EECE 5550
Mobile Robotics

Lecture 9: Mapping

Derya Aksaray

Assistant Professor

Department of Electrical and Computer Engineering

Northeastern University

Recap

Last week: Bayesian networks and the Bayes Filter

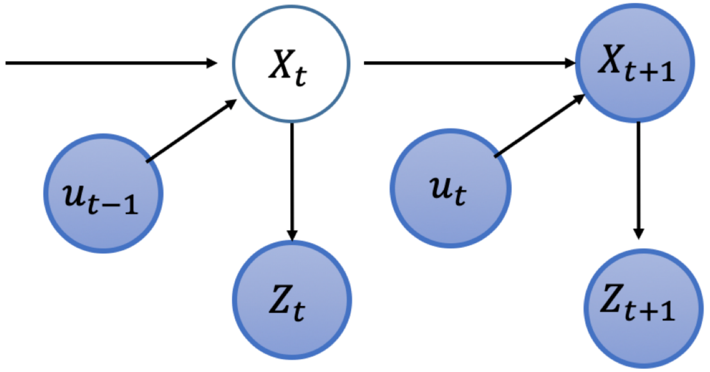
Bayes Filter: For t = 1, 2 … repeat the following operations:

- **Predict** belief for current state X_t given previous control u_{t-1}:

$$p(X_t|u_{0:t-1}, Z_{1:t-1}) = \int p(X_t|X_{t-1}, u_{t-1}) \cdot p(X_{t-1}|u_{0:t-2}, Z_{1:t-1})\ dX_{t-1}$$

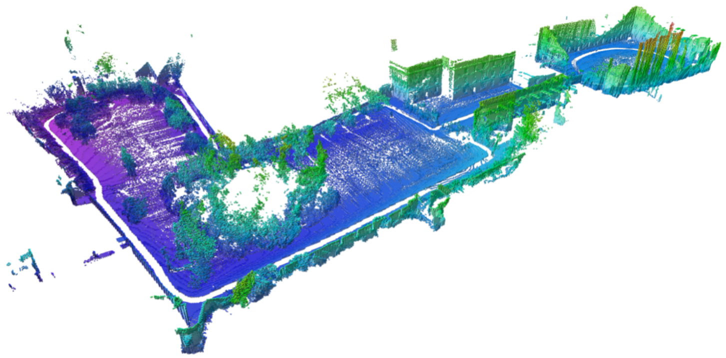
- **Update** belief after incorporating measurement Z_t at current state X_t:

$$p(X_t|u_{0:t-1}, Z_{1:t}) = \frac{p(Z_t|X_t)p(X_t|u_{0:t-1}, Z_{1:t-1})}{\int p(Z_t|X_t)p(X_t|u_{0:t-1}, Z_{1:t-1})\ dX_t}$$



Next two weeks: Mapping and localization!

1. Mapping
2. Localization
3. The Big One: Simultaneous localization and mapping (SLAM)

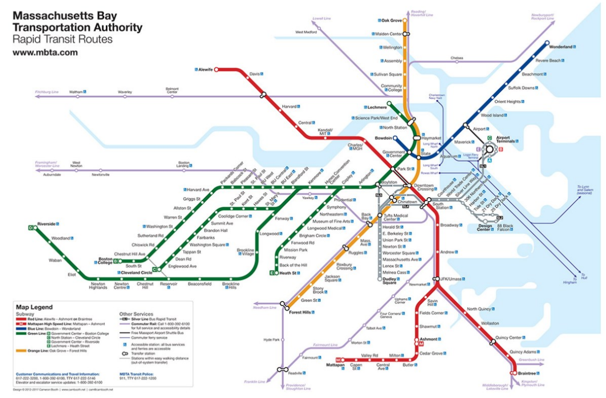


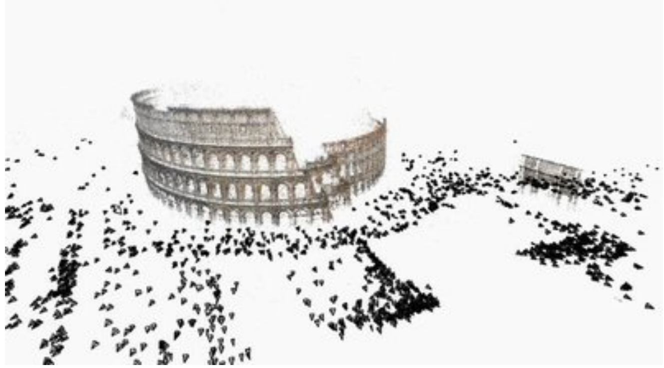
What is a map?

Simple answer: A *map* is a simply a *model* of an environment.

Less simple questions:

- What *properties* of the environment should a map capture?
- How should these properties be *represented* in the map?





Examples:

Connectivity/traversability: Topological maps (graph)

- Good for: Route planning

Free and occupied space: Occupancy grids / voxels

- Good for: 2D / 3D navigation

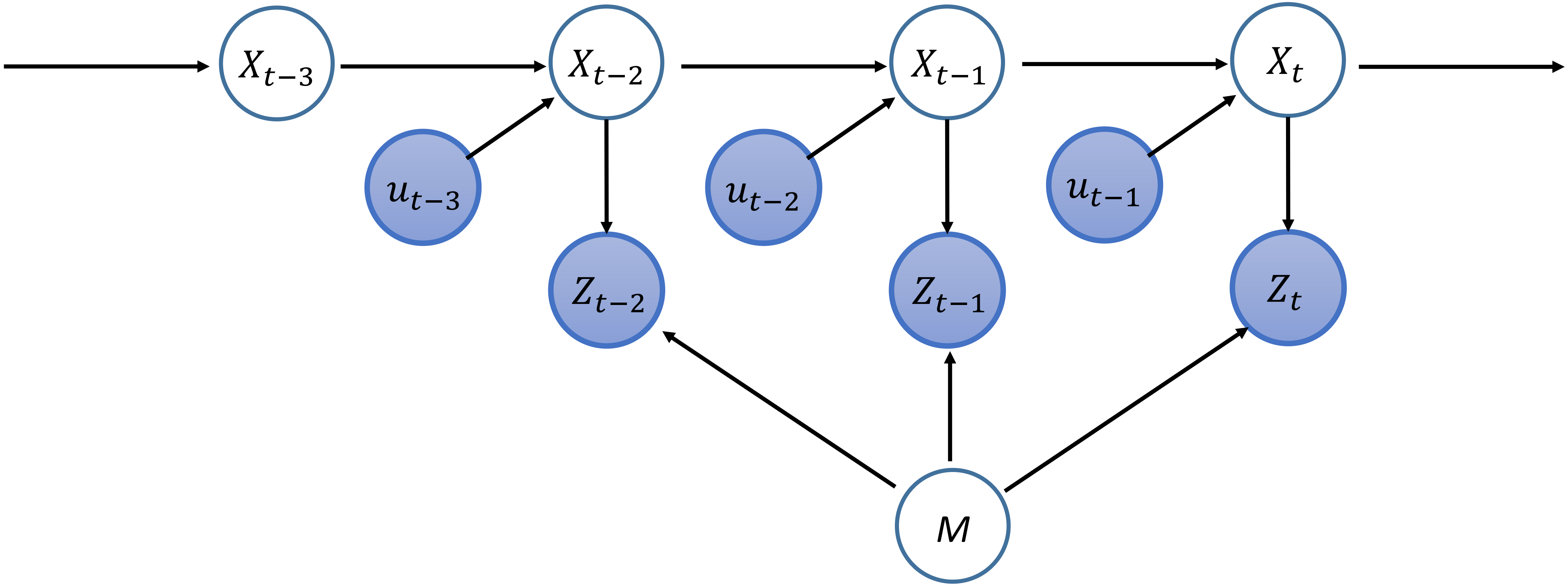
Appearance: Visual maps (keyframes)

- Good for: Visual reconstruction, place recognition

Objects & affordances: Semantic maps, scene graphs, etc.

- Good for: high-level autonomy, environmental interaction

The Mapping Problem as a Bayesian Network

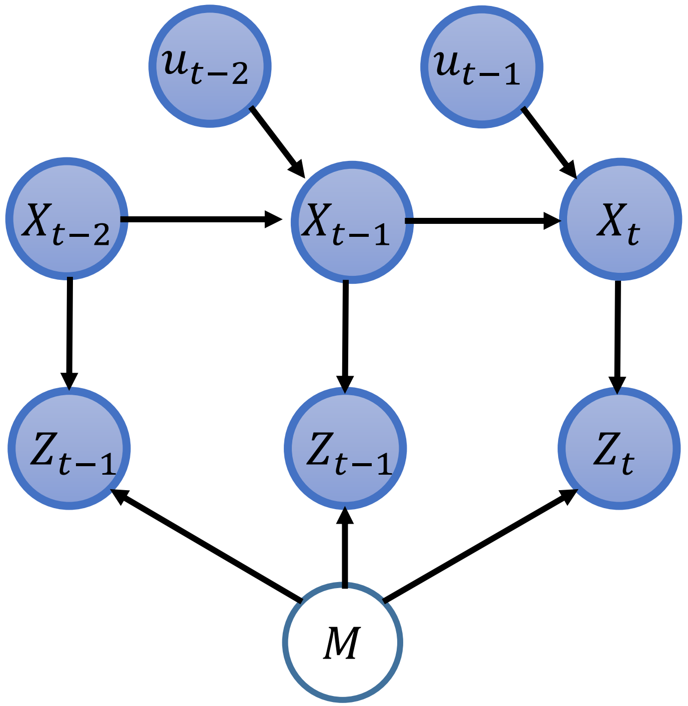


Two fundamental problems in robotic perception

Mapping

Given: Robot poses $x_{0:t}$, measurements $z_{1:t}$

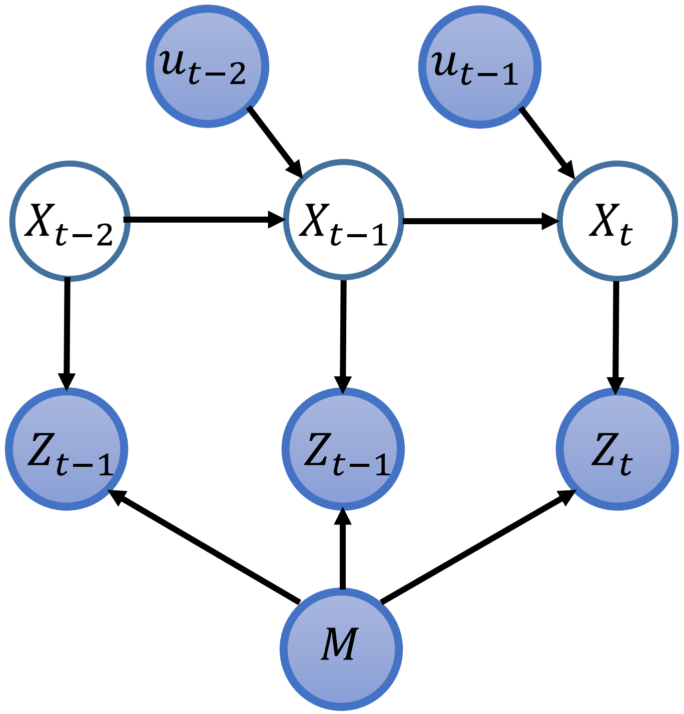
Estimate: Belief $p(m|x_{0:t}, z_{1:t})$ over the map *M*



Localization

Given: Map *m*, measurements $z_{1:t}$

Estimate: Belief $p(x_t| \, m, z_{1:t})$ over the robot pose

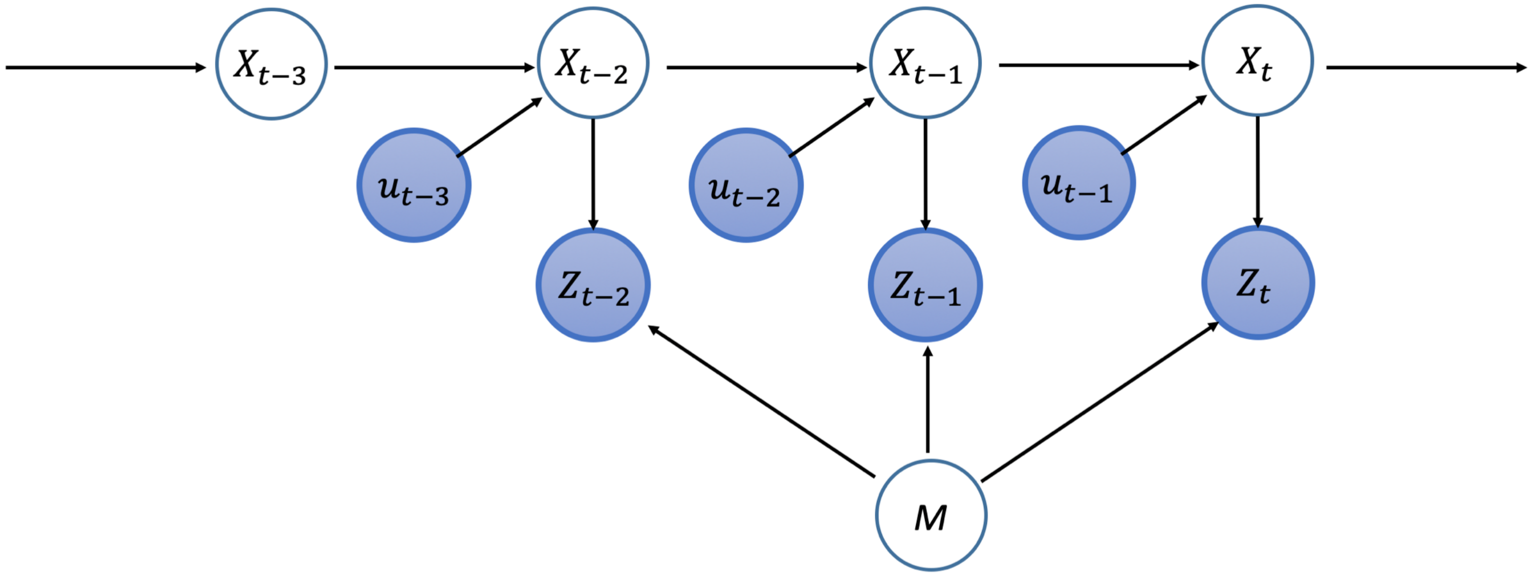


Key observation:
Given *either* poses $x_{0:t}$ *or* map *m*, we can efficiently recover the other via *recursive Bayesian estimation* (Bayes Filter)!

Simultaneous Localization and Mapping (SLAM)

Given: Prior $p(x_0)$ for initial pose x_0, sequence of motor commands $u_{0:t-1}$, observations $z_{1:t}$

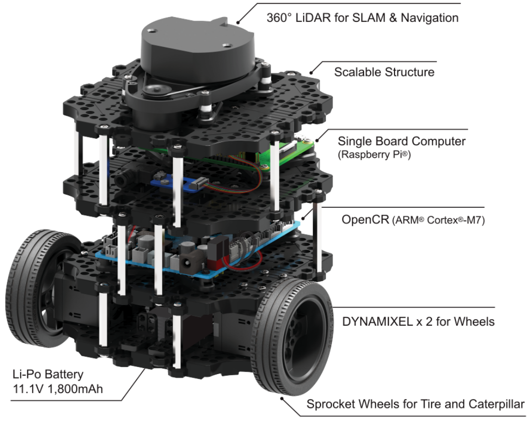
Estimate: Joint posterior $p(x_{0:t}, m \,|u_{0:t-1}, z_{1:t})$ over robot poses $x_{0:t}$ and map m given $u_{0:t-1}$ and $z_{1:t}$



Key point: This coupled problem is *much* harder than either mapping or localization alone.

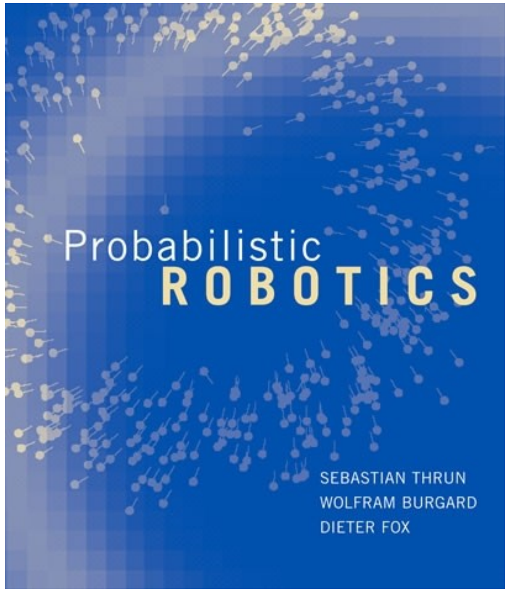
Plan of the day

Today we will consider the specific problem of 2D occupancy mapping using beam sensors, as a prelude to/subcomponent of *full SLAM* (next week!)

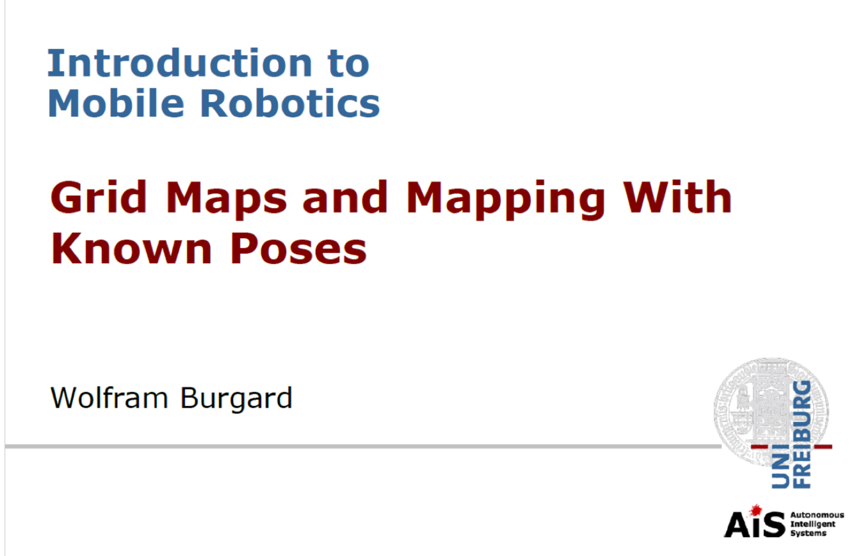


- Probabilistic occupancy grid maps
- Recursive Bayesian estimation for occupancy mapping with known poses
- (Inverse) sensor models for beam sensors

References



Chapter 9



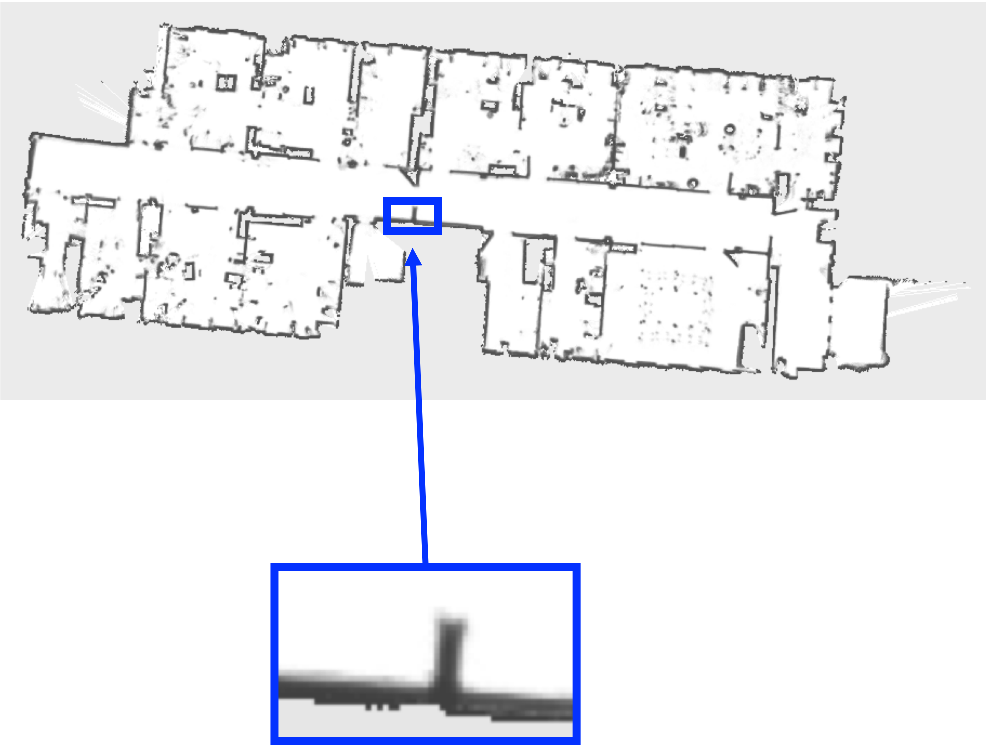
Wolfram Burgard's lecture "Grid Maps and Mapping with Known Poses"

https://www.cs.cmu.edu/~16831-f12/notes/F12/16831_lecture05_vh.pdf

Grid Maps

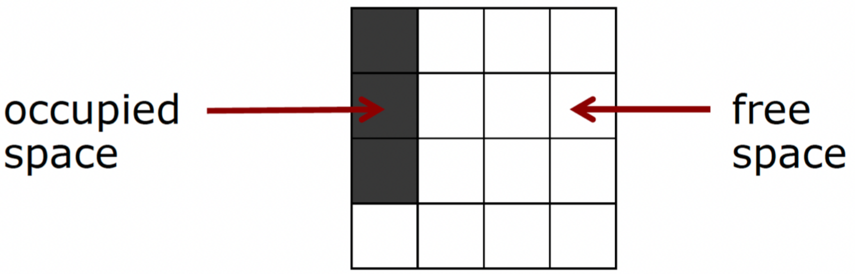
Discretize the world into cells

- The grid structure is rigid
- Each cell: occupied or free
- Large maps require substantial memory resources
- E. g. , 10m by 20m space, 5cm resolution -> 80000 cells -> 2^{80000} possible maps.
- Once constructed, cannot change the resolution
- Useful for combining different sensor scans and modalities
  - Sonar, laser, bump, etc.

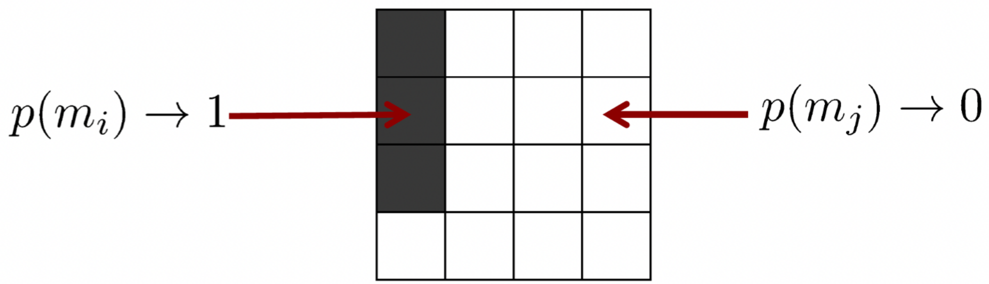


Assumption 1

- The area that corresponds to a cell is either completely free or occupied.



- Each cell is binary random variable that models the occupancy.
- Cell is occupied $p(m_i) = 1$
- Cell is not occupied $p(m_i) = 0$
- No knowledge $p(m_i) = 0.5$
- The environment is assumed to be static.
    - Always free or always occupied



Occupancy probability

Notation

$$p(m_i = occ) = p_{occ}(m_i) = p(m_i)$$

The opposite event:

$$p(m_i = free) = p_{free}(m_i)$$

$$= 1 - p(m_i) = p(\neg m_i)$$

■ $p(m_i = occ) = p(m_i) = 1$

$p(m_i = free) = p(\neg m_i) = 1 - p(m_i)$=0

□ $p(m_i = occ) = p(m_i) = 0$

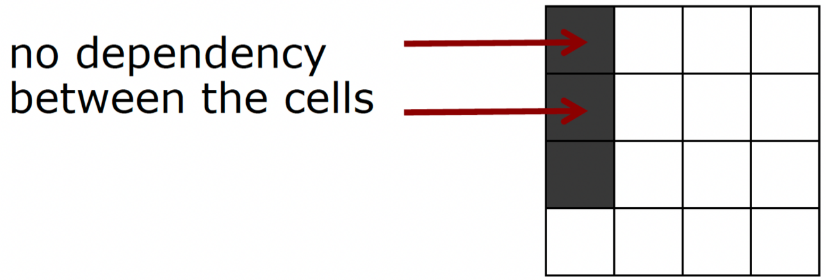
$p(m_i = free) = p(\neg m_i) = 1 - p(m_i)$=1

■ $p(m_i = occ) = p(m_i) = 0.75$

$p(m_i = free) = p(\neg m_i) = 1 - p(m_i) = 0.25$

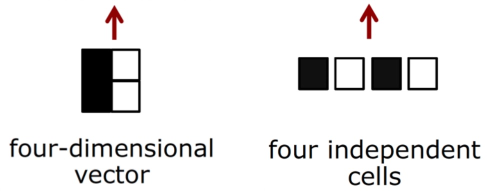
Assumption 2

- The cells (the random variables) are **independent** of each other.



- The probability distribution of the map is given by the product of the probability distributions of the individual cells.

$$p(m) = \prod_i p(m_i)$$



This is a bad assumption to make since obstacles usually span multiple cells. However, this makes the problem tractable.

Occupancy grid maps (OGM)

Key assumptions

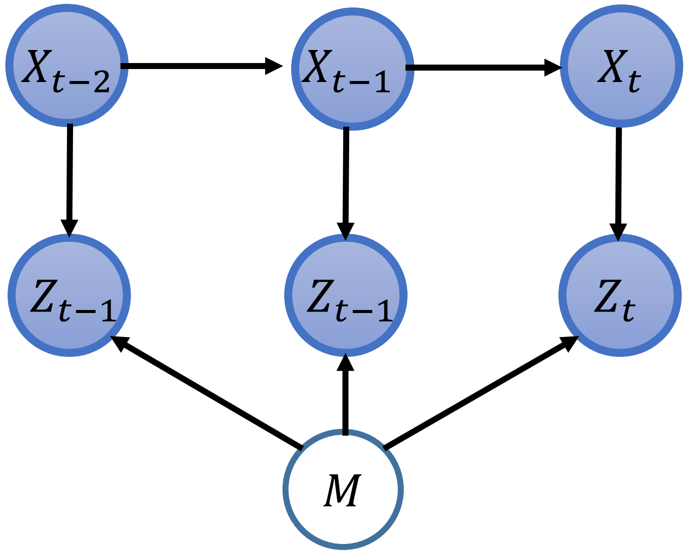
- Environment is **static**.
- Robot positions are **known.**
- Occupancy of individual cells (m_i) are **independent**.

$$p(m|z_{1:t}, x_{1:t}) = \prod_i p(m_i|z_{1:t}, x_{1:t})$$

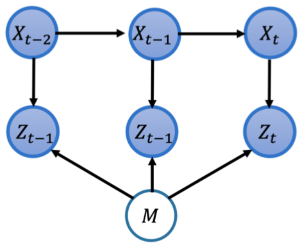
Binary random variable
(Binary Bayes filter)
for a static state – map is not changing

OGM graphical model

- z and x are known.
- **Goal:** infer the map m
- Controls u are not explicitly shown here since x are given (no need for prediction).



Estimating a map from data



Bayes' Rule

$$p(A|B) = \frac{p(B|A)p(A)}{p(B)}$$

Conditional Bayes' Rule

$$p(A|B,C) = \frac{p(B|A,C)p(A|C)}{p(B|C)}$$

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

Bayes rule

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

Markov assumption

$$p(z_t|m_i, x_t) = \frac{p(m_i|z_t, x_t)p(z_t|x_t)}{p(m_i|x_t)}$$

Bayes rule

Inverse measurement model: what is the likelihood of the map cell being occupied given the current state and measurement?

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

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

Independence

Knowing the current state provides no info about whether cell is occupied, if there are no observations

Static State Binary Bayes Filter

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

Do exactly the same for the opposite event:

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

$$\frac{p(m_i|z_{1:t},x_{1:t})}{p(\neg m_i|z_{1:t},x_{1:t})} = \frac{\dfrac{p(m_i|z_t,x_t)p(\cancel{z_t|x_t})p(m_i|z_{1:t-1},x_{1:t-1})}{p(m_i)p(\cancel{z_t|z_{1:t-1},x_{1:t}})}}{\dfrac{p(\neg m_i|z_t,x_t)p(\cancel{z_t|x_t})p(\neg m_i|z_{1:t-1},x_{1:t-1})}{p(\neg m_i)p(\cancel{z_t|z_{1:t-1},x_{1:t}})}} = \frac{p(m_i|z_t,x_t)p(m_i|z_{1:t-1},x_{1:t-1})p(\neg m_i)}{p(\neg m_i|z_t,x_t)p(\neg m_i|z_{1:t-1},x_{1:t-1})p(m_i)}$$

Occupancy update rule

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

$$\frac{p(m_i|z_{1:t},x_{1:t})}{1-p(m_i|z_{1:t},x_{1:t})} = \underbrace{\frac{p(m_i|z_t,x_t)}{1-p(m_i|z_t,x_t)}}_{\text{Uses } z_t} \underbrace{\frac{p(m_i|z_{1:t-1},x_{1:t-1})}{1-p(m_i|z_{1:t-1},x_{1:t-1})}}_{\text{Recursive term}} \underbrace{\frac{1-p(m_i)}{p(m_i)}}_{\text{Prior}}$$

Odds of whether a grid cell is occupied or not

How to recover probability?

From ratio to probability

$$Odds(x) = \frac{p(x)}{1 - p(x)} = Y$$

$$p(x) = Y - Yp(x)$$

$$p(x)(1 + Y) = Y$$

$$p(x) = \frac{Y}{1 + Y}$$

$$p(x) = \frac{1}{1 + \frac{1}{Y}}$$

Occupancy update rule

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

$$\frac{p(m_i|z_{1:t}, x_{1:t})}{1-p(m_i|z_{1:t}, x_{1:t})} = \underbrace{\frac{p(m_i|z_t, x_t)}{1-p(m_i|z_t, x_t)}}_{\text{Uses } z_t} \underbrace{\frac{p(m_i|z_{1:t-1}, x_{1:t-1})}{1-p(m_i|z_{1:t-1}, x_{1:t-1})}}_{\text{Recursive term}} \underbrace{\frac{1-p(m_i)}{p(m_i)}}_{\text{Prior}}$$

$$Bel(m_t^i) = \left[1 + \frac{1-p(m_t^i|z_t, x_t)}{p(m_t^i|z_t, x_t)} \frac{1-Bel(m_{t-1}^i)}{Bel(m_{t-1}^i)} \frac{p(m_t^i)}{1-p(m_t^i)}\right]^{-1}$$

There might be some numerical issues

Log odds notation

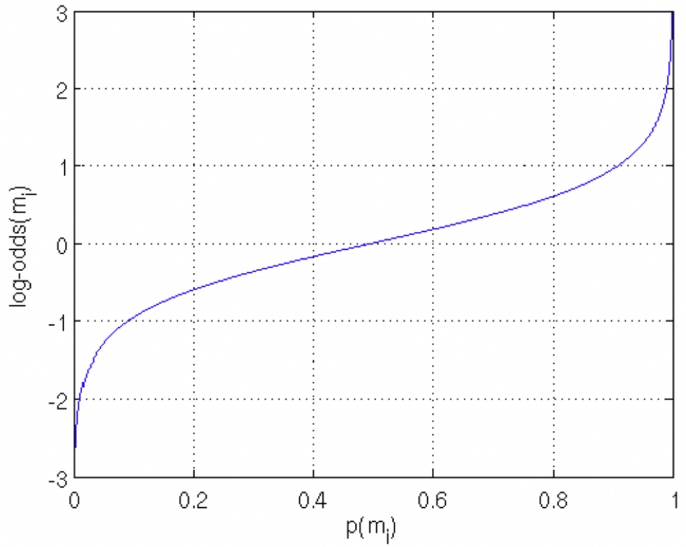
- Log odds ratio is defined as

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

- And with the ability to retrieve $p(x)$

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

- Why?
  - $l(x) \in [-\infty, \infty]$
  - Computationally elegant for updating beliefs in log-odds form because updates are additive and avoids truncation problems arise for probabilities close to 0 and 1.



Occupancy grid in log-odds form

$$\frac{p(m_i|z_{1:t},x_{1:t})}{1-p(m_i|z_{1:t},x_{1:t})} = \underbrace{\frac{p(m_i|z_t,x_t)}{1-p(m_i|z_t,x_t)}}_{\text{Uses } z_t} \underbrace{\frac{p(m_i|z_{1:t-1},x_{1:t})}{1-p(m_i|z_{1:t-1},x_{1:t})}}_{\text{Recursive term}} \underbrace{\frac{1-p(m_i)}{p(m_i)}}_{\text{Prior}}$$

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

$$\log xyz = \log x + \log y + \log z$$

$$l(m_i|z_{1:t},x_{1:t}) = \color{red}{l(m_i|z_t,x_t)} + \color{blue}{l(m_i|z_{1:t-1},x_{1:t})} - \color{green}{l(m_i)}$$

Inverse sensor model Recursive term Prior

$$l_{t,i} = \text{inv\_sensor\_model}(m_i,z_t,x_t) + l_{t-1,i} - l_0$$

Occupancy mapping algorithm

occupancy_grid_mapping$(\{l_{t-1,i}\}, x_t, z_t)$**:**

    for all cells m_i *do*
        if m_i *in perceptual field of* z_t *then*
            $l_{t,i} = l_{t-1,i} + \text{inv\_sensor\_model}(m_i, x_t, z_t) - l_0$
        else
            $l_{t,i} = l_{t-1,i}$
        endif
    endfor
    return $\{l_{t,i}\}$

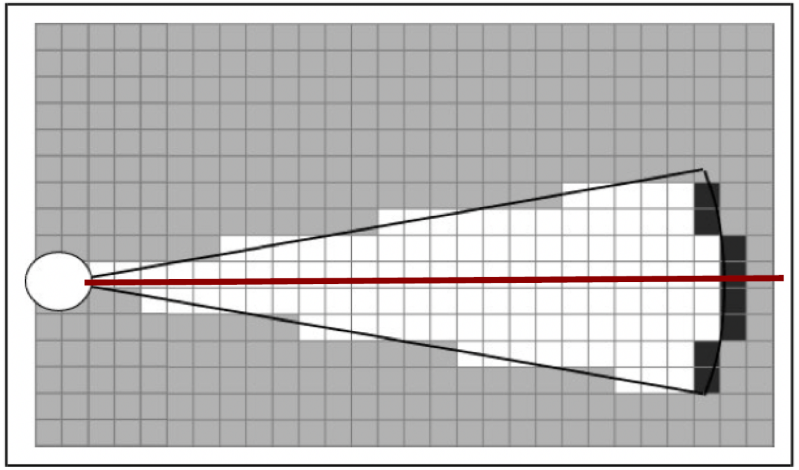
Developed in the mid 80’s by Moravec and Elfes

Inverse sensor model

Sensor (or forward) model: $p(z_t|m_i, x_t)$

Inverse sensor model: $p(m_i|z_t, x_t)$

- specifies a distribution over the (binary) state variable as a function of z_t and x_t.

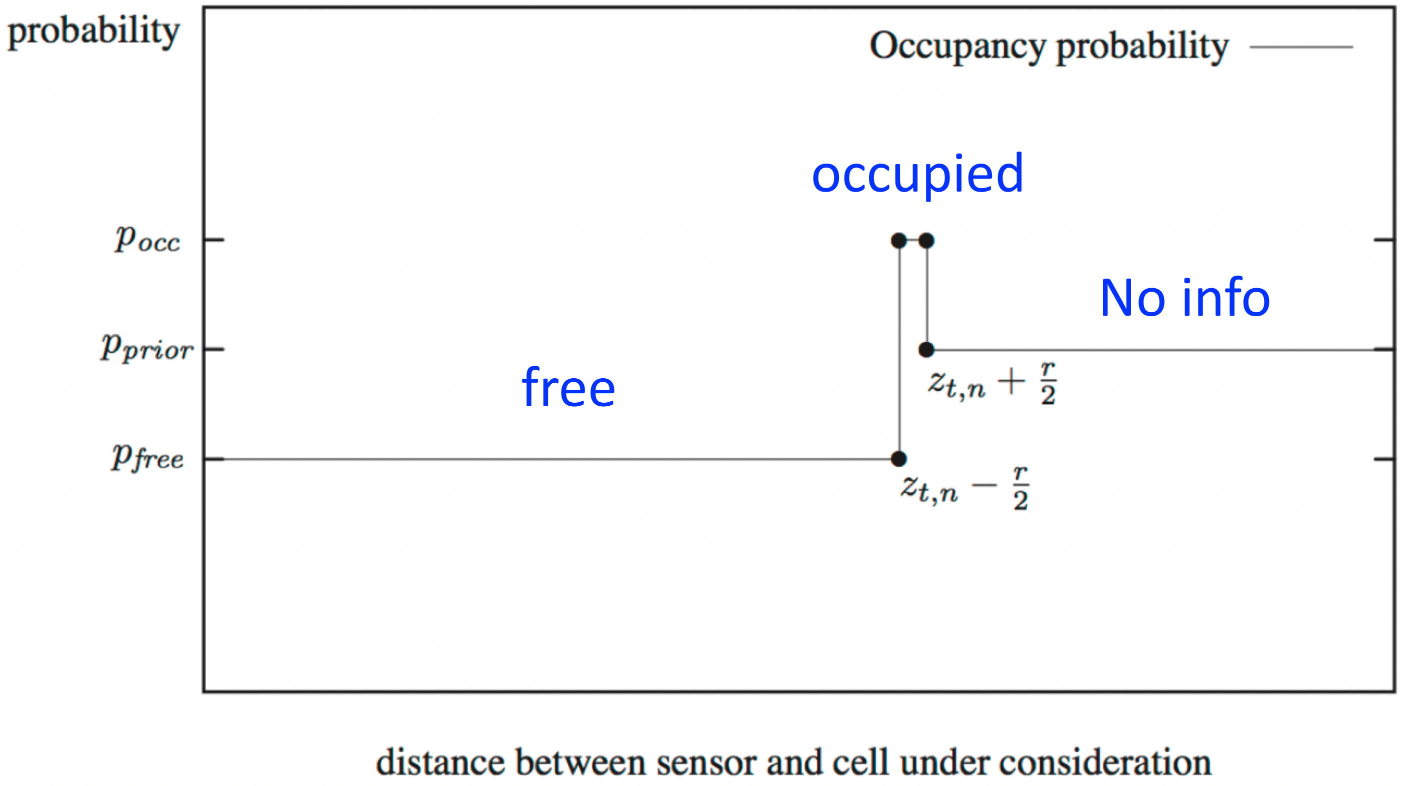


Consider a sensor (e.g., Sonar, laser) which provides the probability a grid cell is occupied given a sensor reading:

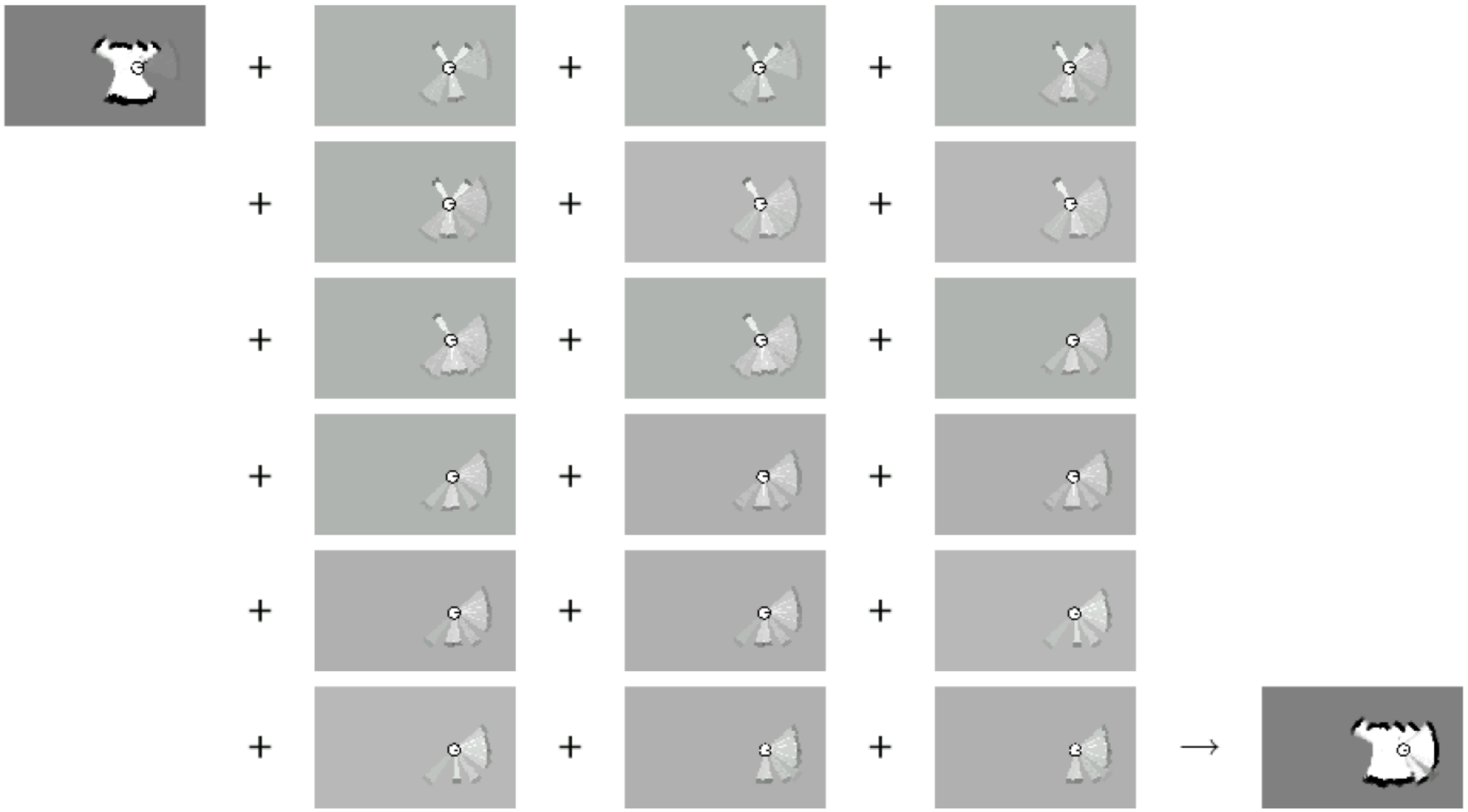
Gray: a probability equal to prior
White: low probability
Black: high probability

Inverse sensor model for laser range finders

Occupancy Value Depending on the Measured Distance



Incremental updating of occupancy grids



Examples

https://www.youtube.com/watch?v=ykOTfrS5pO4

Examples

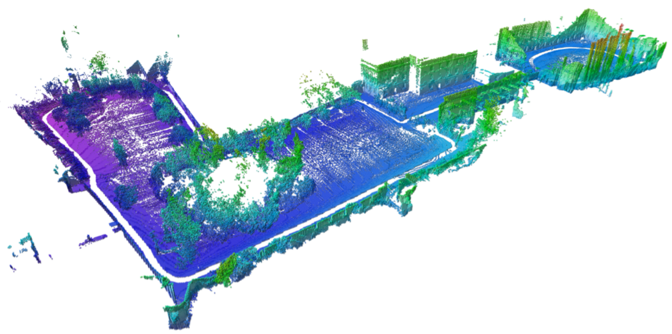
https://www.youtube.com/watch?v=6QgijvWFhFM

Occupancy map summary

- Represent environment of a mobile robot given known poses
- Each cell is considered independently from all others.
- It stores the posterior probability that the corresponding area in the environment is occupied.
- OGM can be learned efficiently using a probabilistic approach.

Limitations: Memory required

- 2D case reasonable
- Dramatic increase in 3D case



3D occupancy mapping
OctoMap

https://link.springer.com/content/pdf/10.1007/s10514-012-9321-0.pdf