Universiteit van Amsterdam

Autonomous Mobile Robots

Fourth assignment

Particle Filter Based Simultaneous Localization And Mapping

Ruben Janssen, 10252657
David van Erkelens, 10264019

Department of Computer Science
University of Amsterdam

December 15, 2013

All code published on
`www.github.com/David1209/AMR/assignment4`

Contents

1 Introduction

In order for an autonomous mobile robot to be able to navigate without a priori knowledge about the map, it has to determine its position whilst building the map. This can be accomplished by using the Simultaneous Localization and Mapping (SLAM) algorithm. One of the variations of SLAM is FastSLAM, a particle filter based algorithm which has a better performance in real time map building than most other variations. FastSLAM scales logarithmically with the number of landmarks in the map, and is therefore much more suitable for real-time applications.

2 Setup

A NXT robot has been equipped with an omnidirectional camera. Using this NXT, a dataset of images has been made instead of a live video feed. Eventhough using the dataset is less accurate, the FastSLAM algorithm should be able to compensate for this. A basic environment has been taped out, and from the point of view of the robot some snapshots have been made.

insert picture of environment here

iets uitgebreider? wat?

3 FastSLAM

To implement FastSLAM, a pre-recorded dataset is used. The input of the SLAM algorithm consists of odometry data and observable features. Here, corners are used as landmarks, detected as endpoint segments of lines extracted from data using the onmidirectional camera. The corner detection algorithm can be reduced to two steps: line extraction and endpoint collection.

3.1 Line extraction

A snapshot from the onmidirectional camera describes a 2D slice of the environment. Points in such a snapshot are specified in the polar coordinate system $(pi, \theta i)$, with the origin at the location of the camera. In order to fit

a line through the points, a line is expressed in polar coordinates:

$$x \cos \alpha + y \sin \alpha = r \tag{1}$$

In which $-\pi < \alpha < \pi$ is the angle between the line and the x-axis, $r >= 0$ is the distance of the line to the origin and (x, y) are the Cartesian coordinates of the point lying on the plane.

In order to identify the best line segments for the obtained measurements, the Split-and-Merge algorithm is employed.

3.2 Endpoint collection

3.3 Parameters

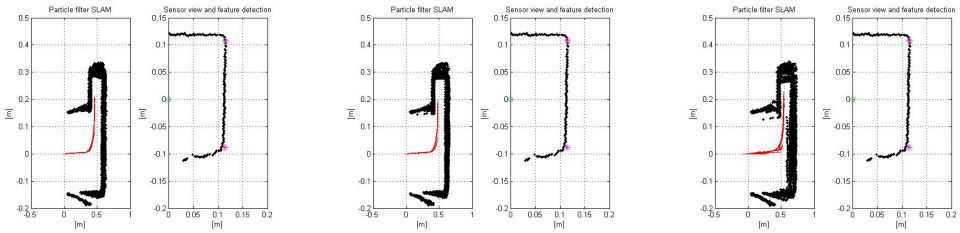
The FastSLAM approach estimates the pose of the robots and the features using a particle filter. Each particle represents a possible robots position and its map of the environment. A few parameters of the SLAM algorithm can be tweaked, in order to improve performance of the algorithm:

1. **The number of particles** - saved as `PARAMS.NPARTICLES`: this influences the number of particles in the cloud. More particles gives an higher chance of a particle representing the exact position of the robot, but requires more computational power and also introduces more wrong particles.

2. **The variance of the odometry** - saved as `sigmaX` and `sigmaTH`: this indicates the possible error value for the odometry of the robot. `sigmaX` indicates the error in the speed, while `sigmaTH` indicates the error in rotation. Higher values for these parameters make the size of the particle cloud bigger, and therefore the locations of the walls in the map less certain.

3. **The variance of the range sensor** - saved as `sigmaR` and `sigmaB`: this indicates the possible error value for the range sensor. Higher values for these parameters make the locations of the walls less certain.

Results of tweaking the parameters can be seen in figure 1.

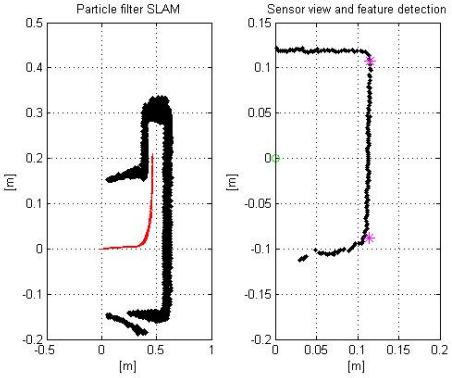


(a) Normal result (b) More particles (c) Using a higher σ

Figure 1: Different parameters

4 Experiments

When running the algorithm on the data set provided, it returned the following result:



5 Results

6 Conclusion