Statistical Techniques in Robotics (16-831, F09) Lecture #3 (09/01/2009)

Sensor Models / Occupancy Mapping / Density Mapping

Lecturer: Drew Bagnell *Scribe: Mehmet R. Dogar*

1 Sensor Models

1.1 Beam Model of Range Finders

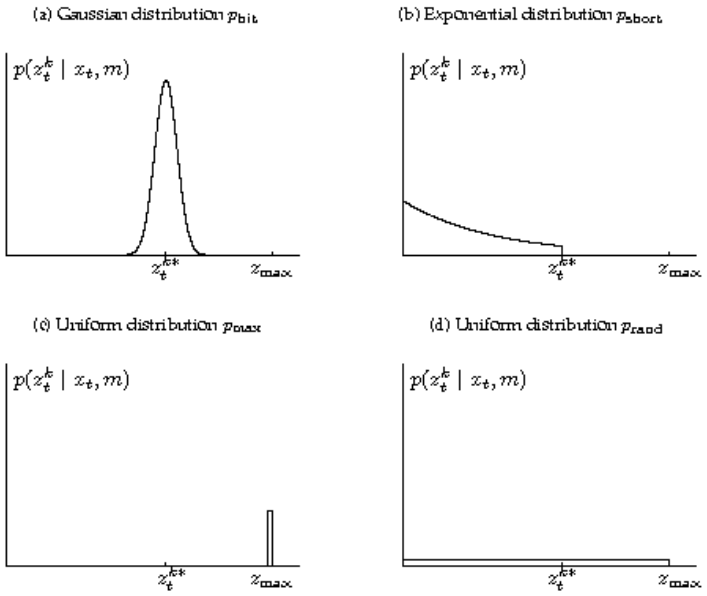

Figure 1: Components of the range finder sensor model.

The measurement model for range finders is a combination of different components:

- Correct range with local measurement noise (Fig. 1-a).
- Unexpected objects (Fig. 1-b).
- Failures to get any readings, e.g. specular reflections, sensing black, light-absorbing objects. (Fig. 1-c).
- Random measurements (Fig. 1-d).

These individual densities are combined we have a resulting mixture density that looks like Fig 2.

When we want to use our own range finder, we can collect data and then fit the data to the model and find the parameters for our sensor (Fig. 3).

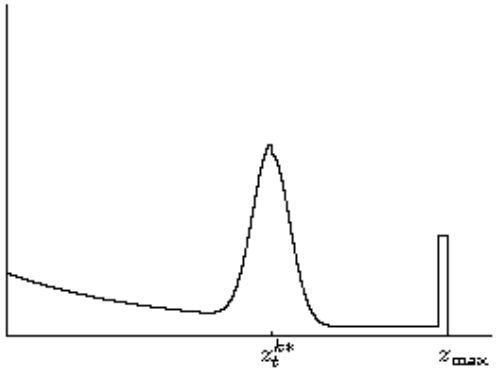

Figure 2: Resulting mixture density.

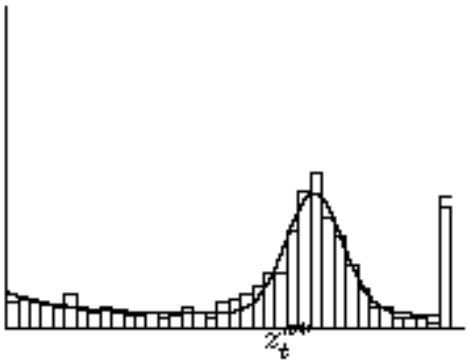

Figure 3: Approximation of the beam model based on real laser data.

In practice, when we are using this model we can do ray-tracing on our map, find the nearest obstacle, and use this to find the density function of our sensor for the given state. But all of these are for just one beam. As an example, a SICK laser range finder throws 181 beams in one cycle. How are we going to combine these?

We make independence assumption between the beams:

$$p(\vec{z}|x) = \prod_i p(z_i|x) \tag{1}$$

where $\vec{z}$ is the vector of all observations/beams, z_i's are individual beams, and x is the location of the robot for a given map.

But, *can the independence assumption between the beams be justified?*

This can be justified for random noise, but not always. E.g. reflectivity of surfaces, calibration errors make neighboring observations dependent on each other. [1]

[1] A similar argument can be made in the time domain. We generally make temporal independence assumptions, but there is actually temporal correlation in the observation data. E.g. a person walking in front of the robot will create a temporal obstacle.

How to fix this problem?

- If we really want to fix this problem, we shall make a new sensor model that associates beams close to each other.
- To "hack" the problem, we can:
  * Subsample the measurements: this makes us less susceptible to correlated noise in adjacent measurements.
  * Smooth the observation model: For example we can take the cube root of the model shown in Fig. 2 to smooth it. This method makes the measurements less informative. But still this is usually better than the first option since we are not completely throwing away any data.

In HW2 we will probably need to do something like this. If we use all sensor measurements directly the robot will get overconfident about its location. Therefore we will need to either throw away some data or oversmooth the sensor model to get better results.

2 Occupancy Mapping

Instead of doing localization, we will try to solve the opposite problem: Given the position of the robot, find the map of the world.

What will be the state x if we don't have a given map? The state will be the whole map!

We will partition the world into cells, with each being one of two states: filled or not. The individual grid cells are m_i, and $\vec{m}$ is the vector of all grid cells. Unfortunately, the curse of dimensionality prevents us from filtering $\vec{m}$. Instead, we will filter each cell independently, assuming that they are in fact independent. In truth, this is a very bad assumption to make, but it is practical.

This is called "occupancy mapping".

2.1 Derivation

Let x represent the state of grid cell m_i (essentially $x \leftarrow m_i$). In the following, also assume that x means 'the grid cell is empty'; and $\bar{x}$ means 'the grid cell is filled'.

$$p(x|z_{1:t}) = \frac{p(z_t|x, z_{1:t-1})p(x|z_{1:t-1})}{p(z_t|z_{1:t-1})} \tag{2}$$

We make the Markov assumption $p(z_t|x, z_{1:t-1}) = p(z_t|x)$ (This should worry us!). Hence:

$$p(x|z_{1:t}) = \frac{p(z_t|x)p(x|z_{1:t-1})}{p(z_t|z_{1:t-1})} \tag{3}$$

Doing the Bayes rule over $p(z_t|x)$ once again:

$$p(x|z_{1:t}) = \frac{p(x|z_t)p(z_t)}{p(x)} \frac{p(x|z_{1:t-1})}{p(z_t|z_{1:t-1})} \tag{4}$$

$$\tag{5}$$

Here we need to use a trick because we have many nasty terms that are hard to calculate. We will start by computing (by analogy) the likelihood for the opposite event $\bar{x}$

$$p(\bar{x}|z_{1:t}) = \frac{p(\bar{x}|z_t)p(z_t)}{p(\bar{x})} \frac{p(\bar{x}|z_{1:t-1})}{p(z_t|z_{1:t-1})}, \tag{6}$$

and then divide the two:

$$\frac{p(x|z_{1:t})}{p(\bar{x}|z_{1:t})} = \left[\frac{p(x|z_t)}{p(\bar{x}|z_t)}\right] \left[\frac{p(\bar{x})}{p(x)}\right] \left[\frac{p(x|z_{1:t-1})}{p(\bar{x}|z_{1:t-1})}\right] \tag{7}$$

The second term on the right may seem counter-intuitive, but if we rearrange the terms, it makes sense:

$$\frac{p(x|z_{1:t})}{p(\bar{x}|z_{1:t})} = \left[\frac{p(x|z_t)}{p(x)}\right] \left[\frac{p(\bar{x})}{p(\bar{x}|z_t)}\right] \left[\frac{p(x|z_{1:t-1})}{p(\bar{x}|z_{1:t-1})}\right] \tag{8}$$

If the probability of $\bar{x}$ decreases with the observation z_t: $p(\bar{x}) > p(\bar{x}|z_t)$, it causes the belief on x to increase relative to $\bar{x}$.

The next step is to take the log of (8) to get the log odds of the belief $bel_t(x)$, denoted as $l_t(x)$:

$$l_t(x) = l_{t-1}(x) + log\frac{p(x|z_t)}{p(x)} + log\frac{p(\bar{x})}{p(\bar{x}|z_t)} \tag{9}$$

Here we need only $p(x|z_t)$ and $p(x)$ to be specified, and we can get the others by subtracting these from 1.

An example z_t here can be pass-through / hit information.

The algorithm derived above uses 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.

2.2 Problems with this Derivation

At the beginning of this derivation we make the Markov assumption and say that: $p(z_t|x, z_{1:t-1}) = p(z_t|x)$. This would have been a reasonable assumption if x were to represent the state of the complete map. But we should not forget that x in occupancy map represents the state of a single grid cell. Hence, Markov assumption does not really make sense: we can not say that the observation data is independent of everything else if we know the state of a single cell. Fig. 4 presents an example case where this assumption can be problematic.

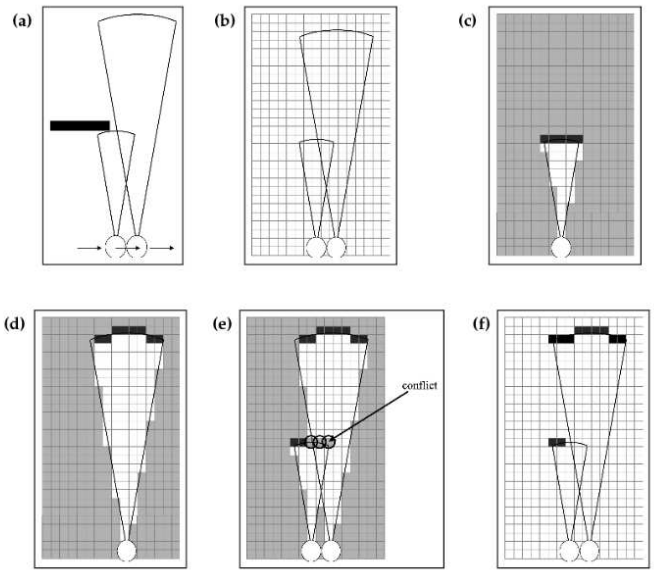

Figure 9.10 The problem with the standard occupancy grid mapping algorithm in Chapter **??**: For the environment shown in Figure (a), a passing robot might receive the (noise-free) measurement shown in (b). The factorial approach maps these beams into probabilistic maps separately for each grid cell 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.

Figure 4:

3 Density Mapping

In occupancy grid mapping every grid cell is either filled or empty. But there may be cases when we want a cell to be partially filled. This can happen in different ways; we may want the density of a cell to represent when some part of the grid cell is filled and the rest is empty; or when the objects that "fill" the grid cell have different characteristics: we may want a grid filled with tree branches to be "less filled" than a grid filled with a solid rock.

For this we use a real number between $[0, 1]$, representing the density of a grid cell. Beta distributions are usually used for this purpose:

$$Beta(\alpha_1, \alpha_2) \tag{10}$$

Some examples of Beta distributions are shown in Fig. 5.

One good way to parametrize the Beta distribution can be:

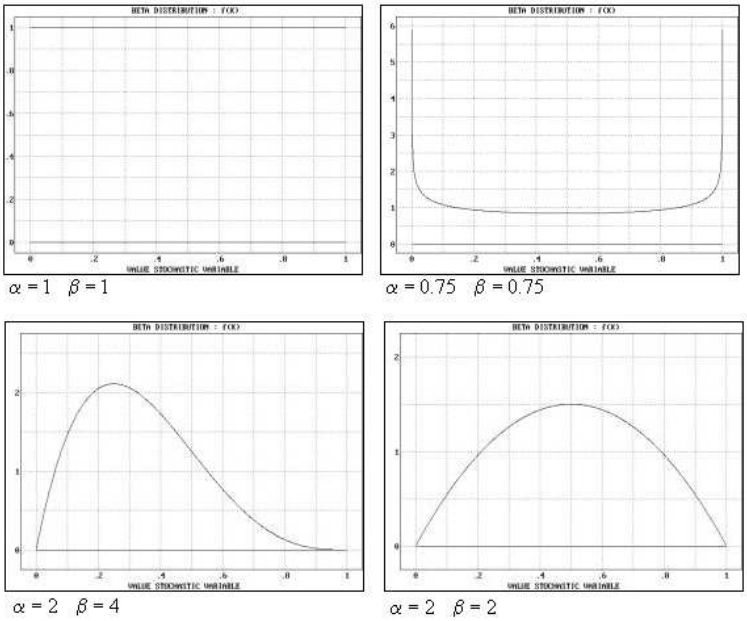

Figure 5:

$$\alpha_1 = \text{number of hits} \tag{11}$$

$$\alpha_2 = \text{number of pass-throughs} \tag{12}$$

The density of a cell can be represented as:

$$\frac{\alpha_1}{\alpha_1 + \alpha_2} \tag{13}$$