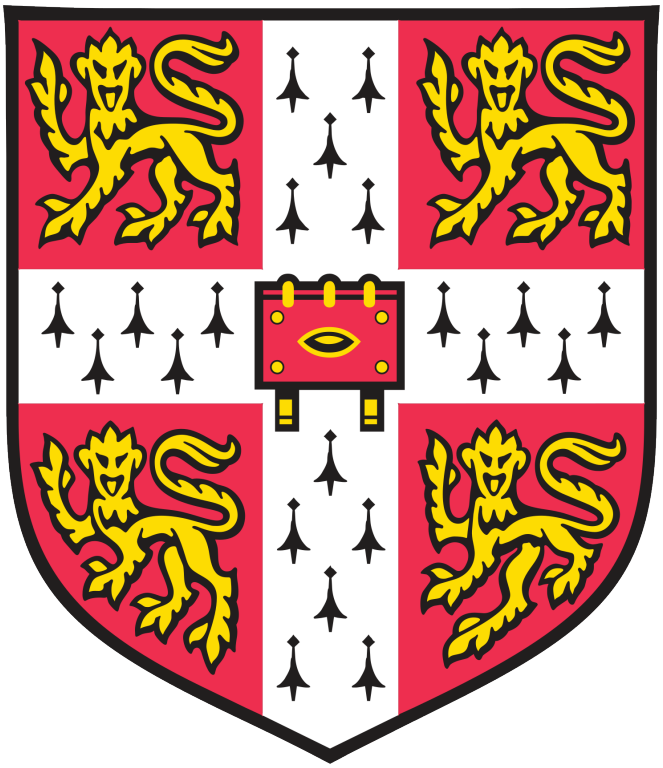
Joshua Bird

Distributed Visual Simultaneous Localization and Mapping



Computer Science Tripos – Part II
Queens' College

February 20, 2024

Declaration

I, Joshua Bird of Queens' College, being a candidate for Part II of the Computer Science Tripos, hereby declare that this dissertation and the work described in it are my own work, unaided except as may be specified below, and that the dissertation does not contain material that has already been used to any substantial extent for a comparable purpose. In preparation of this dissertation I did not use text from AI-assisted platforms generating natural language answers to user queries, including but not limited to ChatGPT. I am content for my dissertation to be made available to the students and staff of the University.

Signed: Joshua Bird
Date: February 20, 2024

Proforma

Candidate number:	⟨CANDIDATE NUMBER⟩
Project Title:	**Distributed Visual Simultaneous Localization and Mapping**
Examination:	**Computer Science Tripos – Part II, 2024**
Word Count:	⟨WORD COUNT⟩[1]
Code Line Count:	⟨LINE COUNT⟩[2]
Project Originator:	Joshua Bird, Jan Blumenkamp
Project Supervisor:	Jan Blumenkamp

Original Aims of the Project

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Fusce ac turpis quis ligula lacinia aliquet. Mauris ipsum. Nulla metus metus, ullamcorper vel, tincidunt sed, euismod in, nibh.

Work Completed

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Fusce ac turpis quis ligula lacinia aliquet. Mauris ipsum. Nulla metus metus, ullamcorper vel, tincidunt sed, euismod in, nibh.

Special Difficulties

None.

[1] This word count was computed using `texcount`.

[2] This code line count was computed using `cloc` (excluding autogenerated test output).

Contents

Chapter 1

Introduction

Visual Simultaneous Localization and Mapping (visual SLAM) serves as the foundation of countless modern technologies, with self-driving cars, augmented reality devices and autonomous drones just being a few examples. By using purely visual inputs, visual SLAM is able to create a 3D map of the surroundings while also localizing the camera's position within this map in real time.

Unlike other SLAM systems which may use expensive and heavy sensor such as LIDAR, RGB-Depth cameras or Radar, visual SLAM only requires the ubiquitous camera sensor, allowing the technology to be used in many commercial applications such as Google's ARCore[1], Boston Dynamic's GraphNav system used on their robot Spot[2], and several models of DJI quadcopters[3] making this a particularly practical field of research.

1.1 Relevant Work

While there have been many advanced implementations of visual SLAM over the last decade, very few of them have focused on distributed multi-agent systems, instead focusing on single-agent or centralized multi-agent systems. ORB-SLAM3 [1] is perhaps the most popular single-agent SLAM system, and often ranks at the top of benchmarks in a variety of environments [2], however it is fundamentally a single-agent system with no considerations in place for collaboration among peers. Out of the few multi-agent systems that do exist, the majority of them require a centralized server to manage communications between the agents and perform map merging, such as CCM-SLAM(2019) [3] and COVINS(2021) [4] among many others.

todo: add comparison table?

1.2 Motivation

Multi-robot systems are only becoming more common as automation in numerous fields continues to grow, such as self-driving cars, drone swarms and warehouse robots. These systems require the agents to understand the world around them, as know each other's locations within that world for collision avoidance. This task is often achieved with technologies such as GPS or motion capture setups, however we can not assume that all environments will have access to these systems. A few emerging examples include:

[1] https://developers.google.com/ar/develop/fundamentals
[2] https://support.bostondynamics.com/s/article/GraphNav-Technical-Summary
[3] DOI: 10.1016/j.vrih.2019.09.002

- Search and rescue operations in large indoor systems, assisted by drone swarms.
- Self-driving cars in underground road networks.
- Multi-agent cave/subsea exploration.

These are scenarios where multi-agent SLAM is able to assist, as it enables us to build a map of an unknown environment and keep agents aware of their relative poses. However, as noted in section 1.1, the majority of existing multi-agent visual SLAM implementations are centralized systems, requiring the agents to maintain reliable communications with the central server in order to operate. This is somewhat counterintuitive, as environments which don't have access to GPS or motion capture systems are more than likely to also have very poor communication channels – greatly limiting the use cases of these centralized multi-agent SLAM systems.

Naturally, this leaves us with distributed multi-agent visual SLAM systems which do not rely on a centralized management server, allowing the agents to be used in environments where network infrastructure may be lacking. Instead of a central node, the agents are able to communicate peer-to-peer when they come within close proximity with one another.

It is easy to see the broad reaching uses cases this opens up. Agents will be able to explore sections of the world independently or in small teams, sharing new world locations with their peers as they come into communication range using an ad-hoc network. Agents will only know their peer's locations when they are within communication range, but this is sufficient if we only need the relative position for collision avoidance (as is common in multi-robot systems).

My project implements a distributed multi-agent visual SLAM system, where agents are able to share information with each other to enable more accurate localization, relative positioning, and collaborative map building.

1.3 Project Overview

In this project, I:

1. Design and implement a distributed multi-agent visual SLAM system that is capable of localization, relative pose estimation and collaborative mapping, all while being tolerant to degraded network conditions and not reliant on any single leader agent.

2. Create a simulation environment for testing and evaluating my system locally.

3. Evaluate the performance of my system on standardized datasets, comparing its performance to other visual SLAM systems.

4. TODO: Deploy my system on physical robots, demonstrating the practical use cases of this system and benchmarking real world performance.

5. TODO: Map compression algorithms?

Chapter 2

Preparation

2.1 Starting Point

As noted in the *Relevant Work* section, visual SLAM systems are a mature and well researched subfield of Computer Science with many advanced implementations. To avoid spending the majority of my time re-implementing a visual SLAM system from scratch, I instead used a **single-agent** visual SLAM implementation as the starting point for my project. This has allowed me to focus my efforts on the distributed multi-agent aspect of my project, which I believe is novel and under-researched aspect in the field. Furthermore, by using a cutting edge single-agent SLAM system as a foundation for my project, I have been able to create a distributed SLAM system that is accurate and performant enough to have real-world use cases.

At the time of submitting my project proposal, I had forked the ORB-SLAM3 [1] git repository[1] and made minor changes to the codebase to allow it to compile on my machine. ORB-SLAM3 is licensed under GPL-3.0, and as such, I have open sourced my code under the same license[2].

I have no prior experience working with SLAM systems, but I have done research on the current state of multi-agent visual SLAM systems to evaluate the feasibility of my project and to try to prevent it being a duplication of past work.

2.2 Visual SLAM Background

Before developing a distributed multi-agent SLAM system, we must first understand the basics of a visual SLAM. This is a topic on which numerous books [5] and research papers [6] have discussed in depth, which I will attempt to summarize here.

2.3 Development Tools & Frameworks

2.3.1 Robot Operating System 2

Robot Operating System (ROS) 2 is a popular development framework (not operating system!) used in robotics applications both in research and industry. ROS's primary purpose is to provide a middleware to facilitate reliable communication between independent processes, called *nodes*.

[1] `https://github.com/UZ-SLAMLab/ORB_SLAM3`

[2] `https://github.com/jyjblrd/part_II_project`

These nodes can be on the same device or on a device within the local area network, and may be written in C++ or Python. Nodes communicate by *publishing* and *subscribing* to different *topics*.

This is best illustrated with an example. Below is a toy distributed SLAM system. Given agents $\{\texttt{agent}_n \mid n \in \{1, 2\}\}$, each agent has a camera which publishes to the $\texttt{/agent}_n\texttt{/camera}$ topic. The $\texttt{SLAM\_Processor}_n$ node subscribes to the $\texttt{/agent}_n\texttt{/camera}$ topic, and performs simultaneous localization and mapping using the image stream. The $\texttt{SLAM\_Processor}_n$ node then publishes to the $\texttt{/agent}_n\texttt{/new\_map\_data}$ topic, which the other agent can subscribe to and use to improve to their local map.

todo: add diagram

Since every node is abstracted away behind the interface provided by the various topics, we can easily swap out nodes in this system. For example, we can easily substitute the real camera for a simulated camera to test our system in a virtual environment without having to change any other part of our system. This makes transitioning between the real and simulated world almost seamless, which I knew would be essential for this project as all the testing would be conducted in a simulated environment.

Furthermore, using the ROS framework allows my code to be far more portable, as anyone can download my nodes, link the camera topics up to their robot's camera, and run my SLAM system. This turns my project from simply being a nice codebase to something that anyone can take and run on their own robots.

There are two versions of ROS: ROS 1 and ROS 2. ROS 2 has slightly less software support than ROS 1, but I chose to use it due to its better decentralized properties, which align with the goals of this project. ROS 2 conforms to the Data Distribution Service (DDS)[1] specification, which guarantees a reliable broadcast and, unlike ROS 1, it does not require a leader node when used in a multi-agent setup.

2.3.2 Webots Simulator

2.4 Datasets

2.5 Requirements Analysis

[1] `https://en.wikipedia.org/wiki/Data_Distribution_Service`

Chapter 3

Implementation

Implementation...

3.1 Implementation Section 1

Implementation Section 1...

Chapter 4

Evaluation

Evaluation...

4.1 Evaluation Section 1

Evaluation Section 1...

Chapter 5

Conclusions

Conclusions...

5.1 Conclusions Section 1

Conclusions Section 1...

Bibliography

[1] Carlos Campos et al. "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM". In: *IEEE Transactions on Robotics* 37.6 (2021), pp. 1874–1890.

[2] Dinar Sharafutdinov et al. "Comparison of modern open-source visual SLAM approaches". In: *CoRR* abs/2108.01654 (2021). arXiv: 2108.01654. URL: https://arxiv.org/abs/2108.01654.

[3] Patrik Schmuck and Margarita Chli. "CCM-SLAM: Robust and efficient centralized collaborative monocular simultaneous localization and mapping for robotic teams". In: *Journal of Field Robotics* 36.4 (2019), pp. 763–781.

[4] Patrik Schmuck et al. *COVINS: Visual-Inertial SLAM for Centralized Collaboration*. 2021. arXiv: 2108.05756 [cs.RO].

[5] Xiang Gao and Tao Zhang. *Introduction to visual SLAM: from theory to practice*. Springer Nature, 2021.

[6] Hugh Durrant-Whyte and Tim Bailey. "Simultaneous localization and mapping: part I". In: *IEEE robotics & automation magazine* 13.2 (2006), pp. 99–110.

Appendix A

⟨APPENDIX A NAME⟩

Appendix A...

Appendix B

⟨APPENDIX B NAME⟩

Appendix B...

Appendix C

Proposal

Proposal...