Introduction to Mobile Robotics

Grid Maps and Mapping With Known Poses

Wolfram Burgard, Diego Tipaldi
Why Mapping?

- Learning maps is one of the fundamental problems in mobile robotics.
- Maps allow robots to efficiently carry out their tasks, allow localization ...
- Successful robot systems rely on maps for localization, path planning, activity planning etc.
The General Problem of Mapping

What does the environment look like?
The General Problem of Mapping

- Formally, mapping involves, given the sensor data

\[ d = \{u_1, z_1, u_2, z_2, \ldots, u_t, z_t\} \]

- to calculate the most likely map

\[ m^* = \arg\max_m P(m \mid d) \]
The General Problem of Mapping

- Formally, mapping involves, given the sensor data
  \[ d = \{ u_1, z_1, u_2, z_2, \ldots, u_t, z_t \} \]

- to calculate the most likely map
  \[ m^* = \arg\max_m P(m \mid d) \]

- Today we describe **how to calculate a map given the robot’s pose**
The General Problem of Mapping with Known Poses

- Formally, mapping with known poses involves, given the measurements and the poses
  \[ d = \{ x_1, z_1, x_2, z_2, \ldots, x_t, z_t \} \]

- to calculate the most likely map
  \[ m^* = \arg\max_m P(m \mid d) \]
Features vs. Volumetric Maps

Courtesy by E. Nebot
Grid Maps

- We discretize the world into cells
- The grid structure is rigid
- Each cell is assumed to be occupied or free
- It is a non-parametric model
- It requires substantial memory resources
- It does not rely on a feature detector
Example
Assumption 1

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

- Each cell is a **binary random variable** that models the occupancy.
Occupancy Probability

- Each cell is a **binary random variable** that models the occupancy
  - Cell is occupied $p(m_i) = 1$
  - Cell is not occupied $p(m_i) = 0$
  - No information $p(m_i) = 0.5$
  - The **state** of the environment is assumed to be **static**
Assumption 2

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

no dependency between the cells
Representation

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

\[ p(m) = \prod_{i} p(m_i) \]
Representation

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

\[
p(m) = \prod_i p(m_i)
\]

four-dimensional vector

four independent cells
Estimating a Map From Data

- Given sensor data $z_{1:t}$ and the poses $x_{1:t}$ of the sensor, estimate the map

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

binary random variable

Binary Bayes filter (for a static state)
Static State Binary Bayes Filter

\[ p(m_i \mid z_{1:t}, x_{1:t}) \quad \text{Bayes rule} \quad \frac{p(z_t \mid m_i, z_{1:t-1}, x_{1:t}) \ p(m_i \mid z_{1:t-1}, x_{1:t})}{p(z_t \mid z_{1:t-1}, x_{1:t})} \]
Static State Binary Bayes Filter

\[ p(m_i \mid z_{1:t}, x_{1:t}) \]

Bayes rule

\[ \frac{p(z_t \mid m_i, z_{1:t-1}, x_{1:t}) \cdot p(m_i \mid z_{1:t-1}, x_{1:t})}{p(z_t \mid z_{1:t-1}, x_{1:t})} \]

Markov

\[ \frac{p(z_t \mid m_i, x_t) \cdot p(m_i \mid z_{1:t-1}, x_{1:t-1})}{p(z_t \mid z_{1:t-1}, x_{1:t})} \]
Static State Binary Bayes Filter

\[ p(m_i \mid z_{1:t}, x_{1:t}) \]

Bayes rule
\[
\frac{p(z_t \mid m_i, z_{1:t-1}, x_{1:t}) \ p(m_i \mid z_{1:t-1}, x_{1:t})}{p(z_t \mid z_{1:t-1}, x_{1:t})}
\]

Markov
\[
\frac{p(z_t \mid m_i, x_t) \ p(m_i \mid z_{1:t-1}, x_{1:t-1})}{p(z_t \mid z_{1:t-1}, x_{1:t})}
\]

\[ p(z_t \mid m_i, x_t) \]

Bayes rule
\[
\frac{p(m_i \mid z_t, x_t) \ p(z_t \mid x_t)}{p(m_i \mid x_t)}
\]
Static State Binary Bayes Filter

\[ p(m_i \mid z_{1:t}, x_{1:t}) = \frac{p(z_t \mid m_i, z_{1:t-1}, x_{1:t}) \ p(m_i \mid z_{1:t-1}, x_{1:t})}{p(z_t \mid z_{1:t-1}, x_{1:t})} \]

Markov

\[ = \frac{p(z_t \mid m_i, x_t) \ p(m_i \mid z_{1:t-1}, x_{1:t-1})}{p(z_t \mid z_{1:t-1}, x_{1:t})} \]

Bayes rule

\[ = \frac{p(m_i \mid z_t, x_t) \ p(z_t \mid x_t) \ p(m_i \mid z_{1:t-1}, x_{1:t-1})}{p(m_i \mid x_t) \ p(z_t \mid z_{1:t-1}, x_{1:t})} \]
Static State Binary Bayes Filter

\[ p(m_i \mid z_{1:t}, x_{1:t}) \]

= \[ \frac{p(z_t \mid m_i, z_{1:t-1}, x_{1:t}) \; p(m_i \mid z_{1:t-1}, x_{1:t})}{p(z_t \mid z_{1:t-1}, x_{1:t})} \]

= \[ \frac{p(z_t \mid m_i, x_t) \; p(m_i \mid z_{1:t-1}, x_{1:t-1})}{p(z_t \mid z_{1:t-1}, x_{1:t})} \]

= \[ \frac{p(m_i \mid z_t, x_t) \; p(z_t \mid x_t) \; p(m_i \mid z_{1:t-1}, x_{1:t-1})}{p(m_i \mid x_t) \; p(z_t \mid z_{1:t-1}, x_{1:t})} \]

= \[ \frac{p(m_i \mid z_t, x_t) \; p(z_t \mid x_t) \; p(m_i \mid z_{1:t-1}, x_{1:t-1})}{p(m_i) \; p(z_t \mid z_{1:t-1}, x_{1:t})} \]
Static State Binary Bayes Filter

\[ p(m_i \mid z_{1:t}, x_{1:t}) \]

Bayes rule

\[ = \frac{p(z_t \mid m_i, z_{1:t-1}, x_{1:t}) p(m_i \mid z_{1:t-1}, x_{1:t})}{p(z_t \mid z_{1:t-1}, x_{1:t})} \]

Markov

\[ = \frac{p(z_t \mid m_i, x_t) p(m_i \mid z_{1:t-1}, x_{1:t-1})}{p(z_t \mid z_{1:t-1}, x_{1:t})} \]

Bayes rule

\[ = \frac{p(m_i \mid z_t, x_t) p(z_t \mid x_t) p(m_i \mid z_{1:t-1}, x_{1:t-1})}{p(m_i \mid x_t) p(z_t \mid z_{1:t-1}, x_{1:t})} \]

Markov

\[ = \frac{p(m_i \mid z_t, x_t) p(z_t \mid x_t) p(m_i \mid z_{1:t-1}, x_{1:t-1})}{p(m_i) p(z_t \mid z_{1:t-1}, x_{1:t})} \]

Do exactly the same for the opposite event:

\[ p(\neg m_i \mid z_{1:t}, x_{1:t}) \]

the same

\[ = \frac{p(\neg m_i \mid z_t, x_t) p(z_t \mid x_t) p(\neg m_i \mid z_{1:t-1}, x_{1:t-1})}{p(\neg m_i) p(z_t \mid z_{1:t-1}, x_{1:t})} \]
Static State Binary Bayes Filter

- By computing the ratio of both probabilities, we obtain:

\[
\frac{p(m_i \mid z_{1:t}, x_{1:t})}{p(\neg m_i \mid z_{1:t}, x_{1:t})} = \frac{p(m_i \mid z_t, x_t) p(z_t \mid x_t) p(m_i \mid z_{1:t-1}, x_{1:t-1})}{p(m_i) p(z_t \mid z_{1:t-1}, x_{1:t})} \frac{p(\neg m_i \mid z_t, x_t) p(z_t \mid x_t) p(\neg m_i \mid z_{1:t-1}, x_{1:t-1})}{p(\neg m_i) p(z_t \mid z_{1:t-1}, x_{1:t})}
\]
Static State Binary Bayes Filter

- By computing the ratio of both probabilities, we obtain:

\[
\frac{p(m_i \mid z_{1:t}, x_{1:t})}{p(\neg m_i \mid z_{1:t}, x_{1:t})} = \frac{p(m_i \mid z_t, x_t) p(m_i \mid z_{1:t-1}, x_{1:t-1}) p(\neg m_i)}{p(\neg m_i \mid z_t, x_t) p(\neg m_i \mid z_{1:t-1}, x_{1:t-1}) p(m_i)}
\]

\[
= \frac{p(m_i \mid z_t, x_t)}{1 - p(m_i \mid z_t, x_t)} \frac{p(m_i \mid z_{1:t-1}, x_{1:t-1})}{1 - p(m_i \mid z_{1:t-1}, x_{1:t-1})} \frac{1 - p(m_i)}{p(m_i)}
\]
By computing the ratio of both probabilities, we obtain:

\[
\frac{p(m_i \mid z_{1:t}, x_{1:t})}{1 - p(m_i \mid z_{1:t}, x_{1:t})} = \frac{p(m_i \mid z_t, x_t) \; p(m_i \mid z_{1:t-1}, x_{1:t-1}) \; p(\neg m_i)}{p(\neg m_i \mid z_t, x_t) \; p(\neg m_i \mid z_{1:t-1}, x_{1:t-1}) \; p(m_i)}
\]

\[
= \frac{p(m_i \mid z_t, x_t)}{1 - p(m_i \mid z_t, x_t)} \; \frac{p(m_i \mid z_{1:t-1}, x_{1:t-1})}{1 - p(m_i \mid z_{1:t-1}, x_{1:t-1})} \; \frac{1 - p(m_i)}{p(m_i)}
\]

uses \( z_t \)  recursive term  prior
Occupancy Update Rule

- Recursive rule

\[
p(m_i \mid z_{1:t}, x_{1:t}) \over 1 - p(m_i \mid z_{1:t}, x_{1:t}) = \frac{p(m_i \mid z_t, x_t)}{1 - p(m_i \mid z_t, x_t)} \cdot \frac{p(m_i \mid z_{1:t-1}, x_{1:t-1})}{1 - p(m_i \mid z_{1:t-1}, x_{1:t-1})} \cdot \frac{1 - p(m_i)}{p(m_i)}
\]

uses \( z_t \)

recursive term

prior
Occupancy Update Rule

- **Recursive rule**

\[
p(m_i \mid z_{1:t}, x_{1:t}) \frac{1 - p(m_i \mid z_{1:t}, x_{1:t})}{1 - p(m_i \mid z_t, x_t)} = \frac{p(m_i \mid z_t, x_t)}{1 - p(m_i \mid z_t, x_t)} \frac{p(m_i \mid z_{1:t-1}, x_{1:t-1})}{1 - p(m_i \mid z_{1:t-1}, x_{1:t-1})} \frac{1 - p(m_i)}{p(m_i)}
\]

uses \( z_t \) recursive term prior

- **Often written as**

\[
Bel(m^i_t) = \left[ 1 + \frac{1 - p(m^i_t \mid z_t, x_t)}{p(m^i_t \mid z_t, x_t)} \frac{p(m^i_t)}{1 - p(m^i_t)} \frac{1 - Bel(m^i_{t-1})}{Bel(m^i_{t-1})} \right]^{-1}
\]
Log Odds Notation

- Log odds ratio is defined as

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

- and with the ability to retrieve \( p(x) \)

\[ p(x) = 1 - \frac{1}{1 + \exp l(x)} \]
Occupancy Mapping in Log Odds Form

- The product turns into a sum

$$l(m_i \mid z_{1:t}, x_{1:t}) = l(m_i \mid z_t, x_t) + l(m_i \mid z_{1:t-1}, x_{1:t-1}) - l(m_i)$$

  inverse sensor model  recursive term  prior

- or in short

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

$\text{occupancy\_grid\_mapping} \left( \{l_{t-1,i}\}, x_t, z_t \right)$:

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

highly efficient, only requires to compute sums
Occupancy Grid Mapping

- Developed in the mid 80’s by Moravec and Elfes
- Originally developed for noisy sonars
- Also called “mapping with know poses”
Inverse Sensor Model for Sonars Range Sensors

In the following, consider the cells along the optical axis (red line)
Occupancy Value Depending on the Measured Distance

- $z + d_1$
- $z + d_2$
- $z - d_1$
- $z$
- $z + d_3$

Distance between the cell and the sensor

Occupancy probability

Prior

measured dist.
Occupancy Value Depending on the Measured Distance

The graph shows the occupancy probability as a function of the distance between the cell and the sensor. The distance is measured as $z-d_1$, $z$, $z+d_1$, $z+d_2$, and $z+d_3$. The occupancy probability $P$ is represented on the y-axis, ranging from 0 to 1. The text "free" indicates the area where the sensor is likely to be unobstructed.
Occupyancy Value Depending on the Measured Distance

![Graph showing occupancy probability depending on measured distance between the cell and the sensor.](image-url)

- Occupancy probability
- "occ"
- distance between the cell and the sensor
- Prior
- Measured dist.
- $z + d_1$, $z + d_2$, $z + d_3$
- $z - d_1$, $z$
Occupancy Value Depending on the Measured Distance

P

0 0.2 0.4 0.6 0.8 1

0 0.5 1 1.5 2 2.5 3

distance between the cell and the sensor

Occupancy probability

z+d₁  z+d₂  z+d₃

“no info” prior

z  measured dist.

z-d₁
Update depends on the Measured Distance and Deviation from the Optical Axis

- Linear in $z_t$
- Gaussian in $\theta$

$$m_l = m_{d,\theta}(x_t)$$
Intensity of the Update
Resulting Model
Example: Incremental Updating of Occupancy Grids
Resulting Map Obtained with Ultrasound Sensors
Resulting Occupancy and Maximum Likelihood Map

The maximum likelihood map is obtained by rounding the probability for each cell to 0 or 1.
Inverse Sensor Model for Laser Range Finders

probability

\( P_{\text{occ}} \)

\( P_{\text{prior}} \)

\( P_{\text{free}} \)

Occupancy probability

\[ z_{t,n} + \frac{r}{2} \]

\[ z_{t,n} - \frac{r}{2} \]

distance between sensor and cell under consideration
Occupancy Grids
From Laser Scans to Maps
Example: MIT CSAIL 3\textsuperscript{rd} Floor
Uni Freiburg Building 106
Alternative: Counting Model

- For every cell count
  - $\text{hits}(x,y)$: number of cases where a beam ended at $<x,y>$
  - $\text{misses}(x,y)$: number of cases where a beam passed through $<x,y>$

\[
Bel(m^{xy}) = \frac{\text{hits}(x,y)}{\text{hits}(x,y) + \text{misses}(x,y)}
\]

- Value of interest: $P(\text{reflects}(x,y))$
The Measurement Model

- Pose at time $t$: $x_t$
- Beam $n$ of scan at time $t$: $z_{t,n}$
- Maximum range reading: $\zeta_{t,n} = 1$
- Beam reflected by an object: $\zeta_{t,n} = 0$

$\text{measured dist. in } \#\text{cells}$
The Measurement Model

- Pose at time $t$: $x_t$
- Beam $n$ of scan at time $t$: $z_{t,n}$
- Maximum range reading: $\zeta_{t,n} = 1$
- Beam reflected by an object: $\zeta_{t,n} = 0$

max range: “first $z_{t,n}-1$ cells covered by the beam must be free”

$$p(z_{t,n}|x_t, m) = \begin{cases} 
\prod_{k=0}^{z_{t,n}-1} (1 - m_f(x_t,n,k)) & \text{if } \zeta_{t,n} = 1 
\end{cases}$$
The Measurement Model

- Pose at time $t$: $x_t$
- Beam $n$ of scan at time $t$: $z_{t,n}$
- Maximum range reading: $\zeta_{t,n} = 1$
- Beam reflected by an object: $\zeta_{t,n} = 0$

max range: “first $z_{t,n}-1$ cells covered by the beam must be free”

$$p(z_{t,n}|x_t, m) = \begin{cases} 
\prod_{k=0}^{z_{t,n}-1} (1 - m_f(x_t,n,k)) & \text{if } \zeta_{t,n} = 1 \\
m_f(x_t,n,z_{t,n}) \prod_{k=0}^{z_{t,n}-1} (1 - m_f(x_t,n,k)) & \zeta_{t,n} = 0 
\end{cases}$$

otherwise: “last cell reflected beam, all others free”
Computing the Most Likely Map

- Compute values for $m$ that maximize
  
  $$m^* = \arg\max_m P(m \mid z_1, \cdots, z_t, x_1, \cdots, x_t)$$

- Assuming a uniform prior probability for $P(m)$, this is equivalent to maximizing:

  $$m^* = \arg\max_m P(z_1, \cdots, z_t \mid m, x_1, \cdots, x_t)$$

  $$= \arg\max_m \prod_{t=1}^{T} P(z_t \mid m, x_t)$$

  $$= \arg\max_m \sum_{t=1}^{T} \ln P(z_t \mid m, x_t)$$

  since $z_t$ independent and only depend on $x_t$
Computing the Most Likely Map

\[ m^* = \arg \max_m \sum_{j=1}^{J} \sum_{t=1}^{T} \sum_{n=1}^{N} \left( I(f(x_t, n, z_{t,n}) = j) \cdot (1 - \zeta_{t,n}) \cdot \ln m_j \right) \]

\[ + \sum_{k=0}^{z_{t,n}-1} I(f(x_t, n, k) = j) \cdot \ln(1 - m_j) \]
Computing the Most Likely Map

\[ m^* = \arg\max_{m} \sum_{j=1}^{J} \sum_{t=1}^{T} \sum_{n=1}^{N} \left( I(f(x_t, n, z_{t,n}) = j) \cdot (1 - \zeta_{t,n}) \cdot \ln m_j \right) \]

\[ + \sum_{k=0}^{z_{t,n}-1} I(f(x_t, n, k) = j) \cdot \ln(1 - m_j) \]
Computing the Most Likely Map

\[ m^* = \arg\max_m \sum_{j=1}^{J} \sum_{t=1}^{T} \sum_{n=1}^{N} \left( I(f(x_t, n, z_{t,n}) = j) \cdot (1 - \zeta_{t,n}) \cdot \ln m_j \right) \]

\[ + \sum_{k=0}^{z_{t,n}-1} I(f(x_t, n, k) = j) \cdot \ln(1 - m_j) \]

“beam \( n \) ends in cell \( j \)”

“beam \( n \) traversed cell \( j \)”
Computing the Most Likely Map

\[ m^* = \arg\max_m \sum_{j=1}^{J} \sum_{t=1}^{T} \sum_{n=1}^{N} \left( I(f(x_t, n, z_{t,n}) = j) \cdot (1 - \zeta_{t,n}) \cdot \ln m_j \right) \]

\[ + \sum_{k=0}^{z_{t,n}-1} I(f(x_t, n, k) = j) \cdot \ln(1 - m_j) \]

Define

\[ \alpha_j = \sum_{t=1}^{T} \sum_{n=1}^{N} I(f(x_t, n, z_{t,n}) = j) \cdot (1 - \zeta_{t,n}) \]

\[ \beta_j = \sum_{t=1}^{T} \sum_{n=1}^{N} \sum_{k=0}^{z_{t,n}-1} I(f(x_t, n, k) = j) \]
Meaning of $\alpha_j$ and $\beta_j$

$$\alpha_j = \sum_{t=1}^{T} \sum_{n=1}^{N} I(f(x_t, n, z_{t,n}) = j) \cdot (1 - \zeta_{t,n})$$

Corresponds to the number of times a beam that is not a maximum range beam ended in cell $j$ ($hits(j)$)

$$\beta_j = \sum_{t=1}^{T} \sum_{n=1}^{N} \sum_{k=0}^{z_{t,n}-1} I(f(x_t, n, k) = j)$$

Corresponds to the number of times a beam traversed cell $j$ without ending in it ($misses(j)$)
Computing the Most Likely Map

Accordingly, we get

\[ m^* = \arg \max_m \sum_{j=1}^{J} \left( \alpha_j \ln m_j + \beta_j \ln(1 - m_j) \right) \]

As the \( m_j \)'s are independent we can maximize this sum by maximizing it for every \( j \)

If we set

\[ \frac{\partial}{\partial m_j} = \frac{\alpha_j}{m_j} - \frac{\beta_j}{1-m_j} = 0 \]

we obtain

\[ m_j = \frac{\alpha_j}{\alpha_j + \beta_j} \]

Computing the most likely map reduces to counting how often a cell has reflected a measurement and how often the cell was traversed by a beam.
Difference between Occupancy Grid Maps and Counting

- The counting model determines how often a cell reflects a beam.
- The occupancy model represents whether or not a cell is occupied by an object.
- Although a cell might be occupied by an object, the reflection probability of this object might be very small.
Example Occupancy Map
Example Reflection Map

glass panes
Example

- Out of $n$ beams only 60% are reflected from a cell and 40% intercept it without ending in it.

- Accordingly, the reflection probability will be 0.6.

- Suppose $p(\text{occ} \mid z) = 0.55$ when a beam ends in a cell and $p(\text{occ} \mid z) = 0.45$ when a beam traverses a cell without ending in it.

- Accordingly, after $n$ measurements we will have

$$
\left( \frac{0.55}{0.45} \right)^{n \cdot 0.6} \ast \left( \frac{0.45}{0.55} \right)^{n \cdot 0.4} = \left( \frac{11}{9} \right)^{n \cdot 0.6} \ast \left( \frac{11}{9} \right)^{-n \cdot 0.4} = \left( \frac{11}{9} \right)^{n \cdot 0.2}
$$

- The reflection map yields a value of 0.6, while the occupancy grid value converges to 1 as $n$ increases.
Summary (1)

- Grid maps are a popular model for representing the environment
- Occupancy grid maps discretize the space into independent cells
- Each cell is a binary random variable estimating if the cell is occupied
- We estimate the state of every cell using a binary Bayes
- This leads to an efficient algorithm for mapping with known poses
- The log odds model is fast to compute
Summary (2)

- Reflection probability maps are an alternative representation.
- The key idea of the sensor model is to calculate for every cell the probability that it reflects a sensor beam.
- Given the this sensor model, counting the number of times how often a measurement intercepts or ends in a cell yields the maximum likelihood model.