Robot Sensing

Javier González Jiménez

Reference Books:

- Probabilistic Robotics. S. Thrun, W. Burgard, D. Fox. MIT Press. 2001
- Simultaneous Localization and Mapping for Mobile Robots: Introduction and Methods. Juan-Antonio Fernández-Madrigal and José Luis Blanco Claraco. IGI-Global. 2013.

Classification of Sensors

- Proprioceptive sensors :
  - Measure the internal status of the robot: battery, position, acceleration, inclination, …
  - Examples: shaft encoder, IMU (Inertial Measurement Unit): accelerometer + gyroscope, potentiometer, inclinometer, …
- Exteroceptive sensors
  - Gather information from the environment: distance and/or angle to objects, light intensity reflected by objects, …
  - Examples: cameras, laser scanner, RGB-D cameras, sonar, infrared, …

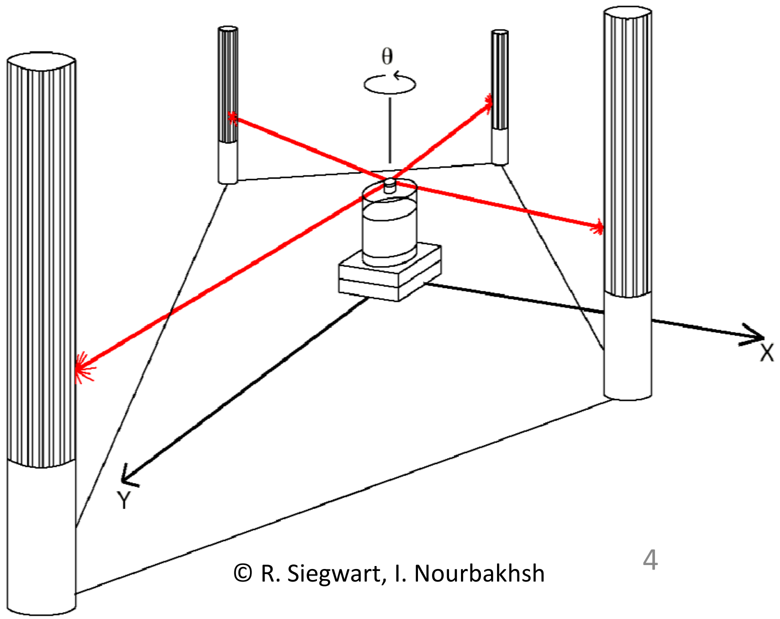
We will focus on these

Classification of Sensors

- Passive sensors
  - energy coming from the environment: e.g. camera
  - better range and coverage
- Active sensors
  - emit their own energy: e.g. GPS, laser scanner, sonar, …
  - better performance in changing-light conditions, but more power consumption

Beacons

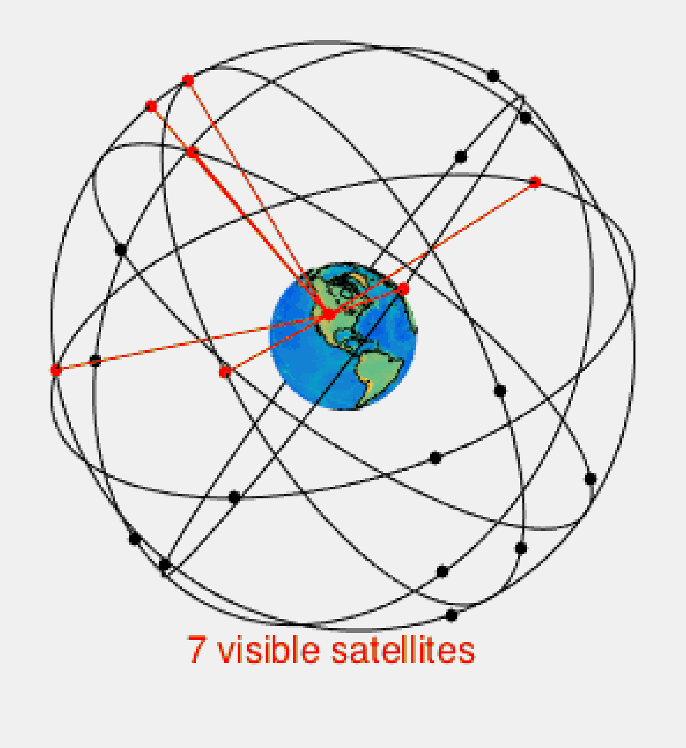
- Are navigation guiding devices with a precise known position (used since humans started to travel! e.g. lighthouses, stars, …)
- Natural beacons (called **landmarks**): distinctive features of the environment
- Artificial beacons: GPS, reflectors, radio antennas, WIFI, visual patterns (QR, datamatrix, ARUCOs, …)
  - **Advantages**:
    - Give absolute positioning (*wrt* the beacon positions)
    - Simple and efficient algorithms
  - **Drawbacks:**
    - Costly: require changes in the environment
    - Limit flexibility and adaptability to changing environments.

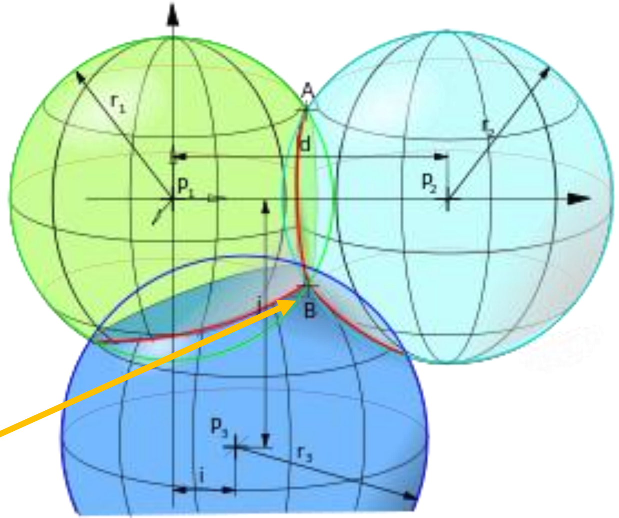


Global Positioning System (GPS)

Key system for **outdoor** mobile robotics

- Mobile, active artificial beacons
- Started in 1973, fully operational in 1995
- 31 satellites in 6 planes (4+ satellites per plane) orbiting the earth every 12 hours at 20.200 km.
- Based on the **trilateration method**: range-based positioning





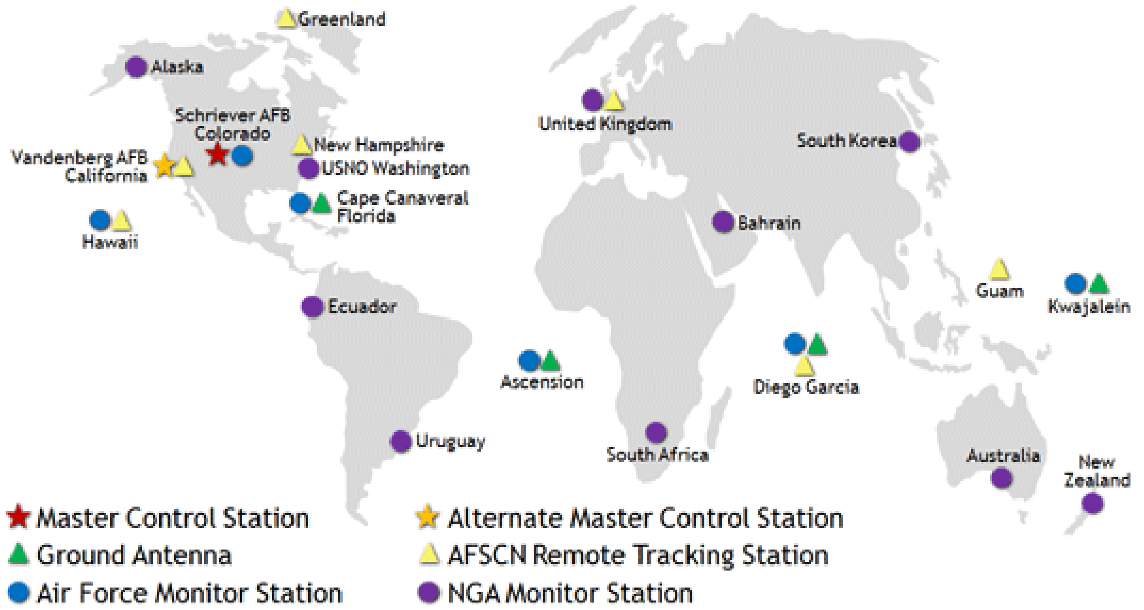
Interception of the 3 spheres

Triangulation vs. trilateration

- Triangulation: process to determine the location of a point by **measuring angles** to it from known points.
- Trilateration: same but **measuring distances** to the point.

Global Positioning System (GPS)

- Each satellite broadcasts coded radio waves containing
  - Identity and location of satellite
  - Data and time when signal was sent
- The GPS control segment:
  - global network of **ground facilities** that track the GPS satellites, monitor their transmissions, perform analyses, and send commands and data to satellites



- **Monitor/Tracking Stations** constantly receive satellite data and forward them to a **Master Control Station** (MCS)
- **MCS** computes corrections to the satellites' orbital and clock information which are sent back to the satellites

Global Positioning System (GPS)

Two levels of service:

– Standard Positioning Service (SPS)
  - available to all users, no restrictions or direct charge
  - high-quality receivers have accuracies of 3m (worse in height)

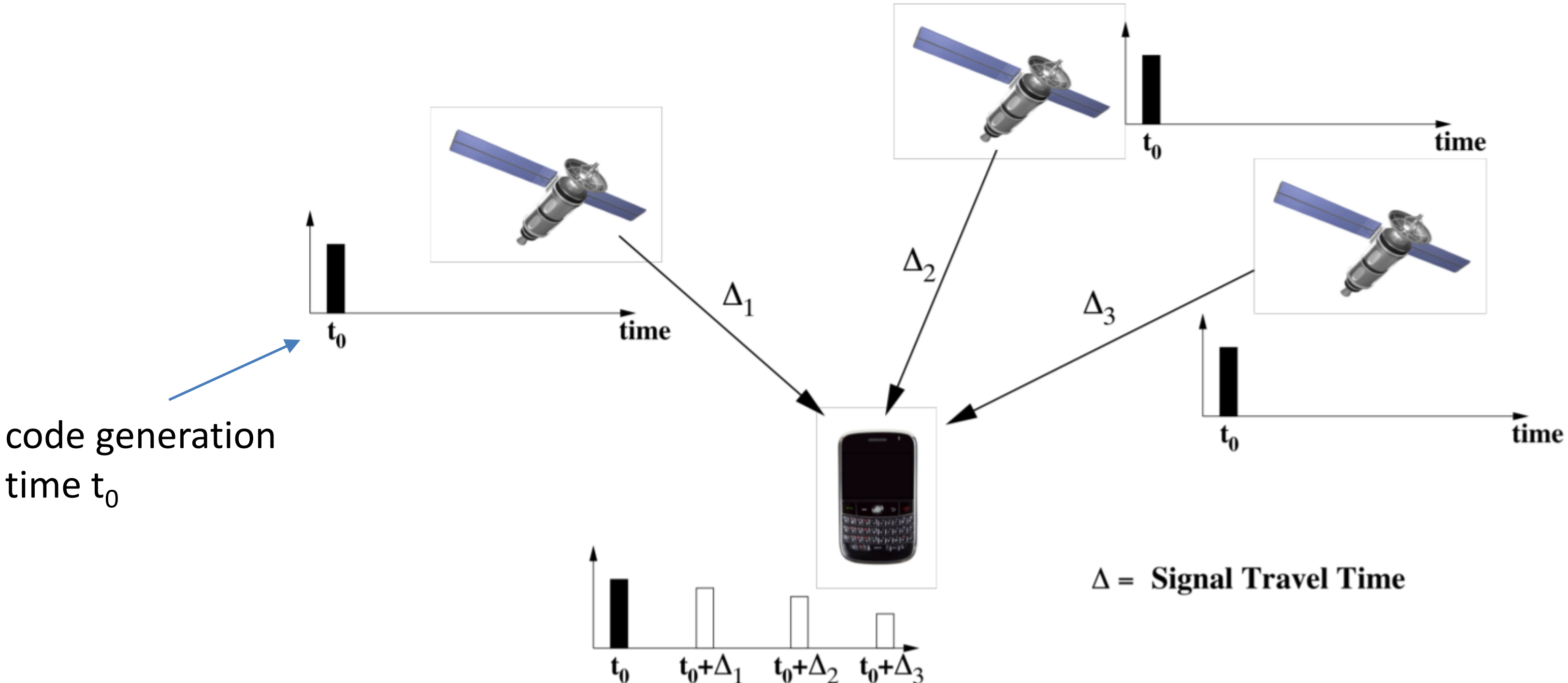
– Precise Positioning Service (PPS)
  - used by US and allied military users
  - uses two signals to reduce transmission errors

GPS signals contain two ranging codes:

- the coarse/acquisition (C/A) code, which is freely available to the public, and
- the restricted precision (P) code, usually reserved for military applications.

Location of a receiver determined through **T**ime of **F**light (ToF)

- Satellites and receivers use accurate and synchronized clocks
- The actual code generation time in the satellite is computed (precision depending on the code of the receiver (C/A or P))
- The time difference Δ between code generation time and current time is computed. This gives the travel time of the code from satellite to receiver

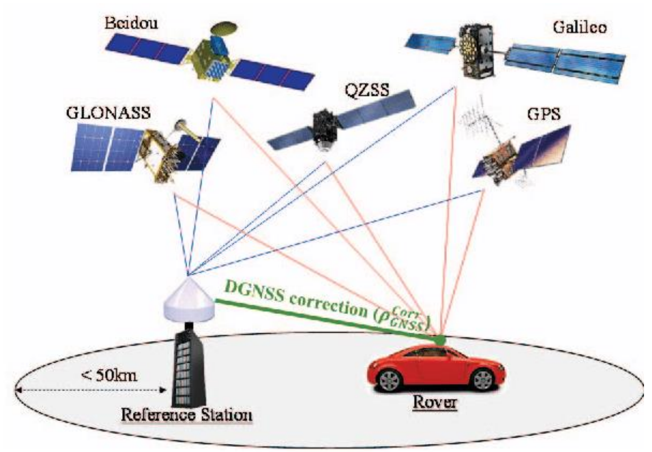


From: Fundamentals of Wireless Sensor Networks: Theory and Practice Waltenegus Dargie and Christian Poellabauer © 2010 John Wiley & Sons Ltd

Global Positioning System (GPS)

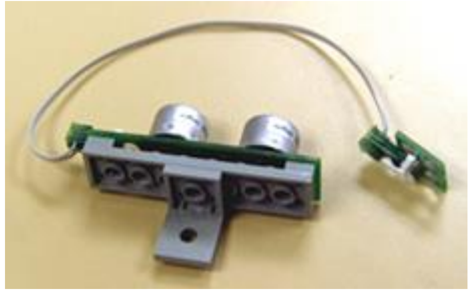
Errors

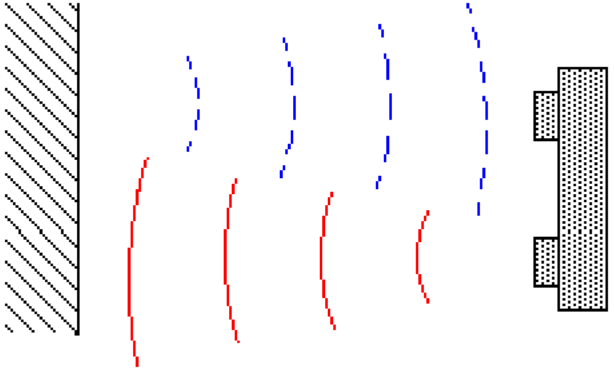
- Key for accuracy: Time synchronization between the satellites and the GPS receiver
- Errors due to interferences with atmosphere, synchronization, number of satellites in view, surrounding conditions (satellite visibility and multipath), …
- Error depends on the GPS techniques employed:
  - **Stand Alone**
    GPS on its own with no additional correction: <10 m.
    (cellular phones)
  - **Differential GPS** (DGPS)
    Use a reference base for corrections: < 5m.
  - **RTK GPS**:
    It's a DGPS with more complex GPS data processing: <10 cm.
    (in topography)



Range sensors: Sonar

- Originally developed by Polaroid for camera range finding (to focus the scene)
- Works as follows:
  - An emitter sends a short burst of ultrasonic sound (>20 KHz)
  - Part of the signal bounces off the obstacle and is sensed by a receiver
  - Emitter and receiver are perfectly synchronized (built in the same unit) so the object distance is computed from the elapsed time (Time of Flight (TOF) principle)



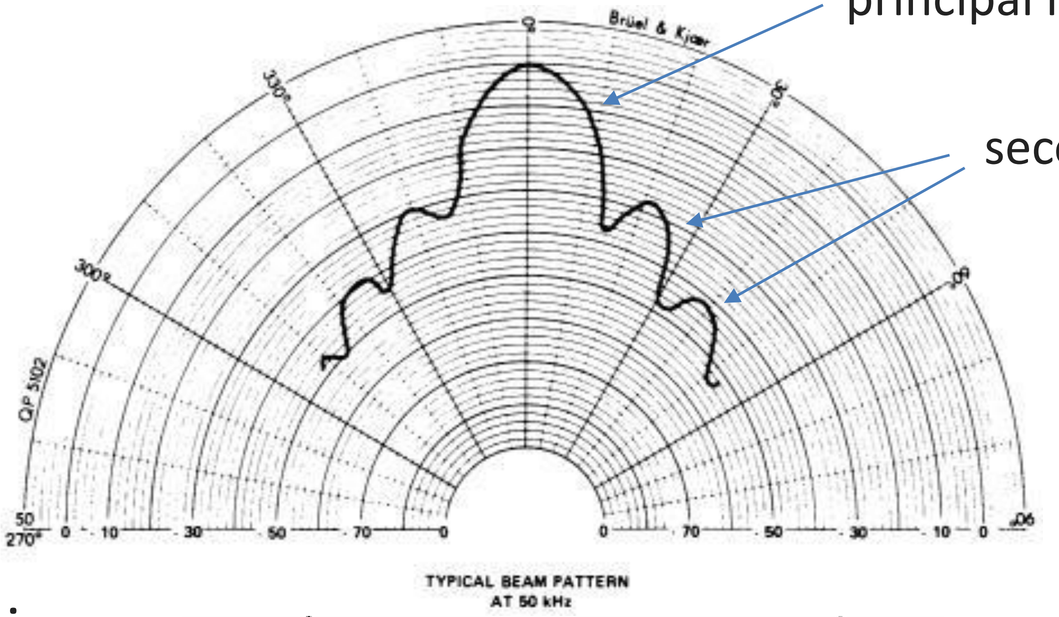


Range sensors: Sonar

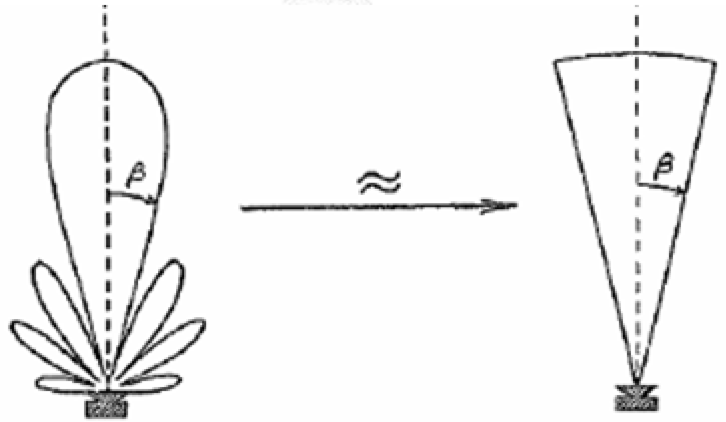
The effective beam width is about 30 degrees, but there are secondary "lobes"

principal lobe

secondary lobes



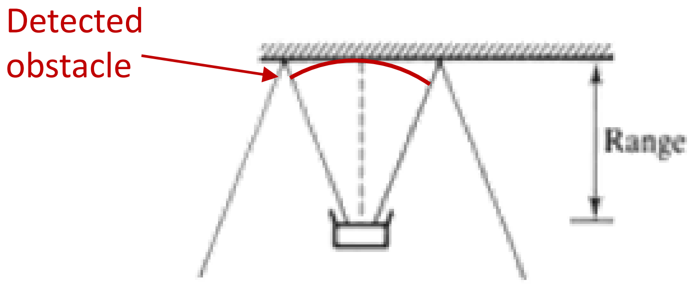
Approximation:



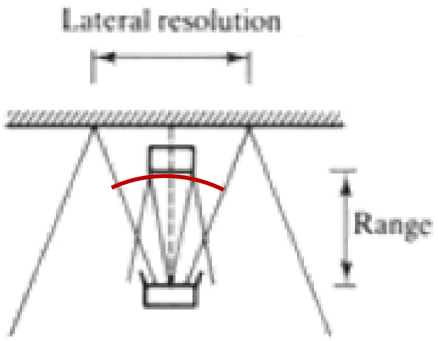
secondary lobes are discarded

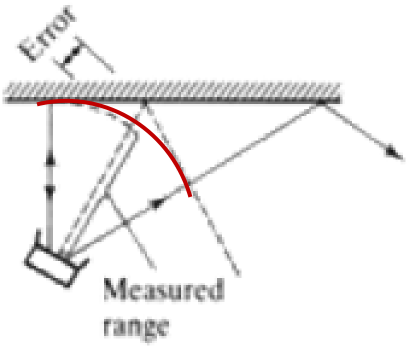
Range sensors: Sonar

Performance:



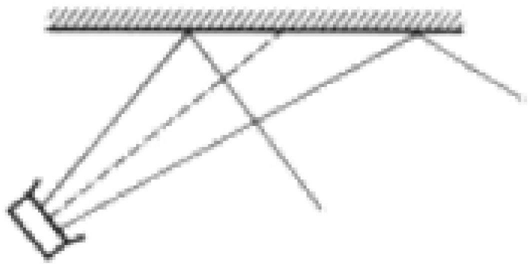
Accurate distance



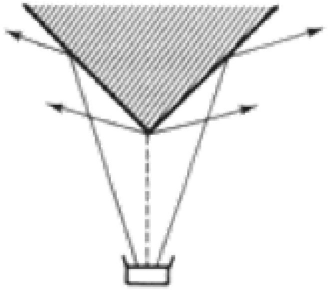


Lateral resolution not very precise; the closest object in the beam's cone provides the response

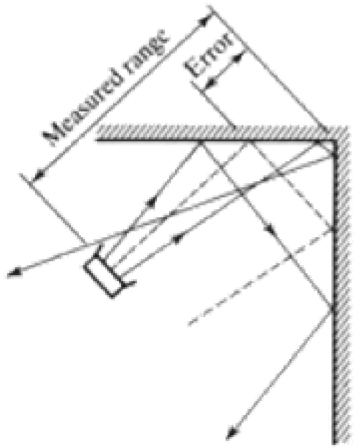
Missed echo:



Specular reflections cause walls to disappear(missed echo)



Open corners produce a weak spherical wavefront (missed echo)



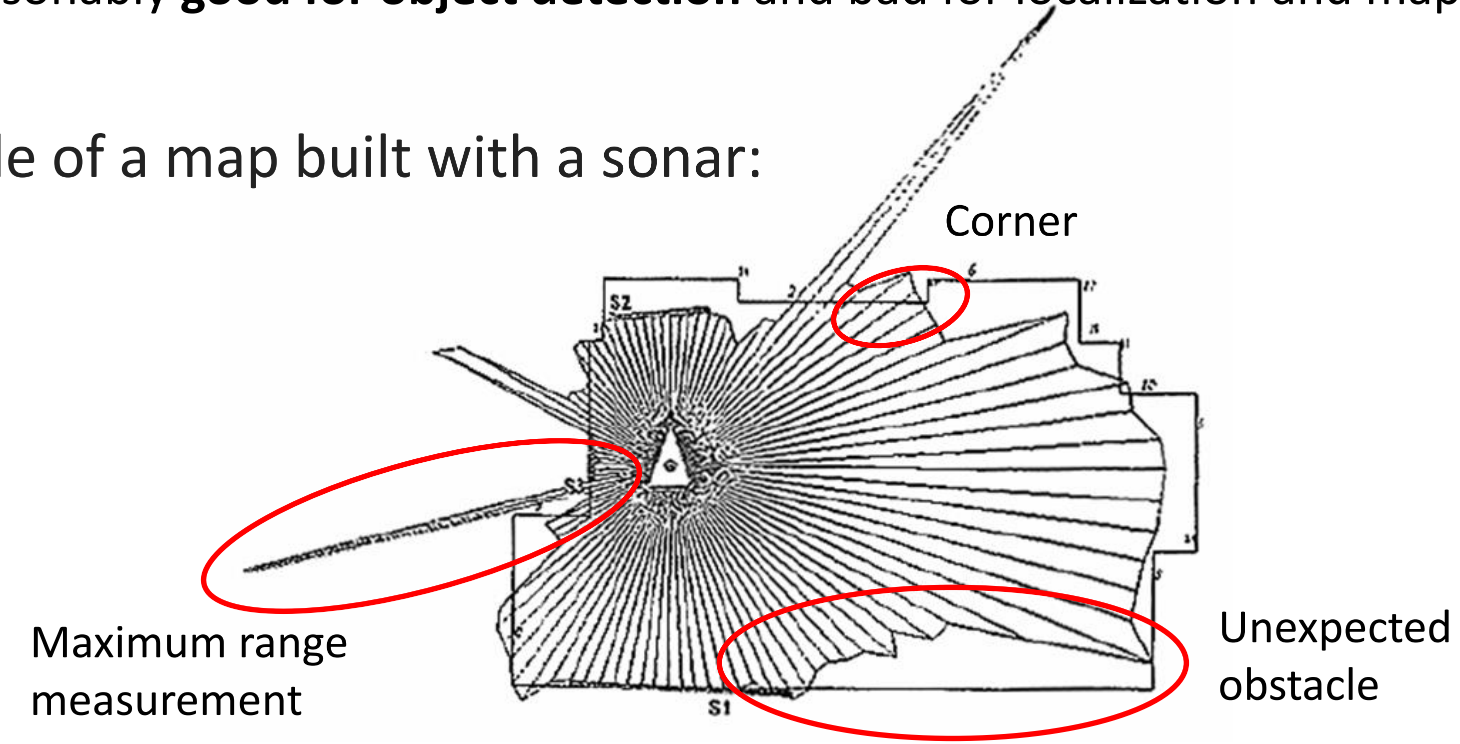
Error because of multiple reflections

Range sensors: Sonar

Features:

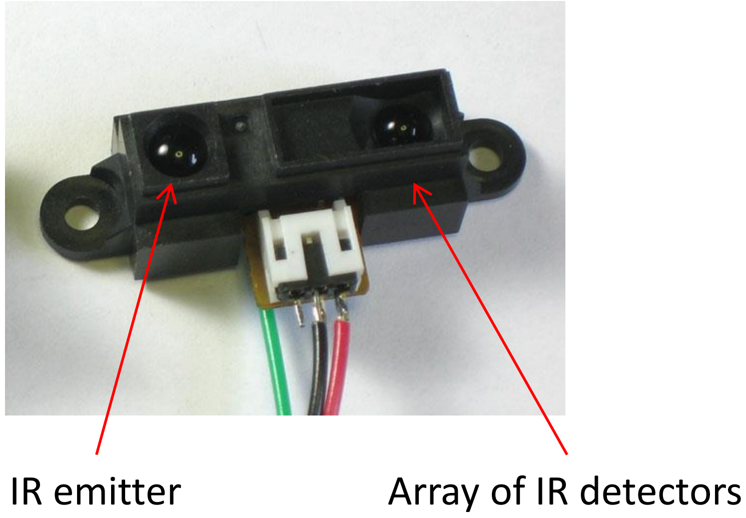
- Small, **cheap**, low power consumption
- **Range until 3m**
- Very **poor angular resolution**
- **Problems** with specular objects, and corners
- Reasonably **good for object detection** and bad for localization and mapping

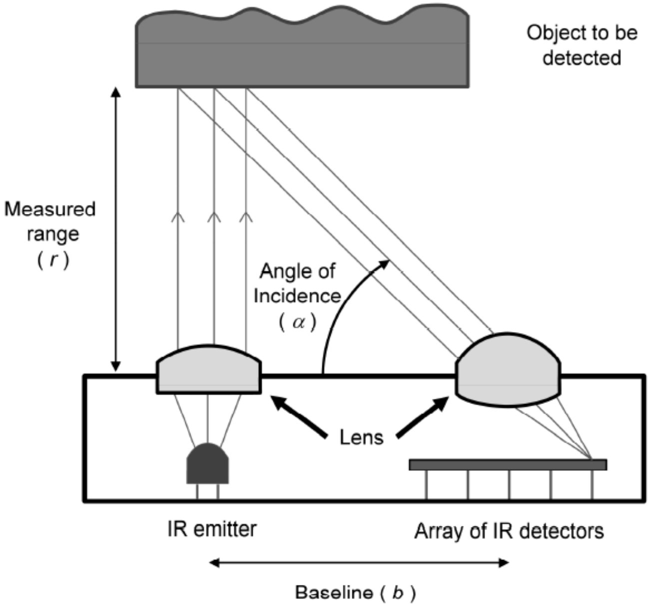
Example of a map built with a sonar:



Range sensors: Infrared

Based on triangulation of light





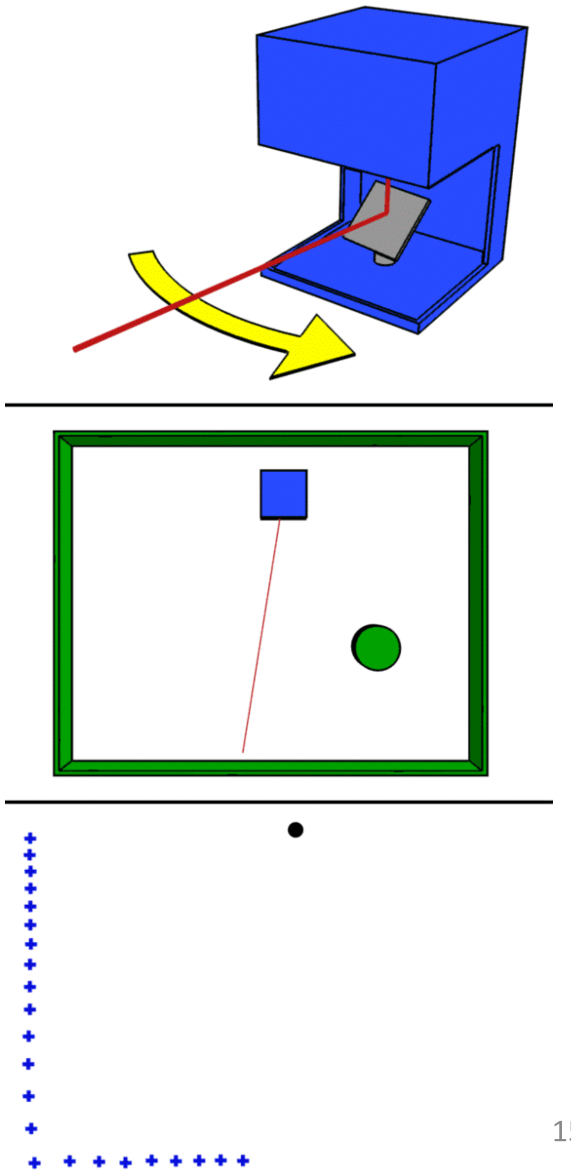
Distance is computed from the detected angle

Features:

- Small, cheap, low power consumption
- **Range until 2m.**, good angular resolution (the beam is much narrower than for a sonar)
- Problems with transparent, glossy and **black** objects, and sun light

Range sensors: Laser scanner

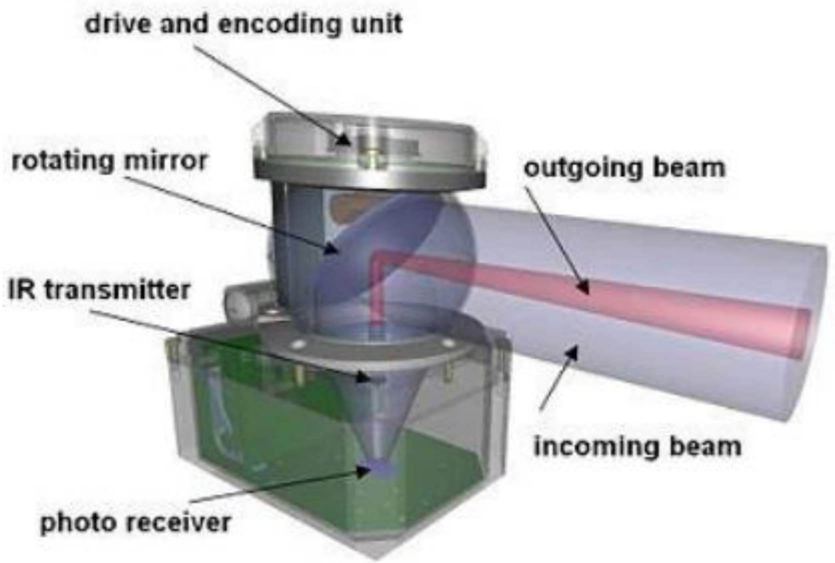
- Transmitter illuminates a target with a laser beam (nearIR, not visible) → more energy (= more range) than standard IR light
- Receiver detects the time needed for round-trip: 2*distance = c*time
- A motor rotates a tilted mirror so the laser beam sweeps on a plane
- Very precise (accuracy < 5cm.) and long range (until 50 m.)
- Relative affordable (1000-2500€)



Range sensors: Laser scanner

HOKUYO radial laser scanner

- Long detection range: 30m
- Wide Angle: 270°
- Valid for Outdoor Environment
- Compact and Light: W60xD60xH87mm, 370g



Range sensors: Laser scanner

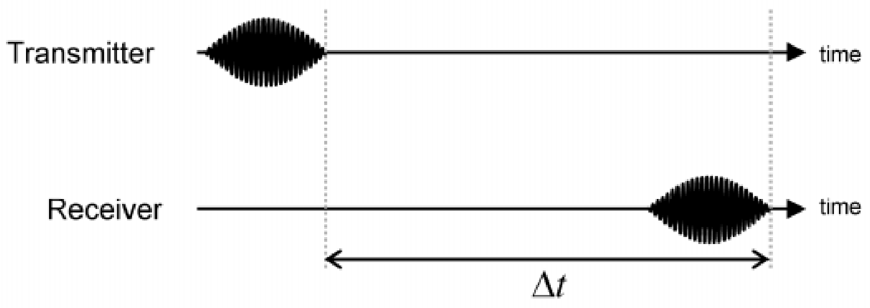
SICK radial laser scanner

- Long detection range: 50m
- Wide Angle: 180°
- Better than Hokuyo for outdoor
- Much heavier and bigger that Hokuyo (~4.5 kg.)

Range sensors: Laser scanner

Two main working principles:

Pulse time-of-flight

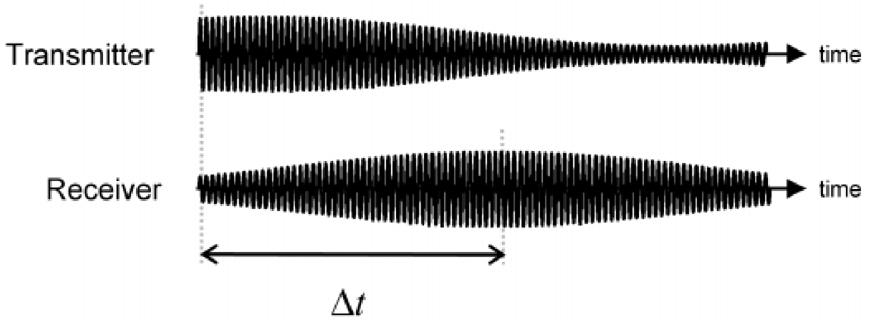


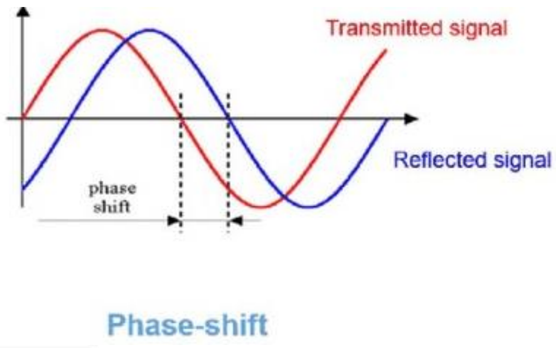
- longer range
- slightly slower
- less accuracy

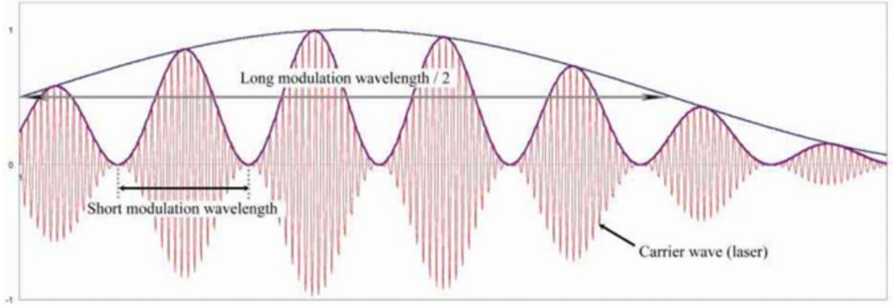
$$range = c.\Delta t/2$$

$$c: light\ speed$$

Phase shift





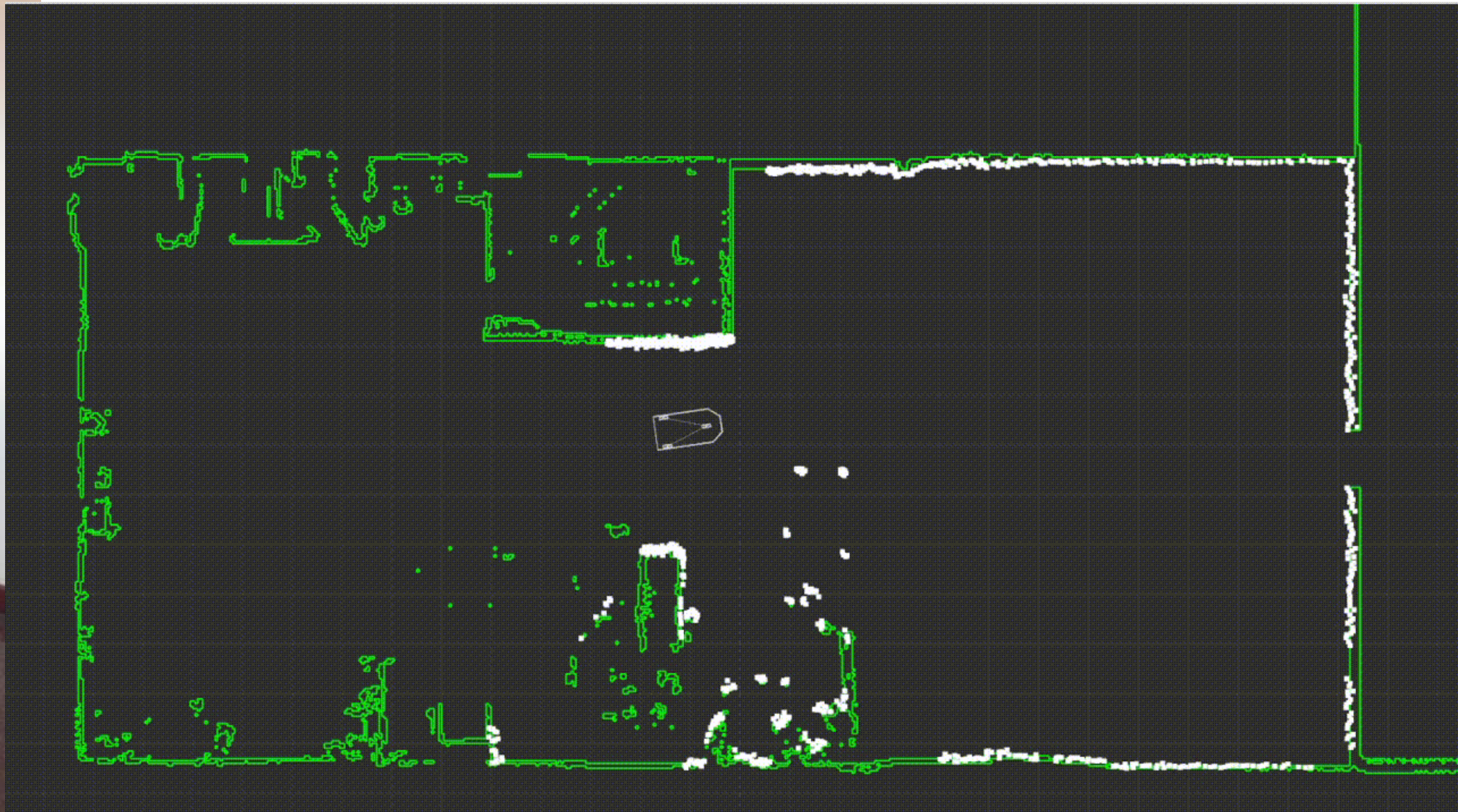


The carrier laser wave is modulated in amplitude to have a long wavelength

- medium range (given by the carrier wavelength)
- high accuracy
- very fast

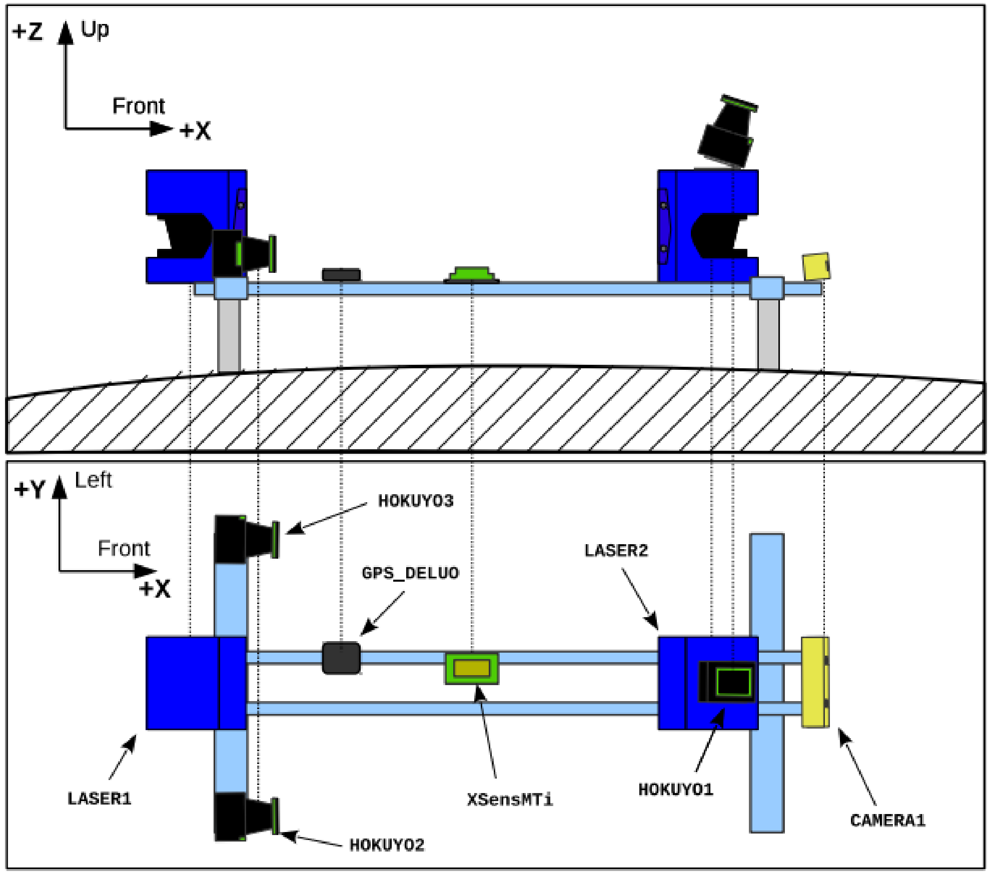
Hokuyo Radial
Laser Scanner

Example of a scan



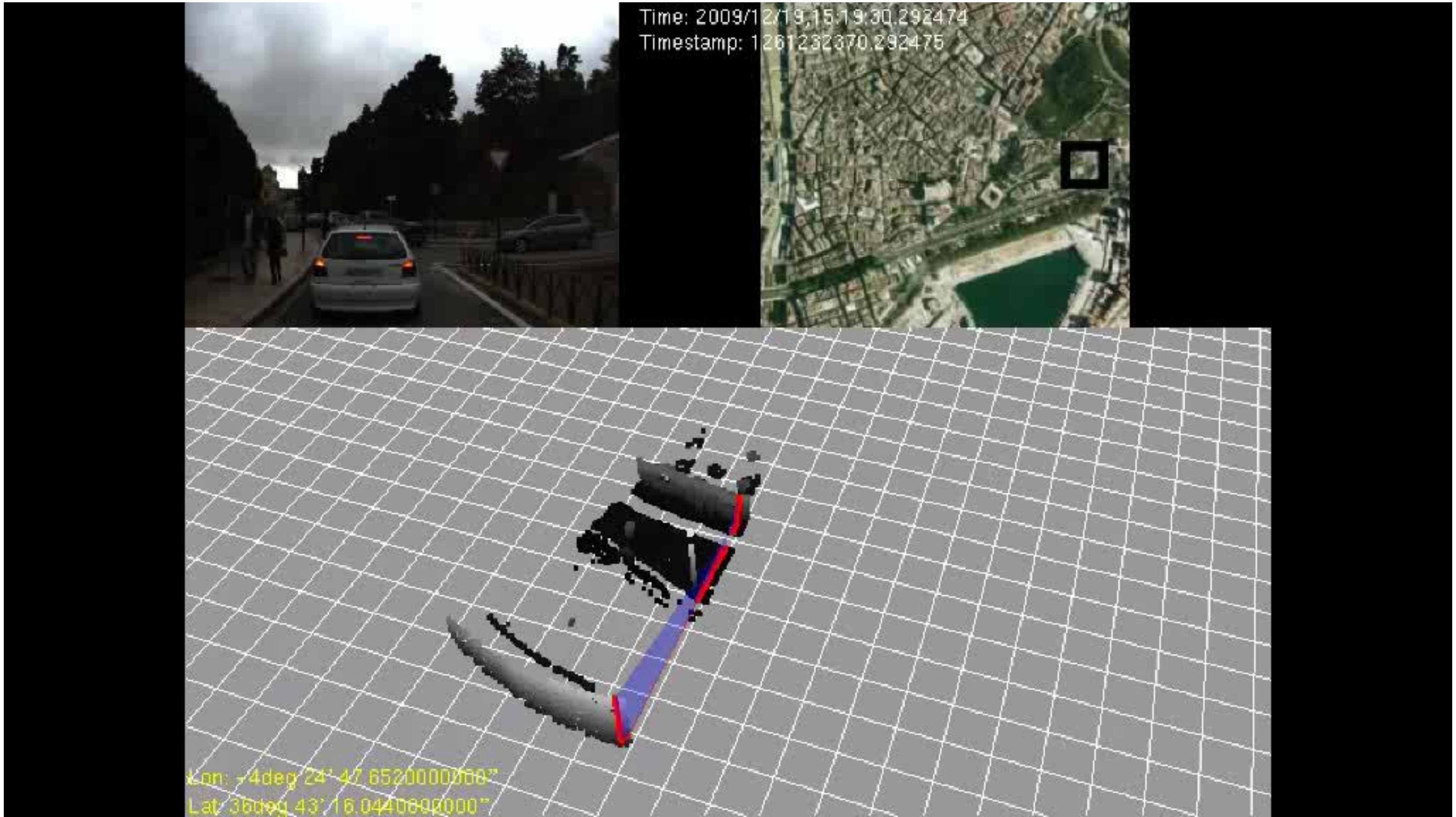
Range sensors: Laser scanner

Many possible configurations in a robot



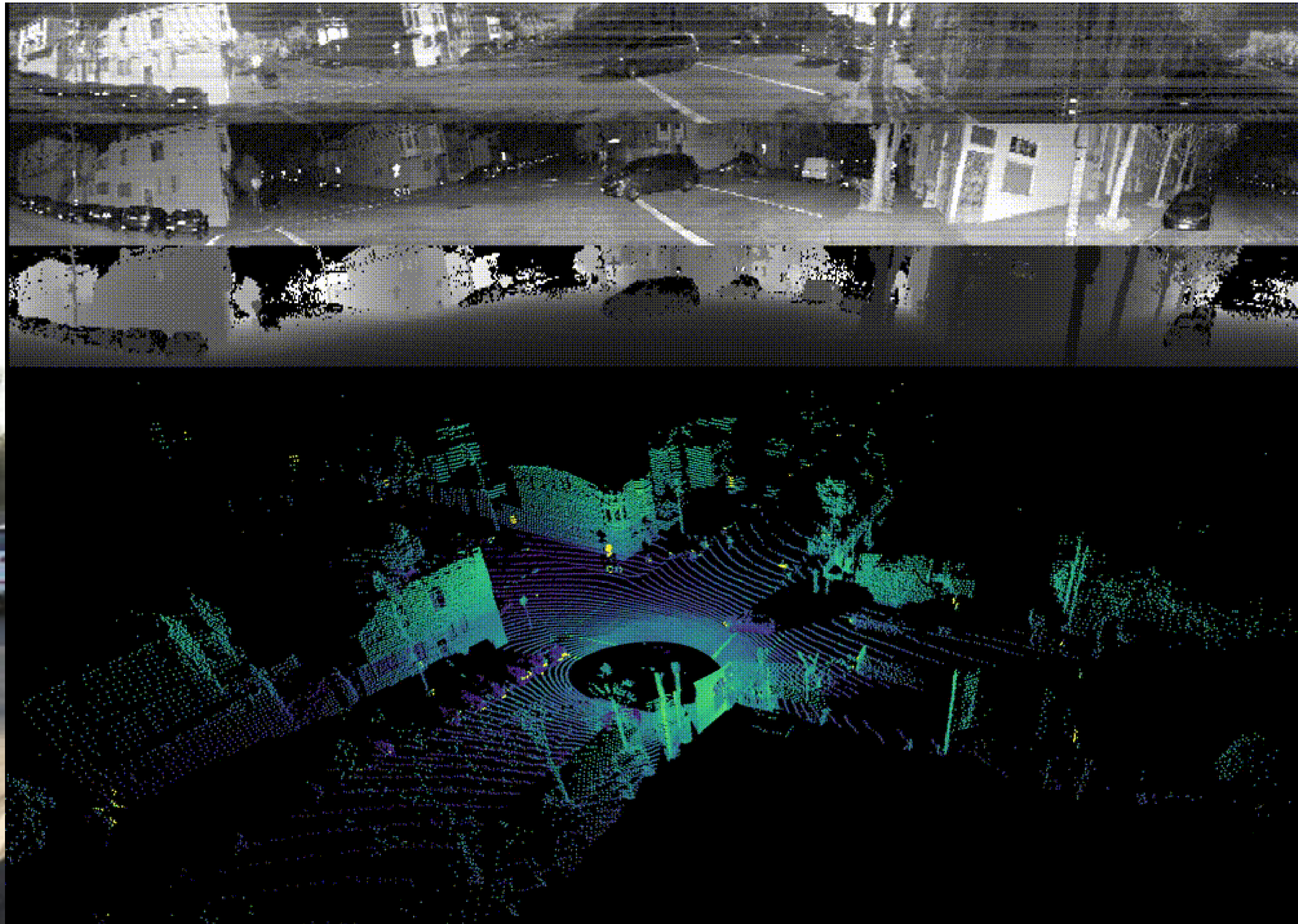
Range sensors: Laser scanner

Example:



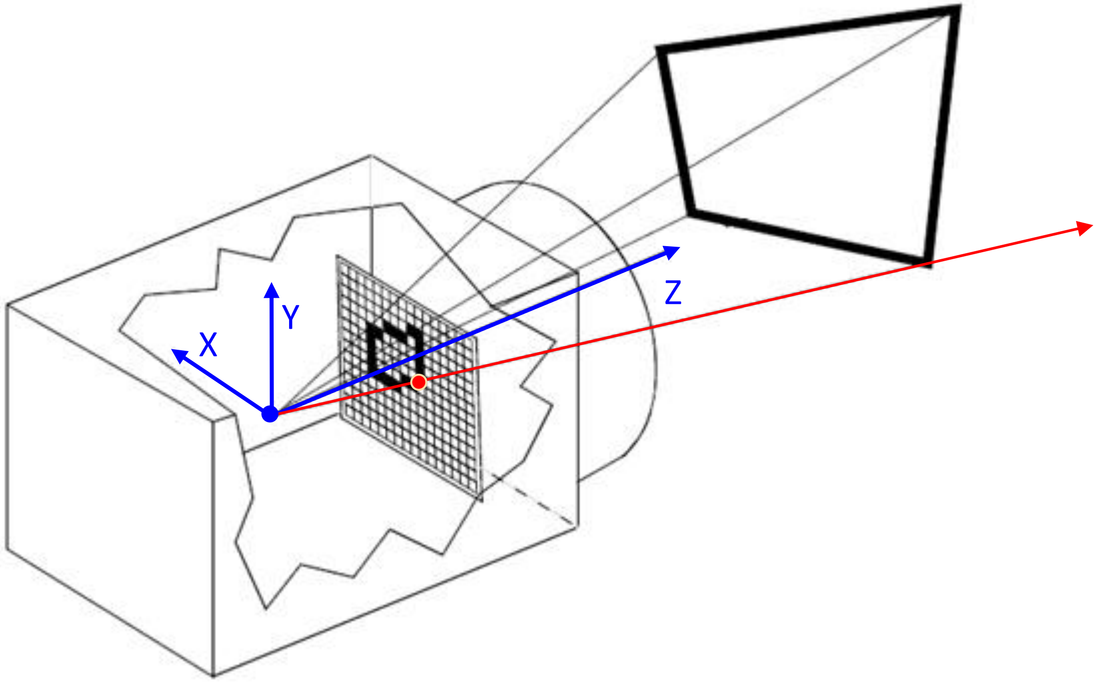
Range sensors: 3D Laser scanner

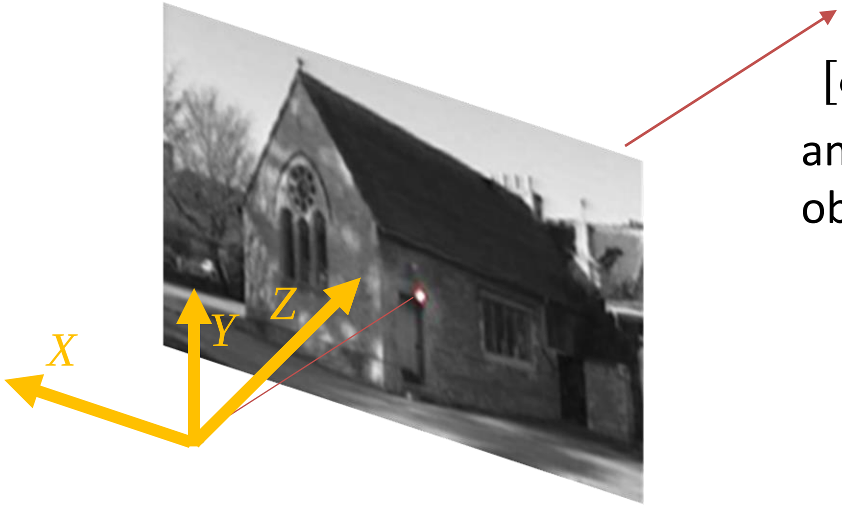
Ouster OS1 3D Lidar



Cameras

They only provide angles of observation → Bearing sensor



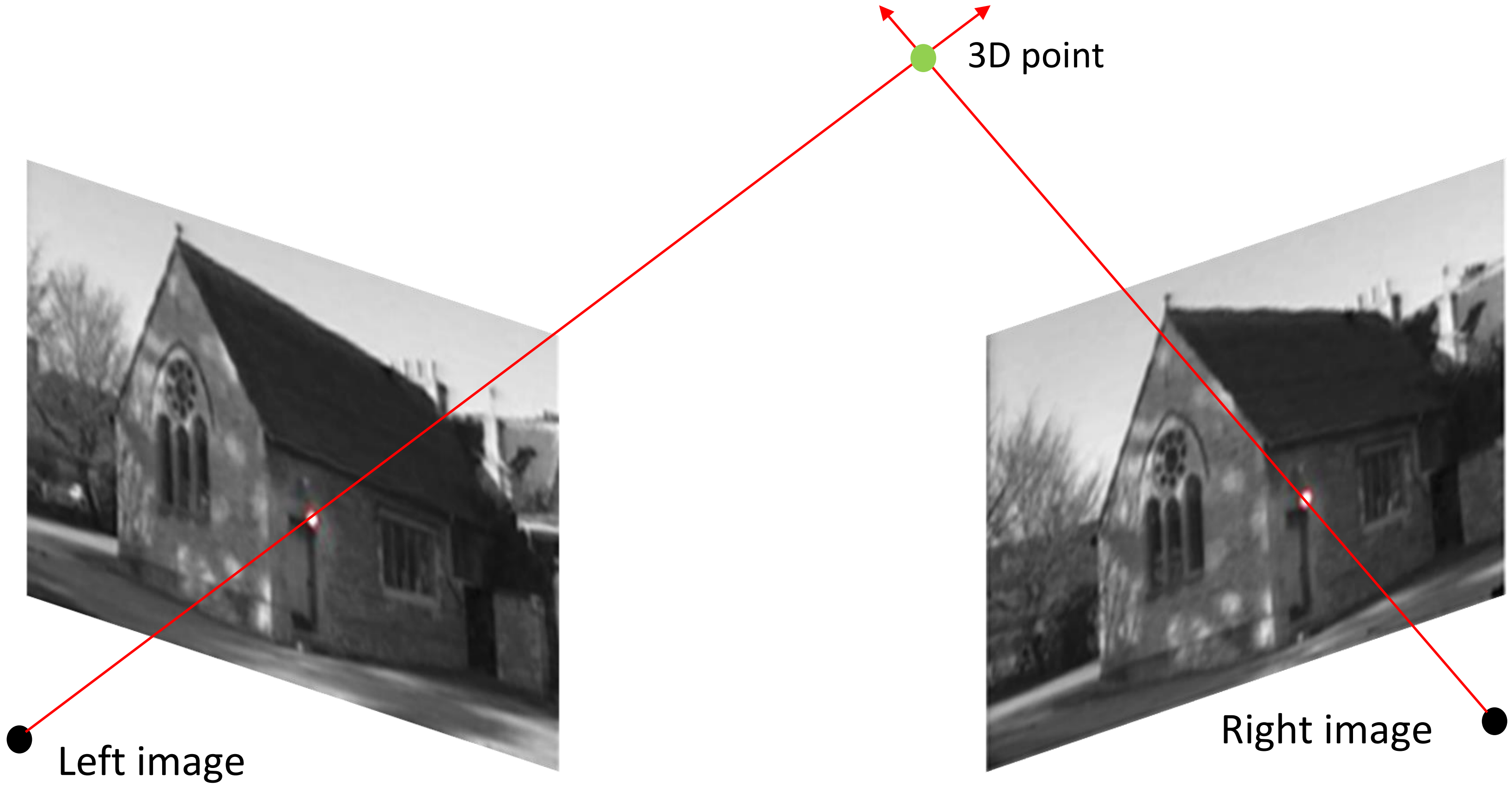


$[\varphi, \theta]$

angle of observation

Cameras

Bearing-range sensor (through stereo)



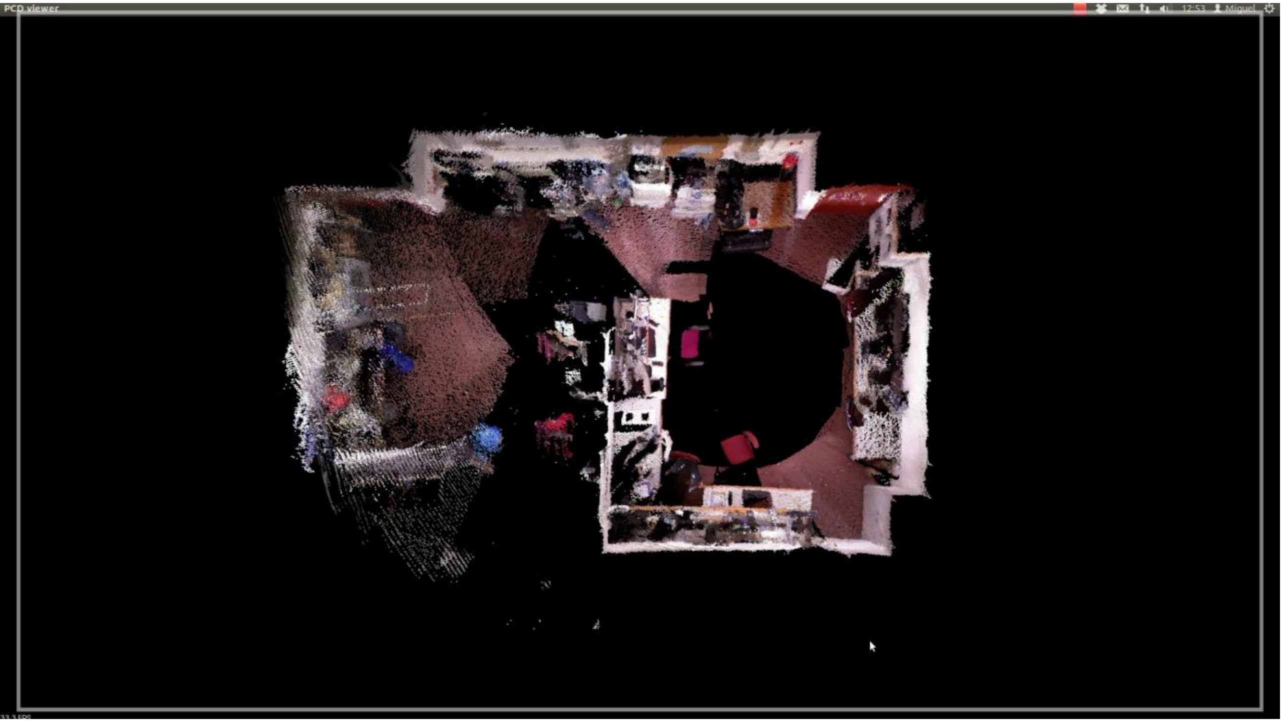
3D point obtained though triangulation of the line-of-sight of two pixels

RGB-D Cameras

- Bearing-range sensor + RGB
- Contrary to 3D laser scanners, no moving components
- Has revolutionized robotics (revolution still ongoing!)

RGB-D Cameras

Used for:



Localization&mapping (SLAM)

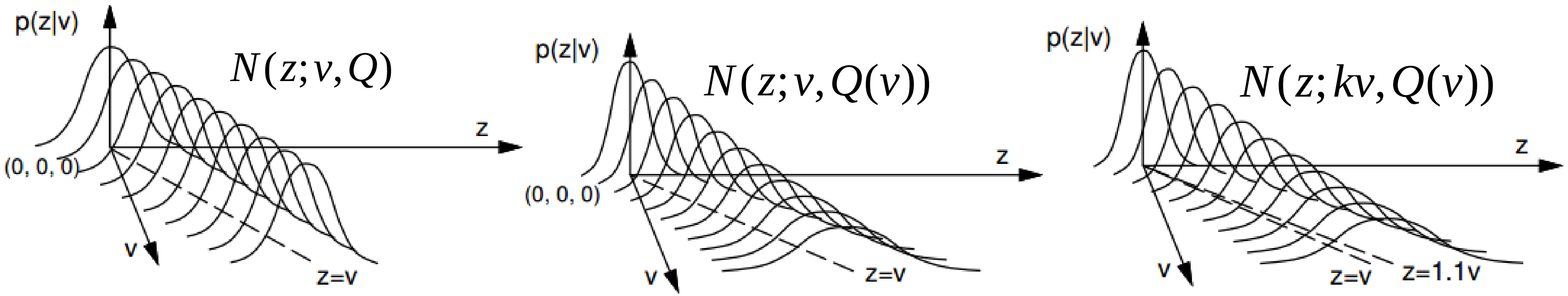
Human-robot interaction

Probabilistic Sensor Models

- Real sensors do not deliver the exact truth of the quantities they are measuring but a perturbed version: truth + error
- We need to determine the function $p(z/v)$ which describes the sensor performance: z measurement variables, v ground truth
- $p(z/v)$, as a function of both z and v and can be plotted as a probability surface.

Example

For a depth sensor: probability function of a measurement z given that the robot is at a distance v to the target. Q is the covariance



Constant Uncertainty Growing Systematic Error (Biased)

Probabilistic Sensor Models

The measurements (observations) z depend on z, which is a function h of the pose x and the environment (map m):

$$p(z|v = h(x, m)) = p(z|x, m)$$

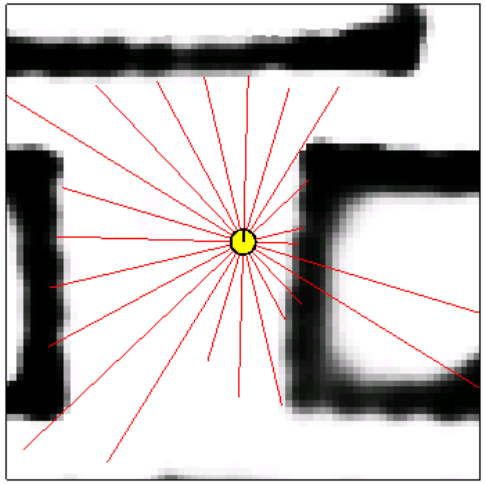
observation

map of the environment

sensor pose

The observation z may consist of a set of measurements (a vector), either from **different sensors** or from the **same device** (e.g. each range of a scan)

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



Example: a laser scan contains many measurements

Typically, we assume independency between the different measurements

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

RECALL: x conditionally independent of y (and viceversa) **given** z *if*

$$P(x, y|z) = P(x|z)P(y|z)$$

Probabilistic Sensor Models

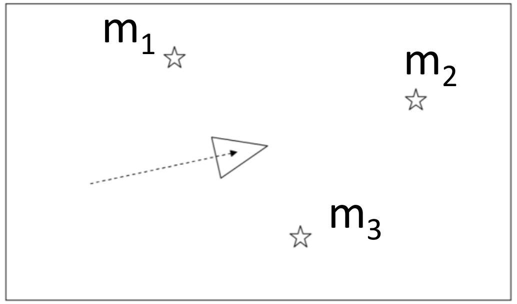
A *map* is a list of objects in the environment represented by their locations and, sometimes some properties (e.g. visual descriptor)

$$m = \{m_1, m_2, \ldots, m_N\}$$

N: total number of elements (landmarks, cells) in the environment

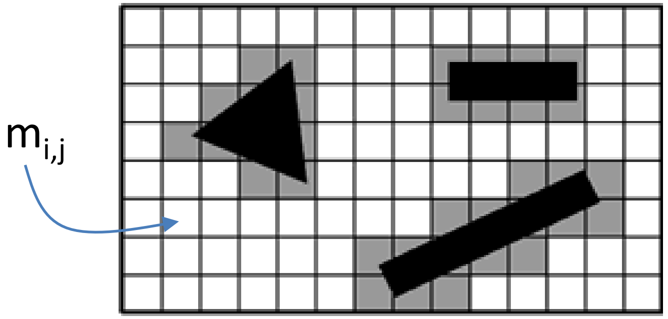
Maps are indexed in two ways:

landmark-based map



- Used by landmark-based sensors (Cameras, beacons)
- A landmarks is described by its **position (x,y)** and some **descriptor**.
- Sensor models used:
  - Bearing (angle)
  - Range (distance)
  - Range and bearing

occupancy grid map



- Used by scan-based sensor and depth cameras
- Each element of the map is a **cell** described by its position (index) in the grid (2D or 3D), and a probability (represented by an intensity value)
- Sensor models used:
  - Beam model
  - Likelihood field

Probabilistic Sensor Models

Remember why we need this $p(z_t|x_t,m)$ [*lecture 2*]:

$$Bel(x_t) = \eta\ p(z_t|x_t,m) \int p(x_t|u_t,x_{t-1})\, Bel(x_{t-1})\ dx_{t-1}$$

An expression that depends on $[x,m,z]$.

- As a conditional distribution: $[x,m]$ known, z is a RV.
- Here it's a Likelihood $\mathcal{L}(x_t|m,z_t)$: z, m are given, the variable is x.

How is the RV z related to x and m?

$$z = h(x,m) + w$$

- Since the sensor is not perfect, z is corrupted by an error w
- $h(x,m)$ is the Observation (or measurement or prediction) function: it predicts the value of z **given the state values x and m**

Assuming **Gaussian error** for the sensor measurements:

Uncertainty in the observation

$$z \sim N(h(x,m),Q) \qquad w = [h(x,m) - z] \sim N(0,Q)$$

$$p(z\,|\,x,m) = K \cdot \exp\{-\frac{1}{2}[h(x,m)-z]^T Q^{-1}[h(x,m)-z]\}$$

Landmark observation models

- The map is a collection of landmarks: $m = \{m_i\}$ $i=1, ..N$
- Sensor provides a measurement to those landmarks
  - Distance (*e.g.*, WIFI, GPS): $z_i = d_i = h_i\,(x,m) + w_i$
  - Bearing (*e.g.*, camera): $z_i = \theta_i = h_i\,(x,m) + w_i$
  - Distance and bearing (stereo, features in a scan,…)

$$z_i = [d_i\ ,\ \theta_i]^T = h_i\,(x,m) + w_i$$

2D vectors

- A new problem arises: DATA ASSOCIATION

Which landmark m_i does a given observation z_i correspond to ?

$$z_i = h_i\,(x,m) = h(x,m_i\,)$$

We know/estimate that the observation z_i comes from m_i

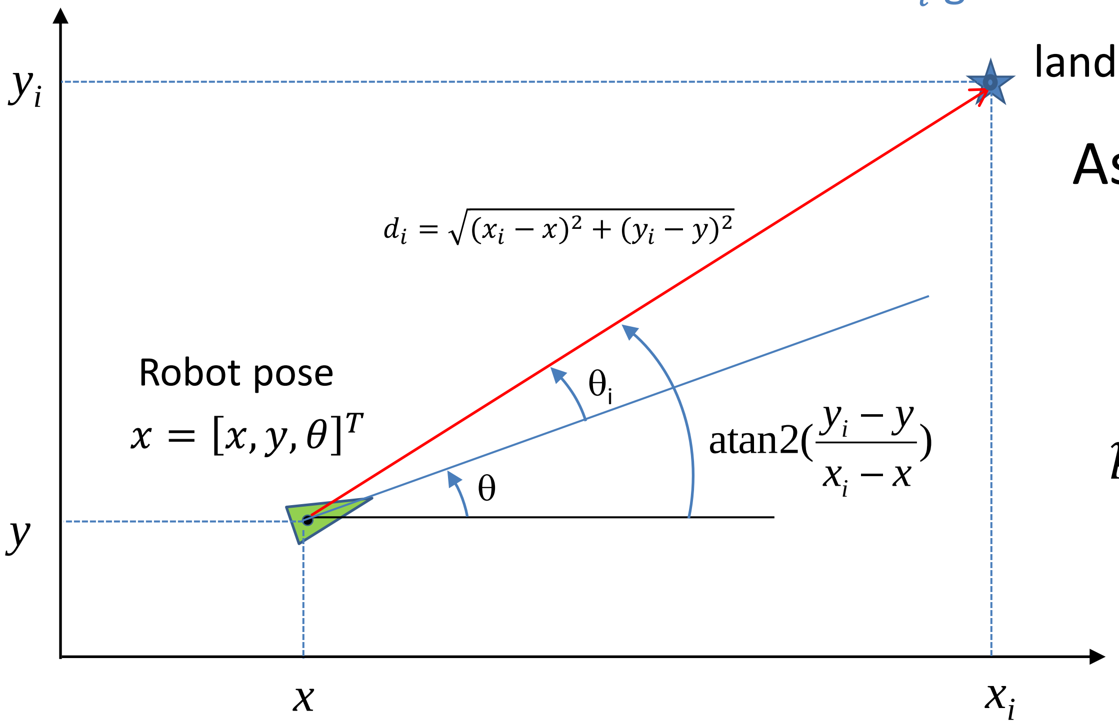
Big issue!!
Mostly addressed by applying Chi-squared tests

Landmark observation models: **Distance and bearing**

The observation consists of a range (distance) d_i and a bearing (angle) θ_i to the i-th landmark:

Measurement z_i comes from landmark m_i

$$z_i \triangleq \begin{bmatrix} d_i \\ \theta_i \end{bmatrix} = \underbrace{h(x, m_i)}_{\text{predicted value of } z_i \text{ given } x \text{ and } m_i} + w_i = \begin{bmatrix} \sqrt{(x_i - x)^2 + (y_i - y)^2} \\ \operatorname{atan}\left(\frac{y_i - y}{x_i - x}\right) - \theta \end{bmatrix} + w_i$$



Assuming that the error w_i is Gaussian:

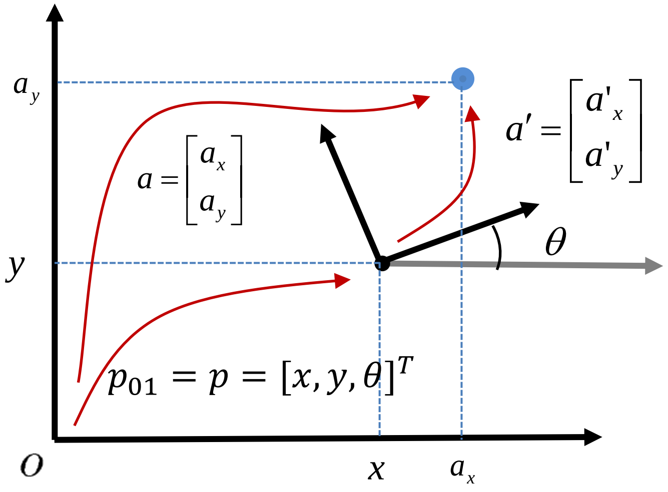
$$w_i = [h(x, m_i) - z_i] \sim N(0, Q_i)$$

Covariance matrix 2x2

$$p(z_i | x, m_i) = K \cdot \exp\{-\frac{1}{2}[h(x, m_i) - z_i]^T Q_i^{-1} [h(x, m_i) - z_i]\}$$

Notice: $p(z_i | x, m_i)$ is defined to be gaussian for z_i, BUT $\mathcal{L}(x_t | m, z_t)$, a function of x, is not gaussian!!

Composition of a **pose** and a **landmark point**: $a = p \oplus a'$



$$\text{pose: } p_{01} = p = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} = \begin{bmatrix} p_{xy} \\ \theta \end{bmatrix} \qquad \text{landmark point: } a' = \begin{bmatrix} a'_x \\ a'_y \end{bmatrix}$$

$$\tilde{a} = \begin{bmatrix} a \\ 1 \end{bmatrix} = \overbrace{\begin{bmatrix} \boldsymbol{R}_\theta & p_{xy} \\ 0^T & 1 \end{bmatrix}}^{{}^0\boldsymbol{T}_1} \begin{bmatrix} a' \\ 1 \end{bmatrix} = {}^0\boldsymbol{T}_1 \tilde{a}' \Rightarrow a = \begin{bmatrix} a'_x \cos\theta & -a'_y \sin\theta \\ a'_x \sin\theta & a'_y \cos\theta \end{bmatrix} + \begin{bmatrix} x \\ y \end{bmatrix}$$

pose

point

Expressed as a **composition**: $a = p \oplus a = \begin{bmatrix} x + a'_x \cos\theta - a'_y \sin\theta \\ y + a'_x \sin\theta + a'_y \cos\theta \end{bmatrix} = f(p, a')$

Jacobians:

$$\frac{\partial a}{\partial p} = \frac{\partial f(p,a')}{\partial p} = \frac{\partial\{a_x, a_y\}}{\partial\{x, y, \theta\}} = \begin{bmatrix} 1 & 0 & -a'_x \sin\theta - a'_y \cos\theta \\ 0 & 1 & a'_x \cos\theta - a'_y \sin\theta \end{bmatrix}$$

$$\frac{\partial a}{\partial a'} = \frac{\partial f(p,a')}{\partial a'} = \frac{\partial\{a_x, a_y\}}{\partial\{a'_x, a'_y\}} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

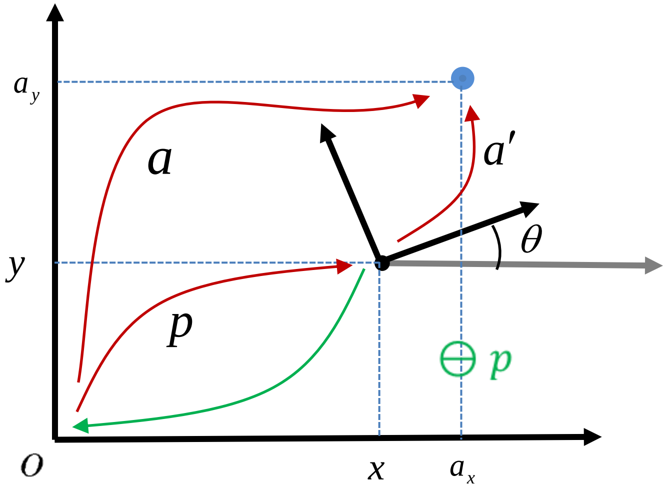
```
function Jac   = J1(p1,p2)
s1  = sin(p1(3));
c1  = cos(p1(3));
Jac   = [1  0  -p2(1)*s1-p2(2)*c1;
          0  1  p2(1)*c1-p2(2)*s1;
          0  0  1];
```

```
function Jac   = J2(p1,p2)
s1  = sin(p1(3));
c1  = cos(p1(3));
Jac   = [c1 -s1 0;s1 c1 0;0 0 1];
```

Note.- Same as for pose composition $p_1 \oplus p_2$ but only 2x3 or 2x2 submatrices are used

Inverse composition of a pose and a point $a' = \ominus p \oplus a$:

(eg.: where will a known landmark be seen from a robot?)



$$a = \begin{bmatrix} a_x \\ a_y \end{bmatrix} \quad a' = \begin{bmatrix} a'_x \\ a'_y \end{bmatrix} \quad p_{01} = p = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} = \begin{bmatrix} p_{xy} \\ \theta \end{bmatrix}$$

$$a = p \oplus a' = \begin{bmatrix} x + a'_x \cos\theta - a'_y \sin\theta \\ y + a'_x \sin\theta + a'_y \cos\theta \end{bmatrix} = f(p, a')$$

$$\tilde{a} = \begin{bmatrix} a \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_\theta & p_{xy} \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} a' \\ 1 \end{bmatrix} = {}^0\boldsymbol{T}_1 \widetilde{a'}$$

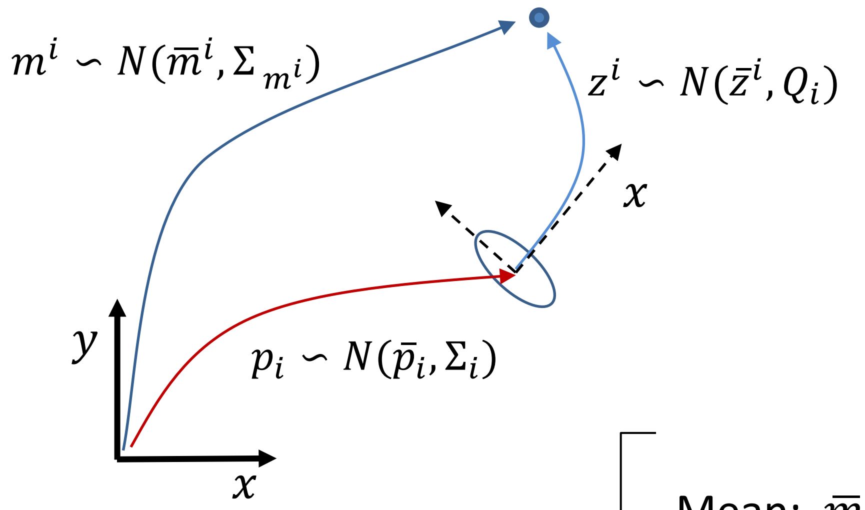
$$\widetilde{a'} = {}^0\boldsymbol{T}_1{}^{-1} = {}^1\boldsymbol{T}_0 \tilde{a} \quad \Rightarrow \quad a' = \ominus p \oplus a = p \ominus a = \begin{bmatrix} \widetilde{a'}(1) \\ \widetilde{a'}(2) \end{bmatrix} = \begin{bmatrix} (a_x - x)\cos\theta + (a_y - y)\sin\theta \\ -(a_x - x)\sin\theta + (a_y - y)\cos\theta \end{bmatrix} = f'(p, a)$$

Jacobians:

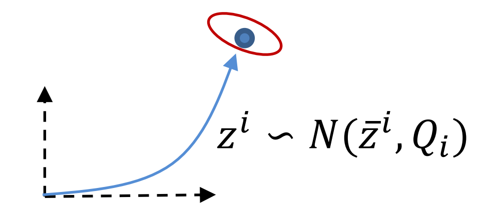
$$\frac{\partial a'}{\partial p} = \frac{\partial f'(p,a)}{\partial p} = \frac{\partial\{a'_x, a'_y\}}{\partial\{x, y, \theta\}} = \begin{bmatrix} -\cos\theta & -\sin\theta & -(a_x - x)\sin\theta + (a_y - y)\cos\theta \\ \sin\theta & -\cos\theta & -(a_x - x)\cos\theta - (a_y - y)\sin\theta \end{bmatrix}$$

$$\frac{\partial a'}{\partial a} = \frac{\partial f'(p,a)}{\partial a} = \frac{\partial\{a'_x, a'_y\}}{\partial\{a_x, a_y\}} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$$

COVARIANCE of an observed 2D landmark (in cartesian coordinates)



Observation



Q_i is the covariance of z^i in the robot frame

For coherence with later notation we have change from subscript (z_i) to superscript z^i

$$m^i = p_i \oplus z^i = \mathrm{f}(p_i, z^i)$$

Mean: $\overline{m}^i = \bar{p}_i \oplus \bar{z}^i$

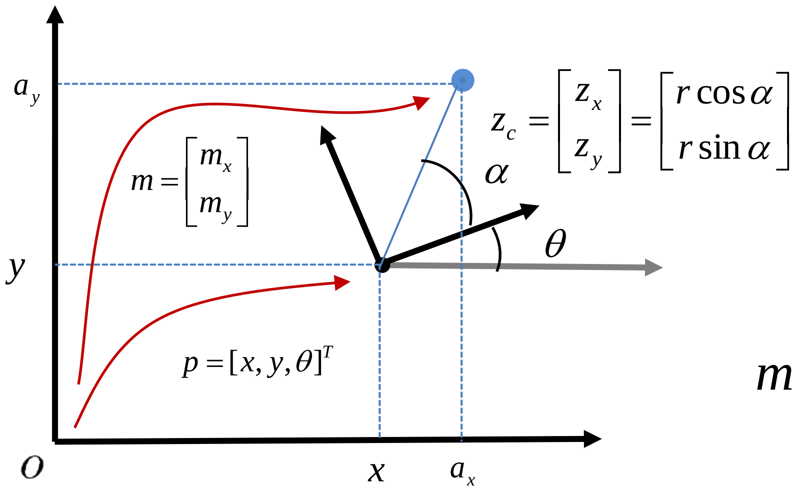
Covariance:
$$\underbrace{\Sigma_{m^i}}_{2\text{x}2} = \overbrace{\underbrace{\frac{\partial\, m^i}{\partial p_i}}_{2\text{x}3} \Sigma_i \left(\frac{\partial\, m^i}{\partial p_i}\right)^T}^{\text{uncertainty in } m^i \text{ due to uncertainty in } p_i} + \overbrace{\underbrace{\frac{\partial\, m^i}{\partial z^i}}_{2\text{x}2} Q_i \left(\frac{\partial\, m^i}{\partial z^i}\right)^T}^{\text{uncertainty in } m^i \text{ due to uncertainty in } z^i}$$

RECALL: if $Z=f(X,Y)$

$$\boxed{\Sigma_Z = \frac{\partial f}{\partial X} \Sigma_X \left(\frac{\partial f}{\partial X}\right)^T + \frac{\partial f}{\partial Y} \Sigma_Y \left(\frac{\partial f}{\partial Y}\right)^T}$$

Σ_{m^i} is the sum of two covariance matrices!

COVARIANCE of an observed 2D landmark (in polar coordinates)



Sensor pose: $p = [x, y, \theta]^T$

Cartesian observation: $z_c = [z_x \ \ z_y]^T$

$$m = p \oplus z_c = f(p, z_c) = \begin{bmatrix} x + z_x\cos\theta - z_y\sin\theta \\ y + z_x\sin\theta + z_y\cos\theta \end{bmatrix}$$

$$\Sigma_m = \frac{\partial m}{\partial p}\Sigma_p\left(\frac{\partial m}{\partial p}\right)^T + \frac{\partial m}{\partial z_c}\Sigma_{z_c}\left(\frac{\partial m}{\partial z_c}\right)^T$$

Since

$$z_c = \begin{bmatrix} z_x \\ z_y \end{bmatrix} = \begin{bmatrix} r\cos\alpha \\ r\sin\alpha \end{bmatrix} = f(r, \alpha) \Rightarrow \Sigma_{z_c} = \frac{\partial z_c}{\partial z_p}\Sigma_{z_p}\left(\frac{\partial z_c}{\partial z_p}\right)^T \quad \text{with} \quad \frac{\partial z_c}{\partial z_p} = \begin{bmatrix} \cos\alpha & -r\sin\alpha \\ \sin\alpha & r\cos\alpha \end{bmatrix}$$

Same result if done in polar or cartesian coordinates

$$\frac{\partial m}{\partial z_c}\frac{\partial z_c}{\partial z_p}\Sigma_{z_p}\left(\frac{\partial z_c}{\partial z_p}\right)^T\left(\frac{\partial m}{\partial z_c}\right)^T = \frac{\partial m}{\partial z_p}\Sigma_{z_p}\left(\frac{\partial m}{\partial z_p}\right)^T$$

Chain rule for derivative of functions

Probabilistic Sensor Models

Scan observation models

- The map is given as a dense discrete representation (typically, a **grid map**)

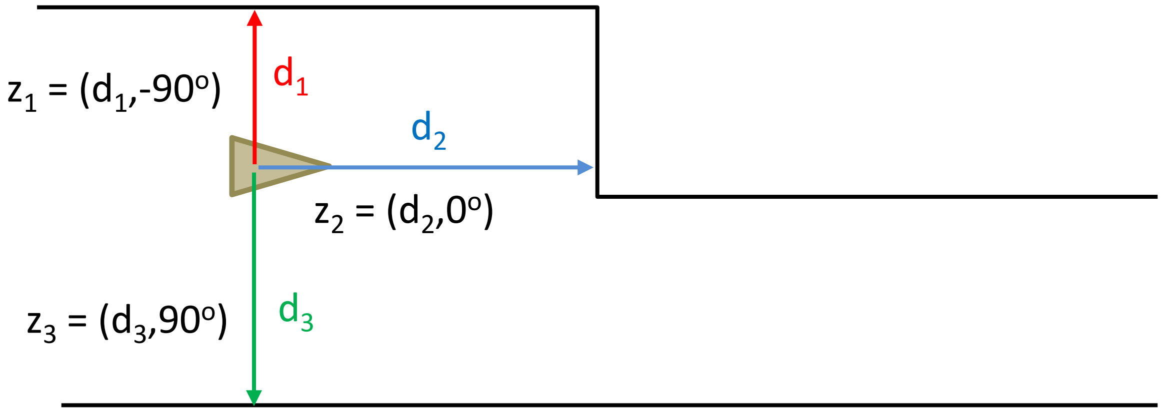
- ***Data Association*** no explicitly addressed

- Sensor **models**:
  - Beam model
  - Likelihood field

Beam model

Used for range sensors (i.e. sonar, infrared, laser), which give the distance (*range*) to the closest obstacle in the direction at which it points

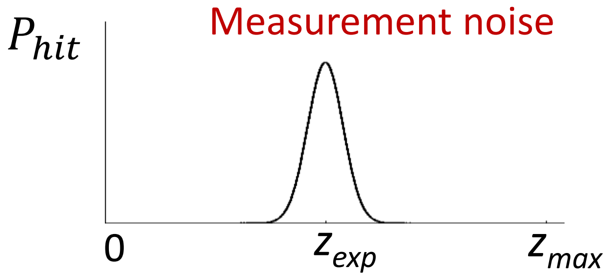
<u>Example</u>: Robot with 3 proximity sensors



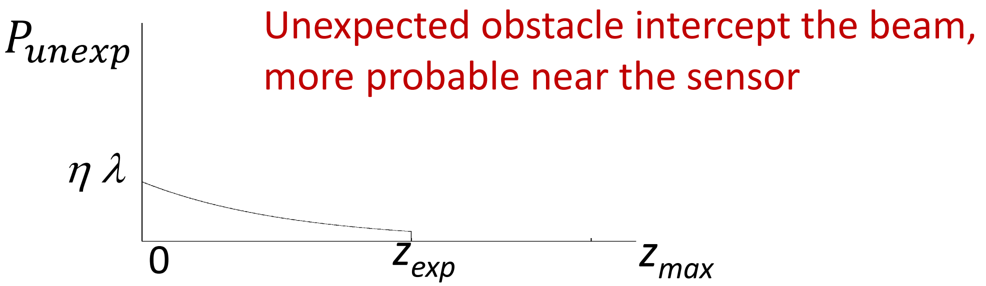
Usually, the angle is not considered a random variable (error free), only error in d, thus we seek the pdf ($z = d$):

$$p(d \mid x, m)$$

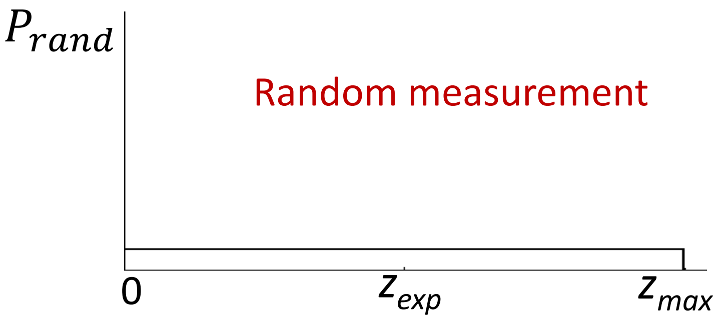
Beam model



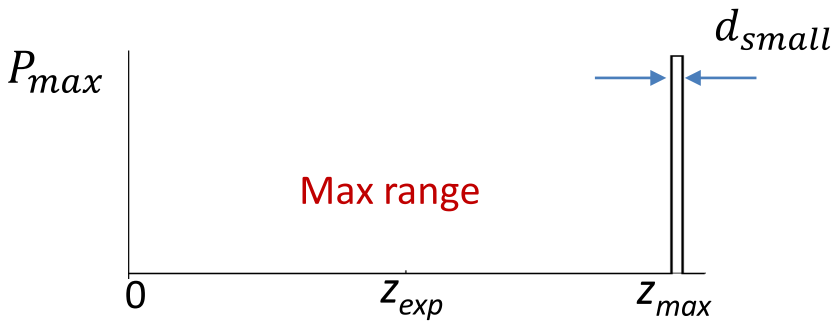
$$P_{hit}(z|x,m) = \eta \, \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\frac{\left(z-z_{exp}\right)^2}{\sigma^2}}$$



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



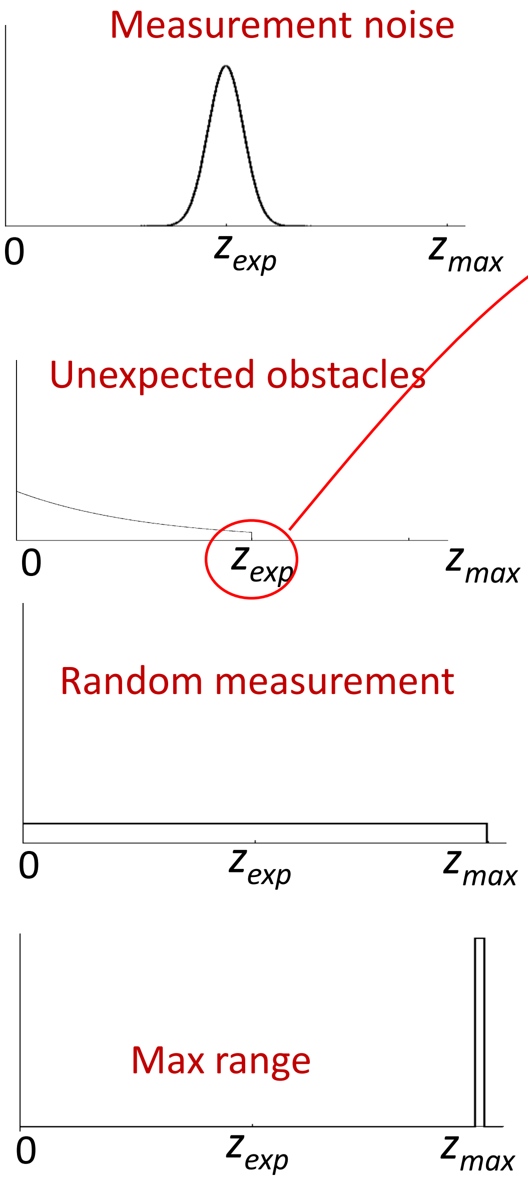
$$P_{rand}(z|x,m) = \eta \, \frac{1}{z_{max}}$$

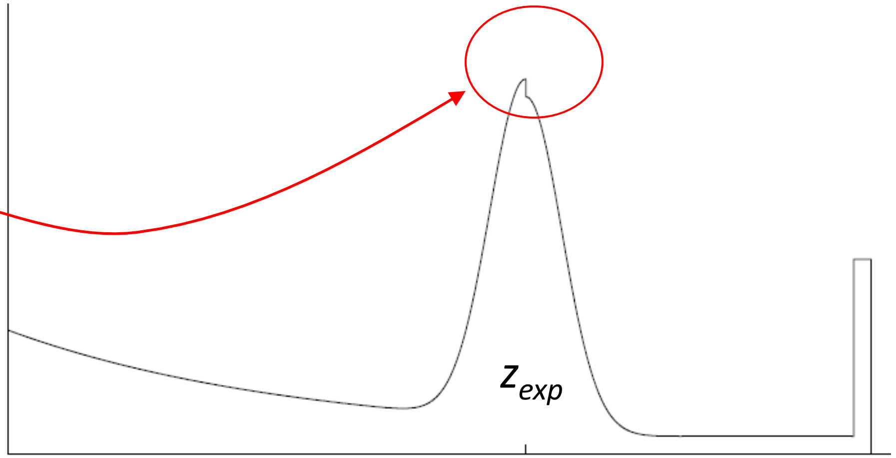


$$P_{max}(z|x,m) = \eta \, \frac{1}{d_{small}}$$

Z_{exp} given by ray casting over the known map (given x and m)

Observation probability for the Beam model





$$p(z|x,m) = \alpha_{hit} P_{hit}(z|x,m) + \alpha_{unexp} P_{unexp}(z|x,m) + \alpha_{rand} P_{exp}(z|x,m) + \alpha_{max} P_{max}(z|x,m)$$

To calculate the value of $p(z|x,m)$ for a given beam we need to trace the ray from the sensor pose x to the map (m) (**ray casting**). This gives us z_{exp}

Very costly for a laser scanner!!
(there are hundred of rays)

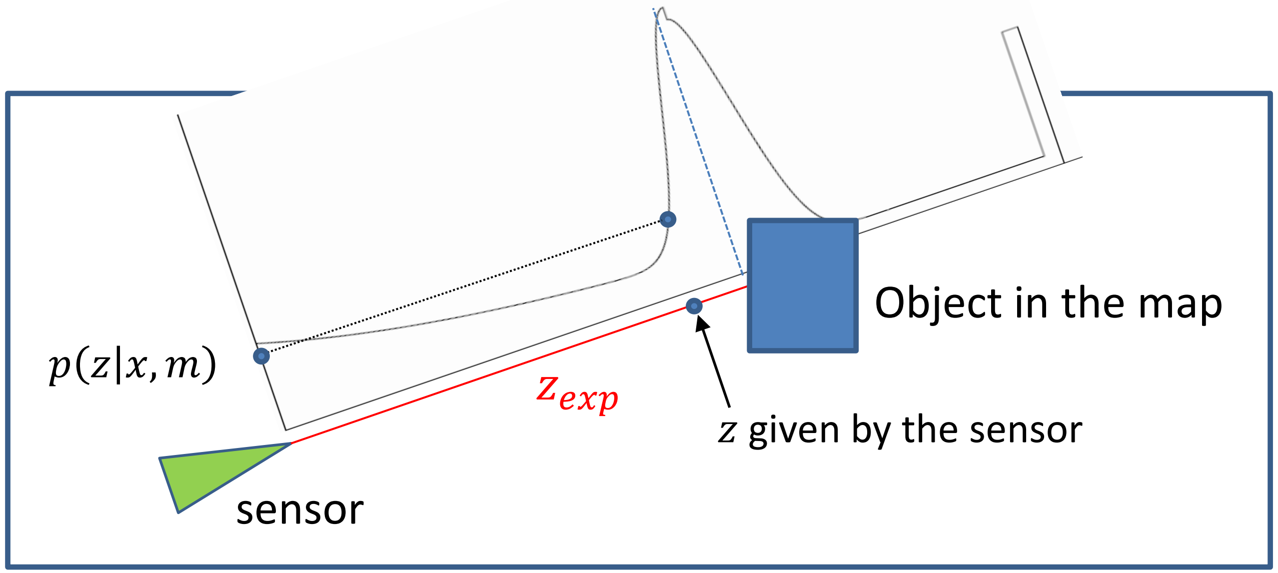
Remember that:

$$\int_0^\infty p(z|x,m)dz = 1$$

Beam model

How to calculate the value of $p(z|x,m)$ for a given beam:

- We trace the ray from the sensor pose x to the map (m) (ray casting)
- The first hit gives us z_{exp}



Drawbacks of the beam model:

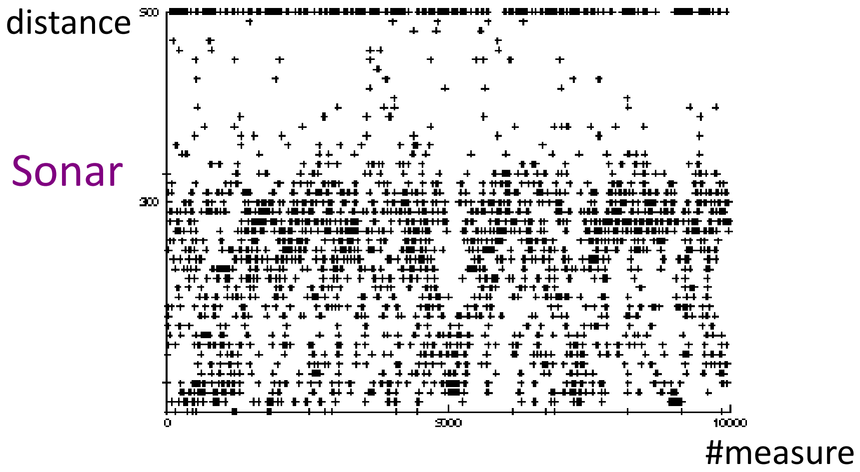
- not smooth for x because of small obstacles and edges
- not efficient (*ray casting* is computationally intensive)

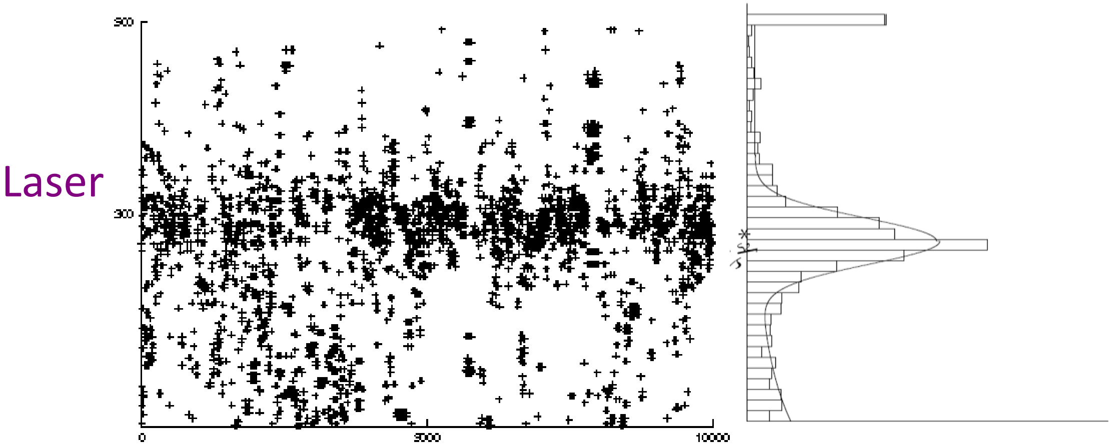
Beam model

$$p(z|x,m) = \alpha_{hit} P_{hit}(z|x,m) + \alpha_{unexp} P_{unexp}(z|x,m) + \alpha_{rand} P_{rand}(z|x,m) + \alpha_{max} P_{max}(z|x,m)$$

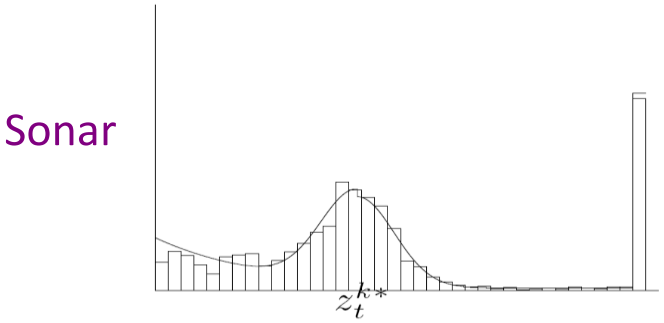
How can we determine the model parameters $\alpha_{hit}, \alpha_{unexp}, \alpha_{rand}, \alpha_{max}$?

Measured distances for expected distance of 300 cm.

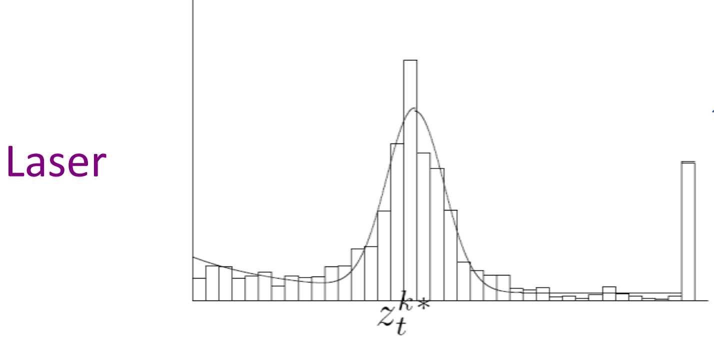




Maximize likelihood of the data



$P(z|z_{\text{experiment}})$



Beam model

map

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

robot pose

complete range scan containing K values

$q = 1$

for $k = 1$ to K do

Compute z_{exp} given x, m using ray casting, and from that $P_{hit}, P_{unexp}, P_{rand}, P_{max}$

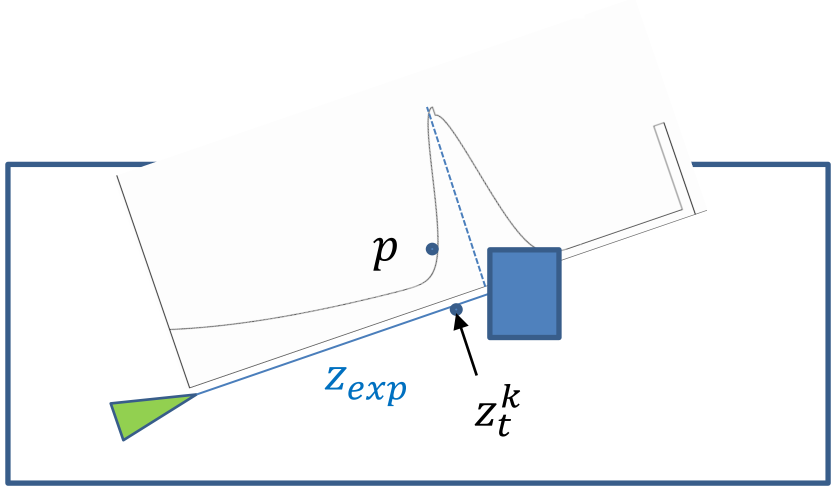
$$p = \alpha_{hit} P_{hit}(z_t|x,m) + \alpha_{unexp} P_{unexp}(z_t|x,m) + \alpha_{rand} P_{rand}(z_t|x,m) + \alpha_{max} P_{max}(z_t|x,m)$$

$q = q \cdot p$

return q

Likelihood of the complete range scan

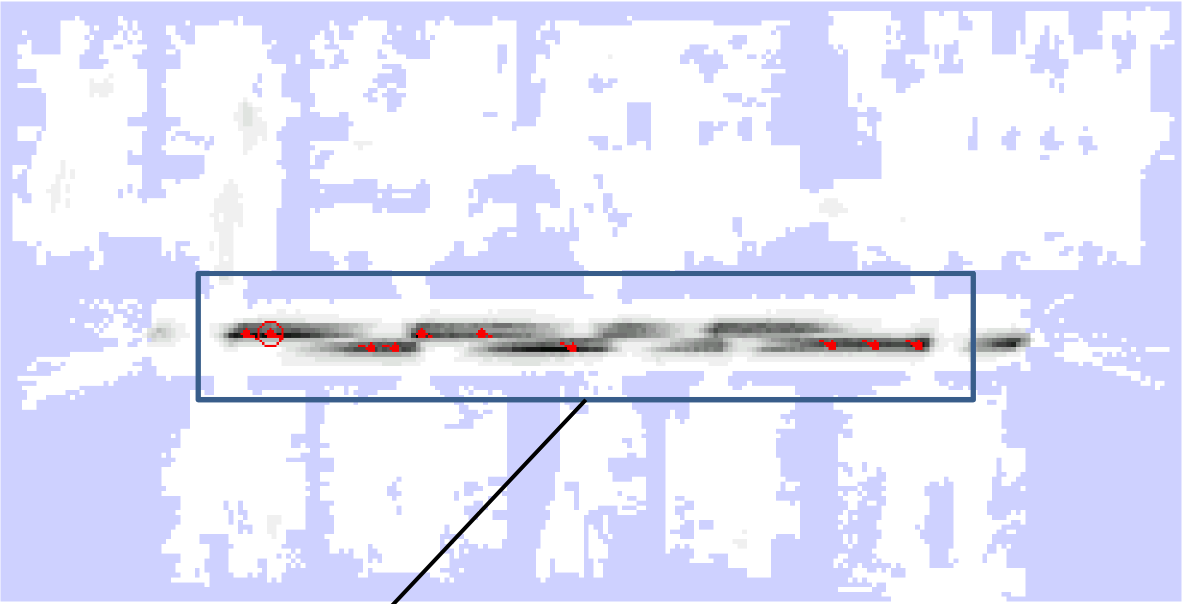
$$P(z \mid x, m) = \prod_{k=1}^{K} P(z_k \mid x, m)$$



Beam model

Example for a laser scanner

z

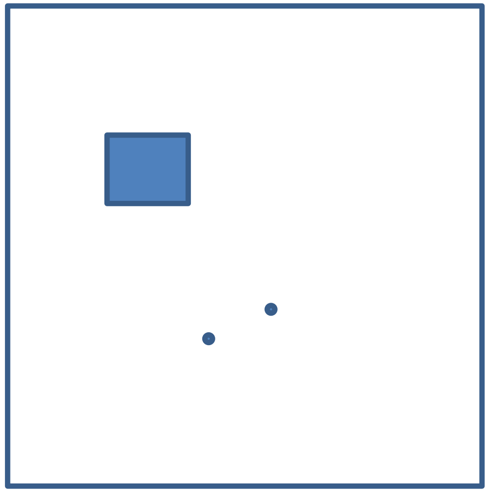


$P(z|x,m)$

Zoom

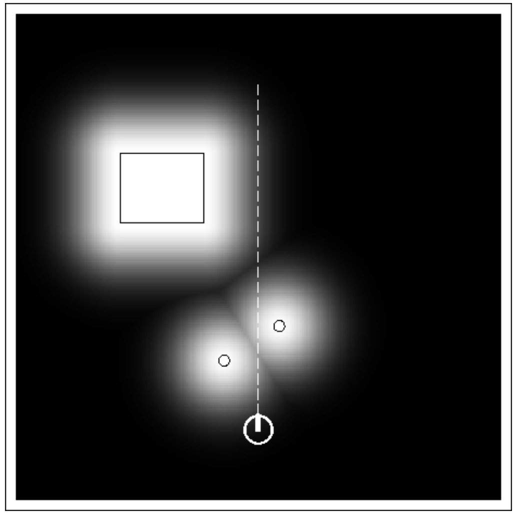
Likelihood Field model

- **Idea** of the *Likelihood Field model*: Instead of following along the beam (ray casting), only check the end point (this model is also known **endpoint model)**
- Given a map, we compute the likekihood field without any assumption where the sensor is.
- If the map is considered static, this is computed only once



Map *m*

The robot pose is not used



Likelihood field

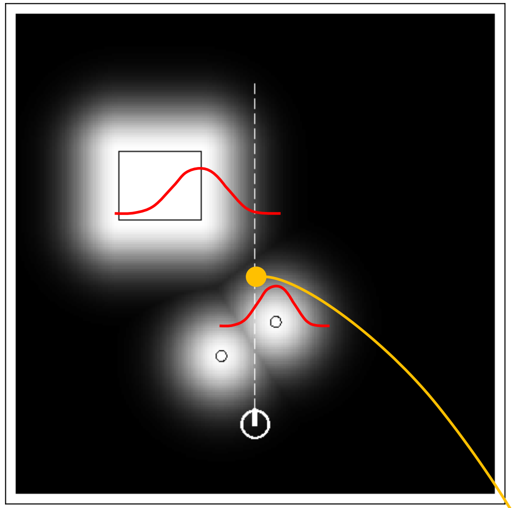
The brighter the bigger the likelihood

if you have a range sensor anywhere in the map, its measurements must lay following this distribution

Likelihood Field model

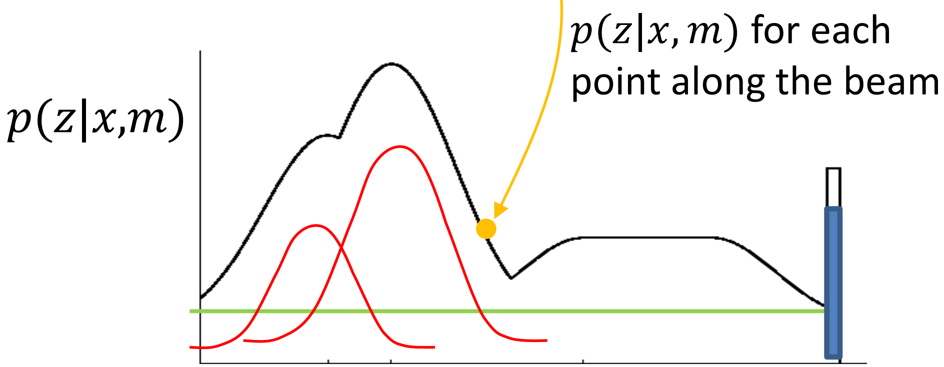
The likelihood at each point in the map is the combination of …

- a Gaussian distribution with mean at distance to closest obstacle
- a Uniform distribution for random measurements
- a small Uniform distribution for max range



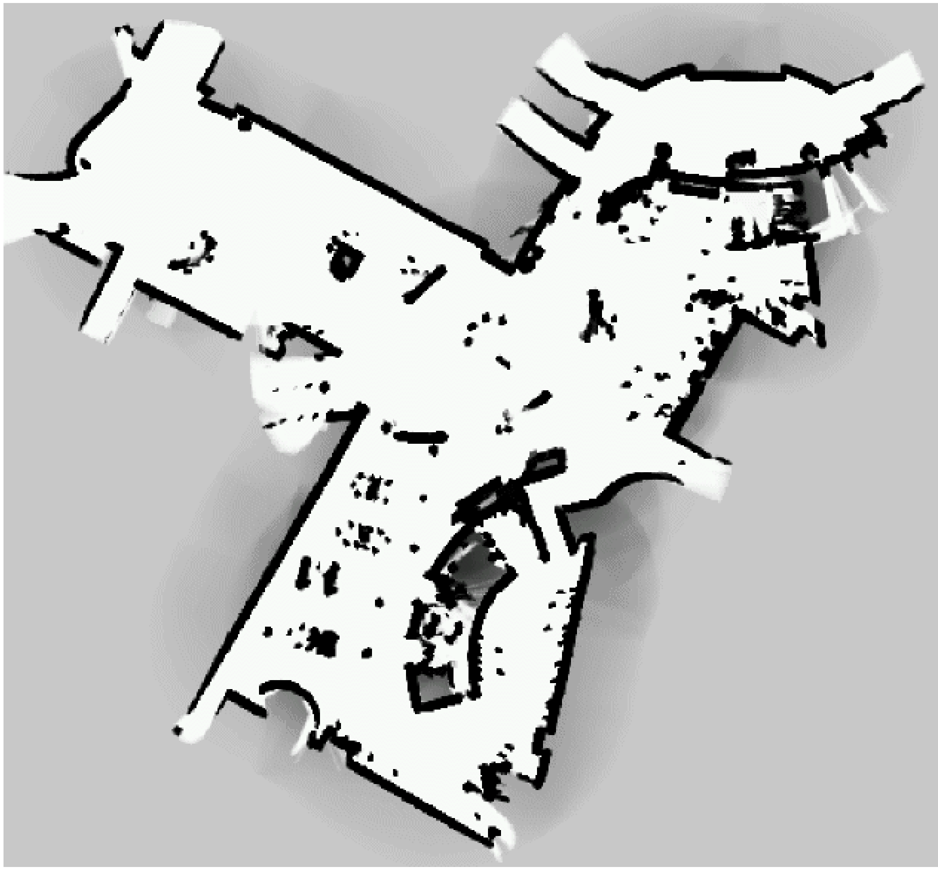
Likelihood field

How to compute the $p(z|x,m)$ for a given x:

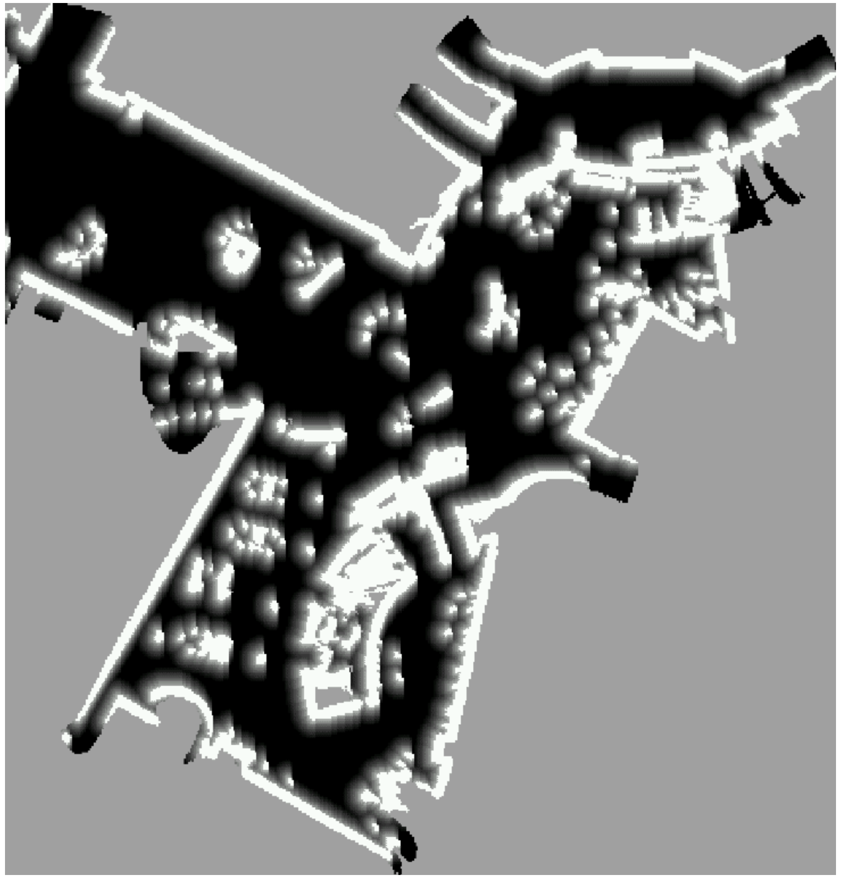


Likelihood Field model

Example



Occupancy grid map



Likelihood field

Likelihood Field model

Properties

- **Highly efficient**, uses 2D tables only
- **Smooth** w.r.t. to small changes in robot position
- Ignores physical properties of beams, e.g. the sonar/laser **beam passes through objects**.
- But, in practice, it works quite well!