Introduction to Mobile Robotics

Probabilistic Sensor Models

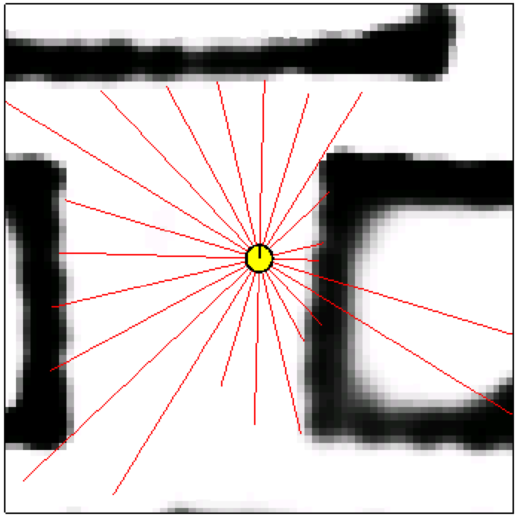
Wolfram Burgard, Diego Tipaldi,

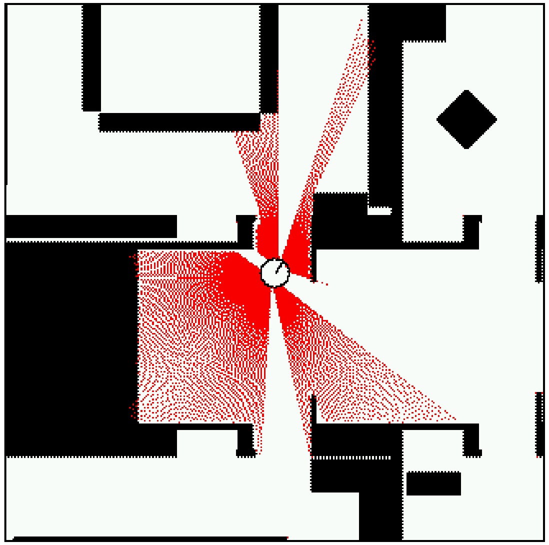
Michael Ruhnke, Bastian Steder

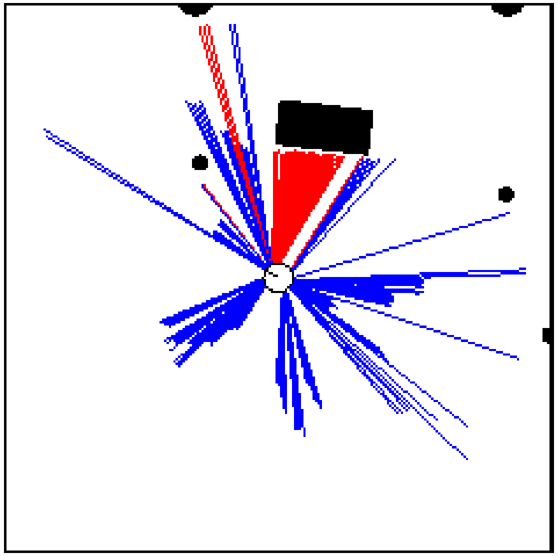
Sensors for Mobile Robots

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

Proximity Sensors







- The central task is to determine *P(z|x)*, i.e., the probability of a measurement *z* given that the robot is at position *x*.
- **Question**: Where do the probabilities come from?
- **Approach**: Let's try to explain a measurement.

Beam-based Sensor Model

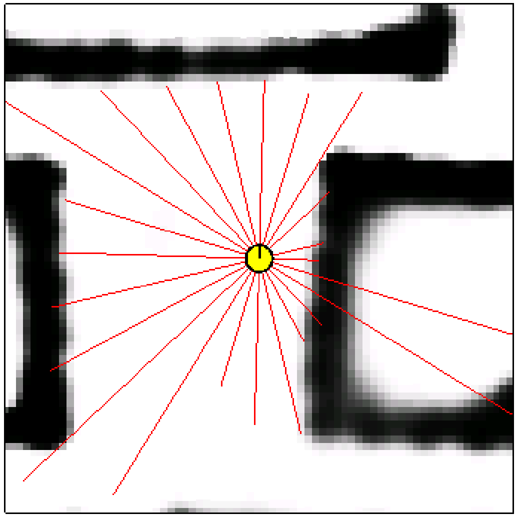
- Scan z consists of K measurements.

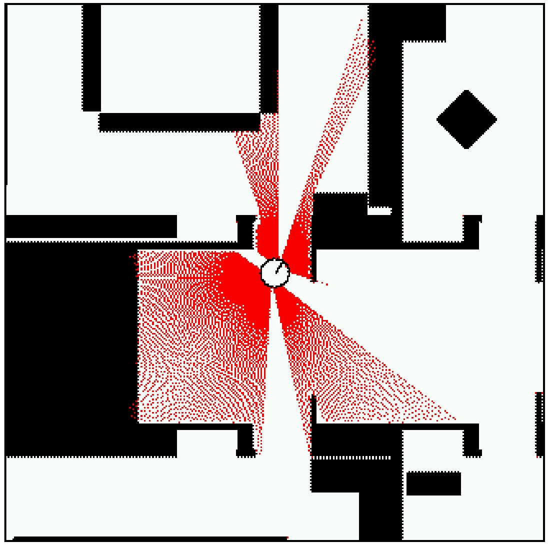
$$z = \{z_1, z_2, ..., z_K\}$$

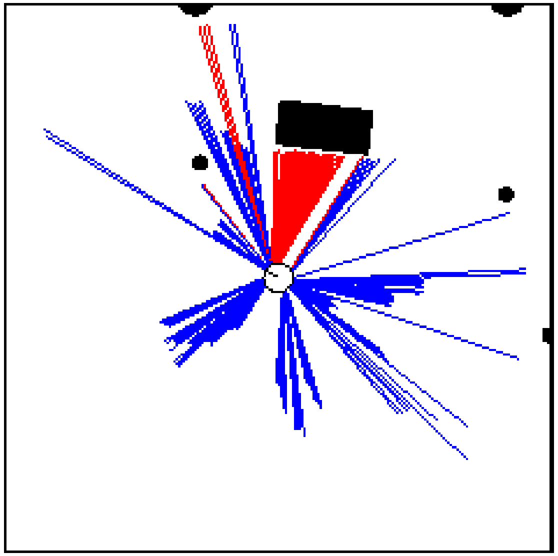
- Individual measurements are independent given the robot position.

$$P(z \mid x, m) = \prod_{k=1}^{K} P(z_k \mid x, m)$$

Beam-based Sensor Model



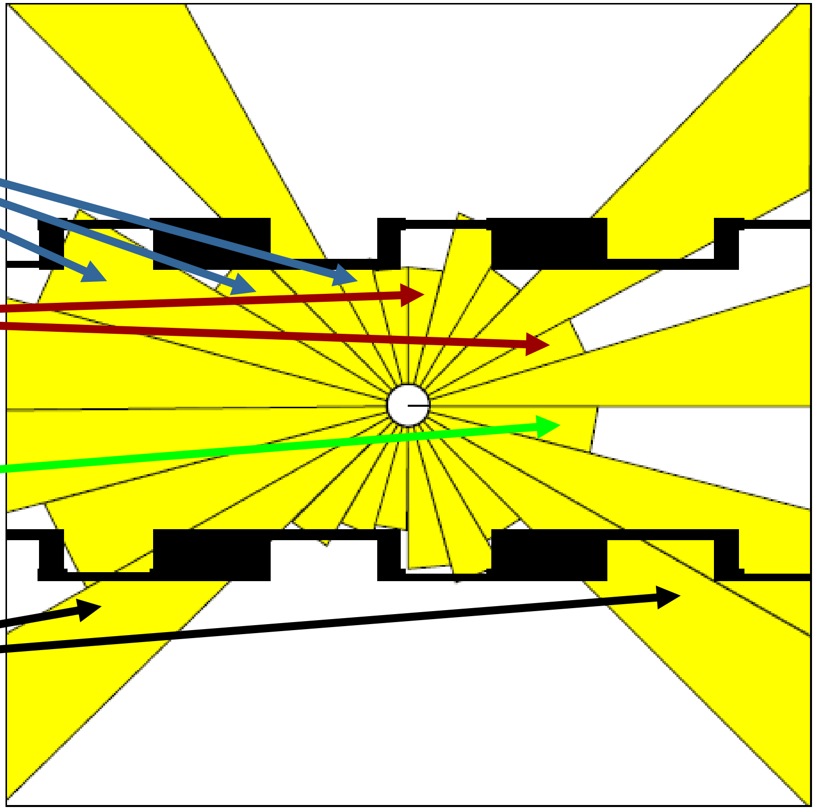




$$P(z \mid x, m) = \prod_{k=1}^{K} P(z_k \mid x, m)$$

Typical Measurement Errors of an Range Measurements

1. Beams reflected by obstacles
2. Beams reflected by persons / caused by crosstalk
3. Random measurements
4. Maximum range measurements

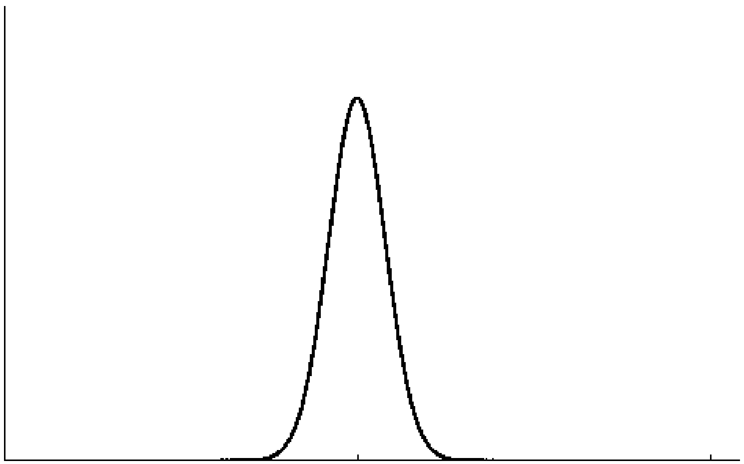


Proximity Measurement

- Measurement can be caused by …
  - a known obstacle.
  - cross-talk.
  - an unexpected obstacle (people, furniture, …).
  - missing all obstacles (total reflection, glass, …).
- Noise is due to uncertainty …
  - in measuring distance to known obstacle.
  - in position of known obstacles.
  - in position of additional obstacles.
  - whether obstacle is missed.

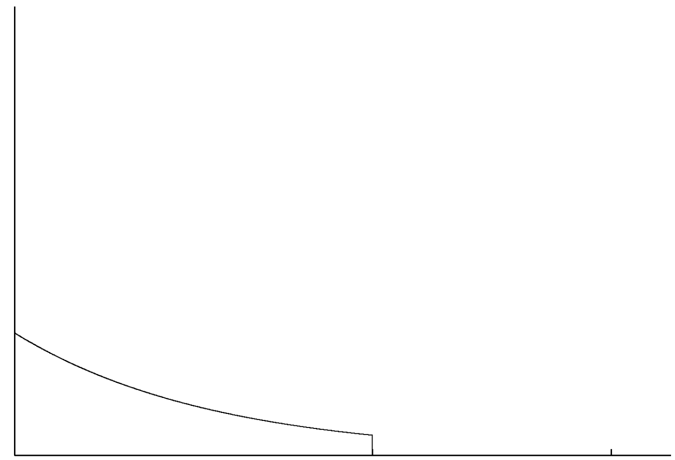
Beam-based Proximity Model

Measurement noise



$$P_{hit}(z \mid x,m) = \eta \frac{1}{\sqrt{2\pi b}} e^{-\frac{1}{2}\frac{(z-z_{exp})^2}{b}}$$

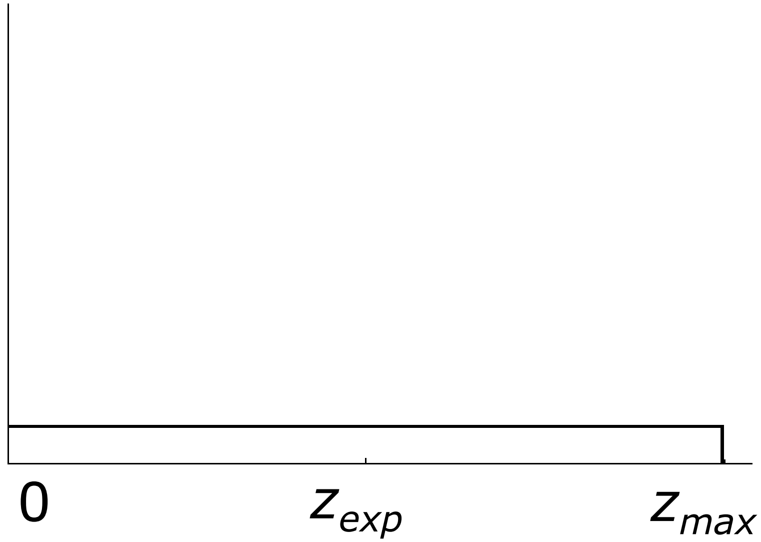
Unexpected obstacles



$$P_{unexp}(z \mid x,m) = \begin{cases} \eta\ \lambda\ e^{-\lambda z} & z < z_{exp} \\ 0 & otherwise \end{cases}$$

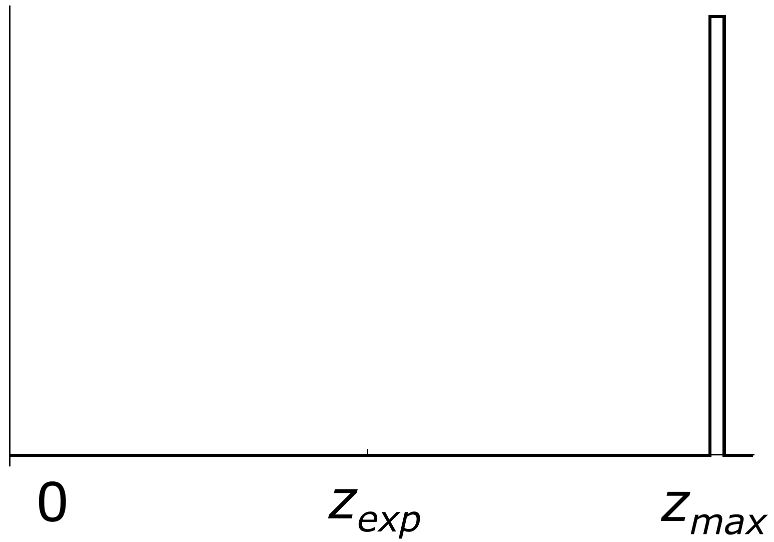
Beam-based Proximity Model

Random measurement



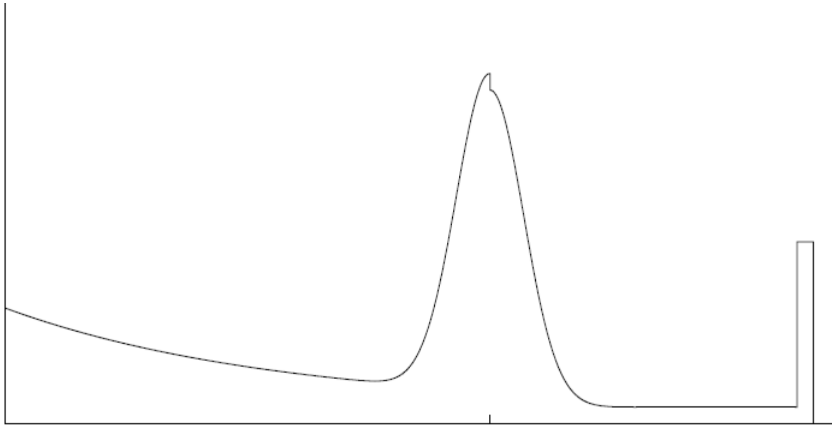
$$P_{rand}(z \mid x, m) = \eta \frac{1}{z_{\max}}$$

Max range



$$P_{\max}(z \mid x, m) = \eta \frac{1}{z_{small}}$$

Resulting Mixture Density

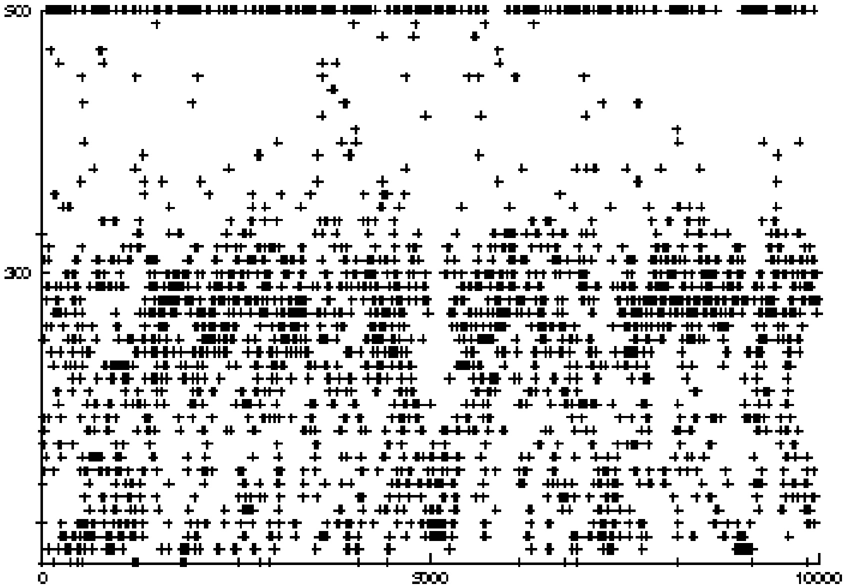


$$P(z \mid x, m) = \begin{pmatrix} \alpha_{\text{hit}} \\ \alpha_{\text{unexp}} \\ \alpha_{\max} \\ \alpha_{\text{rand}} \end{pmatrix}^T \cdot \begin{pmatrix} P_{\text{hit}}(z \mid x, m) \\ P_{\text{unexp}}(z \mid x, m) \\ P_{\max}(z \mid x, m) \\ P_{\text{rand}}(z \mid x, m) \end{pmatrix}$$

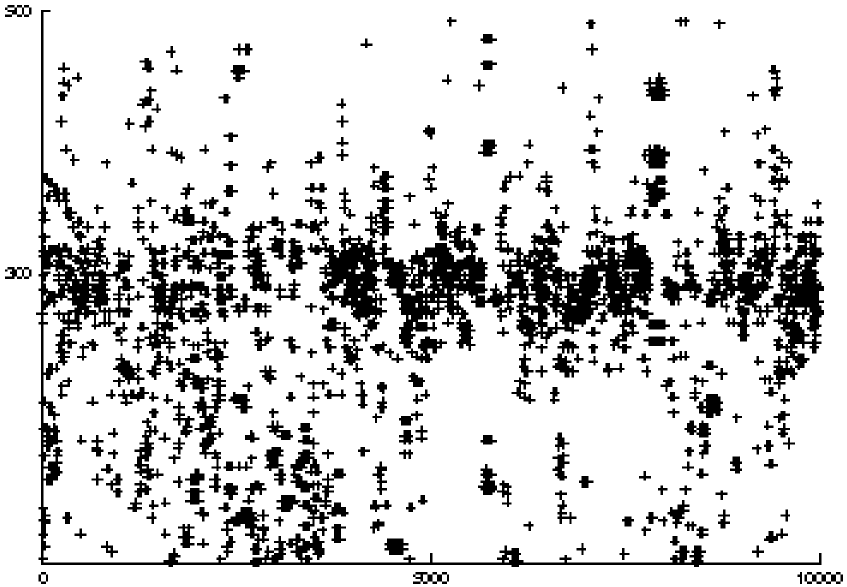
How can we determine the model parameters?

Raw Sensor Data

Measured distances for expected distance of 300 cm.



Sonar



Laser

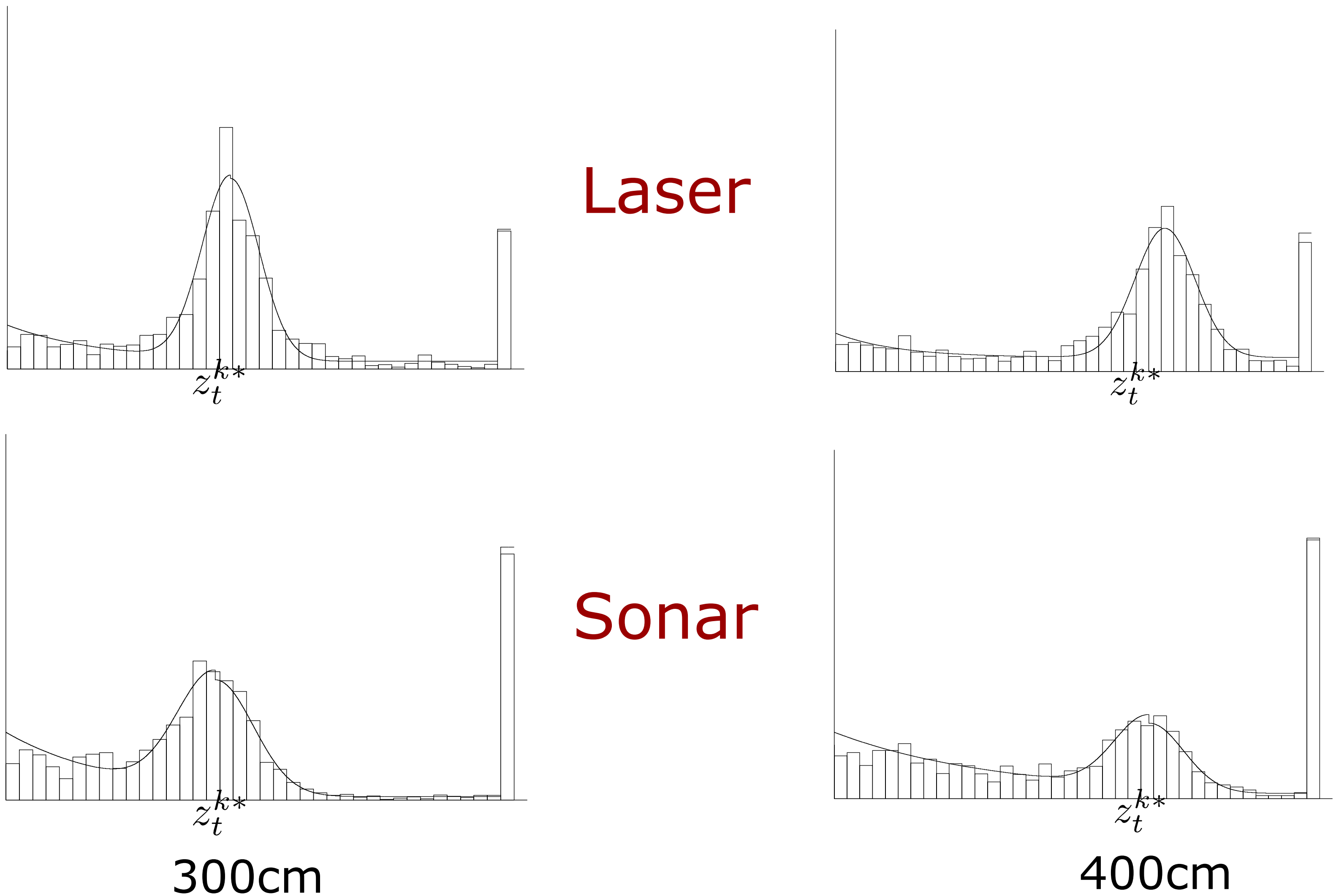
Approximation

- Maximize log likelihood of the data

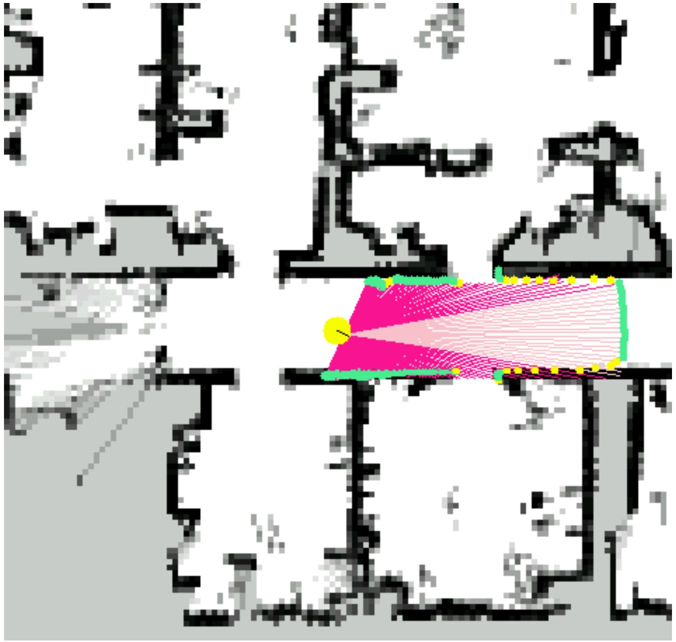
$$P(z \mid z_{\exp})$$

- Search space of n-1 parameters.
  - Hill climbing
  - Gradient descent
  - Genetic algorithms
  - …
- Deterministically compute the n-th parameter to satisfy normalization constraint.

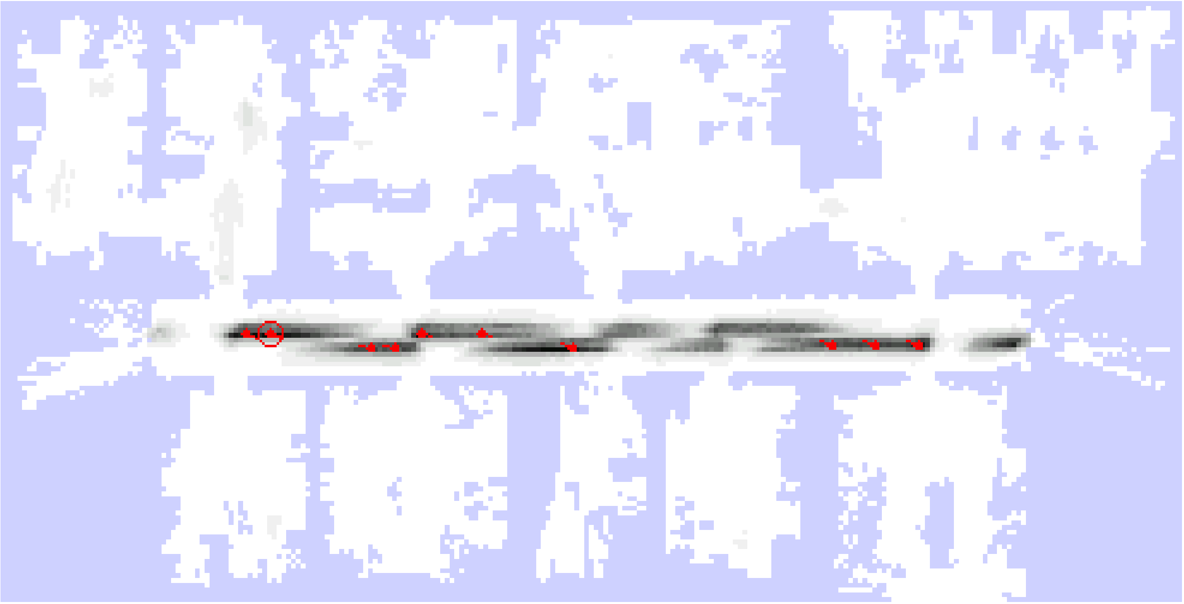
Approximation Results



Example



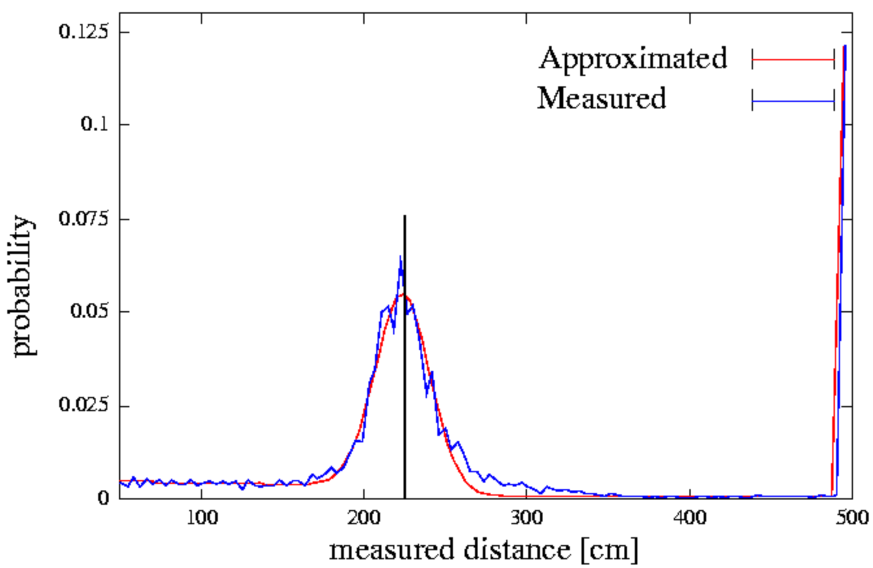
z



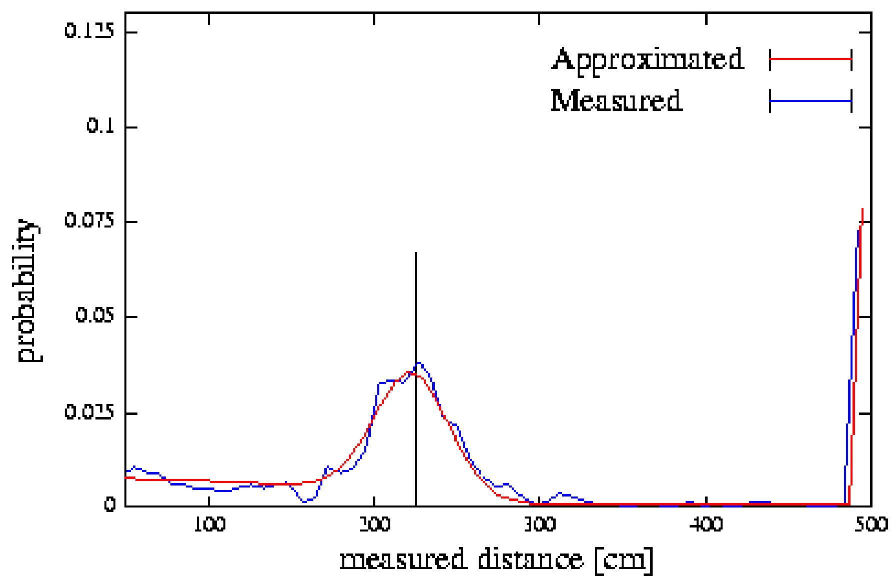
$P(z|x,m)$

Discrete Model of Proximity Sensors

- Instead of densities, consider discrete steps along the sensor beam.

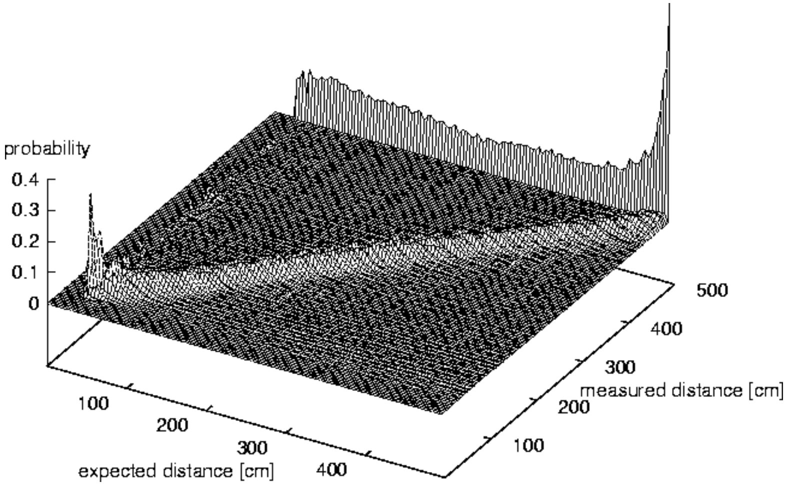


Laser sensor

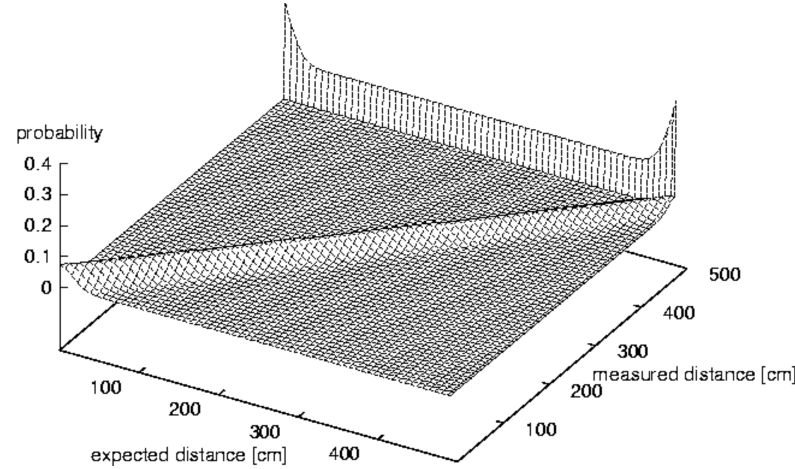


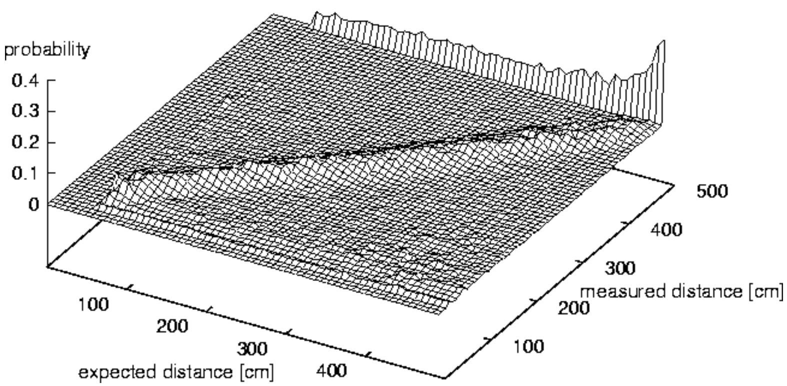
Sonar sensor

Approximation Results

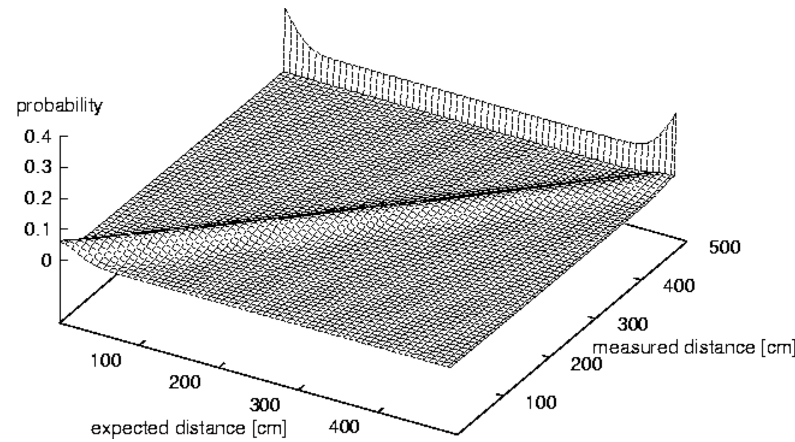


Laser

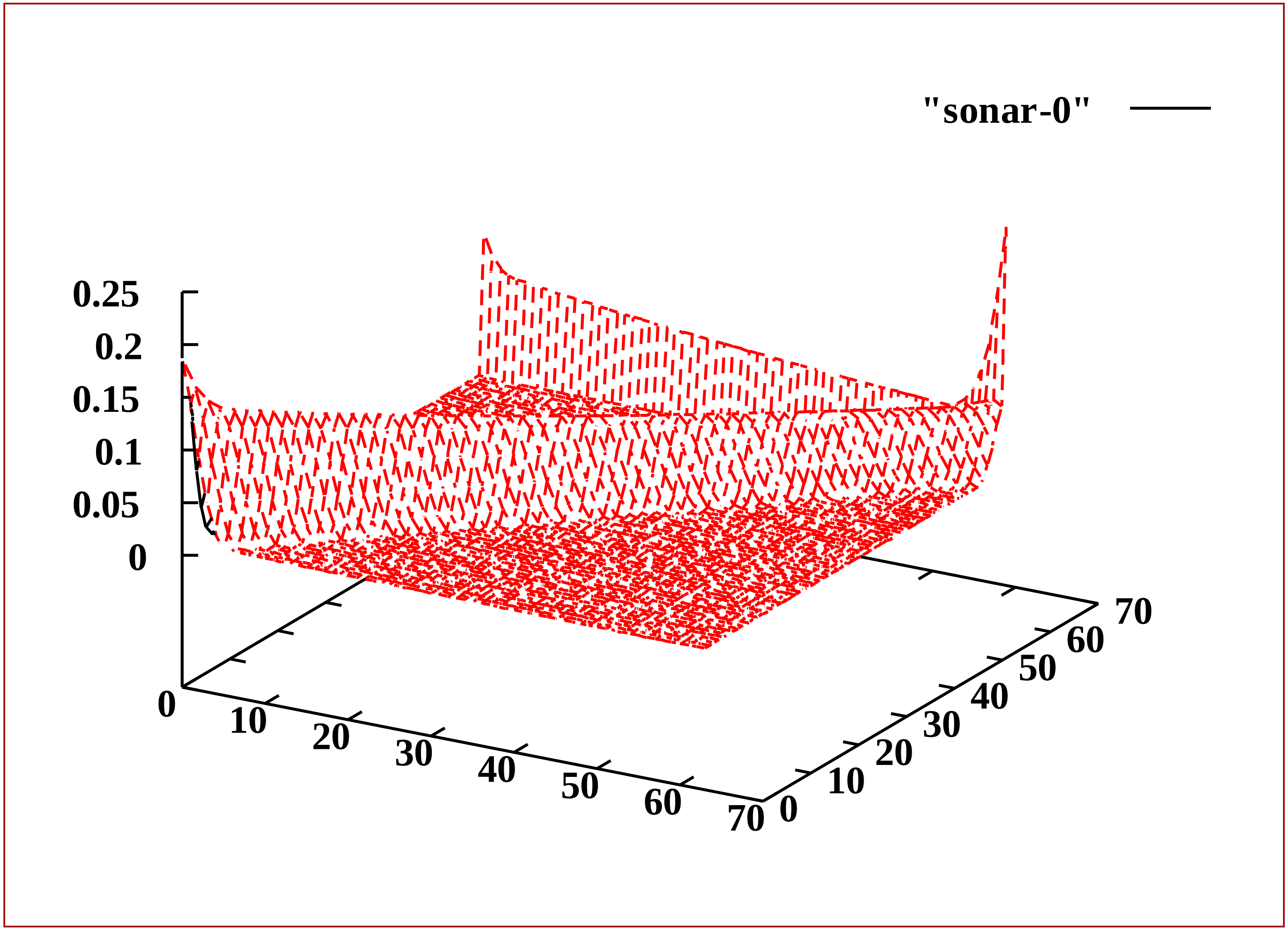




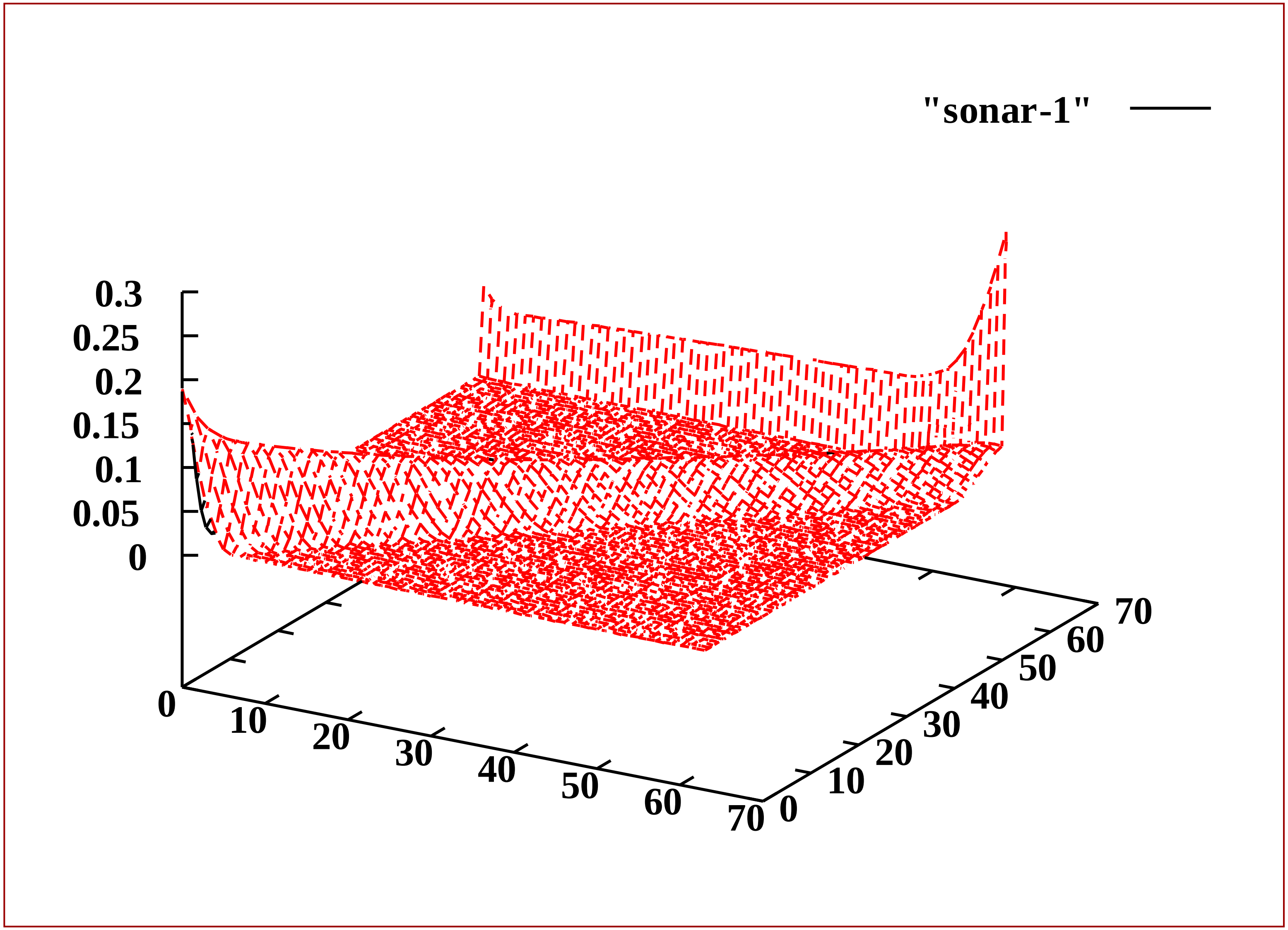
Sonar



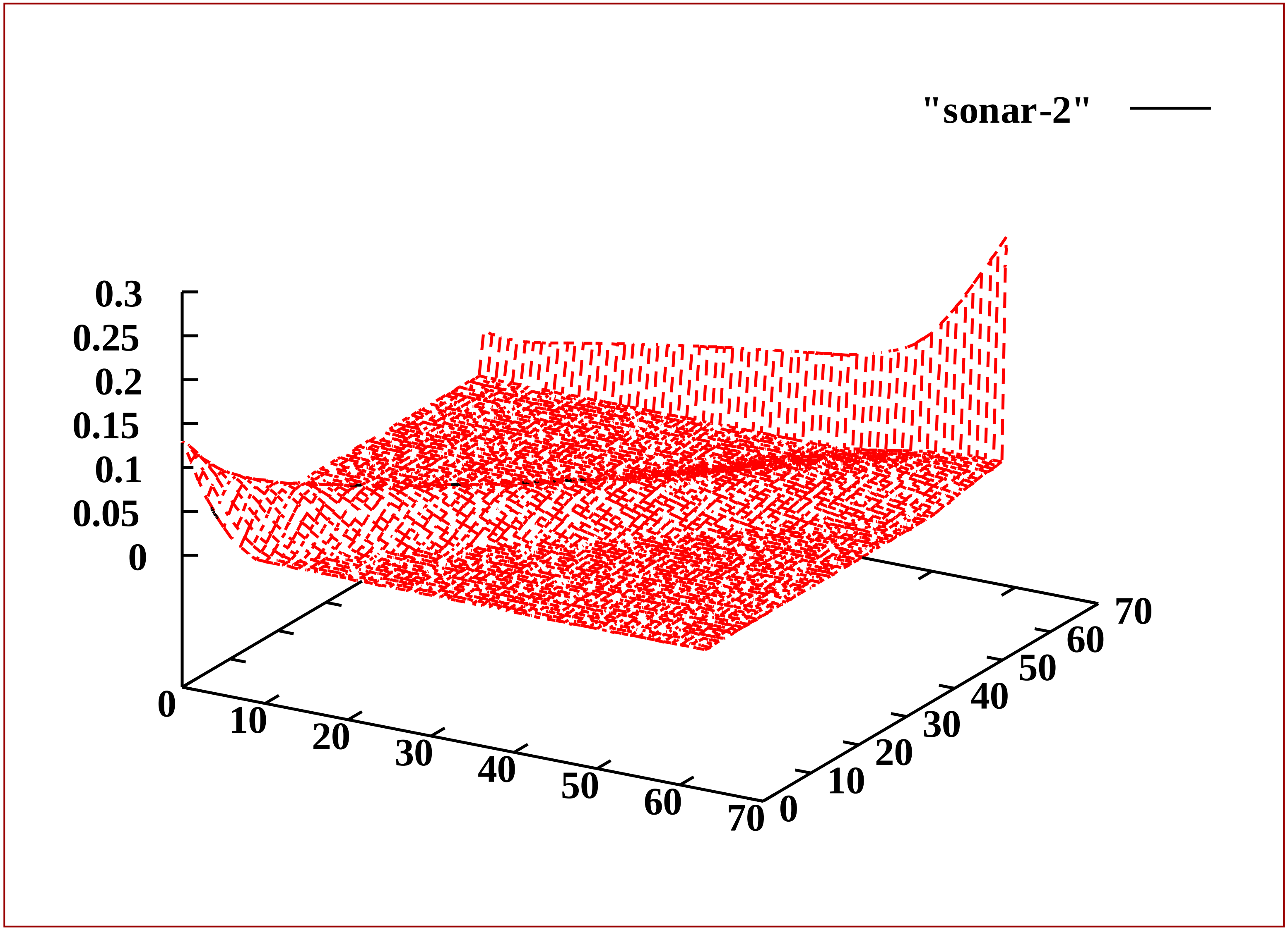
Influence of Angle to Obstacle



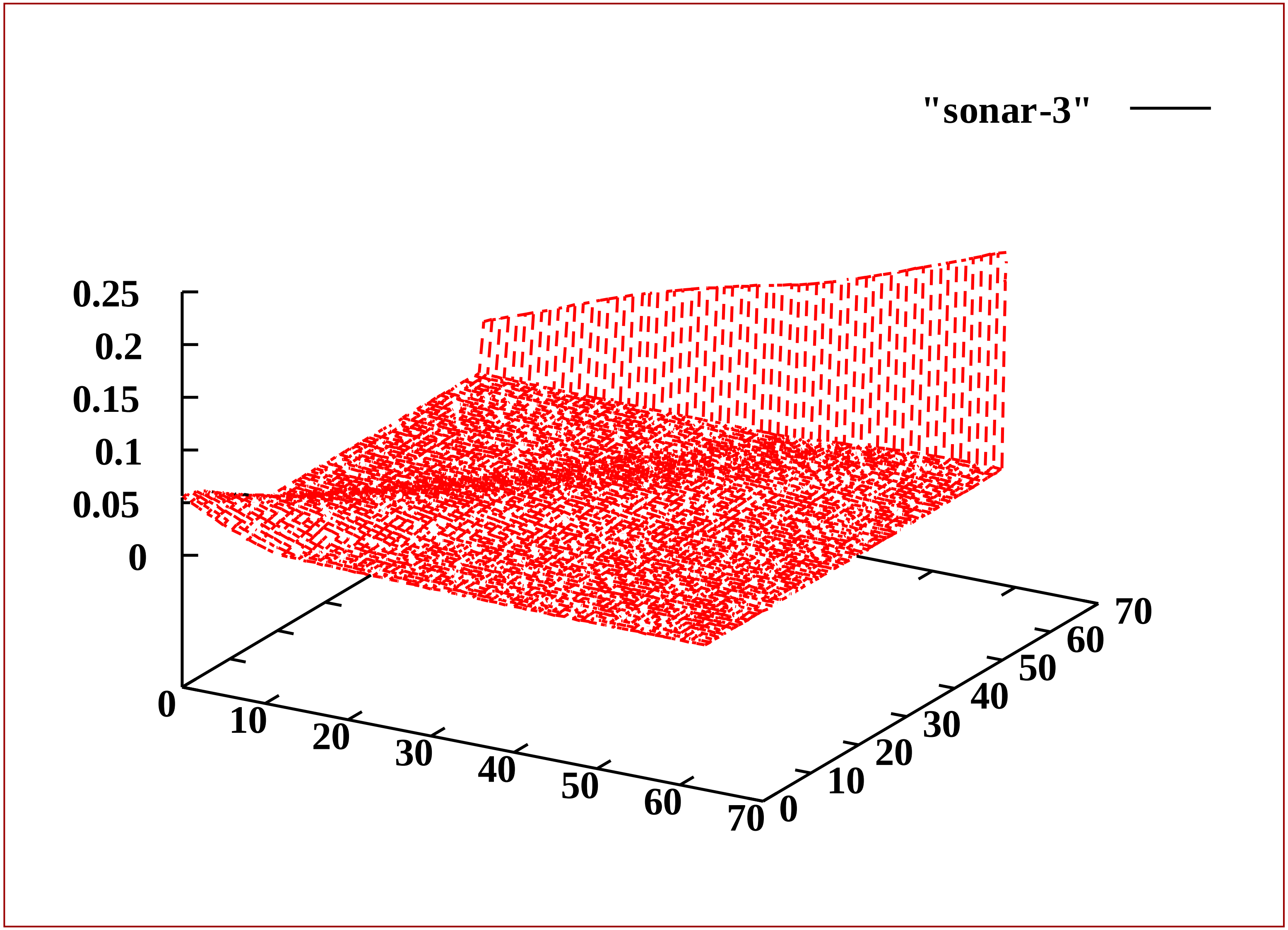
Influence of Angle to Obstacle



Influence of Angle to Obstacle



Influence of Angle to Obstacle



Summary Beam-based Model

- Assumes independence between beams.
  - Justification?
  - Overconfident!
- Models physical causes for measurements.
  - Mixture of densities for these causes.
  - Assumes independence between causes. Problem?
- Implementation
  - Learn parameters based on real data.
  - Different models should be learned for different angles at which the sensor beam hits the obstacle.
  - Determine expected distances by ray-tracing.
  - Expected distances can be pre-processed.

Scan-based Model

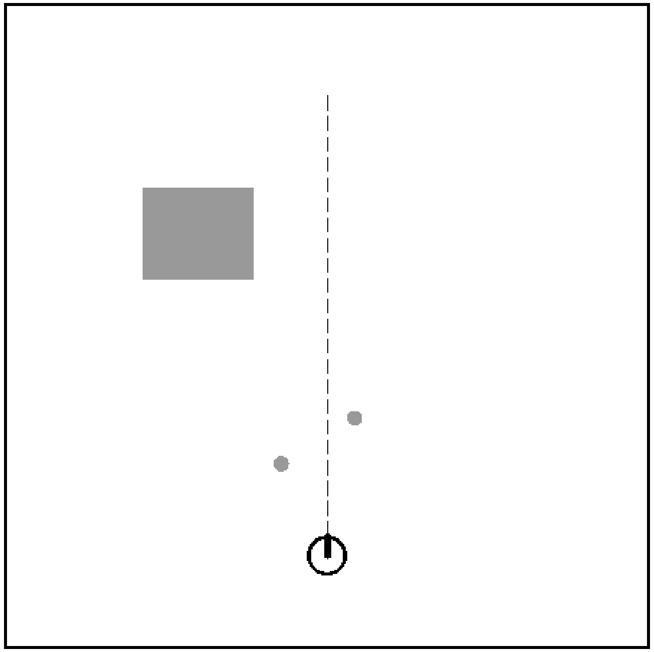
- Beam-based model is …
  - not smooth for small obstacles and at edges.
  - not very efficient.

- Idea: Instead of following along the beam, just check the end point.

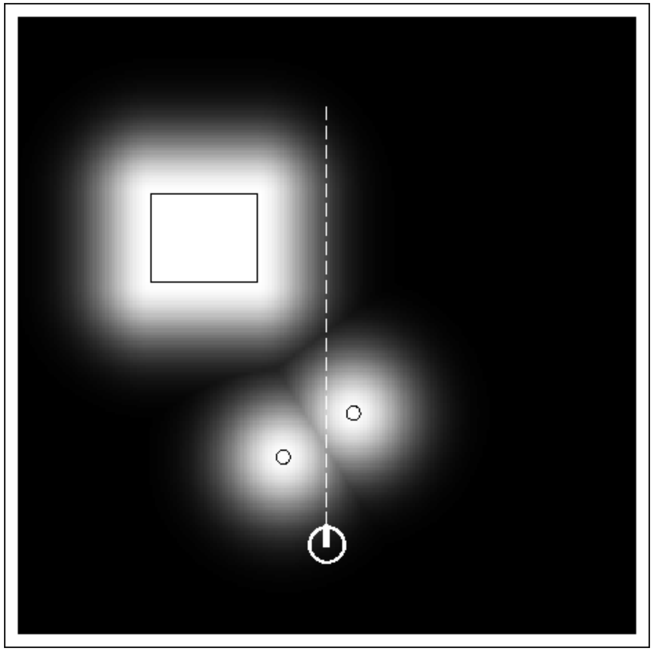
Scan-based Model

- Probability is a mixture of …
  - a Gaussian distribution with mean at distance to closest obstacle,
  - a uniform distribution for random measurements, and
  - a small uniform distribution for max range measurements.
- Again, independence between different components is assumed.

Example

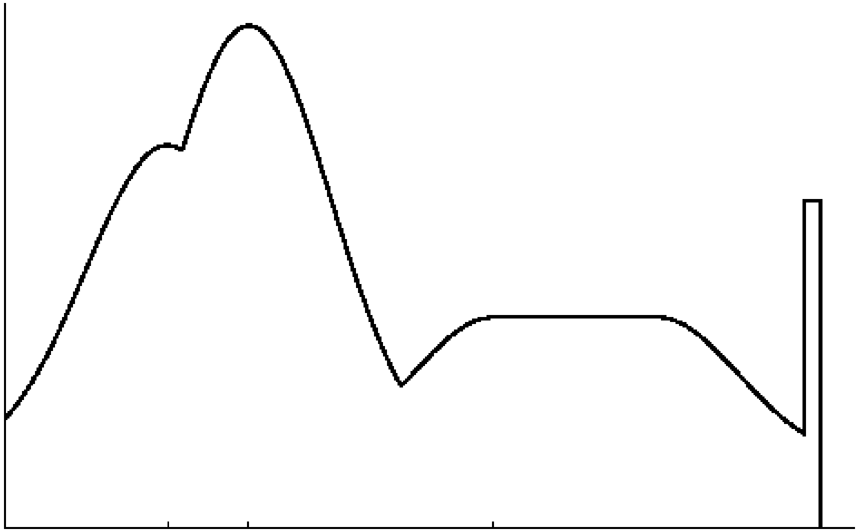


Map *m*

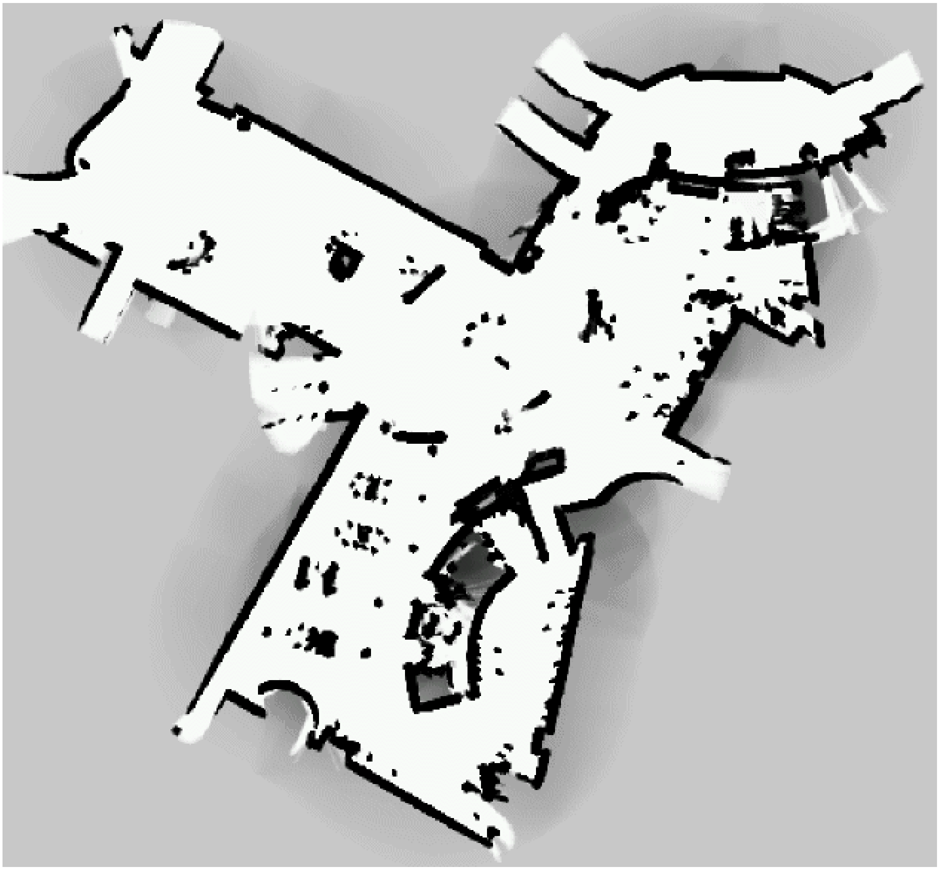


Likelihood field

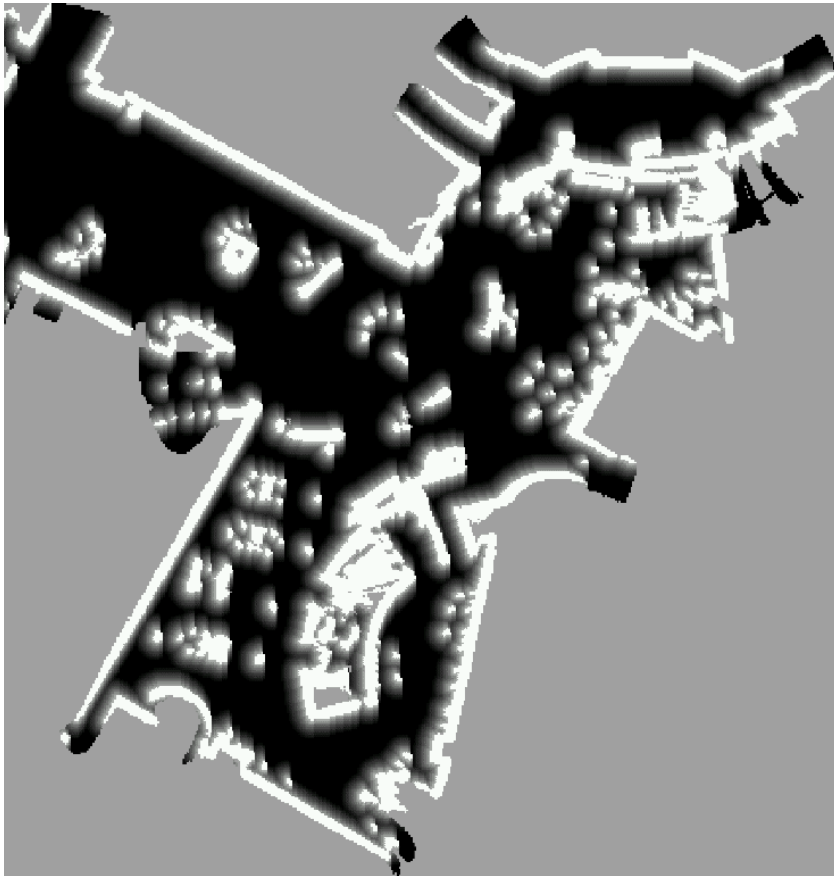
$P(z|x,m)$



San Jose Tech Museum



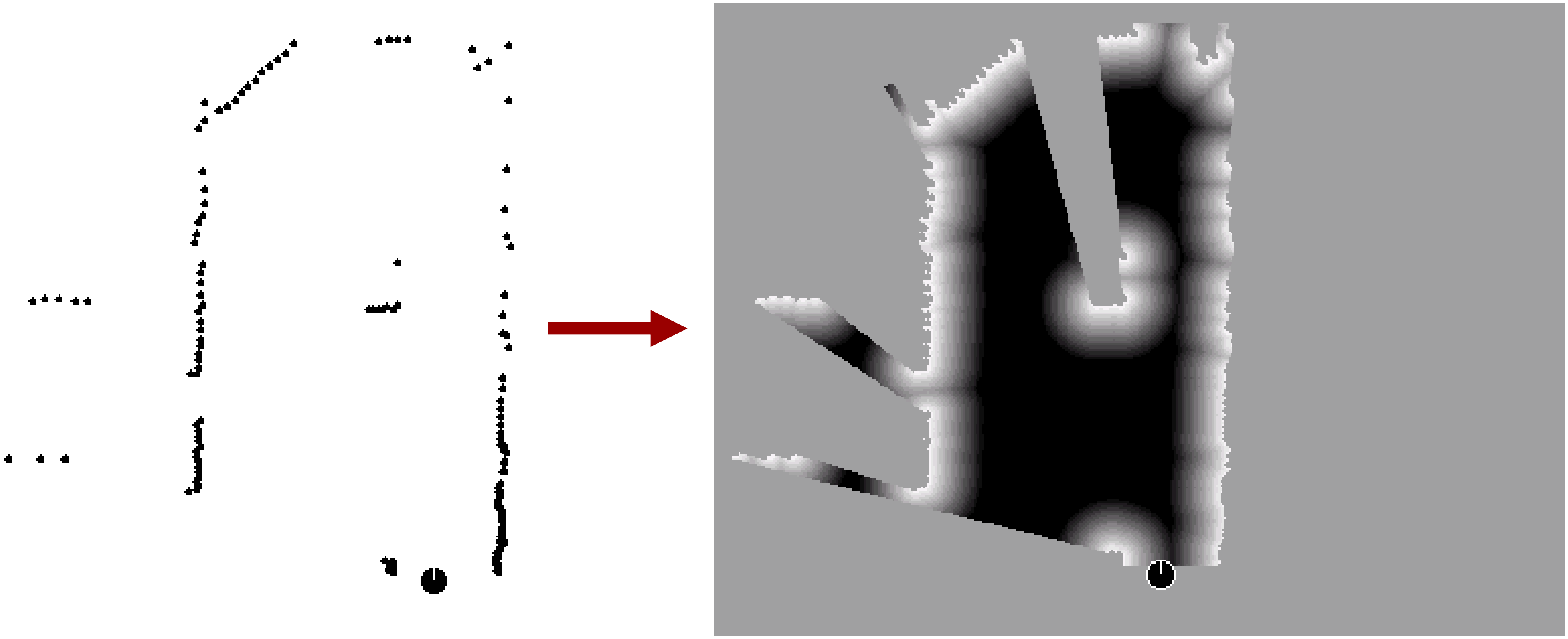
Occupancy grid map



Likelihood field

Scan Matching

- Extract likelihood field from scan and use it to match different scan.



Properties of Scan-based Model

- Highly efficient, uses 2D tables only.
- Distance grid is smooth w.r.t. to small changes in robot position.
- Allows gradient descent, scan matching.
- Ignores physical properties of beams.

Additional Models of Proximity Sensors

- Map matching (sonar, laser): generate small, local maps from sensor data and match local maps against global model.
- Scan matching (laser): map is represented by scan endpoints, match scan into this map.
- Features (sonar, laser, vision): Extract features such as doors, hallways from sensor data.

Landmarks

- Active beacons (*e.g.*, radio, GPS)
- Passive (*e.g.*, visual, retro-reflective)
- Standard approach is triangulation

- Sensor provides
  - distance, or
  - bearing, or
  - distance and bearing.

Distance and Bearing

Probabilistic Model

1. Algorithm **landmark_detection_model**(z,x,m):

   $z = \langle i, d, \alpha \rangle, x = \langle x, y, \theta \rangle$

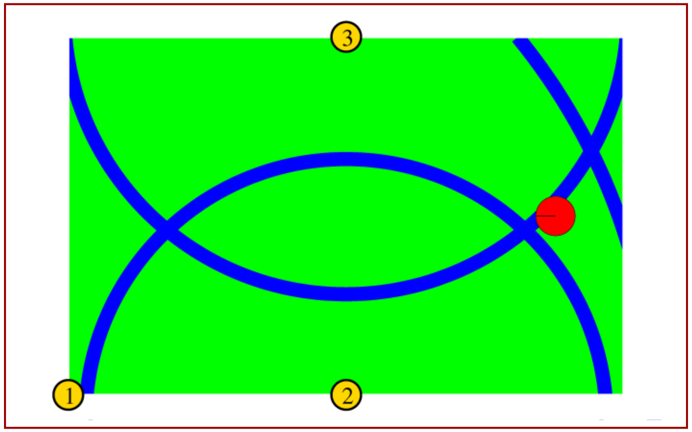
2. $\hat{d} = \sqrt{(m_x(i) - x)^2 + (m_y(i) - y)^2}$

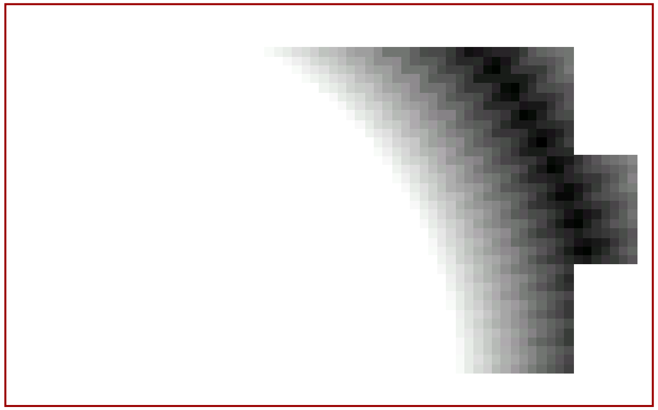
3. $\hat{\alpha} = \text{atan2}(m_y(i) - y, m_x(i) - x) - \theta$

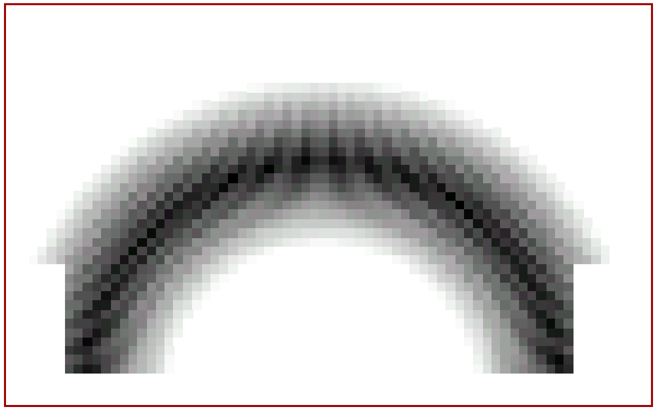
4. $p_{\text{det}} = \text{prob}(\hat{d} - d, \varepsilon_d) \cdot \text{prob}(\hat{\alpha} - \alpha, \varepsilon_\alpha)$

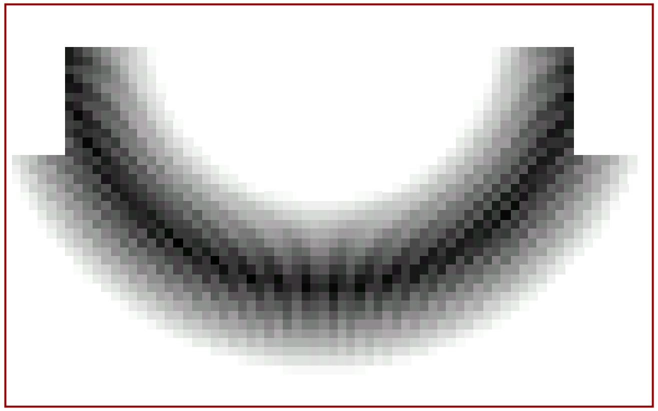
5. Return p_{det}

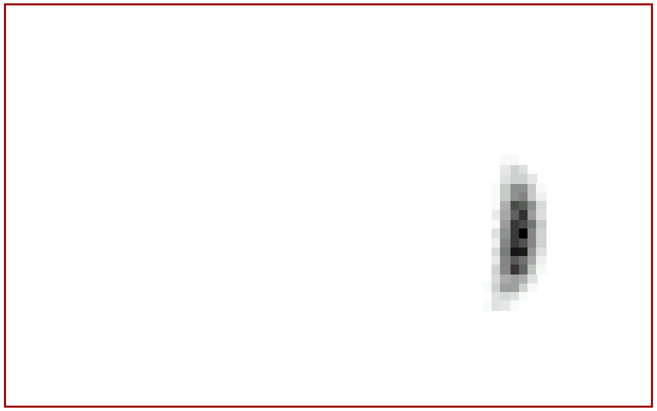
Distributions







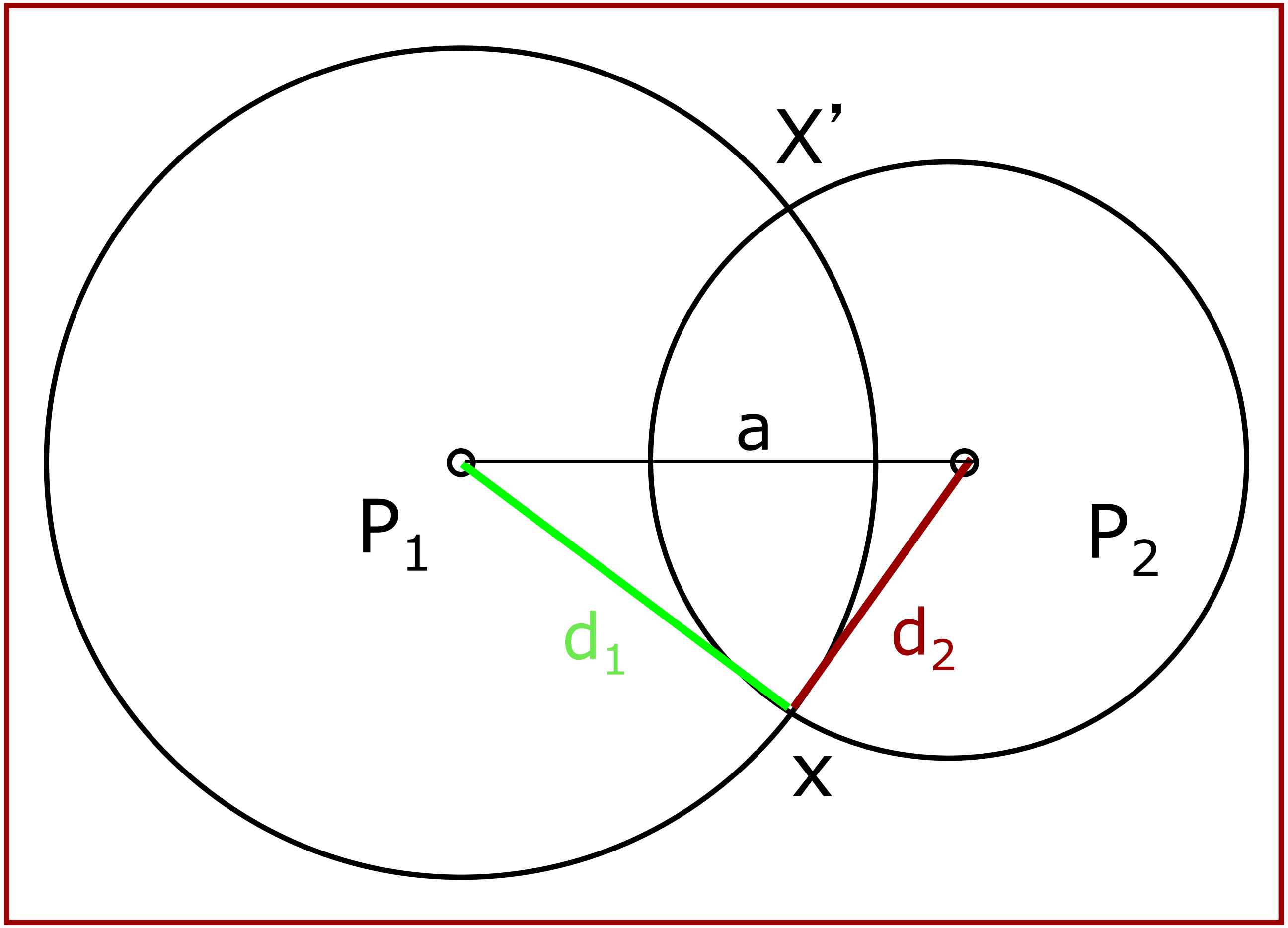




Distances Only
No Uncertainty

$$x = (a^2 + d_1^2 - d_2^2) / 2a$$

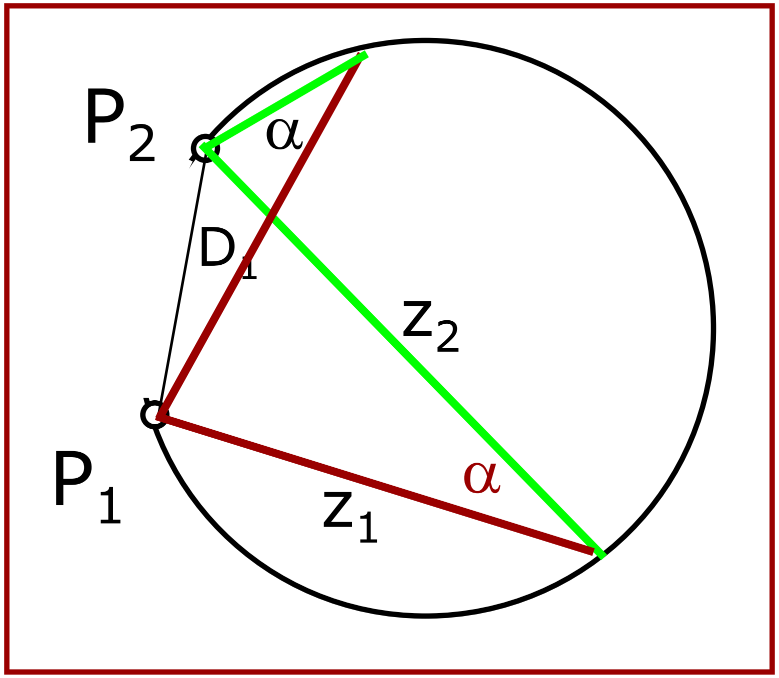
$$y = \pm\sqrt{(d_1^2 - x^2)}$$

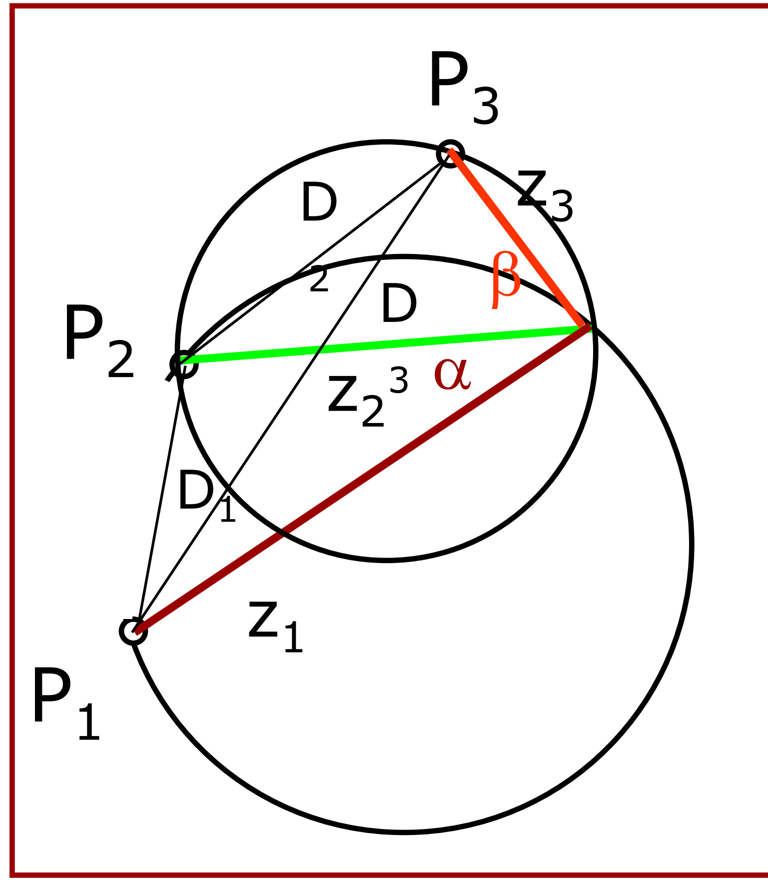


P_1=(0,0)

P_2=(a,0)

Bearings Only
No Uncertainty





Law of cosine

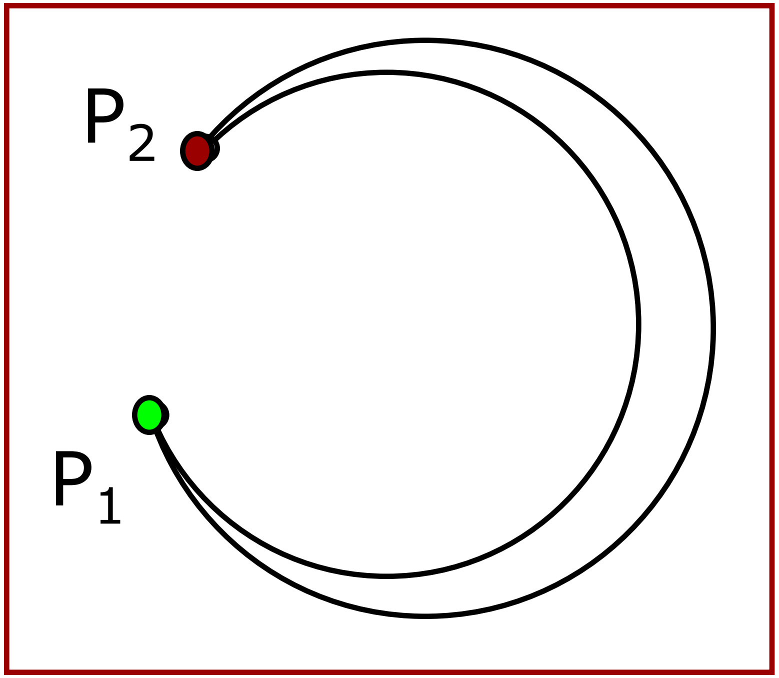
$$D_1^2 = z_1^2 + z_2^2 - 2\, z_1 z_2 \cos\alpha$$

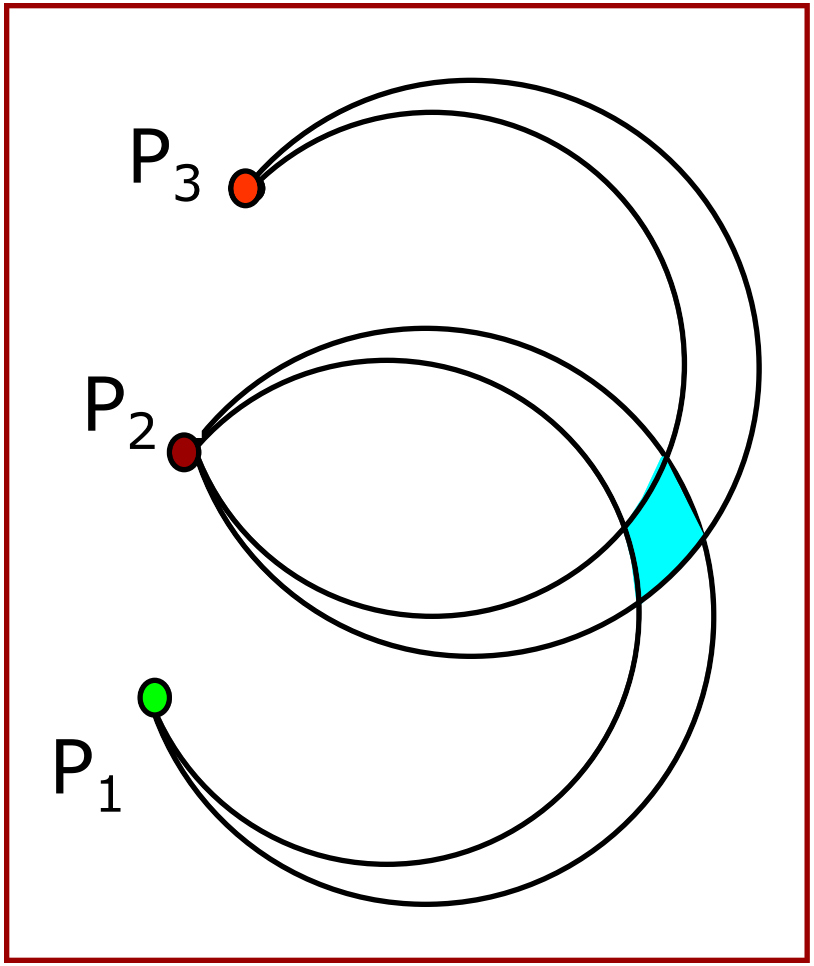
$$D_1^2 = z_1^2 + z_2^2 - 2\, z_1 z_2 \cos(\alpha)$$

$$D_2^2 = z_2^2 + z_3^2 - 2\, z_1 z_2 \cos(\beta)$$

$$D_3^2 = z_1^2 + z_3^2 - 2\, z_1 z_2 \cos(\alpha + \beta)$$

Bearings Only With Uncertainty





Most approaches attempt to find estimation mean.

Summary of Sensor Models

- Explicitly modeling uncertainty in sensing is key to robustness.
- In many cases, good models can be found by the following approach:
  1. Determine parametric model of noise free measurement.
  2. Analyze sources of noise.
  3. Add adequate noise to parameters (eventually mix in densities for noise).
  4. Learn (and verify) parameters by fitting model to data.
  5. Likelihood of measurement is given by "probabilistically comparing" the actual with the expected measurement.
- This holds for motion models as well.
- It is extremely important to be aware of the underlying assumptions!