YaleNUSCollege

Real-time Filtering for Multi-sensory SLAM Benchmarking

Jianglong Liao

Capstone Final Report for BSc (Honours) in

Mathematical, Computational and Statistical Sciences

Supervised by: Dr. Bruno Bodin

AY 2019/2020

Yale-NUS College Capstone Project

DECLARATION & CONSENT

1. I declare that the product of this Project, the Thesis, is the end result of my own work and that due acknowledgement has been given in the bibliography and references to ALL sources be they printed, electronic, or personal, in accordance with the academic regulations of Yale-NUS College.

2. I acknowledge that the Thesis is subject to the policies relating to Yale-NUS College Intellectual Property (Yale-NUS HR 039).

ACCESS LEVEL

3. I agree, in consultation with my supervisor(s), that the Thesis be given the access level specified below: [check one only]

   o Unrestricted access
   Make the Thesis immediately available for worldwide access.

   o Access restricted to Yale-NUS College for a limited period
   Make the Thesis immediately available for Yale-NUS College access only from _____________ (mm/yyyy) to _______________ (mm/yyyy), up to a maximum of 2 years for the following reason(s): (please specify; attach a separate sheet if necessary):
   ____.

   After this period, the Thesis will be made available for worldwide access.

   o Other restrictions: (please specify if any part of your thesis should be restricted)
   ____
   ____

____
Name & Residential College of Student

____________________________________ ___________________________
Signature of Student Date

____________________________________ ___________________________
Name & Signature of Supervisor Date

Acknowledgements

Acknowledgements go here.

YALE-NUS COLLEGE

Abstract

B.Sc (Hons)

Real-time Filtering for Multi-sensory SLAM Benchmarking

by Jianglong LIAO

Abstract goes here.

Contents

Chapter 1

Introduction and Motivation

The simultaneous localization and mapping (SLAM) problem questions the autonomy of a mobile robot in an unfamiliarised environment: would a robot be able to explore and incrementally construct the information of its surroundings, yet at the same time accurately locate its position in the complex, realistic environment? Solutions to the SLAM problem would imply that a robot has become "truly autonomous" as it no longer requires arbitral instructions or fixed input in order to understand its relation with its situated context [1]. Over the decades, multiple solutions have been proposed in different technical and mathematical forms, and have been applied to various fields of robots, including group aerial, underwater, indoor, etc. From a theoretical perspective, SLAM is deemed as a solved problem [1].

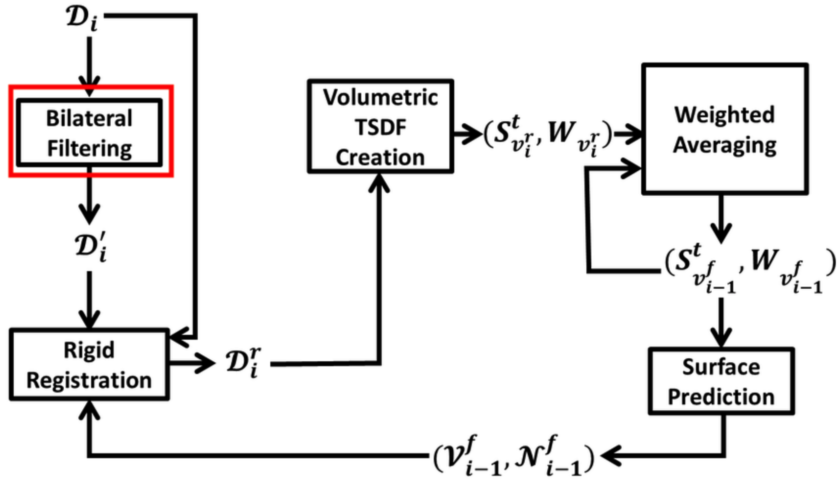
However, in a realistic environment, different solutions will showcase their advantages and disadvantages. For example, algorithms such as iterative closet point (ICP) perform exceptionally well in an indoor context, but when it comes to complex, outdoor environment, they may fail to reach the expectation for accuracy [2]. Moreover, substantial issue remains in implementing these solutions. Problems such as the trade-off between accuracy, speed, memory and power consumption continue to challenge the researchers and practical commercialization. Hence, there has been sustained research in practical implementation and evaluation of different SLAM solutions.

In recent year, the creation of new sensors and increase in processing power have enabled the releases of newer, more accurate and efficient SLAM systems. As SLAM systems continue to evolve and develop, needs for evaluating diverse SLAM algorithms surge exponentially. Unfortunately, despite heated discussion over SLAM, there has been little attention devoted to standardizing the evaluation of all SLAM systems. Different approaches have been used by researchers to test and experiment the SLAM algorithms, making it difficult to compare the performance of various SLAM systems. Furthermore, most of the testing and evaluation can take considerable effort and have been done in a specific, manual procedure. Researchers need a plug and play system that can easily evaluate the performance different SLAM systems and streamline the testing process.

Hence, SLAMBench has been created by numerous SLAM researchers to streamline and standardize the benchmarking process of SLAM systems. The program is structured to be dataset-agnostic and SLAM-system-agnostic, so that researchers can conveniently input their own SLAM model and testing data. In addition, by having a standard SLAM benchmarking program, researchers would also be able to rest assured that the quantitative testing results are reproducible [3]. The program now is developed in multiple languages and supports eight state-of-the-art SLAM systems.

Yet, an essential component is missing in the current SLAMBenck software architecture: a real-time filter. While undoubtedly extracting as much information from sensors as possible would render the most accurate solution, fixed computational bounds and energy consuming limitation of a mobile robot will restrict the quantity and quality of sequential inputs that an algorithm can ingest [4]. Therefore, before feeding the sensor data into the correspondence SLAM system, real-time preprocessing is an indispensable step to improve the computational speed and lower the energy consumption.

FIGURE 1.1: Example of filtering in KinectFusion



Currently, the SLAMBench framework consists of two configurable parts – dataset and SLAM algorithms. To test the performance of filtering and other preprocessing method, researchers are forced to manually customized their filter with deliberate calibration, for which these tinkering approaches will defeat the purpose of SLAMBench and complicate the benchmarking process. In addition, current SLAMBench framework only allows one data pipeline to deliver sensor data sequentially from the

I/O system, where data are ingested, to the loader, where the SLAM algorithms are evaluated. This simplistic design of data pipeline does not account for concurrent data delivery from multiple sensors and does not provide room for filtering and preprocessing multi-sensory data based on sequential time frames. Hence, restructuring the software architecture is required to decouple and modularize the new filtering module and ensure its compatibility with the rest parts of SLAMBench.

In this capstone project I introduce real-time filtering system for the current SLAMBench program. This filtering system enables researchers to evaluate the impact of various filtering and preprocessing techniques on different open source or proprietary SLAM systems. It is a tool that ensures the reproducibility of results for existing filtering methods and allows the integration and evaluation of new filters for SLAM algorithms. Through a series of testing and experimentations, we demonstrate its configurability, extensibility and its ease of use to benchmark different filters for SLAM systems. In summary, we present the following contributions:

- A publicly available filtering system, integrated with SLAMBench, that supports quantitative, reproducible comparison of different filtering techniques for SLAM systems.

- A filtering system that is configurable, extensible and enables plug and play of new filters.

- Various experimenting cases of different filters (resize, blur, random drop, etc.) on state-of-art SLAM systems.

Chapter 2

Background and Related Work

This chapter serves to introduce key and essential concepts of robotics system benchmarking and SLAM systems. Prior knowledge of SLAM and relevant technical understanding is not presupposed for reading this chapter, so the internal workings of specific SLAM systems, including the applied mathematics and codes, will not be elaborated. Rather, we introduce modular conceptualization of robotics system benchmarking and SLAM to assist the reader in understanding the structure of our filtering system. In-depth knowledge of SLAM systems is not strictly required, but we reference some of them in evaluating the results of our testing and experiments.

2.1 Robotics System Benchmarking

2.1.1 Background

Collaborative operations between man and machine has not only improve productivity, but also in some cases provide solutions not yet available to humans alone (disaster search and rescue robots) [5]. Co-development between engineering and computer sciences have started to lay the foundations for mechanical hardware and software systems of generic and domain-specific robots. Yet without standardization, developing robotic technologies becomes not only ineffective, but also unprofessional and dangerous. For example, with proper standardization of video medium, there was a wide confusion and frustration of choosing HD DVD or Blue Ray for home-video appliance among consumers. On the other hand, successful standardization would bring synergies among different fields and cohesion between various research projects. For instance, the immense progress in the wireless communication technology has largely attribute to the establishment of widely accepted industrial standards such as IEEE 802.11. [5].

2.1.2 Methodology

Objective performance evaluation is required for each field of robotic systems to continue the progress of research and to facilitate the acceptance of robotic technologies. Yet it is not a trivial task to ensure that the evaluation is repeatable, unambiguous and holistic. Specifically, in the field of robot navigation and mapping, distinct research projects may take different approaches to measure the accuracy of mapping [6] [7]. However, even if a standard for accuracy comparison do exists, this benchmark is not enough to characterise the mapping system as a whole: there are still considerations such as energy consumption and performance. Recently, there has been initiative, such as RoboBench, that intends to construct a framework that allows researchers to holistically benchmark all aspects of robotic systems and form a set of comparable, reproducible measure.

Hence, when building a benchmarking framework for robots, systematic approach needs to be taken to account the cost of computing over the task, the performance of robotics algorithms, as well as the costs incurred in infrastructure such as energy and networks [8].

To ensure the completeness of Robotic system benchmarking, two aspects of benchmark categories need to be considered: the method that determines how we benchmark, and the focus that determines what we benchmark [9]. In general, there are two methods of benchmarking: analytical approach that concerns with evaluation of a robotics system on its own, and functional approach that test how a system solves a specific problem. Similarly, in terms of the focus, there are also two targets of benchmarking: component – evaluation on one specific part of the system, and system – evaluation on the overall system. By combining these criteria, we are able to form a matrix of our robotics benchmark system:

TABLE 2.1: Robotics benchmark matrix.

Analytical	1	3
Functional	2	4
	Component (filter)	System (SLAM system)

2.1.3 Application

In the case of our filtering system in the SLAMBench, to apply the framework of holistic evaluation, we identify four potential areas of benchmark that we intend to explore based on Table 2.1.

1. **Use the same dataset and apply different filters.**

   Since we already understand the task of each filter, this process merely allows us to observe whether data set has been correctly filtered.

2. **Use different datasets and apply the same filter.**

   Targeting problem: Supposed task of a filter

   Benchmark purpose: what is the performance of a filter at carrying out its filtering tasks.

3. **Use the same filter and apply to different SLAM systems.**

   Benchmark purpose: observe how the same filtering technique would impact different SLAM systems.

4. **Use different filters and apply to the same SLAM systems.**

   Targeting problem: SLAM benchmarking outcome (accuracy, computational speed, energy consumption)

   Benchmark purpose: how does different filters affect a specific SLAM system? What is the optimal filtering technique for a SLAM system?

These benchmarking methods are designed to be holistic to test the filtering system and its correlation with the SLAM systems. In addition, to ensure that the benchmarking results are repeatable, only minimal configuration should be required throughout the system. Therefore, modularization of each part of the system, where every module functions independently from each other, is the key to protect the integrity of the whole benchmarking system. Further details with regards to benchmarking and system design are included in the following chapters ().

2.2 Simultaneous Localization and Mapping

2.2.1 SLAM Background

As mentioned in the Background section, we provide a conceptual state-of-art description of the Simultaneous Localization and Mapping (SLAM) system. The system, as the name suggests, consists of two inseparable parts: mapping and localization. Mapping refers to the process of incrementally building a map representation of an environment, while localization denotes the method to locate the robot in the current map in order to minimize the error [10]. SLAM can be implemented by combination of hardware and software, including laser, monocular vision, and visual-inertial SLAM system, which leverages on RGB-D camera and inertial measurement unit (IMU).

For now, whether SLAM is solved is still hard to answer in reality [11], since under the SLAM domain, there are still various subdomains that are categorized by combining specific type of robot, operational environment and performance requirements. Some, such as deterministic navigation environment and highly robust laser robot, can largely considered solved [12]. On the other hand, for a visual-inertial SLAM system, the accuracy of mapping still varies dramatically with regards to the motion of robot and environment, including fast movement and highly dynamic and challenging environment. Some small alternation in the navigation environment can still easily induce failure and largely drop the accuracy of mapping. In addition, performance of a SLAM system is also directly determined by the computational bound and system requirement, in which case the evaluation of a SLAM system goes beyond theoretical mathematical investigation and aims to provide practical systematic optimization.

2.2.2 SLAM Benchmarking

Therefore, benchmarking a SLAM system now requires a renewed set of requirements to fulfill the current research demand and development of SLAM [13]. First, SLAM system needs to be tested on the robustness of performance. To what extent can a SLAM system operate with a large variety of dataset, yet maintain low failure rate for a long period of time? This benchmarking requirement is essential for generic SLAM system that would be applied to different scenarios and conditions of robots. In this regard, SLAMBench provides interfaces that allow researchers to test a SLAM system with different standardized datasets. Second, SLAM system needs to benchmarked on a high-level understanding. Basic geometry reconstruction is no longer the only measure to represent the robot's understanding of the environment. High-level geometry and semantics now become the main objective measure for accuracy of mapping. To address this requirement, SLAM runs iterative closest point (ICP) algorithm on the point cloud models to gain a high-level understanding of differences between reconstruction and groud truth. Third, SLAM system will need to be evaluated on its efficient usage of limited resources. These resources include type and number of sensors, computational power, memory and energy storage. Beyond algorithmic accuracy, SLAMBench provides performance metrics that consists of computation speed, power consumption and memory usage to evaluate SLAM system's performance with limited resources.

Finally, a SLAM system should be tested on specific task-driven functionality. This test will evaluate the capability of a SLAM system to process the relevant information and filter out inessential sensor data before feeding these data through core algorithms. For this task, a decoupled filtering system is required to allow real-time prepossessing of sensor data from multiple channels. The filtering system could be utilized for the following purposes.

- **Expose the advantage and disadvantage of certain SLAM.** In visual SLAM, there are "direct" algorithms that directly use all intensity values [14], and "indirect" methods that first extract features before ingestion [15]. On the other hand, there are "sparse" method that only uses a subset of features [14], and "dense" method that ingest all pixel values [16]. By using a filtering system, characteristics of a dataset (brightness, sharpness, etc.) could be extended to amplify strengths and weaknesses of all methods;

- **Explore methods to optimize SLAM systems.** The filtering system allows real-time preprocessing of dataset to test certain parameters and improve the robustness of SLAM system when applied to different situations. Moreover, applying a specific filter has the potential to lessen power consumption, memory storage, and at the same time, improve computational speed.

However, this filtering system is not present in the SLAMBench framework. Currently, there are other general-purpose SLAM benchmarking tools such as KITTI Benchmark Suite [17] and TUM RGB-D benchmarking [6]. Yet compared to SLAMBench, these benchmark tools do not allow flexible integration of real-world datasets. In addition, these systems do not entirely address the SLAM benchmarking requirements as mentioned above. Hence, an addition of filtering system to the SLAMBench framework makes the streamlined benchmarking procedure even more comprehensive, and at the same time, maintains the flexibility and scalability of SLAMBench.

Bibliography

[1] H. Durrant-Whyte and T. Bailey, “Simultaneous localization and mapping: Part i”, *IEEE robotics & automation magazine*, vol. 13, no. 2, pp. 99–110, 2006.

[2] K. Krinkin, A. Filatov, A. yom Filatov, A. Huletski, and D. Kartashov, “Evaluation of modern laser based indoor slam algorithms”, in *2018 22nd Conference of Open Innovations Association (FRUCT)*, IEEE, 2018, pp. 101–106.

[3] B. Bodin, H. Wagstaff, S. Saecdi, L. Nardi, E. Vespa, J. Mawer, A. Nisbet, M. Luján, S. Furber, A. J. Davison, *et al.*, “Slambench2: Multi-objective head-to-head benchmarking for visual slam”, in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2018, pp. 1–8.

[4] H. Strasdat, J. Montiel, and A. J. Davison, “Real-time monocular slam: Why filter?”, in *2010 IEEE International Conference on Robotics and Automation*, IEEE, 2010, pp. 2657–2664.

[5] R. Madhavan, R. Lakaemper, and T. Kalmár-Nagy, “Benchmarking and standardization of intelligent robotic systems”, in *2009 International Conference on Advanced Robotics*, IEEE, 2009, pp. 1–7.

[6] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, “A benchmark for the evaluation of rgb-d slam systems”, in *2012*

IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2012, pp. 573–580.

[7] A. Handa, T. Whelan, J. McDonald, and A. J. Davison, "A benchmark for rgb-d visual odometry, 3d reconstruction and slam", in *2014 IEEE international conference on Robotics and automation (ICRA)*, IEEE, 2014, pp. 1524–1531.

[8] J. Weisz, Y. Huang, F. Lier, S. Sethumadhavan, and P. Allen, "Robobench: Towards sustainable robotics system benchmarking", in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2016, pp. 3383–3389.

[9] A. P. del Pobil, R. Madhavan, and E. Messina, "Benchmarks in robotics research", in *Workshop IROS*, Citeseer, 2006.

[10] S. Perera, D. N. Barnes, and D. A. Zelinsky, "Exploration: Simultaneous localization and mapping (slam)", *Computer Vision: A Reference Guide*, pp. 268–275, 2014.

[11] U. Frese, "Interview: Is slam solved?", *KI-Künstliche Intelligenz*, vol. 24, no. 3, pp. 255–257, 2010.

[12] K. Robotics, *Kuka navigation solution, 2016*.

[13] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age", *IEEE Transactions on robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[14] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. W. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking.", in *ISMAR*, vol. 11, 2011, pp. 127–136.

[15] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "Monoslam: Real-time single camera slam", *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 6, pp. 1052–1067, 2007.

[16] T. Whelan, S. Leutenegger, R Salas-Moreno, B. Glocker, and A. Davison, "Elasticfusion: Dense slam without a pose graph", Robotics: Science and Systems, 2015.

[17] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite", in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2012, pp. 3354–3361.