MEN791
Autonomous Unmanned Vehicles

Sensor models

CONTACT

Ulsan National Institute of Science and Technology

Address 50 UNIST-gil, Ulju-gun, Ulsan, 44919, Korea
Tel. +82 52 217 0114 **Web.** www.unist.ac.kr

School of Mechanical, Aerospace, and Nuclear Engineering
5th Engineering Building Room 801-4
Tel. +82 52 217 3048 **Web.** https://sites.google.com/site/aslunist/

Sensors of Wheeled Robots

Perception of the environment

Active:

- Ultrasound
- Laser range finder
- Infrared

Passive:

- Cameras
- Tactiles

Time of flight

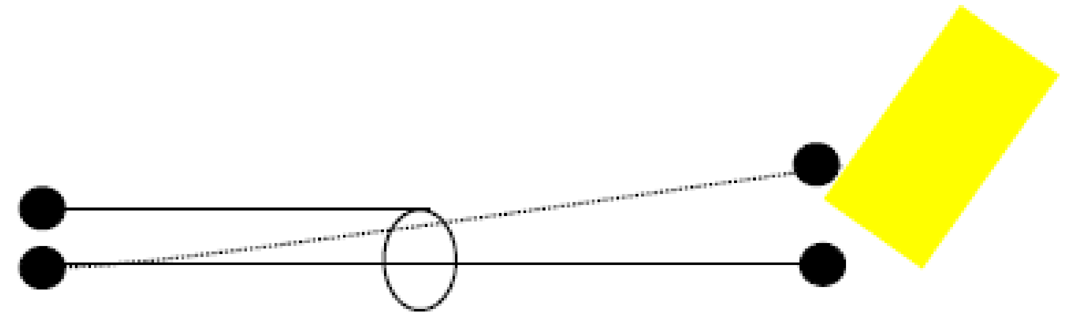
Intensity-based

Sensors for Mobile Robots

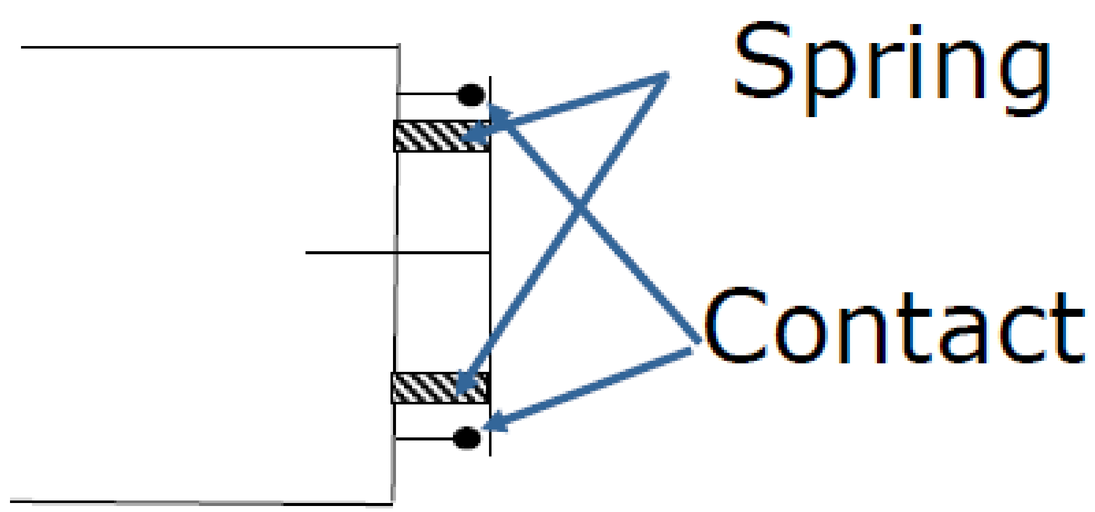
- Contact sensors: Bumpers
- Internal sensors
  - Accelerometers (spring-mounted masses)
  - Gyroscopes (spinning mass, laser light)
  - Compasses, inclinometers (earth magnetic field, gravity)
- Proximity sensors
  - Sonar (time of flight)
  - Radar (phase and frequency)
  - Laser range-finders (triangulation,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

Tactile Sensors

Measure contact with objects



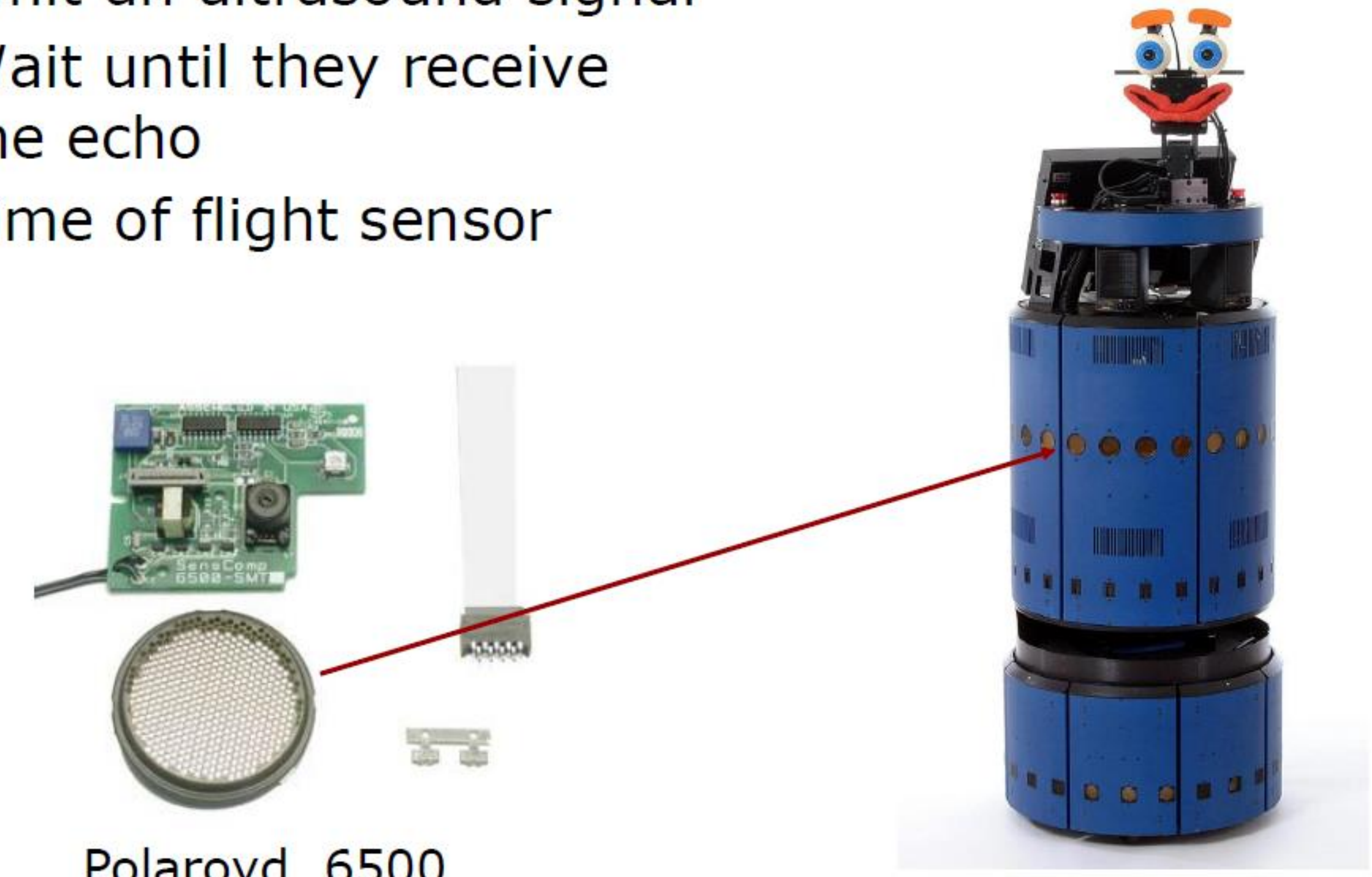
Touch sensor



Bumper sensor

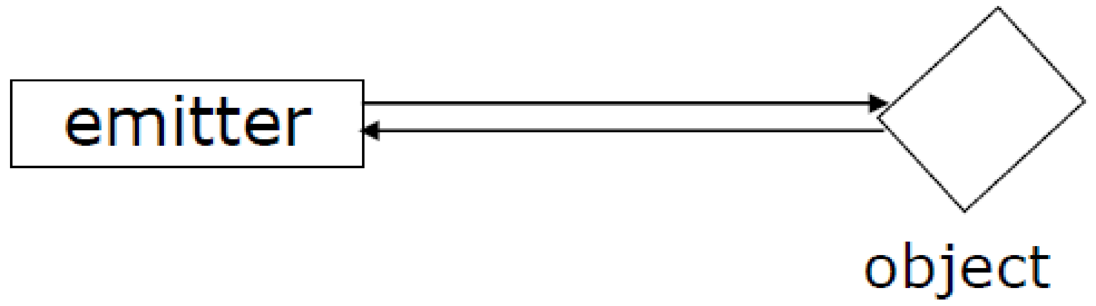
Ultrasound Sensors

- Emit an ultrasound signal
- Wait until they receive the echo
- Time of flight sensor



Polaroyd 6500

Time of Flight Sensors



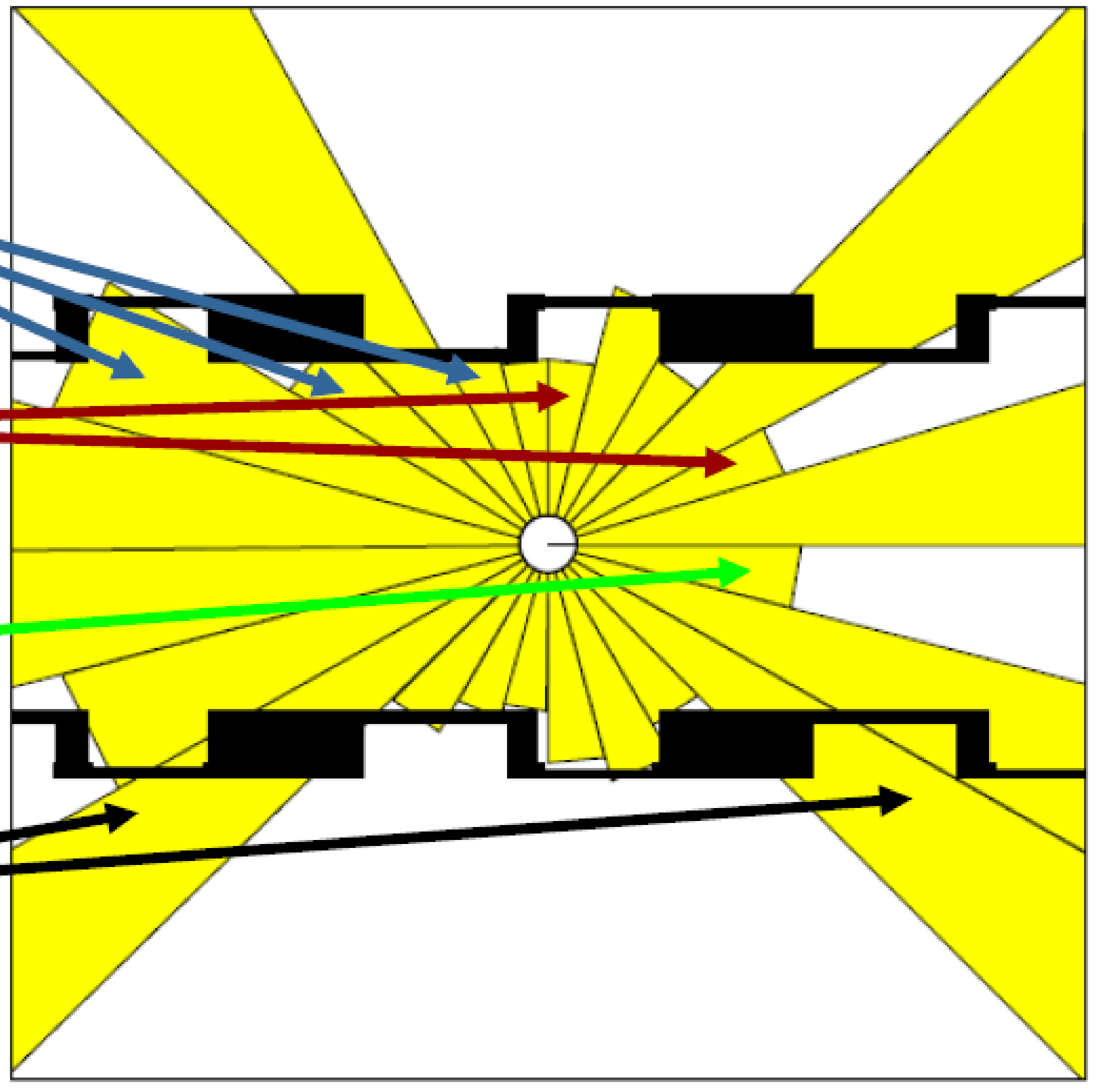
$$d = v \times t / 2$$

v: speed of the signal

t: time elapsed between broadcast of signal and reception of the echo.

Typical Measurement Errors of an Range Measurements

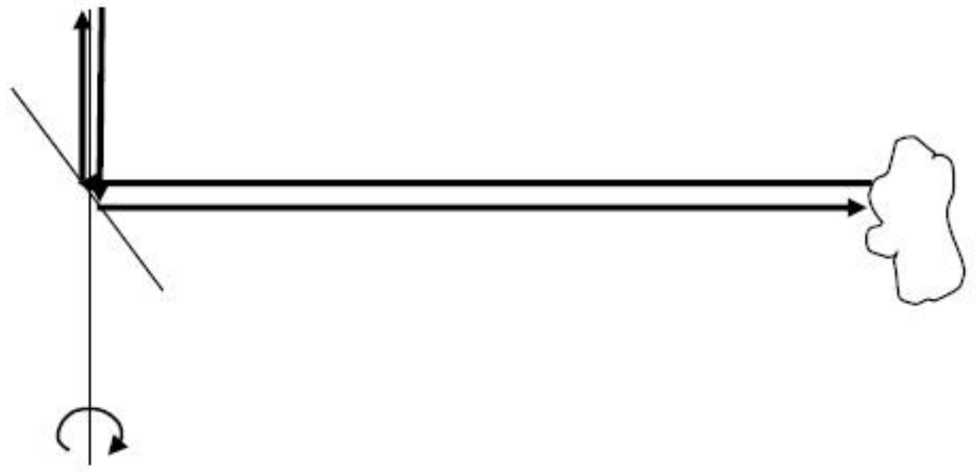
1. Beams reflected by obstacles
2. Beams reflected by persons / caused by crosstalk
3. Random measurements
4. Maximum range measurements



Parallel Operation

- Given a 15 degrees opening angle, 24 sensors are needed to cover the whole 360 degrees area around the robot.
- Let the maximum range we are interested in be 10m.
- The time of flight then is 2*10/330 s=0.06 s
- A complete scan requires 1.45 s
- To allow frequent updates (necessary for high speed) the sensors have to be fired in parallel.
- This increases the risk of crosstalk

Laser Range Scanner

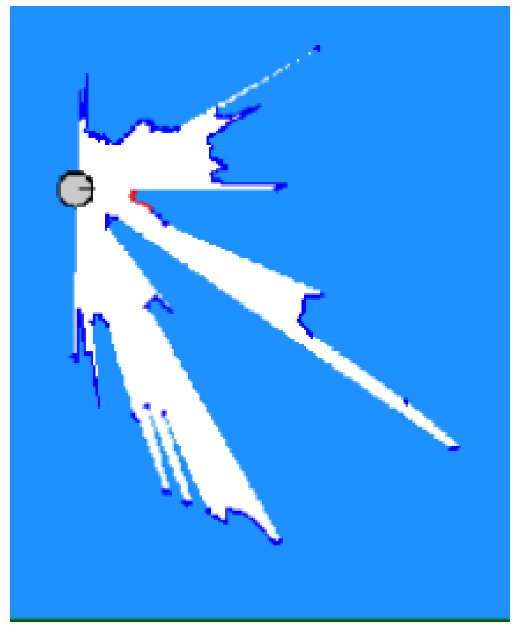


Properties

- High precision
- Wide field of view
- Some laser scanners are security approved for emergency stops (collision detection)

Computing the End Points

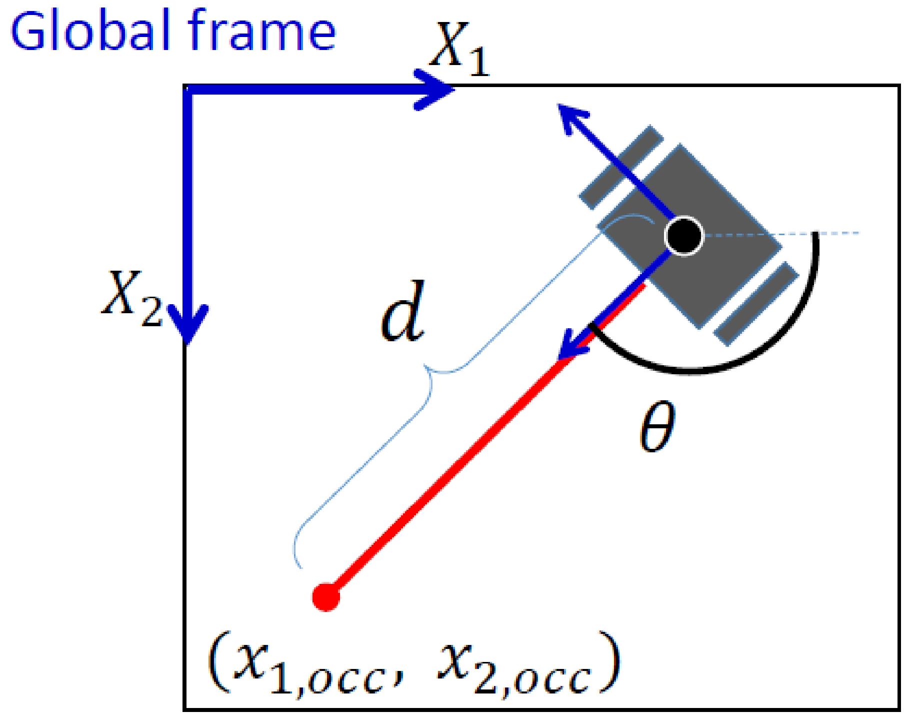
- Laser data comes as an array or range readings, e.g. [1; 1.2; 1.5; 0.1; 81.9; …]
- Assume an field of view of 180 deg
- First beams starts at -½ of the fov
- Maximum range: ~80 m (SICK LMS)



Blackboard:

- Where are the end points relative to the sensor location?
- Where are the end points in an external coordinate system?

Handling Range Measurement on Grid

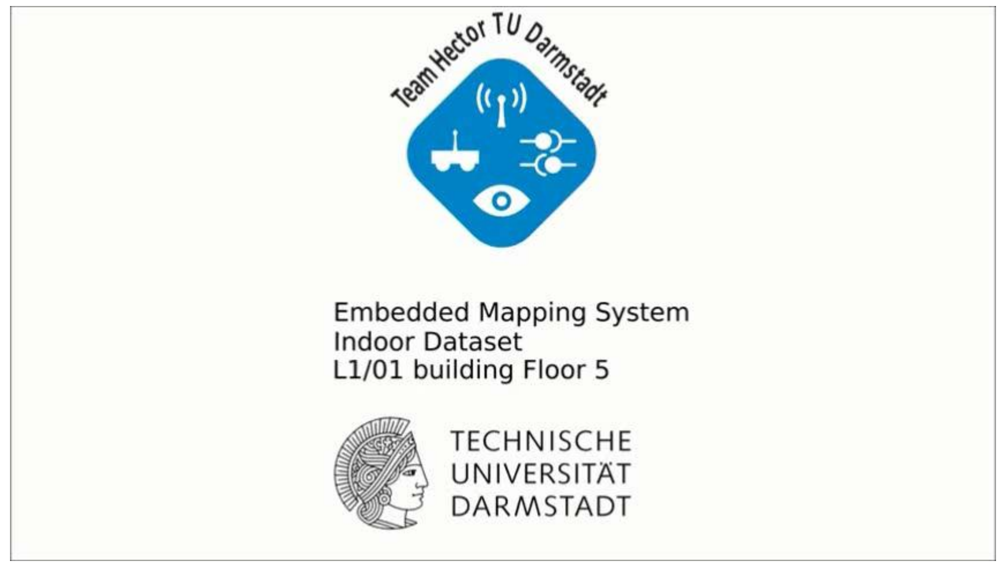


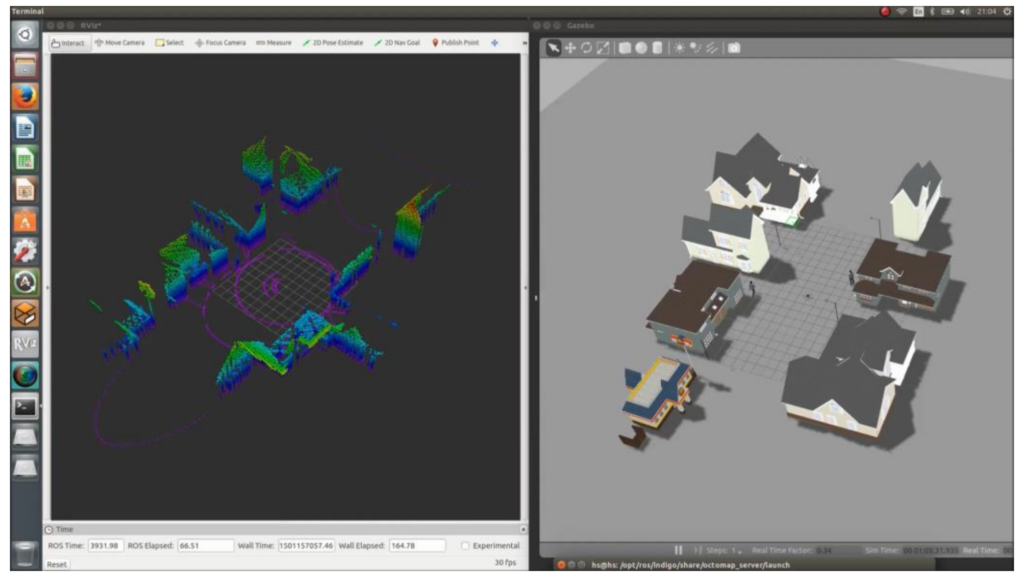
The Map

Distance measurement: d

Known state: (x_1, x_2, θ)

$$\begin{bmatrix} x_{1,occ} \\ x_{2,occ} \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} d \\ 0 \end{bmatrix} + \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$



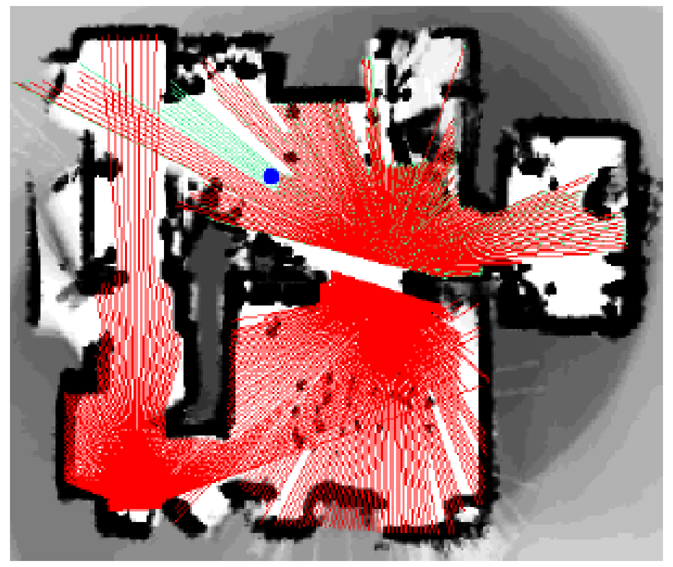


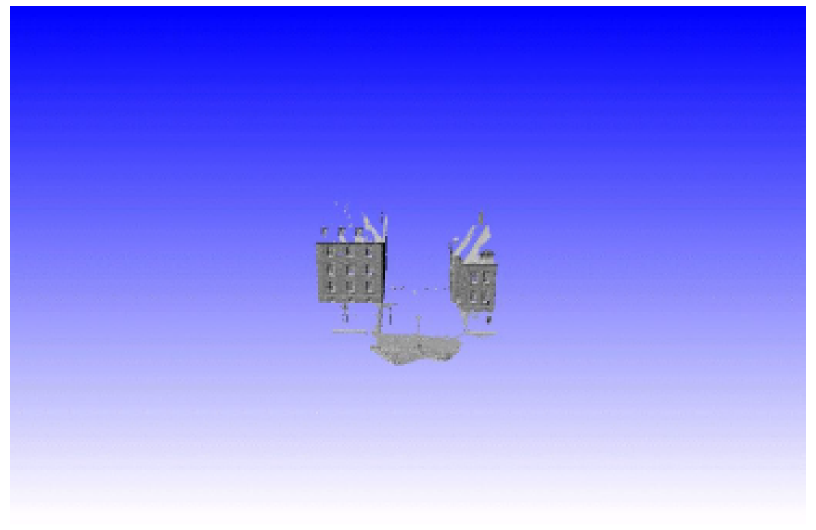
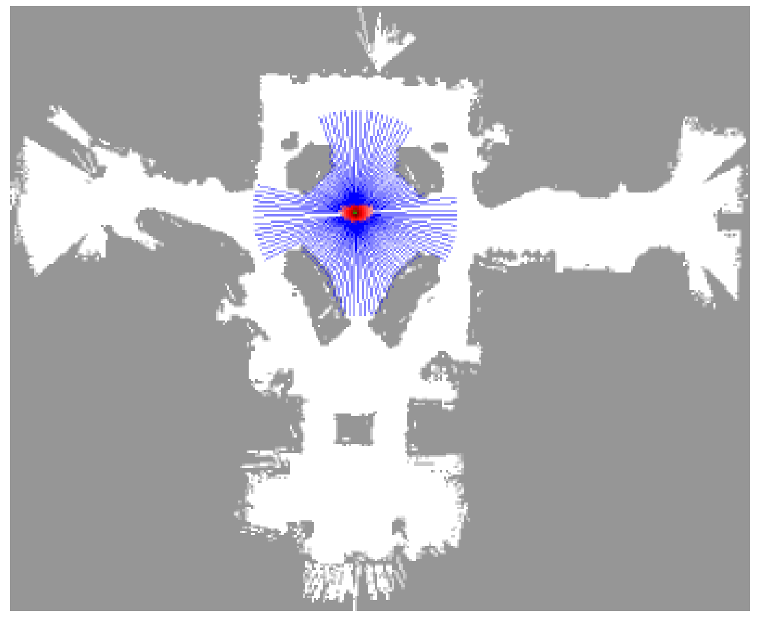
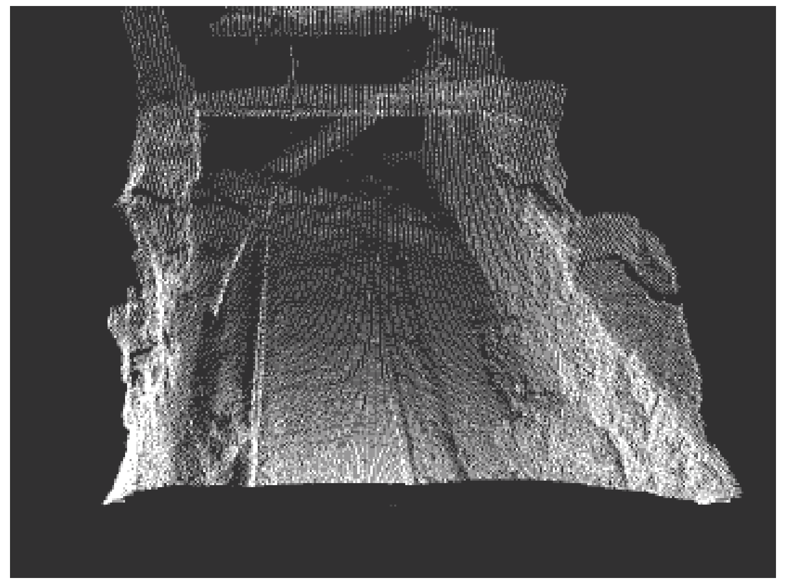
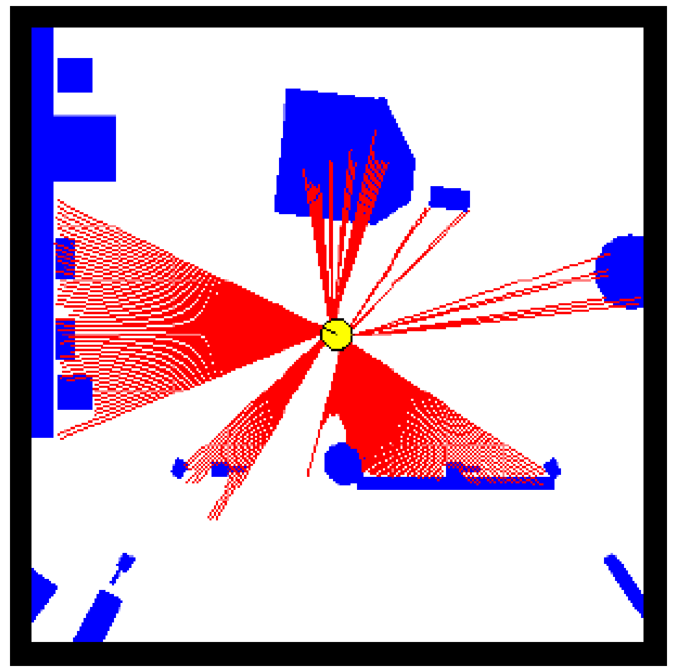
Robots Equipped with Laser Scanners

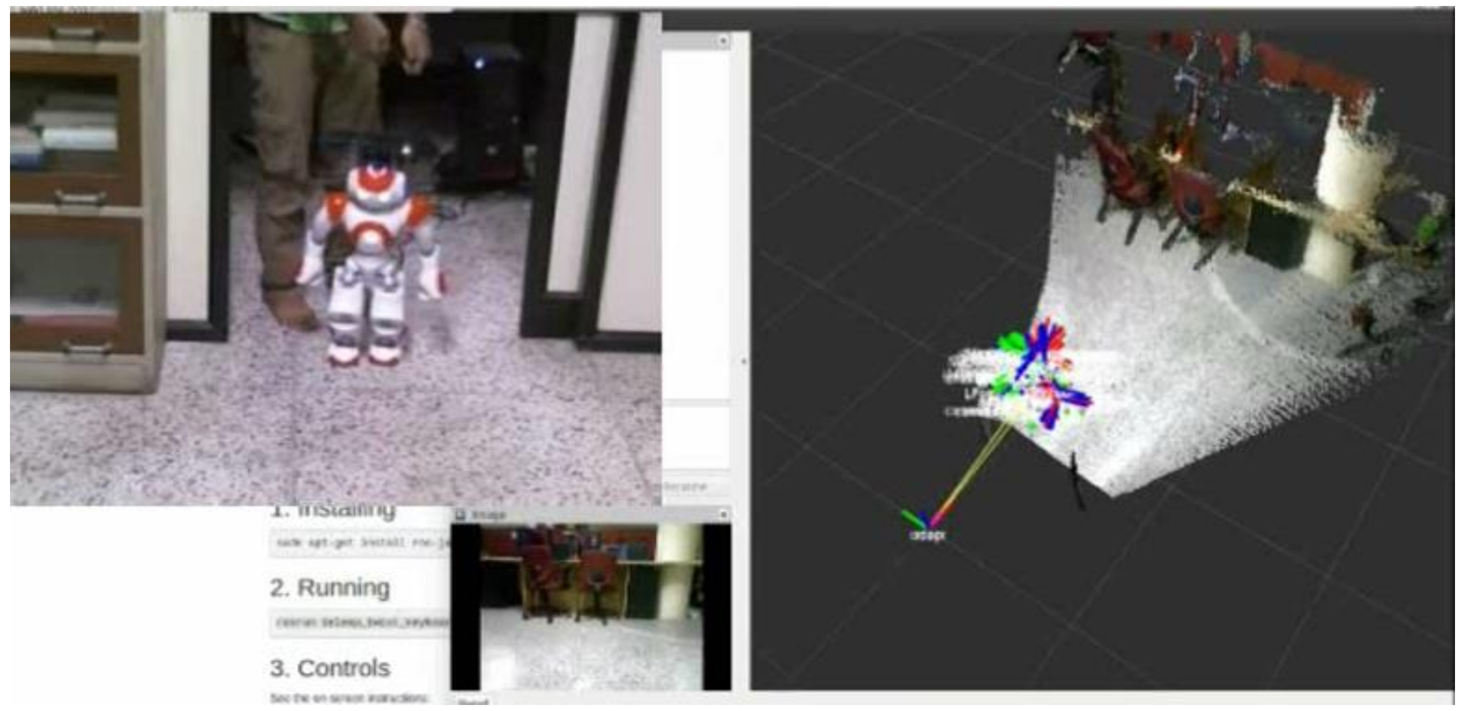


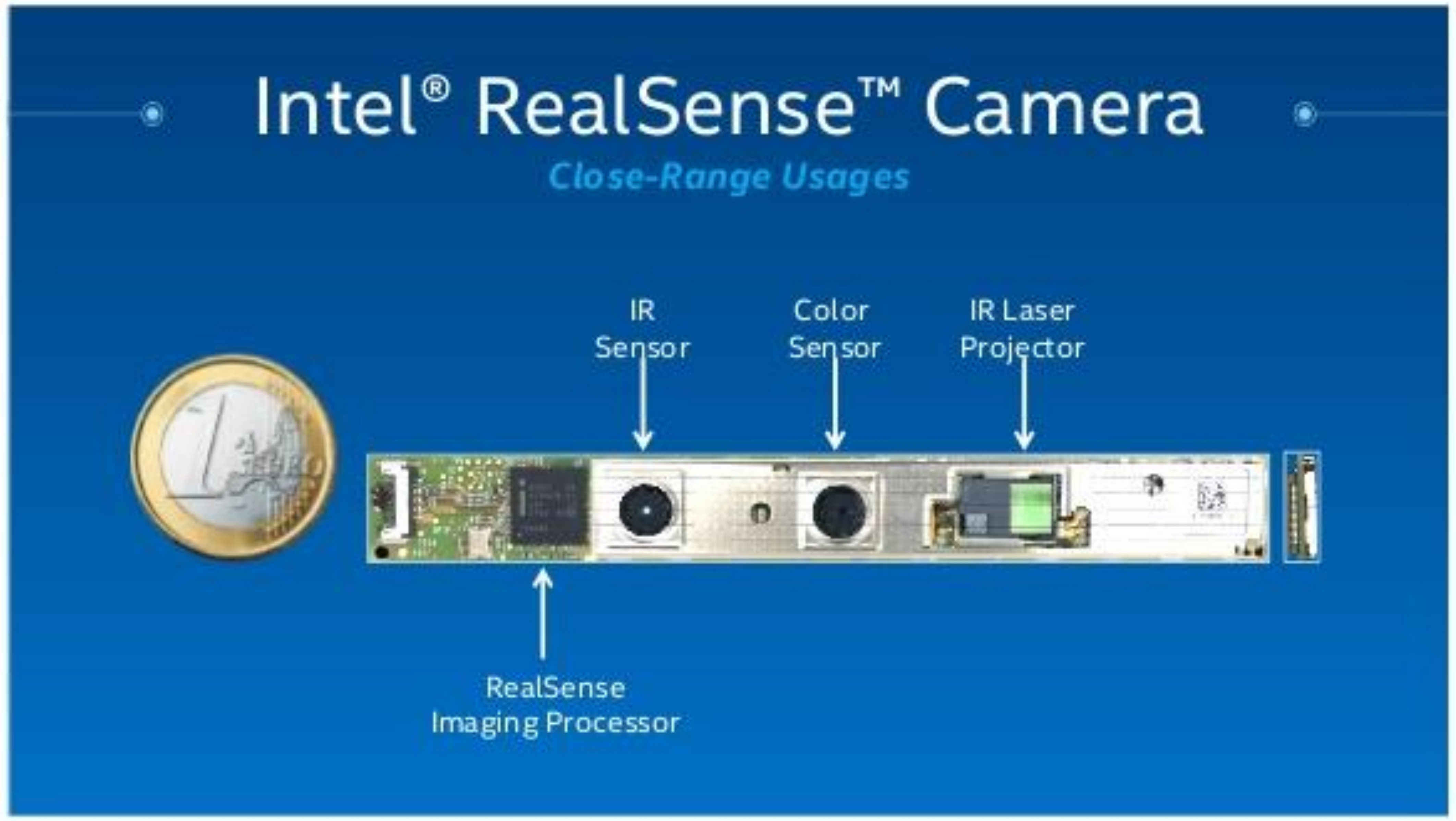


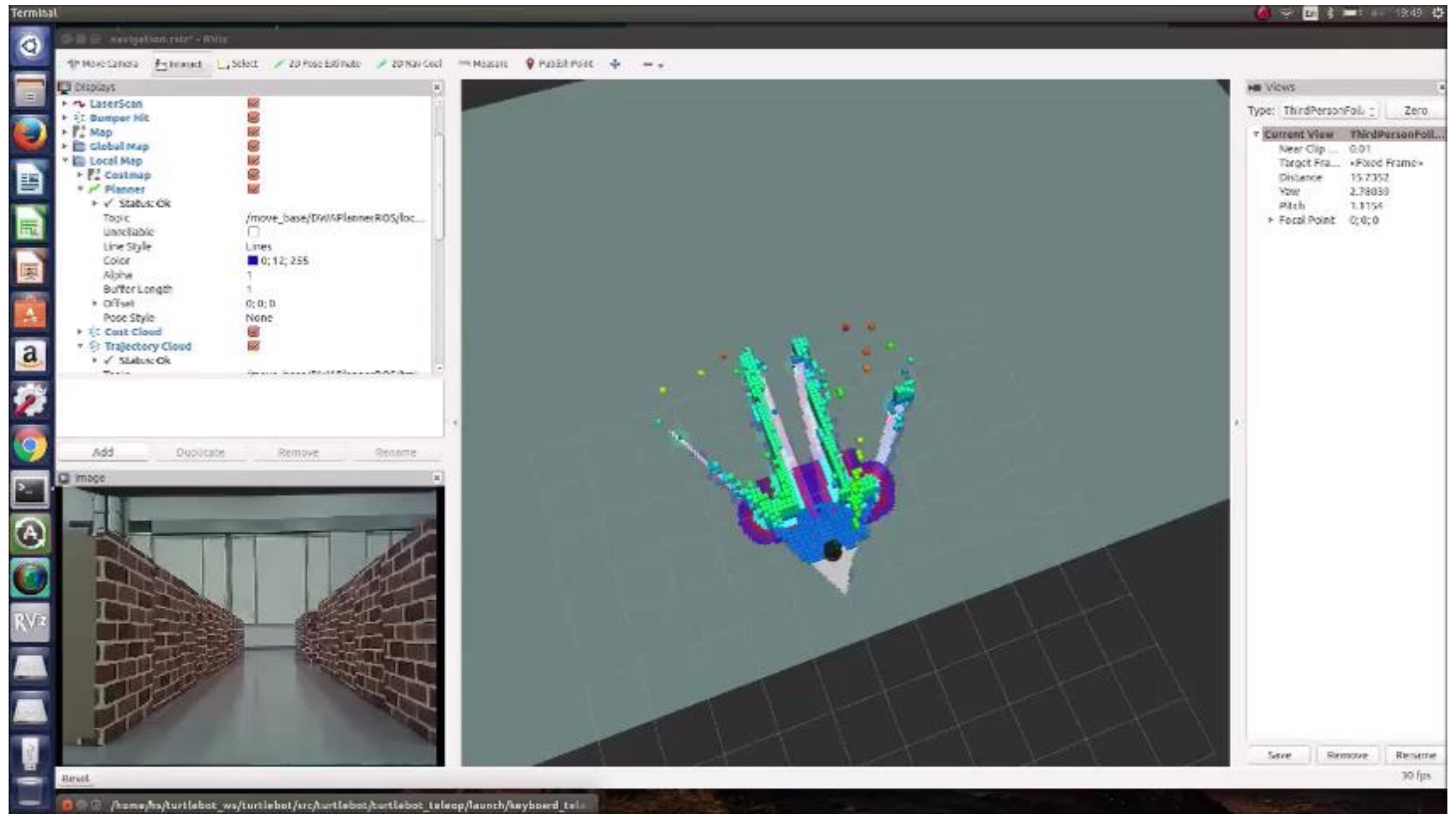

Typical Scans



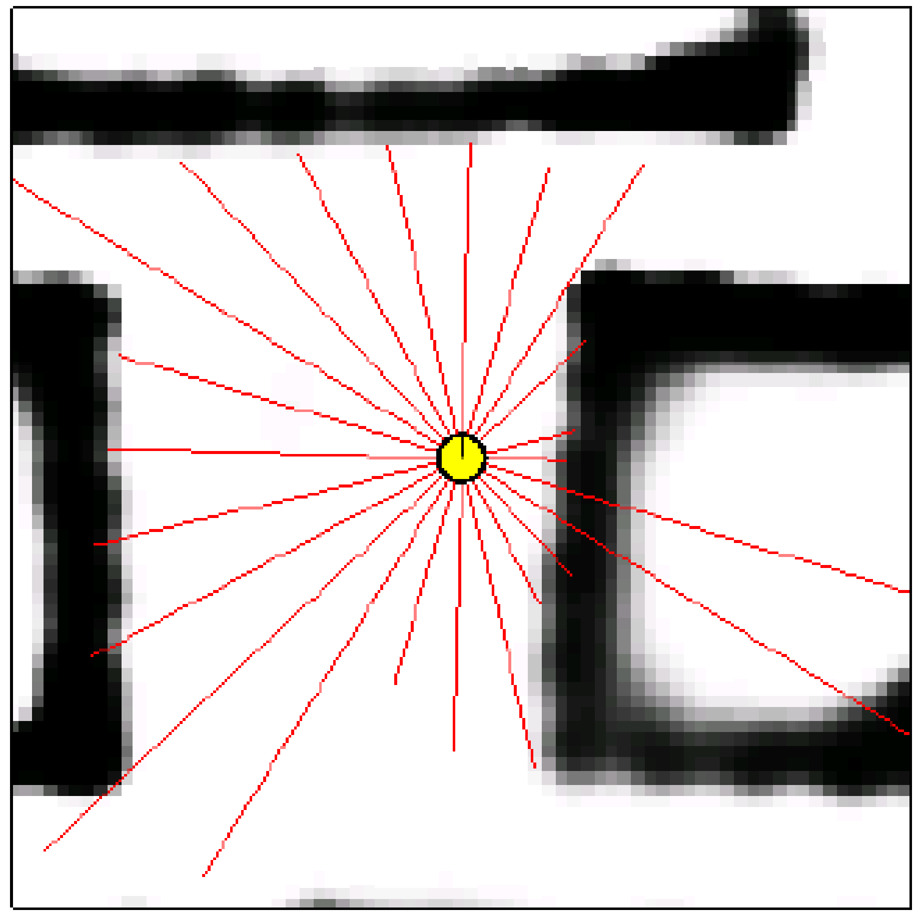
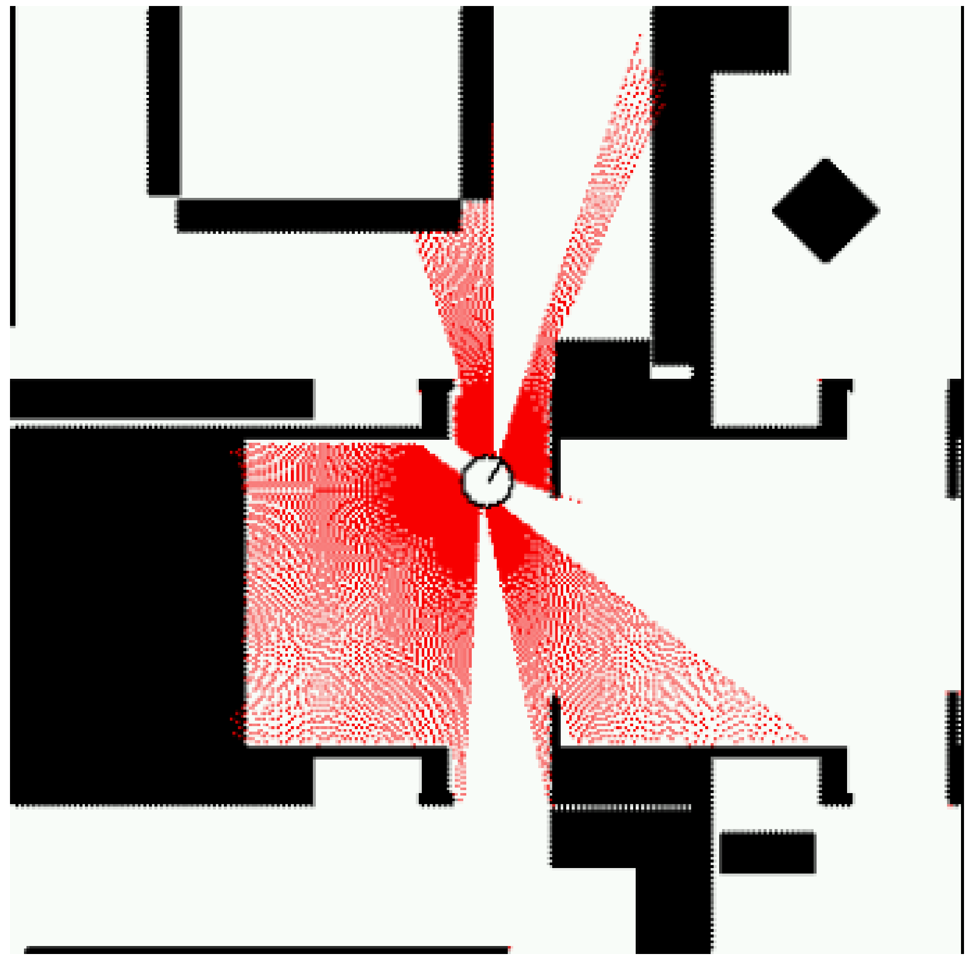
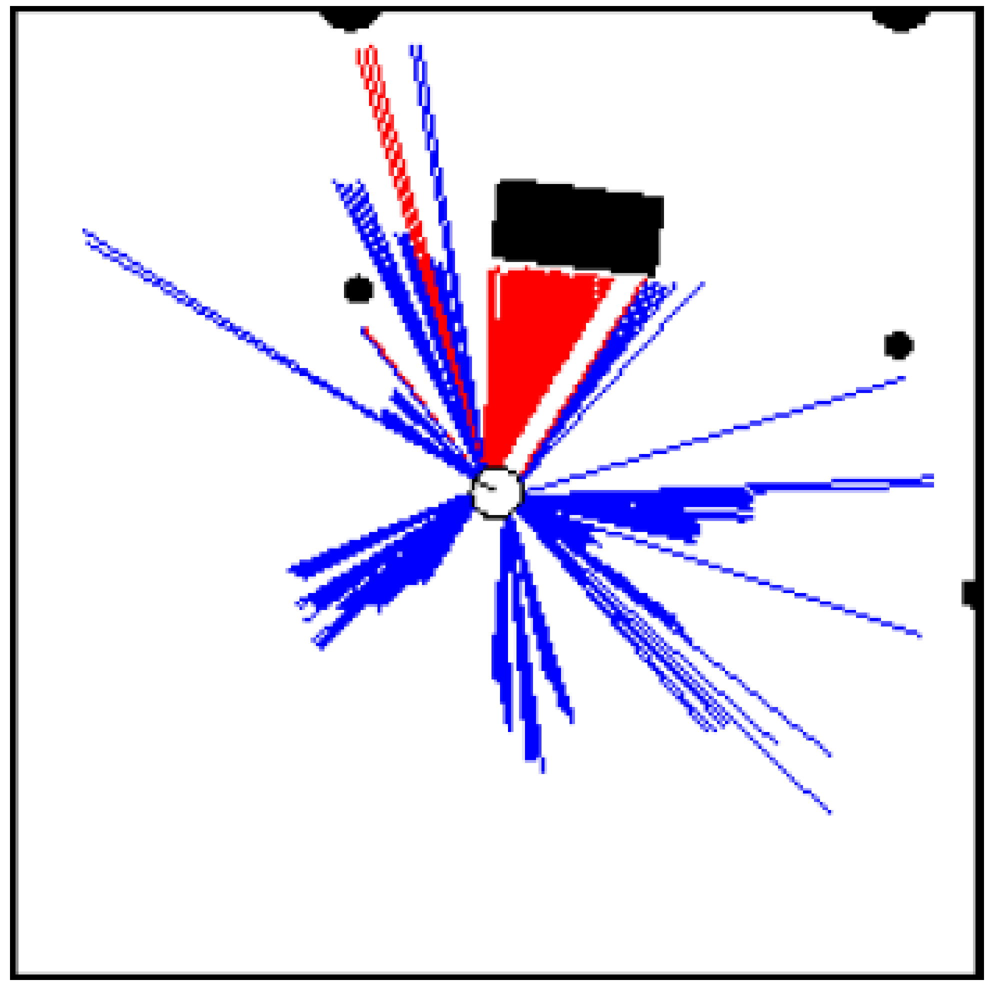






2. Running
3. Controls


Intel® RealSense™ Camera
Close-Range Usages
IR Sensor
Color Sensor
IR Laser Projector
RealSense Imaging Processor



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.

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

- Individual measurements are independent given the robot position.

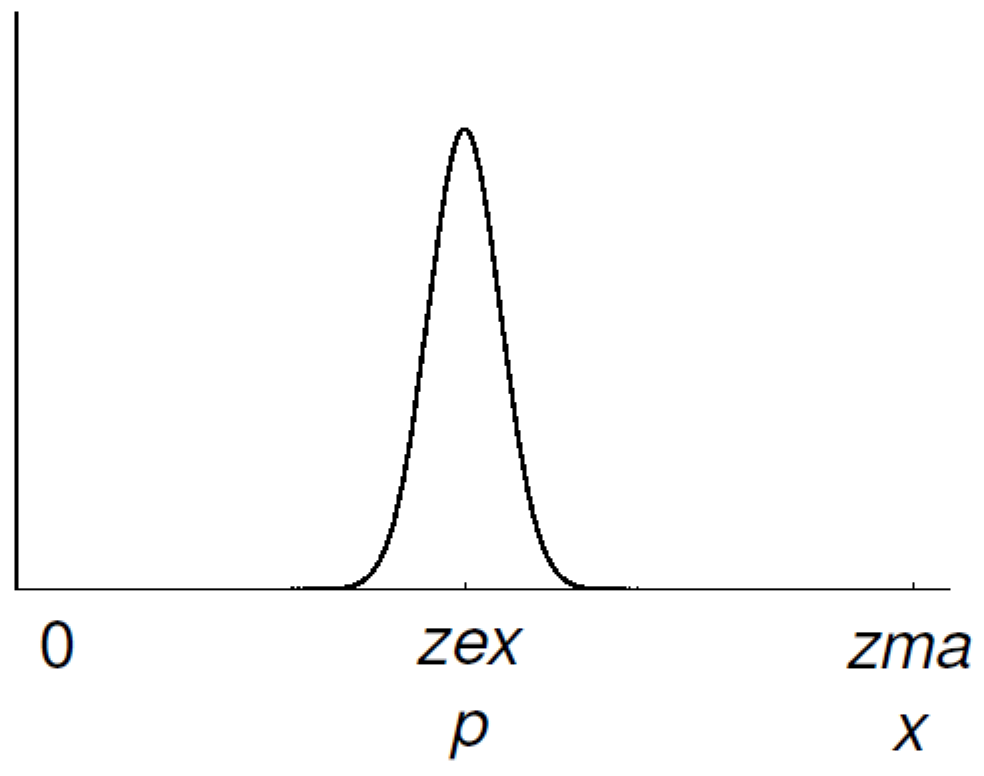
$$P(z \mid x, m) = \prod_{k=1}^{K} P(z_k \mid x, m)$$

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.

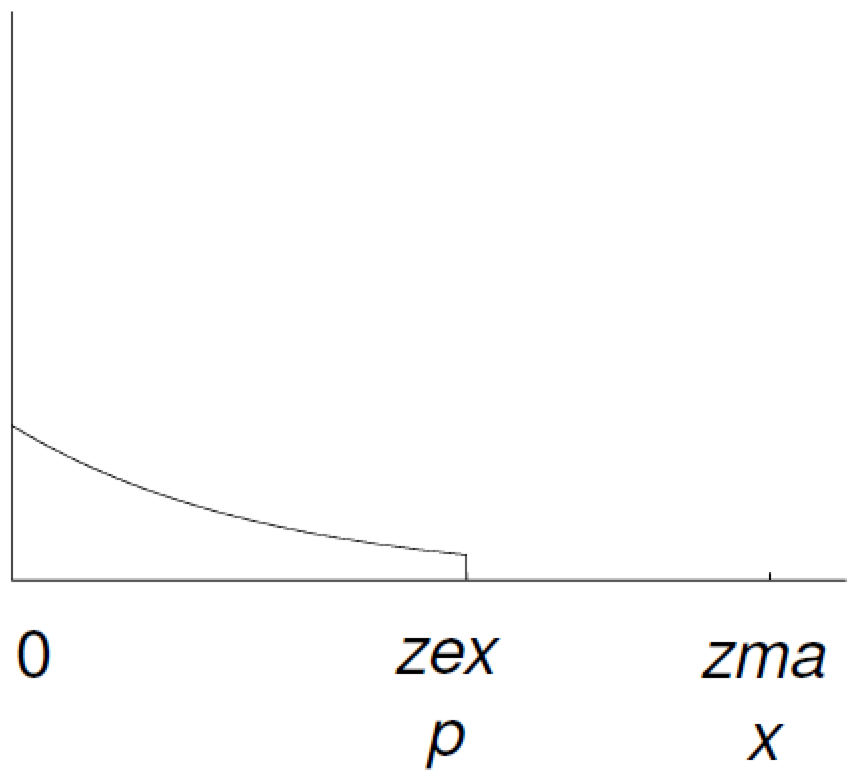
Beam-based Proximity Model

Measurement noise



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

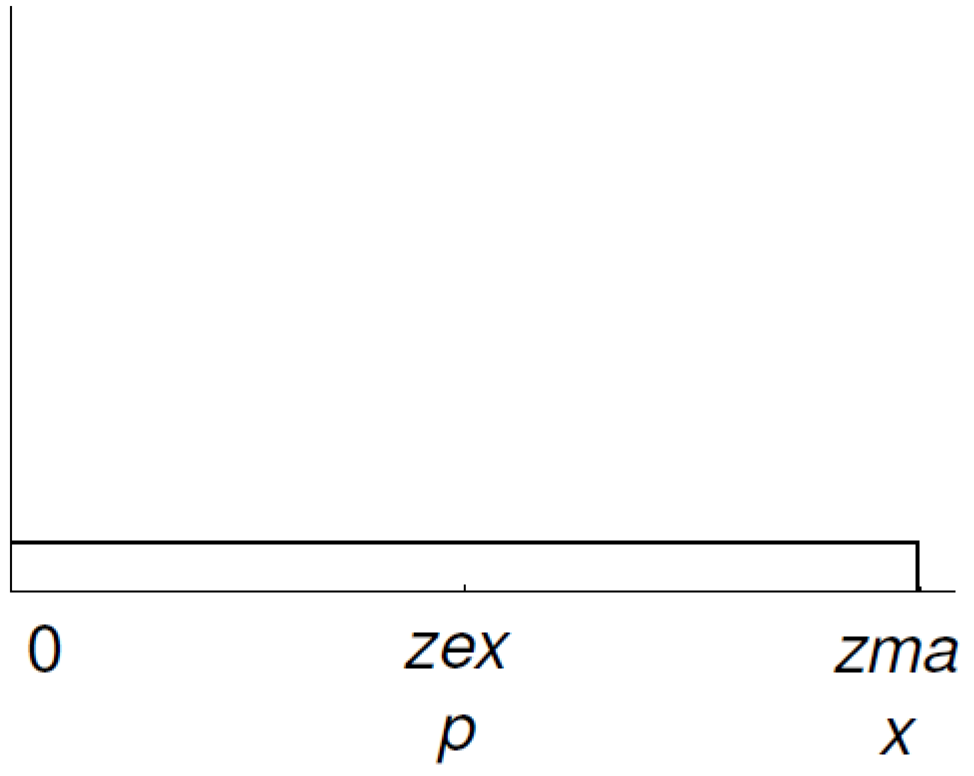
Unexpected obstacles



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

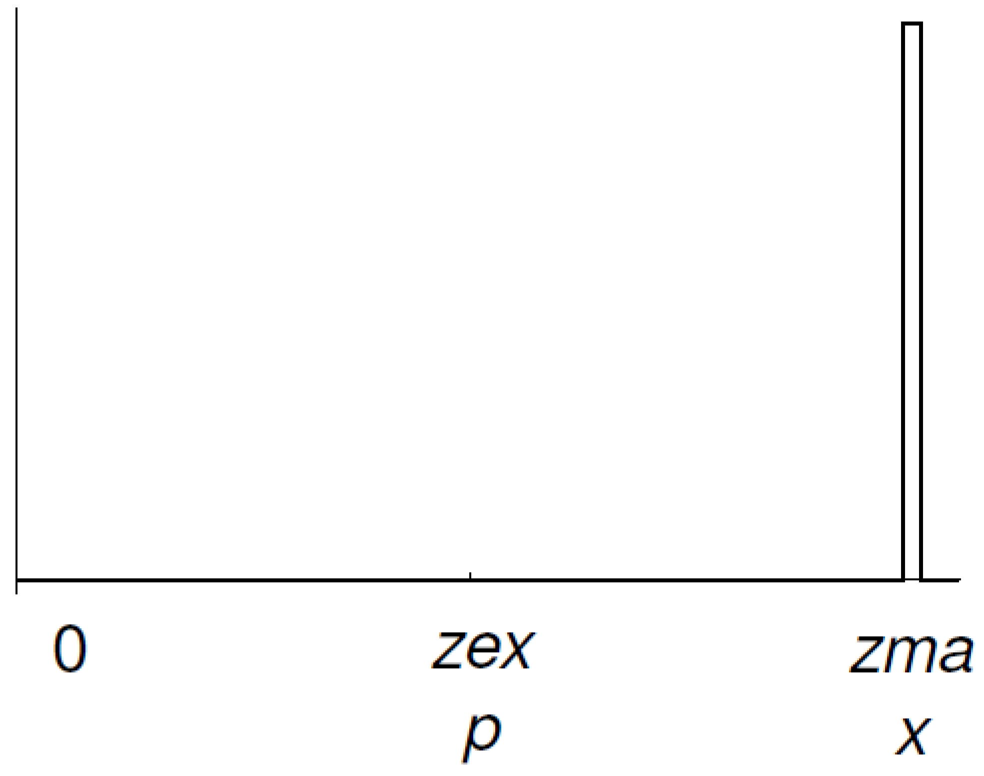
Beam-based Proximity Model

Random measurement



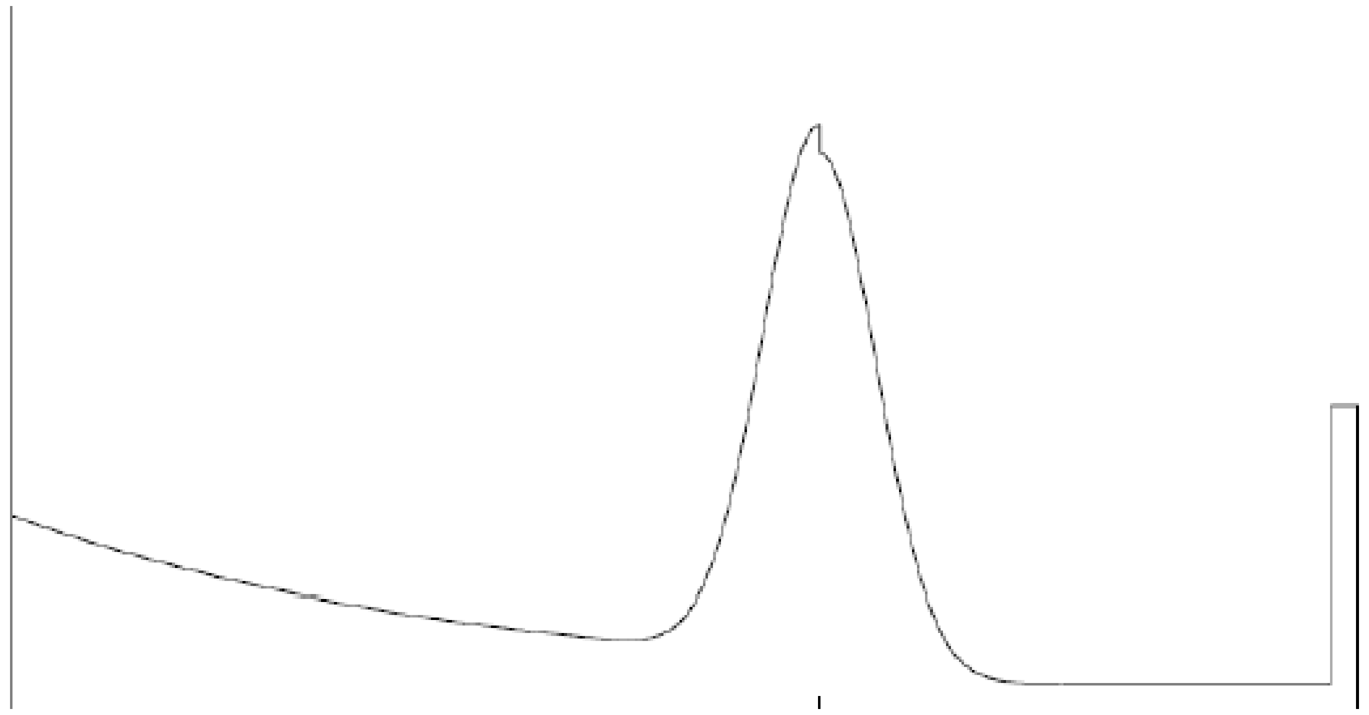
$$P_{rand}(z \mid x, m) = \eta \frac{1}{z_{\max}}$$

Max range



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

Resulting Mixture Density

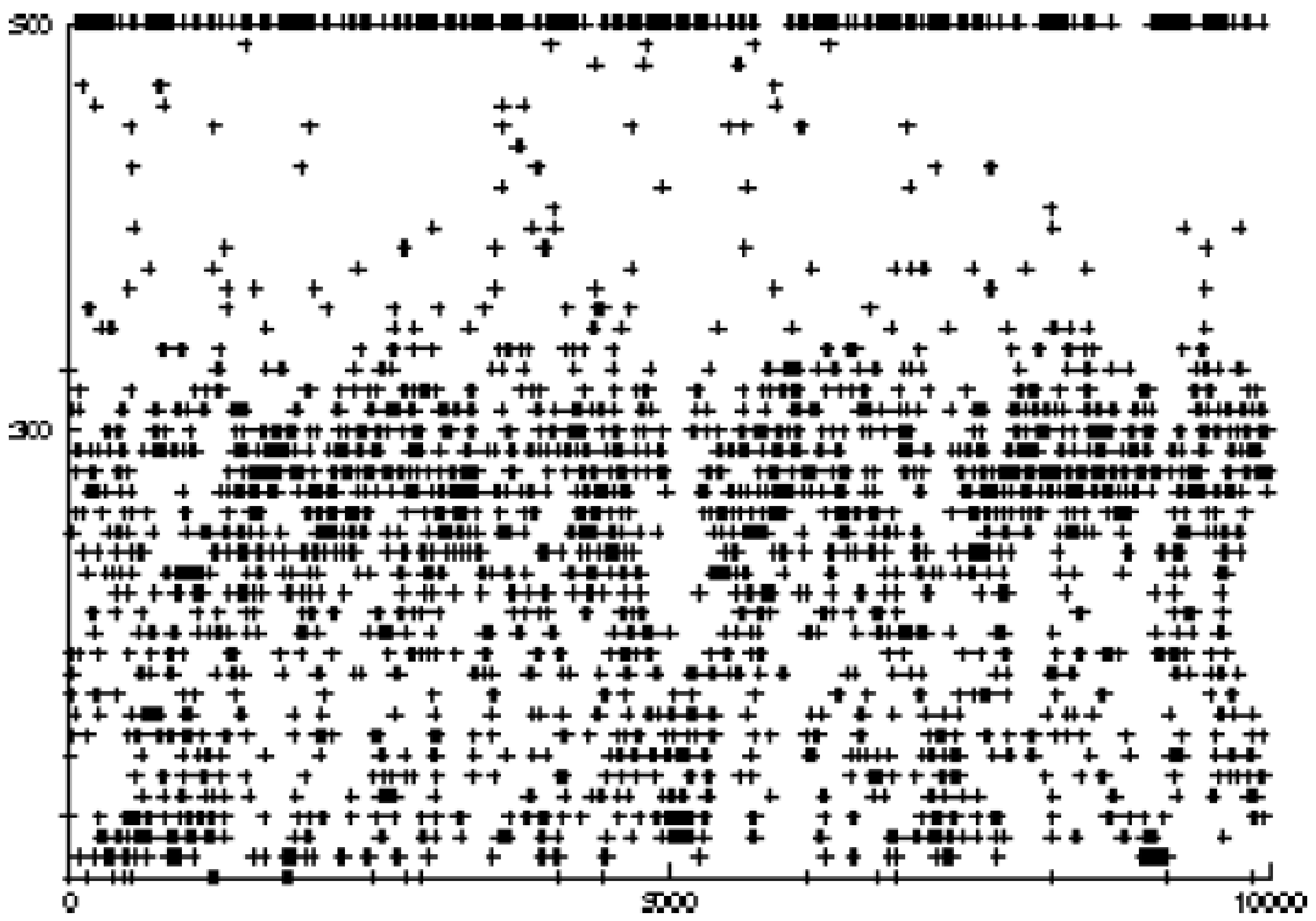


$$P(z \mid x, m) = \begin{pmatrix} \alpha_{\mathrm{hit}} \\ \alpha_{\mathrm{unexp}} \\ \alpha_{\max} \\ \alpha_{\mathrm{rand}} \end{pmatrix}^T \cdot \begin{pmatrix} P_{\mathrm{hit}}(z \mid x, m) \\ P_{\mathrm{unexp}}(z \mid x, m) \\ P_{\max}(z \mid x, m) \\ P_{\mathrm{rand}}(z \mid x, m) \end{pmatrix}$$

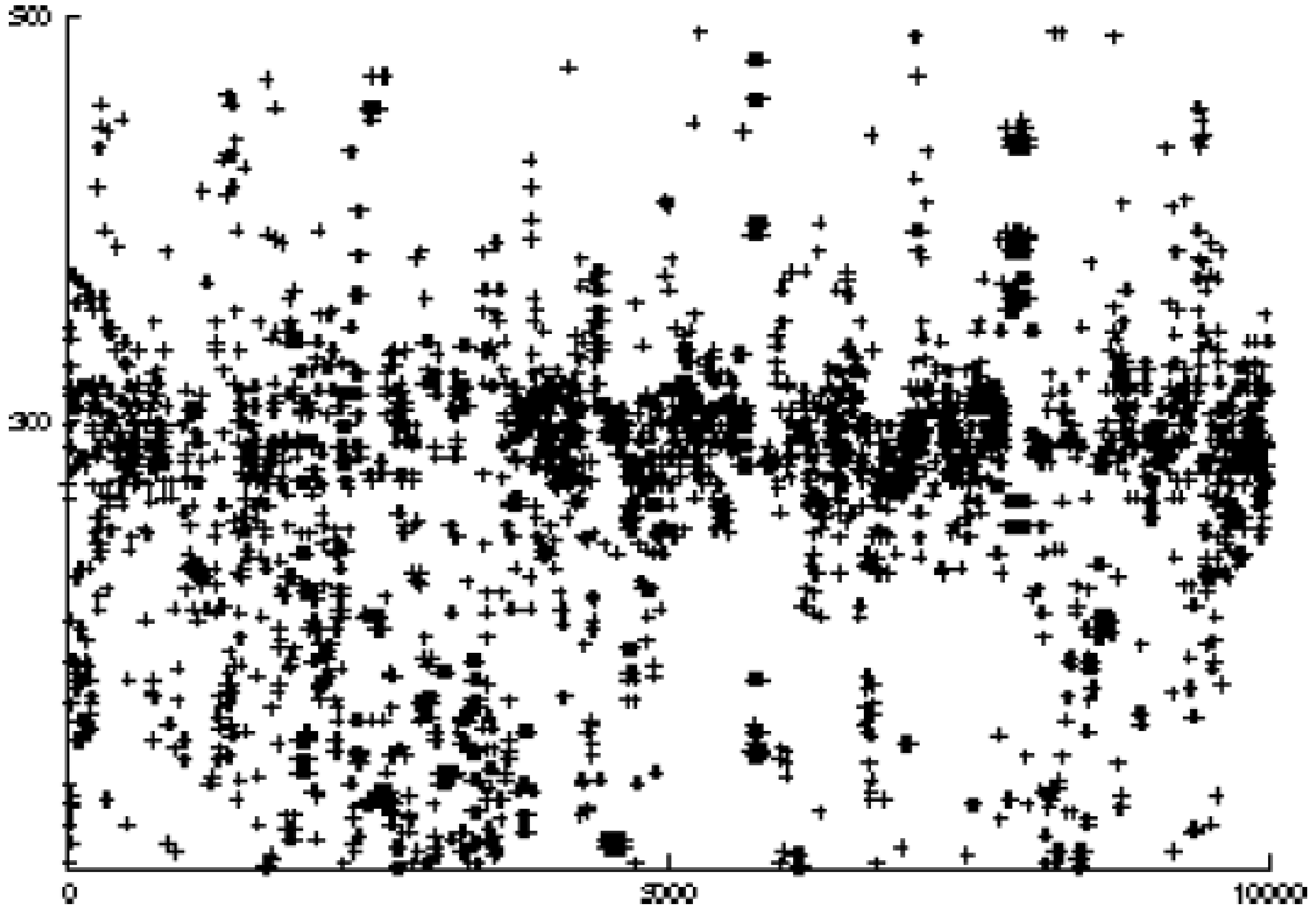
How can we determine the model parameters?

Raw Sensor Data

Measured distances for expected distance of 300 cm.



Sonar



Laser

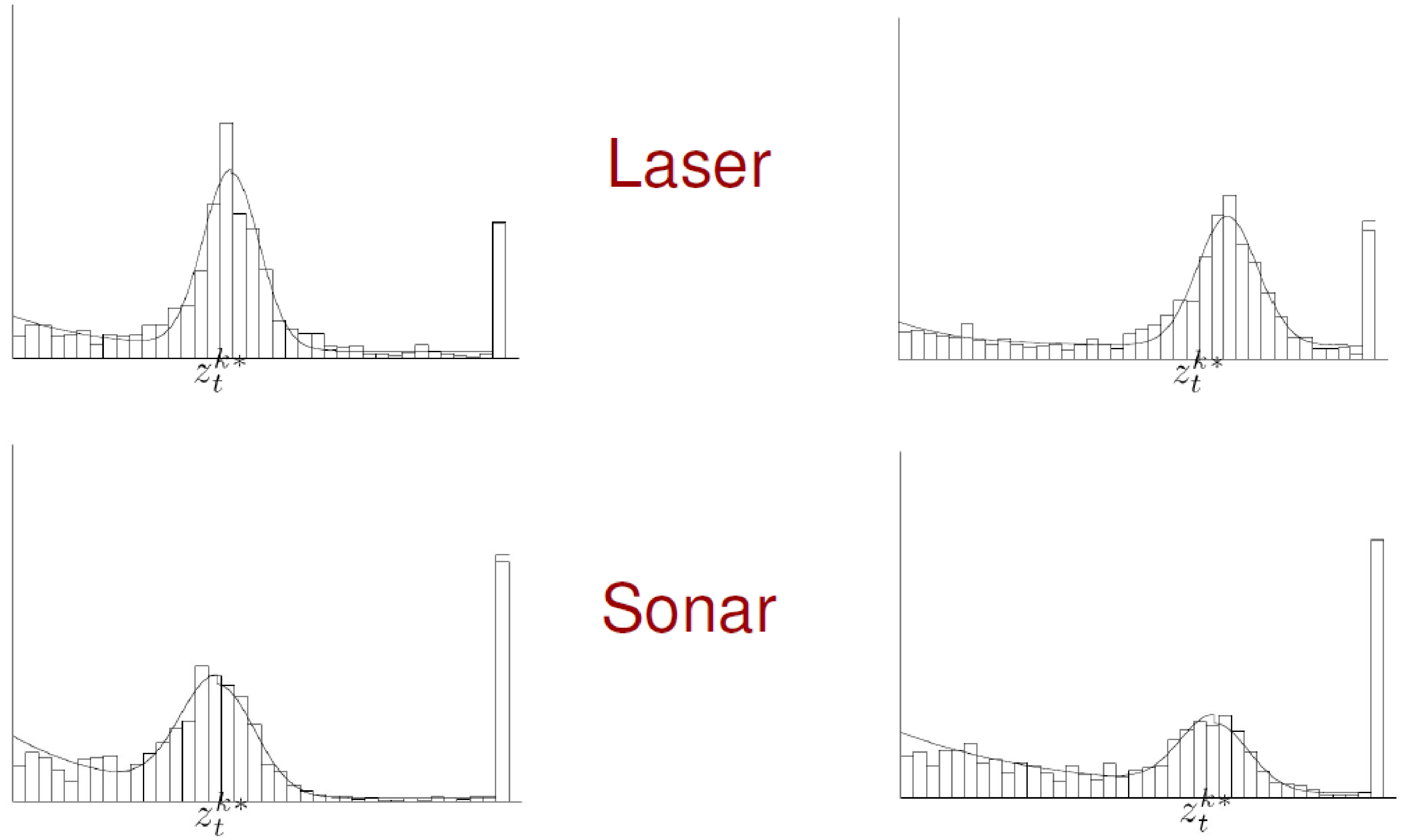
Approximation

- Maximize log likelihood of the data

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

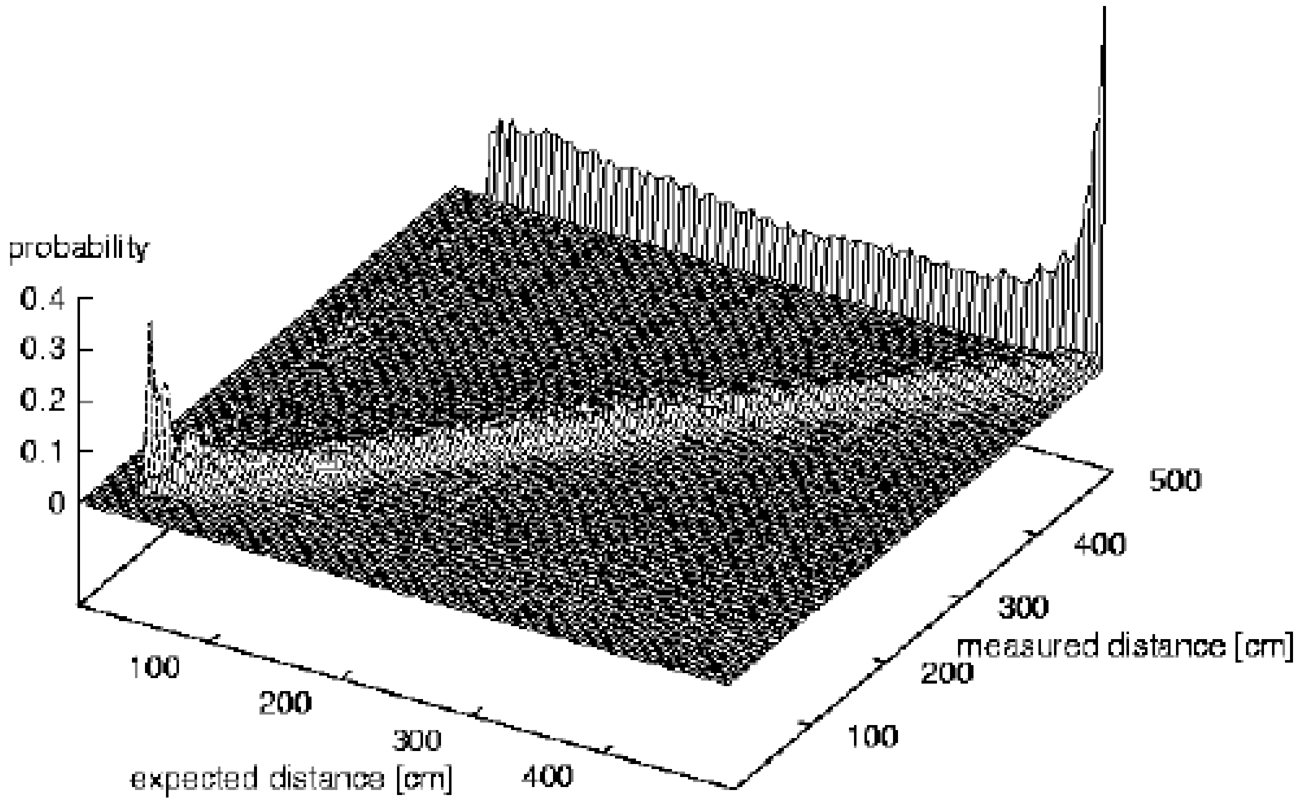
- Search space of n-1 parameters.
  - Hill climbing
  - Gradient descent
  - Genetic algorithms
  - …
- Deterministically compute the n-th parameter to satisfy normalization constraint.

Approximation Results

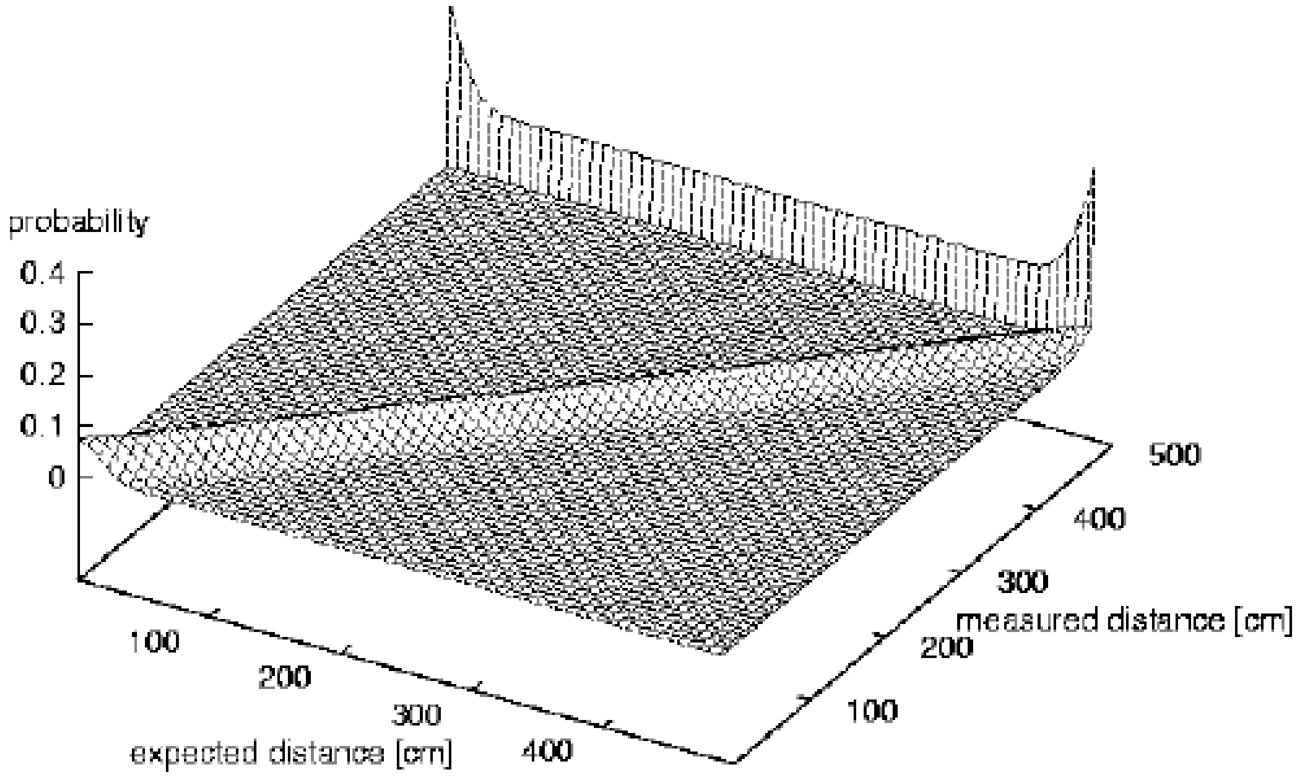


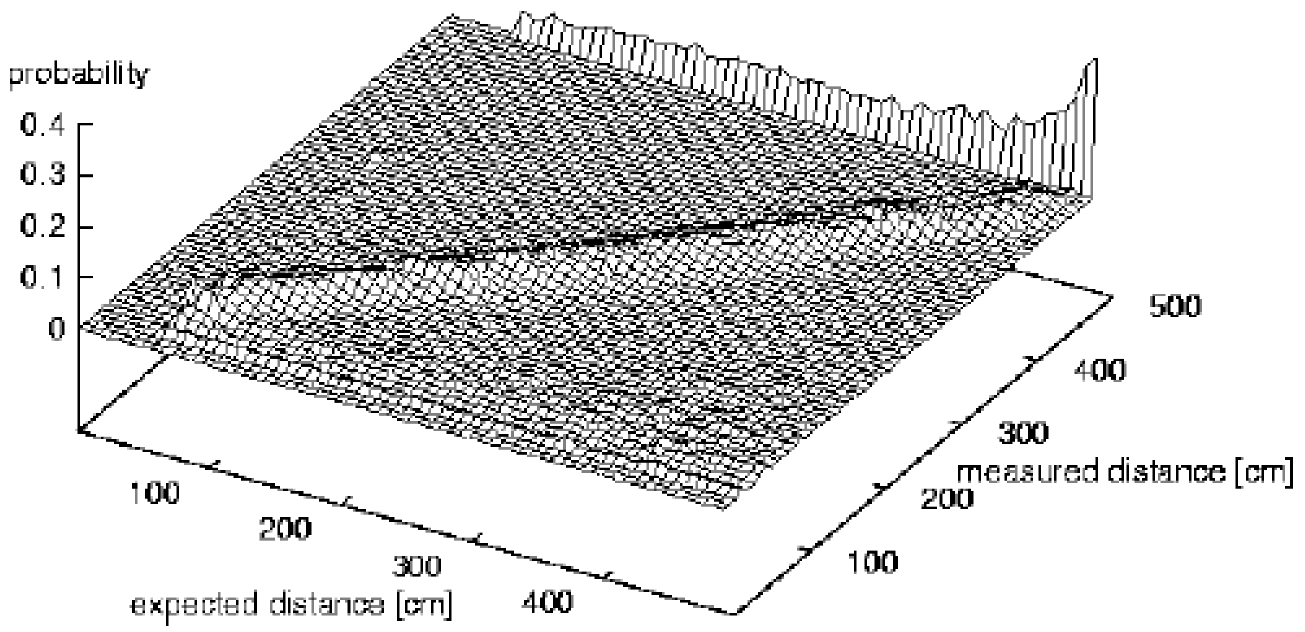
300cm 400cm

Approximation Results

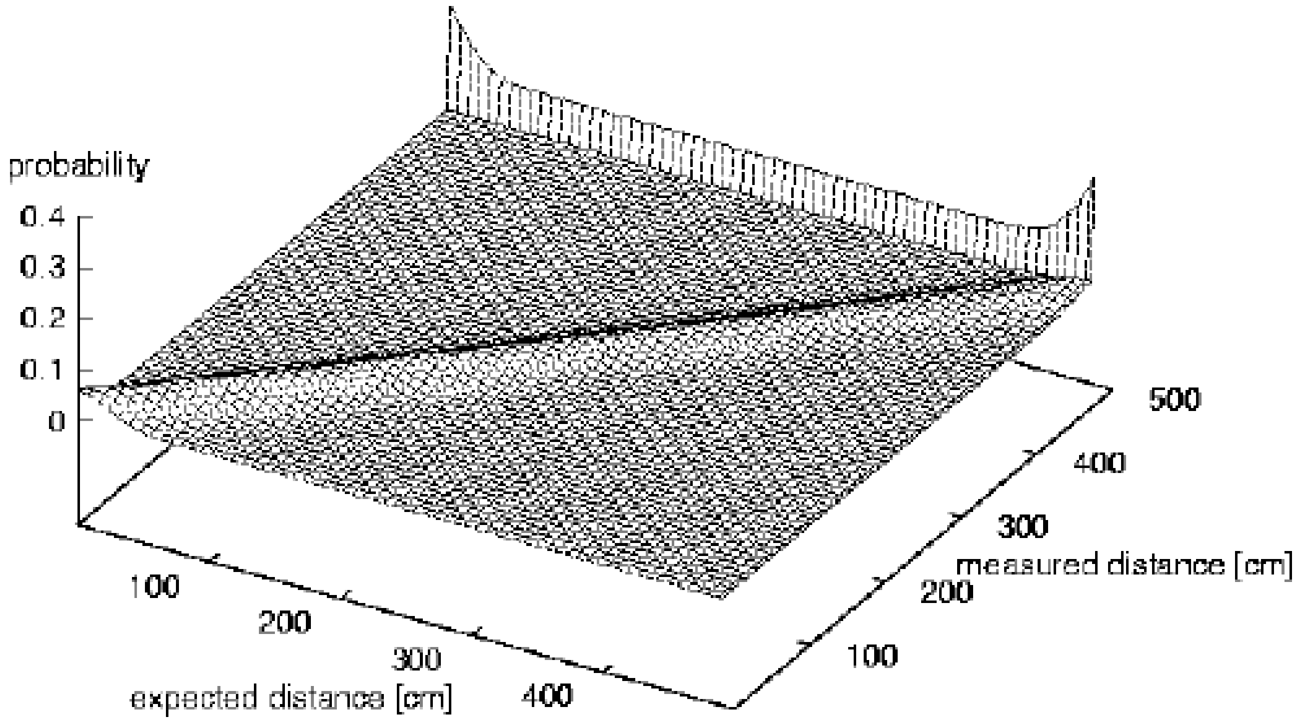


Laser





Sonar



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.

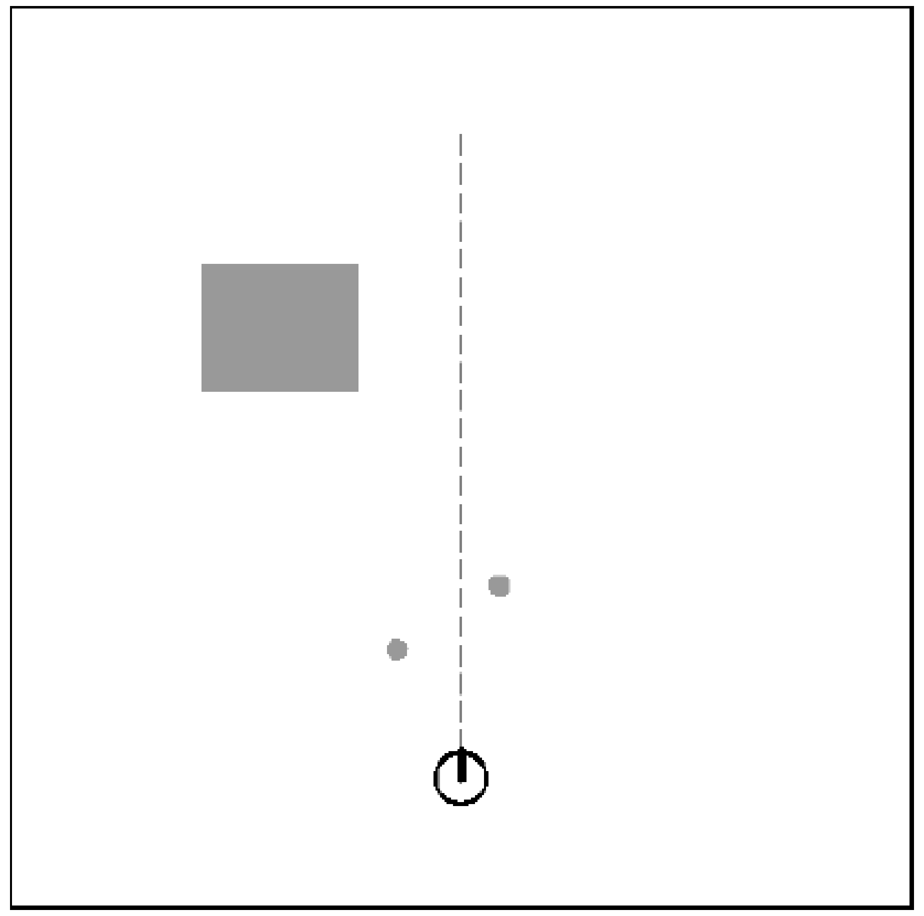
- Idea: Instead of following along the beam, just check the end point.

Scan-based Model

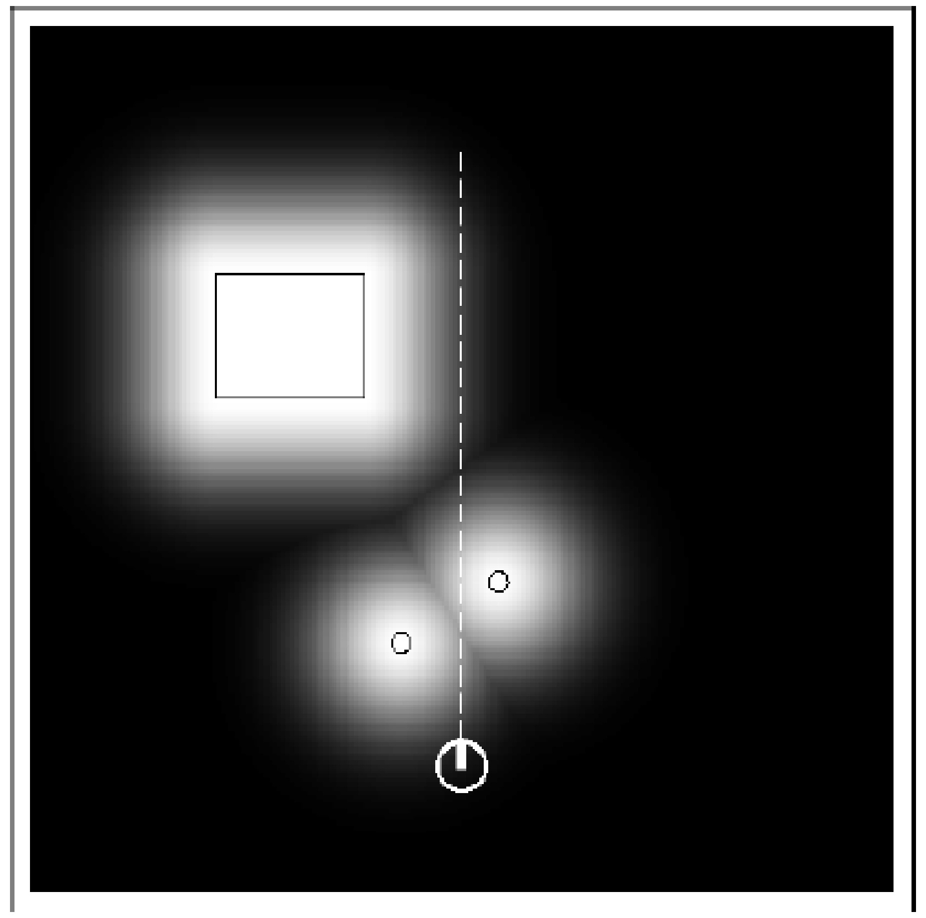
- Probability is a mixture of …
  - a Gaussian distribution with mean at distance to closest obstacle,
  - a uniform distribution for random measurements, and
  - a small uniform distribution for max range measurements.
- Again, independence between different components is assumed.

The likelihood of an obstacle detection as a function of global x-y-coordinates

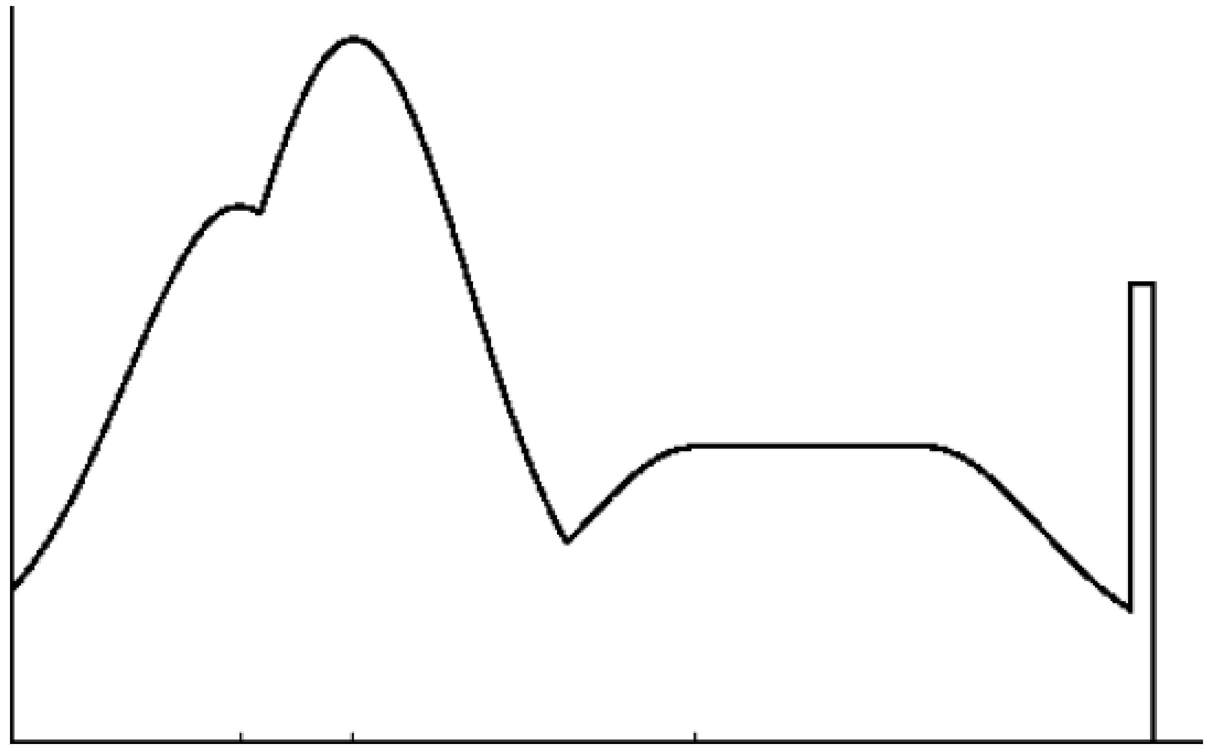
Example



Map m



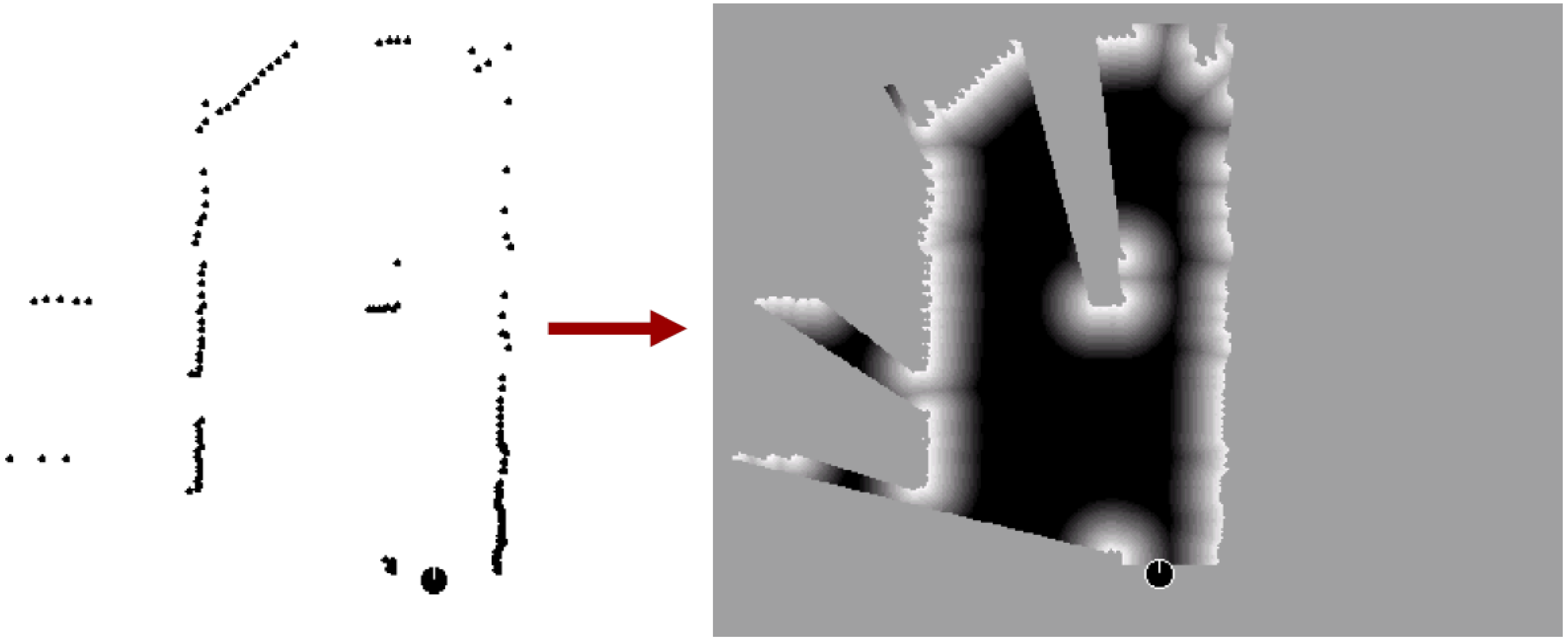
Likelihood field



$P(z|x,m)$

Scan Matching

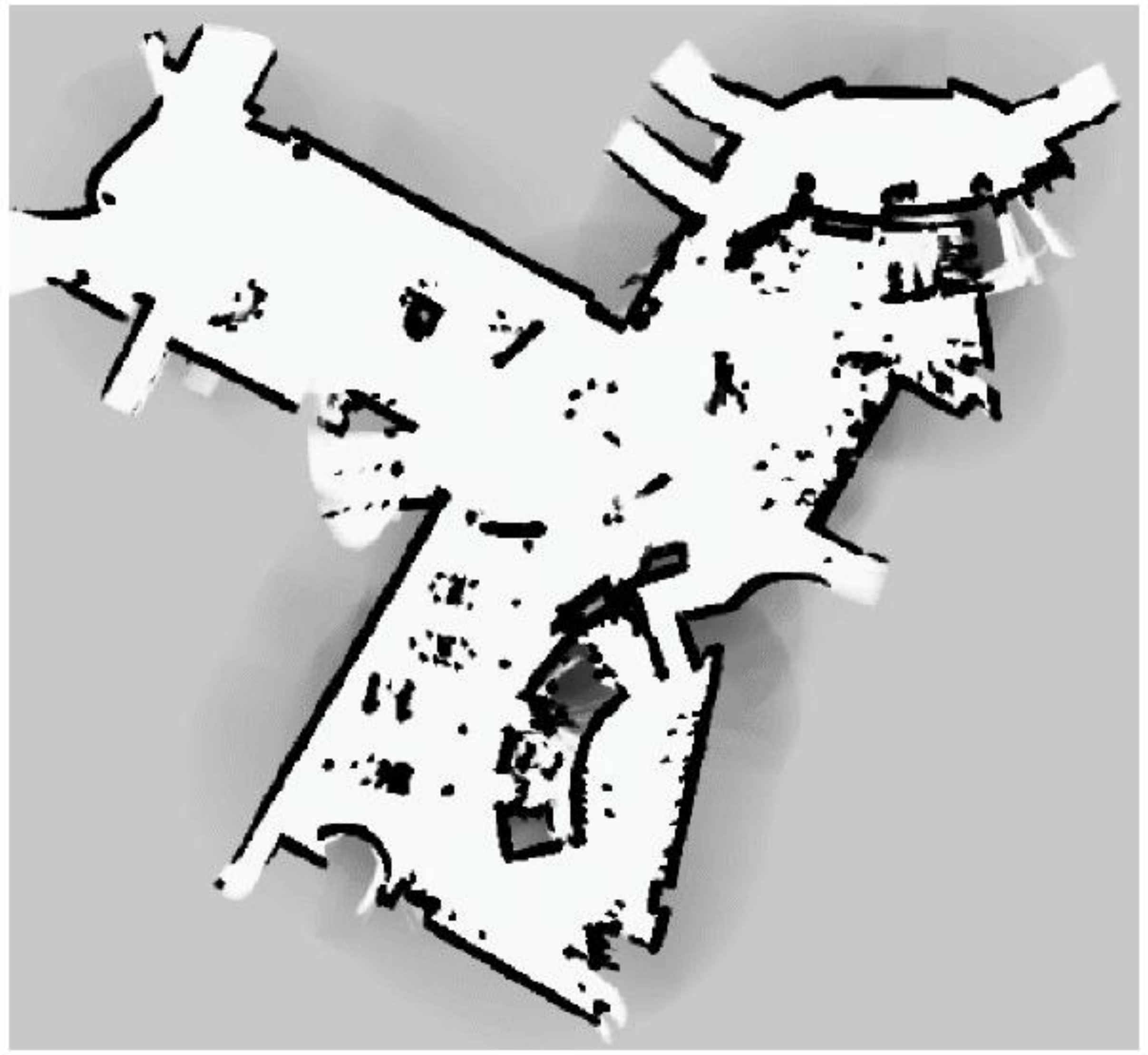
- Extract likelihood field from scan and use it to match different scan.



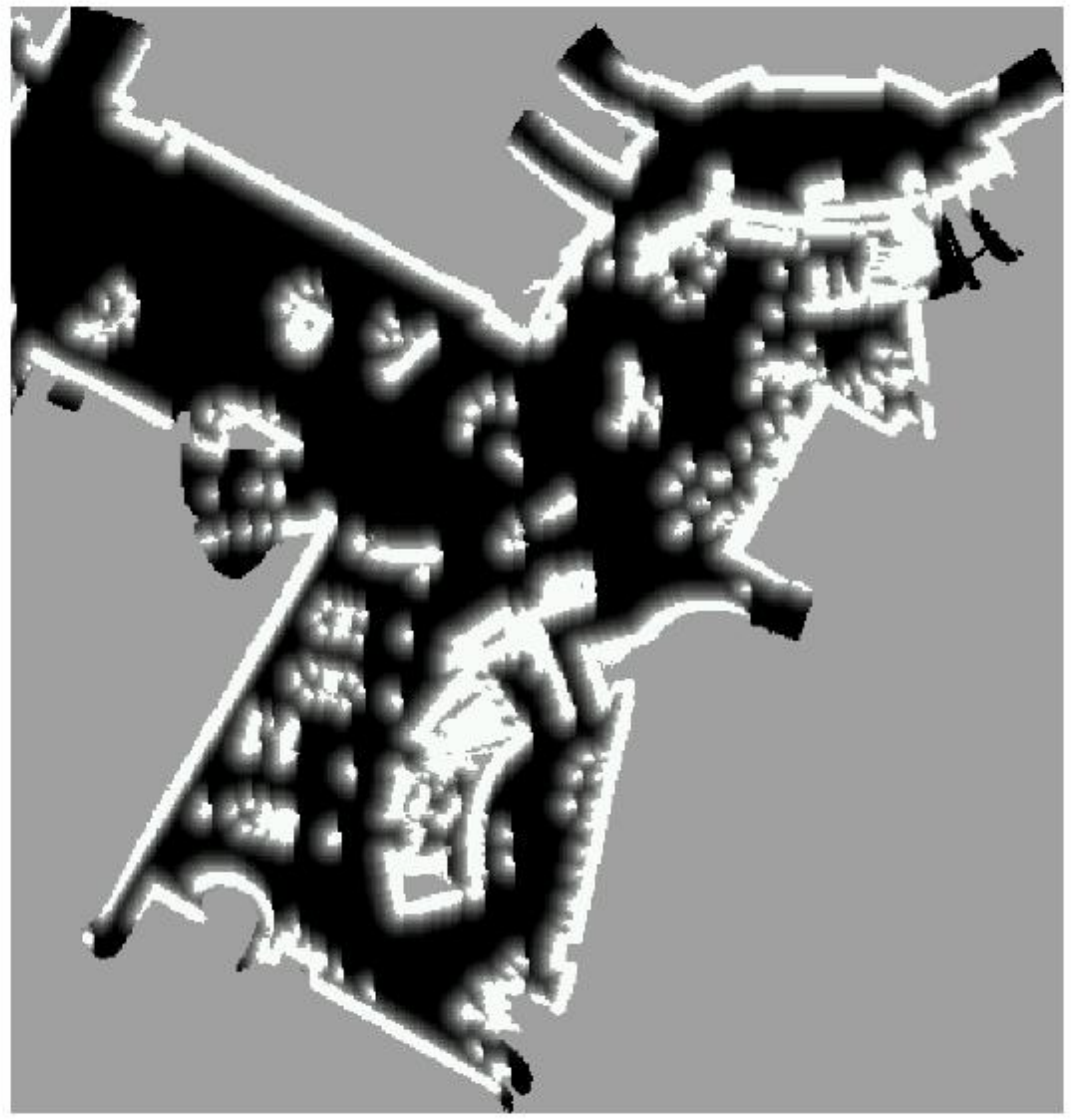
Properties of Scan-based Model

- Highly efficient, uses 2D tables only.
- Smooth w.r.t. to small changes in robot position.
- Allows gradient descent, scan matching.
- Ignores physical properties of beams.
- Will it work for ultrasound sensors?

San Jose Tech Museum



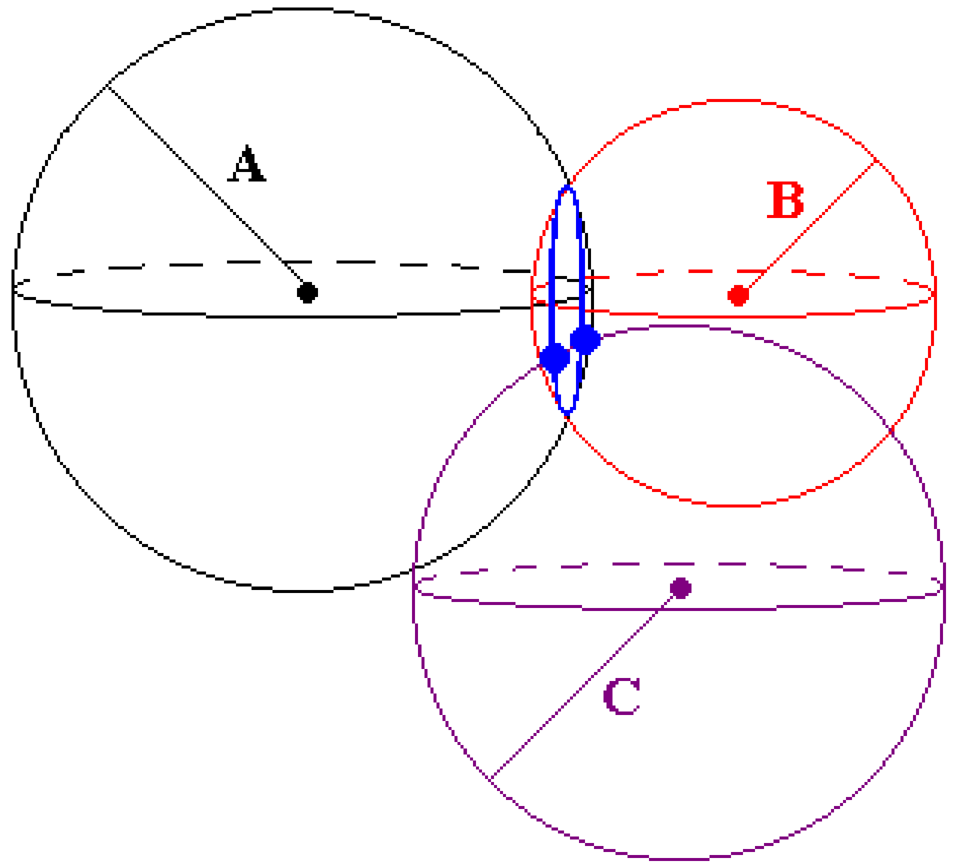
Occupancy grid map



Likelihood field

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.

$$
\begin{pmatrix} r_t^i \\ \phi_t^i \\ s_t^i \end{pmatrix} = \begin{pmatrix} \sqrt{(m_{j,x}-x)^2+(m_{j,y}-y)^2} \\ \operatorname{atan2}(m_{j,y}-y, m_{j,x}-x)-\theta \\ s_j \end{pmatrix} + \begin{pmatrix} \varepsilon_{\sigma_r^2} \\ \varepsilon_{\sigma_\phi^2} \\ \varepsilon_{\sigma_s^2} \end{pmatrix}
$$

Algorithm landmark_model_known_correspondence(f_t^i, c_t^i, x_t, m)**:**

$j = c_t^i$

$\hat{r} = \sqrt{(m_{j,x}-x)^2+(m_{j,y}-y)^2}$

$\hat{\phi} = \operatorname{atan2}(m_{j,y}-y, m_{j,x}-x)$

$q = \mathbf{prob}(r_t^i - \hat{r}, \sigma_r^2) \cdot \mathbf{prob}(\phi_t^i - \hat{\phi}, \sigma_\phi^2) \cdot \mathbf{prob}(s_t^i - s_j, \sigma_s^2)$

return q

Table 6.4 Algorithm for computing the likelihood of a landmark measurement. The algorithm requires as input an observed feature $f_t^i = (r_t^i \ \ \phi_t^i \ \ s_t^i)^T$, and the true identity of the feature c_t^i, the robot pose $x_t = (x \ \ y \ \ \theta)^T$, and the map m. It's output is the numerical probability $p(f_t^i \mid c_t^i, m, x_t)$.

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!

ULSAN NATIONAL INSTITUTE OF SCIENCE AND TECHNOLOGY 2009

THANK YOU