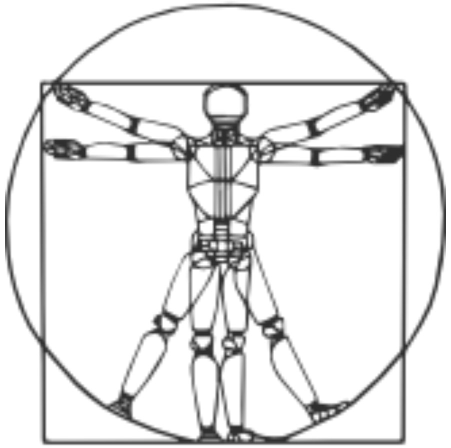
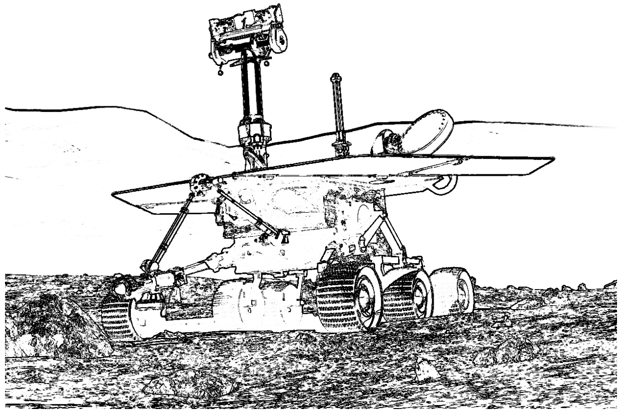




COMP417
Introduction to Robotics and Intelligent Systems

Lecture 7: Occupancy Grid Mapping With Known Poses

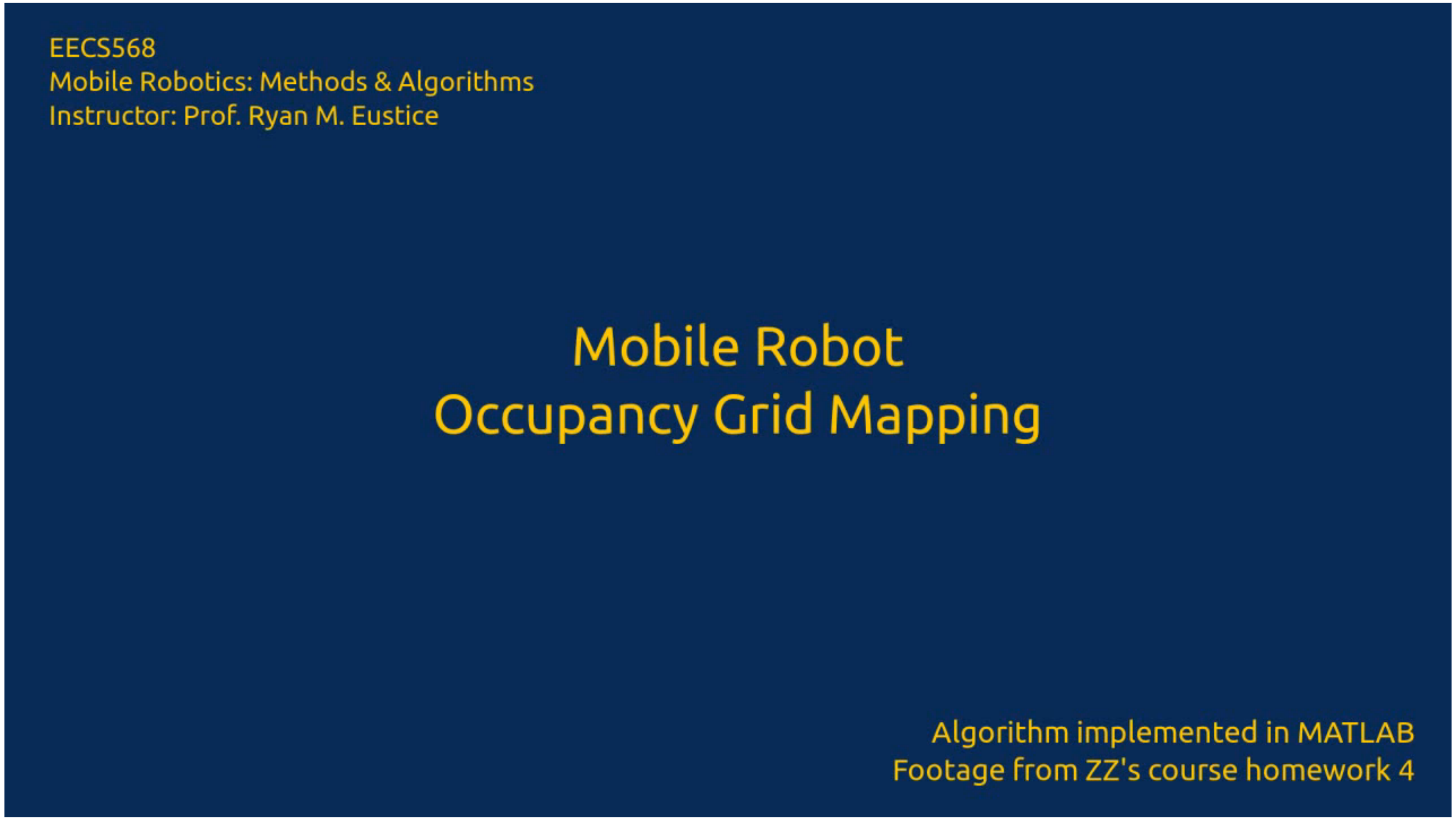
Sept 24th, 2019

Material from Florian Shkurti @ U of T and Probabilistic Robotics Text

McGill

MRL Mobile Robotics Lab at McGill University

What we want to do



Terminology

- Pose: the rotation and translation of a robot, or in general its full state configuration

- Odometry: the transformation of the body frame with respect to its initial pose (fixed frame of reference).

$$ {}^{B_0}_{B_t}T $$

- Dynamics model: what is the next state given current state and control?

$$ x_{t+1} = f(x_t, u_t) $$

- Sensor model: value will be returned on a given sensor channel from a given state?

$$ z_t = h(x_t) $$

Robotic Map Building

- From each pose, a robot senses its immediate surroundings only
- In order to build a full map, it must move to cover the space
- Mapping means accumulating information, z, from the sensor across many states/poses, x. From all x and z, we form m.

Why is mapping a problem?

Don't we have all the information we need to build a map?

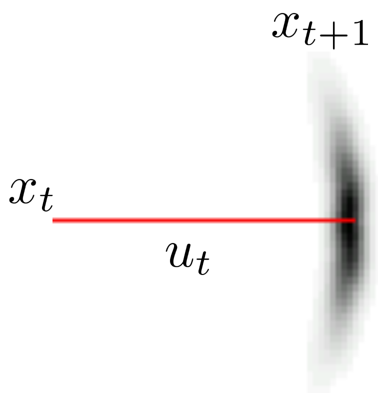
- Well, are humans perfect?
  - https://www.nationalgeographic.com/news/2016/10/northwest-passage-map-history/#/01-northwest-passage-maps.JPG

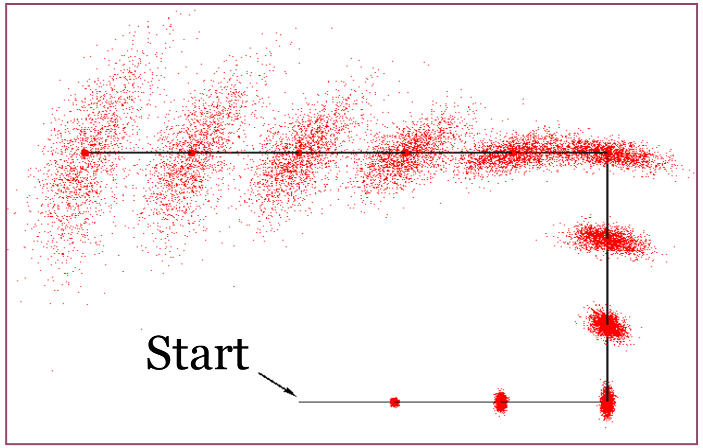
Why is mapping a problem?

Don't we have all the information we need to build a map?

- Two main sources of uncertainty in robotic mapping:
    - accumulating uncertainty in the dynamics

$$p(x_{t+1}|x_t, u_t)$$



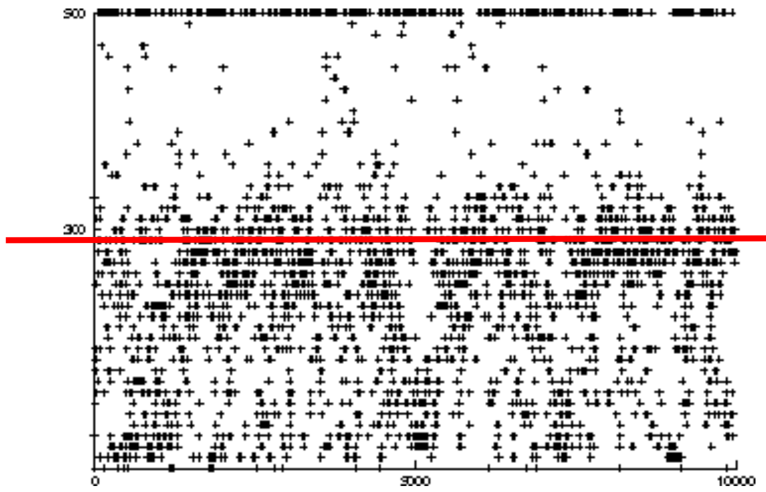


Uncertainty in the dynamics compounds into increasing uncertainty in odometry, as time passes.

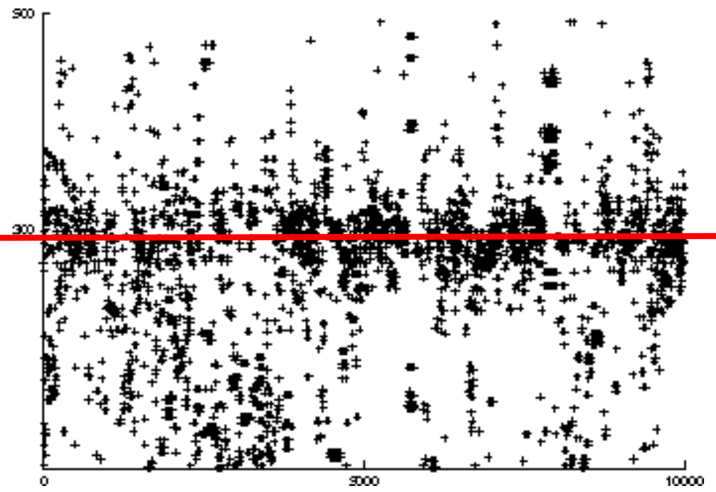
Why is mapping a problem?

Don't we have all the information we need to build a map?

- Two main sources of uncertainty:
  - uncertainty in the dynamics $p(x_{t+1}|x_t, u_t)$
  - uncertainty in sensor measurements



Sonar



Laser

True value: 300cm

but measurements have noise

Why is mapping a problem?

Don't we have all the information we need to build a map?

- If we had no uncertainty, i.e. $x_{t+1} = f(x_t, u_t)$ and $z_t = h(x_t)$ perfectly described reality, then mapping would be trivial.

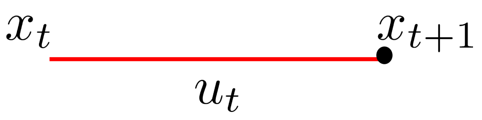
- Today we will assume perfect dynamics and odometry, but noisy sensor measurements: $p(z_t | x_t)$

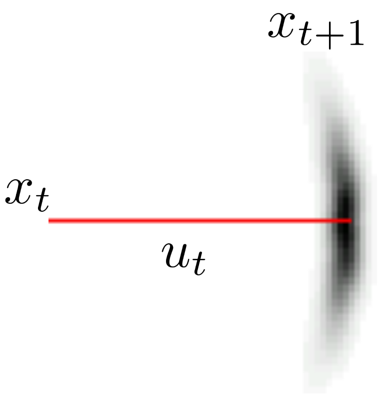
- We are also going to assume a static map, no moving objects

Perfect models vs. Reality

Dynamics

$x_{t+1} = f(x_t, u_t)$





Noise as a random variable

$x_{t+1} = f(x_t, u_t) + w_t$

Idealized dynamics / odometry tells us that the physical reality should perfectly match what we command and sense.

Dynamics / odometry noise will capture:

- Imperfect calibration (e.g., we think the wheel is 20 cm radius, but it wore down to be 19.8 cm – we move less on each turn than expected!)
- Wheels slipping on the ground. A bit of error each time it happens - we can't predict when or by how much.
- Drive belts stretching, gears slipping, etc.
- And many more!

Perfect models vs. Reality

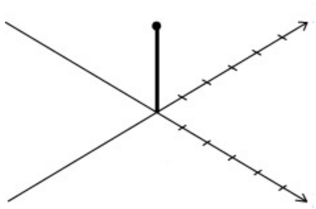
Idealized sensing says the range reading we receive is precisely the right distance in the physical world. “Trust” the sensor.

Sensor Measurements

$z_t = h(x_t)$

$z_t = x_t$

e.g. GPS (simplified)

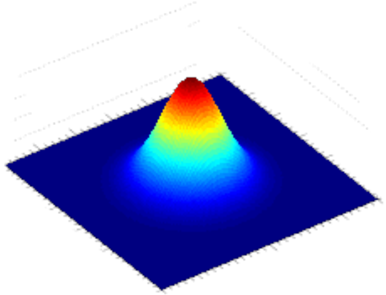


Sensing noise will capture:

- Aspects of the sensor’s physics we do not model (some sensors read a higher value when the heat up – space looks bigger in summer than in winter!)
- Complicated aspects of the environment (reflection/absorption by materials is a big one):
  - A lidar hitting a mirror, then a wall, then back to us
  - GPS bouncing off buildings

Noise as a random variable

$$z_t = x_t + v_t, \quad v_t \sim \mathcal{N}(0, \sigma^2 I)$$

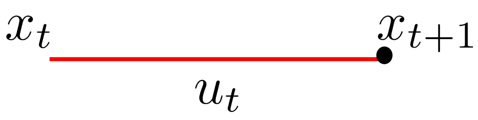


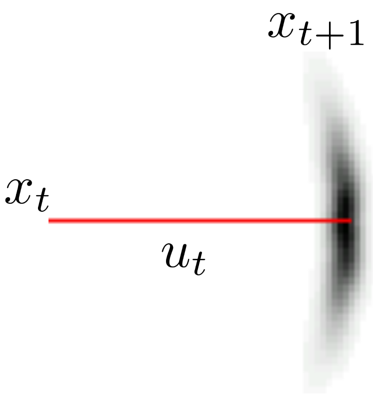
w and v do not necessarily follow the same distribution

Perfect models vs. Reality

$x_{t+1} = f(x_t, u_t)$

Dynamics





$$p(x_{t+1} | x_t, u_t)$$

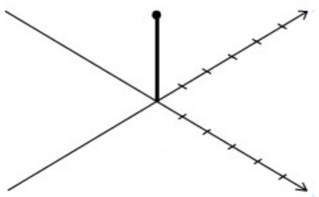
probabilistic
dynamics model

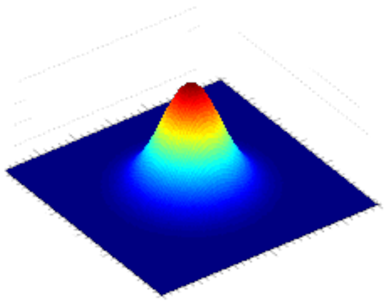
Sensor
Measurements

$z_t = h(x_t)$

e.g. GPS (simplified)

$z_t = x_t$





$$p(z_t | x_t)$$

probabilistic
measurement model

Very brief probability check-up

- $p(A)$ for random variable A means the chance it occurs:
  - A – the grass is wet
  - Outcomes: yes or no – binary random variable (don't assume the level of wetness matters here)
  - We can estimate $p(A)$ by counting: every morning visit your yard, touch the grass, tally wet or not:
    - $p(A = yes)$ = #wet_days / #days_total
    - $p(A = no)$ = #dry_days / #days_total

- $p(A, B)$ captures more detailed outcomes:
  - B – it rained last night
  - We now want to know $p(A = yes, B = yes)$ how likely that the grass is wet AND it rained
  - Also $p(A = no, B = no)$, and all other combinations

Very brief probability check-up

- $p(A|B)$ for random variables:
  - A – the grass is wet
  - B – it rained last night

- Each of A and B are binary random variables:
  - 1 = the event occurred (rain or wetness)
  - 0 = the event did not (sunny or dry)

- Our probability formula reflects a one-way relationship:
  - Last night's weather changes the chances of wet grass today, but it is not a simple function.
  - It can rain and then wind dries the grass. It can be sunny but our neighbor spills their coffee.

Staring at the odometry probability

- The probability of being at a next state give a previous state and control.
  - The mean is very often $x_{t+1} = f(x_t, u_t)$
  - To pick the variance and/or shape, we need information from the physics and hardware

$$p(x_{t+1}|x_t, u_t)$$

- Note that it's crucial to condition on the previous state and control: we don't warp to a location but end up there due to a series of motions in time.

Staring at the sensing probability

- The probability of our sensor returning a value z, given that we are at state x
  - The mean is very often $z_t = h(x_t)$
  - To pick the variance and/or shape, we need information from the physics and hardware

$$p(z_t|x_t)$$

- Note that it's crucial to condition on the previous state. This controls the physical circumstances, and therefore our sensor's chance to return any given value.
  - Note that the sensor return is also very influenced by the environment (map). We will expand to show this in the relationship soon.

Defining the problem

- The occupancy grid map is a binary random variable

$$\mathbf{m} = \{m_{ij}\} \in \{0, 1\}^{W \times H}$$

width = #columns
height = #rows
of the occupancy grid

Defining the problem

- The occupancy grid map is a binary random variable

$$\mathbf{m} = \{m_{ij}\} \in \{0, 1\}^{W \times H}$$

width = #columns
height = #rows
of the occupancy grid

- The path of the robot up to time t is a sequence of random variables $\mathbf{x}_{1:t} = \mathbf{x}_1, \ldots, \mathbf{x}_t$ with $\mathbf{x}_i = (x_i, y_i, \theta_i)$

Odometry coordinates

Defining the problem

- The occupancy grid map is a binary random variable

$$\mathbf{m} = \{m_{ij}\} \in \{0, 1\}^{W \times H}$$

width = #columns
height = #rows
of the occupancy grid

- The path of the robot up to time t is a sequence of random variables $\mathbf{x}_{1:t} = \mathbf{x}_1, \ldots, \mathbf{x}_t$ with $\mathbf{x}_i = (x_i, y_i, \theta_i)$

Odometry coordinates

- At each time step the robot makes a measurement (sonar/laser). Measurements up to time t are a sequence of random variables $\mathbf{z}_{1:t} = \mathbf{z}_1, \ldots, \mathbf{z}_t$ with $\mathbf{z}_i = \{(r_i, \psi_i)\}^K$

K = #beams, or
#points in the scan

(range, angle) in
the laser's local
coordinates

The goal of mapping

- To estimate the probability of any map, given path and measurements $p(\mathbf{m}|\mathbf{z}_{1:t}, \mathbf{x}_{1:t})$?

The goal of mapping

- To estimate the probability of any map, given path and measurements $p(\mathbf{m}|\mathbf{z}_{1:t}, \mathbf{x}_{1:t})$?
- This is intractable. E.g. for a 100 x 100 grid there are 2^{10000} possible binary maps.

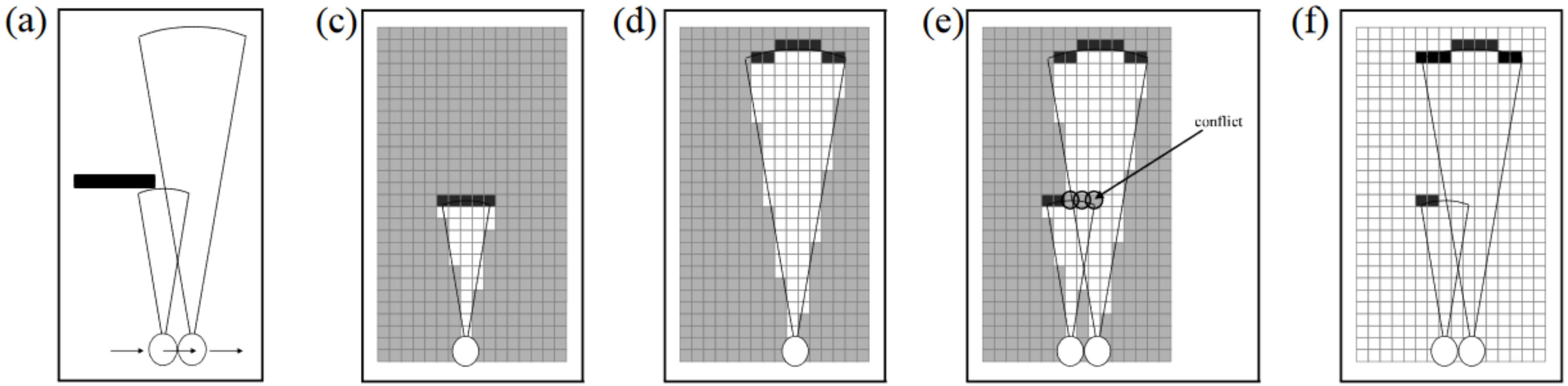
The goal of mapping

- To estimate the probability of any map, given path and measurements $p(\mathbf{m}|\mathbf{z}_{1:t}, \mathbf{x}_{1:t})$?
- This is intractable. E.g. for a 100 x 100 grid there are 2^{10000} possible binary maps.

- We can approximate $p(\mathbf{m}|\mathbf{z}_{1:t}, \mathbf{x}_{1:t}) \simeq \prod_{i,j} p(m_{ij}|\mathbf{z}_{1:t}, \mathbf{x}_{1:t})$

Approximation ignores all dependencies between map cells, given known info. Assumes (for tractability) that cells are independent given path and measurements

Why is it an approximation?



Scenario

Nearby measurements

Resulting map when considering cells independently

Resulting map when considering cells jointly

Evaluating the occupancy of a map cell

- How do we evaluate $p(m_{ij} = 1|\mathbf{z}_{1:t}, \mathbf{x}_{1:t})$?

Evaluating the occupancy of a map cell

- How do we evaluate $p(m_{ij} = 1 | \mathbf{z}_{1:t}, \mathbf{x}_{1:t})$?

Bayes' Rule

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

Conditional Bayes' Rule

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

Evaluating the occupancy of a map cell

- How do we evaluate $p(m_{ij} = 1|\mathbf{z}_{1:t}, \mathbf{x}_{1:t})$?
- Using conditional Bayes' rule we get

$$p(m_{ij} = 1|\mathbf{z}_{1:t}, \mathbf{x}_{1:t}) = \frac{p(\mathbf{z}_t|\mathbf{z}_{1:t-1}, \mathbf{x}_{1:t}, m_{ij} = 1)p(m_{ij} = 1|\mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})}{p(\mathbf{z}_t|\mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})}$$

Bayes' Rule

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

Conditional Bayes' Rule

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

Evaluating the occupancy of a map cell

- How do we evaluate $p(m_{ij} = 1 | \mathbf{z}_{1:t}, \mathbf{x}_{1:t})$?
- Using conditional Bayes' rule we get

$$p(m_{ij} = 1 | \mathbf{z}_{1:t}, \mathbf{x}_{1:t}) = \frac{p(\mathbf{z}_t | \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t}, m_{ij} = 1) p(m_{ij} = 1 | \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})}{p(\mathbf{z}_t | \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})}$$

- And we simplify

$$p(m_{ij} = 1 | \mathbf{z}_{1:t}, \mathbf{x}_{1:t}) = \frac{p(\mathbf{z}_t | \mathbf{x}_t, m_{ij} = 1) p(m_{ij} = 1 | \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-1})}{p(\mathbf{z}_t | \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})}$$

If C is independent of A given B, then C provides no extra information about A after we know B.

$$p(A|B, C) = p(A|B)$$

"Conditional Independence"

Evaluating the occupancy of a map cell

- How do we evaluate $p(m_{ij} = 1|\mathbf{z}_{1:t}, \mathbf{x}_{1:t})$?
- Using conditional Bayes' rule we get

$$p(m_{ij} = 1|\mathbf{z}_{1:t}, \mathbf{x}_{1:t}) = \frac{p(\mathbf{z}_t|\mathbf{z}_{1:t-1}, \mathbf{x}_{1:t}, m_{ij} = 1)p(m_{ij} = 1|\mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})}{p(\mathbf{z}_t|\mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})}$$

Current measurement only depends on current state and map cell

Current state without current measurement provides no additional information about the occupancy of the map cell

- And we simplify

$$p(m_{ij} = 1|\mathbf{z}_{1:t}, \mathbf{x}_{1:t}) = \frac{p(\mathbf{z}_t|\mathbf{x}_t, m_{ij} = 1)p(m_{ij} = 1|\mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-1})}{p(\mathbf{z}_t|\mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})}$$

Evaluating the occupancy of a map cell

- And we simplify:

$$p(m_{ij} = 1|\mathbf{z}_{1:t}, \mathbf{x}_{1:t}) = \frac{p(\mathbf{z}_t|\mathbf{x}_t, m_{ij} = 1)p(m_{ij} = 1|\mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-1})}{p(\mathbf{z}_t|\mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})}$$

- Another way to write this:

$$belief_t(m_{ij} = 1) \ = \eta \, p(\mathbf{z}_t|\mathbf{x}_t, m_{ij} = 1) \, belief_{t-1}(m_{ij} = 1)$$

- Belief at time t-1 was updated to belief at time t based on likelihood of measurement received at time t.

Evaluating the occupancy of a map cell

- And we simplify:

$$p(m_{ij} = 1|\mathbf{z}_{1:t}, \mathbf{x}_{1:t}) = \frac{p(\mathbf{z}_t|\mathbf{x}_t, m_{ij} = 1)p(m_{ij} = 1|\mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-1})}{p(\mathbf{z}_t|\mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})}$$

- Another way to write this:

$$belief_t(m_{ij} = 1) \;\; = \eta \; p(\mathbf{z}_t|\mathbf{x}_t, m_{ij} = 1) \; belief_{t-1}(m_{ij} = 1)$$

So, as long as we can evaluate the measurement likelihood

$$p(\mathbf{z}_t|\mathbf{x}_t, m_{ij} = 1)$$

and the normalization factor

$$\eta = 1/p(\mathbf{z}_t|\mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})$$

we can do the belief update.

Evaluating the occupancy of a map cell

- And we simplify:

$$p(m_{ij} = 1|\mathbf{z}_{1:t}, \mathbf{x}_{1:t}) = \frac{p(\mathbf{z}_t|\mathbf{x}_t, m_{ij} = 1)p(m_{ij} = 1|\mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-1})}{p(\mathbf{z}_t|\mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})}$$

- Another way to write this:

$$belief_t(m_{ij} = 1) \; = \eta \; p(\mathbf{z}_t|\mathbf{x}_t, m_{ij} = 1) \; belief_{t-1}(m_{ij} = 1)$$

So, as long as we can evaluate the measurement likelihood

$$p(\mathbf{z}_t|\mathbf{x}_t, m_{ij} = 1)$$

and the normalization factor

$$\eta = 1/p(\mathbf{z}_t|\mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})$$

we can do the belief update.

Problem: this is hard to compute. How can we avoid it?

The log-odds trick for binary random variables

- We showed $belief_t(m_{ij} = 1) \; = \eta \; p(\mathbf{z}_t | \mathbf{x}_t, m_{ij} = 1) \; belief_{t-1}(m_{ij} = 1)$

- Define the log odds ratio $l_t^{(ij)} = \log \dfrac{p(m_{ij} = 1 | \mathbf{z}_{1:t}, \mathbf{x}_{1:t})}{p(m_{ij} = 0 | \mathbf{z}_{1:t}, \mathbf{x}_{1:t})} = \log \dfrac{belief_t(m_{ij} = 1)}{belief_t(m_{ij} = 0)}$

The log-odds trick for binary random variables

- We showed $belief_t(m_{ij} = 1) \;= \eta\, p(\mathbf{z}_t|\mathbf{x}_t, m_{ij} = 1)\; belief_{t-1}(m_{ij} = 1)$ (1)

- Define the log odds ratio $l_t^{(ij)} = \log \dfrac{p(m_{ij} = 1|\mathbf{z}_{1:t}, \mathbf{x}_{1:t})}{p(m_{ij} = 0|\mathbf{z}_{1:t}, \mathbf{x}_{1:t})} = \log \dfrac{belief_t(m_{ij} = 1)}{belief_t(m_{ij} = 0)}$

- Then (1) becomes $l_t^{(ij)} = \log \dfrac{p(\mathbf{z}_t|\mathbf{x}_t, m_{ij} = 1)}{p(\mathbf{z}_t|\mathbf{x}_t, m_{ij} = 0)} + l_{t-1}^{(ij)}$

The log-odds trick for binary random variables

- We showed $belief_t(m_{ij} = 1) \ = \eta \, p(\mathbf{z}_t | \mathbf{x}_t, m_{ij} = 1) \, belief_{t-1}(m_{ij} = 1)$ (1)

- Define the log odds ratio $l_t^{(ij)} = \log \dfrac{p(m_{ij} = 1 | \mathbf{z}_{1:t}, \mathbf{x}_{1:t})}{p(m_{ij} = 0 | \mathbf{z}_{1:t}, \mathbf{x}_{1:t})} = \log \dfrac{belief_t(m_{ij} = 1)}{belief_t(m_{ij} = 0)}$

- Then (1) becomes $l_t^{(ij)} = \log \dfrac{p(\mathbf{z}_t | \mathbf{x}_t, m_{ij} = 1)}{p(\mathbf{z}_t | \mathbf{x}_t, m_{ij} = 0)} + l_{t-1}^{(ij)}$

- We can recover the original belief as

$$belief_t(m_{ij} = 1) = 1 - \frac{1}{1 + \exp(l_t^{(ij)})}$$

The log-odds trick for binary random variables

- We showed $$belief_t(m_{ij}=1) = \eta\, p(\mathbf{z}_t|\mathbf{x}_t, m_{ij}=1)\, belief_{t-1}(m_{ij}=1) \qquad (1)$$
- Define the log odds ratio $$l_t^{(ij)} = \log \frac{p(m_{ij}=1|\mathbf{z}_{1:t}, \mathbf{x}_{1:t})}{p(m_{ij}=0|\mathbf{z}_{1:t}, \mathbf{x}_{1:t})} = \log \frac{belief_t(m_{ij}=1)}{belief_t(m_{ij}=0)}$$
- Then (1) becomes $$l_t^{(ij)} = \log \frac{p(\mathbf{z}_t|\mathbf{x}_t, m_{ij}=1)}{p(\mathbf{z}_t|\mathbf{x}_t, m_{ij}=0)} + l_{t-1}^{(ij)}$$

So, as long as we can evaluate the log odds ratio for the measurement likelihood:

$$\log \frac{p(\mathbf{z}_t|\mathbf{x}_t, m_{ij}=1)}{p(\mathbf{z}_t|\mathbf{x}_t, m_{ij}=0)}$$

we can do the belief update.

Log-odds ratio for the measurement likelihood

- We want to compute $\log\frac{p(\mathbf{z}_t|\mathbf{x}_t, m_{ij}=1)}{p(\mathbf{z}_t|\mathbf{x}_t, m_{ij}=0)}$ to do the belief update

- We apply conditional Bayes' rule again: $p(\mathbf{z}_t|\mathbf{x}_t, m_{ij}=1) = \frac{p(m_{ij}=1|\mathbf{z}_t, \mathbf{x}_t)\ p(\mathbf{z}_t|\mathbf{x}_t)}{p(m_{ij}=1|\mathbf{x}_t)}$

Log-odds ratio for the measurement likelihood

- We want to compute $\log \frac{p(\mathbf{z}_t|\mathbf{x}_t, m_{ij}=1)}{p(\mathbf{z}_t|\mathbf{x}_t, m_{ij}=0)}$

- We apply conditional Bayes' rule again: $p(\mathbf{z}_t|\mathbf{x}_t, m_{ij}=1) = \frac{p(m_{ij}=1|\mathbf{z}_t, \mathbf{x}_t)\, p(\mathbf{z}_t|\mathbf{x}_t)}{p(m_{ij}=1|\mathbf{x}_t)}$

- If we take the log-odds ratio: $\log \frac{p(\mathbf{z}_t|\mathbf{x}_t, m_{ij}=1)}{p(\mathbf{z}_t|\mathbf{x}_t, m_{ij}=0)} = \log \frac{p(m_{ij}=1|\mathbf{z}_t, \mathbf{x}_t)}{p(m_{ij}=0|\mathbf{z}_t, \mathbf{x}_t)} + \log \frac{p(m_{ij}=0|\mathbf{x}_t)}{p(m_{ij}=1|\mathbf{x}_t)}$

- We can simplify further:

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

$$\log \frac{p(\mathbf{z}_t|\mathbf{x}_t, m_{ij}=1)}{p(\mathbf{z}_t|\mathbf{x}_t, m_{ij}=0)} = \log \frac{p(m_{ij}=1|\mathbf{z}_t, \mathbf{x}_t)}{p(m_{ij}=0|\mathbf{z}_t, \mathbf{x}_t)} + \log \frac{p(m_{ij}=0)}{p(m_{ij}=1)}$$

Log-odds ratio for the measurement likelihood

- We want to compute $\log \frac{p(\mathbf{z}_t|\mathbf{x}_t, m_{ij} = 1)}{p(\mathbf{z}_t|\mathbf{x}_t, m_{ij} = 0)}$

- We apply conditional Bayes' rule again: $p(\mathbf{z}_t|\mathbf{x}_t, m_{ij} = 1) = \frac{p(m_{ij} = 1|\mathbf{z}_t, \mathbf{x}_t)\, p(\mathbf{z}_t|\mathbf{x}_t)}{p(m_{ij} = 1|\mathbf{x}_t)}$

- If we take the log-odds ratio: $\log \frac{p(\mathbf{z}_t|\mathbf{x}_t, m_{ij} = 1)}{p(\mathbf{z}_t|\mathbf{x}_t, m_{ij} = 0)} = \log \frac{p(m_{ij} = 1|\mathbf{z}_t, \mathbf{x}_t)}{p(m_{ij} = 0|\mathbf{z}_t, \mathbf{x}_t)} + \log \frac{p(m_{ij} = 0|\mathbf{x}_t)}{p(m_{ij} = 1|\mathbf{x}_t)}$

- We can simplify further:

$$\log \frac{p(\mathbf{z}_t|\mathbf{x}_t, m_{ij} = 1)}{p(\mathbf{z}_t|\mathbf{x}_t, m_{ij} = 0)} = \log \frac{p(m_{ij} = 1|\mathbf{z}_t, \mathbf{x}_t)}{p(m_{ij} = 0|\mathbf{z}_t, \mathbf{x}_t)} + \log \frac{p(m_{ij} = 0)}{p(m_{ij} = 1)}$$

Prior probability of cell being occupied.
Can choose uniform distribution, for example.

Log-odds ratio for the measurement likelihood

- We want to compute $\log \frac{p(\mathbf{z}_t|\mathbf{x}_t, m_{ij}=1)}{p(\mathbf{z}_t|\mathbf{x}_t, m_{ij}=0)}$

- We apply conditional Bayes' rule again: $p(\mathbf{z}_t|\mathbf{x}_t, m_{ij}=1) = \frac{p(m_{ij}=1|\mathbf{z}_t,\mathbf{x}_t)\, p(\mathbf{z}_t|\mathbf{x}_t)}{p(m_{ij}=1|\mathbf{x}_t)}$

- If we take the log-odds ratio: $\log \frac{p(\mathbf{z}_t|\mathbf{x}_t, m_{ij}=1)}{p(\mathbf{z}_t|\mathbf{x}_t, m_{ij}=0)} = \log \frac{p(m_{ij}=1|\mathbf{z}_t,\mathbf{x}_t)}{p(m_{ij}=0|\mathbf{z}_t,\mathbf{x}_t)} + \log \frac{p(m_{ij}=0|\mathbf{x}_t)}{p(m_{ij}=1|\mathbf{x}_t)}$

- We can simplify further:

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

$$\log \frac{p(\mathbf{z}_t|\mathbf{x}_t, m_{ij}=1)}{p(\mathbf{z}_t|\mathbf{x}_t, m_{ij}=0)} = \log \frac{p(m_{ij}=1|\mathbf{z}_t,\mathbf{x}_t)}{p(m_{ij}=0|\mathbf{z}_t,\mathbf{x}_t)} + \log \frac{p(m_{ij}=0)}{p(m_{ij}=1)}$$

Log-odds ratio for the measurement likelihood

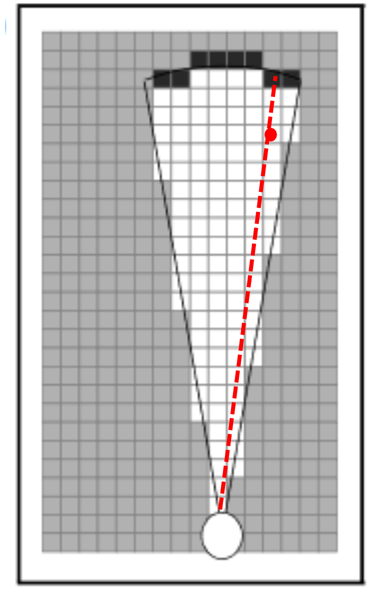
- We want to compute $\log \frac{p(\mathbf{z}_t | \mathbf{x}_t, m_{ij} = 1)}{p(\mathbf{z}_t | \mathbf{x}_t, m_{ij} = 0)}$ but it's hard

- Instead, we can compute the log-odds ratio of the measurement likelihood in terms of the inverse measurement model:

$$\log \frac{p(\mathbf{z}_t | \mathbf{x}_t, m_{ij} = 1)}{p(\mathbf{z}_t | \mathbf{x}_t, m_{ij} = 0)} = \log \frac{p(m_{ij} = 1 | \mathbf{z}_t, \mathbf{x}_t)}{p(m_{ij} = 0 | \mathbf{z}_t, \mathbf{x}_t)} + \log \frac{p(m_{ij} = 0)}{p(m_{ij} = 1)}$$

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

Inverse sensor measurement model

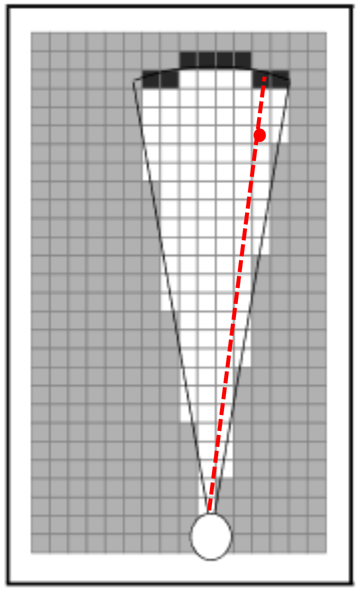


$p(m_{ij} = 1 | \mathbf{z}_t, \mathbf{x}_t)$

Given map cell (i, j), the robot's state $\mathbf{x} = (x, y, \theta)$, and beams $\mathbf{z} = \{(r_k, \psi_k)\}$

Find index k of sensor beam that is closest in heading to the cell (i, j)

Inverse sensor measurement model



$p(m_{ij} = 1|\mathbf{z}_t, \mathbf{x}_t)$

Given map cell (i, j), the robot's state $\mathbf{x} = (x, y, \theta)$, and beams $\mathbf{z} = \{(r_k, \psi_k)\}$

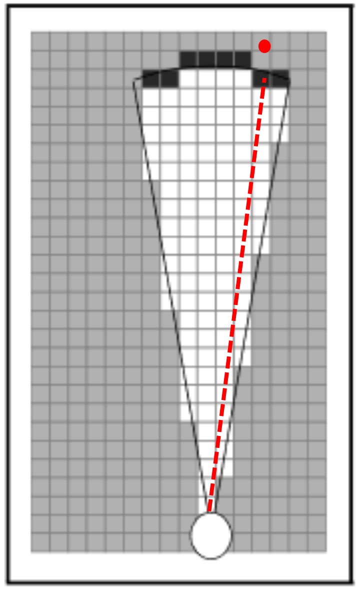
Find index k of sensor beam that is closest in heading to the cell (i, j)

If the cell (i, j) is sufficiently closer than r_k

// Cell is most likely free

Return p_{occupied} that is well below 0.5

Inverse sensor measurement model



$p(m_{ij} = 1|\mathbf{z}_t, \mathbf{x}_t)$

Given map cell (i, j), the robot's state $\mathbf{x} = (x, y, \theta)$, and beams $\mathbf{z} = \{(r_k, \psi_k)\}$

Find index k of sensor beam that is closest in heading to the cell (i, j)

If the cell (i, j) is sufficiently farther than r_k or out of the field of view

// We don't have enough information to decide whether cell is occupied

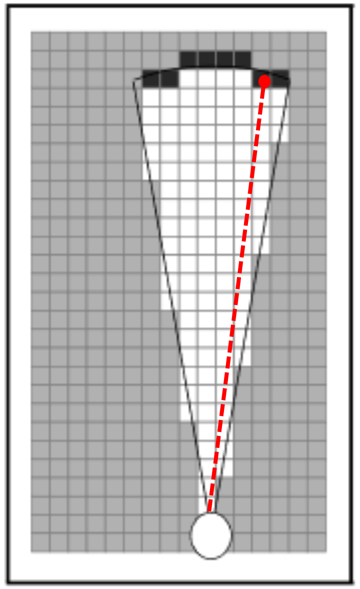
Return prior occupation probability $p(m_{ij} = 1)$

If the cell (i, j) is sufficiently closer than r_k

// Cell is most likely free

Return p_{occupied} that is well below 0.5

Inverse sensor measurement model



$p(m_{ij} = 1|\mathbf{z}_t, \mathbf{x}_t)$

Given map cell (i, j), the robot's state $\mathbf{x} = (x, y, \theta)$, and beams $\mathbf{z} = \{(r_k, \psi_k)\}$

Find index k of sensor beam that is closest in heading to the cell (i, j)

If the cell (i, j) is sufficiently farther than r_k or out of the field of view

// We don't have enough information to decide whether cell is occupied
Return prior occupation probability $p(m_{ij} = 1)$

If the cell (i, j) is nearly as far as the measurement r_k

// Cell is most likely occupied
Return p_{occupied} that is well above 0.5

If the cell (i, j) is sufficiently closer than r_k

// Cell is most likely free
Return p_{occupied} that is well below 0.5

inverse_sensor_measurement_model((i, j), $\mathbf{x} = (x, y, \theta)$, $\mathbf{z} = \{(r_k, \psi_k)\}$)

From Probabilistic Robotics, chapter 9.2

- Let (x_i, y_i) be the center of the cell (i, j)
- Let $r = ||(x_i, y_i) - (x, y)||$
- Let $\phi = \text{atan2}(y_i - y, x_i - x) - \theta$ // Might need to ensure this angle difference is in $[-\pi, \pi]$
- The index of the closest-in-heading beam to (x_i, y_i) is $k^* = \operatorname*{argmin}_k |\phi - \psi_k|$
- If $r > \min\{r_{\max},\ r_k + \alpha/2\}$ or $|\phi - \psi_k| > \beta/2$
  - Return the log odds ratio of the prior occupancy $\log \dfrac{p(m_{ij} = 1)}{p(m_{ij} = 0)}$
- If $r_k < r_{\max}$ and $|r - r_k| < \alpha/2$
  - Return the log odds ratio of being occupied (corresponding to occupation probability > 0.5)
- If $r \leq r_k$
  - Return the log odds ratio of being free (corresponding to occupation probability < 0.5)

Recap

- We wanted to compute the likelihood of any map based on known states and observations

$$p(\mathbf{m}|\mathbf{z}_{1:t}, \mathbf{x}_{1:t}) \quad \simeq \quad \prod_{i,j} p(m_{ij}|\mathbf{z}_{1:t}, \mathbf{x}_{1:t})$$

Recap

- We wanted to compute the likelihood of any map based on known path and observations

$$p(\mathbf{m}|\mathbf{z}_{1:t}, \mathbf{x}_{1:t}) \simeq \prod_{i,j} p(m_{ij}|\mathbf{z}_{1:t}, \mathbf{x}_{1:t})$$

- To evaluate $p(m_{ij} = 1|\mathbf{z}_{1:t}, \mathbf{x}_{1:t}) = belief_t(m_{ij} = 1)$ we had to apply Bayes' theorem, which revealed a way to recursively update the belief

$$belief_t(m_{ij} = 1) = \eta \, p(\mathbf{z}_t|\mathbf{x}_t, m_{ij} = 1) \, belief_{t-1}(m_{ij} = 1)$$

Very frequent reasoning in probabilistic robotics

Recap

- We wanted to compute the likelihood of any map based on known path and observations

$$p(\mathbf{m}|\mathbf{z}_{1:t}, \mathbf{x}_{1:t}) \simeq \prod_{i,j} p(m_{ij}|\mathbf{z}_{1:t}, \mathbf{x}_{1:t})$$

- To evaluate $p(m_{ij} = 1|\mathbf{z}_{1:t}, \mathbf{x}_{1:t}) = belief_t(m_{ij} = 1)$ we had to apply Bayes' theorem, which revealed a way to recursively update the belief

$$belief_t(m_{ij} = 1) = \eta \, p(\mathbf{z}_t|\mathbf{x}_t, m_{ij} = 1) \, belief_{t-1}(m_{ij} = 1)$$

- To avoid evaluating η we used the log odds ratio

$$l_t^{(ij)} = \log \frac{p(\mathbf{z}_t|\mathbf{x}_t, m_{ij} = 1)}{p(\mathbf{z}_t|\mathbf{x}_t, m_{ij} = 0)} + l_{t-1}^{(ij)}$$

Can do this for binary random variables

Recap

- We wanted to compute the likelihood of any map based on known path and observations

$$p(\mathbf{m}|\mathbf{z}_{1:t}, \mathbf{x}_{1:t}) \quad \simeq \quad \prod_{i,j} p(m_{ij}|\mathbf{z}_{1:t}, \mathbf{x}_{1:t})$$

- To evaluate $p(m_{ij} = 1|\mathbf{z}_{1:t}, \mathbf{x}_{1:t}) = belief_t(m_{ij} = 1)$ we had to apply Bayes' theorem, which revealed a way to recursively update the belief

$$belief_t(m_{ij} = 1) \;\; = \eta \, p(\mathbf{z}_t|\mathbf{x}_t, m_{ij} = 1) \; belief_{t-1}(m_{ij} = 1)$$

- To avoid evaluating η we used the log odds ratio

$$l_t^{(ij)} = \log\frac{p(\mathbf{z}_t|\mathbf{x}_t, m_{ij} = 1)}{p(\mathbf{z}_t|\mathbf{x}_t, m_{ij} = 0)} + l_{t-1}^{(ij)}$$

- Computing the forward measurement model $p(\mathbf{z}_t|\mathbf{x}_t, m_{ij} = 1)$ was hard, so we applied Bayes' rule again, to get an inverse measurement model $p(m_{ij} = 1|\mathbf{z}_t, \mathbf{x}_t)$ and an easier-to-compute log-odds ratio:

$$l_t^{(ij)} = l_{t-1}^{(ij)} + \log\frac{p(m_{ij} = 1|\mathbf{z}_t, \mathbf{x}_t)}{p(m_{ij} = 0|\mathbf{z}_t, \mathbf{x}_t)} - \log\frac{p(m_{ij} = 1)}{p(m_{ij} = 0)}$$

Occupancy Grid Algorithm

- Upon reception of a new laser/sonar/scan measurement $\mathbf{z}_t = \{(r_k, \psi_k)\}$
- Let the robot's current state be $\mathbf{x}_t = (x_t, y_t, \theta_t)$
- Let the previous log-odds ratio of the occupancy belief be the 2D array $l_{t-1}^{(ij)}$ where i is a row, j is a column

  In the beginning we set the prior $l_0^{(ij)} = \log \frac{p(m_{ij}=1)}{p(m_{ij}=0)}$ where the occupancy probability is a design decision.

Occupancy Grid Algorithm

- Upon reception of a new laser/sonar/scan measurement $\mathbf{z}_t = \{(r_k, \psi_k)\}$
- Let the robot's current state be $\mathbf{x}_t = (x_t, y_t, \theta_t)$
- Let the previous log-odds ratio of the occupancy belief be the 2D array $l_{t-1}^{(ij)}$ where i is a row, j is a column

  In the beginning we set the prior $l_0^{(ij)} = \log\frac{p(m_{ij}=1)}{p(m_{ij}=0)}$ where the occupancy probability is a design decision.

- For all cells (i,j) in the grid
  - If the cell (i,j) is in the field of view of the robot's sensor at state $\mathbf{x}_t$

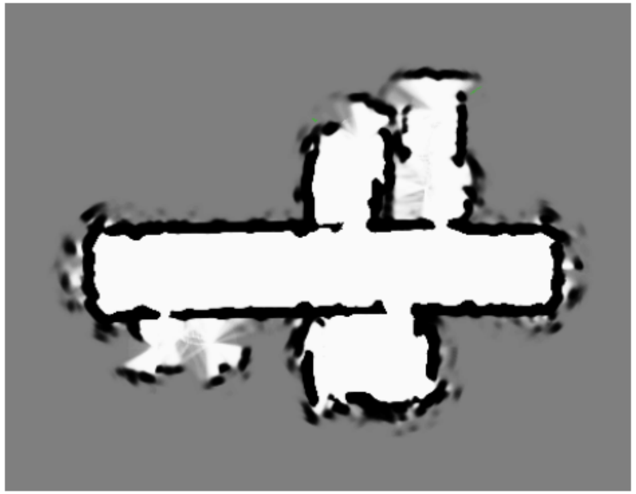
    $$l_t^{(ij)} = l_{t-1}^{(ij)} + \text{inverse-sensor-measurement-model}((i,j),\ \mathbf{x}_t,\ \mathbf{z}_t)\ - l_0^{(ij)}$$

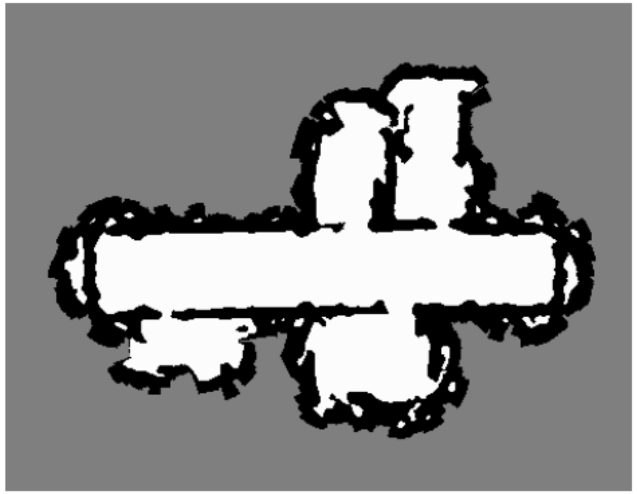
  - Else

    $$l_t^{(ij)} = l_{t-1}^{(ij)}$$

- If asked, return the following 2D matrix of occupancy probabilities: $belief_t(m_{ij}=1) = 1 - \frac{1}{1+\exp(l_t^{(ij)})}$

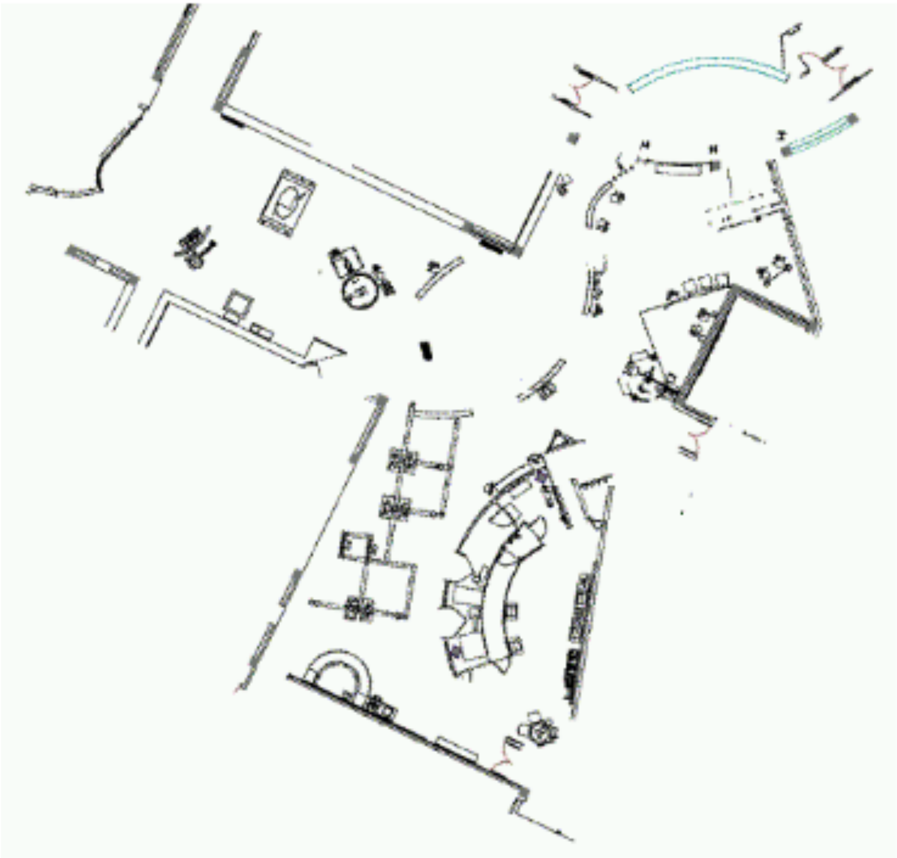
Results





The maximum likelihood map is obtained by clipping the occupancy grid map at a threshold of 0.5

Tech Museum, San Jose



CAD map

occupancy grid map