QUANSER
INNOVATE·EDUCATE

Concept Review

Occupancy Grid Mapping

Why Construct a Map?

Modern autonomous vehicles make decisions on where to travel in their environment based on perception interpretation techniques. These techniques can be applied to a global understanding of the vehicle's position or a local planner in charge of deciding what areas are safe to drive on. In this document occupancy grids are used as one technique for environment interpretation.

Introduction

For a robot to autonomously navigate through its environment safely, it must have some insight into what areas are freely traversable, and what areas are not traversable (due to the presence of obstacles, unsafe terrain, etc...). Given the continuous nature of the world, building an exact representation of the environment surrounding a robot is a challenging task. Instead, we can simplify things by discretizing the world into a grid of equally spaced cells, as shown in Figure *1*. In this type of model (referred to as an 'occupancy grid'), we care only about whether each cell is free (traversable) or occupied (not traversable). Since each cell can only be in one of two states (occupied or free), each cell is modelled as an independent binary variable, having a value of 0 if free or 1 if occupied.

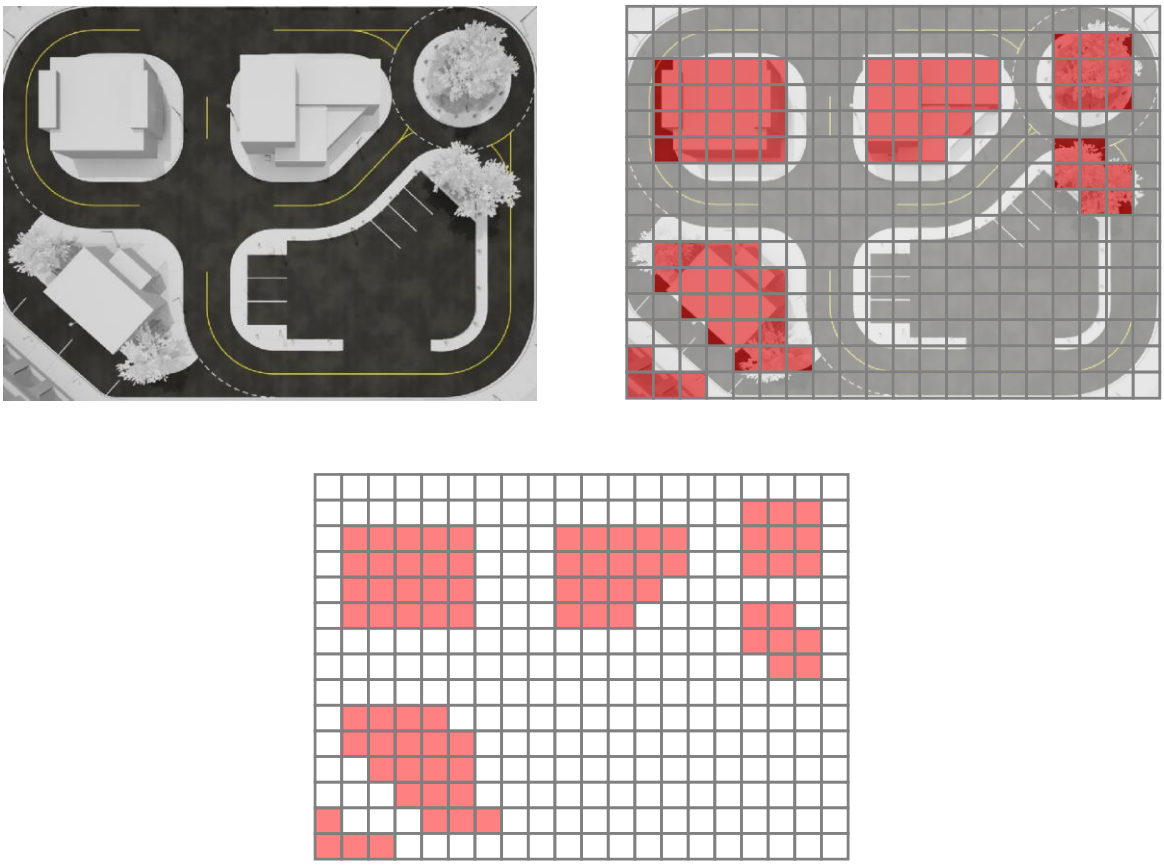

Figure 1: Grid-based environment representation: white and red cells represent free and occupied areas respectively.

The simplicity of the occupancy grid representation is especially useful for tasks such as path planning and constructing the map of an unknown environment using sensor data. The main drawback of grid-based representations is that information is lost during the discretization process. The accuracy of a map can be improved by decreasing the cell-size, but as the size of the grid grows, so too does the computational burden. Thus, there is a tradeoff between accuracy and performance.

Map Definition and Interpretation

Mathematically, an occupancy grid can be seen as an $m \times n$ matrix, $\mathbf{M}$, where $m_{i,j}$ corresponds to the cell in the ith row and jth column of $\mathbf{M}$. The minimum size of an occupancy grid required to cover a given environment can be determined by three parameters:

- Minimum required length in the x direction,
- Minimum required length in the y direction,
- Width of each cell.

These three parameters are denoted as L_x, L_y, and w respectively and can be used to calculate the minimum required m and n values as follows:

$$\begin{aligned} m &= \left(\frac{L_y}{w}\right) \\ n &= \left(\frac{L_x}{w}\right) \end{aligned}$$

To discretize the occupancy grid, the ratios represented by m or n need to be rounded up to the nearest integer value. This will ensure the actual grid size is at least as large as the minimum required size specified by L_x and L_y.

For map $\mathbf{M}$ to be useful in describing a robot's surroundings, there must be a 1-to-1 correspondence between a point's location in the real world, and its location in the grid. To this end, we can define two frames: a grid frame $G = \{i, j\}$, and a world frame $W = \{x, y\}$, each with coordinate axes oriented as shown in Figure 2. Superscript notation identifies the frame in which a particular item is expressed. For example, $\mathbf{p}^G$ denotes point $\mathbf{p}$ expressed in frame G while $\mathbf{p}^W$ denotes the same location expressed in frame W. Additionally, $\mathbf{c}_{i,j}$ denotes the location at the centre of cell i, j.

Given a point, $\mathbf{p}$, expressed in the grid frame, it can be converted to its equivalent in the world frame by the following relation:

$$\mathbf{p}^W = \begin{bmatrix} w & 0 \\ 0 & -w \end{bmatrix} \mathbf{p}^G + \begin{bmatrix} \frac{-n \cdot w}{2} \\ \frac{m \cdot w}{2} \end{bmatrix}$$

Similarly, to convert from frame W to frame G by rearranging the equation above:

$$\mathbf{p}^G = \begin{bmatrix} \frac{1}{w} & 0 \\ 0 & \frac{1}{-w} \end{bmatrix} \left(\mathbf{p}^W - \begin{bmatrix} \frac{-n \cdot w}{2} \\ \frac{m \cdot w}{2} \end{bmatrix} \right)$$

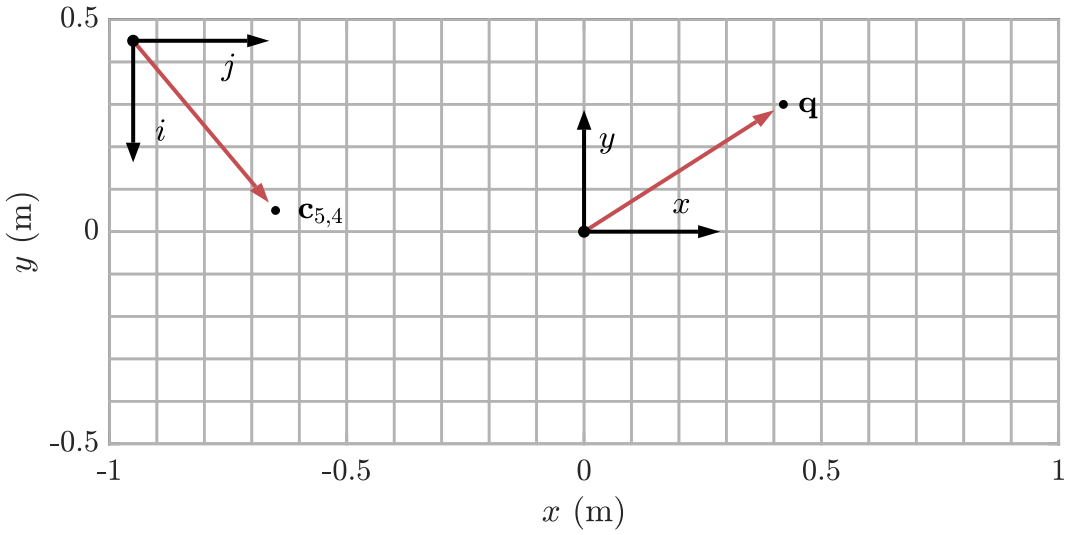

Figure 2: Orientation of grid frame and world frame coordinate axes

Occupancy Grid Mapping

The goal of occupancy grid mapping algorithms is to learn the map of an environment using a sequence of given robot states and sensor measurements. For an occupancy grid of size $m \times n$, there are mn binary variables that together must be identified from a set of 2^{mn} possible map configurations. For even moderately sized grids, the set of all possible maps is intractably large, making it impossible to estimate an entire map directly. Instead, we break the problem down into a collection of subproblems by assuming the state of each cell is independent of all others. This allows us to replace our original mn-dimensional problem with mn 1-dimensional subproblems (one for each cell) each of which can easily be solved using a binary Bayes filter.

Binary Bayes Filter with Static State

For each cell, $m_{i,j}$ in $\mathbf{M}$, the aim is to estimate the likelihood of $m_{i,j}$ being occupied given a sequence of sensor measurements and vehicle states over time ($\mathbf{z}_{1:t}$ and $\mathbf{x}_{1:t}$ respectively). Specifically, what we seek to estimate is

$$p(m_{i,j} \mid \mathbf{z}_{1:t}, \mathbf{x}_{1:t})$$

for each cell. In order to deal with numerical issues such as rounding and truncation errors that occur as p approaches its limits of 0 or 1, it is common to restate p to be in log odds form using the 'logit' function, defined as follows:

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

The 'logit' function extends the range of probabilities $(0, 1)$ to a range of $(-\infty, \infty)$, as shown in Figure 3. Using the log odds form has a secondary benefit in that filter

updates require only addition operations to be performed. The original probability can be returned by simply rearranging the equation above:

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

Applying the logit transformation, we obtain the binary Bayes filter in log-odds form:

$$l_t(m_{i,j}) = l_{t-1}(m_{i,j}) + \log\frac{p(m_{i,j}|\ \mathbf{z}_t, \mathbf{x}_t)}{1 - p(m_{i,j}|\ \mathbf{z}_t, \mathbf{x}_t)} - l_0$$

Where l_0 constitutes the initial likelihood before any measurements have been taken. Without prior knowledge of the environment, assume each cell is equally likely to be occupied as it is to be free, giving $p_0\left(m_{i,j}\right) = 0.5$. Thus, for each cell

$$l_0 = \log\frac{p_0(m_{i,j})}{1 - p_0(m_{i,j})} = \log\frac{0.5}{1 - 0.5} = 0$$

The bayes filter update equation can then be simplified to

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

$p(m_{i,j}|\ \mathbf{z}_t, \mathbf{x}_t)$ comes from the sensor specific inverse measurement model (the likelihood of $m_{i,j}$ being occupied given measurement $\mathbf{z}_t$). For an example of an inverse measurement model refer to Concept Review - LiDAR Inverse Measurement model

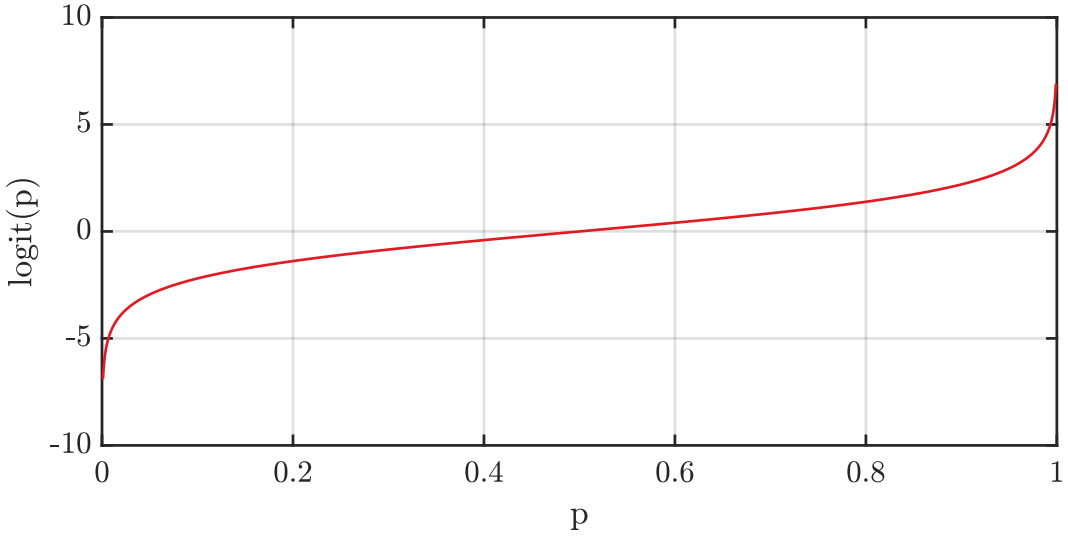

Figure 3: Logit function for the range of probabilities from 0 to 1

Solutions for teaching and research. Made in Canada.