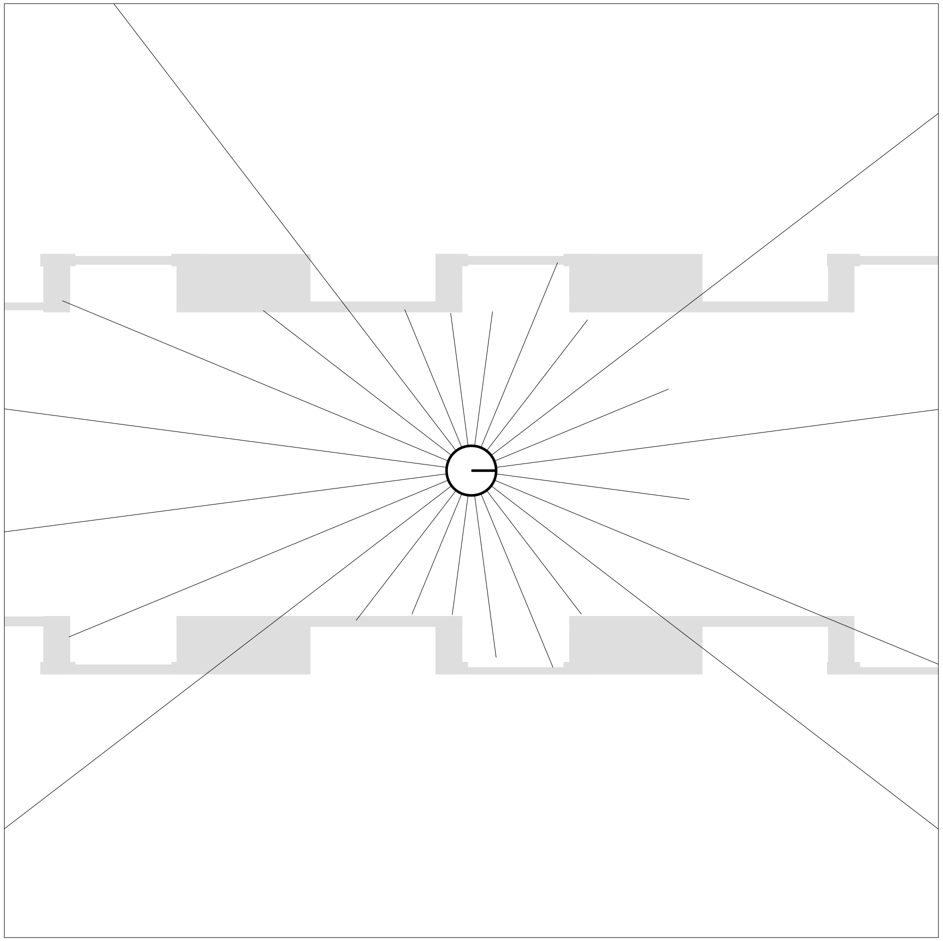
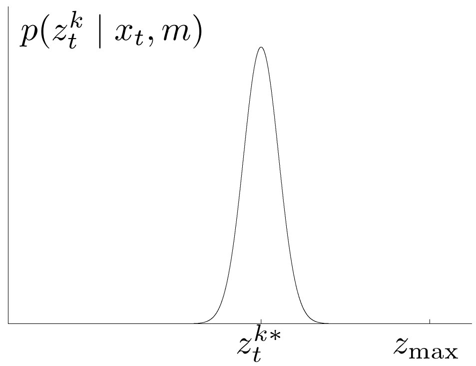
Measurement model Lab

Beam model for range finders

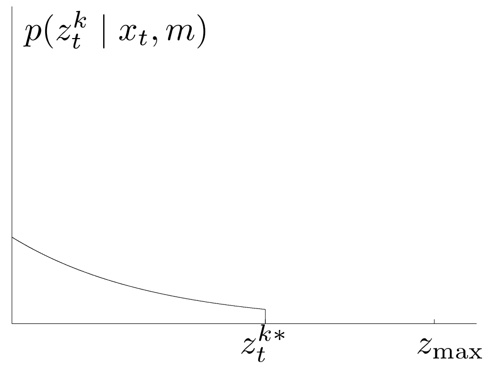


Different sources of errors

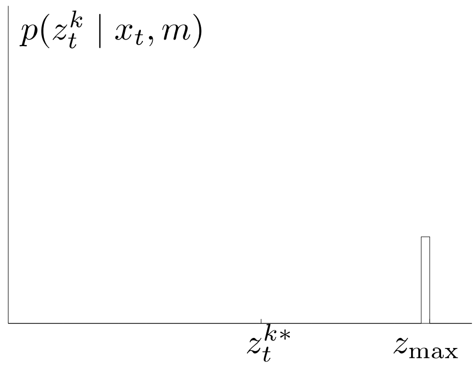
(a) Gaussian distribution p_{hit}



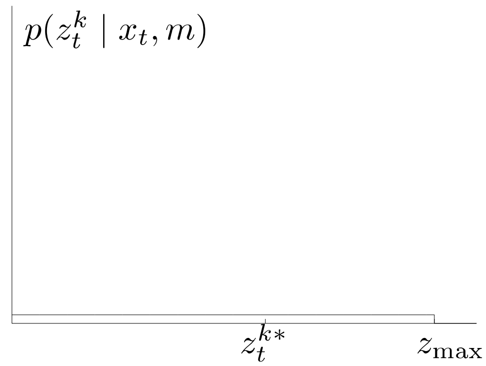
(b) Exponential distribution p_{short}



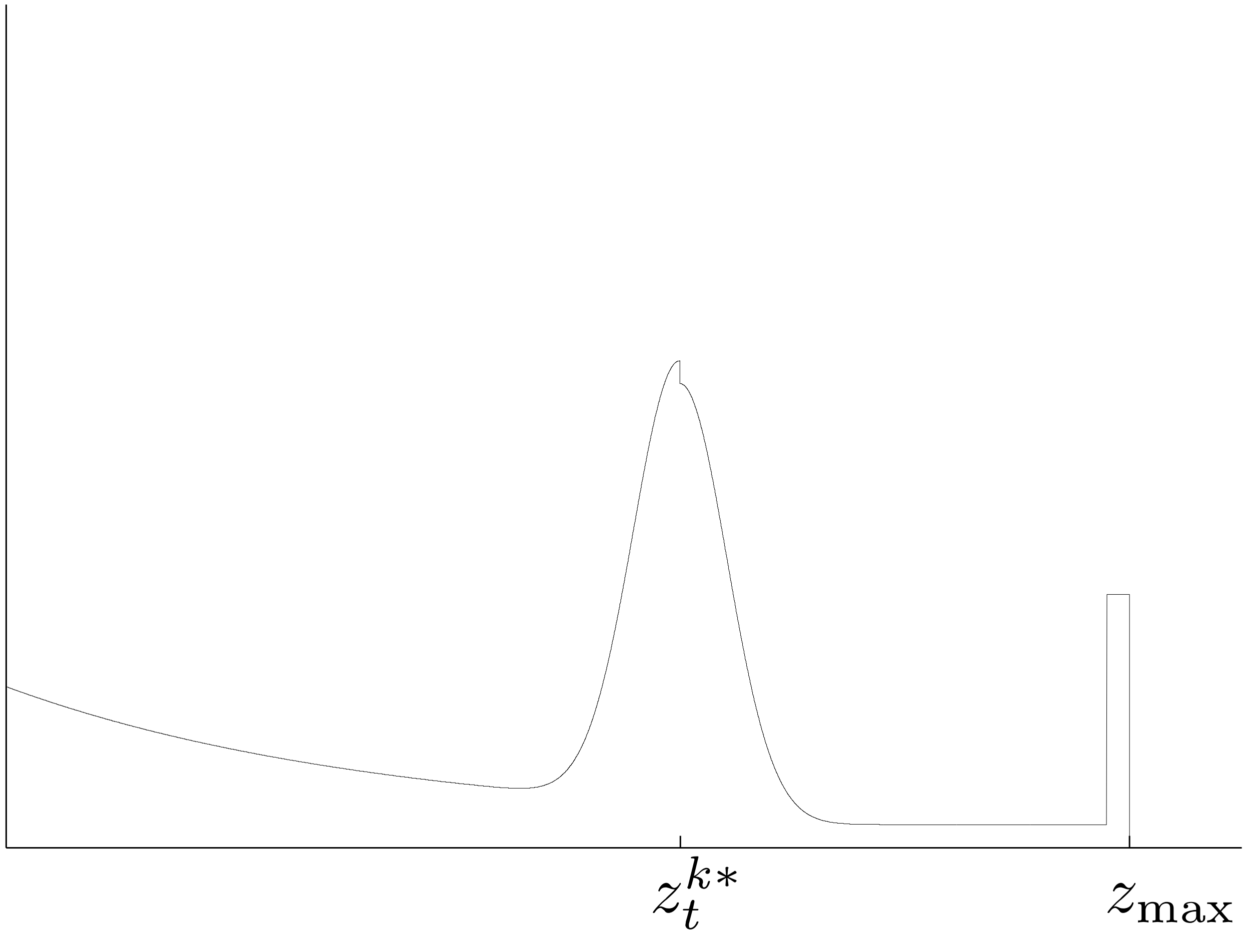
(c) Uniform distribution p_{max}



(d) Uniform distribution p_{rand}



Final distribution



Beam range finder algorithm

Algorithm beam_range_finder_model(z_t, x_t, m**):**

$q = 1$

for $k = 1$ *to* K *do*

compute z_t^{k*} *for the measurement* z_t^k *using ray casting*

$p = z_{\text{hit}} \cdot p_{\text{hit}}(z_t^k \mid x_t, m) + z_{\text{short}} \cdot p_{\text{short}}(z_t^k \mid x_t, m)$

$+ z_{\text{max}} \cdot p_{\text{max}}(z_t^k \mid x_t, m) + z_{\text{rand}} \cdot p_{\text{rand}}(z_t^k \mid x_t, m)$

$q = q \cdot p$

return q

Measurement noise

- The measurement probability is given by:

Found through ray casting

$$p_{hit}\left(z_t^k \mid x_t, m\right) = \begin{cases} \eta N(z_t^k; z_t^{k*}; \sigma_{hit}^2) & \text{if } 0 < z_t^k < z_{\max} \\ 0 & \text{Otherwise} \end{cases}$$

$$N(z_t^k; z_t^{k*}; \sigma_{hit}^2) = \frac{1}{\sqrt{2\pi\sigma_{hit}^2}} e^{-\frac{1}{2}\frac{\left(z_t^k - z_t^{k*}\right)^2}{\sigma_{hit}^2}}$$

$$\eta = \left(\int_0^{z_{max}} \mathcal{N}(z_t^k; z_t^{k*}, \sigma_{hit}^2) dz_t^k \right)^{-1}$$

Unexpected objects

- The probability of having such a object (or short reading) is:

$$p_{short}\left(z_t^k \mid x_t, m\right) = \begin{cases} \eta \lambda_{short} e^{-\lambda short\ z_t^k} & \text{if } 0 \le z_t^k \le z_t^{k*} \\ 0 & \text{Otherwise} \end{cases}$$

- The cumulative probability is given by:

$$\int_0^{z_t^{k*}} \lambda_{short} e^{-\lambda_{short} z_t^k}\, dz_t^k = -e^{-\lambda_{short} z_t^{k*}} + e^{-\lambda_{short}\, 0}$$

$$= 1 - e^{-\lambda_{short} z_t^{k*}}$$

- η is found as:

$$\eta = \frac{1}{1 - e^{-\lambda_{short} z_t^{k*}}}$$

Failures

- Sometimes obstacles are missed altogether. Examples include sonars measuring specular objects, or lasers sensing black, light absorbing objects, or in bright sunlight.
- A sensor error is returned as a *max-range measurement* (z_{max})
- This will be modeled as a point-mass distribution:

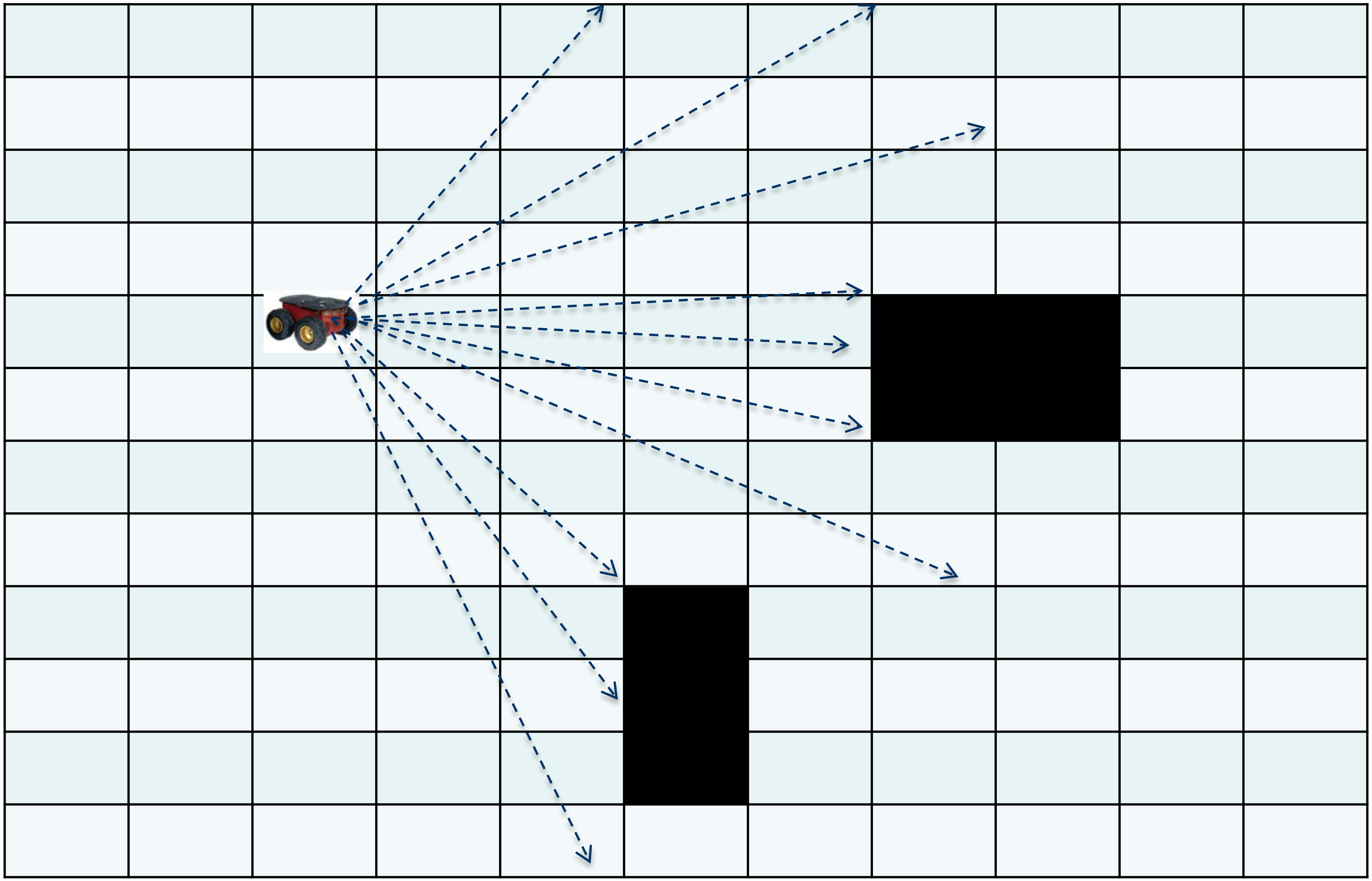
$$p_{\max}\left(z_t^k \mid x_t, m\right) = I(z = z_{\max}) = \begin{cases} 1 & \text{if } z = z_{\max} \\ 0 & \text{Otherwise} \end{cases}$$

Random measurements

- Range finders occasionally produce entirely unexplained measurements (phantom sonar readings, when bouncing off a wall).
- These measurements are modeled by a uniform distribution spread over the entire sensor measurement range.

$$p_{rand}\left(z_t^k \mid x_t, m\right) = \begin{cases} \dfrac{1}{z_{\max}} & \text{if } 0 \leq z_t^k < z_{\max} \\ 0 & \text{Otherwise} \end{cases}$$

How to compute the mean value z_k^{t*}?



How to perform ray casting

- Given a map and the position and the orientation of the SENSOR in the map, it is possible to calculate the “true” distance from the sensor to the nearest object.
- Simplest raycasting methods consider a straight line starting from the center of the sensor at an angle specified by the orientation. The algorithm returns the distance from the sensor to the nearest object.
- You will be provided with two functions, castrays and castraysingle.

Castraysingle

castraysingle(xc,yc,theta, map,lidarrange) takes as input:

- Coordinates of the center of the sensor and its orientation.
- The map should be imported using
  A = imread('maze.jpg');
  map_bw = im2bw(A,0.5);
- Lidarrange is the range of the sensor in pixels

The challenge

- Import the map file provided
- Position your robot at a specified location in the map
- Read from the sensor
- Apply raycasting at given angle and position of the sensor
- Calculate the different probabilities
- Output the final probability q

Tips: You can consider a very high weight z_{hit} and very low probabilities for unexpected objects.