



UNIVERSITEIT VAN AMSTERDAM

AUTONOMOUS MOBILE ROBOTS

THIRD ASSIGNMENT

---

PROBABILISTIC POSE ESTIMATION BASED ON A TOPOLOGICAL MAP

---

RUBEN JANSSEN, 10252657

DEPARTMENT OF COMPUTER SCIENCE  
UNIVERSITY OF AMSTERDAM

DECEMBER 8, 2013

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Setup</b>	<b>2</b>
<b>3</b>	<b>Implementation</b>	<b>3</b>
3.1	Detecting Lines . . . . .	3
3.2	Comparing Lines . . . . .	3
3.3	Detecting Blobs . . . . .	3
3.4	Comparing Blobs . . . . .	3
3.5	Combining Blobs and Lines . . . . .	3
<b>4</b>	<b>Conclusion</b>	<b>3</b>

# 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 most 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 (1). Second, the detected walls are mapped on a grid corresponding with their distance from the vehicle (2). Third, the lines are extracted from the grid (3). pints, grid, lines

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

### **3.2 Comparing Lines**

### **3.3 Detecting Blobs**

### **3.4 Comparing Blobs**

### **3.5 Combining Blobs and Lines**

## **4 Conclusion**

dd