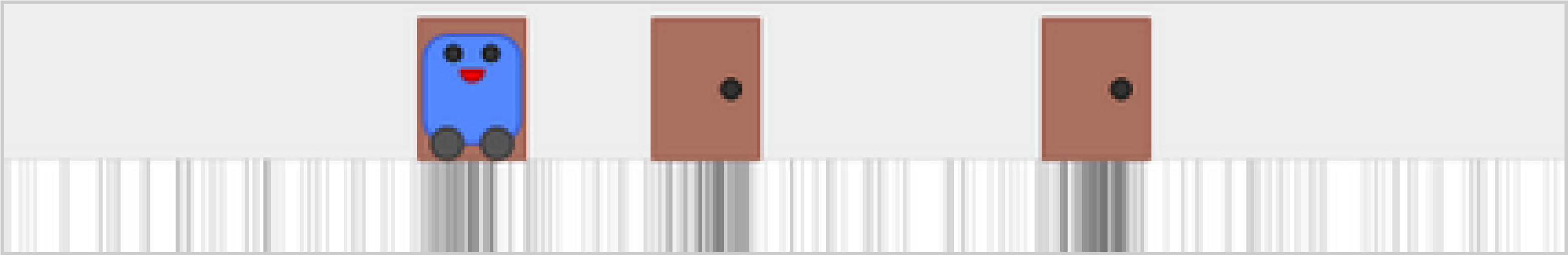
Observation

$t = 0$



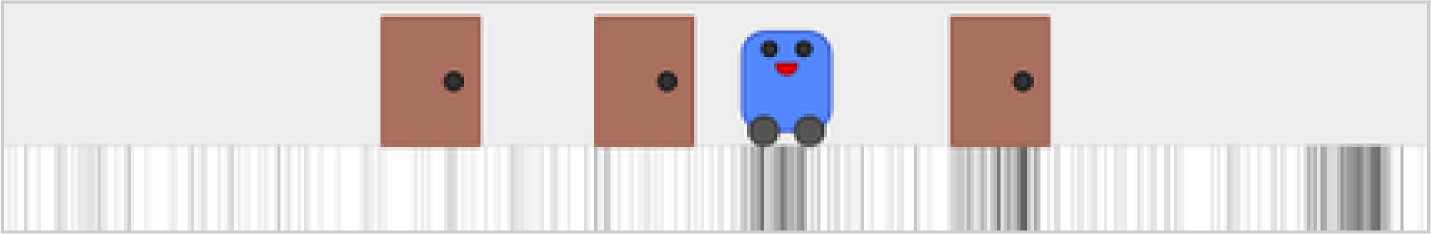
Sensor update: the robot detects **a door**. It assigns a weight to each of the particles. The particles which are likely to give this sensor reading receive a higher weight.

Resampling



Resampling: the robot generates a set of new particles, with most of them generated around the previous particles with more weight. It now believes it is at one of the three doors.

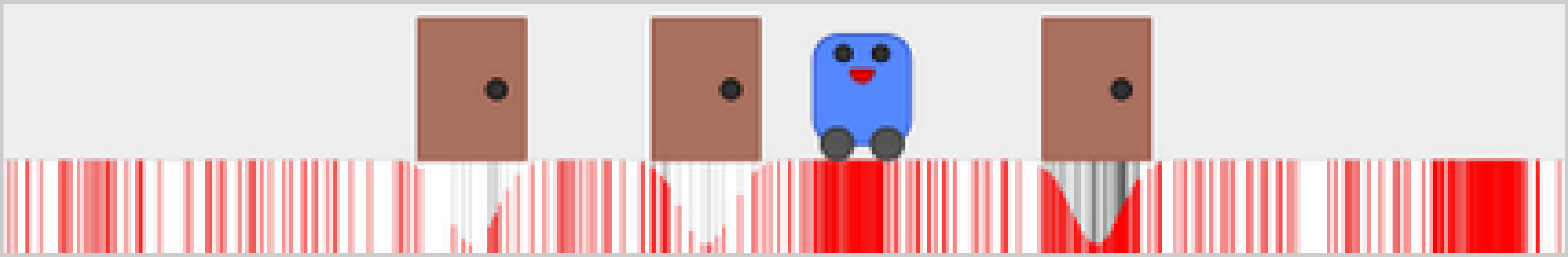
Motion Update



Motion update: the robot moves some distance to the right. All particles also move right, and some noise is applied. The robot is physically between the second and third doors.

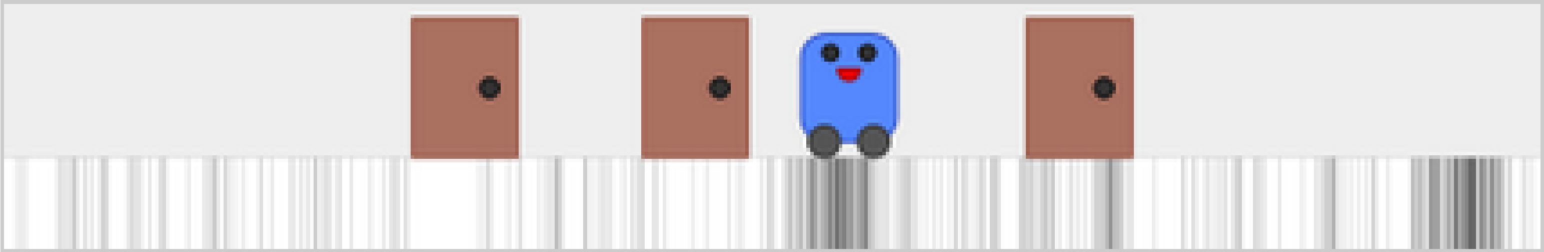
Observation

$$t = 1$$



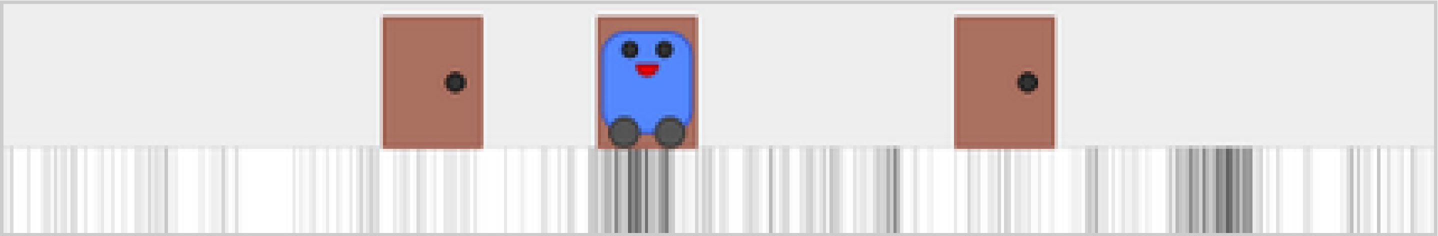
Sensor update: the robot detects **no door**. It assigns a weight to each of the particles. The particles likely to give this sensor reading receive a higher weight.

Resampling



Resampling: the robot generates a set of new particles, with most of them generated around the previous particles with more weight. It now believes it is at one of two locations.

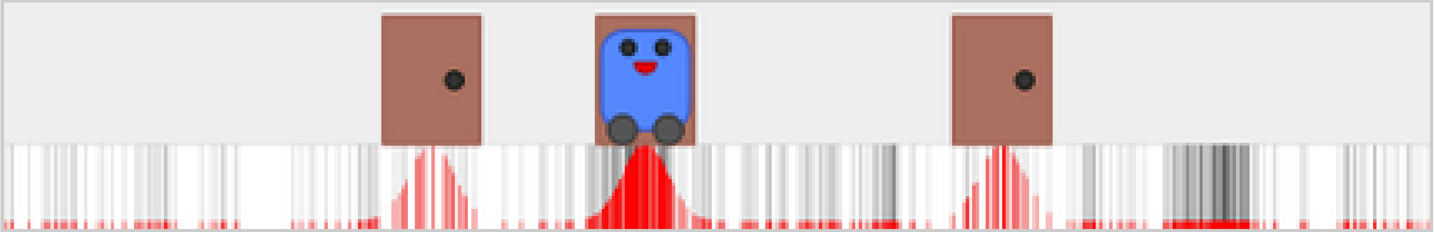
Motion Update



Motion update: the robot moves some distance to the left. All particles also move left, and some noise is applied. The robot is physically at the second door.

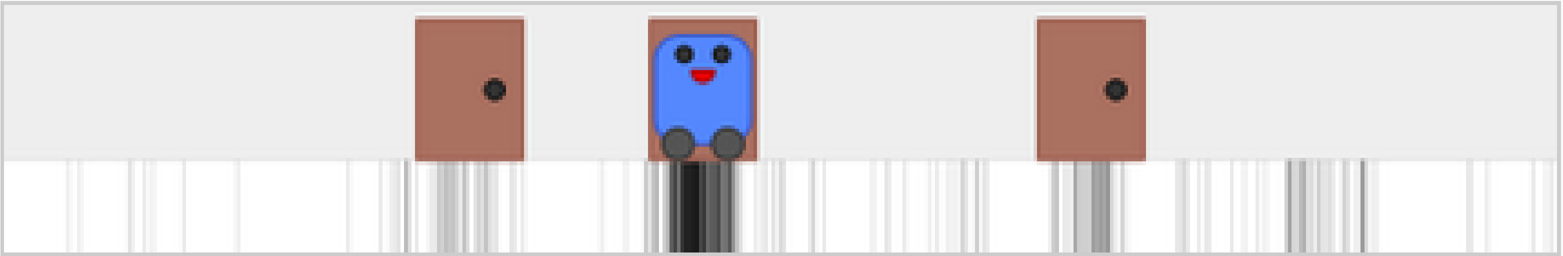
Observation

$$t = 2$$



Sensor update: the robot detects **a door**. It assigns a weight to each of the particles. The particles likely to give this sensor reading receive a higher weight.

Resampling



Resampling: the robot generates a set of new particles, with most of them generated around the previous particles with more weight. The robot has successfully localized itself.

Monte Carlo localization

Algorithm MCL(X_{t-1}, u_t, z_t):

$\bar{X}_t = X_t = \emptyset$

for $m = 1$ to M:

$x_t^{[m]} =$ **motion_update**$(u_t, x_{t-1}^{[m]})$

$w_t^{[m]} =$ **sensor_update**$(z_t, x_t^{[m]})$

$\bar{X}_t = \bar{X}_t + \langle x_t^{[m]}, w_t^{[m]} \rangle$

endfor

for $m = 1$ to M:

draw $x_t^{[m]}$ from $\bar{X}_t$ with probability $\propto w_t^{[m]}$

$X_t = X_t + x_t^{[m]}$

endfor

return X_t