Universiteit van Amsterdam

Autonomous Mobile Robots

Third assignment

Probabilistic Pose Estimation based on a Topological Map

Ruben Janssen, 10252657

David van Erkelens, 10264019

Department of Computer Science
University of Amsterdam

December 8, 2013

Contents

1 Introduction

In order for an autonomous mobile robot to drive and navigate, it requires some method by which to determine its position. This is called the localization problem. One of the methods to solve this problem is by using probabilistic pose estimation based on a topological map. The robot tries to estimate its position with respect to known locations in the environment. This can be achieved by comparing its current position to known locations and estimating which locations is most similar. The most similar location is probably the current location of the robot.

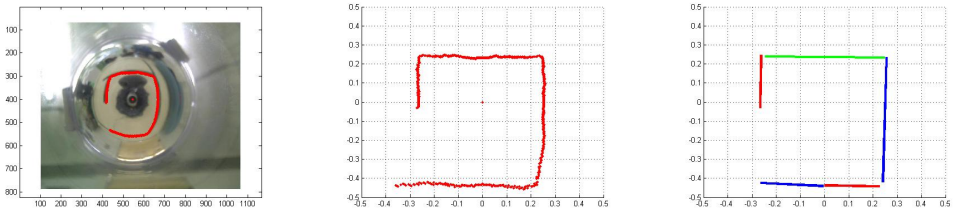
2 Setup

In order to compare the robots current position with that of known locations, the robot is provided a training set of pictures. These pictures are made in different positions in the training area by using an omnidirectional camera. Based on lines and blobs detected in the pictures, the robot can compare the lines and blobs detected in the picture taken at the current position to the provided pictures. Unfortunately however, not enough robots were available to test with. Instead, the software compares the known locations to a set of given photos rather than to a live video feed.

3 Implementation

3.1 Detecting Lines

Before pictures can be compared, they have to be analysed. First, the walls are detected around the robot (figure 1a). Second, the detected walls are mapped on a grid corresponding with their distance from the vehicle (figure 1b). Third, the lines are extracted from the grid (figure 1c). pints, grid,



(a) Detecting walls (b) Distance grid (c) Forming lines

Figure 1: Detecting lines

lines.jpg

The code to determine points and calculate the distance grid are provided. To extract the lines from the distance grid, a simpified version of the Split & Merge algorithm is used.

GetLinePattern.m

3.2 Comparing Lines

lineschance.jpg checkLinePattern.m

3.3 Detecting Blobs

blobs.jpg GetBlobPattern.m

3.4 Comparing Blobs

blobschance.jpg checkBlobPattern.m

3.5 Combining Blobs and Lines

4 Conclusion

dd