Occupancy Grid Mapping

COMP.4510 and COMP.5490
Fall 2017

These lecture slides were adapted from slides on the book’s website (PR) and from Cyrill Stachniss (CS)
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,

\[ d = \{u_1, z_1, u_2, z_2, \ldots, u_n, z_n\} \]

...to calculate the most likely map

\[ m^* = \text{arg max}_m P(m \mid d) \]
Mapping as a Chicken and Egg Problem

• So far we learned how to estimate the pose of the vehicle given the data and the map.
• Mapping, however, involves to simultaneously estimate the pose of the vehicle and the map.
• The general problem is therefore denoted as the simultaneous localization and mapping problem (SLAM).
• Throughout this section we will describe how to calculate a map given we know the pose of the vehicle.
Types of SLAM-Problems

- **Grid maps or scans**

  [Lu & Milios, 97; Gutmann, 98; Thrun 98; Burgard, 99; Konolige & Gutmann, 00; Thrun, 00; Arras, 99; Haehnel, 01; ...]

- **Landmark-based**

  [Leonard et al., 98; Castelanos et al., 99; Dissanayake et al., 2001; Montemerlo et al., 2002; ...]
Problems in Mapping

• **Sensor interpretation**
  • How do we *extract relevant information from raw sensor data*?
  • How do we represent and *integrate* this information *over time*?

• **Robot locations have to be estimated**
  • How can we identify that we are at a *previously visited place*?
  • This problem is the so-called *data association problem*.
Grid Maps

- Discretize the world into cells
- Grid structure is rigid
- Each cell is assumed to be occupied or free space
- Non-parametric model
- Large maps require substantial memory resources
- Do not rely on a feature detector
Example
Occupancy Grid Maps

- Introduced by Moravec and Elfes in 1985
- Represent environment by a grid
- Estimate the probability that a location is occupied by an obstacle

**Key assumptions**
- Occupancy of individual cells \((m[xy])\) is independent

\[
Bel(m_t) = P(m_t \mid u_1, z_2, \ldots, u_{t-1}, z_t) = \prod_{x,y} Bel(m_{t[xy]})
\]

- Robot positions are known!
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

\[ p(m_i) \rightarrow 1 \quad p(m_j) \rightarrow 0 \]
Occupyancy 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 knowledge: \( p(m_i) = 0.5 \)
Assumption 2

- The world is **static** (most mapping systems make this assumption)
Assumption 3

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

![Diagram showing no dependency between cells]
Representation

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

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

map \quad cell
Representation

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

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

Example map (4-dim state) 4 individual 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

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

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

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

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

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

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

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

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

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

Markov

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

Bayes rule

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

Markov

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

Bayes rule

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

Markov

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

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

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

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

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

Do exactly the same for the opposite event:

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

CS 24
Static State Binary Bayes Filter

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

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

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

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

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

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

- uses $z_t$
- recursive term
- prior
From Ratio to Probability

We can turn the ratio into a probability:

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

\[ p(x) = Y - Y \cdot p(x) \]

\[ p(x) \cdot (1 + Y) = Y \]

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

\[ p(x) = \frac{1}{1 + \frac{1}{Y}} \]
From Ratio to Probability

- Using \( p(x) = [1 + Y^{-1}]^{-1} \) directly leads to

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

For reasons of efficiency, one performs the calculations in the log odds notation.
Log Odds Notation

- The log odds notation computes the logarithm of the ratio of probabilities

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

uses $z_t$ recursive term prior

\[
l(m_i | z_{1:t}, x_{1:t}) = \log \left( \frac{p(m_i | z_{1:t}, x_{1:t})}{1 - p(m_i | z_{1:t}, x_{1:t})} \right)
\]
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

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

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

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} + 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, we only have to compute sums
Inverse Sensor Model for Sonar Range Sensors

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

![Graph showing the relationship between occupancy probability and distance between the cell and the sensor.](image)

- "free" label
- Prior and measured distance markers
- Distance values: z, z+d1, z+d2, z+d3
- Occupancy probability scale: 0.0 to 1.0

(distance between the cell and the sensor)
Occupancy Value Depending on the Measured Distance

- Occupancy probability
- "occ"
- $z + d_1$
- $z + d_2$
- $z - d_1$
- $z + d_3$
- Prior
- Measured dist.

Distance between the cell and the sensor
Occupancy Value Depending on the Measured Distance

![Graph showing occupancy probability with distance between the cell and the sensor.]
Incremental Updating of Occupancy Grids (Example)
Resulting Map Obtained with Ultrasound Sensors
Resulting Occupancy and Maximum Likelihood Map

The maximum likelihood map is obtained by clipping the occupancy grid map at a threshold of 0.5 (rounding to 0 or 1)
Inverse Sensor Model for Laser Range Finders

\[
p_{\text{occ}} \quad p_{\text{prior}} \quad p_{\text{free}}
\]

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

distance between sensor and cell under consideration
Occupancy Grids
From Laser Scans to Maps
Example: MIT CSAIL 3rd Floor
Occuancy Grid Map 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
Grid Mapping Meets Reality...
Mapping With Raw Odometry
Incremental Scan Alignment

- Motion is noisy, we cannot ignore it
- Assuming known poses fails!
- Often, the sensor is rather precise

- Scan-matching tries to incrementally align two scans or a map to a scan, without revising the past/map
Pose Correction Using Scan-Matching

Maximize the likelihood of the current pose relative to the previous pose and map

\[ x_t^* = \underset{x_t}{\text{argmax}} \left\{ p(z_t \mid x_t, m_{t-1}) \ p(x_t \mid u_{t-1}, x_{t-1}^*) \right\} \]

- current measurement
- robot motion
- map constructed so far
Incremental Alignment

Courtesy by E. Olson
Incremental Alignment

Courtesy by E. Olson
Various Different Ways to Realize Scan-Matching

- Iterative closest point (ICP)
- Scan-to-scan
- Scan-to-map
- Map-to-map
- Feature-based
- RANSAC for outlier rejection
- Correlative matching
- ...

CS
With and Without Scan-Matching

Courtesy by D. Hähnel
Scan Matching Summary

- Scan-matching improves the pose estimate (and thus mapping) substantially
- Locally consistent estimates
- Often scan-matching is not sufficient to build a (large) consistent map
Summary

• Occupancy grid maps are a popular approach to represent the environment of a mobile robot given known poses.
• In this approach each cell is considered independently from all others.
• It stores the posterior probability that the corresponding area in the environment is occupied.
• Occupancy grid maps can be learned efficiently using a probabilistic approach.
Reading

Static state binary Bayes filter

- Probabilistic Robotics
  Section 4.2

Occupancy grid mapping

- Probabilistic Robotics
  Sections 9.1 and 9.2