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 sdfsdf

Contents

Chapter 1

Introduction

I completed a program which allowed a turtlebot to autonomously map an environment using SLAM as well as create an inventory of items found in the environment.

1.1 Importance/Context/Motivation

Mobile robots are becoming more and more affordable and accessible which has allowed developers to take advantage of their applications in many different ways.

SLAM has it's application in rescue robots which http://www.aaai.org/Pressroom/Releases/release-02-0910.php these robots required realtime control and utilised only video streams to identify and rescue people.

This compares to this which utilises rugged mobile robots to create SLAM maps of mine shafts with minimal supersvision. This can then be applied to areas which are too unsafe/ small for humans to access. https://miningrox.informatik.tu-freiberg.de/en/

More affordable Drones can also be used to increasingly accurate create maps of property as well

Object detection can be used to detect humans via heat sensors etc. Aswell as identifying bombs etc.

1.1.1 Sketch of how I achieved it

1.1.2 Objectives/Hypothesis Karl Popper/Problem statement

1.1.3 Description of Objectives

1.2 Outline of the dissertation

Chapter 2

Background

Existing/similar applications

Include Photos of everything

2.1 Papers

Semantic Slam

2.2 Robot Operating System (ROS)

ROS provides a meta-operating system to for robots to use which includes an abstraction of hardware, low level controllers, message passing and package management.

This means that ROS is generally independent of hardware specifics and has standard implementations in well known programming languages such as Python and C++.

Helps encourage code reuse through engineering principles of loose coupling which has helped provide a vibrant ecosystem of developers and researchers working on ROS applications.

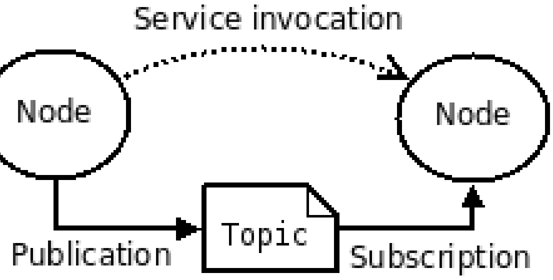
ROS includes a powerful simulator in the form of Gazebo which allows whole applications to be written without use of a physical robot.

How ROS works?

Nodes are modular processes which can communicate with each other through messages which are data structures supporting typed fields such as primitives or arrays. These messages are transported via topics which can be subscribed or published to.

For example one node can perform camera data processing while another one can control the robot's movement. These nodes can be publishing many messages to different topics, such as

Figure 2.1: Illustration of the ROS concepts



pointcloud messages to /camera/depth/points. The movement node can be subscribed to the /camera/depth/points topic and process it's messages.

Coordinate transformation.

2.3 Mobile Robots

2.3.1 Sensors

2.4 Turtlebot

2.5 Cameras

2.5.1 RGB-D

2.5.2 Stereo

2.6 SLAM

2.6.1 RtabMap

2.6.2 Others

2.7 Frontier Exploration

2.8 Object Detection

At low levels of the evolutionary scale the eye acts as a type of goal detector rather than a camera.

The information provided by past experience have a greater say on how a scene has been interpreted than immediate information provided by external organs. The eye the brain the computer p208

However in contrast to simple organisms whose focus is on detection resulting in a sensitive and broad field of vision the programme must have the ability to classify which requires precise resolution when required.

Everything is seen for the first time. Our sensory systems does not keep telling us things we already know as most important environmental information stays constant, this process is called adaption and happens not only at a retinal level but at higher levels in the brain.

What different types of neural structure are used to extract information from a visual signal

1) areas of high level of features 2) depth/closeness 3) shapes/edge detection - particularly light and dark

Not feasible to search for objects at all locations in a visual field Partioning or perceptual organisation.

Law of proximity = Stimulus that are close together are percieved to be a group.

Geometric, Photometric modeling scene degmentation, naming+labeling,

Image matching: correlation approach feature matching approach relational matching approach

2.8.1 Feature Detection

2.8.2 Deep Learning

Tensorflow

Haar Cascades

Google Vision API

3D Detection

2.8.3 Summary

Chapter 3

Approach/Implementation

Describe what I did and How

What problems I encounted etc.

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