Albert-Ludwigs-Universität Freiburg — Institut für Informatik
Lecture: Introduction to Mobile Robotics
Summer term 2009 — Prof. Dr. W. Burgard

Sheet 5

Topic: Particle Filter and Monte Carlo Localization

Submission deadline: Tuesday 26.5.2009 (before class)

Exercise 1:

A range scan z consists of a set of K range measurements

$$z = \{z_1, \ldots, z_K\} .$$

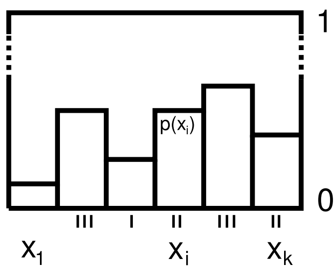
The probability $p(z \mid x, m)$ of a range scan given the pose x and map m is usually computed as

$$p(z \mid x, m) = \prod_{i=1}^{K} p(z_i \mid x, m) .$$

1. What is the underlying assumption behind this computation?
2. If the assumption is valid, explain why? Otherwise, explain what can be done to alleviate this?

Exercise 2:

Consider rejection sampling for a discrete probability distribution p: We are given k states $x_1, \ldots, x_k$ with associated probabilities $p(x_1), \ldots, p(x_k)$.



We will use N samples. Let $c(x_i) \in \{0, \ldots, N\}$ be the number of *accepted* (!) samples for state x_i. Prove that the expected probability mass $\tilde{p}(x_i) = \frac{E(c(x_i))}{\sum_{j=1}^{k} E(c(x_j))}$ assigned to state x_i by rejection sampling equals the true probability $p(x_i)$: $\forall i \in \{1, \ldots, k\} : \frac{E(c(x_i))}{\sum_{j=1}^{k} E(c(x_j))} = p(x_i)$.

Exercise 3:

Based on the odometry motion model presented in the lecture,

1. Write an *Octave* function that given two consecutive robot poses x_{t-1} and x_t computes the corresponding odometry command $u_t = \langle \delta_{rot_1}, \delta_{trans}, \delta_{rot_2} \rangle$

2. Write an *Octave* function that given a robot pose x_{t-1} and a odometry command u_t samples a new pose x_t. Use the following error parameters: $\alpha_1 = 0.2$, $\alpha_2 = 0.01$, $\alpha_3 = 0.2$ and $\alpha_4 = 0.01$.

3. Write an *Octave* program program that reads a file containing range scans (see exercise sheet 4 for a description of the file format). Use the functions implemented above to generate 250 samples for each odometry command computed from file *log2.log* (see exercise sheet 4). Plot all sampled poses.

Exercise 4:

Write an *Octave* program that reads a file containing range scans (see exercise sheet 4 for a description of the file format) and builds a likelihood field map. Additionally, write a function that given a likelihood field map m, a range scan z and a pose x, computes the likelihood $p(z \mid x, m)$ of scan z given pose x and map m.

A likelihood field map is a discrete grid that represents the environment where each cell in the grid contains the likelihood of an obstacle detection for the space represented by that cell.

To build a likelihood field map from a log file containing range scans, do the following:

1. Determine the size of the gird by first determining the size of the measured environment. Use a grid resolution of 0.1 meters (if you run out of memory, use a larger resolution).

2. Create an auxiliary grid where each cell indicates whether a range measurement ended in the space represented by that cell or not. If a range measurement ends in a cell, the space corresponding to that cell is considered to be occupied and the cell is called an "occupied cell".

3. Using the auxiliary grid, create a distance grid where each cell contains the distance to the nearest obstacle (that is, the distance to the nearest occupied cell in the auxiliary grid).

4. Use the distance grid to compute the likelihood field map. As sensor model use a simplified scan-based model that consists of a Gaussian distribution with mean at the distance to the closest obstacle and standard deviation $\sigma = 0.02m$.

Plot the likelihood field map for file *log2.log* (see exercise sheet 4).