Statistical Techniques in Robotics (16-831, F10) Lecture#06(Thursday September 11)

Occupancy Maps

Lecturer: Drew Bagnell *Scribes: {agiri, dmcconac, kumarsha, nbhakta}* [1]

1 Occupancy Mapping: An Introduction

Occupancy Grid Mapping refers to a family of computer algorithms in probabilistic robotics for mobile robots which address the problem of generating maps from noisy and uncertain sensor measurement data, with the assumption that the robot pose is known.

The basic idea of the occupancy grid is to represent a map of the environment as an evenly spaced field of binary random variables each representing the presence of an obstacle at that location in the environment. Occupancy grid algorithms compute approximate posterior estimates for these random variables.

Earlier, 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), we need to 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. There is no explicit motion model defined in the case of mapping, i.e., we assume that the maps do not change with time and that calculations are only done on the basis of measurements. 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.

For our first attempt at a 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 labelled 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:

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

In truth, this is a very bad assumption to make since obstacles usually span multiple cells. However, this makes the problem tractable. In class, Drew mentions that this is more correctly called an

[1] Contect adapted from previous scribes: Victor Hwang, Nathan Brooks, Brian Coltin and Mehmet R. Dogar. Some wording is taken from Wikipedia

approximation rather than an assumption, because we are not assuming that cells are independent but rather we are approximating them as being independent.

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.

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

We then 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 assumption does not hold right here. When we talk about a single cell then given the current state previous observations do not tell us anything new about what we should observe now. In general however we do not have a single celled map. The reasons are described in more detail 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.

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

Intuitively, this tells us that if the probability of an event, say a door being open, is just as likely to remain open on observing a new sensor reading, then the reading is uninformative and does not change the posterior.

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.

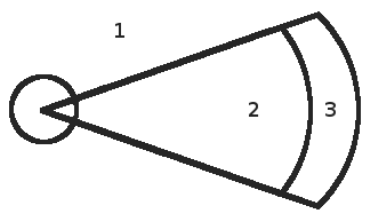

Figure 1: A sample sensor model for a laser scanner device which provides the probability a grid cell is occupied given a sensor reading. Grid cells in (1) would have probability equal to the prior, grid cells in (2) would have a low probability (since the beam does not return from those locations) and grid cells in (3) would have a high probability.

1.2 Problems with the Markov Assumption

At the beginning of the 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 in the case of a laser beam model: we can't say that an observation z_t is independent of all prior observations given only the state of a single cell, since the beam model necessarily couples observations by virtue of the beam passing through multiple cells (states).

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.

It should be noted, however, that this assumption is perfectly valid when the sensor directly observes the state, for instance in the case of a downward looking camera used for visual SLAM.

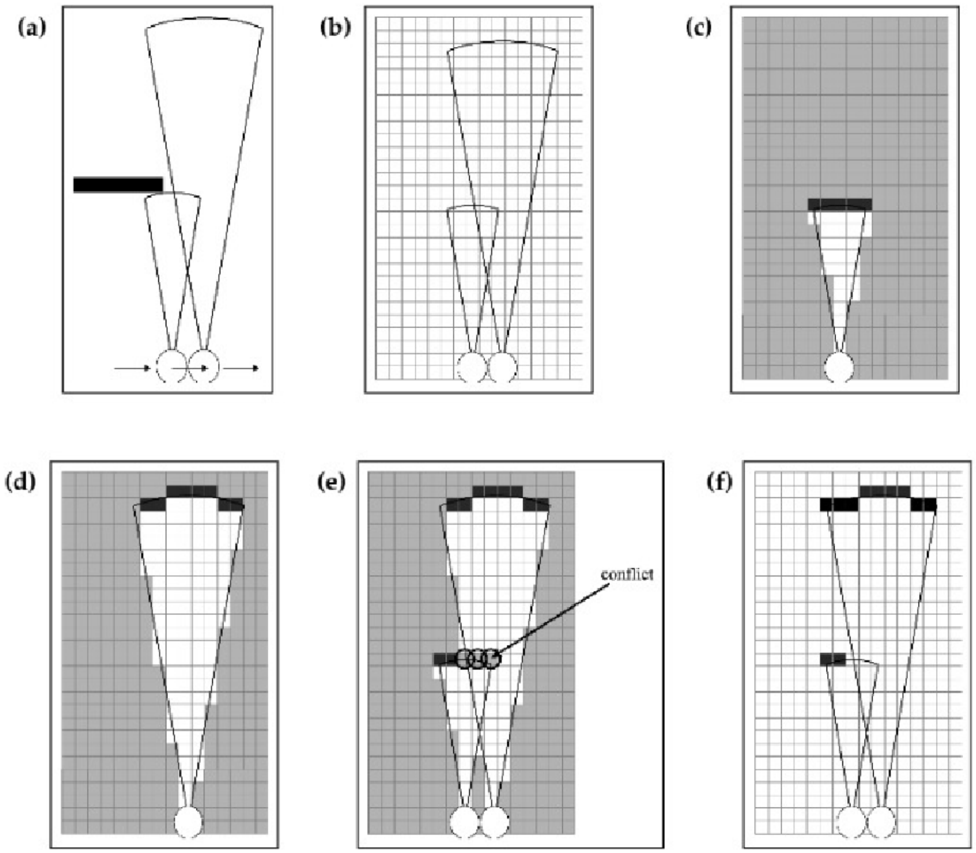

Figure 2: The problem with the standard occupancy grid mapping algorithm in Chapter 9.2 of Probabilistic Robotics: For the environment shown in Figure (a), a passing robot might receive the (noise-free) measurement shown in (b). The occupancy grid map approach maps these beams into probabilistic maps 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.

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.

Semi-transparent obstacles and Mitigation

Classical 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.

A possible solution to this problem would be to consider a more continous measure that measures the 'density' or probability of a beam to have reflected and not passed. A straightforward approach would be to treat the random variable as a biased coin and for each state keep a count of the number of hits and pass throughs. Thus, the only difference is that each state tracks two numbers. Then, the probability is empirically the ration of the hits to the sum of hits and passthroughs. For the inverse sensor model, we either spread the high probability zone (section 3 of Fig. 1) or use fractional number of hits and misses for each state.