Statistical Techniques in Robotics (16-831, F12) Lecture#05 (Wednesday, September 12)

Mapping

Lecturer: Alex Styler (in for Drew Bagnell) *Scribe: Victor Hwang*[1]

1 Occupancy Mapping

When solving the localization problem, we had a map of the world and tried to find the position of the robot in it. Now we will solve the opposite problem: given the position of the robot (perhaps through some sensor such as GPS), find the map of the world. Similar to localization, we will solve the mapping problem as a filtering problem.

We need two components to solve this filtering problem: a state and a measurement model. The state is the map of the world which we are trying to find. For example, it could be a pixelized grid of cells or a map of landmarks. The measurement model is $p(z_t|m, l_t)$, or the probability of making an observation z_t given a map m and a location on the map l_t. We do not need to concern ourselves with the robot's controls since we have the position of the robot. As such, there is no motion model explicitly defined.

For our first solution to the mapping problem, we will partition the world into cells, with each being one of two states: filled or empty. The individual grid cells are labeled m_i, and $\vec{m}$ is the vector of all grid cells. Our goal is to calculate the posterior

$$p(m|z_{1:t}, x_{1:t})$$

Unfortunately, the curse of dimensionality prevents us from filtering $\vec{m}$, since there are $2^{|\vec{m}|}$ possible states. If we filter each cell independently, assuming that they are in fact independent, we can reduce the complexity to $2|\vec{m}|$ by reconstructing the map from products of the map's marginal probability:

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

where m_i is each individual grid cell. In truth, this is a very bad assumption to make since obstacles usually span multiple cells. However, this makes the problem tractable.

This is called "occupancy mapping".

1.1 Derivation

Let X^i represent the state of a grid cell m_i. The state is either x, meaning filled, or $\bar{x}$, meaning empty. We look at the probability that a cell is filled given the measurements, starting with a Bayesian filter.

[1] Content adapted from previous scribes: Nathan Brooks, Brian Coltin and Mehmet R. Dogar

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

Now we will make the Markov assumption that $p(z_t|x, z_{1:t-1}) = p(z_t|x)$.

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

However, this is totally bogus! This is a statement about a single cell, but the observation does not depend on just a single cell. This is detailed in Section 1.2.

We expand the equation once again using Bayes' rule to get

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

Now $p(x|z_{1:t})$ is based on the *inverse sensor model*, $p(x|z_t)$, instead of the familiar forward model $p(z_t|x)$. The inverse sensor model specifies a distribution over the (binary) state variable as a function of the measurement z_t. A sensor model for a laser scanner device might look like Figure 1, where z_t is pass-through / hit information.

Using the same proof, we can derive a matching update rule for $p(\bar{x}|z_{1:t})$:

$$p(\bar{x}|z_{1:t}) = \frac{p(\bar{x}|z_t)p(z_t)}{p(\bar{x})} \frac{p(\bar{x}|z_{1:t-1})}{p(z_t|z_{1:t-1})}.$$

Next, we divide the two update rules.

$$\begin{aligned}
\frac{p(x|z_{1:t})}{p(\bar{x}|z_{1:t})} &= \frac{p(x|z_t)}{p(\bar{x}|z_t)} \frac{p(\bar{x})}{p(x)} \frac{p(x|z_{1:t-1})}{p(\bar{x}|z_{1:t-1})} \frac{p(z_t|z_{1:t-1})}{p(z_t|z_{1:t-1})} \\
\frac{p(x|z_{1:t})}{p(\bar{x}|z_{1:t})} &= \frac{p(x|z_t)}{p(\bar{x}|z_t)} \frac{p(\bar{x})}{p(x)} \frac{p(x|z_{1:t-1})}{p(\bar{x}|z_{1:t-1})} \\
&= \frac{p(x|z_t)}{p(x)} \frac{p(\bar{x})}{p(\bar{x}|z_t)} \frac{p(x|z_{1:t-1})}{p(\bar{x}|z_{1:t-1})}
\end{aligned}$$

Now we have a recursive update rule. If the probability of $\bar{x}$ decreases with the observation z_t, then $p(\bar{x}) > p(\bar{x}|z_t)$, which causes the belief on x to increase relative to $\bar{x}$.

Next, we take the log of this update rule to find the log odds of the belief the square is filled over the belief it isn't filled, which we label $l_t(x)$. Using the log odds reduces numerical errors from multiplying minuscule floating point numbers.

$$\begin{aligned}
\log \frac{p(x|z_{1:t})}{p(\bar{x}|z_{1:t})} &= \log \frac{p(x|z_t)}{p(\bar{x}|z_t)} \frac{p(\bar{x})}{p(x)} \frac{p(x|z_{1:t-1})}{p(\bar{x}|z_{1:t-1})} \\
l_t(x) &= \log \frac{p(x|z_t)}{p(\bar{x}|z_t)} + \log \frac{p(\bar{x})}{p(x)} + l_{t-1}(x)
\end{aligned}$$

For this update rule, we need only to specify $p(x|z_t)$, the inverse sensor model, and $p(x)$, the prior. $p(\bar{x}|z_t)$ and $p(\bar{x})$ are the complements of these two terms. Note that the inverse sensor model must respond to updates to the prior: consider section (1) in Figure 1.

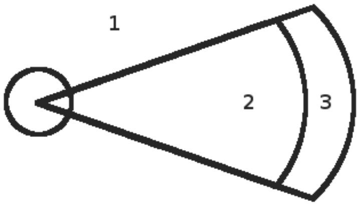

Figure 1: A sample sensor model for a laser scanner device which provides the probability a grid cell is occupied given a sensor reading. Grid cells in (1) would have probability equal to the prior, grid cells in (2) would have a low probability and grid cells in (3) would have a high probability.

1.2 Problems with the Markov Assumption

At the beginning of this derivation we used the Markov assumption to claim that $p(z_t|x, z_{1:t-1}) = p(z_t|x)$. This would have been reasonable had x represented the state of the complete map. However, x represents the state of a single grid cell. The Markov assumption in this context doesn't make much sense: we can't say that an observation z_t is independent of all prior observations given only the state of a single cell.

Figure 2 shows an example using a wide laser beam where this Markov assumption fails. In that case, the odds of the conflicting cell would increase, then decrease later. Without the Markov assumption, we could explain away this inconsistency by taking into account the entire map. Given this natural error, narrow beam sensors (such as LIDAR) are very popular.

1.3 Limitations of Occupancy Mapping

In occupancy grid mapping every grid cell is one of two states: filled or empty. But in some situations it makes sense for a cell to be partially filled. This may occur when only part of the grid cell is filled and the rest is empty, or when the objects that "fill" the grid cell have special characteristics: we may want a grid filled with vegetation to be "less filled" than a grid filled with a solid rock.

Furthermore, occupancy grids have trouble dealing with semi-transparent obstacles such as glass and vegetation. These obstacles may return hits to the laser rangefinder about half of the time, but eventually the occupancy grid will converge to either filled or not filled, both of which are incorrect.

2 Density Mapping

To address the above problems, we use a density filter where the state $x \in [0, 1]$ is the probability a beam passes through rather than a binary filled / empty state. Assuming that a beam returns with a probability x according to a Bernoulli distribution, we model this probability using a Beta

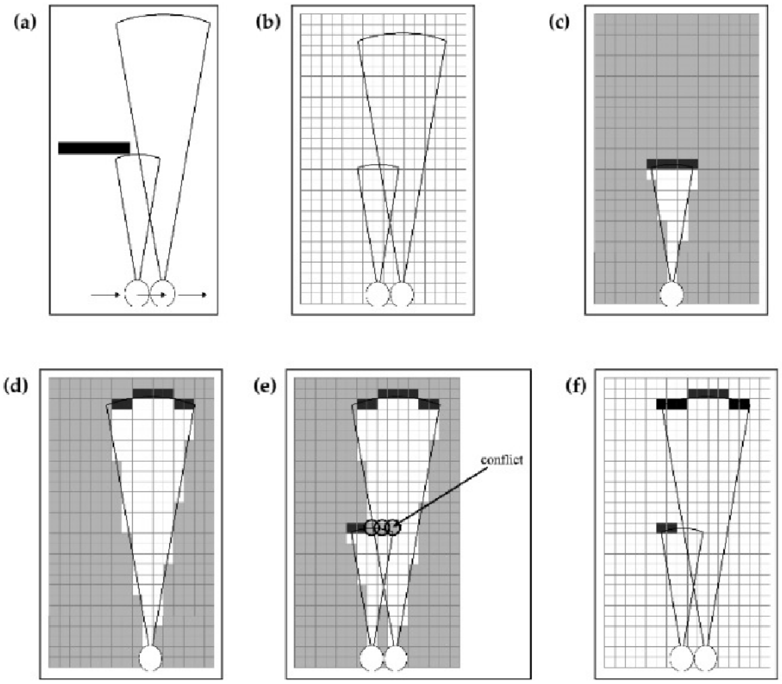

Figure 2: The problem with the standard occupancy grid mapping algorithm in Chapter 9.2: For the environment shown in Figure (a), a passing robot might receive the (noise-free) measurement shown in (b). The occupancy grid map approach maps these beams into probabilistic maps separately for each grid cell and each beam, as shown in (c) and (d). Combining both interpretations yields the map shown in (e). Obviously, there is a conflict in the overlap region, indicated by the circles in (e). The interesting insight is: There exist maps, such as the one in diagram (f), that perfectly explain the sensor measurement without any such conflict. For a sensor reading to be explained, it suffices to assume an obstacle *somewhere* in the cone of a measurement, and not everywhere.

distribution. If α is the number of beams that pass through and β is the number of beams that hit, the most probable value of a cell's density is

$$\frac{\alpha}{\alpha + \beta}$$

and we use this as the value in the density grid. A comparison between occupancy mapping and density mapping highlighting this difference can be seen in Figures 3 and 4.

Note that density grids make the same flawed Markov assumption as occupancy grids, but do not assume the cells can only be completely full or completely empty. To further take advantage this partial occupancy model, the sensor model seen in Figure 1 could be modified so that sensor model returns for areas near borders in the model are “blurred.“

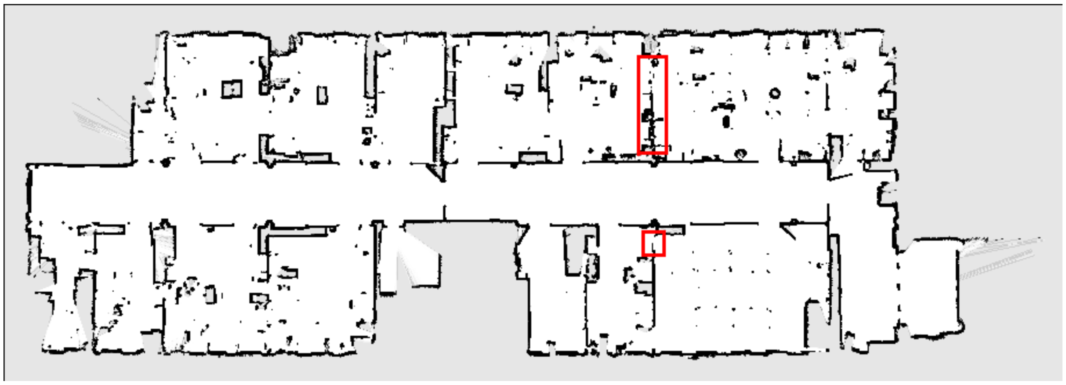

Figure 3: A map produced using an occupancy mapping technique. Regions in red indicate glass walls which are indicated as being either fully occupied or non-occupied.

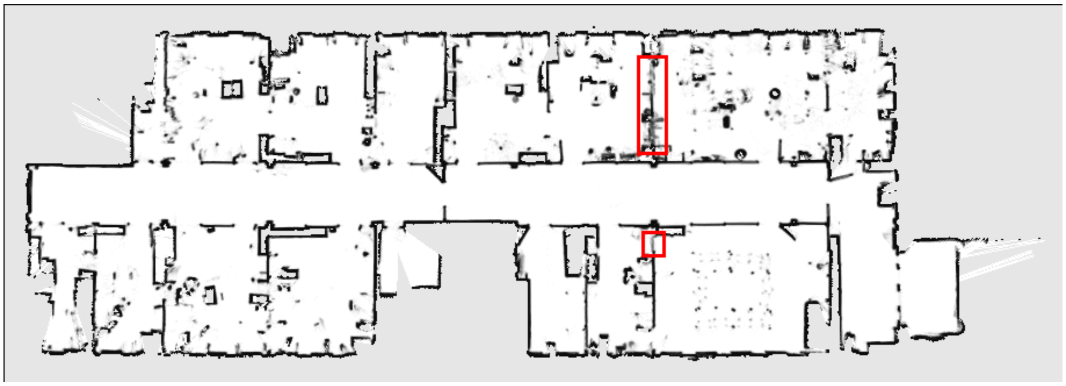

Figure 4: A map produced using a density mapping technique. Regions in red indicate glass walls which are indicated as having partial returns.