This is a repository copy of *Evaluation of SLAM algorithms for search and rescue applications*.

White Rose Research Online URL for this paper:
https://eprints.whiterose.ac.uk/201928/

Version: Accepted Version

Proceedings Paper:

Yang, Zhiyu, Naz, Nabila, Liu, Pengcheng orcid.org/0000-0003-0677-4421 et al. (1 more author) (2023) Evaluation of SLAM algorithms for search and rescue applications. In: The 24th Annual Conference Towards Autonomous Robotic Systems (TAROS2023). 24th Towards Autonomous Robotic Systems Conference (TAROS), 13-15 Sep 2023 Springer , GBR .

Reuse

This article is distributed under the terms of the Creative Commons Attribution (CC BY) licence. This licence allows you to distribute, remix, tweak, and build upon the work, even commercially, as long as you credit the authors for the original work. More information and the full terms of the licence here: https://creativecommons.org/licenses/

Takedown

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk/

Evaluation of SLAM algorithms for Search and Rescue applications

Zhiyuan Yang[1], Nabila Naz[2], Pengcheng Liu[3], and M. Nazmul Huda[4]

[1] Shanghai Huawei Technology Co.,Ltd.
[2] School of Engineering, Technology and Design, Canterbury Christ Church University, UK
[3] Department of Computer Science, University of York, UK
[4] Department of Electronic and Electrical Engineering, Brunel University London, UK
mdnazmul.huda@brunel.ac.uk

Abstract. Search and rescue robots have been widely investigated to detect humans in disaster scenarios. SLAM (Simultaneous Localisation And Mapping), as a critical function of the robot, can localise the robot and create the map during the rescue tasks. In this paper, prominent 2D SLAM algorithms are investigated and three of them (Gmapping, Hector, and Karto) are implemented on a low-cost search and rescue robot to demonstrate their feasibility. Moreover, experiments containing various ground surface scenarios are performed. Maps created by various SLAM algorithms are compared to identify the best SLAM algorithm search and rescue tasks using low-cost robots. The experimental results suggest that Karto SLAM performs best for low-cost search and rescue robots among the three SLAM algorithms.

Keywords: Search and Rescue, low-cost robot, SLAM, Karto, Gmapping.

1 Introduction

Robots are widely used in highly repeatable, dangerous and precise tasks in industry, service, military, healthcare, scientific research and so on. The search and rescue robots [1], have been employed in rescue tasks for more than 20 years [2]. However, performing rescue tasks in an extreme situation has many requirements [3]. Firstly, the size of the robot should be suitable. It should be small enough to enter the narrow area and big enough to move on the rough terrain without being stuck, which means the hardware of the robot is also required to be simplified [4]. Secondly, its operation should not be difficult so that ordinary humans can operate it. Thirdly, it should be low cost so that it can be widely used by people in various scenarios [5]. Fourthly, it should have basic functions, such as SLAM for map building and real-time video streaming for human life detection [6].To keep the robot's cost down, the hardware, sensors (e.g., LIDAR) and computing resources need to be low-cost as well.

SLAM [7] (simultaneous localization and mapping) aims to implement the robot localization and map building in an unknown environment at the same time [8]. Many SLAM algorithms have been developed and implemented on the ground mobile robots. Several studies have also been performed comparing the performance of various SLAM algorithms [9]–[13] either through simulations [9], [10], [12] or experiments [13] or both [11]. However, all of them aim to display the optimum performance of the algorithms. As a result, some high-cost hardware such as the high precision Lidar [14] and high-performance controller [11], have been employed. Besides, the environmental constraints have been ignored, which means all of these researches assumed the environment to be a flat ground surface. Therefore, it is critical to implement the SLAM algorithms on a low-cost robot and explore the influence of various ground surfaces which are comparable to the rescue scenarios, in order to find the most effective and efficient SLAM algorithms for search and rescue robot to be applied in specific disaster scenarios. This research investigates three SLAM algorithms on a low-cost mobile robot and finds the algorithms' performance through a set of experiments including different types of ground surfaces.

This paper has the following structure. Section 2 presents the low-cost search and rescue robot platform; section 3 investigates five popular SLAM algorithms and identifies three algorithms for further investigation; section 4 presents the maps built by three selected SLAM algorithms and tests their feasibility for a low-cost robot; section 5 presents experiments to investigate the impact of various ground surfaces on map building, to emulate the environment of real disaster scenarios; section 6 analyses the findings of the research and section 7 concludes the paper.

2 Low-cost Search and Rescue Robot Platform

This research aims to evaluate the performance of various SLAM algorithms for a low-cost search and rescue robot (Fig. 1) in disaster scenarios. The software framework and hardware specifications of the low-cost robot platform are provided in this section.

For the software framework, ROS (Robot Operating System) is used on the robot, which is originated from the AI Lab project of Stanford University, developed by Willow Garage in 2010 [15]. The main hardware components of the robot are a tracked Chassis, an RPLidar A1M8-R5, a Raspberry Pi 3 Model B (main controller), a secondary control board based on STM32 (STM32F103RCT6), two DC brushed motor a 9-DoF IMU (GY-85), a 12V 8400mAh li-po battery pack. The total cost of the robot is approximately £210 (£40 - Raspberry Pi, £20 - STM32 based secondary controller board, £90 - Lidar, £40-chassis with two motors, £20 - cables, screws, and others).

Fig. 1. Low-Cost Search and Rescue Robot Platform

The Raspberry Pi is used as the main controller of the robot as it is stable and widely used to work as an embedded controller on robotics. Raspberry Pi 3B is based on Quad Cortex A53 chip with 1.2 GHz operating speed [16]. The RPLidar A1 is used because of its stable performance and low cost as an elementary laser range scanner. The parameters of the Lidar are 12m distance range, 360° angular range, and 4000Hz sample frequency[17].

3 SLAM Algorithms and Discussions

Many 2D SLAM algorithms have been developed. The example includes Gmapping SLAM [18], [19], Hector SLAM [14], [20], Karto SLAM [11], Cartographer SLAM [21], Core SLAM [22] etc. This section briefly introduces the above-mentioned SLAM algorithms and identifies the algorithms that could be employed in low-cost search and rescue robots. The readers who want more details on the algorithms may read the review papers on 2D SLAM algorithms [11], [23], [24].

3.1 SLAM Algorithms

Gmapping SLAM is based on Rao-Blackwellized Particle Filters (RBPF) framework. It obtains the robot pose from localization data first, then creates the map. The process of Gmapping SLAM has four main steps: sampling through Lidar, calculating the weight through the information carried by the particles, resampling according to the weight and finally, estimating the map [18]. This is the most widely used SLAM algorithm [11].

Hector SLAM implements the localization and mapping at the same time. It uses the occupancy grid map, which is divided into limit grid cells, to estimate the map by each grid cell occupancy situation. During initialization, the data of the first frame will be mapped in the grid cell. Then, the data of the next frame will be matched to the previous one [20].

Karto SLAM, developed by Karto Robotics of SRI International, is a graph-based SLAM algorithm which uses highly optimized and non-iterative Cholesky matrix to calculate the solution of the sparse system. The Sparse Pose Adjustment (SPA) method is adopted to be responsible for scan matching and loop-closure procedures [11].

Cartographer SLAM, developed by Google in 2016, aims to provide the solution on map building for real-time obstacle-avoiding and path planning applications with limited computing ability such as in-door service robot (e.g. sweeping robot) [21]. The Cartographer SLAM can be located with low computing consumption, real-time optimization, but the precision is lower than other algorithms [25].

Core SLAM is an algorithm to minimize the loss of SLAM performance. The algorithm is simplified into two steps, distance calculation and map update process [26]. It is based on a simple particle filter algorithm to calculate distance and particle filter matching is used to laser and map matching. However, it requires high computing power [11].

3.2 Discussions

Trejos et al. [23] compared various 2D SLAM algorithms based on four metrics namely pose error, map accuracy, CPU usage, and memory usage. They reported that overall, Karto SLAM outperformed other 2D SLAM algorithms. Karto SLAM uses Sparse Pose Adjustment (SPA) method which has faster speed and better convergence on solving robot pose and large sparse graphs. It will have a great advantage on a large scene map building because only one pose graph needs to be maintained [27]. Therefore, we have selected Karto SLAM to further investigate it in a low-cost search and rescue robot.

The Gmapping SLAM is a good algorithm to build a map in a small scene. It can build a real-time map with high precision. Compared with Hector SLAM, it has low requirement on the Lidar precision as it uses odometer information to estimate the robot pose first. However, as the surface area increases, the particle number and information will rapidly increase, so Gmapping is not suitable for large scene map building. Yet we have selected Gmapping SLAM for further investigation as this is the most widely used SLAM algorithm in robots [11].

Trejos et al. [23] reported that Hector SLAM has the least CPU usage as it does not need to use odometer. However, compared with Gmapping, it has a high requirement of Lidar scanning frequency [28]. Yet, we have selected Hector SLAM for further investigation as it has the least CPU usage. However, through our investigation we will find out whether the scanning frequency of a typical low-cost LIDAR is sufficient for Hector SLAM.

As reported in [11], the maps built by Core SLAM are less impressive with higher error and CPU load. Therefore, Core SLAM will not be used in this research. Hess et al. [27] reported that the Cartographer SLAM was developed for indoor real-time map building, which means this algorithm may not meet the demand of search and rescue tasks because the flat and smooth indoor ground surface is different from the rough terrain formed by collapsed buildings. As a result, Cartographer SLAM will not be used in this research. We will investigate Gmapping SLAM, Hector SLAM and Karto SLAM for low-cost search and rescue applications.

4 Experimentation on Low-cost Robot

In this section, the Gmapping SLAM, Hector SLAM, and Karto SLAM are implemented on the robot to build a 2D map for the office floor (Fig. 2) with a flat surface. This section aims to test the feasibility of these three algorithms for a low-cost robot. The robot was deployed at the same location of the office floor and remotely operated and followed identical trajectory. RViz was used to construct the map.

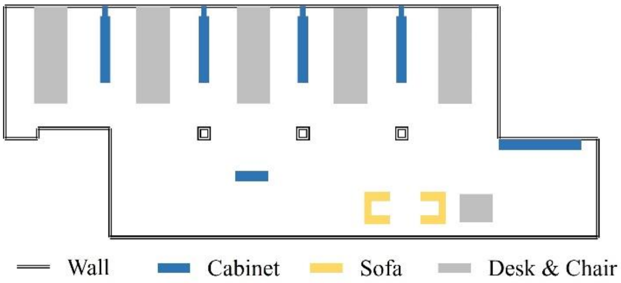

Fig. 2. 2D map of the office floor

4.1 Gmapping SLAM

From the map (Fig. 3) built by Gmapping SLAM, it can be seen that error occurs at the top-right corner. The generated map rotates the wall about 45 degrees counterclockwise. This error happens at the time when the robot makes a turn to avoid obstacle, and the odometer posts incorrect data so that the orientation predicted by the robot is also incorrect which generates the incorrect edge of wall.

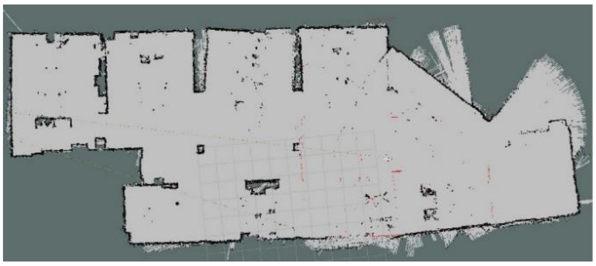

Fig. 3. Map built by Gmapping SLAM for the office floor of Fig. 3

4.2 Hector SLAM

From the map (Fig. 4) generated by the Hector SLAM, it can be seen that most of the area is overlapped. This is because Hector SLAM is based on occupancy grid map without odometer, which means this algorithm has a high requirement on the Lidar scanning frequency. The low-cost Lidar (RPLidar A1) used in this research has a maximum scanning frequency of 10 Hz which does not meet the demand of the Hector SLAM. Additionally, Hector SLAM has no ability to correcting the map so that all subsequent matches will have problems once the map makes an error [28].

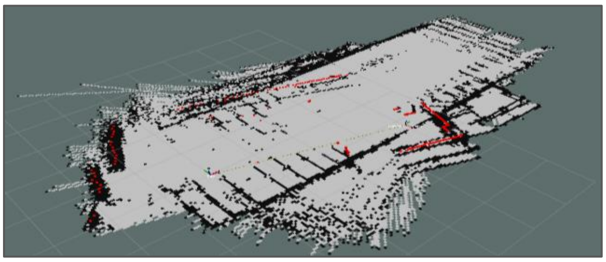

Fig. 4. Map built by Hector SLAM for the office floor of Fig. 3

4.3 Karto SLAM

The map (Fig. 5) built by Karto SLAM has good precision and resolution. However, it can be found that the bottom of the map is not closed. The robot performed a rotation at the middle of the bottom area of the map to avoid an obstacle. The odometer may have an erroneous reading at that time. The accumulation of odometer error could have contributed to the error in mapping the wall.

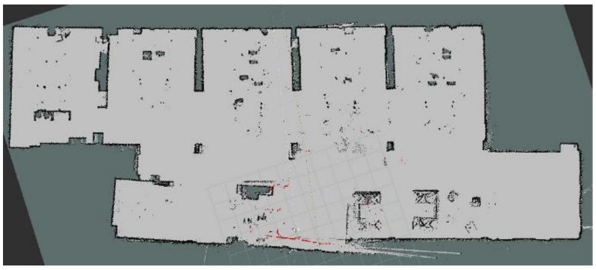

Fig. 5. Map built by Karto SLAM for the office floor of Fig. 3

4.4 Discussion

The experimental result of Fig. 4 shows that the Map built by Hector SLAM for the office floor of Fig. 3 is too far from accurate. Tee et al. [24] also reported failed map construction by Hector SLAM. Thus, Hector SLAM is not suitable for a low-cost robot including a low-cost Lidar. The experimental results of Fig. 3 and Fig. 5 suggest that the map generated using Gmapping and Karto SLAM for the low-cost robot has moderate accuracy. The next section will further investigate Gmapping and Karto SLAM for search and rescue applications.

5 Experimentation on Low-cost Robot for Search and Rescue applications

This experimentation aims to find the best SLAM algorithm in search and rescue scenarios. Gmapping SLAM and Karto SLAM will be investigated to find the best SLAM algorithm. The experiment contains three different ground surfaces in the same room (Fig. 6), flat ground, slope ground (Fig. 8), and rough ground (Fig. 10). In each environment, two SLAM algorithms are employed on map building. The area used is a rectangular room. The robot was deployed at the same location of the room and remotely operated and followed identical trajectory. RViz was used to construct the map.

5.1 Flat ground

Fig. 7 shows the maps built by Gmapping SLAM and Karto SLAM for a rectangular room when the surface is flat. The result shows that both Gmapping and Karto SLAM algorithm can build a map with a closed and clear edge in the flat ground surface. A little difference between them is that the map built by Gmapping SLAM is rough with a low resolution while the map built by Karto SLAM has a precise edge with a high resolution.

Fig. 6. Rectangular room

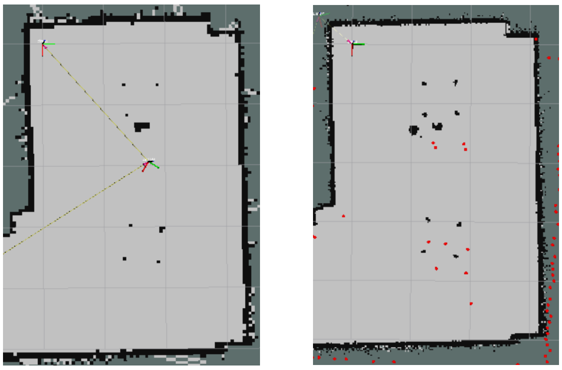

Fig. 7. Maps built by Gmapping SLAM (left) and Karto SLAM (right) for the rectangular room of Fig. 6 when the surface is flat

5.2 Slope Surface

The slope surface (Fig. 8) is made by two boxes in the lab and the gradient is about 37.5%. The maps are created using both the algorithms while the robot moved within the room using the slope surface. From the two results (Fig. 9), it can be seen that the map built by Gmapping SLAM is influenced by the slope, so the area overlap occurs in this map. On the other hand, the slope has very little influence on the map built by Karto SLAM. Therefore, in the slope scenario, the Karto SLAM is more adaptable than the Gmapping SLAM.

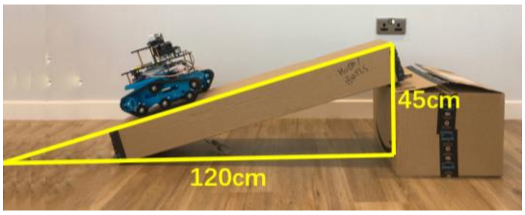

Fig. 8. Slope surface used for map building

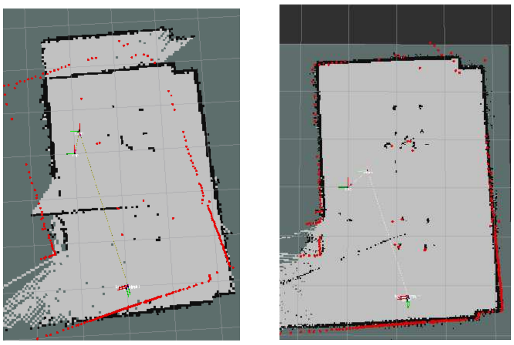

Fig. 9. Maps built by Gmapping SLAM (left) and Karto SLAM (right) for the rectangular room of Fig. 6 when the surface has a slope (Fig. 8)

5.3 Rough Terrain

The rough ground (Fig. 10) is made of several components in the lab, which aims to make the robot moving up and down in a smaller range than the slope scenario. From the maps (Fig. 11) built by each algorithm, it can be seen that there is still overlap in the map built by Gmapping SLAM, however, the result is better than the slope situation. On the other hand, the map built by Karto SLAM is still clear and precise.

Fig. 10. Rough terrain used for map building

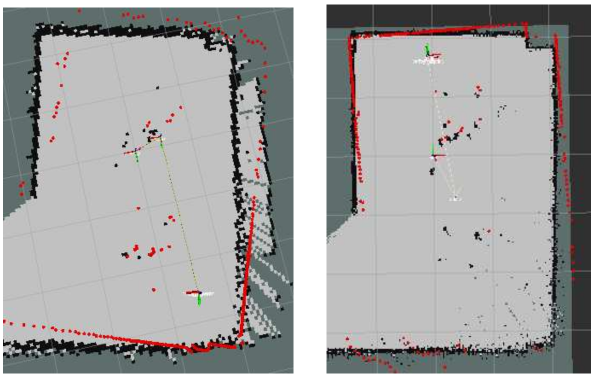

Fig. 11. Maps built by Gmapping SLAM (left) and Karto SLAM (right) for the rectangular room of Fig. 7 when the surface is rough (Fig. 11)

5.4 Discussion

Table 1 compares the performance of Gmapping and Karto SLAM for three different ground surfaces. Similar maps can be built by Gmapping and Karto SLAM in the flat ground scenario, but in the slope and rough terrain scenario, the advantage of Karto SLAM is obvious compared to Gmapping SLAM. As a result, Karto SLAM is more adaptable for the low-cost search and rescue robot in disaster scenarios.

Table 1. Performance Comparison of Gmapping and Karto SLAM

Scenarios	Gmapping SLAM	Karto SLAM
Flat ground	Constructed map has a closed edge but the resolution is low	Constructed map has a closed edge and the resolution is high
Slope surface	Constructed map has overlapped area	Constructed map has a closed edge
Rough ground	Constructed map has unidentified edge	Constructed map has a closed edge

6 Analysis

This paper investigates various SLAM algorithms for low-cost search and rescue applications and presents experimental results and comparison of various SLAM algorithms. The key findings of this research are provided below.

The experimental results show that the Hector SLAM generates an inaccurate map for the low-cost search and rescue robot even at the flat surface. It suggests that the Hector SLAM is inappropriate for low-cost applications. Hector SLAM is based on grid map which has a high requirement on Lidar scanning frequency. Lidar scanning frequency of the low-cost robot does not meet the requirement of Hector SLAM algorithm. This leads to an overlap of areas in the map generated using the Hector SLAM in this paper. The other two algorithms use odometer information as well. However, the Hector SLAM does not use odometer information. Thus, if Lidar provides noisy data Hector SLAM cannot recover from that whereas other two algorithms can.

This paper presents the results of map generation for three different ground surfaces. The results show that the worst influence on map building occurs in the slope scenario among the three situations. This is because the slope surface makes the Lidar out of the scan plane. The new plane formed by the slope surface has an angle with the original plane, which will make the distance scanned by Lidar changed. When the robot is back to the flat ground and the original plane, the scan edge cannot be closed so that the area is overlapped.

In all the experiments, Karto SLAM presents better performance than Gmapping SLAM. Thus, Karto SLAM is best SLAM for the low-cost search and rescue robot in disaster scenarios. Gmapping SLAM relies on odometer and IMU to estimate the robot pose first, which highly depends on the hardware. As a low-cost robot, the hardware cannot meet the requirements of Gmapping algorithm, which makes the error occur. However, the Karto SLAM is graph-based, which result in low influence on different ground surface.

7 Conclusions

This paper implemented three SLAM algorithms on a low-cost search and rescue robot. The result shows that the Hector SLAM algorithm is not suitable for low-cost robots while the Gmapping SLAM and Karto SLAM can create map with acceptable accuracy in flat ground. The performance of Gmapping SLAM and Karto SLAM have been further compared in three different ground surfaces where Karto SLAM performed better over Gmapping on the map resolution, precision, and stability. The Gmapping SLAM is not good enough on complicated ground surface situation, such as slope and rough terrain.

References

[1] F. Niroui, K. Zhang, Z. Kashino, and G. Nejat, ‘Deep Reinforcement Learning Robot for Search and Rescue Applications: Exploration in Unknown Cluttered Environments’, *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 610–617, Apr. 2019.

[2] B. Siciliano and O. Khatib, Eds., *Springer handbook of robotics*. Berlin: Springer, 2008.

[3] J. Liu, Y. Wang, B. Li, and S. Ma, ‘Current research, key performances and future development of search and rescue robots’, *Front. Mech. Eng. China*, vol. 2, no. 4, pp. 404–416, Oct. 2007.

[4] K. Priandana *et al.*, ‘Design of A Task-Oriented Autonomous Wheeled- Robot for Search and Rescue’, in *2018 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, Oct. 2018, pp. 259–263.

[5] L. Dang and J. Kwon, ‘Design of a new cost-effective head for a low-cost humanoid robot’, in *2016 IEEE 7th Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON)*, Oct. 2016, pp. 1–7.

[6] Z. Uddin and M. Islam, ‘Search and rescue system for alive human detection by semi-autonomous mobile rescue robot’, in *2016 International Conference on Innovations in Science, Engineering and Technology (ICISET)*, Oct. 2016, pp. 1–5.

[7] M. Li, H. Zhu, S. You, L. Wang, and C. Tang, ‘Efficient Laser-Based 3D SLAM for Coal Mine Rescue Robots’, *IEEE Access*, vol. 7, pp. 14124–14138, 2019.

[8] C. Cadena *et al.*, ‘Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age’, *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1309–1332, Dec. 2016.

[9] Y. Zhang, T. Zhang, and S. Huang, ‘Comparison of EKF based SLAM and optimization based SLAM algorithms’, in *2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, IEEE, 2018, pp. 1308–1313.

[10] P. Qi and L. Wang, ‘On simulation and analysis of mobile robot SLAM using Rao-Blackwellized particle filters’, in *2011 IEEE/SICE International Symposium on System Integration (SII)*, IEEE, 2011, pp. 1239–1244.

[11] J. M. Santos, D. Portugal, and R. P. Rocha, ‘An evaluation of 2D SLAM techniques available in Robot Operating System’, in *2013 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, Oct. 2013, pp. 1–6.

[12] G. Tuna, K. Gulez, V. Cagri Gungor, and T. Veli Mumcu, ‘Evaluations of different Simultaneous Localization and Mapping (SLAM) algorithms’, in *IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society*, Montreal, QC, Canada: IEEE, Oct. 2012, pp. 2693–2698. Accessed: Sep. 14, 2019. [Online]. Available: http://ieeexplore.ieee.org/document/6389151/

[13] B. M. F. da Silva, R. S. Xavier, T. P. do Nascimento, and L. M. G. Gonsalves, ‘Experimental evaluation of ROS compatible SLAM algorithms for RGB-D sensors’, in *2017 Latin American Robotics Symposium (LARS) and 2017 Brazilian Symposium on Robotics (SBR)*, Curitiba: IEEE, Nov. 2017, pp. 1–6. Accessed: Sep. 14, 2019. [Online]. Available: http://ieeexplore.ieee.org/document/8215331/

[14] S. Khan, D. Wollherr, and M. Buss, ‘Modeling laser intensities for simultaneous localization and mapping’, *IEEE Robot. Autom. Lett.*, vol. 1, no. 2, pp. 692–699, 2016.

[15] M. Quigley *et al.*, ‘ROS: an open-source Robot Operating System’, *ICRA Workshop Open Source Softw.*, p. 6, Jan. 2009.

[16] I. H. Shanavas, P. B. Reddy, and M. C. Doddegowda, ‘A Personal Assistant Robot Using Raspberry Pi’, in *2018 International Conference on Design Innovations for 3Cs Compute Communicate Control (ICDI3C)*, Apr. 2018, pp. 133–136.

[17] Z. Gong, J. Li, and W. Li, ‘A low cost indoor mapping robot based on TinySLAM algorithm’, in *2016 IEEE International Geoscience and Remote Sensing Symposium*, Jul. 2016, pp. 4549–4552.

[18] G. Grisetti, C. Stachniss, and W. Burgard, ‘Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters’, *IEEE Trans. Robot.*, vol. 23, no. 1, pp. 34–46, Feb. 2007.

[19] Y. Abdelrasoul, A. B. S. H. Saman, and P. Sebastian, ‘A quantitative study of tuning ROS gmapping parameters and their effect on performing indoor 2D SLAM’, in *2016 2nd IEEE International Symposium on Robotics and Manufacturing Automation (ROMA)*, Sep. 2016, pp. 1–6.

[20] N. Yu and B. Zhang, ‘An Improved Hector SLAM Algorithm based on Information Fusion for Mobile Robot’, in *2018 5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS)*, Nov. 2018, pp. 279–284.

[21] A. Filatov, A. Filatov, K. Krinkin, B. Chen, and D. Molodan, ‘2D SLAM quality evaluation methods’, in *2017 21st Conference of Open Innovations Association (FRUCT)*, Nov. 2017, pp. 120–126.

[22] B. Steux and O. E. Hamzaoui, ‘tinySLAM: A SLAM algorithm in less than 200 lines C-language program’, in *2010 11th International Conference on Control Automation Robotics Vision*, Dec. 2010, pp. 1975–1979.

[23] K. Trejos, L. Rincón, M. Bolaños, J. Fallas, and L. Marín, ‘2D SLAM Algorithms Characterization, Calibration, and Comparison Considering Pose Error, Map Accuracy as Well as CPU and Memory Usage’, *Sensors*, vol. 22, no. 18, p. 6903, 2022.

[24] Y. K. Tee and Y. C. Han, ‘LiDAR-based 2D SLAM for mobile robot in an indoor environment: A review’, in *2021 International Conference on Green Energy, Computing and Sustainable Technology (GECOST)*, IEEE, 2021, pp. 1–7.

[25] W. Hess, D. Kohler, H. Rapp, and D. Andor, ‘Real-time loop closure in 2D LIDAR SLAM’, in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, Stockholm, Sweden: IEEE, May 2016, pp. 1271–1278. Accessed: Sep. 01, 2019. [Online]. Available: http://ieeexplore.ieee.org/document/7487258/

[26] L. Carlone, R. Aragues, J. A. Castellanos, and B. Bona, ‘A linear approximation for graph-based simultaneous localization and mapping’, *Proc Int Conf Robot. Sci. Syst.*, p. 2011.

[27] K. Konolige, G. Grisetti, R. Kümmerle, W. Burgard, B. Limketkai, and R. Vincent, ‘Efficient Sparse Pose Adjustment for 2D mapping’, in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2010, pp. 22–29.

[28] S. Kohlbrecher, O. von Stryk, J. Meyer, and U. Klingauf, ‘A flexible and scalable SLAM system with full 3D motion estimation’, in *2011 IEEE International Symposium on Safety, Security, and Rescue Robotics*, Nov. 2011, pp. 155–160.