Environment Sensing

Lecture 6
Environment Representation and Static Occupancy Grid Maps

Dr. Tamás Bécsi

Introduction

- One of the key challenges of any (AD) system lies in the perception and **representation** of the driving environment
- Semantical/Cognitive representation of the world is hard
- Environment representation is the buzzword version of mapping (in our scope). Since mapping can be either spatial or temporal, dynamic or static.

The „ideal" environment representation

- **Provides all information** necessary for implementing any ADAS or AD function
- **Compact enough** for transmission between electronic control units or V2V/V2I communication interfaces
- Is **generable real-time** in a computationally inexpensive, robust way with respect to sensor errors/malfunctions
- **Suppress irrelevant** environment **details** to facilitate situation interpretation and planning approaches
- **Represent free space** explicitly to permit safety-related trajectory planning
- Is able to handle **static and dynamic** objects
- Allow to incorporate **uncertainties**

Mapping is pre-computation

- From the point of view of self-driving technology, the mapping operation includes everything we can do to pre-compute things before the AV starts driving.
- Perception and localization of static objects in the world such as roads, intersections, street signs, etc. can be solved offline and in a highly accurate manner.
- **Without it, the AD/ADAS function has to figure out the whole background at the instant**

Mapping improves safety

- The use of maps for navigation is OK
- Though maps could adopt pragmatic best practices that reduce risk during driving
  - E.g. providing not only speed limits, but speed profiles
  - The vehicle can precondition itself for situations

Map is a unique sensor

- Has no range limitations
- It is immune to runtime occlusion from dynamic objects
- Can also be used for sensor fusion

Map can be a global shared state

- Just as any other Cloud service.
- Like a large full-information MMO game

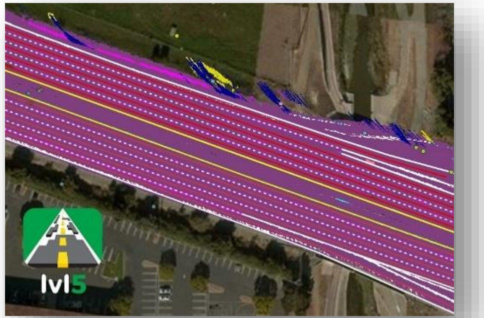
Hierarchical approach

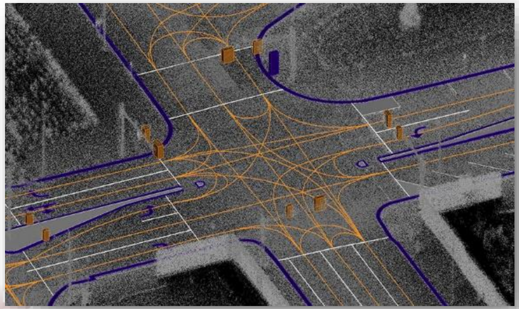
Dynamic	• Updated Real-time • Nearby vehicles, pedestrians, traffic signs
Semi-dynamic	• Minute based • Accidents, congestions, road works, weather
Semi-static	• Hourly • planned traffic/regulation changes, weather forecast
Static	• Monthly • Roads/lanes/3D structure (Basically the SD map)

HD maps

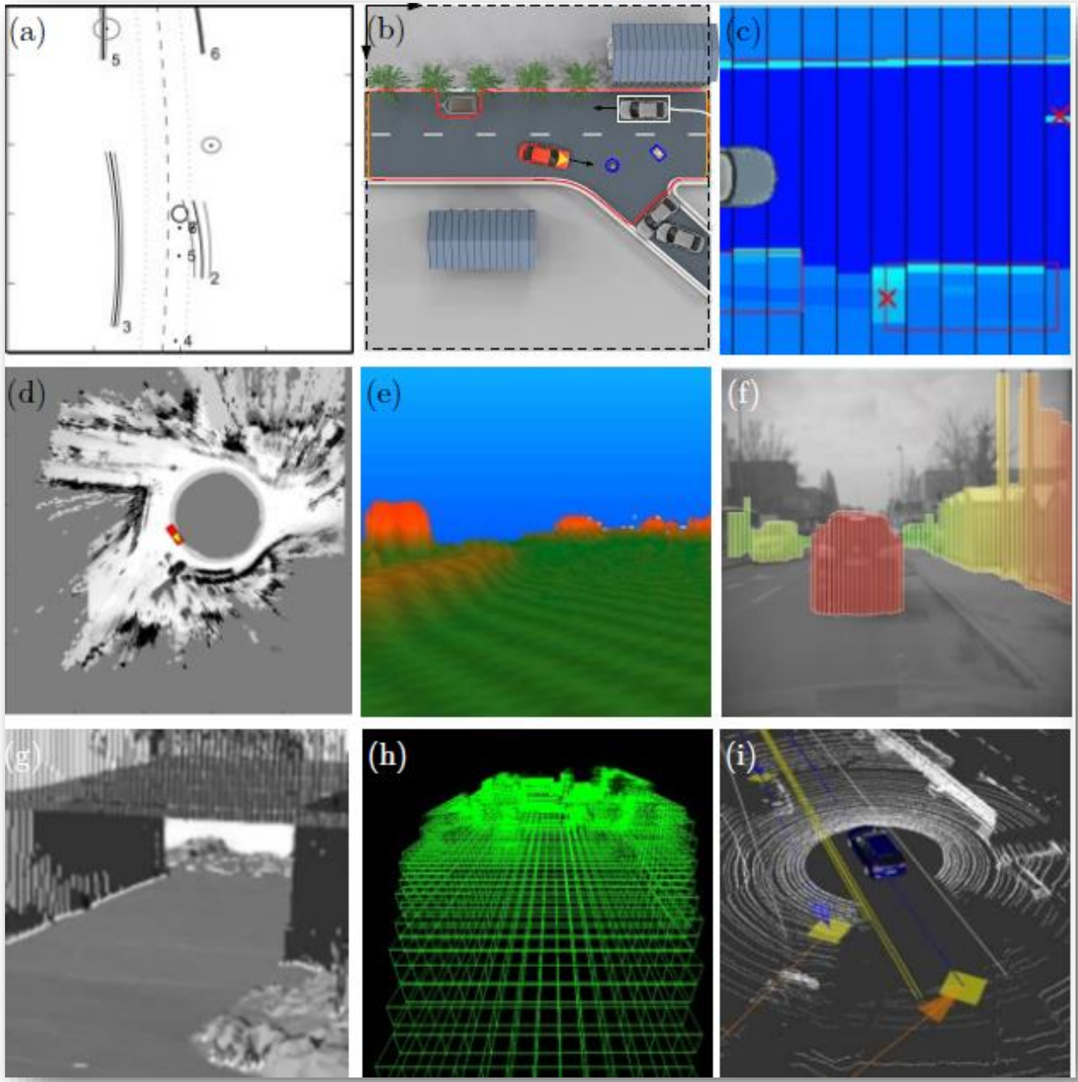
- Many participants,
- Not this course…
  - Localization and mapping
  - ADAS
  - etc…

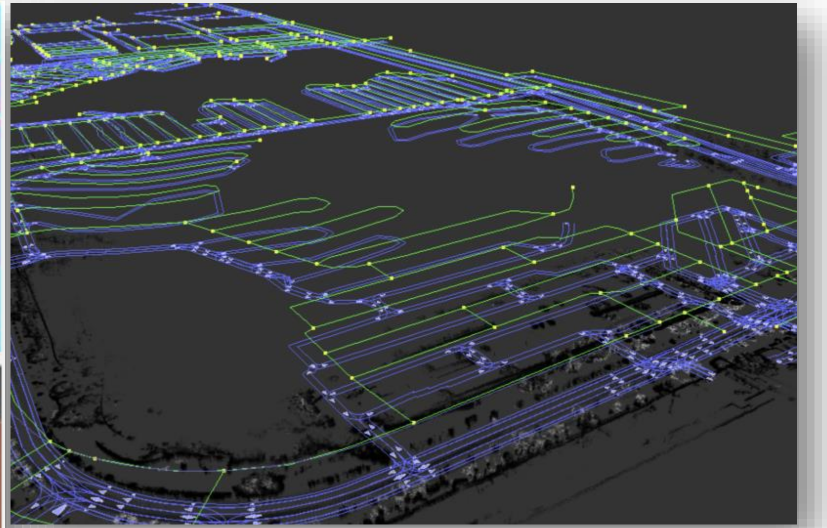
- HERE
- DeepMap
- Civil Maps
- Carmera
- TomTom
- lvl5
- Baidu

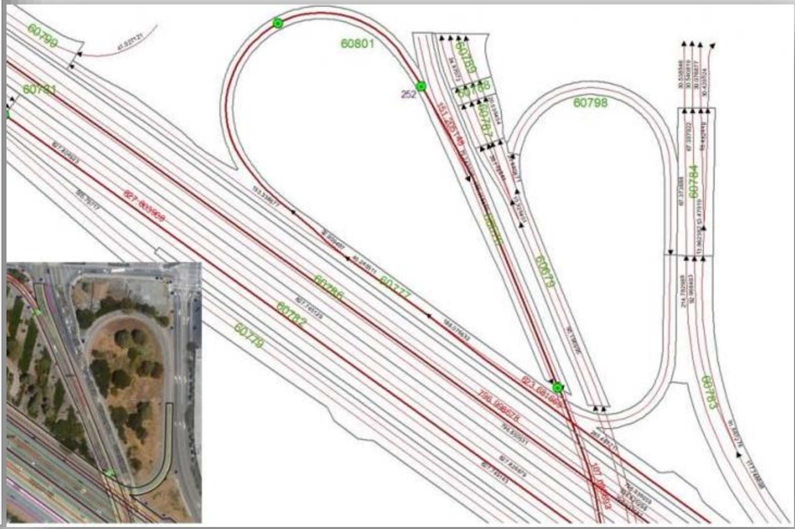




Feature, Volumetric, Semantic







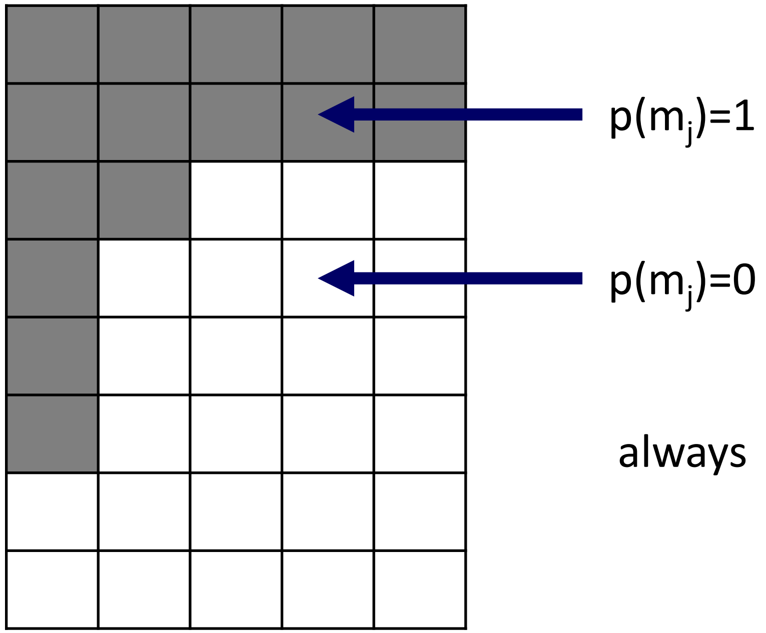
Feature Maps vs Grid Maps

- Natural choice for Kalman filter-based SLAM systems
- Compact representation
- Multiple feature observations improve the position estimate (EKF)

- Discretize the world into cells
- Grid structure is rigid
- Each cell is assumed to be occupied or free space
- Non-parametric model
- Require substantial memory resources
- Does not rely on a feature detector

Grid maps basics

- The area is simplified to 2D space,
- With equidistant grid,
- And each grid cell is a binary random variable, that models occupancy
  - Occupied: $p(m_j)=1$
  - Unoccupied: $p(m_j)=0$
  - Unknown: $p(m_j)=0.5$
- The grid is assumed to be static
- All cells have independent probability



Basics recap

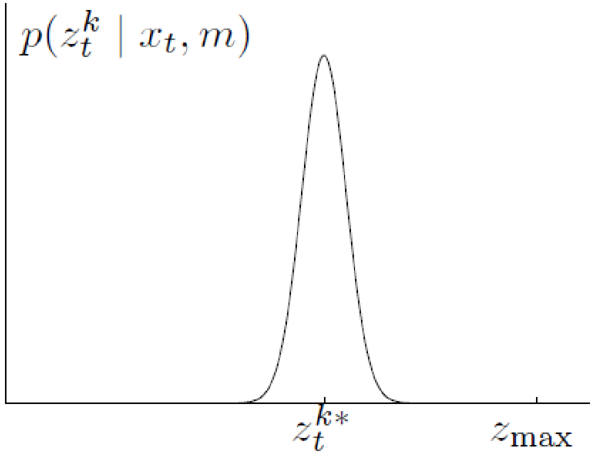
- m – the map, p(m) assumption
- m_i – one cell in the map, p(m$_i$) assumption
  - $p(m) = \prod p(m_i)$
- z_t - measurement in step t
- $z_{1:t} = \{z_1, z_2 \dots z_t\}$ set of measurements from step 1 to t
- All measurement are independent!
  - $p(z_2) = p(z_2|z_1)$ the current measurement is independent for all previous: $p(z_t) = p(z_t|z_{1:t-1})$
- x_t - the state of the sensor in step t

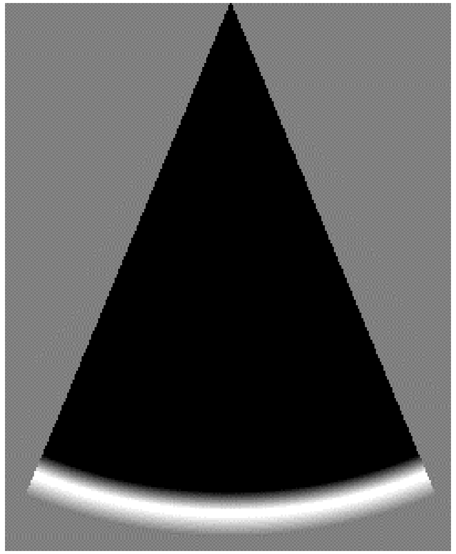
The sensor model of a proximity sensor

- The Sensor model states that
  - given a map m (actually the reality)
  - and a sensor state x_t
  - what is the probability function of the measurement?

$$p(z_t|x_t, m)$$

- Simply: What is the sensor output in a given scenario?



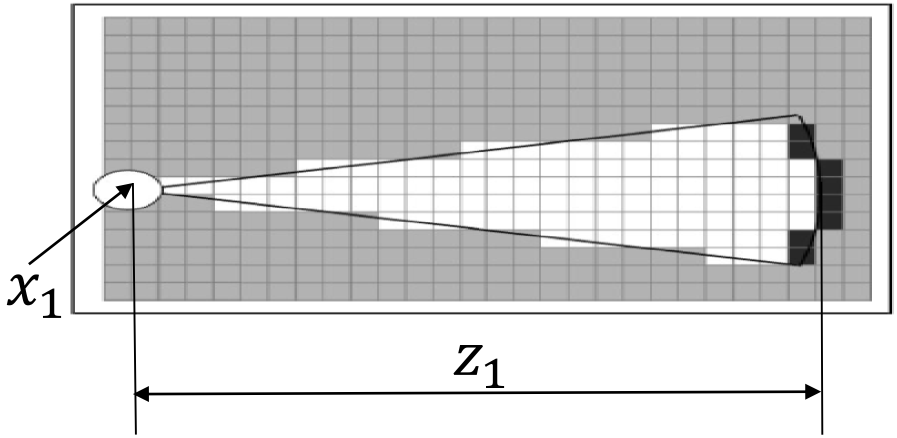


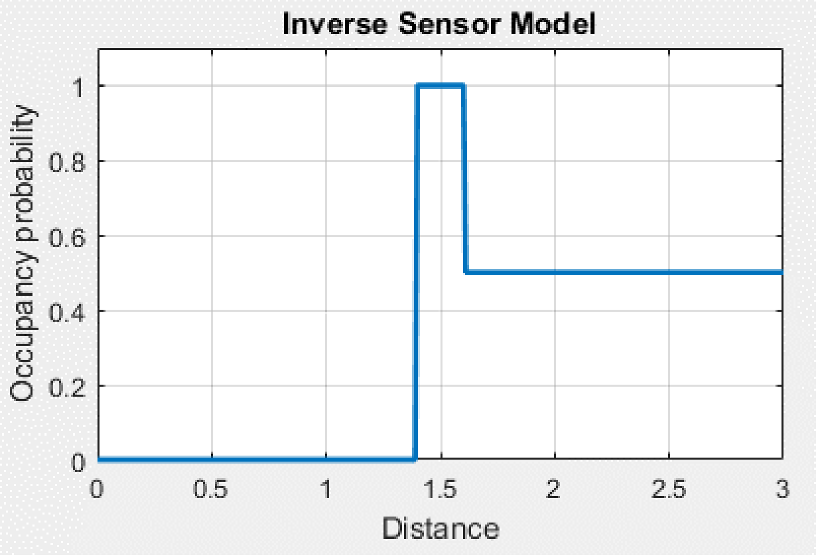
The inverse sensor model of a proximity sensor

- The Inverse Sensor model states that
  - given a measurement z_t
  - and a sensor state x_t
  - what is the probability function for the map cell m_i?

$$p(m_i|z_t, x_t)$$

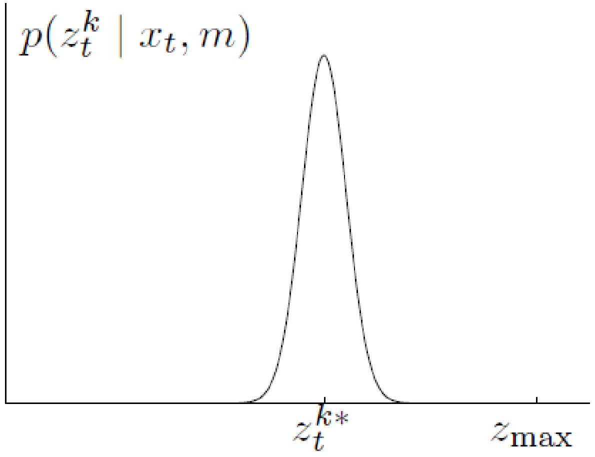
- Simply: What is the sensor output in a given scenario?

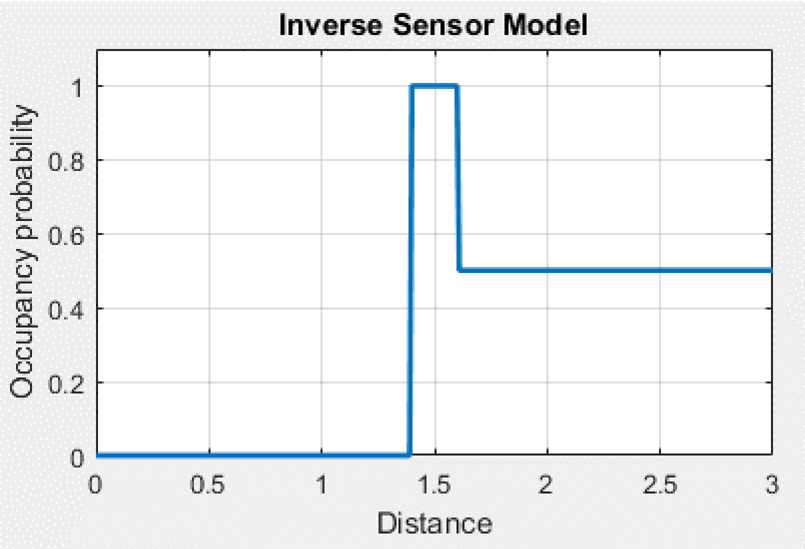


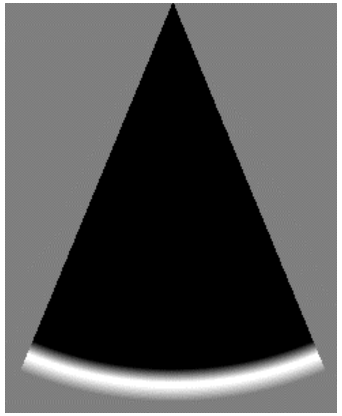


Mapping, „Naive” Approach 1.

- The sensor model for mapping is considered deterministic, with minimal noise on measurement
- The inverse sensor model states, that the sensor output has small deviation from the actual distance of the closest object.







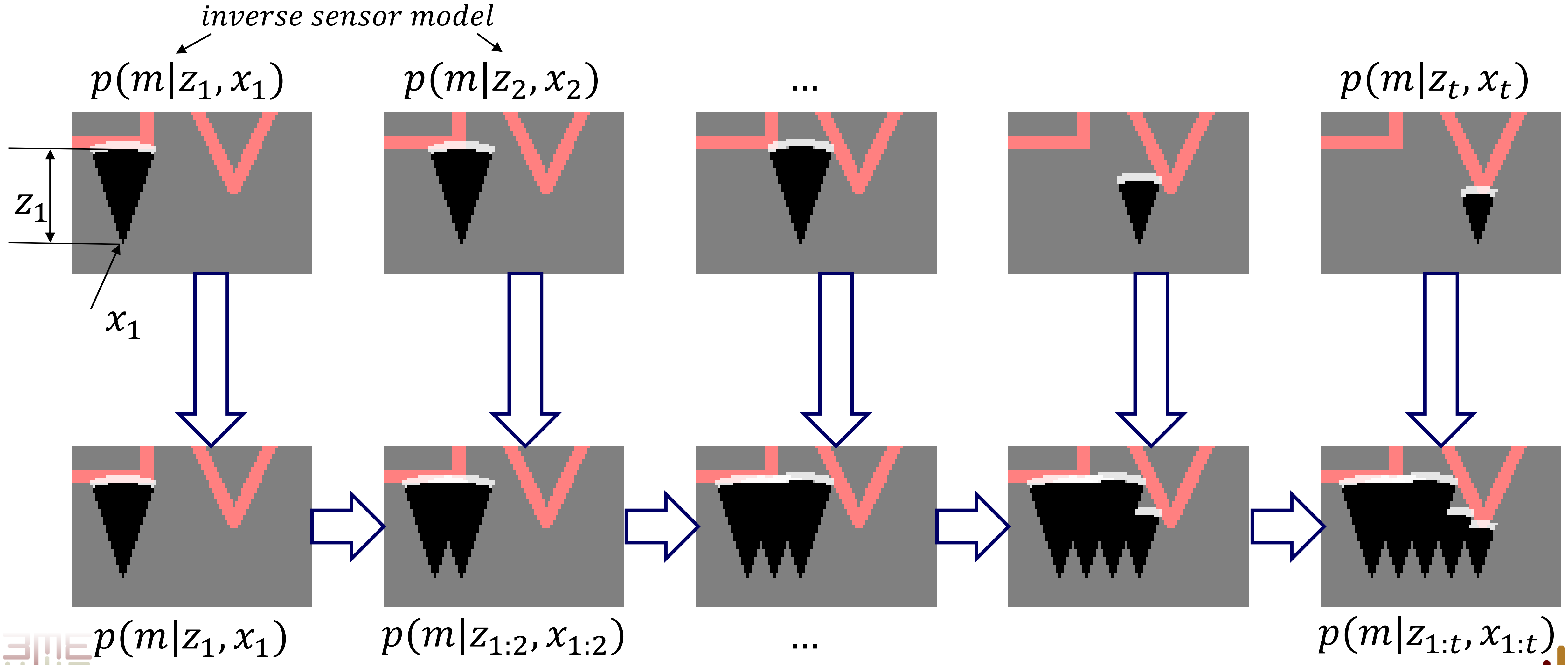
Mapping, Naive Approach 2.

- Without any probabilistic approach,

1. Initialize Map with unknown state for each cell (0.5)
2. Update each map cell $\mathbf{m_j}$ with its corresponding measurement $\mathbf{z_j}$ by following the rule:

z_j \ m_j	0	0.5	1
0	0	0	0
0.5	0	0.5	1
1	0	1	1

Demo


inverse sensor model
$p(m|z_1, x_1)$
$p(m|z_2, x_2)$
...
$p(m|z_t, x_t)$
z_1
x_1
$p(m|z_1, x_1)$
$p(m|z_{1:2}, x_{1:2})$
...
$p(m|z_{1:t}, x_{1:t})$

Demonstration Videos

1. Ideal Sensor

…

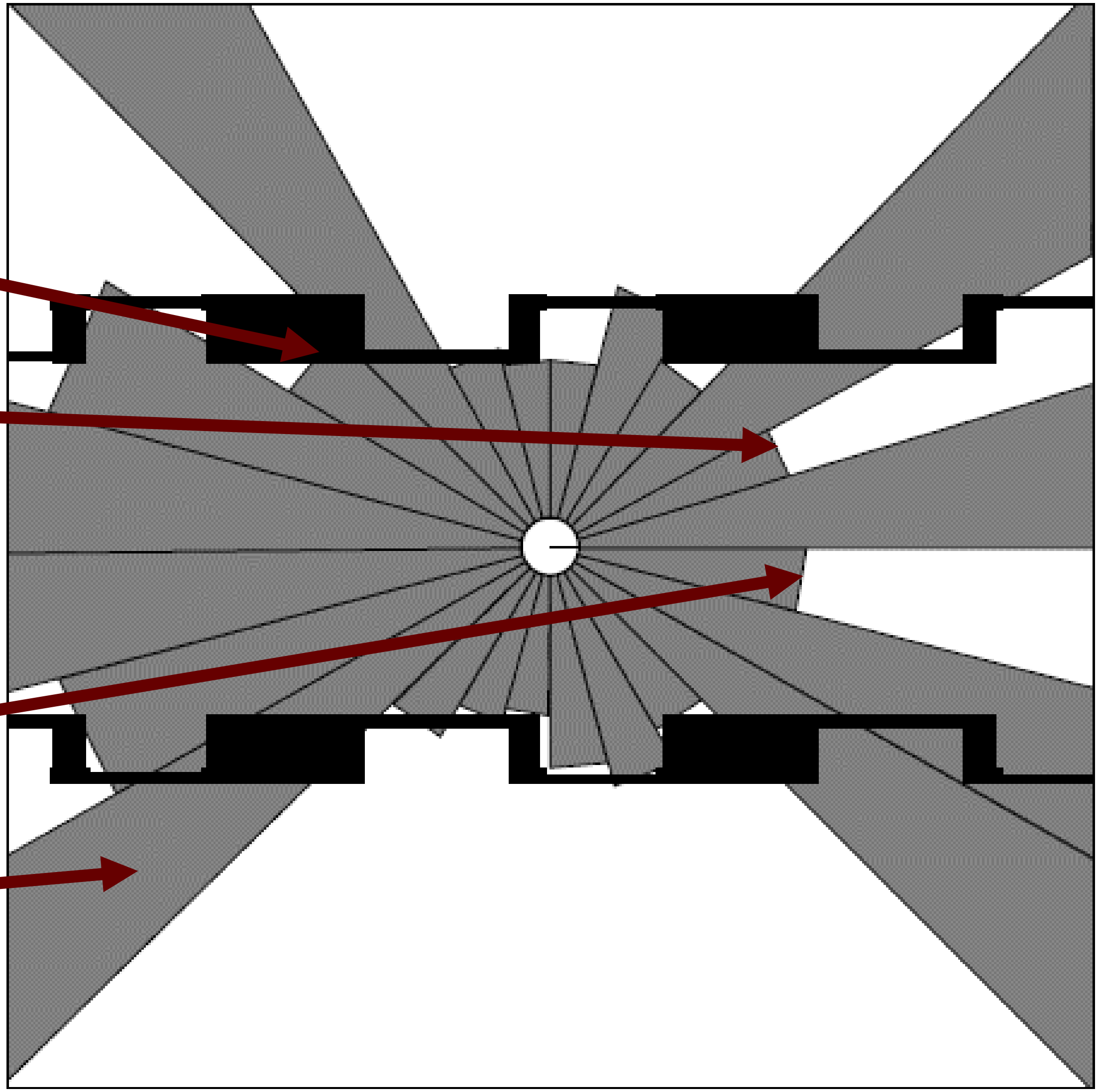
2. What if, we have Gaussian noise on the sensor?

…

And more…

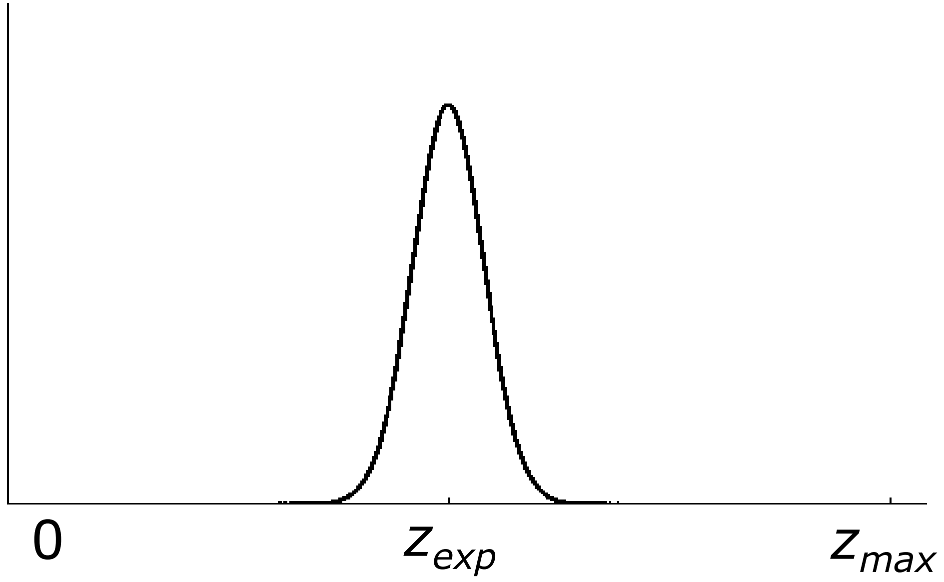
The Problem with the Naive approach

- Beams reflected by obstacles
- Beams reflected by persons / caused by crosstalk
- Random measurements
- Maximum range measurements



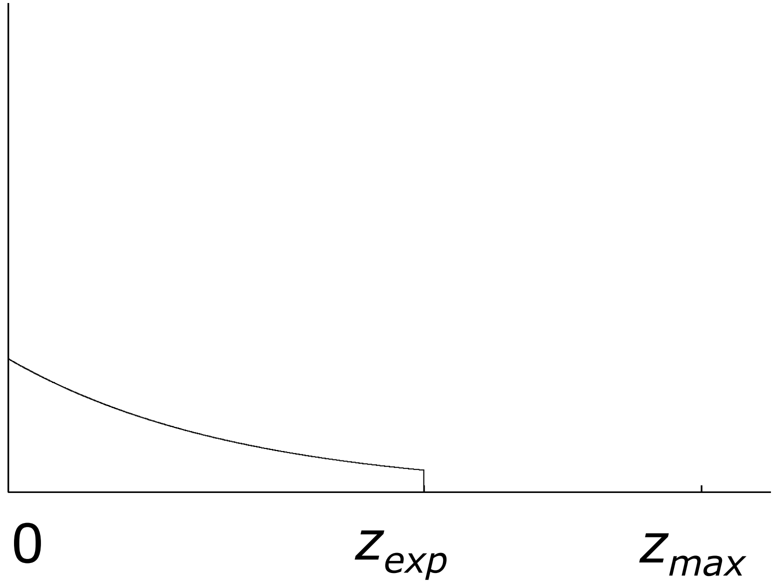
Sensor errors

Measurement noise



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

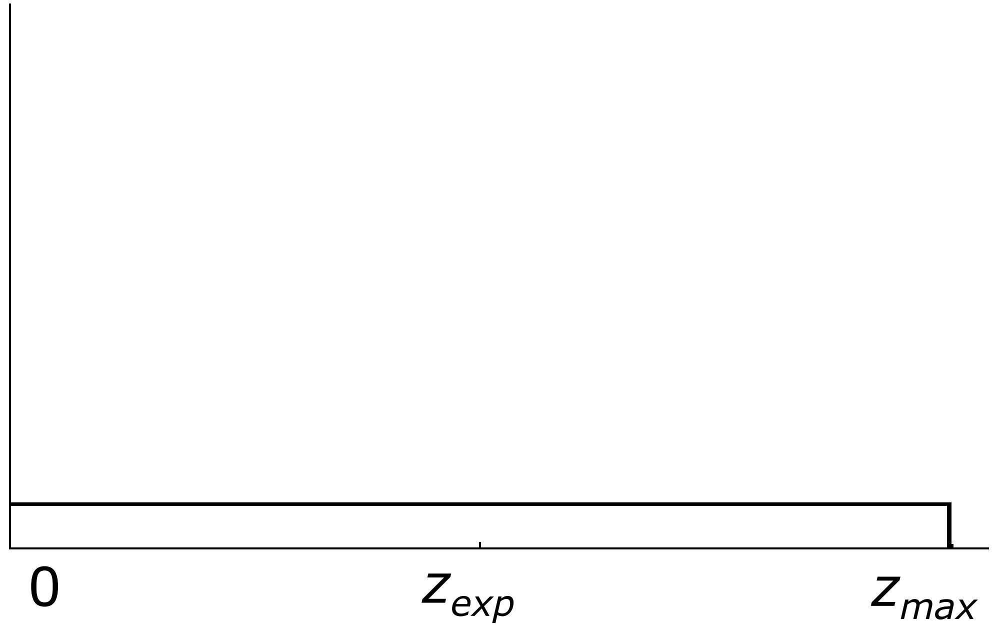
Unexpected obstacles



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

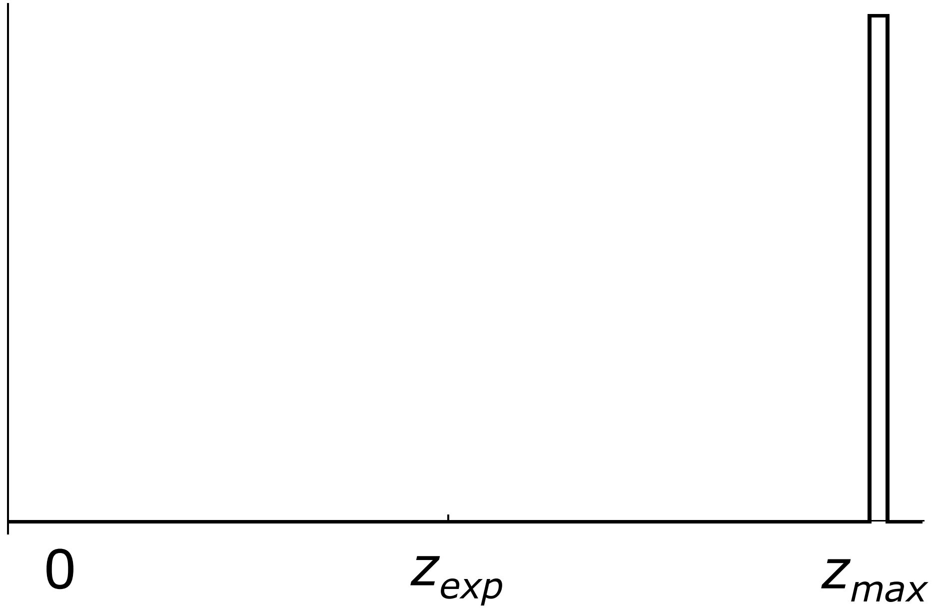
Sensor Errors

Random measurement



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

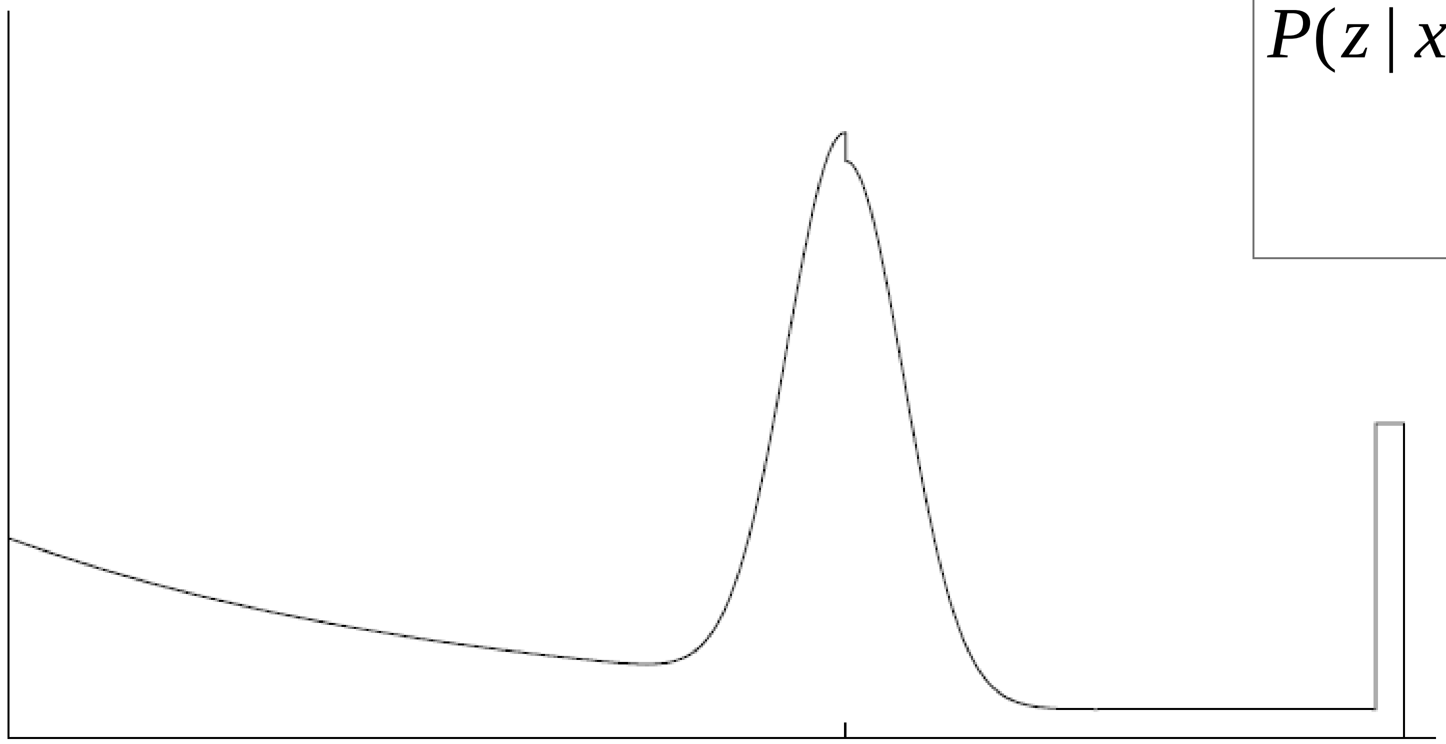
Max range



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

Resulting Mixture Density/Sensor modell

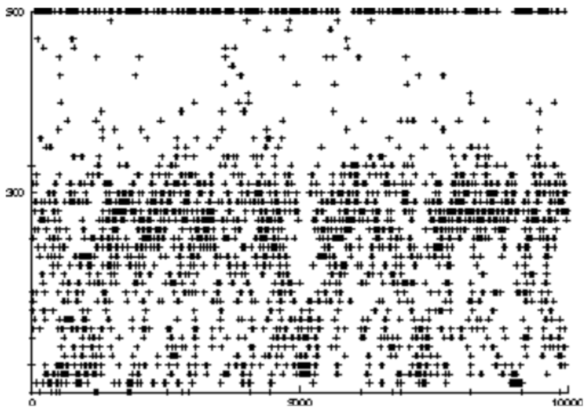
$$P(z \mid x, m) = \begin{pmatrix} \alpha_{\text{hit}} \\ \alpha_{\text{unexp}} \\ \alpha_{\text{max}} \\ \alpha_{\text{rand}} \end{pmatrix}^T \cdot \begin{pmatrix} P_{\text{hit}}(z \mid x, m) \\ P_{\text{unexp}}(z \mid x, m) \\ P_{\text{max}}(z \mid x, m) \\ P_{\text{rand}}(z \mid x, m) \end{pmatrix}$$



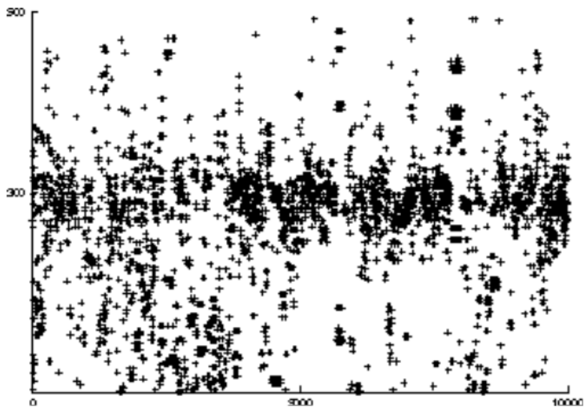
Demonstration Video

- With complex sensor error scheme

Determine model



Sonar

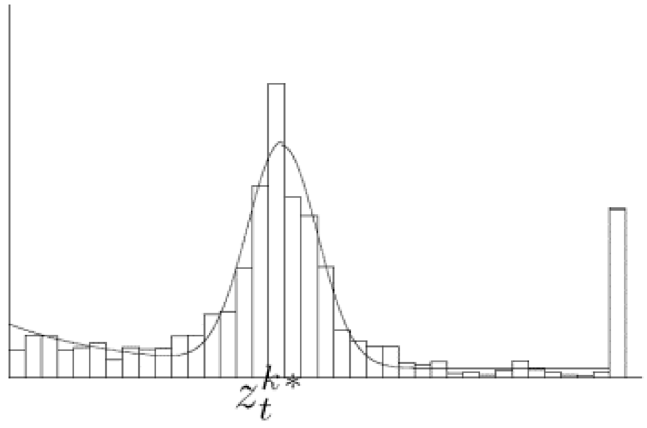


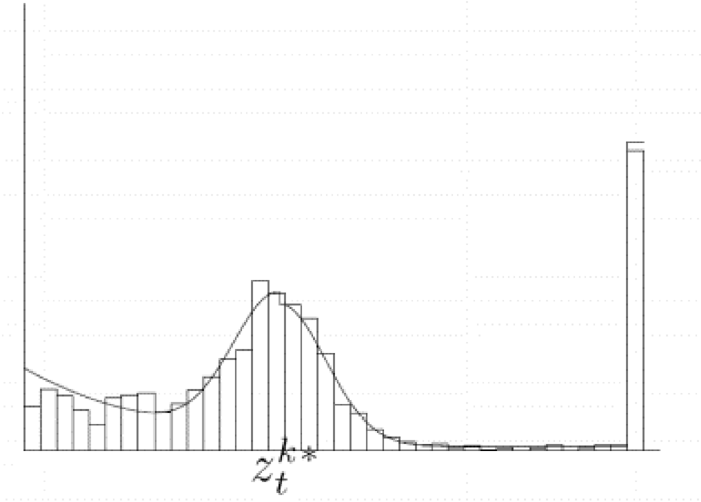
Laser

Mixture of

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

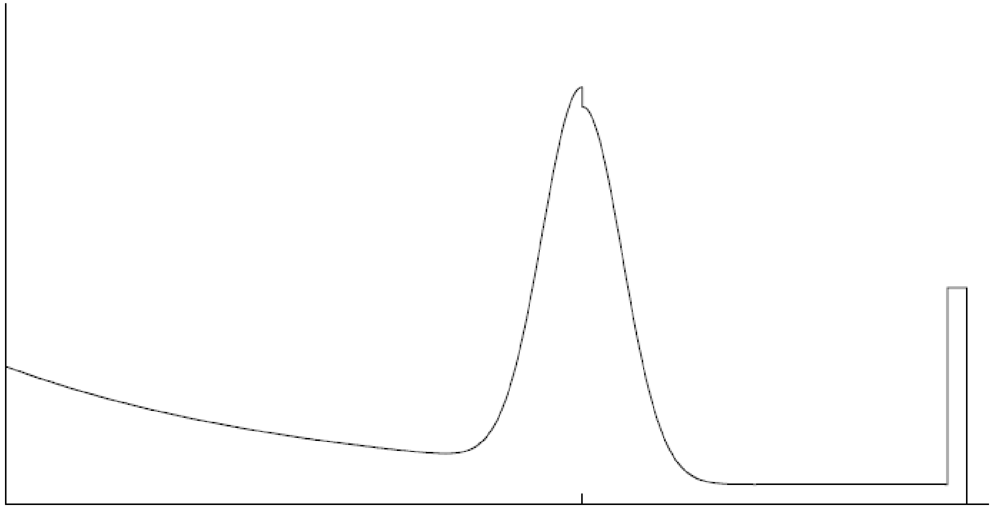
FIT

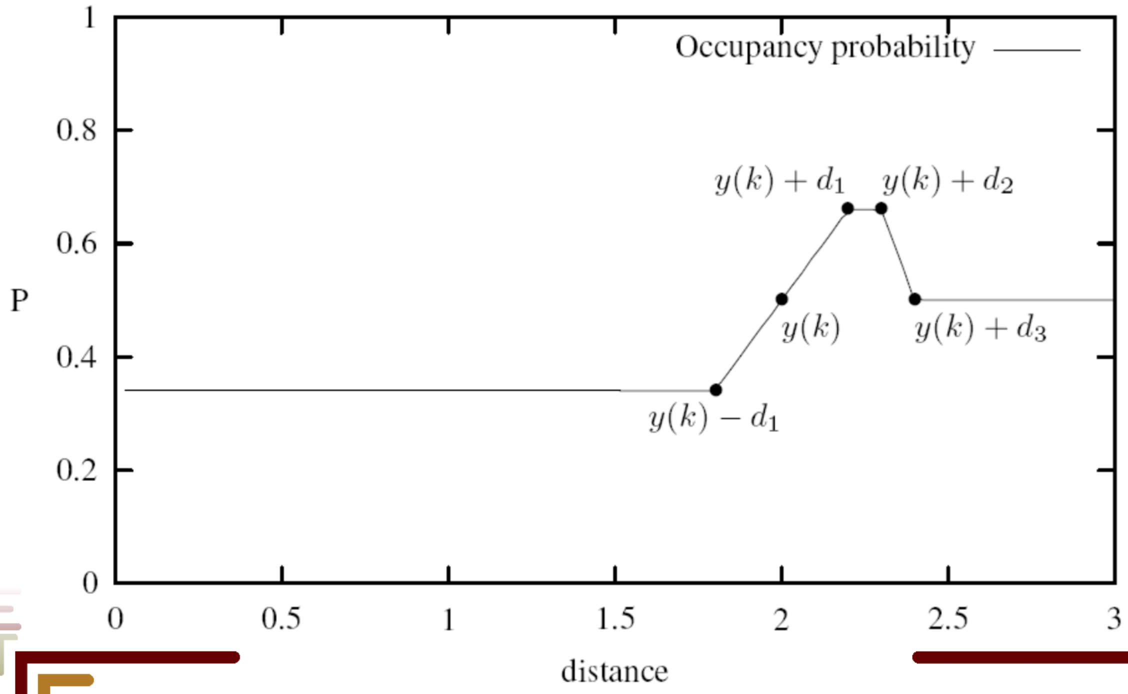




Probabilistic grid mapping

The sensor model for mapping is as described previously





The inverse sensor model should also consider this

Probabilistic grid mapping 1.

We want to determine the map m, based on all previous sensor states and measurements. Actually(1), since all cells are independent, we write the term for one cell:

$$p(m_i|z_{1:t}, x_{1:t}) = \frac{p(z_t|m_i, z_{1:t-1}, x_{1:t})p(m_i|z_{1:t-1}, x_{1:t})}{p(z_t|z_{1:t-1}, x_{1:t})}, \textit{by applying the Bayes Rule}$$

Actually(2), since we won't do anything with the sensor position mathematically, we omit it to make things clearer:

$$p(m_i|z_{1:t}) = \frac{p(z_t|m_i, z_{1:t-1})p(m_i|z_{1:t-1})}{p(z_t|z_{1:t-1})}$$

Estimating a Map From Data

Budapesti Műszaki és Gazdaságtudományi Egyetem
Közlekedésmérnöki és Járműmérnöki Kar
Közlekedés- és Járműirányítási Tanszék

$$p(m_i|z_{1:t}) = \frac{p(z_t|m_i, z_{1:t-1})p(m_i|z_{1:t-1})}{p(z_t|z_{1:t-1})}$$

the measurements are independent

$p(z_t|z_{1:t-1})=p(z_t)$

$$p(m_i|z_{1:t}) = \frac{p(z_t|m_i)(m_i|z_{1:t-1})}{p(z_t)}$$

another Bayes on $p(z_t|m_i)$

$$p(m_i|z_{1:t}) = \frac{p(m_i|z_t)p(z_t)(m_i|z_{1:t-1})}{p(m_i)p(z_t)}$$

simplify by $p(z_t)$

$$p(m_i|z_{1:t}) = \frac{p(m_i|z_t)p(m_i|z_{1:t-1})}{p(m_i)}$$

BME KJIT

Estimating a Map From Data

Current Measurement
(From inverse sensor model)

Map Based on all previous measurements
(Making the algorithm sequential)

$$p(m_i|z_{1:t}) = \frac{p(m_i|z_t)p(m_i|z_{1:t-1})}{p(m_i)}$$

Prior assumption on the map (generally 0.5)

Static State Binary Bayes Filter – Though multiplication causes some problems

Further improvement

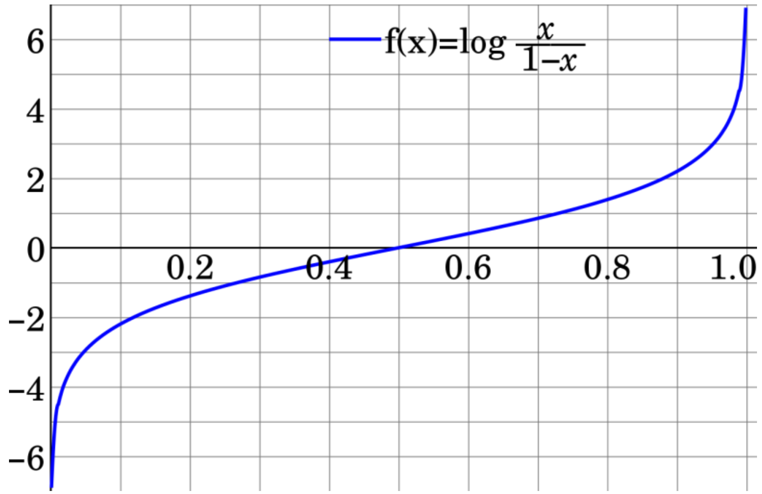
$$p(\neg m|z_{1:t}) = \frac{p(\neg m|z_t)p(\neg m|z_{1:t-1})}{p(\neg m)}$$

- Ratio off probabilities:

$$\frac{p(m|z_{1:t})}{p(\neg m|z_{1:t})} = \frac{p(m|z_{1:t})}{1-p(m|z_{1:t})} = \frac{p(m|z_t)}{1-p(m|z_t)}\frac{p(m|z_{1:t-1})}{1-p(m|z_{1:t-1})}\frac{1-p(m)}{p(m)}$$

Measurement　　Recursion　　Prior

Using log odds notation



$$l(x) = log\frac{p(x)}{1-p(x)};$$

$$p(x) = 1 - \frac{1}{1+\exp(l(x))}$$

$$l(m|z_{1:t}) = l(m|z_t) + l(m|z_{1:t-1}) - l(m)$$

$$l_t = inverse.sensor.model + l_{t-1} - l_0$$

- Demonstration video

Summary

- Occupancy grid maps discretize the space into Independent cells
- Each cell is a binary random variable estimating if the cell is occupied
- Static state binary Bayes filter per cell
- Mapping with known poses is easy
- Log odds model is fast to compute
- No need for predefined features

Summary II.

- „Known" poses is questionable
- Short term quasi-known poses are OK
- wide-angle sensors are still a problem, with biased measurements
- „independent cell" <-> „independent measurement" assumption makes it weaker

References

- Rethinking Maps for Self-Driving, Medium online: https://medium.com/@LyftLevel5/https-medium-com-lyftlevel5-rethinking-maps-for-self-driving-a147c24758d6
- SCHREIER, Matthias. Environment representations for automated on-road vehicles. *at-Automatisierungstechnik*, 2018, 66.2: 107-118.
- Sebastian THRUN, Wolfram BURGARD, Dieter FOX: PROBABILISTIC ROBOTICS Chapters 4.2; 6; 9.1-2
- Robot Mapping http://ais.informatik.uni-freiburg.de/teaching/ws18/mapping/
- Mitsui & Co. Global Strategic Studies Institute Monthly Report July 2018:DEVELOPMENT OF DYNAMIC MAP FOR AUTOMATED DRIVING AND ITS POTENTIAL TO BE NEXT-GEN INDUSTRIAL INFRASTRUCTURE