Cognitive Robotics

05. Probabilistic Sensor Models

AbdElMoniem Bayoumi, PhD

Fall 2022

Acknowledgment

- These slides have been created by Wolfram Burgard, Dieter Fox, Cyrill Stachniss and Maren Bennewitz

State Estimation

- Estimate the state x of a system given observations z and actions u
- **Goal:** Determine $p(x \mid z, u)$

Recursive Bayes Filter (recap)

$$
\begin{aligned}
bel(x_t) &= p(x_t \mid z_{1:t}, u_{1:t}) \\
&= p(z_t \mid x_t, z_{1:t-1}, u_{1:t})\, p(x_t \mid z_{1:t-1}, u_{1:t}) \,/\, p(z_t \mid z_{1:t-1}, u_{1:t}) \\
&= \eta\, p(z_t \mid x_t)\, p(x_t \mid z_{1:t-1}, u_{1:t}) \\
&= \eta\, p(z_t \mid x_t) \int p(x_t \mid x_{t-1}, z_{1:t-1}, u_{1:t}) \\
&\qquad\qquad p(x_{t-1} \mid z_{1:t-1}, u_{1:t})\, dx_{t-1} \\
&= \eta\, p(z_t \mid x_t) \int p(x_t \mid x_{t-1}, u_t)\, p(x_{t-1} \mid z_{1:t-1}, u_{1:t})\, dx_{t-1} \\
&= \eta\, p(z_t \mid x_t) \int p(x_t \mid x_{t-1}, u_t)\, p(x_{t-1} \mid z_{1:t-1}, u_{1:t-1})\, dx_{t-1} \\
&= \underbrace{\eta\, p(z_t \mid x_t)}_{\text{observation model}} \int \underbrace{p(x_t \mid x_{t-1}, u_t)}_{\text{motion model}}\, \underbrace{bel(x_{t-1})}_{\text{recursive term}}\, dx_{t-1}
\end{aligned}
$$

bnshof eh e7tmal en el z de tetl3 given en el state xt 7asal.

lw hya likely, fa handeha kema kbera, lakn lw l 3ks kemtha htb2a olyla, w da by5lene a2dr a3ml correction lel model bta3y,

Motion and Observation Model

- Prediction step

$$\overline{bel}(x_t) = \int \underline{p(x_t \mid u_t, x_{t-1})}\, bel(x_{t-1})\, dx_{t-1}$$

motion model

- Correction step

$$bel(x_t) = \eta\, \underline{p(z_t \mid x_t)}\, \overline{bel}(x_t)$$

sensor or observation model

Previous Lecture: Probabilistic Motion Models

- Robots execute motion commands only inaccurately
- The motion model specifies the probability that action u_t carries the robot from pose x_{t-1} to x_t:

$$p(x_t \mid u_t, x_{t-1})$$

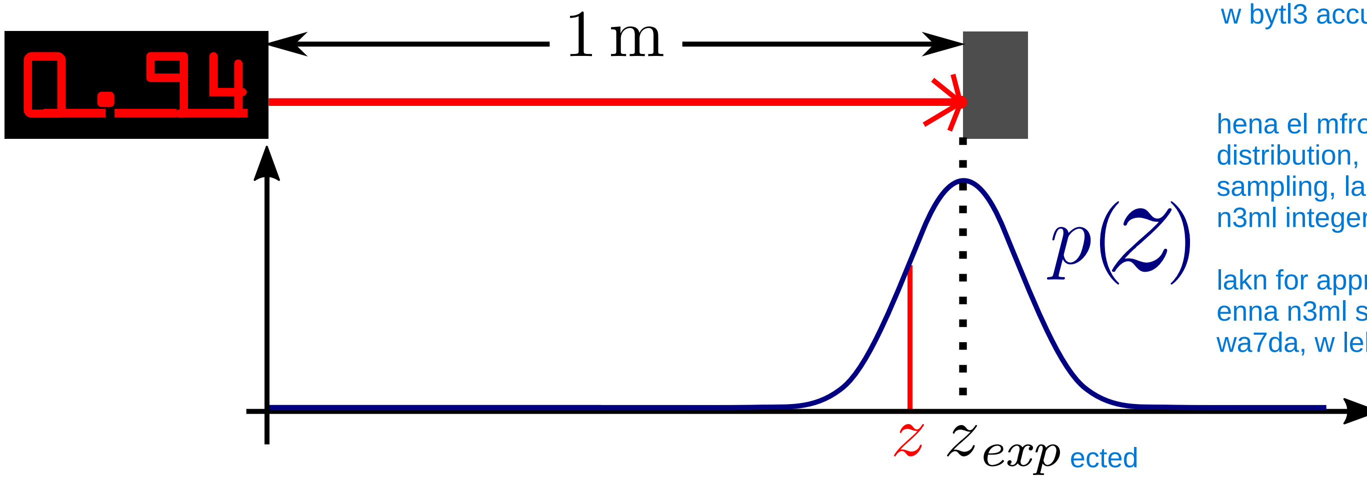
- Defined individually for each type or robot

Measurement Probability

el errors fl sensors, dayman btb2a a2al bkter gedan mn el errors fl actuators.

7ata m3 el sensors el t3bana, brdu el filters btb2a shaghaala w bytl3 accuracy kwysa,

el sensor hena ara 0.93 badal el 1m, 34an byb2a feh noise.



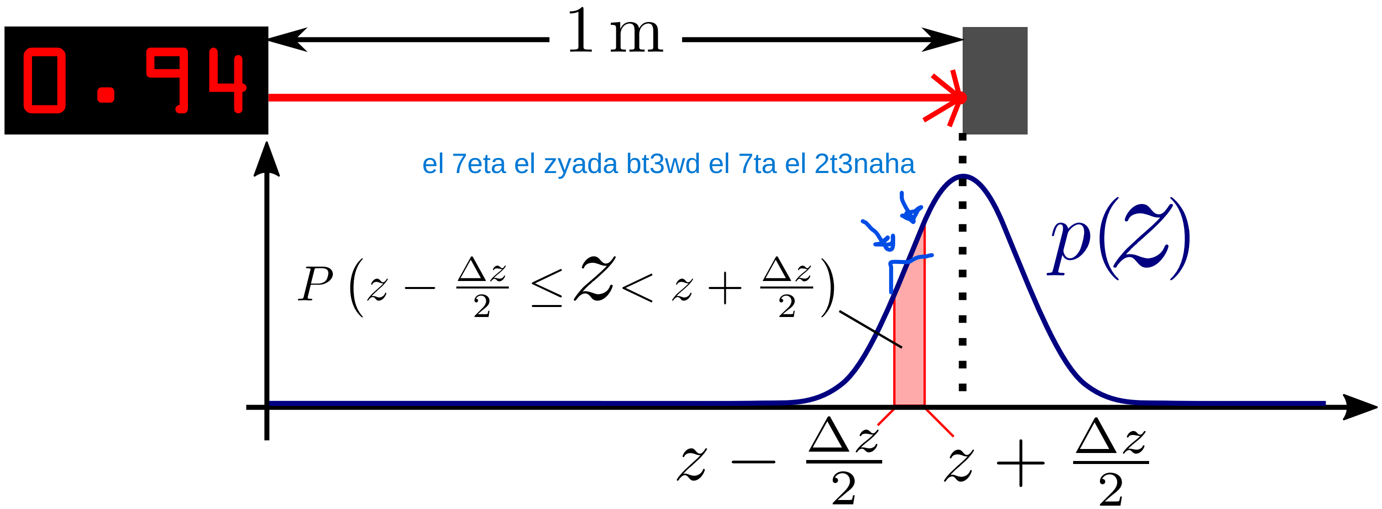
hena el mfrod da continous distribution, fa el mfrod mn3mlsh sampling, laa nakhud range w n3ml integeration 3leh.

lakn for approximation, btmshy enna n3ml sampling 3la point wa7da, w leha proof.

bn3br 3n el noise b gauss distribution.

$$
\begin{aligned}
P(Z = z) &= 0 \\
&= P(Z = 0.9400000\ldots)
\end{aligned}
$$

Measurement Probability



$$P(0.935 \leq Z < 0.945) = P\left(z-\frac{\Delta z}{2} \leq Z < z+\frac{\Delta z}{2}\right)$$

$$= \int_{z-\frac{\Delta z}{2}}^{z+\frac{\Delta z}{2}} p(Z)\,\mathrm{d}Z$$

$$\approx \Delta z \cdot p(Z) \quad \text{(for small } \Delta z\text{)}$$

Continuous vs. Discretized Random Variables

- Z is a continuous random variable with the probability density function

$$p(Z) = \lim_{\Delta z \to 0} \frac{P\left(z - \frac{\Delta z}{2} \leq Z < z + \frac{\Delta z}{2}\right)}{\Delta z}$$

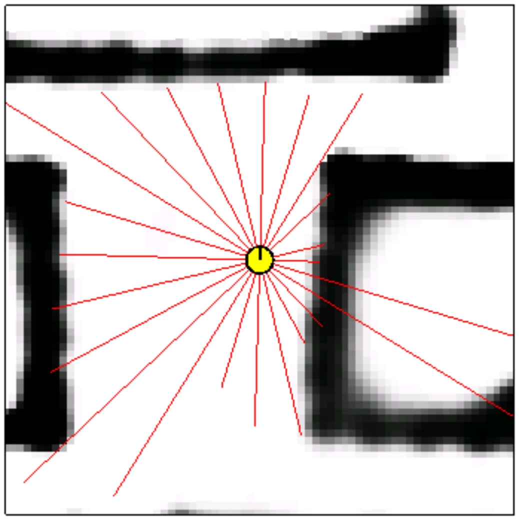
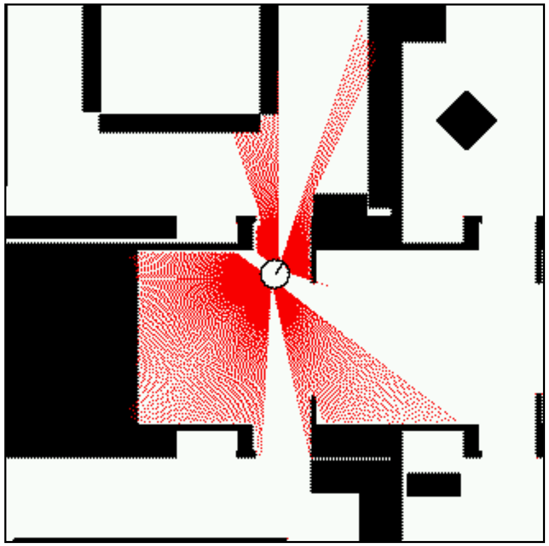
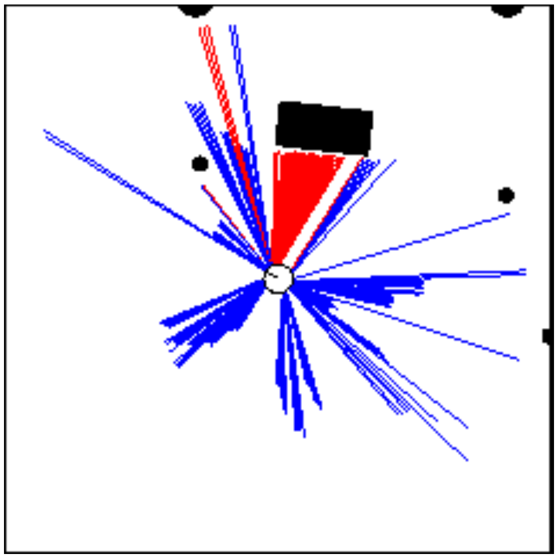
- We can only measure and represent Z in discrete steps Δz
- As Δz is constant for all measurements, we can ignore it when computing $bel(x_t)$ if we normalize $bel(x_t)$ in the end

el klam da valid in case of small delta z only, w bn7tag enna lazm n3ml normalization lama nkhls.

Sensors for Mobile Robots

- Proprioceptive sensors:
  - Accelerometers
  - Gyroscopes
  - Compasses
- Typical proximity sensors:
  - Sonars
  - Laser range-finders
- Visual sensors:
  - (Stereo) Cameras
  - Structured light (RGBD cameras)
- Infrastructure-based sensors: GPS, WLAN

Proximity Sensors

Question: How can we calculate the likelihood of such a measurement given the robot pose?

Beam-Based Sensor Model

- Sensor data consists of K measurements

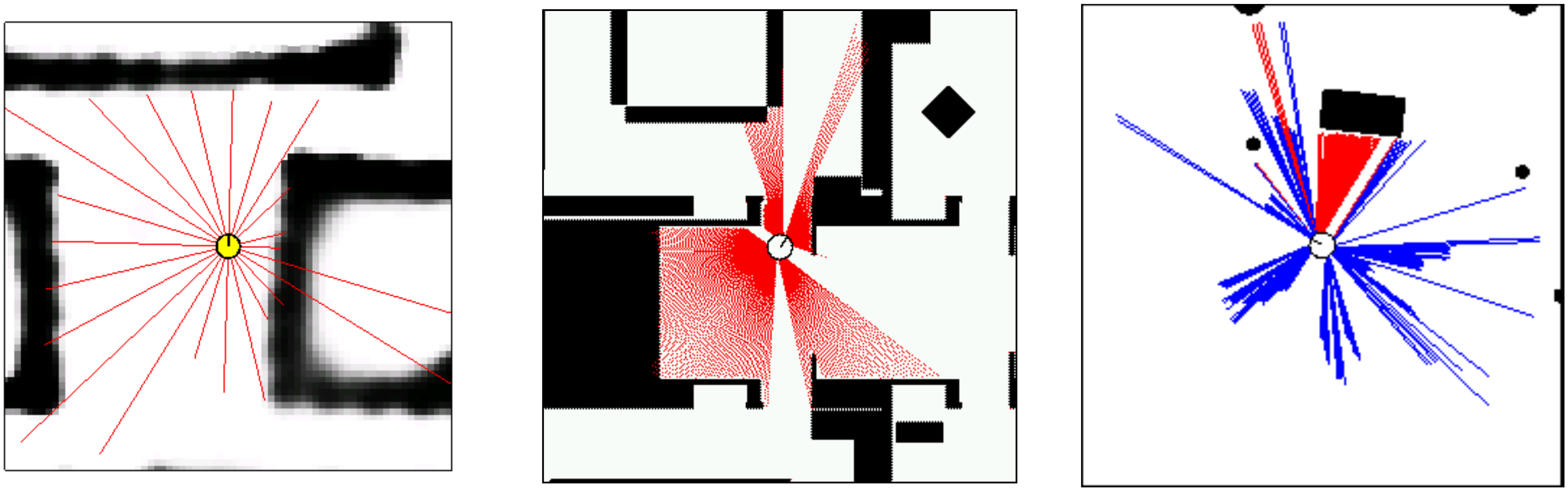
$$z = \{z_1, \ldots, z_K\}$$

- Assumption: The individual measurements are independent given the robot's pose:

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

- "How well can the distance measurements be explained given the pose (and the map)"

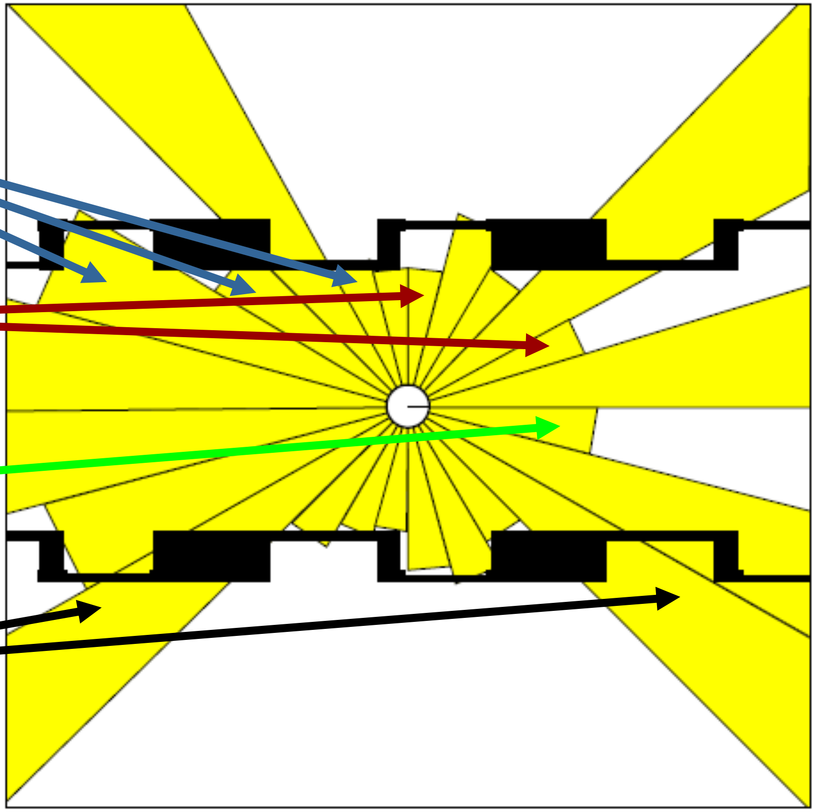
Beam-Based Sensor Model



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

Typical Measurement Errors of an Range Measurements

1. Beams reflected by known obstacles
2. Beams reflected by people / objects
3. Random measurements
4. Maximum range measurements



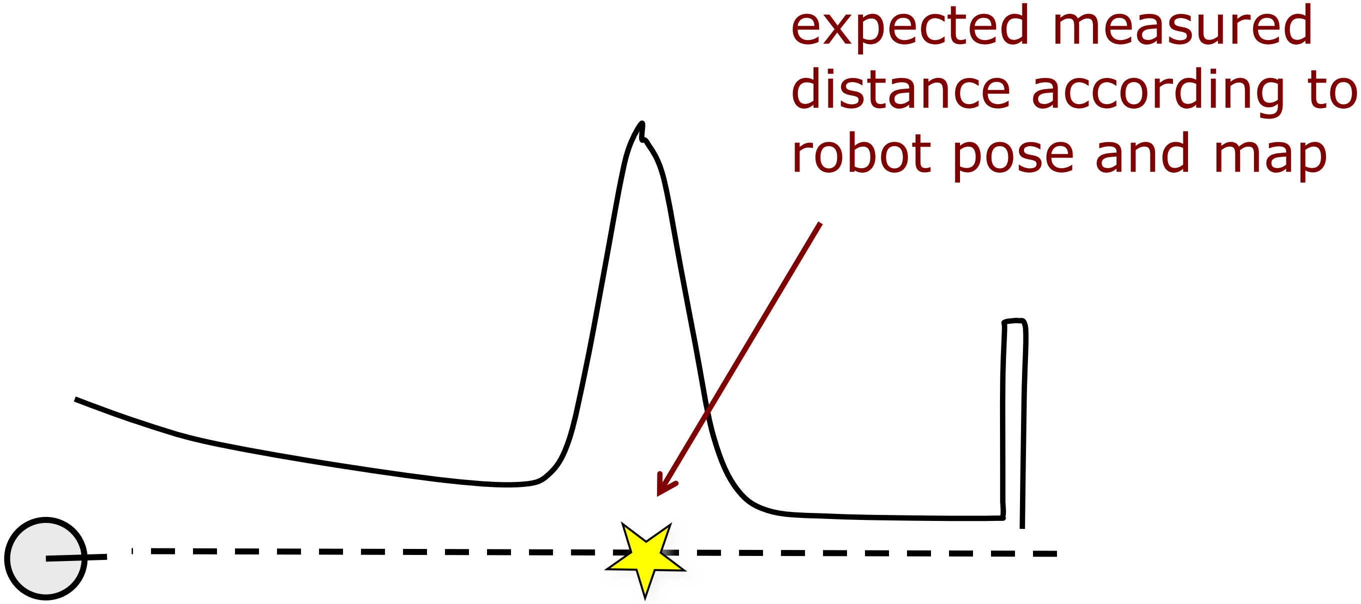
14

Proximity Measurements

- **A measurement can be caused by:**
  - a known obstacle
  - an unexpected obstacle (people, furniture, …)
  - random measurements, cross-talk (sonars)
  - missing all obstacles
- **Noise is due to uncertainty:**
  - in measuring distance to known obstacle (sensor noise)
  - in the position of known obstacles (“map noise”)
  - in the position of additional objects
  - whether an obstacle is missed

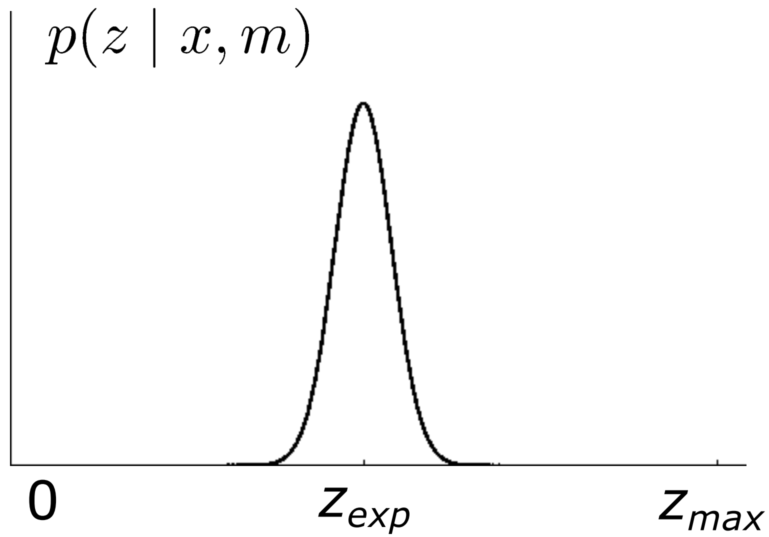
Beam-Based Proximity Model

- Considers the first obstacle along the line of sight
- Mixture of four components

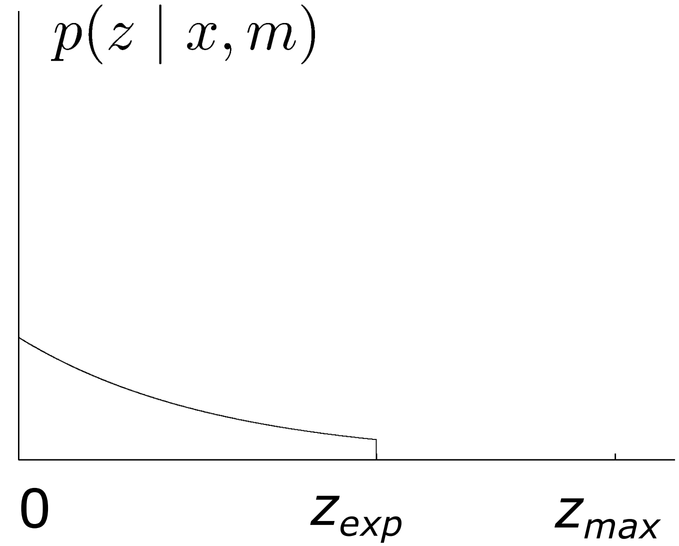


Beam-Based Proximity Model

Measurement noise



Unexpected objects



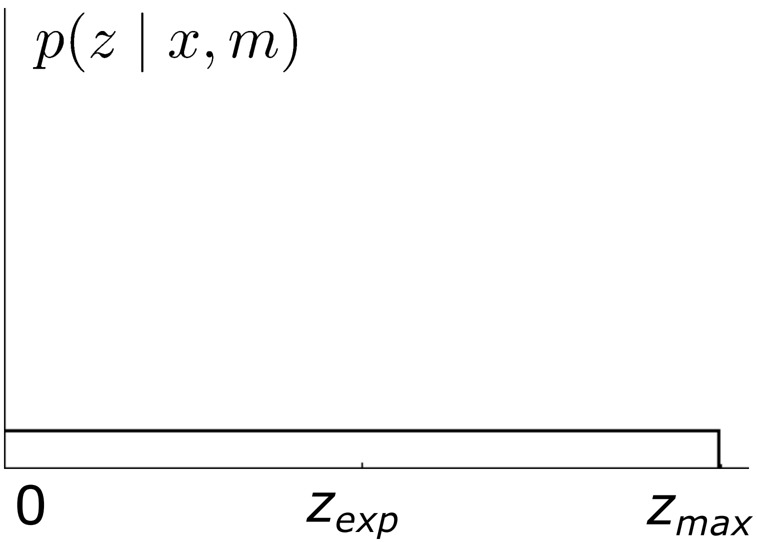
$$p_{hit}(z \mid x, m) = \begin{cases} \frac{\eta}{\sqrt{2\pi\sigma^2}} e^{-\frac{(z - z_{exp})^2}{2\sigma^2}} & \text{if } z \le z_{max} \\ 0 & \text{otherwise} \end{cases}$$

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

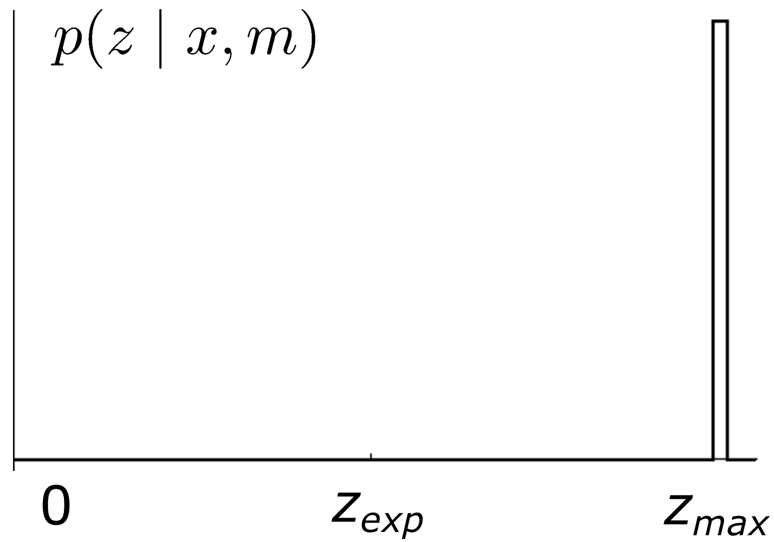
Note: The expected distance z_{exp} is computed by ray casting in the map, starting from the pose x

Beam-Based Proximity Model

Random measurement



Max range

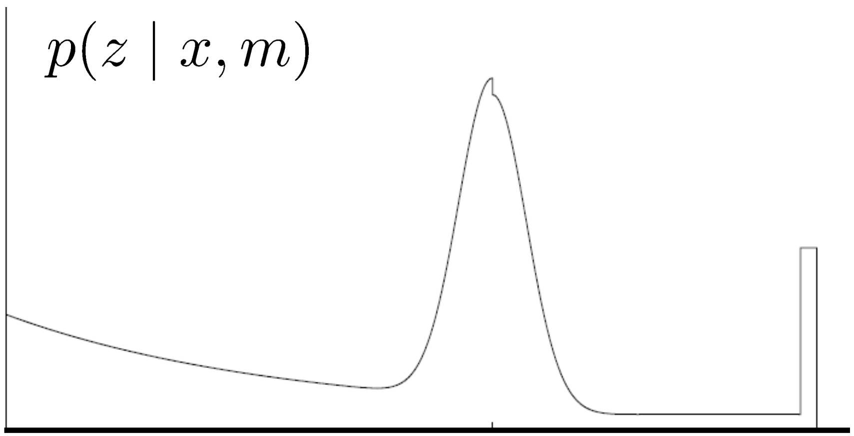


$$p_{rand}(z \mid x, m) = \begin{cases} \frac{1}{z_{max}} & \text{if } z < z_{max} \\ 0 & \text{otherwise} \end{cases}$$

$$p_{max}(z \mid x, m) = \begin{cases} \frac{1}{z_{small}} & \text{if } z \in [z - z_{small}, z_{max}] \\ 0 & \text{otherwise} \end{cases}$$

Note: The expected distance z_{exp} is computed by ray casting in the map, starting from the pose x

Resulting Mixture Density

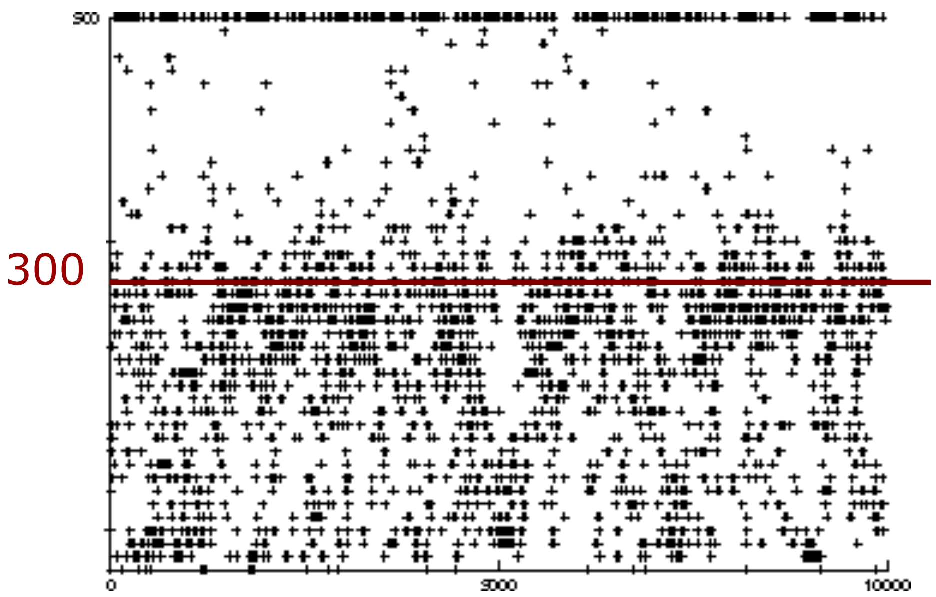


$$p(z \mid x, m) = \begin{pmatrix} \alpha_{\text{hit}} \\ \alpha_{\text{unexp}} \\ \alpha_{\text{max}} \\ \alpha_{\text{rand}} \end{pmatrix}^T \cdot \begin{pmatrix} p_{\text{hit}}(z \mid x, m) \\ p_{\text{unexp}}(z \mid x, m) \\ p_{\text{max}}(z \mid x, m) \\ p_{\text{rand}}(z \mid x, m) \end{pmatrix}$$

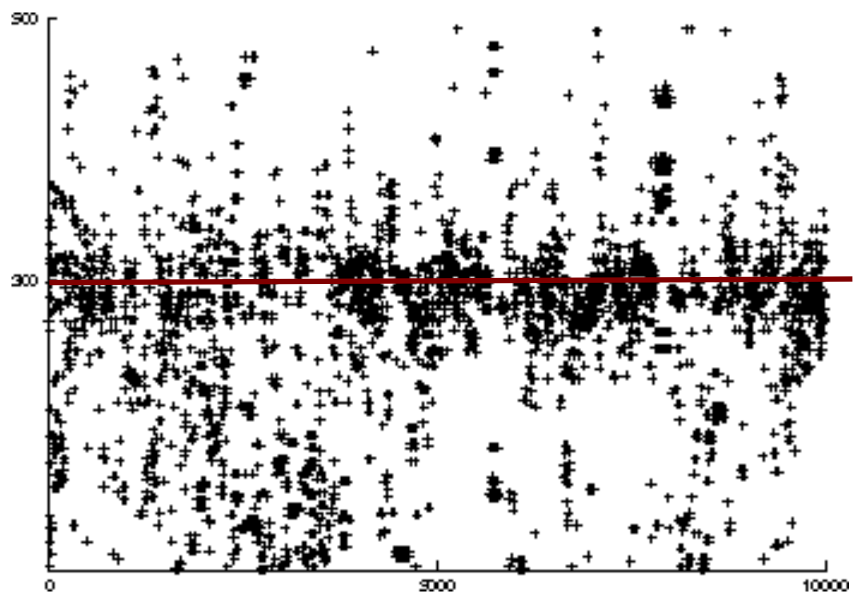
How can we determine the model parameters?

Raw Sensor Data

Measured distances for the “true” expected distance of 300 cm (maximum range 500 cm)



Sonar



Laser

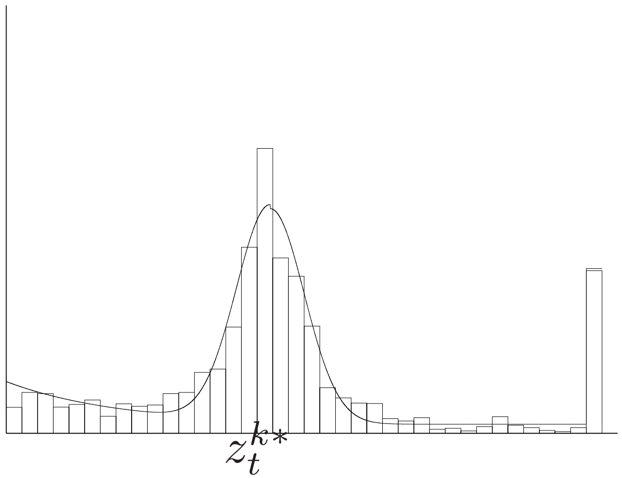
Approximation

- Maximize log likelihood of the data

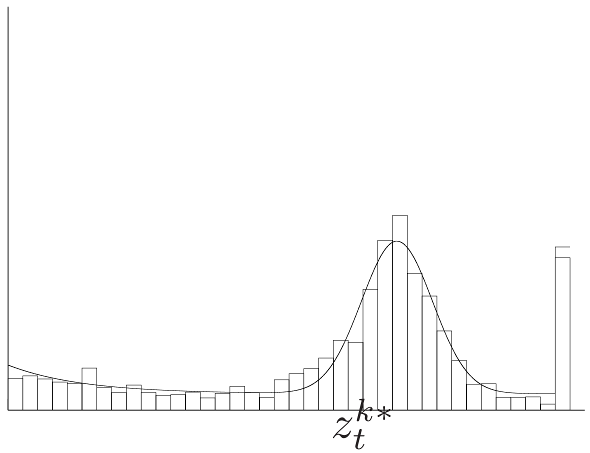
$$p(z \mid z_{\exp})$$

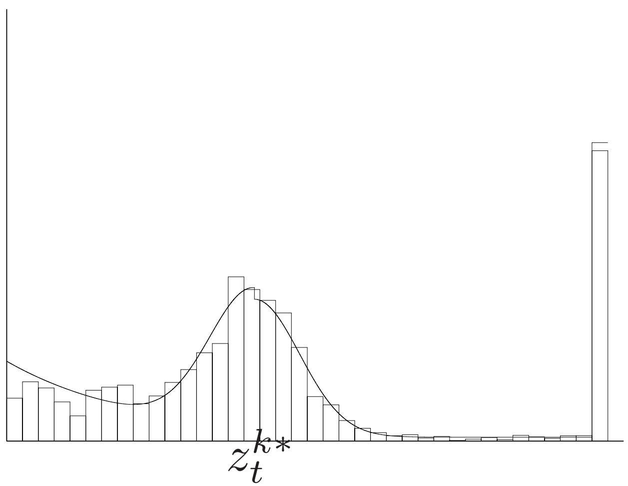
- Search space of n-1 parameters
  - Hill climbing
  - Gradient descent
  - Genetic algorithms
  - …
- Deterministically compute the n-th parameter to satisfy normalization constraint

Approximation Results

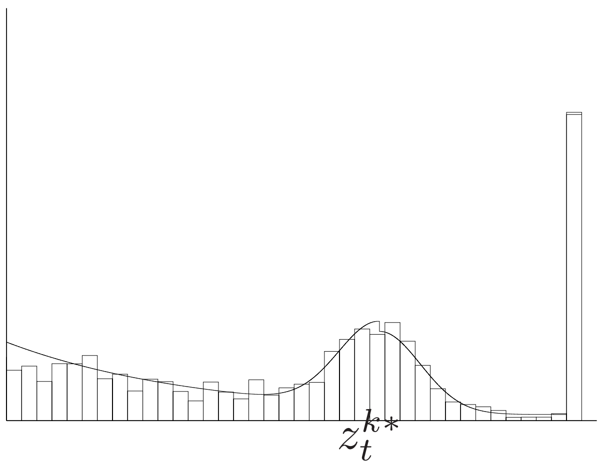


Laser





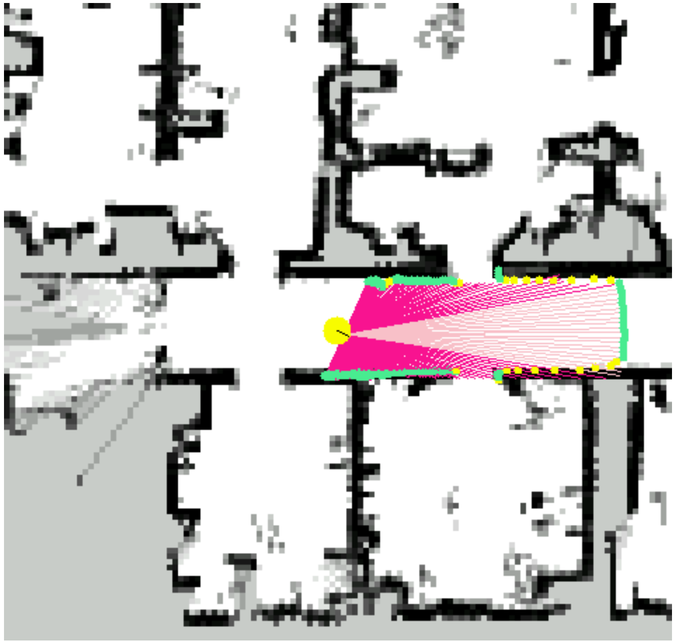
Sonar



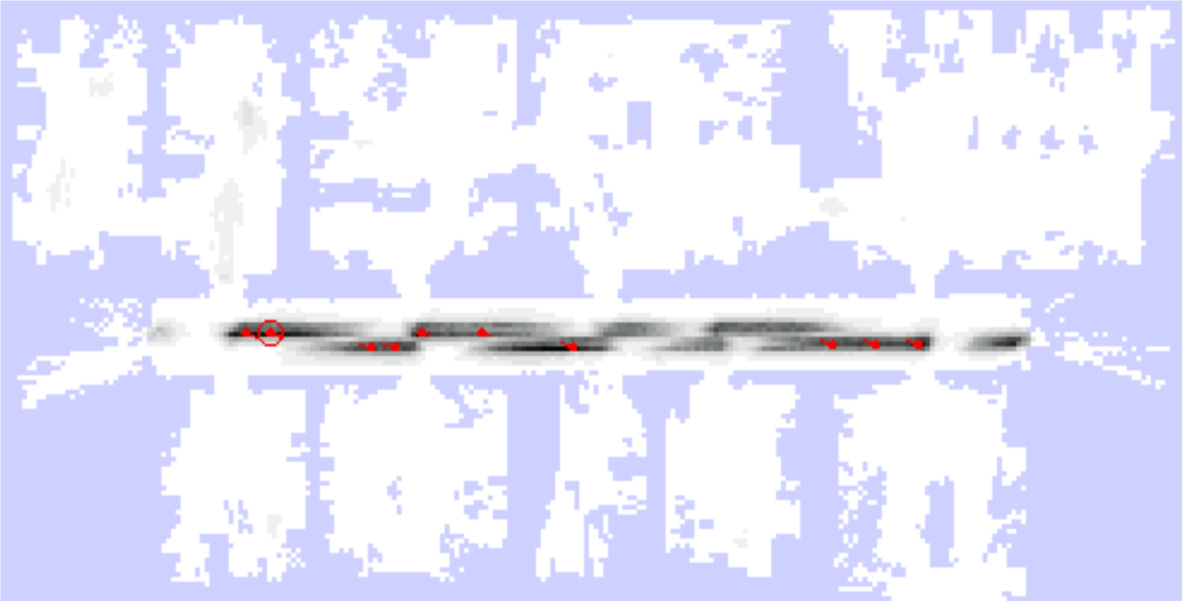
300cm

400cm

Example

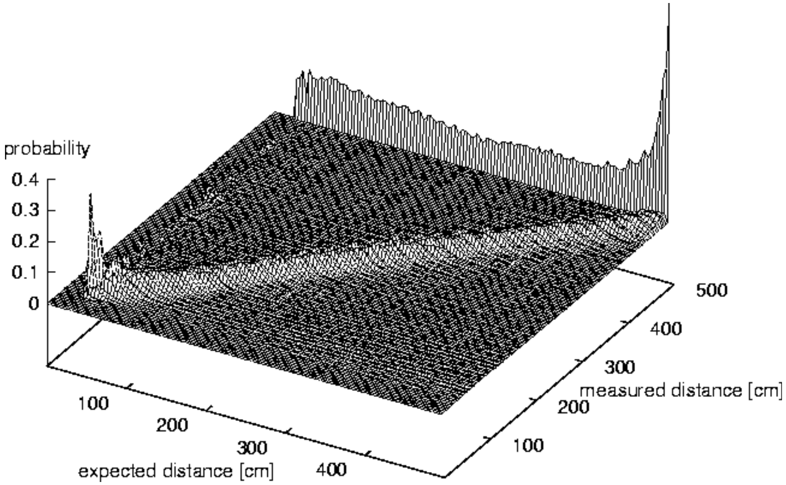


z

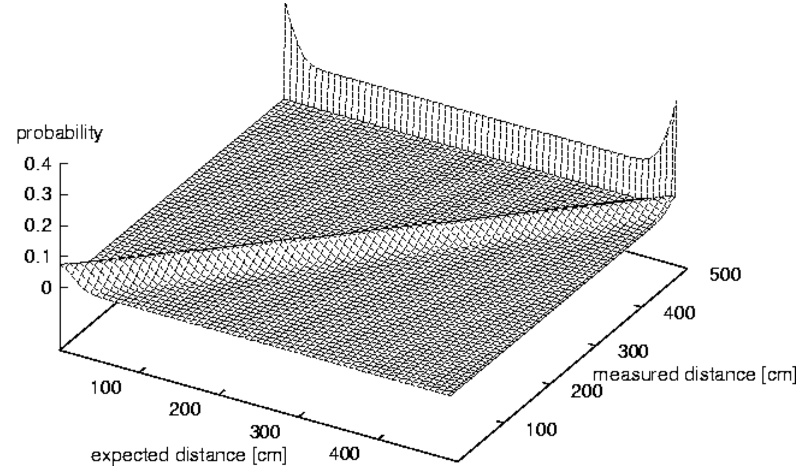


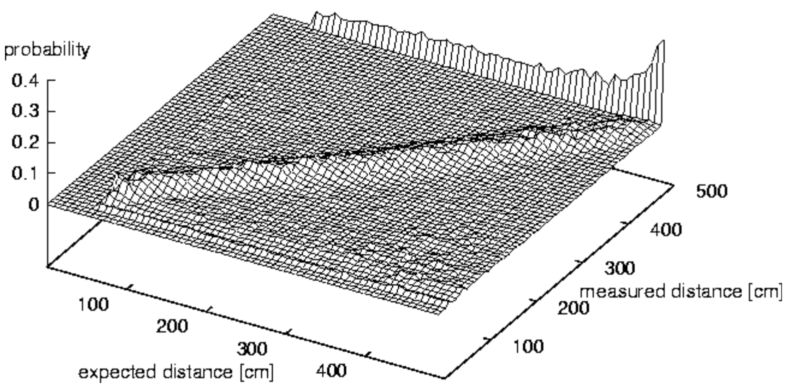
$p(z|x,m)$

Approximation Results

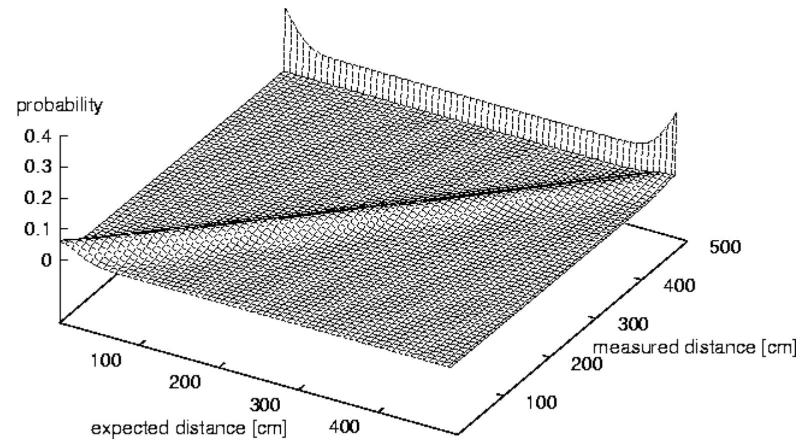


Laser

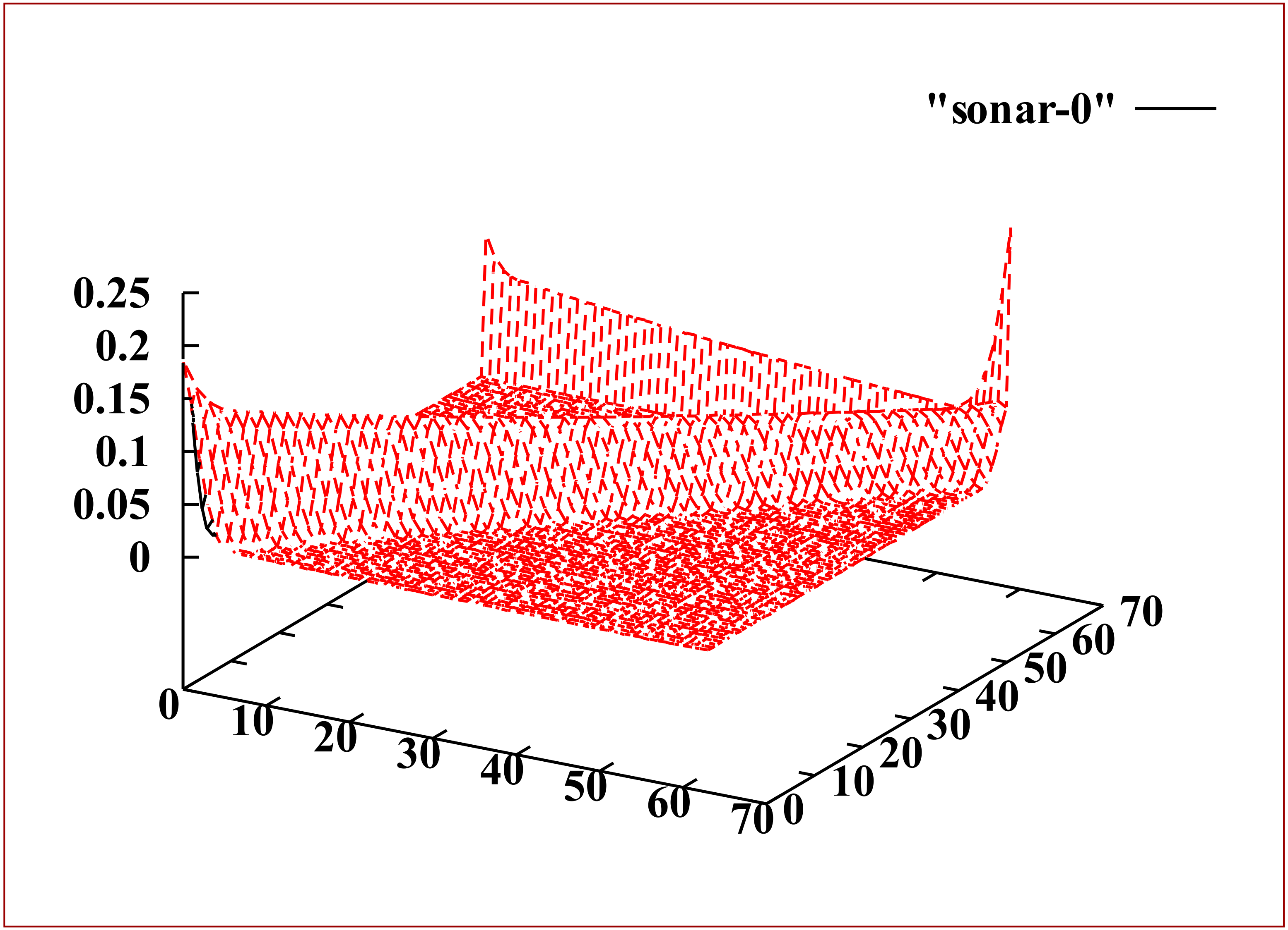




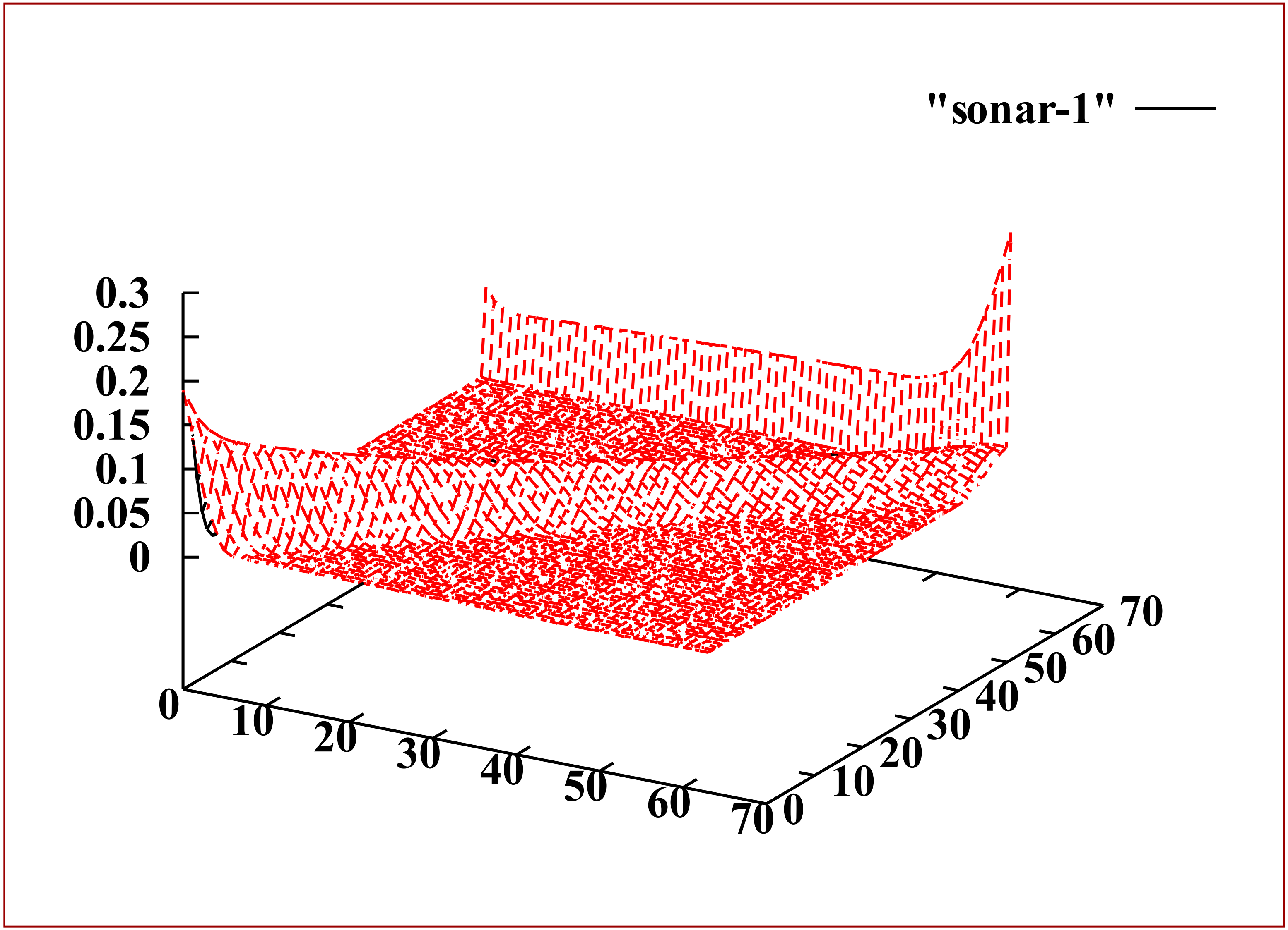
Sonar



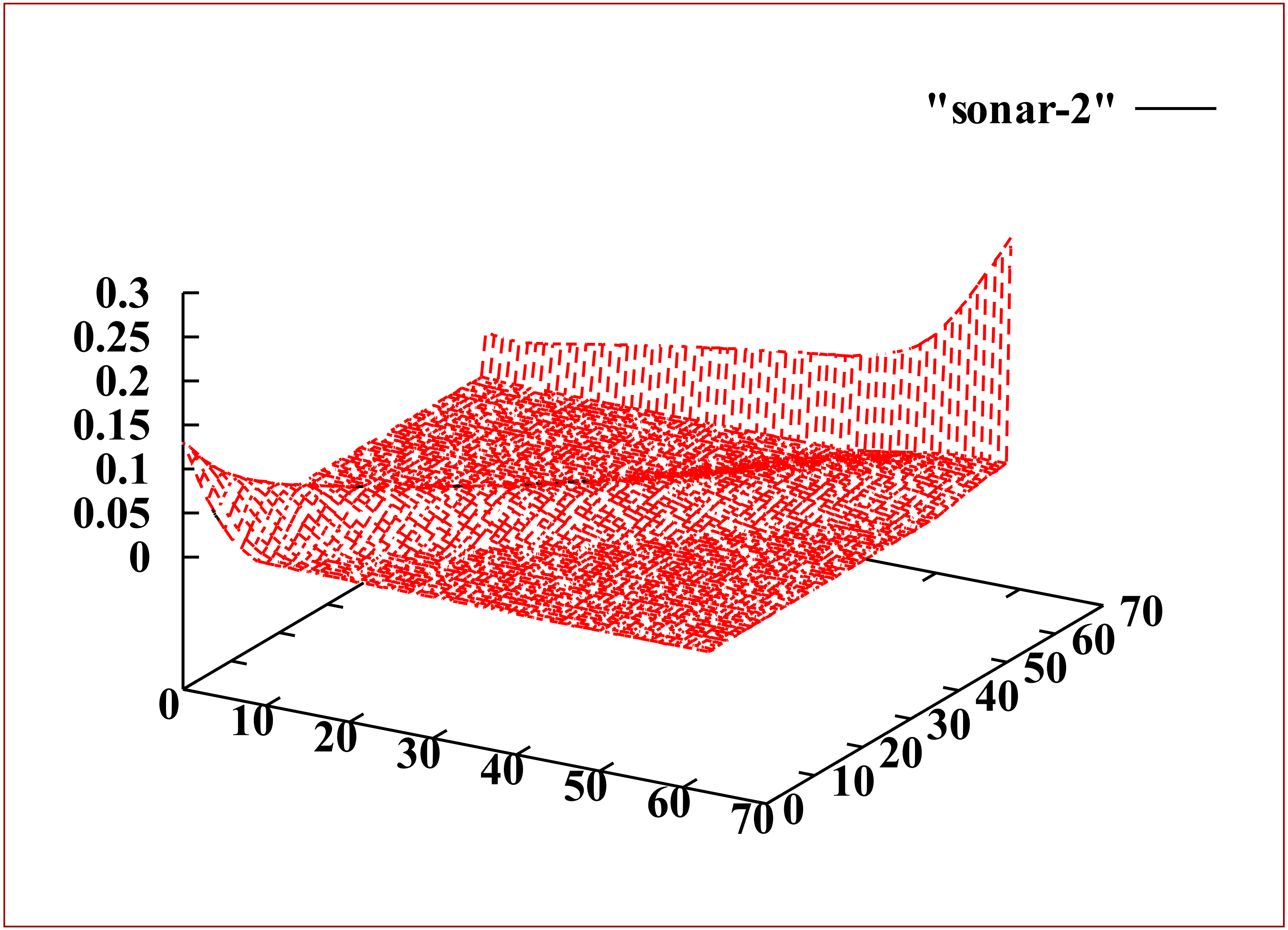
Influence of Angle to Obstacle



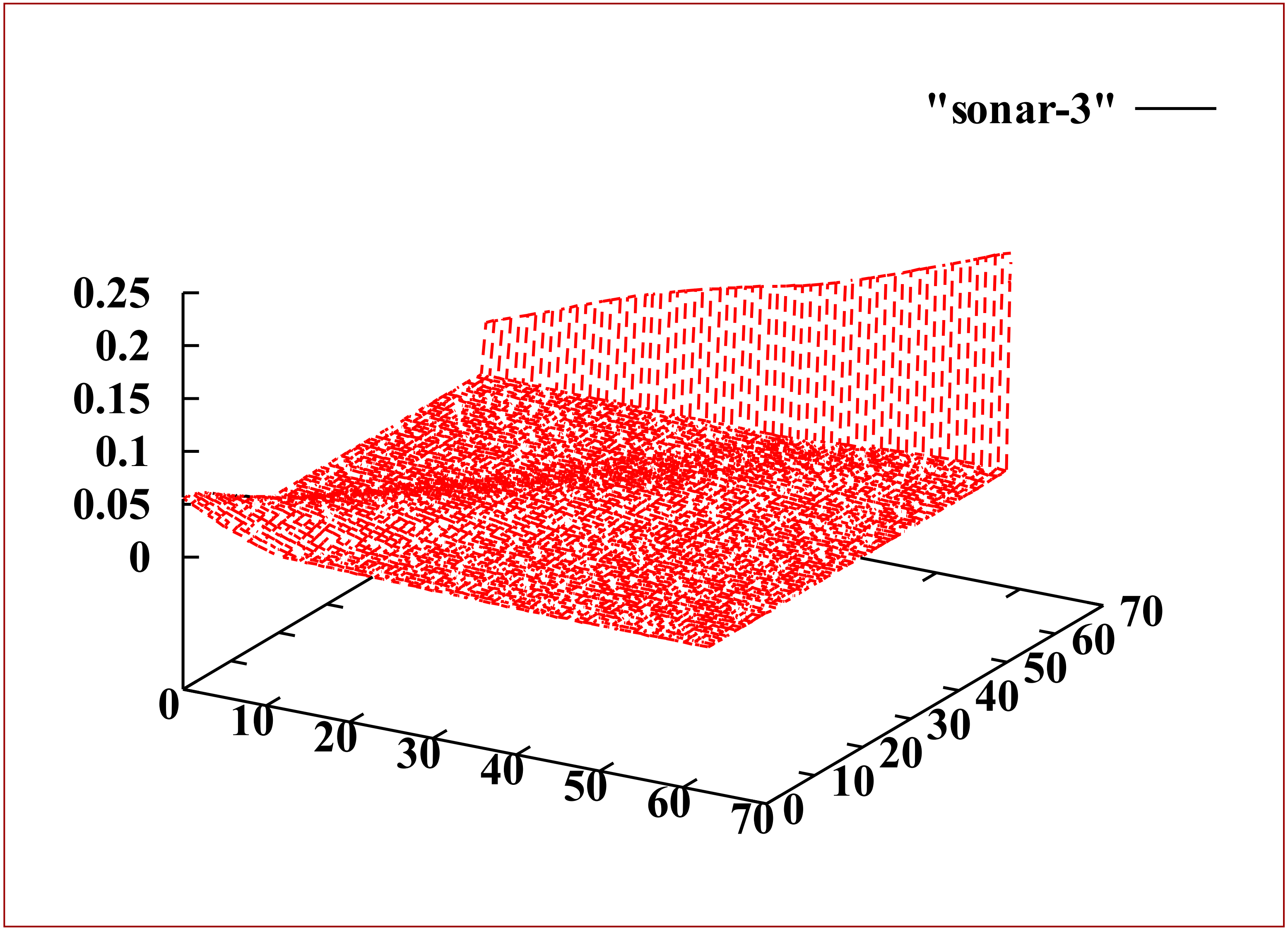
Influence of Angle to Obstacle



Influence of Angle to Obstacle



Influence of Angle to Obstacle



Summary Beam-Based Model (1)

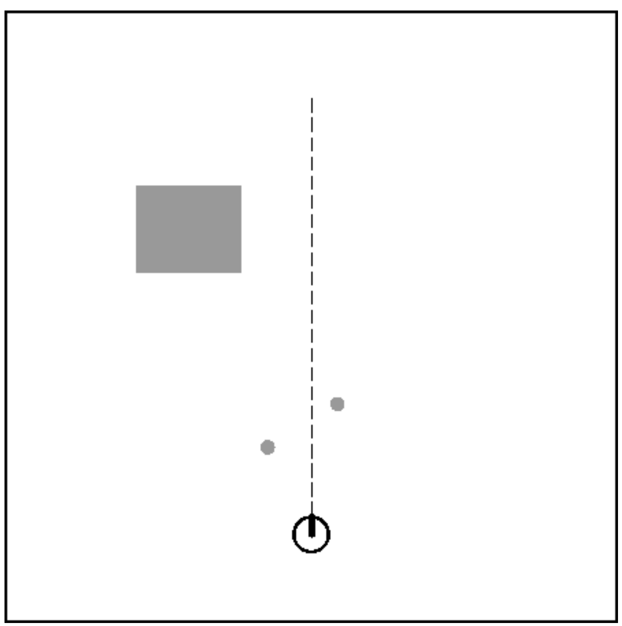
- **Assumes independence between beams**
  - Justification?
  - Problem: Overconfident estimates
- **Models physical causes for measurements**
  - Mixture of densities for these causes
  - Assumes independence between causes

Summary Beam-Based Model (2)

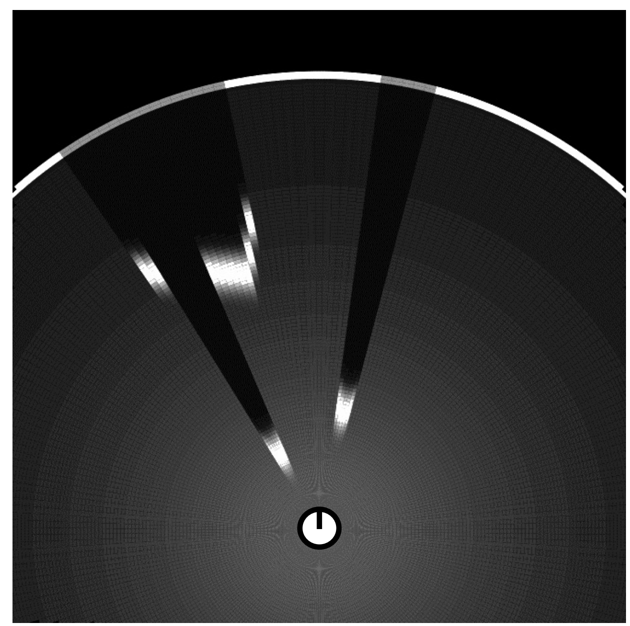
- **Implementation**
  - Learn parameters based on real data
  - Different models should be learned for different angles at which the sensor beam hits the obstacle (sonar)
  - Determine expected distances by ray casting
  - Expected distances can be pre-computed

Summary Beam-Based Model (3)

- **Disadvantages:** The beam-based model is
  - not smooth at edges
  - not very efficient (ray casting or precomputed lookup tables)



Map *m*



Likelihood field

End-Point Model

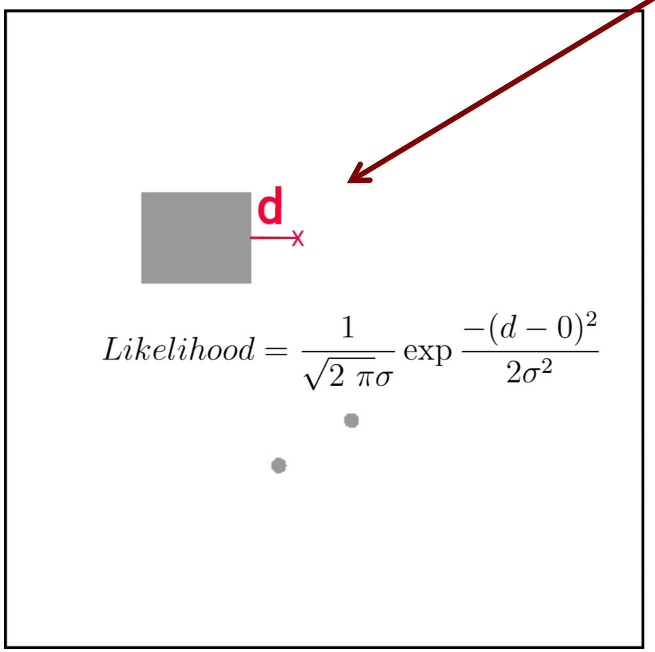
- Idea of the end-point model: Instead of following along the beam, just check the end point of the beam
- Precompute a so-called likelihood field (distance grid)

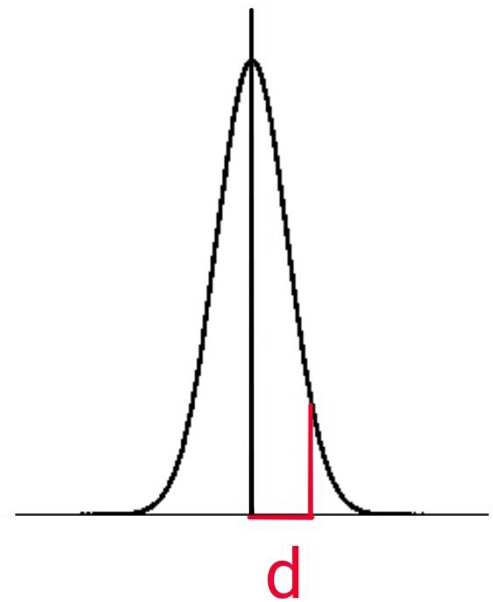
End-Point Model

- Probability is a mixture of:
  - a Gaussian distribution evaluating the **distance to the closest obstacle**,
  - a uniform distribution for random measurements, and
  - a small uniform distribution for max range measurements
- Again, independence between different components is assumed

Gaussian Used within the Likelihood of a Measurement

end point and distance to the closest obstacle

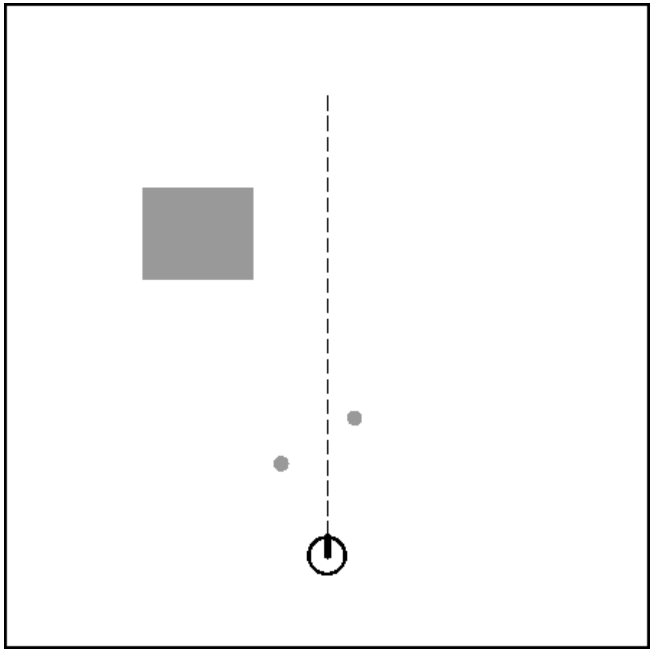




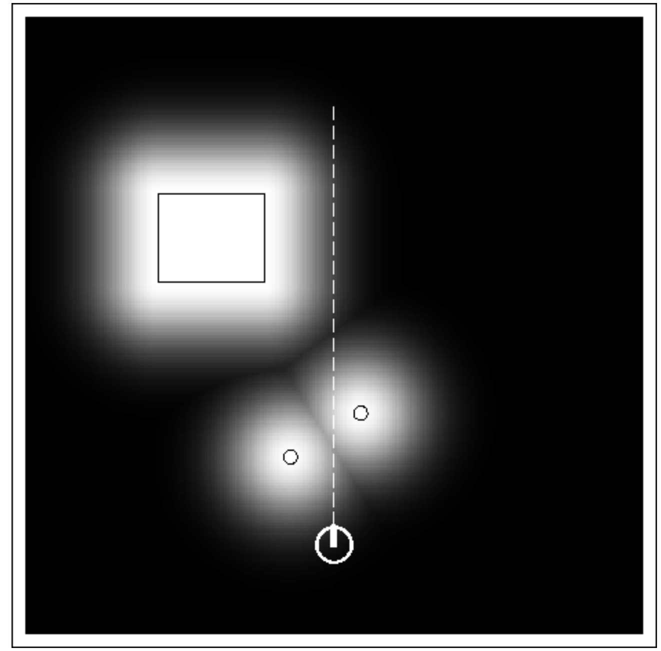
Map with three obstacles

Gaussian to evaluate the distance

Example

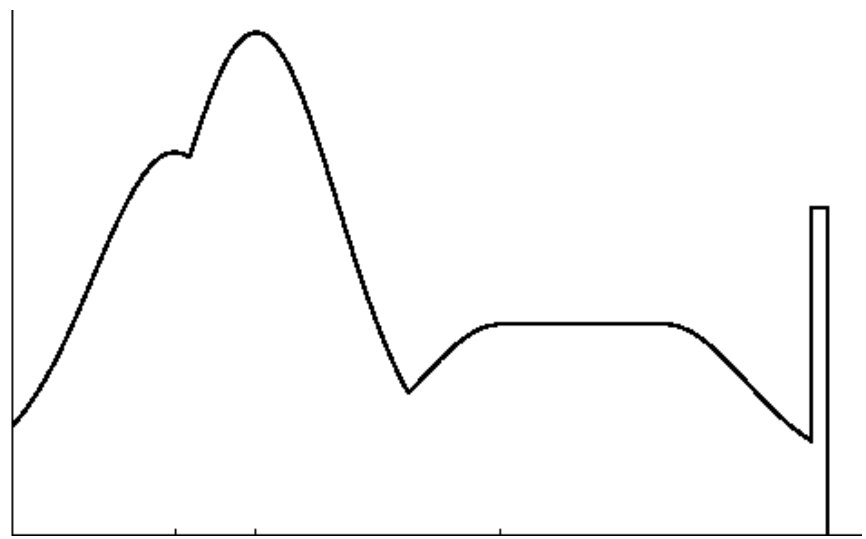


Map m



Likelihood field

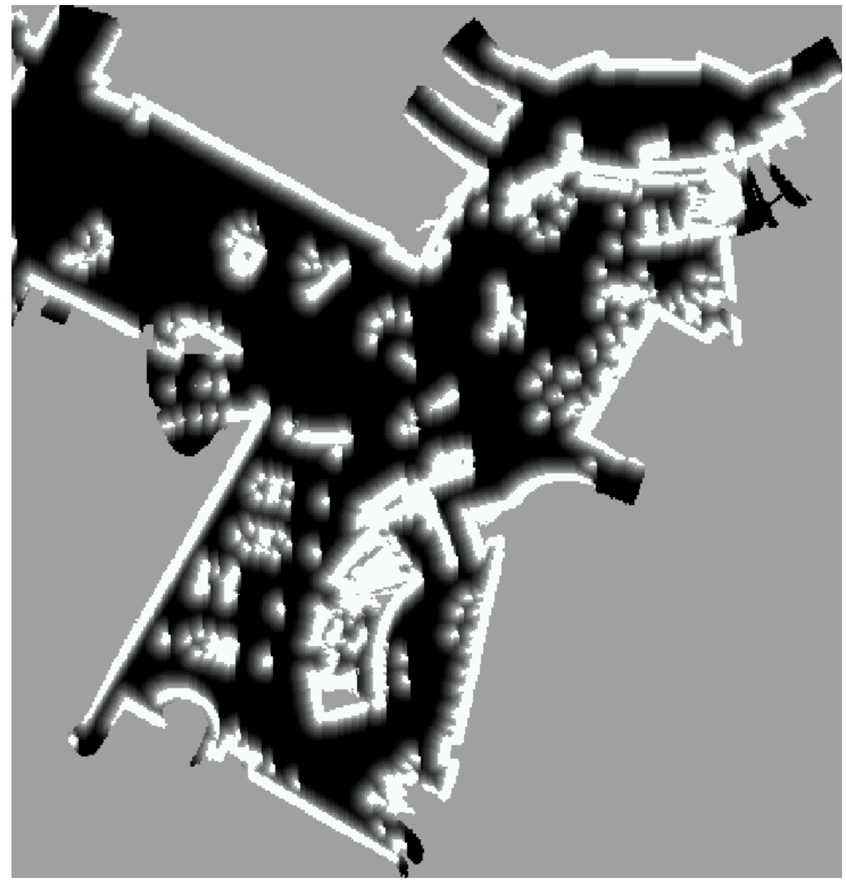
$p(z|x,m)$



measured length z

San Jose Tech Museum

Occupancy grid map



Likelihood field

Note: Precomputed independently of robot pose

Properties End-Point Model

- Highly efficient, uses 2D tables only
- Distance grid is smooth w.r.t. to small changes in robot position
- Ignores physical properties of beams
- Treats sensor as if it can see through walls

Landmarks

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

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

Distance and Bearing

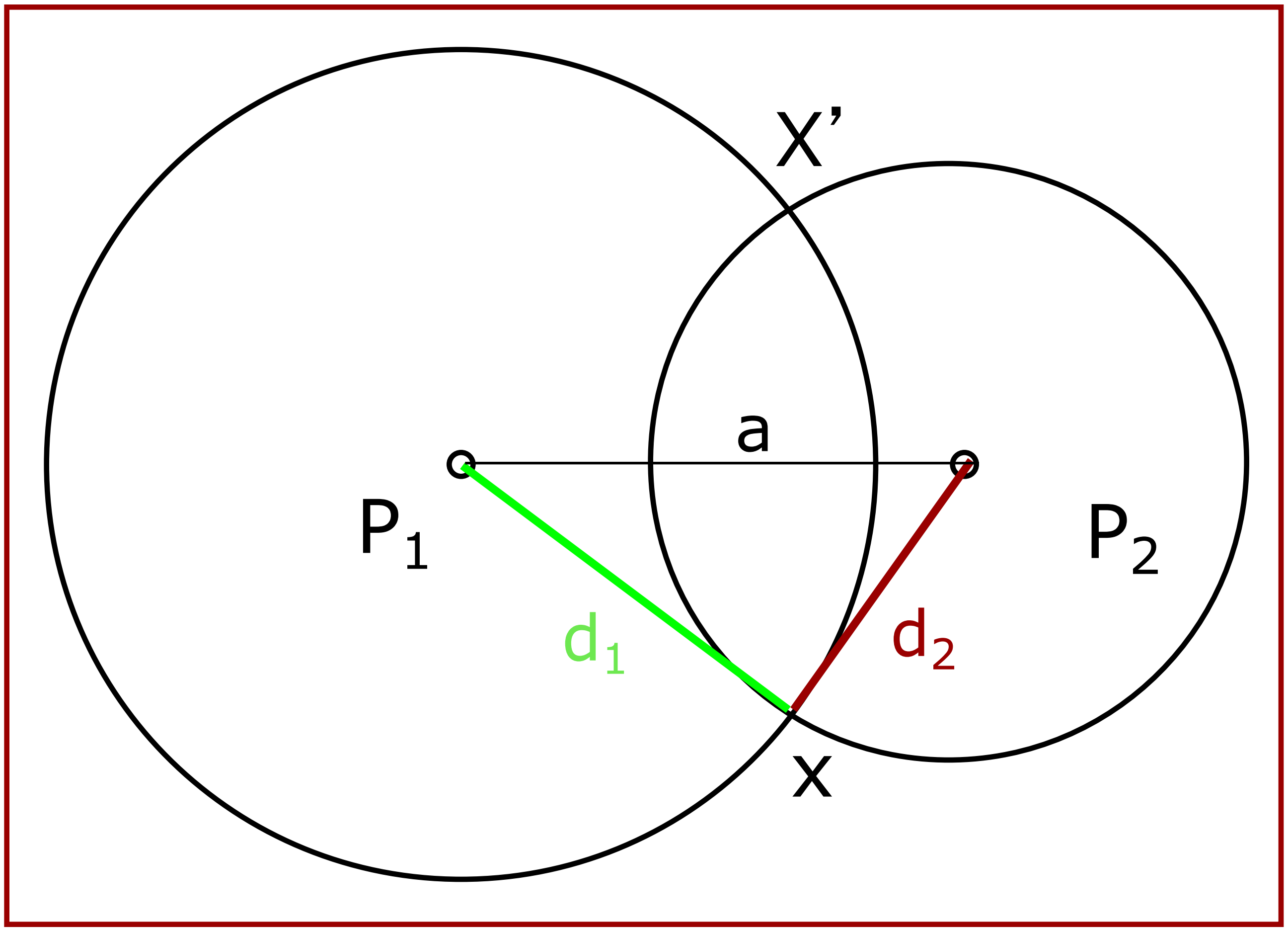
Probabilistic Model

1. Algorithm **landmark_detection_model**(z,x,m):
   $z = \langle i, d, \alpha \rangle, x = \langle x, y, \theta \rangle$
2. $\hat{d} = \sqrt{(m_x(i) - x)^2 + (m_y(i) - y)^2}$ — Expected distance
3. $\hat{\alpha} = \operatorname{atan2}(m_y(i) - y, m_x(i) - x) - \theta$ — Expected angle
4. $p_{\det} = \operatorname{prob}(\hat{d} - d, \varepsilon_d) \cdot \operatorname{prob}(\hat{\alpha} - \alpha, \varepsilon_\alpha)$ — Independence assumption, two Gaussians
5. Return $p_{\det}$

Assumption: Correspondences are known

Distances Only
No Uncertainty

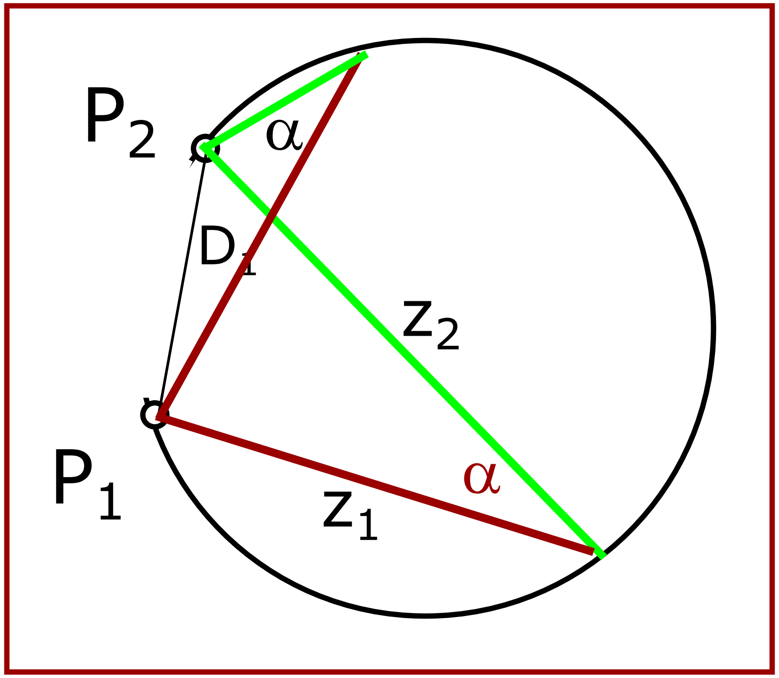
$$x = (a^2 + d_1^2 - d_2^2)/2a$$
$$y = \pm\sqrt{(d_1^2 - x^2)}$$

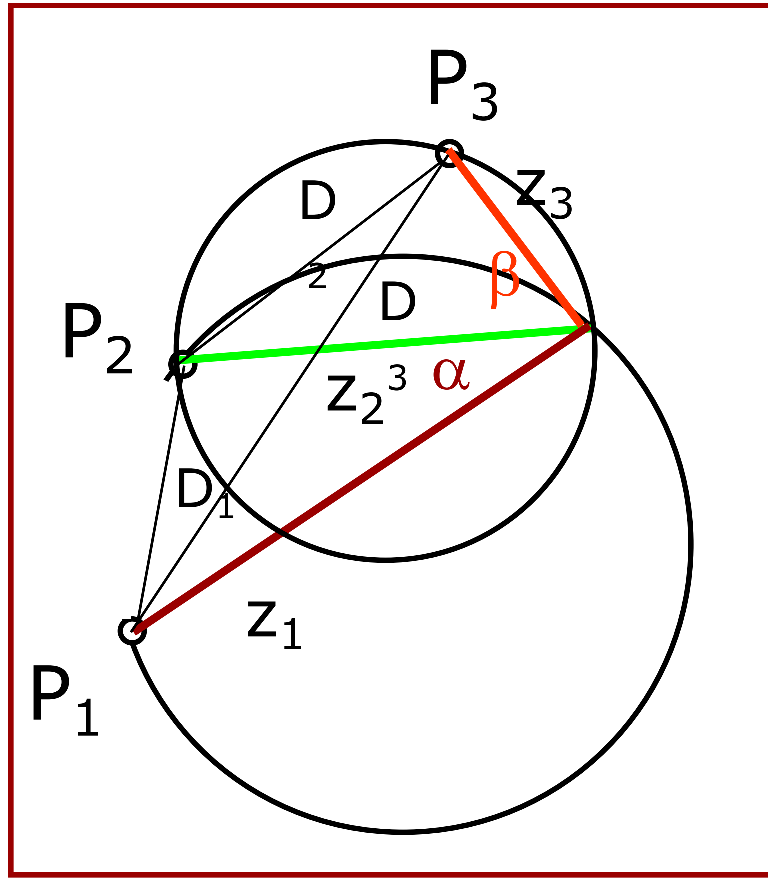


$P_1=(0,0)$

$P_2=(a,0)$

Bearings Only
No Uncertainty





Law of cosine

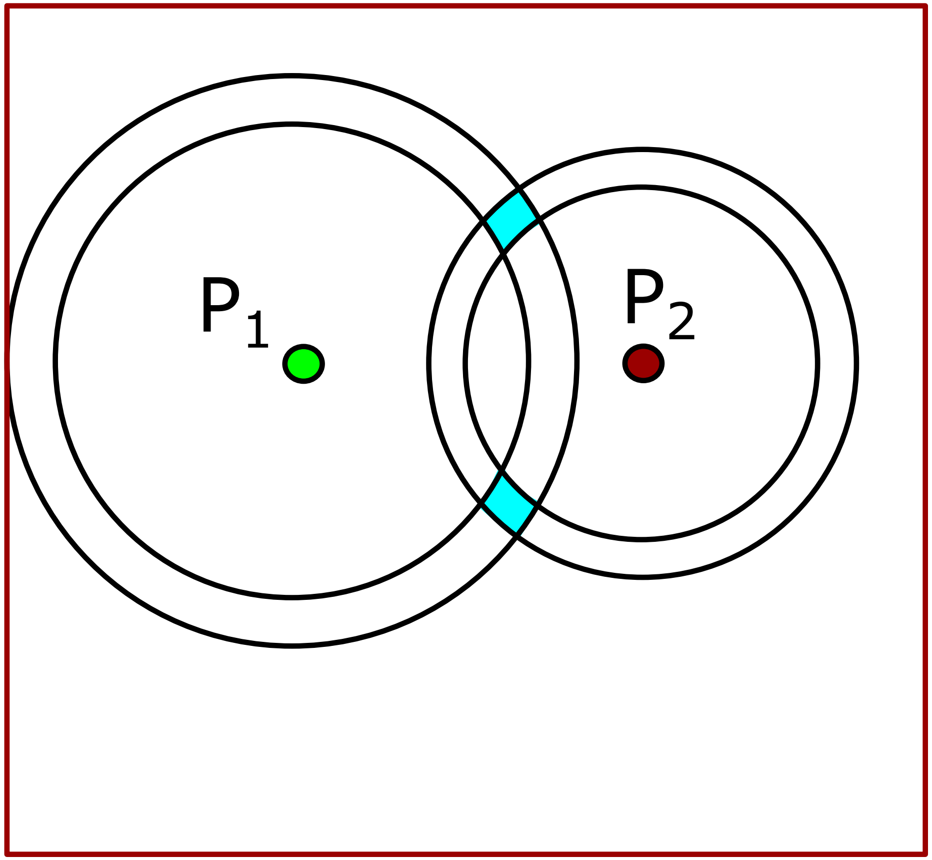
$$D_1^2 = z_1^2 + z_2^2 - 2\, z_1 z_2 \cos\alpha$$

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

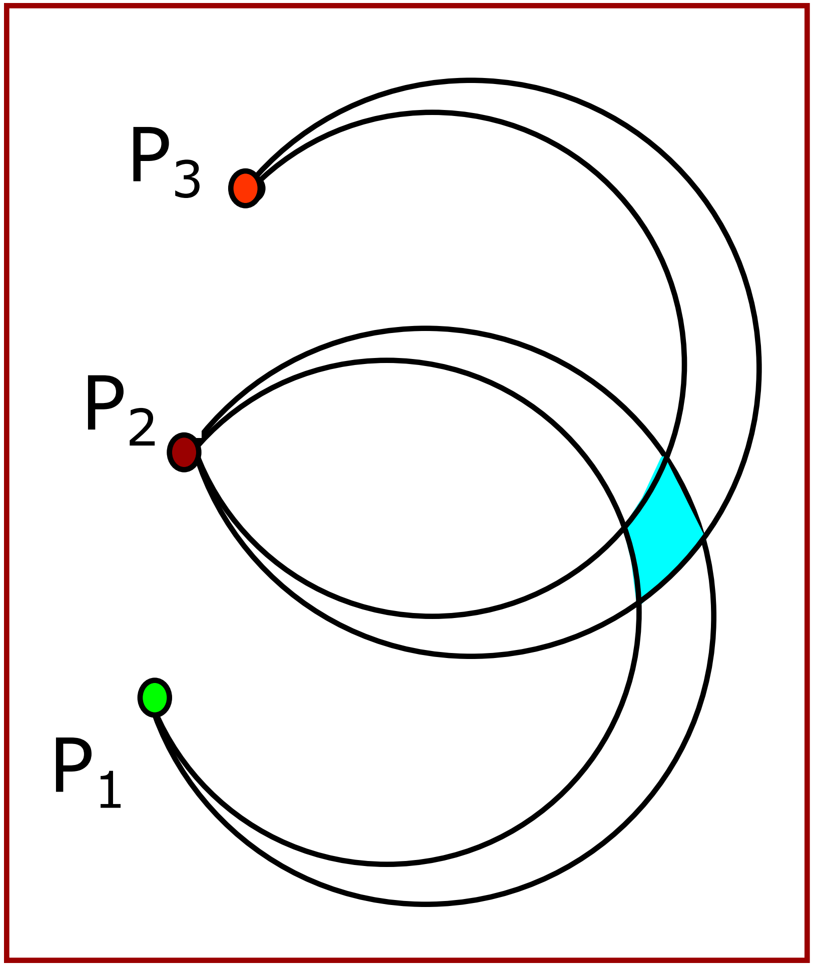
$$D_2^2 = z_2^2 + z_3^2 - 2\, z_1 z_2 \cos(\beta)$$

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

Landmark Measurements with Uncertainty



Distance only



Bearings only

Summary (1)

- **Explicitly modeling uncertainty in sensing is key to robustness**
- In many cases, good models can be found by the following approach:
  - Determine the parametric model of a noise-free measurement
  - Analyze the individual sources of noise
  - Add adequate noise to parameters (add densities for noise)
  - Learn parameters by fitting a model to the data

Summary (2)

- **The likelihood of a measurement is given by "probabilistically comparing" the actual with the expected measurement**
- It is extremely important to be aware of the underlying assumptions!

Midterm

- This lecture is included in the Midterm

Acknowledgment

- These slides have been created by Wolfram Burgard, Dieter Fox, Cyrill Stachniss and Maren Bennewitz