



University  
of Glasgow | School of  
Computing Science

**Title of project placed here**

Name of author placed here

School of Computing Science  
Sir Alwyn Williams Building  
University of Glasgow  
G12 8QQ

A dissertation presented in part fulfilment of the requirements of the  
Degree of Master of Science at The University of Glasgow

Date of submission placed here

## **Abstract**

abstract goes here

## Education Use Consent

I hereby give my permission for this project to be shown to other University of Glasgow students and to be distributed in an electronic format. **Please note that you are under no obligation to sign this declaration, but doing so would help future students.**

Name: \_\_\_\_\_ Signature: \_\_\_\_\_

## **Acknowledgements**

acknowledgements go here sdfsf

# Contents

<b>1</b>	<b>Introduction</b>	<b>6</b>
1.1	Context . . . . .	6
1.2	Problem Objectives . . . . .	6
1.3	Problem Statement . . . . .	6
1.4	Application/ Motivation . . . . .	6
1.5	Outline of the dissertation . . . . .	6
<b>2</b>	<b>Background Literature Survey</b>	<b>7</b>
2.1	Papers? - if required . . . . .	8
2.2	ROS . . . . .	8
2.3	Turtlebot . . . . .	8
2.4	Cameras . . . . .	8
2.4.1	RGB-D . . . . .	8
2.4.2	Stereo . . . . .	8
2.5	SLAM . . . . .	8
2.5.1	RtabMap . . . . .	8
2.5.2	Others . . . . .	8
2.6	Frontier Exploration . . . . .	8
2.7	Object Detection . . . . .	8
2.7.1	Feature Detection . . . . .	8
2.7.2	Tensorflow . . . . .	8

2.7.3	Haar Cascades . . . . .	8
2.7.4	Google Vision API . . . . .	8
2.7.5	3D Detection . . . . .	8
<b>3</b>	<b>Requirements</b>	<b>9</b>
<b>4</b>	<b>System Design</b>	<b>10</b>
<b>5</b>	<b>System Implementation</b>	<b>11</b>
5.1	Mapping . . . . .	11
5.1.1	Transforming data . . . . .	11
5.1.2	Calibration . . . . .	11
5.2	Frontier Exploration . . . . .	11
5.3	Object Detection and Recognition . . . . .	11
5.3.1	Blob Detection . . . . .	11
5.3.2	Detecting Clusters - different methods . . . . .	11
5.3.3	Creating Boxes . . . . .	11
5.3.4	Tracking Boxes . . . . .	11
5.3.5	Positioning Boxes in Map/Loop Closure . . . . .	11
5.3.6	Recognising Objects . . . . .	11
5.3.7	Publishing to Rviz/Rtabmap . . . . .	11
5.4	The whole package - how to utilise . . . . .	11
<b>6</b>	<b>Evaluation</b>	<b>12</b>
6.1	Testing . . . . .	12
<b>7</b>	<b>Conclusion</b>	<b>13</b>
7.0.1	Future work . . . . .	13
<b>A</b>	<b>First appendix</b>	<b>14</b>
A.1	Section of first appendix . . . . .	14



# **Chapter 1**

## **Introduction**

### **1.1 Context**

### **1.2 Problem Objectives**

### **1.3 Problem Statement**

### **1.4 Application/ Motivation**

### **1.5 Outline of the dissertation**



## **Chapter 2**

# **Background Literature Survey**

Divide into hardware and software/implementation technology Do I need to discuss things I haven't used Existing/similar applications

## **2.1 Papers? - if required**

## **2.2 ROS**

## **2.3 Turtlebot**

## **2.4 Cameras**

### **2.4.1 RGB-D**

### **2.4.2 Stereo**

## **2.5 SLAM**

### **2.5.1 RtabMap**

### **2.5.2 Others**

## **2.6 Frontier Exploration**

## **2.7 Object Detection**

### **2.7.1 Feature Detection**

### **2.7.2 Tensorflow**

### **2.7.3 Haar Cascades**

### **2.7.4 Google Vision API**

### **2.7.5 3D Detection**

## **Chapter 3**

# **Requirements**

## **Chapter 4**

# **System Design**

## **Chapter 5**

# **System Implementation**

### **5.1 Mapping**

#### **5.1.1 Transforming data**

#### **5.1.2 Calibration**

### **5.2 Frontier Exploration**

### **5.3 Object Detection and Recognition**

#### **5.3.1 Blob Detection**

#### **5.3.2 Detecting Clusters - different methods**

#### **5.3.3 Creating Boxes**

#### **5.3.4 Tracking Boxes**

#### **5.3.5 Positioning Boxes in Map/Loop Closure**

#### **5.3.6 Recognising Objects**

#### **5.3.7 Publishing to Rviz/Rtabmap**

### **5.4 The whole package - how to utilise**

## **Chapter 6**

# **Evaluation**

### **6.1 Testing**

## **Chapter 7**

# **Conclusion**

### **7.0.1 Future work**

## **Appendix A**

### **First appendix**

#### **A.1 Section of first appendix**



## **Appendix B**

### **Second appendix**

# **Bibliography**