Prof. Kirstin Hagelskjær Petersen
kirstin@cornell.edu

Fast Robots

Sensor Models

Prof. Kirstin Hagelskjær Petersen
kirstin@cornell.edu

ECE 4160/5160
MAE 4910/5910

Fast Robots

Entry survey

Midterm feedback

- We are learning a lot
- *…Ton of work!*
  - Limit tasks
  - Allow teamwork
  - Limit writing
    - Detailed report every 2-4 labs, instead of every lab

Fast Robots

You are almost there!!

- Labs 1-5: Implement your robot
- Labs 6-8: Control and stunts
- Labs 9-12: Localization and Navigation
    - Lab 9: Mapping
    - Flipped classroom: Simulator (Apr 13^{th})
    - https://cei-lab.github.io/FastRobots-2023/FastRobots-Sim.html

You are almost there!!

- Labs 1-5: Implement your robot
- Labs 6-8: Control and stunts
- Labs 9-12: Localization and Navigation
  - Lab 9: Mapping
  - Flipped classroom: Simulator (Apr 13th)
  - Lab 10: Localization (sim), S/U
  - Lab 11: Localization (real)
  - Lab 12: Navigation
- Lectures
  - Bayes Filter/SLAM
  - Ethics
  - Guest lectures: Katie Bradford (Vecna Robotics) and Adam Kane (ASML)
  - Trivia and Showcase (May 9th 9-11.30)

Prof. Kirstin Hagelskjær Petersen
kirstin@cornell.edu

Fast Robots

Sensor models

Bayes Filter

Lecture 17: Motion model

- Odometry model
- Velocity model

1. **Algorithm Bayes_Filter** $(bel(x_{t-1}), u_t, z_t)$:
2. for all x_t do
3. $\overline{bel}(x_t) = \sum_{x_{t-1}} p(x_t|u_t, x_{t-1})\ bel(x_{t-1})$ [Prediction Step]
4. $bel(x_t) = \eta \boxed{p(z_t|x_t)}\ \overline{bel}(x_t)$ [Update/Measurement Step]
5. endfor
6. return $bel(x_t)$

Measurement Probability / Sensor Model

Prof. Kirstin Hagelskjær Petersen
kirstin@cornell.edu

ECE 4160/5160
MAE 4910/5910

Probabilistic Sensor Models

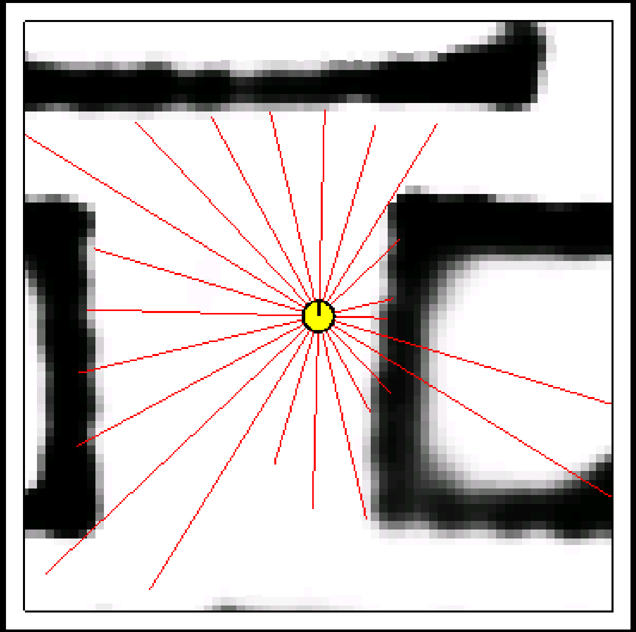
$$p(z|x)$$

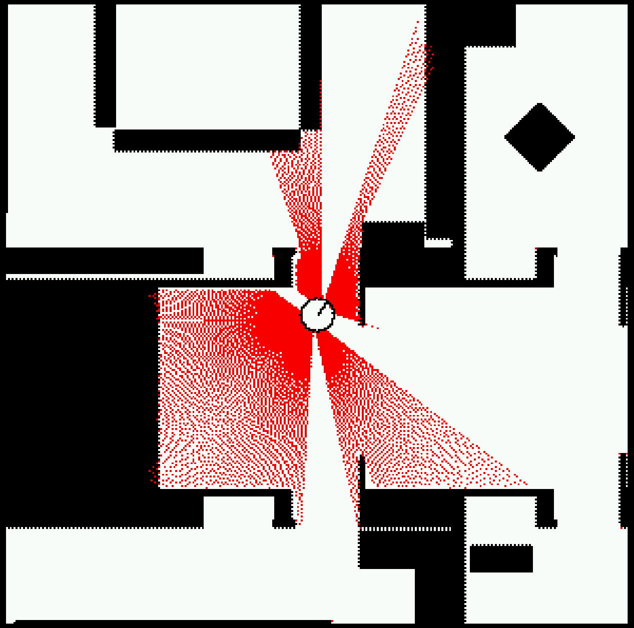
Sensors for Mobile Robots

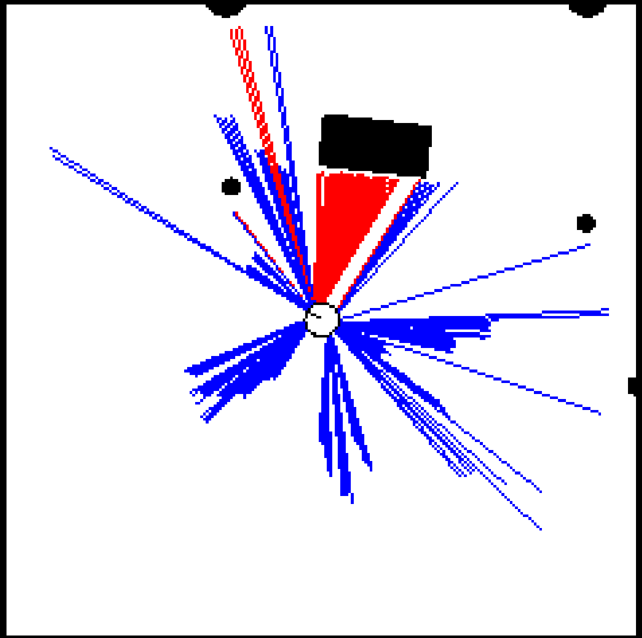
- **Contact Sensors:** Bumpers
- **Internal/Proprioceptive Sensors**
  - Accelerometers (spring-mounted masses)
  - Gyroscopes (spinning mass, laser light)
  - Compasses, inclinometers (earth magnetic field, gravity)
- **Range Sensors**
  - Infrared (intensity)
  - Sonar (time of flight)
  - Radar (phase and frequency)
  - Laser range finders (triangulation, tof, phase)
- **Visual Sensors:** Cameras
- **Satellite-based sensors:** GPS

Sensor Model

- Probabilistic robotics explicitly models the noise in exteroceptive sensor measurements
  - *(What about proprioceptive/odometry sensors?)*
- Where does the noise come from?



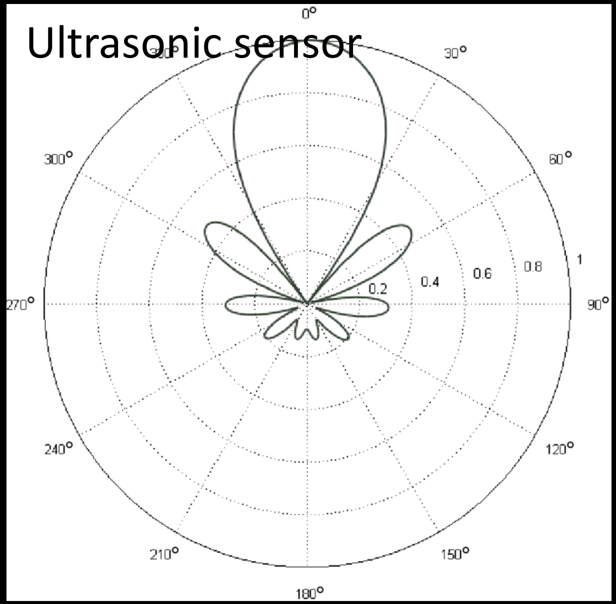


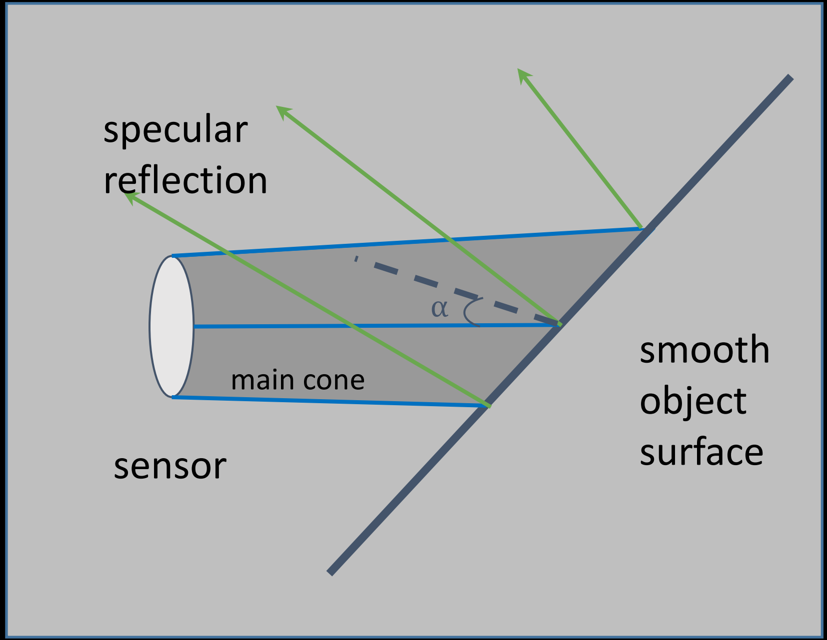


Range Sensor Inaccuracies ("noise")

- Larger readings
  - Surface material
  - Angle between surface normal and direction of sensor cone
  - Width of the sensor cone of measurement
  - Sensitivity of the sensor cone

- Shorter readings
  - Crosstalk between different sensors
  - Unmodeled objects in the proximity of the robot, such as people



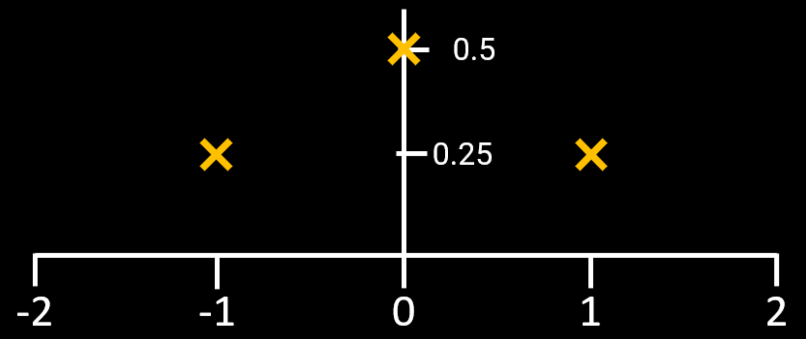


Probabilistic Sensor Model

- Perfect sensor models…
  - $z = f(x)$
  - …practically impossible
  - …computationally intractable
- Practical sensor models…
  - $p(z|x)$

- Three common sensor models
  - Beam model
  - Likelihood model
  - Feature-based model

Up till now our sensor models have been simple
- p(z=correct)
- p(z|X) for a small state space
-



ECE 4160/5160
MAE 4910/5910

Prof. Kirstin Hagelskjær Petersen
kirstin@cornell.edu

Beam Model

Beam Model of Range Finders

- Let there be K individual measurement values within a measurement z_t

$$z_t = \{z_t^1, z_t^2 \dots, z_t^K\}$$

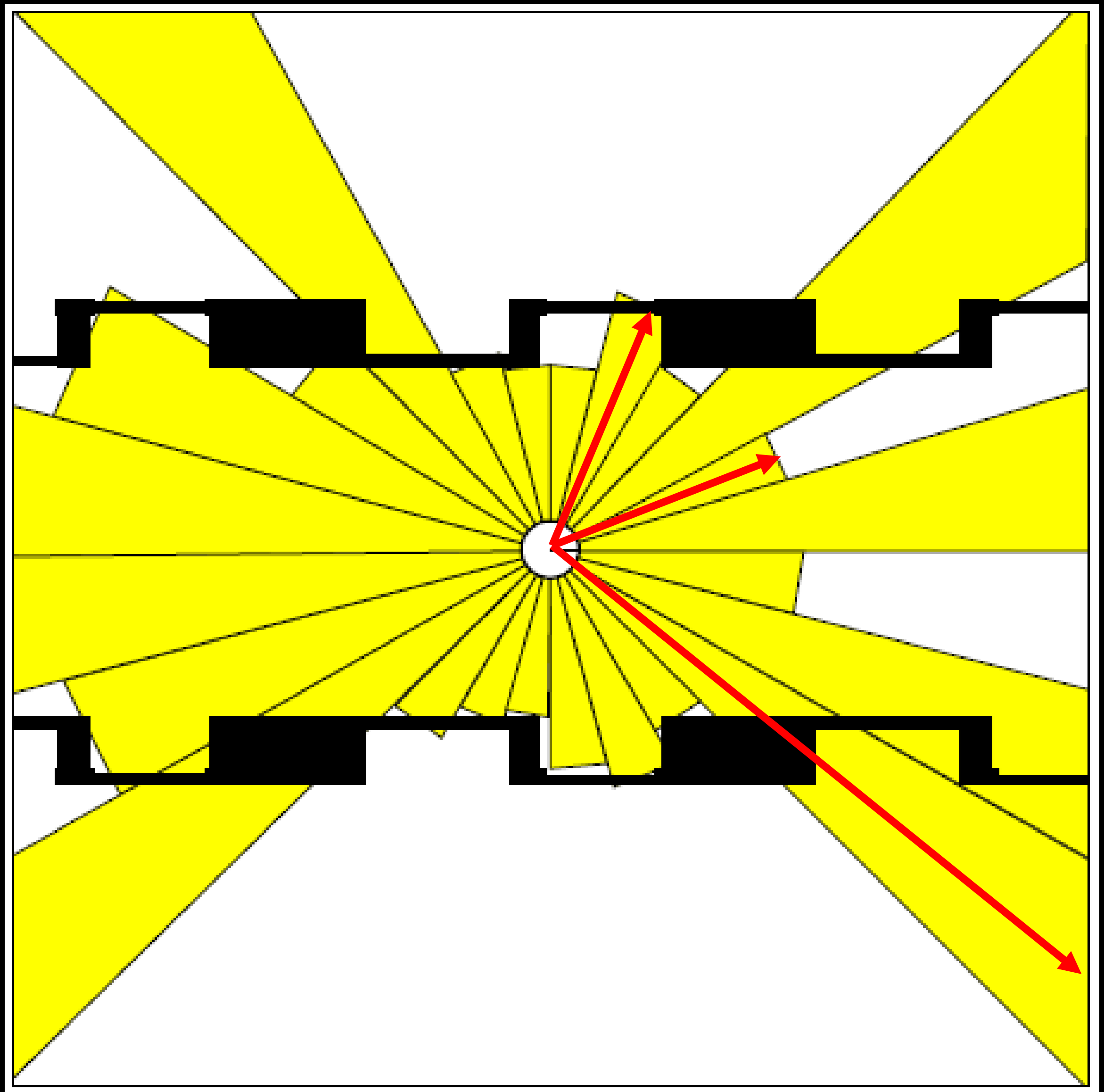
- Individual measurements are independent given the robot state

(Sensor measurements are caused by real world objects)

$$p(z_t|x_t, m) = \prod_{k=1}^{K} p\left(z_t^k \middle| x_t, m\right)$$

- *Can you think of violations to that assumption?*
  - People, errors in the map model *m*, approximations in the posterior, etc.
  - But it makes computation much more tractable…

Typical Measurement Errors of an Range Measurements



Typical Measurement Errors of an Range Measurements

1. Correct Range Measurements

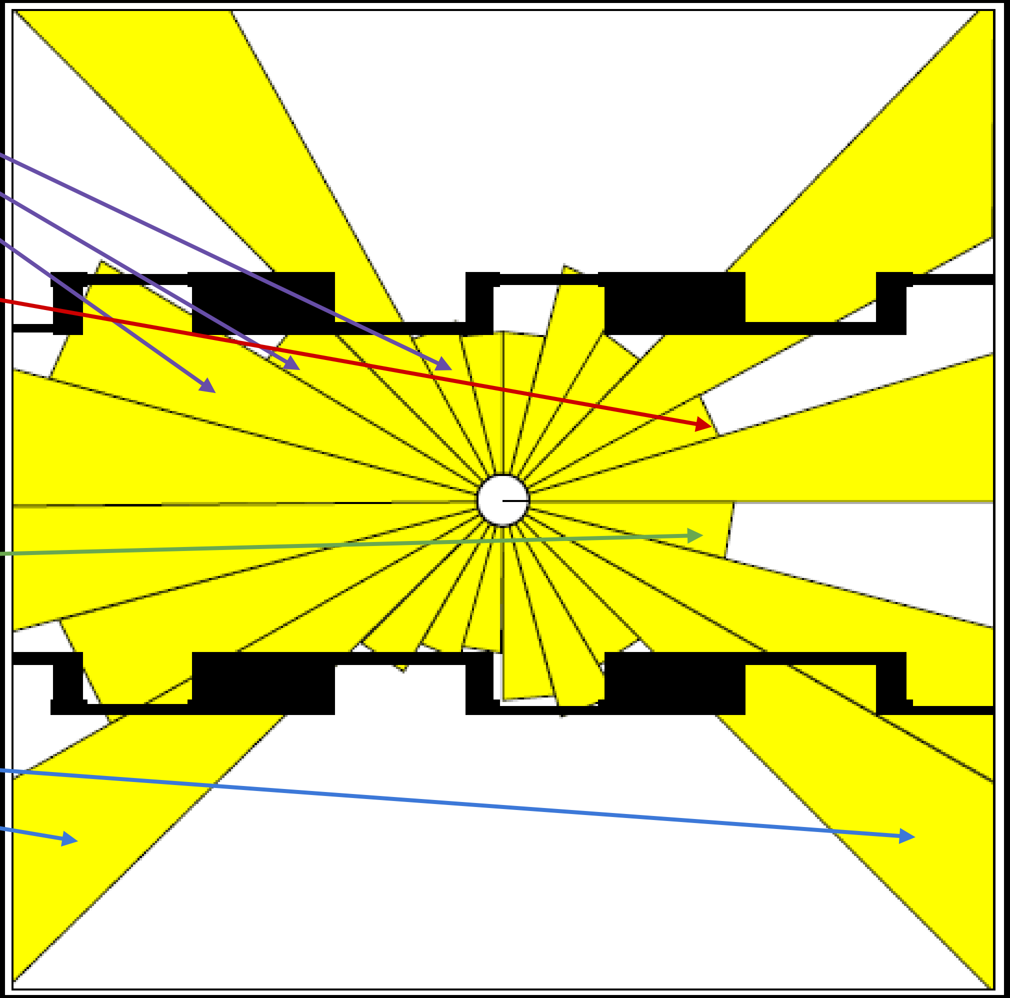
   Beams reflected by obstacles

2. Unexpected Objects

   Beams reflected by persons / caused by crosstalk

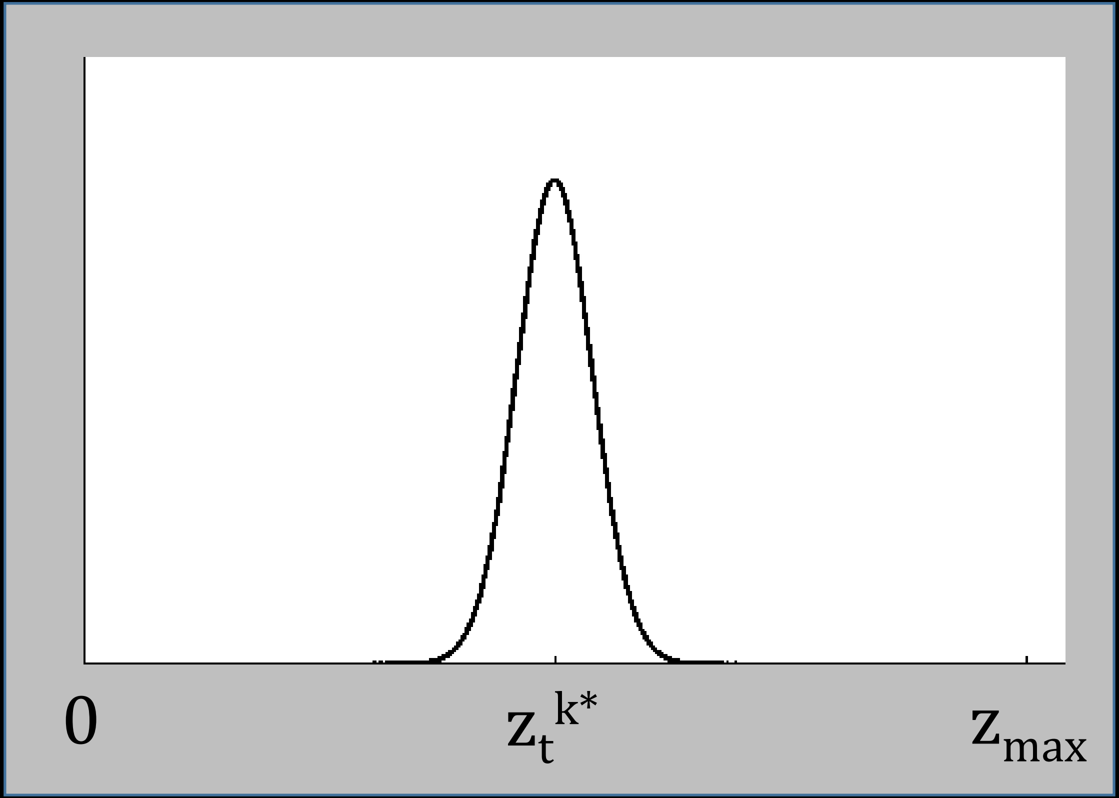
3. Failures

4. Random measurements



1. Correct Range Measurements

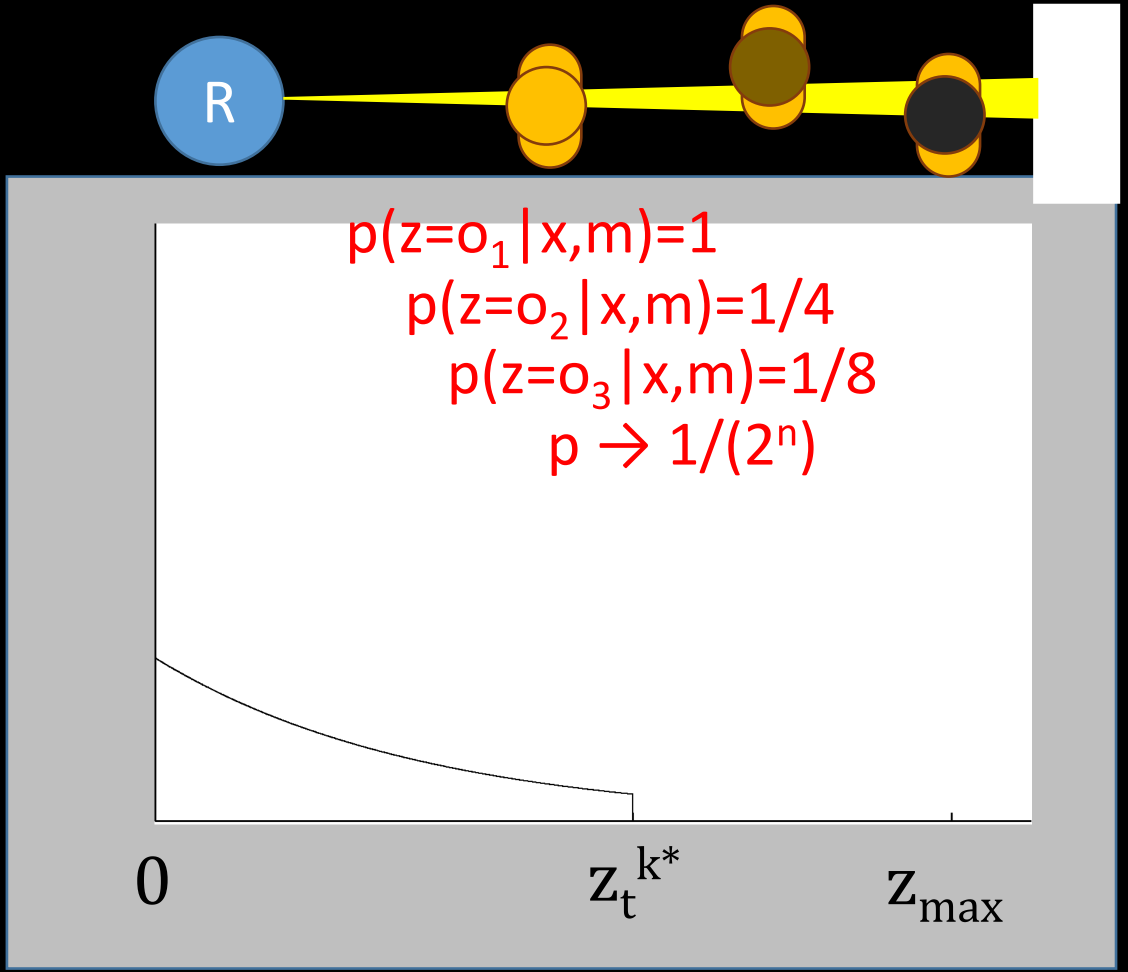
- Reading
  - z_t^k
- True value
  - z_t^{k*}
  - In a location-based map, z_t^{k*} is usually estimated by *ray casting*
- Measurement noise
  - Narrow Gaussian p_{hit} with mean z_t^{k*} and standard deviation σ_{hit}



$$p_{hit}(z_t^k | x_t, m) = \begin{cases} \eta f(z_t^k, z_t^{k*}, \sigma_{hit}) & if\ 0 \le z_t^k \le z_{max} \\ 0 & otherwise \end{cases}$$

2. Unexpected Objects

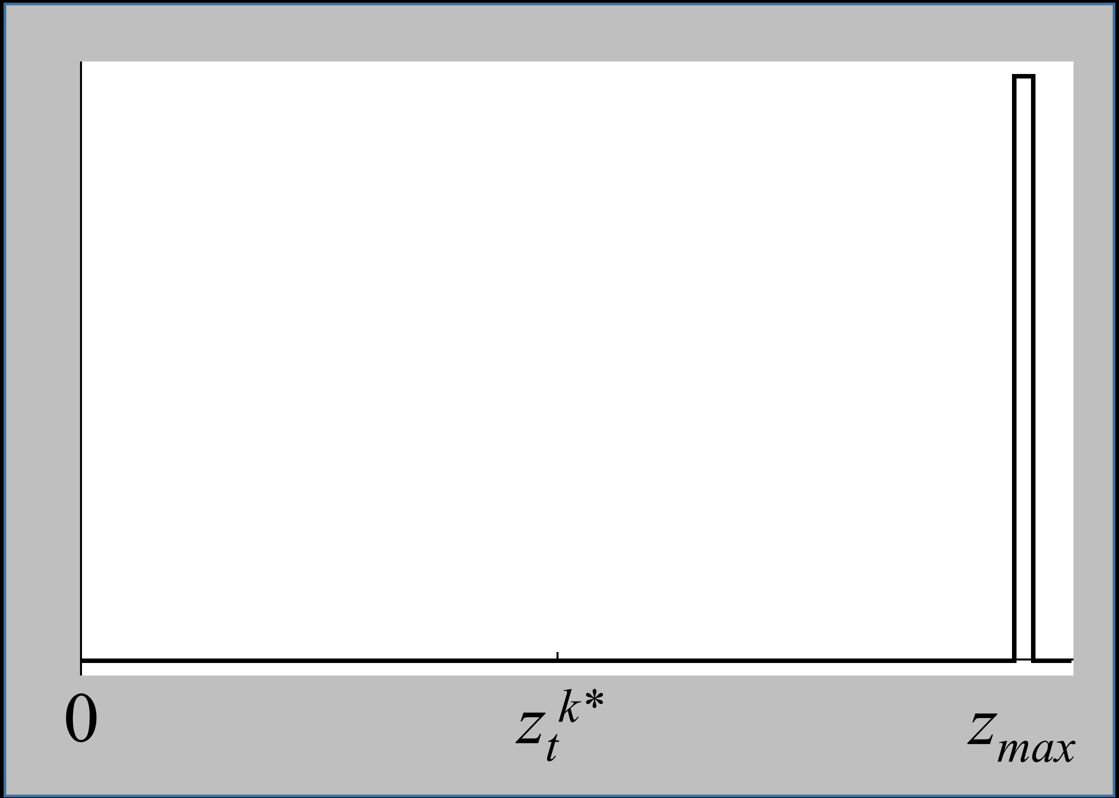
- Real world is dynamic
- Objects not contained in the map can cause shorter readings
  - treat them as part of the state vector and estimate their location
  - Or treat them as sensor noise
- The likelihood of sensing unexpected objects decreases with range
- Model as an exponential distribution p_{short}



$$p_{short}(z_t^k | x_t, m) = \begin{cases} \eta \, \lambda_{short} \, e^{-\lambda_{short} z_t^k} & if\ 0 \le z_t^k \le z_t^{k*} \\ 0 & otherwise \end{cases}$$

3. Failures

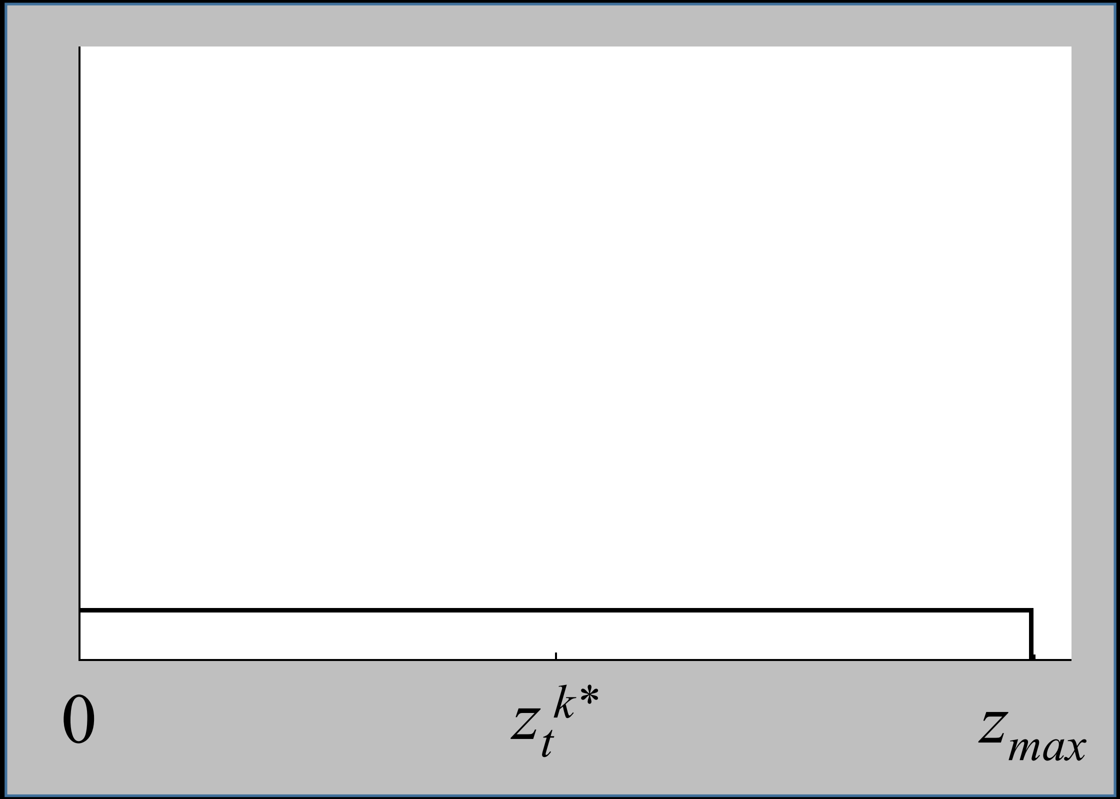
- Obstacles might be missed altogether
- The result is a max-range measurement z_{max}
- Model as a point-mass distribution p_{max}



$$p_{max}(z_t^k | x_t, m) = I(z = z_{max}) = \begin{cases} 1 & if\ z = z_{max} \\ 0 & otherwise \end{cases}$$

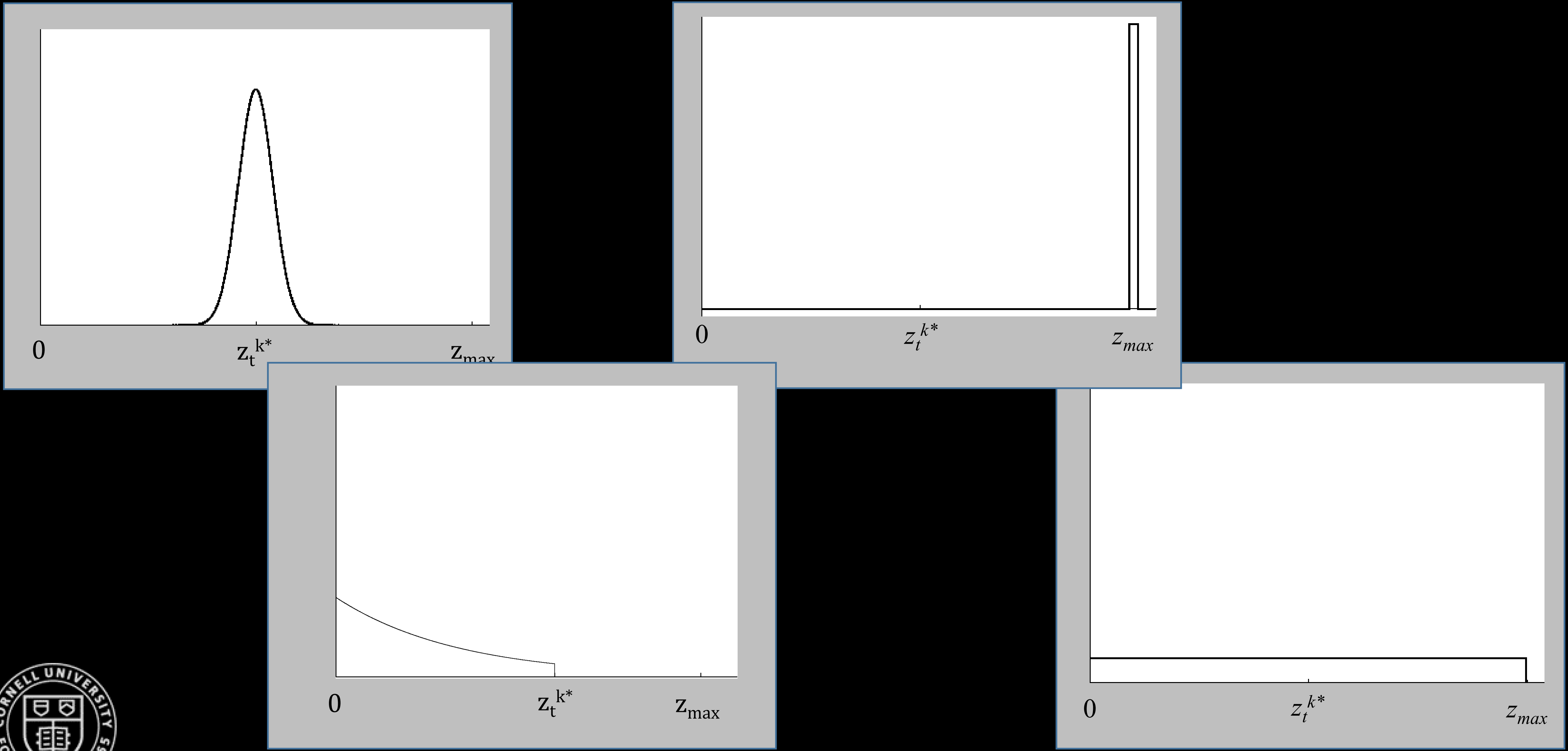
4. Random Measurements

- Range finders can occasionally produce entirely unexplainable measurements
- Modelled as a uniform distribution p_{rand} over the measurement range



$$p_{rand}(z_t^k | x_t, m) = \begin{cases} \frac{1}{z_{max}} & if\ 0 \leq z_t^k \leq z_{max} \\ 0 & otherwise \end{cases}$$

Beam Model

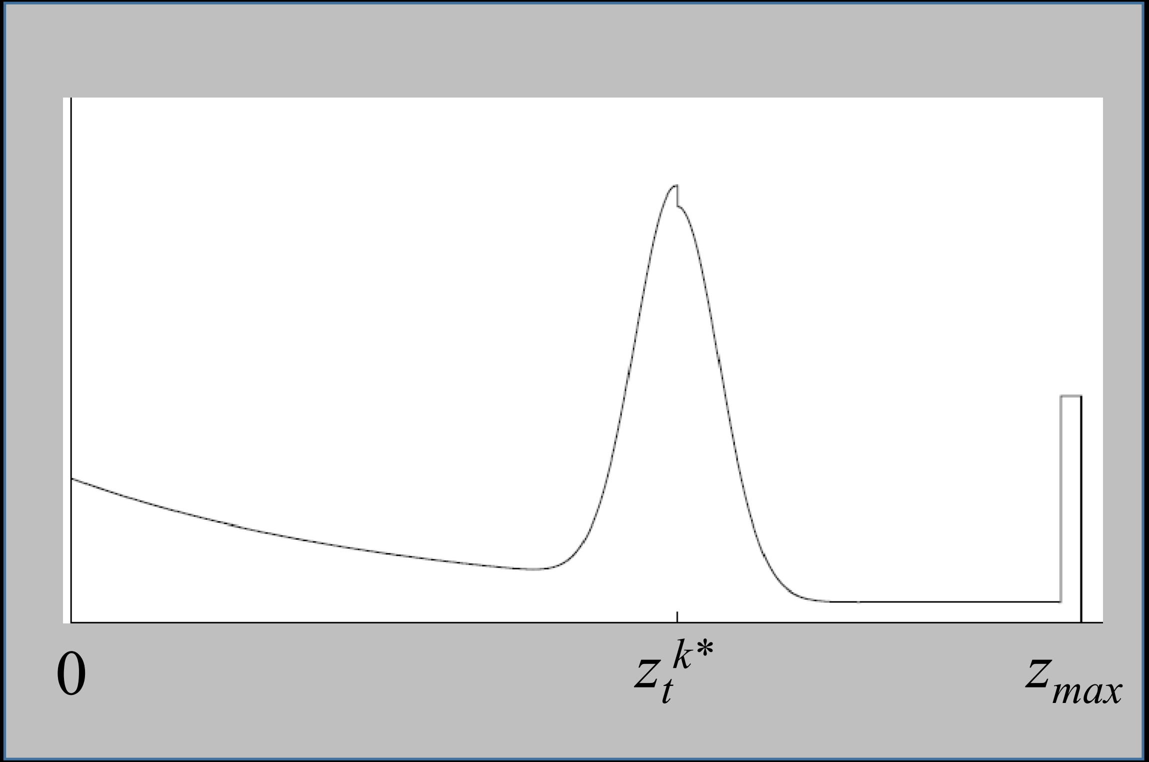


Beam Range Model as a Mixture Density

- The four different distributions are mixed by a weighted average

$$p(z_t^k|x_t,m) = \begin{pmatrix} \alpha_{hit} \\ \alpha_{short} \\ \alpha_{max} \\ \alpha_{rand} \end{pmatrix} \cdot \begin{pmatrix} p_{hit}(z_t^k|x_t,m) \\ p_{short}(z_t^k|x_t,m) \\ p_{max}(z_t^k|x_t,m) \\ p_{rant}(z_t^k|x_t,m) \end{pmatrix}$$

- $\alpha_{hit} + \alpha_{short} + \alpha_{max} + \alpha_{rand} = 1$



Algorithm for Beam Model

Algorithm beam_range_finder_model(z_t, x_t, m):

$q = 1$

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

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

  $p = \alpha_{hit} . p_{hit}(z_t^k | x_t, m) + \alpha_{short} . p_{short}(z_t^k | x_t, m)$

  $\quad + \alpha_{max} . p_{max}(z_t^k | x_t, m) + \alpha_{rand} . p_{rand}(z_t^k | x_t, m)$

  $q = q . p$

return q

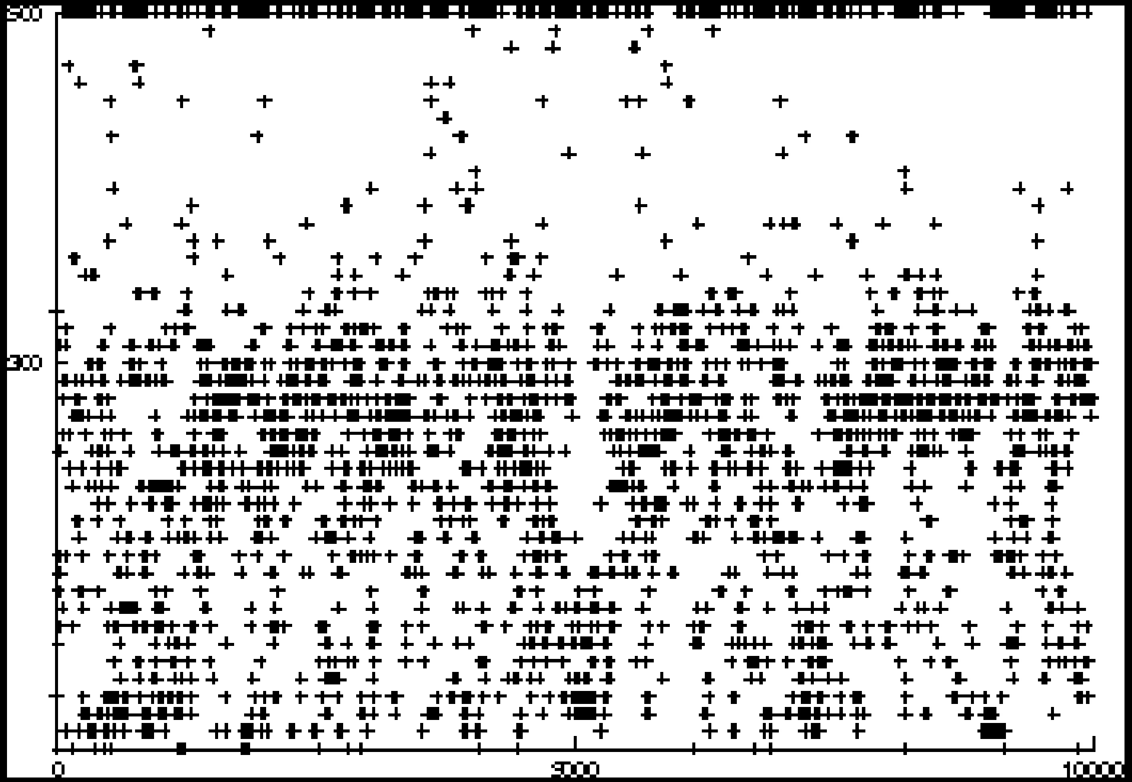
Parameters of Beam Range Model

- Intrinsic Parameters Θ of the beam range model
  - α_{hit} , α_{short} , α_{max} , α_{rand} , λ_{short}
  - Affect the likelihood of any sensor measurement

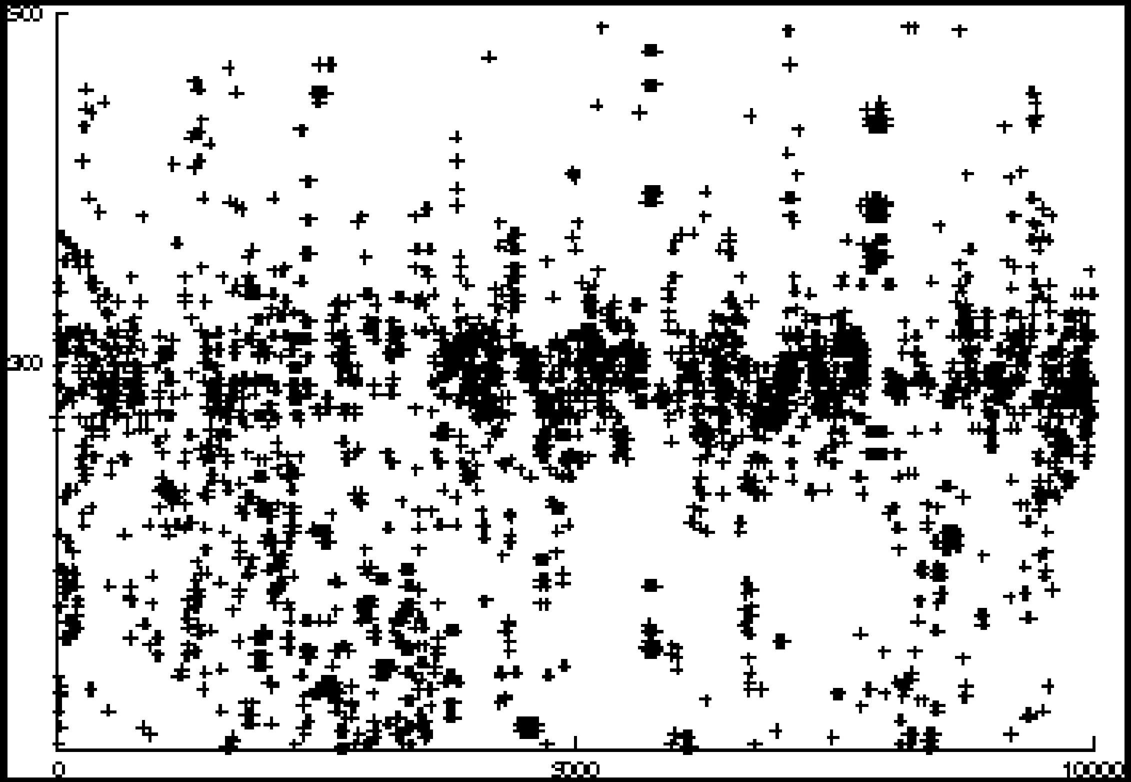
- Estimation Methods
  - Guesstimate the resulting density
  - Learn parameters using a Maximum Likelihood Estimator
  - Hill Climbing, Gradient descent, Genetic algorithms, etc.

Raw Sensor Data

Sonar senor data



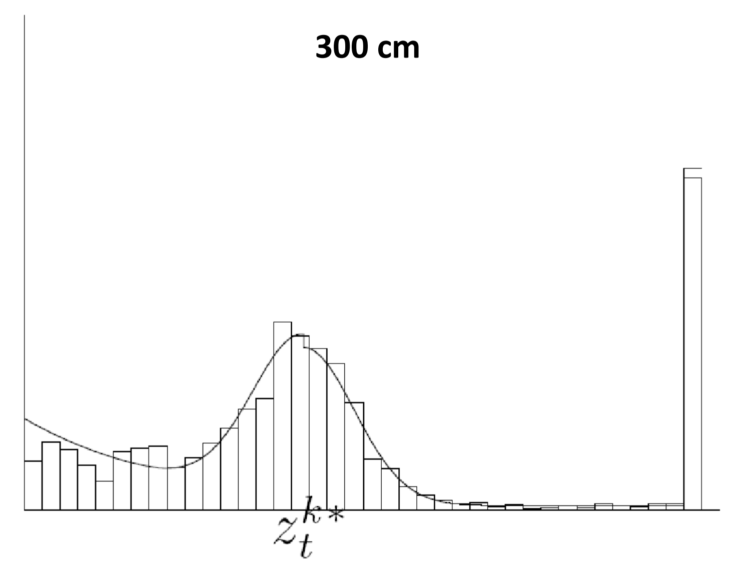
Laser range sensor

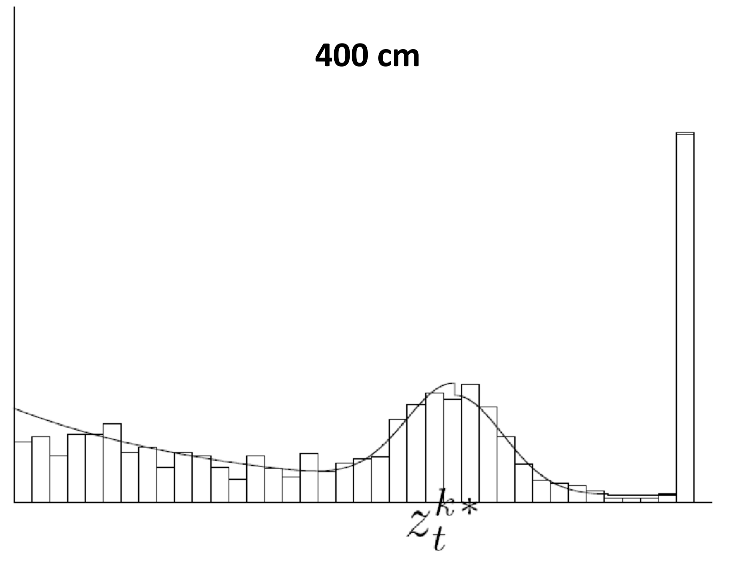


(True range is 300 cm and maximum range is 500 cm)

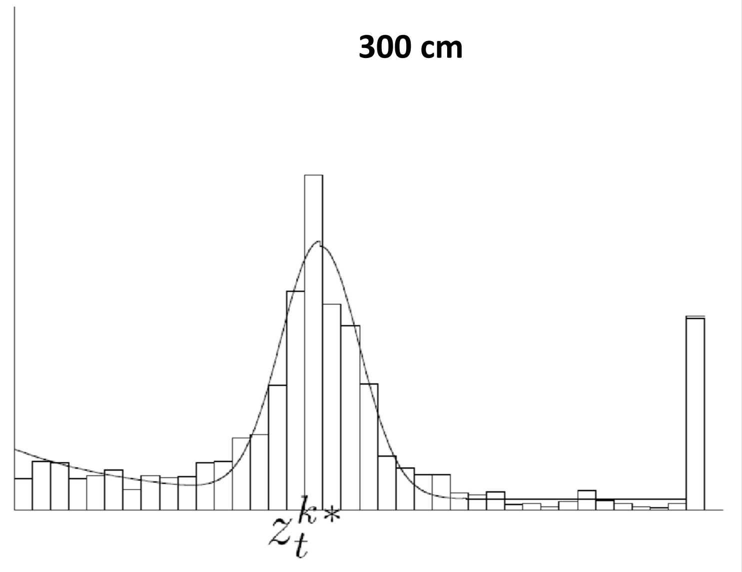
Approximation Results of Beam Model (with MLE)

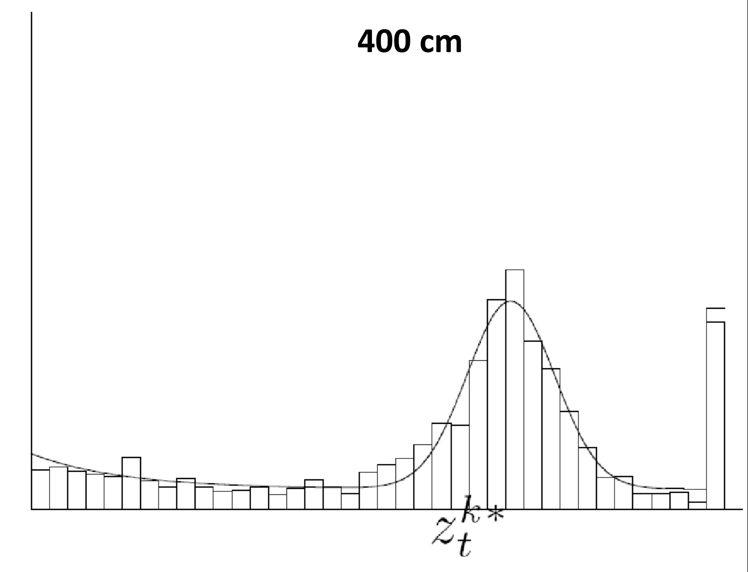
Sonar Data



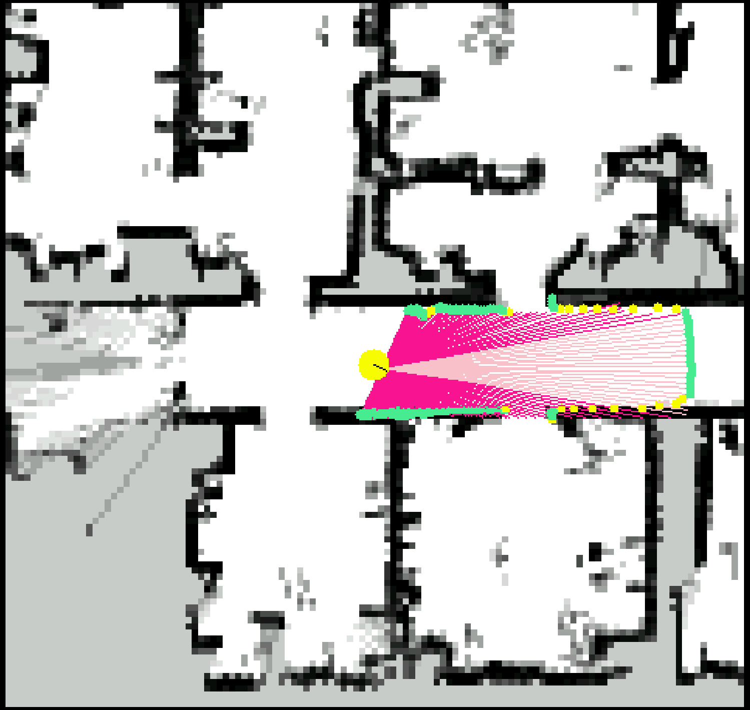


Laser Data

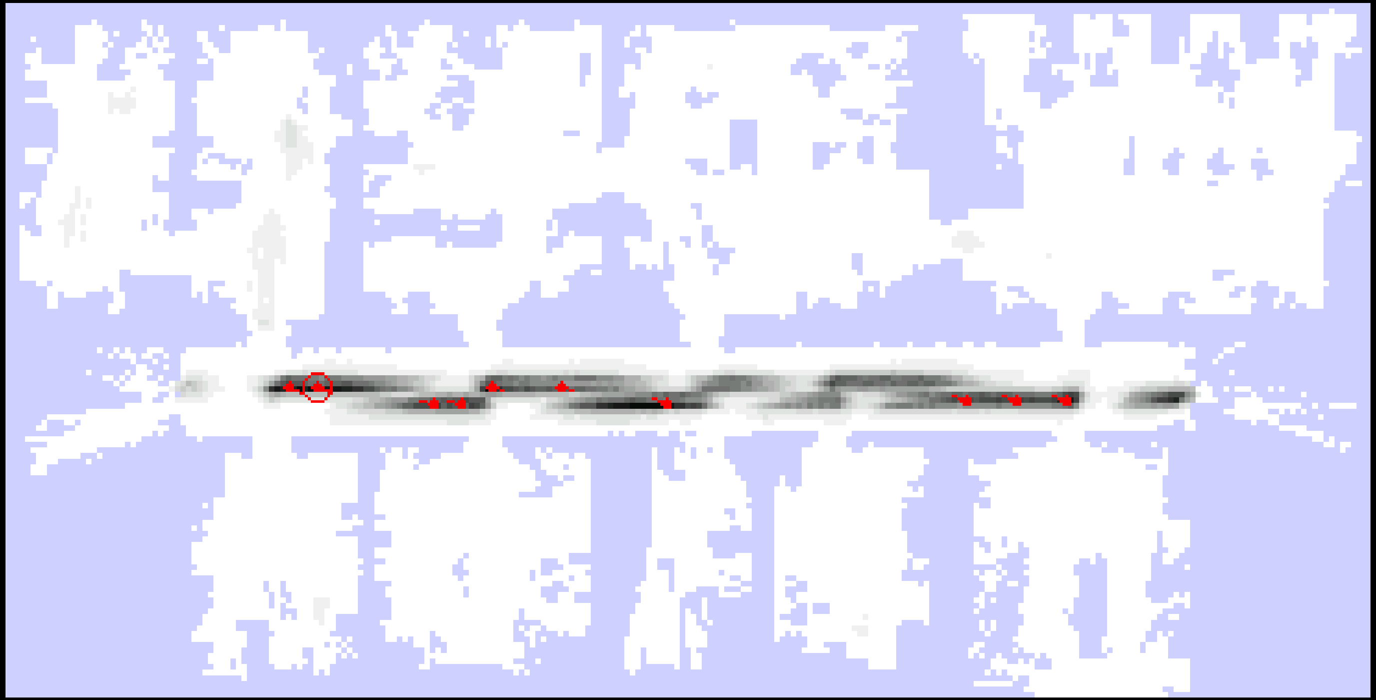




Beam Model in Action



Laser scan projected into a partial map m.



Likelihood $p(z_t|x_t, m)$ for all positions x_t projected into the map. The darker a position, the larger $p(z_t|x_t, m)$

Summary of Beam Model

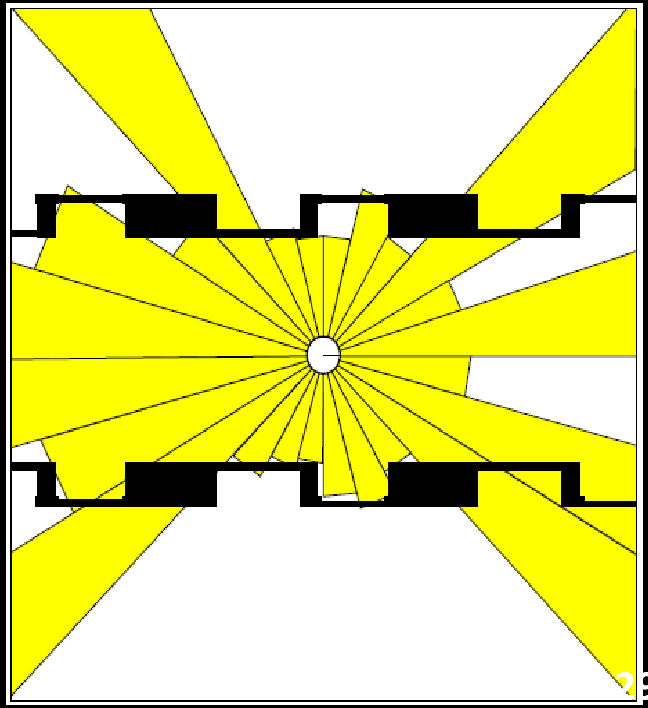
- Overconfident
  - Assumes independence between individual measurements
- Models physical causes for measurements
- Implementation involves learning parameters based on real data
- Limitations
  - Different models are needed for every possible scenario (e.g. hit angles for intensity sensors)
  - Raytracing is computationally expensive
    - But can be pre-processed
  - Not smooth for small obstacles, at edges, or in cluttered environments

Prof. Kirstin Hagelskjær Petersen
kirstin@cornell.edu

Likelihood Fields

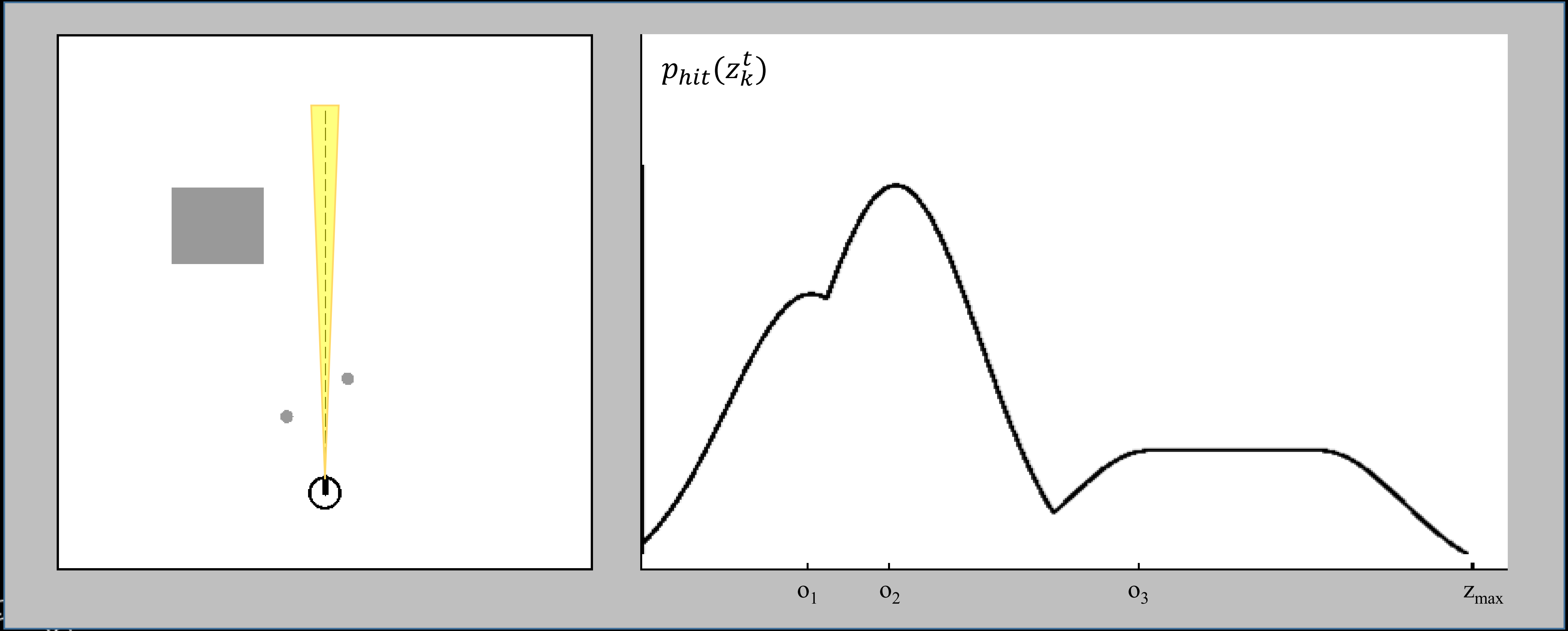
Likelihood Fields of Range Finders

- Instead of following along the beam, just check the end point
- Project sensor scan z_t into the map and compute the closest end point
- Probability function is a mixture of
  - a Gaussian distribution with mean at the distance to closest obstacle
  - a uniform distribution for random measurements
  - a point mass distribution for max range measurements



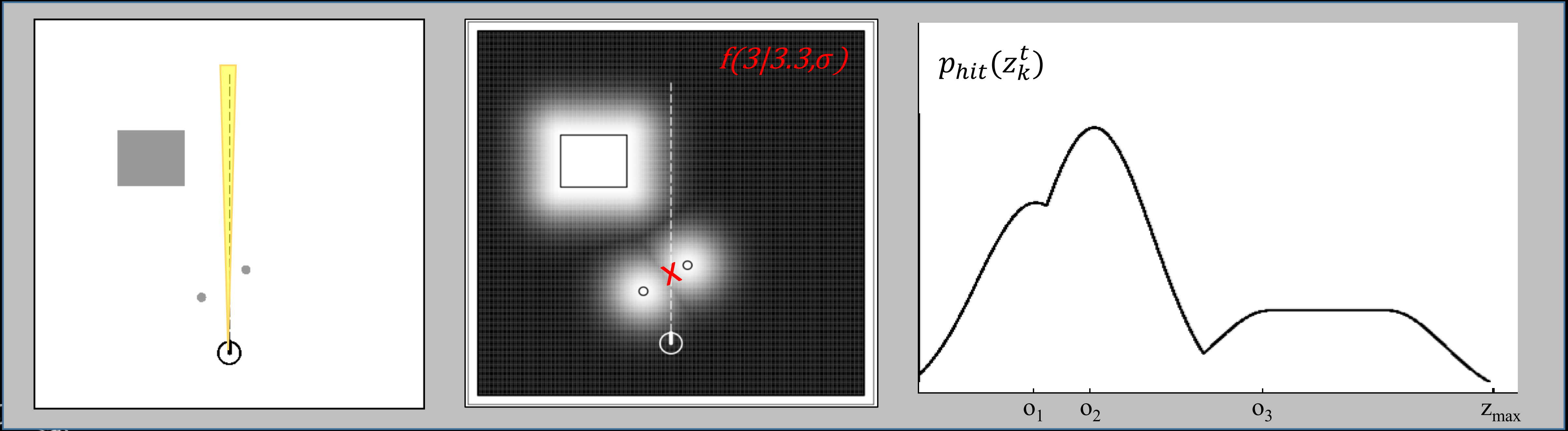
Measurement Noise

- Modelled using Gaussians



Measurement Noise

- Modelled using Gaussians
- In xy space, this involves finding the nearest obstacle in the map
- The probability of a sensor measurement is given by a Gaussian that depends on the euclidean distance between measurement coordinates and nearest object in the map m



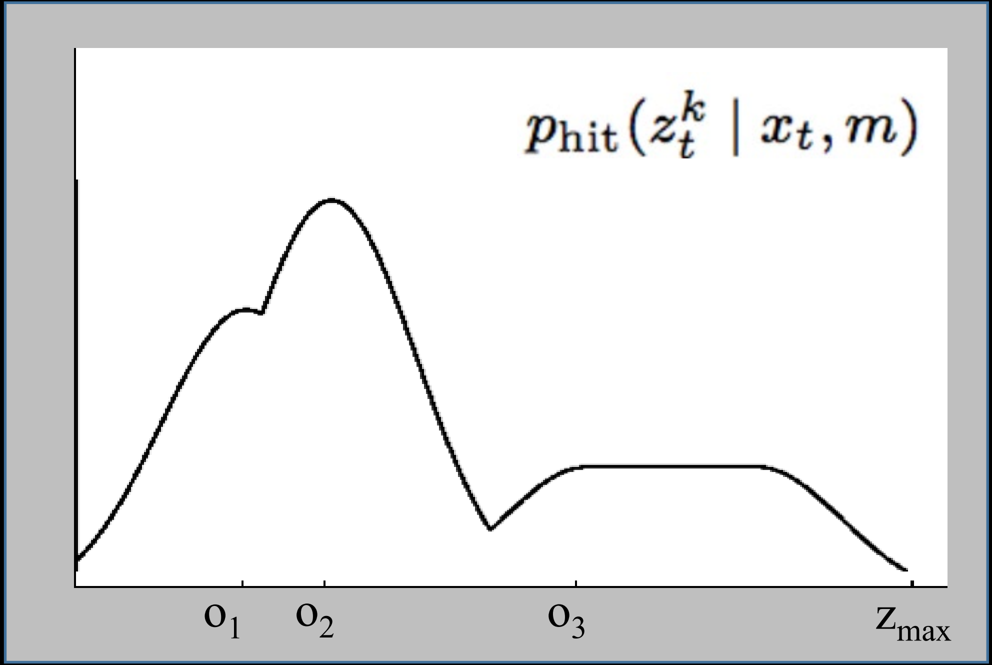
Likelihood Fields for Range Finders

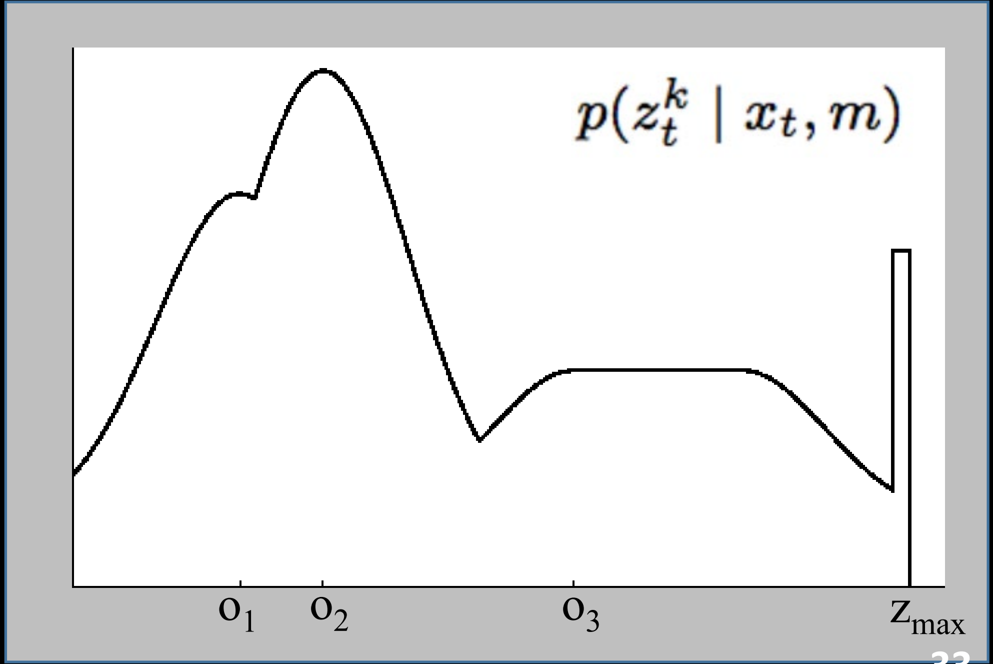
- Robot pose in world frame: $x_t = (x, y, \theta)^T$
- Sensor measurement in the robot frame: $(x_{k,sens}, y_{k,sens}, \theta_{k,sens})$
- $z^k{}_t$ hit/"end" points in the world frame

$$\begin{pmatrix} x_{z_t^k} \\ y_{z_t^k} \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} x_{z^{k,sens}} \\ y_{z^{k,sens}} \end{pmatrix} + z_t^k \begin{pmatrix} \cos(\theta + \theta^{k,sens}) \\ \sin(\theta + \theta^{k,sens}) \end{pmatrix}$$

Likelihood Fields for Range Finders

- Assume independence between individual measurements
- Three types of sources of noise and uncertainty
    - Measurement noise
    - Failures
        - Max range readings are modelled by a point-mass distribution
    - Unexplained random measurements
        - Uniform distribution





Algorithm for Likelihood Fields

1. **Algorithm likelihood_field_range_finder_model**$(z_t\, , x_t\, , m)$:
2. $\mathrm{q} = 1$
3. for $k = 1\ to\ K\ do$
4. $\quad x_{z_k^t} = x + x_{z_k,sens} \cos(\theta) - y_{z_k,sens}\ \sin(\theta) + z_k^t \cos(\theta +\ \theta_{k,sens})$
5. $\quad y_{z_k^t} = y + y_{z_k,sens} \cos(\theta) + x_{z_k,sens}\ \sin(\theta) + z_k^t \sin(\theta +\ \theta_{k,sens})$
6. $\quad dist = \min_{x',y'}\left\{\sqrt{(x_{z_k^t} - x')^2 + (y_{z_k^t} - y')^2}\ |\ \langle x', y' \rangle \text{ occupied in } m\right\}$
7. $\quad q = q.(z_{hit}\, .\, f(dist; 0, \sigma_{hit}) + \frac{z_{rand}}{z_{max}})$
8. return q

Transform sensor reading to world coordinate frame

Find distance to closest object

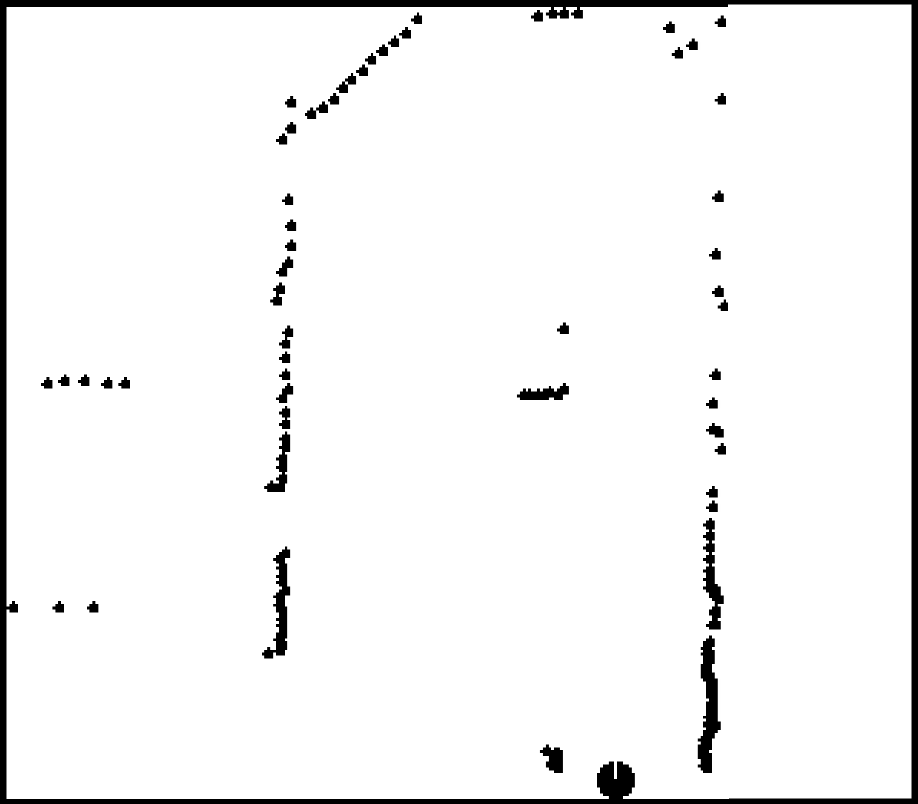
Compute likelihood of the reading

Return the product of likelihoods

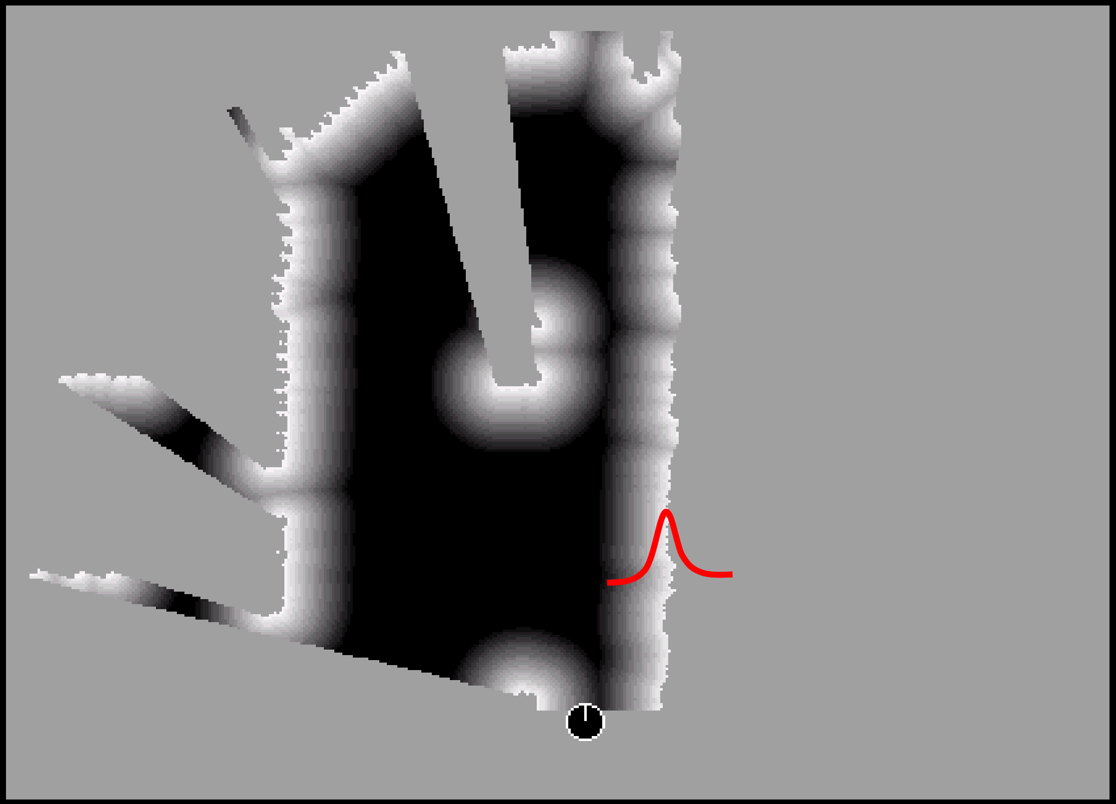
Fast Robots

Likelihood Field from Sensor Data

Sensor data projected into map

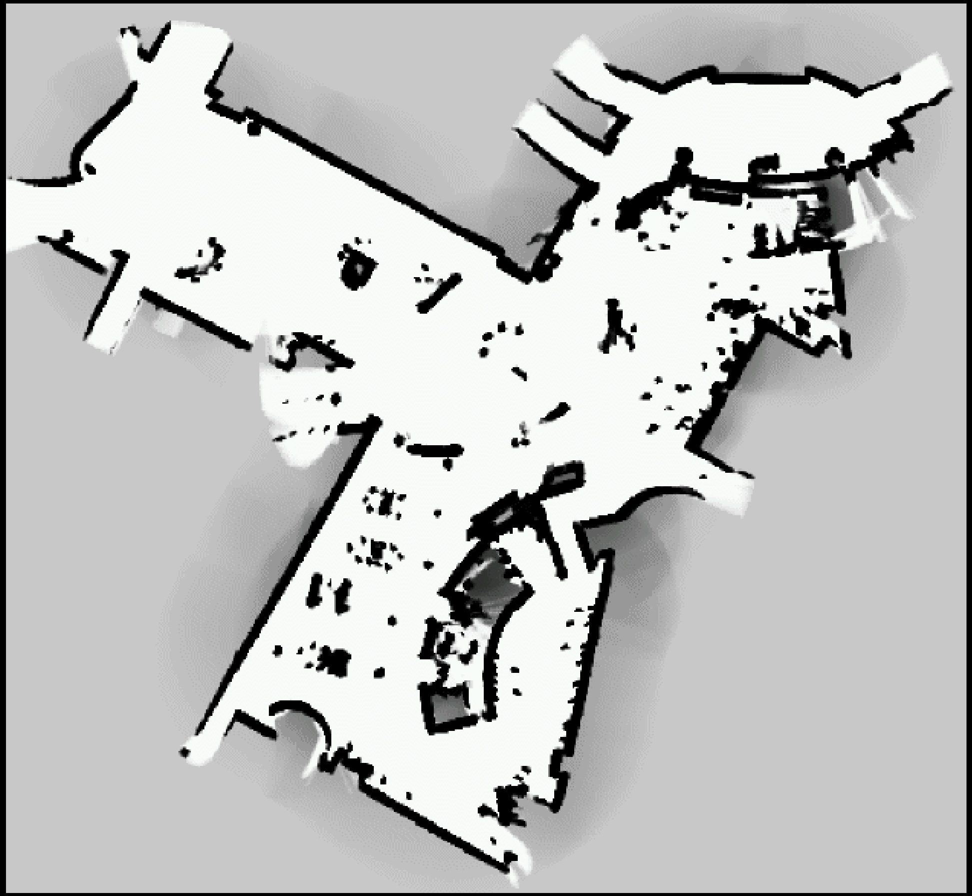


Corresponding likelihood function

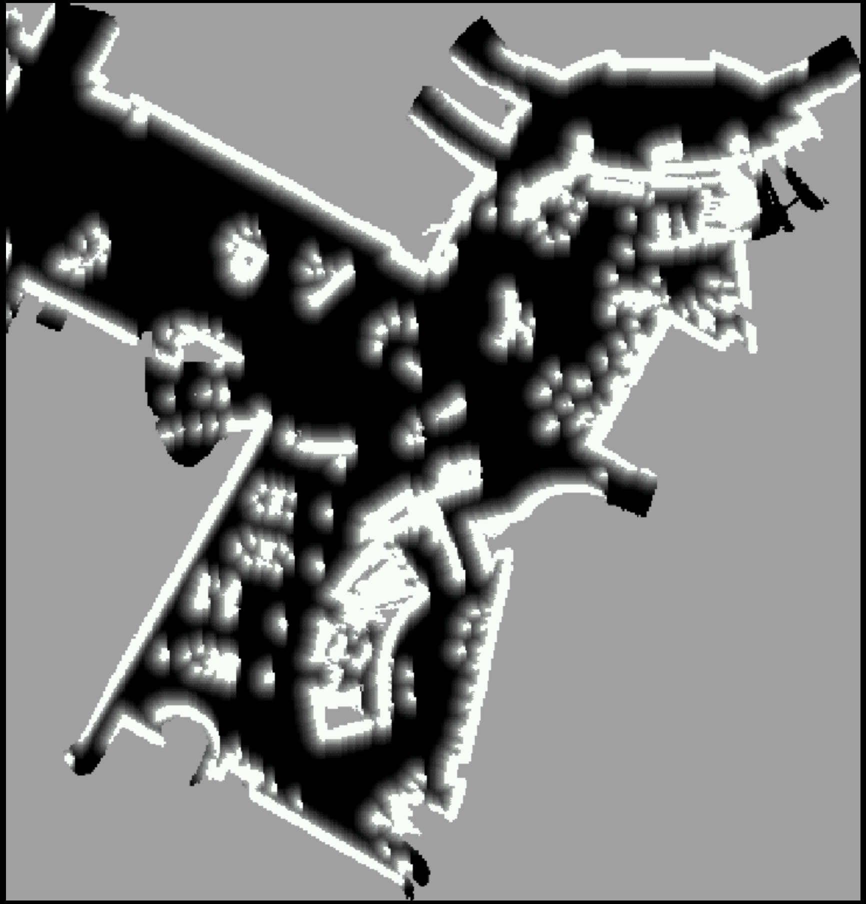


San Jose Tech Museum

Occupancy grid map



Likelihood field



Summary of Likelihood Fields

- Advantages
  - Highly efficient (computation in 2D instead of 3D)
  - Smooth w.r.t. to small changes in robot position
- Limitations
  - Does not model people and other dynamics that might cause short readings
  - Ignores physical properties of beams

Prof. Kirstin Hagelskjær Petersen
kirstin@cornell.edu

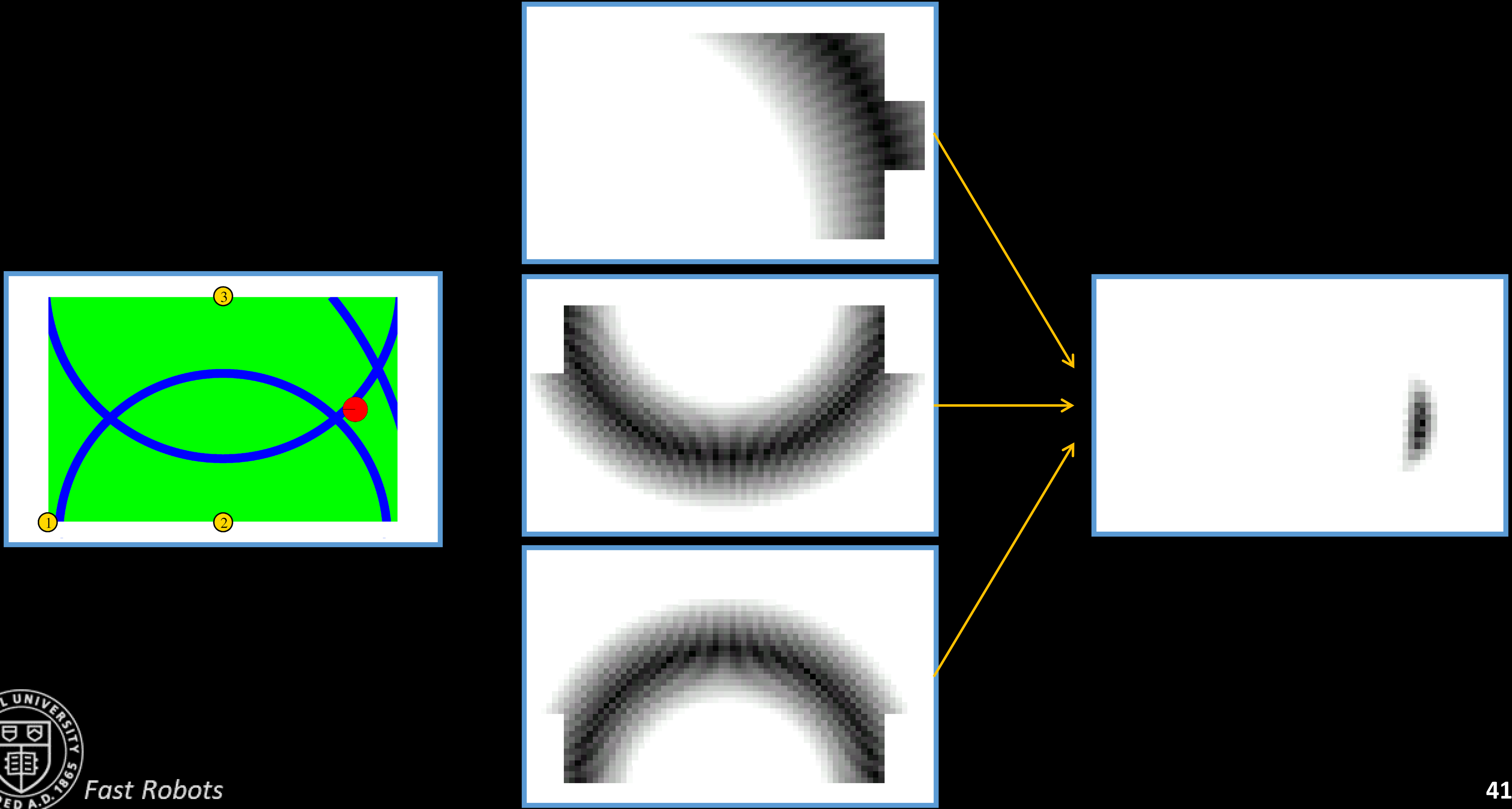
Feature Based Models

Feature Based Models

- Extract features from dense raw measurements
  - For range sensors: lines and corners
  - Often from cameras (edges, corners, distinct patterns, etc.)
- Feature extraction methods
- Features correspond to distinct physical objects in the real world and are often referred to as *landmarks*
  - Sensors output the range and/or bearing of the landmark w.r.t to the robot frame
  - Trilateration
  - Triangulation
  - Inference in the feature space can be more efficient

Trilateration using Range Measurements

Trilateration using Range Measurements



Summary of Sensor Models

- Robustness comes from explicitly modeling sensor uncertainty
- Measurement likelihood is given by "probabilistically comparing" the actual with the expected measurement
- Often, good models can be found by
    1. Determining a parametric model of noise free measurements
    2. Analyzing the sources of noise
    3. Adding adequate noise to parameters (mixed density functions)
    4. Learning (and verifying) parameters by fitting model to data
- It is extremely important to be aware of the underlying assumptions!

Reference

1. Thrun, Sebastian, Wolfram Burgard, and Dieter Fox. Probabilistic robotics. MIT press, 2005.
2. http://ais.informatik.uni-freiburg.de/teaching/ss10/robotics/slides/07-sensor-models.pdf
3. http://www.cs.cmu.edu/~16831-f14/notes/F12/16831_lecture03_mtaylormshomin.pdf
4. Gaussian Distribution: https://www.asc.ohio-state.edu/gan.1/teaching/spring04/Chapter3.pdf
5. Gaussian Distribution: http://www2.stat.duke.edu/~rcs46/modern_bayes17/lecturesModernBayes17/lecture-3/03-normal-distribution.pdf