Algorithm for Calculating Grid Cell Occupancy Belief from 3d Point Data

Jeff Johnson

1 Problem and Proposed Solution

Given a static environment, how can you use noisy 3d point data to build a model of the environment? The approach is to sweep the environment systematically to generate many overlapping sets of 3d point data, then use those points to fill a grid cell map representation of the environment. A previously measured error function for the device in conjunction with some basic assumptions about the probabilistic and physical nature of the environment will then be used to compute occupancy beliefs for the grid cells. The following sections will provide some further detail about each of these steps.

1.1 The Environment

For this problem the environment is assumed to be static and finite, that is, it will not change over time, and it will have definite, observable boundaries that lie within the range of the measuring device. (Specifically, it will be the section of IE010 lab containing the robot arms.) It is further assumed that all objects in the environment are opaque.

1.2 Measuring the Environment

Measurement of the environment will be performed by mounting a range finding device to one of the robotic arms and performing a systematic scan of the environment by moving the arm so that the device measures the entire environment. The scans should significantly overlap in order to provide large redundancy in the data. The specifics of the measurement device, such as its range, error model, field of view, and current position and orientation in the world coordinate frame, are assumed to be known.

1.3 Representation of the Environment

The environment will be represented by a grid cell occupancy map, and its boundaries will lie at the end of the range of the measurement device. Each grid cell will keep track of its own occupancy belief, and a measure of density, which will incorporate the number of points measured within the cell, and their associated certainty beliefs. The grid cell size should be fine enough to reasonably reflect the resolution of the measuring device, but not so fine as to cause computational difficulties (the exact size will be determined empirically, but these will be the criteria).

1.4 Probabilistic and Physical Assumptions

Cell occupancy is assumed to be independent of occupancy of neighboring cells at the same depth. This is clearly not the case, but without an underlying model of the environment, computing this kind of conditional occupancy seems unworkable. That said, occupancy is assumed to be conditional on the cells lying on the same ray between the cell in question and the measuring device. For instance, if points are measured that would occupy a cell that lies directly behind a cell that is believed to be occupied, the belief occupancy of the cell in question is conditioned appropriately.

2 The Algorithm

The algorithm will start by assigning a belief to each pixel value. To do this, it assumes that the rate of change of z should be smooth in all directions in a one pixel neighborhood. Then it looks at derivatives of the z-value in opposing directions radiating from the center of a 3x3 window outward. For instance, the difference in z-value between the center pixel and the top-left pixel, and the center pixel and bottom-right pixel should mostly cancel (so long as the corner pixels are both subtracted from the center pixel). A sharp change in z across the window will increase the value of the error function, E.

$$\begin{aligned} E(u,v)^2 &= |(z_{1,1} - z_{0,0}) + (z_{1,1} - z_{2,2})|^2 \\ &+ |(z_{1,1} - z_{0,1}) + (z_{1,1} - z_{2,1})|^2 \\ &+ |(z_{1,1} - z_{0,2}) + (z_{1,1} - z_{2,0})|^2 \\ &+ |(z_{1,1} - z_{1,0}) + (z_{1,1} - z_{1,2})|^2 \end{aligned}$$

The square root of this value, which is E, in conjunction with the measured error distributions, can be used to assign a belief, b, to the point. Let h be a function that takes a value and returns a probability associated with that value from the PMF associated with the measured error histogram.

$$b(u, v) = h(E)$$

Now that the points have associated certainties, they are entered into a grid cell map of the environment. In doing so, a density, D, of each grid cell, i, is computed, which is the sum of the beliefs for each point in the cell.

$$D_i = \sum_{(u,v)\in i} b(u, v)$$

The density for a given cell accumulates with every iteration, and the individual point data are discarded. Next an opacity for each grid cell is calculated. The opacity, o_i, for a cell should be directly proportional to density, and should be lie in the range $[0, 1]$. It will be defined as the density divided by the number of pixels in the projection of the current grid cell in the current view, where if $o_i > 1$ then $o_i = 1$. (That should make intuitive sense: If the grid cell has more points than it's projection, it should be opaque.) Let f be a function that returns the area of the grid cell when projected onto the 2d imaging plane.

$$o_i = \begin{cases} 1 & : \frac{D_i}{f(i)} > 1 \\ \frac{D_i}{f(i)} & : \frac{D_i}{f(i)} \leq 1 \end{cases}$$

Translucency, t_i, for each cell, then, is defined as one minus the opacity.

$$t_i = 1 - o_i$$

Once translucency for cells has been calculated, an occlusion check occurs. For every cell in which point data is detected, the ray back to the measuring device is traced, and the cell's density is multiplied by the translucency of every intersecting cell. Let S be the set of cells intersected by the ray, and D'_i the updated density of cell i.

$$D'_i = D_i \prod_{j\in S} t_j$$

This should penalize grid cells that occur behind other cells in a way proportional to the accumulated weight of their densities.

The algorithm runs until there are no more data frames left, and the result is the model of the environment in the form of a grid cell occupancy map, where the density of the grid cells is proportional to the certainty of their occupancy.

3 Evaluation

There will be a ground truth measurement of the environment, and the grid cell size along with objects in the environment should be chosen such that the ground truth binary occupancy map represents as closely as possible the real world (that is, the cell size and object dimensions should be chosen so that the environment exactly fills the grid cells). The performance of the above algorithm will then be given in terms of the probability of the ground truth environment, given the representation generated. Probably this will involve generating the most like environment representation and comparing it to the ground truth. The exact mechanics of this have yet to be determined.

4 Next Step

The next step will be to use this grid cell map as the basis for object recognition. The task will be: Given this grid cell occupancy map and a list of known objects, identify any of the know objects in the map. It's conceivable that this could be done in conjunction with map construction, which would greatly aid in determining the likelihood of certain points.

5 Questions

1. When performing the evaluation of the would it be useful to generate a set of random perturbations of the ground truth, and then see how closely the generated model fits?
2. How can the occlusion check step be justified theoretically? Perhaps some kind of limit check that the erroneous readings go to zero?
3. Does this algorithm provide a good foundation for the planned next step?
4. Is the assumption that $\frac{d}{dz}$ is constant for each direction in a pixel window valid?
5. How should grid cell size be determined?
6. Could the opacity check exacerbate errors if there are a lot of erroneous points lying on the intersecting ray?
7. Is it problematic that the certainty of the grid cell occupancy is only proportional to density, and that there isn't an actual probability?