AUTONOMOUS MOBILE ROBOT WITH ORB-SLAM 2

BENAHMED SALAH EDDINE

A thesis submitted in partial fulfilment of the requirement for the award of the
Degree of Master in Electrical Engineering

Faculty of Electrical and Electronic Engineering
University Tun Hussein Onn Malaysia

AUGUST 2021
There are some people to whom I am greatly indebted. Therefore, I would like to dedicate this thesis to:

My parents

My father, Mr Benahmed Mohamed and my mother, Mrs TAIBI SOUHILA for their countless support, care, sacrifice, motivation and encouragement in my education.

And to all my two sisters.
ACKNOWLEDGEMENT

In the name of Allah, the Most Gracious and the Most Merciful. Salawat and salaam are rendered to the beloved Prophet Muhammad SAW and his family and all of his companions. Alhamdulillah, all praises to Allah (SWT) for the strengths and blessing in completing this thesis.

I would like to express my sincere gratitude to my advisor DR. Abu Ubaidah bin Shamsudin for his continuous support, patience, motivation and immense knowledge. The guidance has helped me throughout the time of research and writing of this thesis. Besides of my advisors, I would like to thank the rest of my thesis committee. Additionally, I would like to thank my internal evaluators during the course of this study for their insightful comments and encouragement and questions which have incanted me to widen my research from various perspectives.

Last but not least, I would like to thank my family: my parents, brothers, sisters for supporting me spiritually throughout the writing of this thesis and my life in general.
In this thesis we introduced ORB-SLAM2. It is critical for a mobile robot to be able to identify revisited locations or loop closures while conducting Simultaneous Localization and Mapping (SLAM) in order to be successful during autonomous navigation. When creating maps, it has been determined that one of the most difficult data association problems is loop closure. It is an effective method of eliminating mistakes and increasing the precision of the robot's localization and mapping capabilities. For the purpose of resolving the loop closure issue, the ORB-SLAM method is used, which is a feature-based simultaneous localization and mapping system that works in real time. This system incorporates loop closure and relocalization, as well as the ability to do automated initialization.

The monocular cameras are used to test the algorithm's performance in order to ensure that it is working properly. An important goal of this thesis is to demonstrate the accuracy of the relocalization and loop closure processes while utilizing the ORB SLAM2 algorithm in a range of different environmental conditions. The effectiveness of relocalization and loop closure in a variety of difficult indoor settings is shown via the use of a variety of experiment. According to the results of the studies, the monocular SLAM provides an accurate outcome in the interior environment. The ORB-SLAM 2 findings show the usability of the technique for autonomous navigation and future automated vehicles equipped with a low-cost monocular camera. The ORB-SLAM 2 results also demonstrate the usability of the approach for future driverless vehicles.

Keywords: ORB-SLAM2; keyframe; map points localization, relocalization
ABSTRAK


Kata kunci: ORB-SLAM2; kerangka utama; peta menunjukkan penyetempatan, penempatan semula
# CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>TITLE</td>
<td>i</td>
</tr>
<tr>
<td>DECLARATION</td>
<td>ii</td>
</tr>
<tr>
<td>DEDICATION</td>
<td>iii</td>
</tr>
<tr>
<td>ACKNOWLEDGEMENT</td>
<td>iv</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>v</td>
</tr>
<tr>
<td>ABSTRAK</td>
<td>vi</td>
</tr>
<tr>
<td>CONTENTS</td>
<td>ix</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>xi</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>xiii</td>
</tr>
<tr>
<td>LIST OF SYMBOLS AND ABBREVIATIONS</td>
<td>xv</td>
</tr>
<tr>
<td>LIST OF APPENDICES</td>
<td>xvi</td>
</tr>
</tbody>
</table>

## CHAPTER 1 INTRODUCTION

1.1 Background of study  
1.2 Visual SLAM  
1.3 Problem Statement  
1.4 Objectives of the study  
1.5 Scope of the study  
1.6 Thesis outline

## CHAPTER 2 LITERATURE REVIEW

2.1 Introduction  
2.2 Related work
CHAPTER 3 METHODOLOGY

3.1 Introduction 17
3.2 Flow chart 17
3.3 Keywords 18
3.4 Algorithm part 19
  3.4.1 Tracking 19
  3.4.2 Local Mapping 22
  3.4.3 Loop closing 24
3.5 Installation of utilities of ORB-SLAM2 27
  3.5.1 Configuration process 27
3.6 ROS installation 29
3.7 Installation of ORB-SLAM2 31
3.8 Running and testing in real time ORB-SLAM2 35
3.9 Summary 37

CHAPTER 4 RESULTS AND DISCUSSION

4.1 Introduction 38
4.2 Prerequisites 39
  4.2.1 USB camera 40
  4.2.2 RQT graph 40
  4.2.3 Rviz 41
4.3 Introduction 42
  4.3.1 Experiment 1 (Fixed camera no movement) 44
  4.3.2 Experiment 2 (Slow movement of the camera) 45
  4.3.3 Experiment 3 (empty environment) 46
  4.3.4 Experiment 4 (full furniture environment) 47
  4.3.5 Experiment 5 (light and dark environment) 48
  4.3.6 Experiment 6 (moving a robot on straight path) 49
  4.3.7 Experiment 7 (round trip movement) 53
4.4 Experiment 8 (dynamic environment) 55
4.5 Summary 56

CHAPTER 5 CONCLUSION AND RECOMMENDATION 58

5.1 Conclusion 58
5.2 Recommendation for Future Research 59

REFERENCES 60
LIST OF TABLES

Table 2-1 SUMMARY OF REPRESENTED VISUAL SLAM SYSTEMS PREVIOUSLY STUDIED AND ITS ADVANTAGE 15
LIST OF FIGURES

2.1 Tracking of Davison's Mono SLAM 7
2.2 Mono SLAM Lost tracking 7
2.3 Unscented Kalman Filter (UKF) 8
2.4 Based methodology for monocular SLAM 8
2.5 PTAM (Parallel Tracking and Mapping) 9
2.6 Large-Scale Direct Monocular SLAM-LSD 10
2.7 Light-weight Rover Unit (LRU) with 3D illustration 11
2.8 OctoMap 12
   3D Map Representation for Robotic Systems. Octomap 12
2.9 12
2.10 ORBODOM trajectory 14
2.11 Qualitative evaluation orb-slam2 with 6dof motion 13
3.1 A sampled frame work of ORB-SLAM2 16
3.2 Tracking thread 18
3.3 Local mapping thread 22
3.4 Detailed flowchart 42
3.5 cmake gcc and g++ tool command 43
3.6 Pangolin command 44
3.7 Eigen installation command 44
3.8 Ros melodic cover page 45
3.9 Ros and RQT source download 46
3.10 ROS installation 46
3.11 Python ROS build 46
3.12 ROScor terminal interface 47
3.13 ORB-SLAM2 GitHub downloading 47
3.14 Compilation command 48
3.15 ORB folder files on terminal 48
3.16 Usb cam source instalation
3.17 Usb cam source file path
3.18 Usb cam quality parameter
3.19 Usb cam window test
3.20 RQT graph nodes connections
3.21 Roslaunch terminal interface
3.22 Rviz terminal interface
3.23 Full ORB-SLAM2 terminal launch
3.24 RVIZ principal window
4.1 The usb cam used in the experiment
4.3 Usb cam pose and orientation on laptop
4.4 RQT graph nodes of the system
4.5 RVIZ first images window
4.6 Fixed camera frames images and its 3D plan
4.7 Moving camera frames images
4.8 Moving camera frames and its 3D plan
4.9 Blank wall captured images
4.9 Blank wall keyframes and its 3D plan
4.10 Full furniture environment keyframes images with 3D plan
4.11 Dark and lighted plan images
4.13 Path way drown the robot in movement position
4.14 X Y Z trajectory position of the ORB-SLAM2
4.15 Keyframes and 3D plan points extracted from moving robot
4.16 Roundtrip trajectory 3D plan points and keyframes
4.17 Roundtrip path way robot direction
4.18 Images, keyframes extracted and 3D points of dynamic environment
**LIST OF SYMBOLS AND ABBREVIATIONS**

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLAM</td>
<td>Simultaneously Localization and Mapping</td>
</tr>
<tr>
<td>VO</td>
<td>Visual Odometry</td>
</tr>
<tr>
<td>LSD-SLAM</td>
<td>Large Scale Direct - Simultaneously Localization and Mapping</td>
</tr>
<tr>
<td>ORB</td>
<td>Oriented Features from Accelerated Segment Test and Rotated Binary Robust Independent Elementary Features Mapping</td>
</tr>
<tr>
<td>PnP</td>
<td>Perspective-n-Point Camera Pose Estimation</td>
</tr>
<tr>
<td>BoW</td>
<td>Bag of Words</td>
</tr>
<tr>
<td>BA</td>
<td>Bundle Adjustment</td>
</tr>
<tr>
<td>ROS</td>
<td>Robot Operating System</td>
</tr>
</tbody>
</table>
CHAPTER 1

INTRODUCTION

1.1 Background of the study

For almost a decade, robotics has expanded considerably in areas that have interested academics and researchers. [1] the navigation and localization of mobile robots was one of the trending topics of the moment [2], with the main question being how can the robot move to an unfamiliar area and how can it decide its own direction, recognize its surroundings and guide itself by observation of the environment. In addition to navigating, how the robot may record its corresponding location in world coordinates while also building up a map of the surrounding environment that are based on previous knowledge without the assistance of humans.

1.2 Introduction of Visual SLAM

Research into mobile robots and autonomous systems has garnered considerable interest from scientists across the globe in recent decades, with major advancements in technology, as well as technological discoveries.

At the moment, mobile robots are capable of performing complicated tasks independently, such as package delivery and driving an autonomous vehicle. During the course of these applications, mobile robots must navigate through complicated and dynamic indoor and outdoor settings without the assistance of a human operator. Robots must be able to locate themselves in their surroundings in order to perform
autonomous navigation and route planning effectively and securely in their environment. As a result, researchers have thoroughly investigated the positioning issue and developed a number of methods for resolving the problem of localization.

Simultaneous Localization and Mapping (SLAM) is very important for a robot while it is exploring an unfamiliar area for the first time. There are many mathematical insights and domain expertise required to complete the tasks of determining the present location of the robot, producing robot coordinates from camera inputs, and building a global map of the environment. Several theoretical frameworks, including Probabilistic Robotics [3] and Multiple View Geometry in Computer Vision [4] serve as the foundation for this investigation. SLAM research has produced a large number of different algorithms and building modules that have been suggested.

Since its initial presentation in 1988, the SLAM research field has been around for almost three decades. According to the overall procedure, it may be split into two major parts: environment detection, efficiency, and optimization. Early SLAM research concentrated on the use of filter theory to reduce the motion poses and noise associated with the map's landmark points to a bare minimum. The majority of these efforts are based on optimization.

Previous generations of robots have relied on odometry, which relies on the encoder of the wheel to measure the amount of rotation the robot's wheels have taken place, to determine their location. After combining this measurement with the robot's motion model, the current position of the robot in relation to a global reference frame can be determined for the robot. There are some significant limitations to wheel odometry. Following the movement of ground vehicles, and while the localization data is incremental and based on previously estimated positions, the measurement error will accumulate over time, resulting in the estimated robot pose deviating from its actual position. There are a variety of factors that contribute to errors, such as rough surfaces and greasy wheels on smooth surfaces.

Other methods of localization, such as the use of ultrasonic, GPS, Inertial Measurement Units (IMUs), and Lidar, have been proposed in order to circumvent these limitations. As research has progressed, Lidar has gradually displaced other sensors and has emerged as the most reliable mainstream solution available [35]. A
great deal depends on the type of sensor being used and how it is installed in order for SLAM to be implemented successfully and with ease. SLAM is currently classified into two categories: laser and vision, which are based on the sensor used. It is still early in the history of laser SLAM research, and both the theory and the engineering are well developed. It is called Visual SLAM (VSLAM) because it relies solely on visual information, and the camera is the only sensor that can gather data. In the current state of visual SLAM research, it is only rarely seen in actual product applications and is still in the laboratory research stage. Nonetheless, many innovative VSLAM solutions have been proposed in the past ten years.

In 2007, A.J.Davison introduced the first monocular SLAM solution, dubbed "MonoSLAM: real-time single camera SLAM," to the market. [5]. It is the first SLAM system to operate in real time with monocular vision. However, the system is unable to operate online. It depends on the robot to transport the camera and gather data, after which it does offline localization and mapping. At the same time, G. Klein and D. Murray suggested "Parallel Tracking and Mapping" in 2007, [6], which was published in the journal Nature. It provides a framework for the present visual SLAM system, which will be described in more detail in the next subsections. Newcombe, Richard A., Steven J. Lovegrove, and Andrew J. Davison [7] published a paper in 2011 introducing the concept of "Dense Tracking and Mapping" (DTAM). SLAM is the first visual SLAM solution to be developed utilizing the direct approach, which implies that just the feature points are retrieved from the picture and no descriptor for the feature points is developed. Feature points are used in visual SLAM to construct the map, which is sparse since feature points only extract information from a small portion of the image's total amount of information. SVO: Fast Semi-Direct Monocular Visual Odometry was suggested by Forster, Christian, Matia Pizzoli, and Davide Scaramuzza in 2014 [8], and it is still being used today. In this technique, feature points and the direct method are combined to reduce the amount of computation required, allowing it to be performed quickly and efficiently. It was the same year that J. Engel and T. Schops and D. Cremers published "Large-Scale Direct Monocular SLAM," [9] which was the first time this term was used. It is based on the direct technique of calculation. As a result, it is capable of producing a dense map. Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D. Tardos suggested "Oriented FAST and Rotating BRIEF (ORB) -SLAM" in 2015 [10], and ORB-SLAM2 was added in 2017. They also
proposed "Oriented FAST and rotating BRIEF (ORB) -SLAM" in 2015. There are no feature points in this method, thus it's a fully functional visual SLAM algorithm. Unlike other visual SLAM systems, it can be utilized with monocular, stereo, and RGB-D cameras, as opposed to other SLAM solutions. At the same time, it serves a dual purpose and can be used both indoors and outdoors. As a result, it has been selected for this project. Chapters 3 and 4 provide in-depth introductions to both concepts and performance, respectively.

1.3 Problem Statement

The ORB-SLAM2 bring us a new filed for the technology, a filed which will be so useful and can modify the daily life of the next generation in positive way with his benefits. In this work will utilize ORB-SLAM2 for Simultaneous Localization and Mapping. Currently, most research on ORB-SLAM2 uses expensive stereo camera to achieve high accuracy result for SLAM. However, most of the car today uses DASHCAM which is a low-cost monocular camera. From previous study, the majority of research papers explore the use of RGB-D and high-cost camera. Therefore, this research evaluates the use of low-cost monocular camera for ORB-SLAM2

1.4 Objectives of the study

In this thesis we will suggest and build a low-cost autonomous mobile robot system with ORB-SLAM 2 and monocular camera.

The main contributions of this paper are summarized as follows:

i) To propose an ORB-SLAM 2 with monocular-camera.

ii) To develop an ORB-SLAM 2 with a low-cost monocular camera for autonomous mobile robot.

iii) To evaluate the accuracy of ORB-SLAM 2 with monocular-camera
1.5 Scope of the study

The following are the scope of this research:

1. The operating system which will be used to run the ORB SLAM 2 is Ubuntu
3. We have run ORB-SLAM2 in an Intel Core i7-4790 desktop computer with 8Gb RAM
4. Due to the covid19, the experiment is done in indoor environment

1.6 Thesis outline

The content of this thesis is distributed within five chapters as described below:

Chapter 1: The first chapter give brief description about ORB-SLAM 2 by introducing the system so that the readers can understand what the report is about
Chapter 2: This chapter discuss about the ORB-SLAM2 related works and what previous researchers has done and where they stop their research.
Chapter 3: The chapter introduce the methods and the methodology of how our system will be able to work
Chapter 4: This chapter focuses on the obtained results from the experience on the real time and the analyses of the results obtained.
Chapter 5: The final chapter of this thesis highlights the significant findings of the thesis and suggests recommendation of future work.
CHAPTER 2

LITERATURE REVIEW

2.1 Overview

In this chapter an overview of ORB-SLAM2, historical development of the SLAM, methods, techniques, characterization equipment. This chapter help better understanding of this project through ancient theories

2.2 Related work (Previous Iconic Solutions of Visual SLAM)

To ameliorate visual slam for mobile robot, Davison put forward the MONO SLAM [11] as the first high frame-rate, real-time Monocular SLAM solution. The algorithm generates a sparse but stable map of landmarks based on a probabilistic system and fixed the issue of monocular feature initialization.
In this work they used the EKF (Extended Kalman Filter) algorithm [12]. But a problem occurred in the error drift, which caused the loss of tracking system due to the sparse landmarks map. A solution has been found for improving linearization error and has been applied in VISUAL SLAM the UKF (Unscented Kalman Filter) [13] and improved.
Moreover Sim [14] proposed a particle filter-based monocular SLAM algorithm to avoid linearization even so that this method ameliorate accuracy it also prolongs the computational complication.

Figure 0.3 Comparison of diverse iterations for Localization primarily based totally at the Unscented Kalman Filter (UKF) algorithm.

Figure 2.4: based methodology for monocular SLAM
The previous SLAM is employed for chase and reconstructing a short-run native map (blue), the latter for building a reusable world map (red and green).

- Bottom: (from left to right) this frame, the most recent direct keyframe with color-coded depths.
- The latest feature-based keyframe with the matched options (red) and the projection of direct map points (blue)

Klein and Murray et al [15] proposed a keyframe-based monocular SLAM algorithm, called PTAM (parallel tracking and mapping). The algorithm interpolates a keyframe extraction method, and for the first time, divided the tracking and mapping into two parallel threads excluding PTAM used nonlinear optimization instead of the EKF method to eliminate the linearization error problem.[16][17]

![Figure 2.5: PTAM (Parallel Tracking and Mapping) [15]](image)

PTAM is a camera tracking system for augmented reality. It requires no markers, pre-made maps, known templates, or inertial sensors but the loss tracking of large scale environment due of the deconcentrating in the study about the global optimization of PTAM pushed Engel to propose the Large-Scale Direct Monocular SLAM (LSD-SLAM)
algorithm, [18] which is the monocular SLAM calculation dependent on the direct strategy. LSD-SLAM can build a semi-thick worldwide steady guide, which is a more complete representation of the environmental than a point-cloud map dependent on the element method. What’s more, LSD-SLAM presents a novel direct following strategy which can precisely recognize scale drift, and the algorithm can run continuously on the Central Processing Unit (CPU). In any case, this work actually didn't settle the grey scale invariant speculation of the immediate technique, which made the execution of LSD-SLAM decline quickly when the robot was working in an environment with regular brightening changes.[18]

Large-Scale Direct Monocular SLAM: LSD-SLAM generates a uniform international map, exploitation direct image alignment and probabilistic, semi-dense depth maps rather than key points. In the top Accumulated point clouds of all keyframes of a medium-sized generated in real-time. Down selection of keyframes with color-coded semi-dense inverse depth map. Brand and Schuster put all their concentration in the localization and navigation of the mobile robot in unknown area that’s why they created the on board SLAM [19] algorithm which utilize 2.5D maps that can operate directly in

![Figure 2.6: Large-Scale Direct Monocular SLAM: LSD [18]](image)
fast obstacle avoidance and local path planning, and which also help for to be a suitable input for a 2D grid map.

![Image](image_url)

**Figure 2.7:** light-weight Rover Unit (LRU) with 3D illustration

Figure 2.7: Top: light-weight Rover Unit (LRU) with 3D illustration of a crater in our out of doors testbed, akin to the central a part of the map displayed below. Bottom: top-down read on ensuing 3D map generated by our SLAM system (green path: SLAM estimates offered to the golem at its individual positions, blue: SLAM graph edges between consequent submaps from detector fusion, yellow: edges representing submap matches.

Lee and song et al [20] Proposed an Slam algorithm so the mobile Robot could identify the object without relying on any a priori information, and inserts the detected objects as landmarks into the grid map that’s called autonomous detection and the registration of objects The map created has a high accuracy in positioning and navigation if we compare it with the old visual SLAM. And another time the ignorance of 3D map value and the fail in reconstruction of a complete environmental structure made them lost the main power of VISUAL SLAM, so the solution for this problem was dense 3D
reconstruction method based on the ORB-SLAM algorithm which was proposed by Lv, Q.; Lin, H et al[21] The method enables the ORB-SLAM system to construct an octomap [22] based on the octrees and probabilistic occupancy estimation, and the improved ORB-SLAM can complete the map reconstruction in real time using a Kinect 2.0 camera in the real world. But this work is not able to express the working naturality as a result of using octomap. Which didn’t allow us to check the mapping effect the main cause of the interactivity decrease of the mobile robot.

Figure 2.8: OctoMap

Figure 2.9: OctoMap: A Probabilistic, Flexible, and Compact 3D Map Representation for Robotic Systems.

Scientists have put forth a ton of work to attempt and upgrade the presentation of ORB-SLAM2 Wang [23] proposed a monocular SLAM calculation dependent on the
REFERENCES


