Johnson, P, Nguyen, QH and Latham, D

Performance evaluation of ROS-based SLAM algorithms for handheld indoor mapping and tracking systems

Article

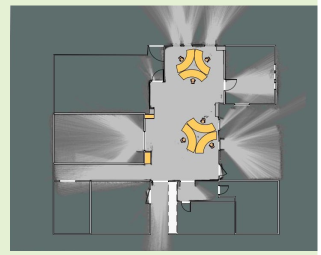
Citation (please note it is advisable to refer to the publisher's version if you intend to cite from this work)

Johnson, P ORCID logoORCID: https://orcid.org/0000-0003-2379-9700, Nguyen, QH and Latham, D (2022) Performance evaluation of ROS-based SLAM algorithms for handheld indoor mapping and tracking systems. IEEE Sensors Journal. 23 (1). pp. 706-714. ISSN 1530-437X

Performance evaluation of ROS-based SLAM algorithms for handheld indoor mapping and tracking systems

Quang Huy Nguyen, Princy Johnson, *Senior Member, IEEE* and David Latham.

***Abstract*— Simultaneous Localization and Mapping is an important field of work not only in robotics, but also in mobile platforms. This research work provides insight into how SLAM techniques are deployed in an indoor environment to aid first responders with their duties. Due to the hazardous nature of the environment and the need for sensitivity due to potential involvement of human subjects, autonomous robots cannot be used. So, the first responders must carry the scanning equipment and perform SLAM at the same time. As a result, unlike standard robot platforms, there will be no reliable odometry source, and SLAM will have to deal with the user's unpredictable movement. In this work, we compare and examine ROS-based SLAM approaches without using any odometry for their application in the above-mentioned circumstances. Gmapping, HectorSLAM, and Cartographer have been chosen as the candidates for this evaluation. We evaluated these approaches in two different environments: a lab office, and a long corridor. The research results show that Cartographer outperforms the other two techniques in our test setup in terms of map quality and trajectory tracking. The Cartographer's mapping error ranged from 0.017m to 0.3548m.**



***Index Terms*— Cartographer, Gmapping, HectorSLAM, Indoor mapping, LiDAR, ROS, SLAM.**

I. Introduction

SIMULTANEOUS Localization and Mapping (SLAM) [1] is a prominent topic among mobile robotics scientists. In environments such as, indoor or underground environments, where the Global Positioning System (GPS) does not have the reach or provide the necessary resolution [2-3], it would be useful to be able to generate and track a map of the region with non-GPS technology. This is one of the most active indoor positioning issues, attracting many researchers who have provided various solutions. [4-5]. When SLAM is used for tracking and mapping in a given space, dealing with sensor accumulative error is a conundrum for academics and engineers [6]. Furthermore, in operations such as firefighting, exploration, military, or rough terrain where robots cannot be used, human beings involved in the operation need to carry the equipment containing sensors that execute SLAM while performing their duties [7]. Additionally, unlike robots, human movements may be irregular and unpredictable, especially while carrying out firefighting tasks [8]. As a result, odometry free SLAM has become one of the most challenging issues to tackle.

Today, Robotic Operating System (ROS) [9] is one of the most widely used robotic framework for developing automated data collection and decision making for indoor mapping space [10]. It is commonly used by both beginners and expert users, as well as startup companies to study robotic problems, and for rapid prototyping [11-12]. ROS provides libraries, drivers, and toolkits to help solve robotics problems efficiently, and it enables the development of both simulation and real-world applications. The SLAM techniques used in this research study, such as Gmapping [13], HectorSLAM [14], and Cartographer [15], all perform well within ROS environment.

Most SLAM systems are equipped with LiDAR or stereo cameras. In addition, RADAR, SONAR, depth camera and Inertial Measurement Unit (IMU) are often used to improve the technique's accuracy. While each sensor has its own benefits and limitations, most studies show that LiDAR, though expensive, is superior to other sensors for high-precision mapping.

The purpose of this research work was to compare and evaluate the performance of ROS-compatible SLAM algorithms based on low-cost LiDAR in wearable and handheld systems. The achieved result is the first step toward our ultimate goal of developing an aid system for emergency fire rescue operations when the environment contains smoke and dust. As a result, examining the most widely used methods is essential, and we believe our study will act as a guideline for our future approaches as well as for other researchers

This work is supported and funded by the Innovate UK (KTP Project 1162). The partner industry is CAL International, and Liverpool John Moores University is the academic partner. Hardware assembly was done by CAL International.

Q. H. Nguyen is with the Liverpool John Moores University, Liverpool, UK and CAL International, Prescot, UK (e-mail: q.h.nguyen@ljmu.ac.uk).

P. Johnson, is with School of Engineering, Liverpool John Moores University, Liveprool, UK. (e-mail: p.johnson@ljmu.ac.uk).

D. Latham is with CAL International, Prescot, UK (e-mail: david.latham@cal-international.com).

working in odometry-free SLAM techniques and ROS in general. Sections 2 and 3 of this paper cover related research and background. Section 4 follows with a description of the system configuration and environment. Section 5 presents and discusses the experimental results. Finally, section 6 concludes with findings and future directions.

II. Related work

In this section, we will review the studies that compared and assessed multiple SLAM techniques, with a particular focus on the LiDAR-based SLAMs used for this research experimentation. Furthermore, as previously mentioned, this research is the initial step toward the development of an assistance system for first responders. Therefore, a comparative evaluation of studies using SLAM techniques that do not strictly rely on odometry or that are deployed on unstable platforms such as handheld and wearable systems will be considered.

There has been a lot of recent studies comparing different SLAM systems for indoor environments, such as [16-18]. The performance of LiDAR-based SLAMs in mobile robotics applications such as Gmapping, KartoSLAM [19], HectorSLAM, CoreSLAM [20], and LagoSLAM [21] were evaluated by the authors in [16]. They demonstrated that the last three techniques provide higher-quality maps. They did emphasize, however, that this is largely due to the efficiency of wheel odometry data. The study in [17] evaluated three SLAM methods, CoreSLAM, Gmapping, and HectorSLAM, using simulations to identify the most suitable technique for military UAVs operating in a variety of terrains. The authors of [18] investigated the performance of Gmapping and Cartographer for autonomous vehicles. In [22], the authors proposed a FastSLAM which has a similar foundation as Gmapping since both use Rao-Blackwellised particle filter [13]. Both [18] and [22] concluded that their approach produced satisfactory results but only with the help of IMU and very reliable wheel odometry data. Authors of [23] described a dual LiDAR system paired with IMU and wheel odometry to achieve mapping accuracy of up to 4cm. Besides, Ali and co-authors focused on the long-term operation of the SLAM process in [24]. They suggested a novel method based on adaptive local mapping algorithms. The experiment using test data yielded results with greater than 90% precision and significantly lower CPU utilisation when compared to typical SLAM methods.

The authors of [25-27] presented several techniques for generating 3D mapping using handheld devices, which employed odometry-free SLAM. The authors not only utilized LiDAR but also integrated it with additional sensors such as an RGBD camera or an IMU sensor to improve accuracy of scanning process in an indoor environment. In [28], Zhou and colleagues proposed a novel plane adjustment for LiDAR in the interior environment. Though the final outcome showed that its performance surpassed that of its competitions, a powerful CPU was required to process the data. In [29], authors ultilized GPS and LiDAR-based SLAM to overcome the odometry-free challenges. Also a camera was required to complete the mapping process. The root mean square error achieved was 3.14m. Zhou *et al.* in [30] proposed a fusion of LiDAR with Ultra Wide Band (UWB) to solve the odometry-free and multi-robot mapping problems. The accuracy error was in the 0.1-0.4m range. However, this technique necessitated prior preparation and in-depth knowledge of the surroundings.

As can be seen, a major part of the research has concentrated on mapping for robots when odometry data is available. Furthermore, odometry-free techniques are being developed for 3D scanning applications, which need the employment of expensive sensors. In the context of this study, we consider three well-known LiDAR-based SLAM algorithms for handheld or wearable systems. The performance of these techniques will be compared in terms of the accuracy of the occupancy grid mapping. The main objective of this paper is to provide an overview of the strengths and weaknesses of all three ROS algorithms mentioned above, and to provide a set of guidelines for ROS users to select an algorithm that best suits their individual odometry-free application requirements.

III. Mapping with ROS

ROS [9] is a specialized open source software platform that is used to program and control robots. ROS includes libraries, programming tools, graphical tools, tools for direct control communication with hardware, and libraries for data retrieval from sensors and devices.

ROS's programming environment makes it easier to construct complex robot capabilities, such as teleoperation and navigation. ROS functions similar to a network made up of several nodes, each with a specific function that corresponds to the robot's parts. Rather than agreeing to use the same programming language, each node can be created and coded in accordance with the developer's concept. C++ can be used to create one node, whereas Python can be used to create another node. As a result, developers no longer need to be concerned about the robot's hardware because ROS provides a common interface for operating the robot's hardware. Thus, instead of dealing with the specific hardware API, the software becomes the primary focus. This makes robot software development considerably easier, independent, and versatile to use.

The focus of this research is on LiDAR-based SLAM. Many SLAM techniques were created in the ROS ecosystem and are widely used in both research and industry environments. Several SLAM techniques, such as HectorSLAM, Gmapping, Cartographer, LagoSLAM, and CoreSLAM, have been developed and widely used in ROS. Many researchers have shown that HectorSLAM, Gmapping, and Cartographer outperform other options for autonomous robotics in indoor environment. Gmapping is one of the most popular SLAM packages developed by Griseetti et al [13]. Gmapping is a laser-based method that employs the Rao-Blackwellized particle filter [13] SLAM approach. Gmapping is usually suggested for use with precise odometry data, such as wheel odometry, to assure accuracy. HectorSLAM is a SLAM method that makes use of the Extended Kalman Filter (EKF) [14]. The high update rate and low noise output of a high-end LiDAR are utilized to predict the robot's movements in real time. Although odometry data may not be used, the Inertial Measurement Unit (IMU) can be used for 3D state estimation. However, HectorSLAM may have problems with low-end LiDARs because its output is often noisy and has a slow update rate. Cartographer is a graph-based SLAM system created in 2016 [15]. It reconstructs the environment using two subsystems: global SLAM and local SLAM. In the most basic scenario, with 2D mapping, the Cartographer requires only the laser scanner and no odometry data. However, this is a technique that can deplete a large amount of resources. The authors of [31] and [32] compared the capabilities of Gmapping, HectorSLAM, and Cartographer for robot indoor mapping. The results showed that the Cartographer performs well with the robot. However, as previously stated, estimating human mobility is significantly more difficult. Hence, this research article specifically considers this circumstance.

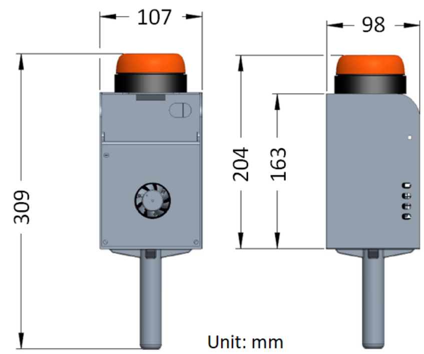
IV. Odometry free SLAM experimentation

A. System setup and environment

To evaluate the performance of each of the LiDAR based SLAM techniques for the handheld devices in the indoor environment, we performed an extensive set of experiments. We developed a handheld device that can do two functions: data collection and data processing. RPLIDAR A2 [33] was the LiDAR utilized in the experiment. It is a low-cost scanner capable of 360-degree scanning within 10m and collecting up to 8000 points per second. Fig. 1a shows the picture of our finished prototype, while Fig. 1b depicts the dimensions of our device in millimeters. The LiDAR and processor are powered by a custom designed 3Ah battery, which allows our prototype to function continuously for more than 1 hour. The enclosure was 3D printed and the prototype weighs nearly 1100 grams in total, which is comparable to the weight of a lightweight modern ultrabook. Thus, our prototype is portable and can be carried comfortably by hand. Table I presents its specifications and our experimental setup. Gmapping, HectorSLAM, and Cartographer have been chosen as the three representations. As previously stated, these three solutions are considered because they are widely used in research and industry.

We conducted the experiment in two different environments and with two different modes of movement to validate its reliability. The first environment was a real lab office with numerous objects and complex lighting. The second location was a long corridor at Liverpool John Moores University. Long corridor SLAM is a typical issue that many SLAM systems deal with. Normal walking with smooth rotations, and quick walking with sharper rotations were the two movement types. We attempted to recreate the handheld system's ambiguity and unpredictability using these two types of movement. This is also the most significant distinction between a portable device and a robot system.

(a)



(b)

Fig. 1: Handheld SLAM system prototype: (a) Built prototype. (b) Prototype dimension.

TABLE I: System setup

Parameters	Configuration
Hardware	Jetson Xavier [34]
Processor	6-Core Carmel ARM v8.2
GPU	NVIDIA Volta architecture
RAM	8GB
LiDAR	RPLiDAR A2
OS	Ubuntu 18.04
ROS	Melodic
Walking Mode 1	Standard speed, smooth rotation
Walking Mode 2	Fast speed, sharp rotation

B. Implementation and results

1) Scene 1: Lab office: The system was used to map the office environment; laser readings, trajectory, and true landmarks were recreated in real time. The user traced a pre-defined path with the device in two different modes: normal walking pace with smooth rotations, and fast walking with abrupt rotations. The lab office that we used for the experimentation is seen in Fig. 2 and 3. Walking path is indicated by the dashed red line. Our lab office environment includes a mixture of objects of varying heights, and the lighting situation is complicated. Light comes from both natural sources such as sunlight passing through large windows and artificial sources such as electric bulb lights and working monitors. There are also glass doors, as illustrated in Fig 3b.

The outcome will be visually evaluated as this is the user's initial impression of the system. The Root Mean Square Error Formula will be used for additional analysis:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(d_e - d_t)^2} \tag{1}$$

where n is the number of sampled points, d_e is the distance estimated by the SLAM approach and d_t is the true distance.

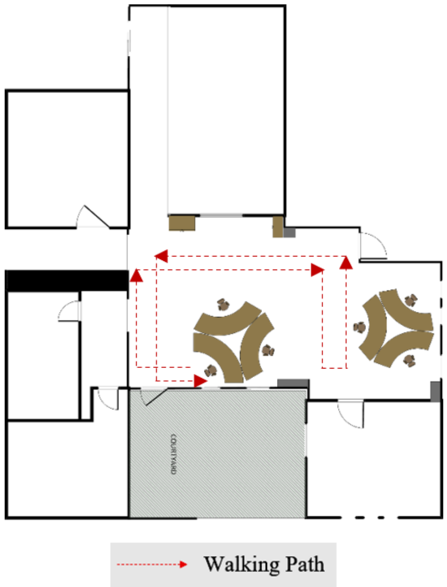
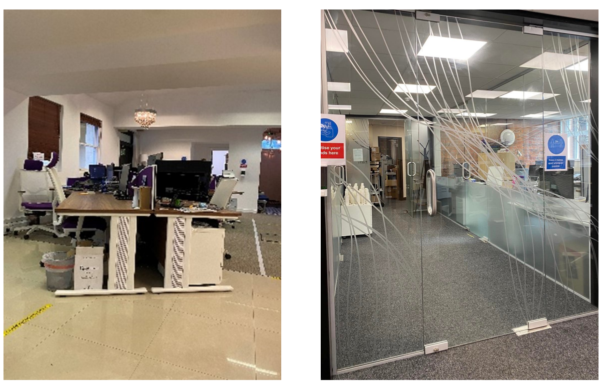


Fig. 2: Lab office floor plan and walking path.



(a) (b)

Fig. 3: Sample views of the lab office environment used in the experimentation: (a) Objects and furniture. (b) Glass doors.

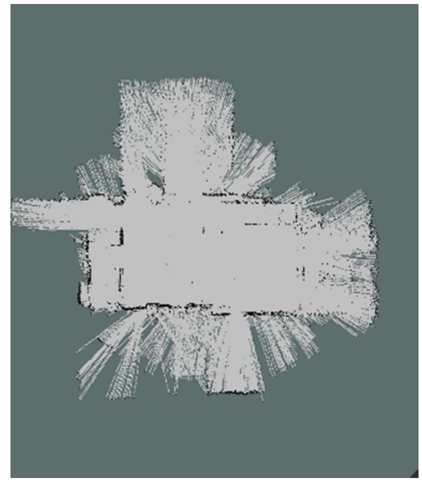
Noting that we calculate this value only when the mapping process was defined as successful.

The result of mapping and tracking in this scene is presented visually in Fig. 4 to 6. RMSE value is shown in Table II.

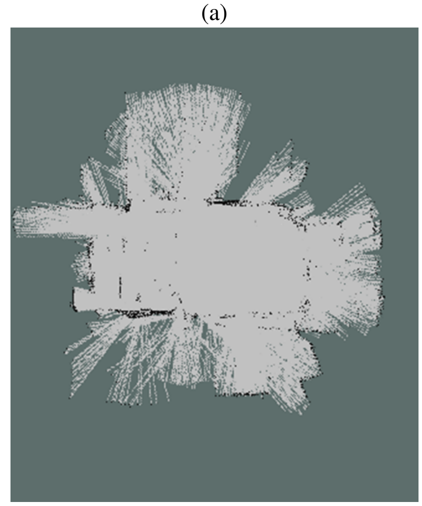
2) Scene 2: long corridor: The system was tested in the LJMU corridor, as shown in Fig. 7 and 8, in the same way that it was examined in the complex office. The walking path is represented as red dashed line in Fig. 7. Similar to the above experimentation, the system was tested using the two walking modes as in the complex office environment. Fig. 9 to 11 present our findings. Table III shows the accuracy calculated using RMSE metric.

TABLE II: Accuracy comparison complex office

Walking Mode	RSME (m)		
	Gmapping	HectorSLAM	Cartographer
Mode 1	N/A	0.454	0.1864
Mode 2	N/A	N/A	0.2789



(a)



(b)

Fig. 4: Maps of the lab office environment produced by the prototype using Gmapping SLAM technique when using two different walking modes: (a) Walking mode 1: standard speed, smooth rotation. (b) Walking mode 2: fast speed, sharp rotation.

V. Discussion of Results

A. Mapping Quality

1) Scene 1: lab office: Visual inspection reveals that Gmapping produced poor results and was incapable of producing a comprehensive map. This is due to the fact that Gmapping requires odometry information, and in our configuration, laser data was transformed to odometry using the ROS package called "laser_scan_matcher" [35]. As can be seen, this was insufficient to assist Gmapping in completing its map, especially in the event of a sharp rotation.

In the first walking mode, when the results from HectorSLAM and Cartographer are compared, it is evident that both approaches produce exceptionally good maps as shown in Fig. 4a and 5a with the RMSE of Cartographer was impressive at about 0.017m while the HectorSLAM's value is not bad at 0.454 m. The discrepancy was most noticeable at the corners

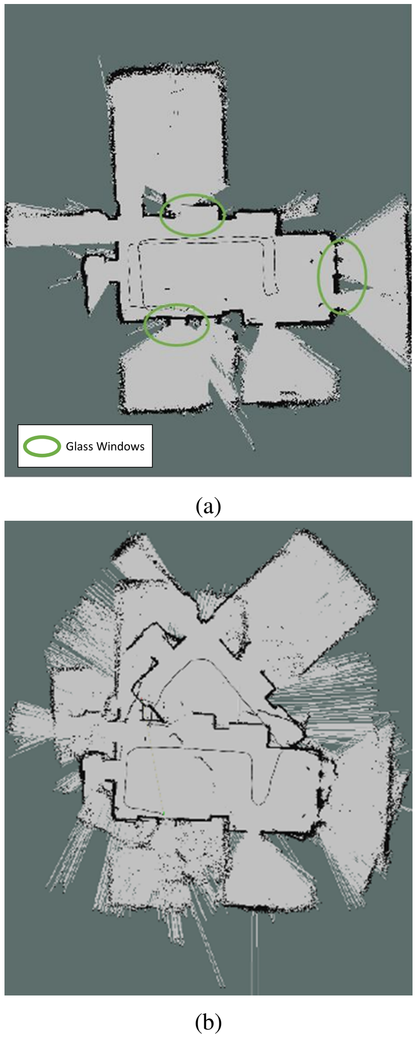


Fig. 5: Maps of the lab office environment produced by the prototype using HectorSLAM technique when using two different walking modes: (a) Walking mode 1: standard speed, smooth rotation. (b) Walking mode 2: fast speed, sharp rotation.

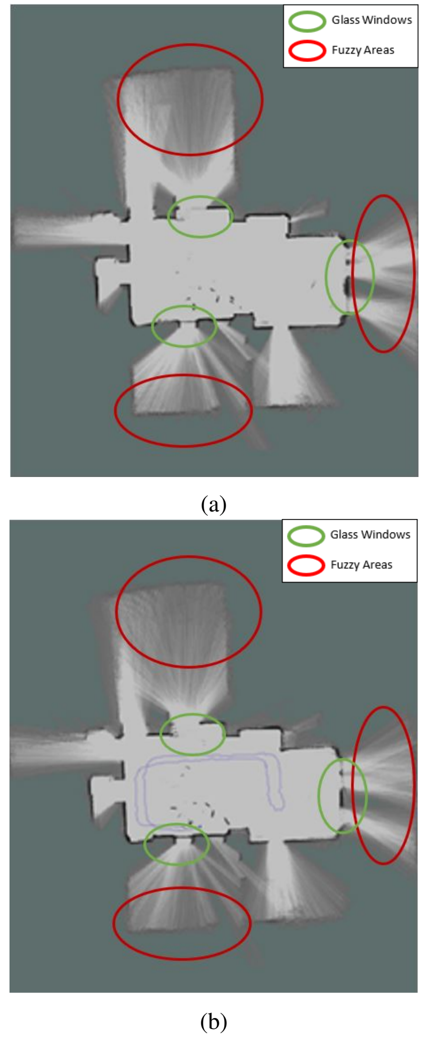


Fig. 6: Maps of the lab office environment produced by the prototype using Catographer SLAM technique when using two different walking modes: (a) Walking mode 1: standard speed, smooth rotation. (b) Walking mode 2: fast speed, sharp rotation.

of walls, where Cartographer is able to display considerably sharper angles with greater accuracy, and a percentage of occupied grids that was around 20% lower than HectorSLAM at these spots. On the other hand, the Cartographer gave the map a slightly blurry appearance in red circled area. Interestingly, this wasn't a bad thing at least in our experiment. In Fig. 5 and 6, the glass window is circled in green, and the fuzzy areas represent the sections behind the window where there is no direct walking route for the operator to access. In fact, LiDAR is the sensor that cannot detect glass, allowing it to look through glass windows. The presence of a fuzzy area in Cartographer's output implies that these locations have not yet been validated as actual landmarks. As a result, it is understandable that there may be some glass objects in the region. Cartographer technique appears to be potentially better at recognizing genuine landmarks inside the user's perspective as a result of their usage of submaps and global maps. For our future work, we will target this problem, which is to clarify real landmark and other moveable objects with the help of other sensors.

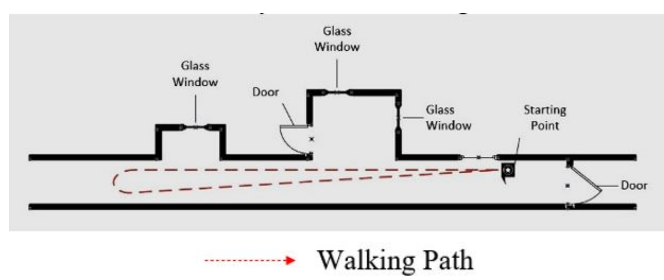
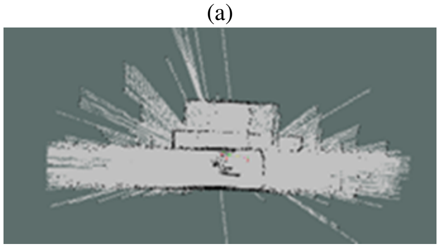


Fig. 7: Long corridor floor plan and walking path.

The distinction though, was obvious in the second walking mode. In this case, the Cartographer generated a quite similar map to that of the first walking mode, thus proving its ability to behave consistently in our setup. HectorSLAM was unable to complete the mapping, as shown in Fig. 4b. This is because HectorSLAM was designed with a considerably higher spec LiDAR than the RPLIDAR A2 we used. As a result, the

Fig. 8: Sample view of the long corridor used in the experimentation.

(a)



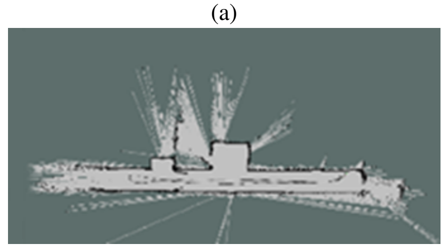
(b)

Fig. 9: Maps of the long corridor environment produced by the prototype using Cartographer SLAM technique when using two different walking modes: (a) Walking mode 1: standard speed, smooth rotation. (b) Walking mode 2: fast speed, sharp rotation.

resolution of our low-cost system is significantly lower than that required by the HectorSLAM technique. In the second walking mode, the user's speed was increased, and the turning curves were made much sharper. As a result, HectorSLAM accumulated errors, leading to the map breaking down after the fourth turn.

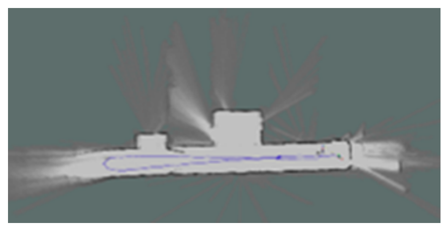
2) Scene 2: long corridor: The long corridor results are unsurprising, with Cartographer still delivering the highest performance. Meanwhile, Gmapping's mapping is still incomplete. More specifically, in both walking modes, the maps generated by Gmapping are quite noisy, with the walls and the room adjacent to the hallway not correctly recognized and reconstructed.

(a)

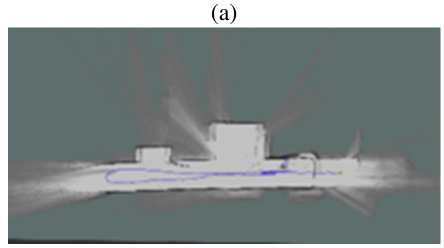


(b)

Fig. 10: Maps of the long corridor environment produced by the prototype using HectorSLAM technique when using two different walking modes: (a) Walking mode 1: standard speed, smooth rotation. (b) Walking mode 2: fast speed, sharp rotation.



(a)



(b)

Fig. 11: Maps of the long corridor environment produced by the prototype using Cartographer SLAM technique when using two different walking modes: (a) Walking mode 1: standard speed, smooth rotation. (b) Walking mode 2: fast speed, sharp rotation.

In the two walking modes, HectorSLAM performed differently and interestingly. In the first walking mode, HectorSLAM was unable to fully reconstruct the area. The two rooms adjacent to the corridor, in particular, were mapped twice: once when the user walked forward and again when the user returned to the starting point. Both of these reconstructed rooms were in the wrong spot. Their difference from the actual location is around 1.2m and 2.5m, respectively. This can be explained as starting with a typical long corridor area, HectorSLAM struggles to recognize its position in an environment with few differentiating features. Surprisingly, the room

TABLE III: Accuracy comparison long corridor

Walking Mode	RSME (m)		
	Gmapping	HectorSLAM	Cartographer
Mode 1	N/A	N/A	0.017
Mode 2	N/A	1.2878	0.3548

repeating issue did not occur in the second walking mode. The quick movement speed allowed the user to enter the room early, assisting HectorSLAM in the mapping task. Regardless, the position of both rooms is incorrect in comparison to the actual floor plan. As a result, the RSME in this situation is fairly high, at around 2.857m.

Whereas, the Cartographer performed admirably, with the RSME only around 0.007m. It can also be noticed that the Cartographer did better in walking mode 1 than in walking mode 2. In walking mode 2, the RMSE was 0.134m.

In general, Cartographer, produced the best mapping results consistently in both circumstances. More crucially, in both movement scenarios, the Cartographer's error was minimal, and its performance was consistent.

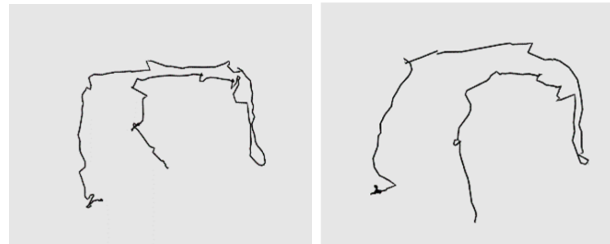
B. Trajectory analysis

Unlike when using a robot or a simulation, when human subjects are used to carry the equipment, it is impractical to preserve all tests with a perfectly matching walking path for mathematical analysis, due to the nature of human movement and the desire to keep walking as natural as possible. Therefore, the trajectory will be examined visually, and we will concentrate on overall tracking pattern, particularly turning around the corners. In addition, significant distinctions will be highlighted and analyzed. Fig. 12 to 17 depict the trajectory tracking results.

Gmapping's trajectory tracking was lost in the office environment since it was unable to complete its map. In the first walking mode, Cartographer and HectorSLAM track the user well and closely to ground reality. A detailed examination of the tracking records reveals that the Cartographer's trajectory has three sudden spikes shown by the black circles in Fig. 14a. This could be due to the sudden turning movement of the user. This, however, was not shown by HectorSLAM.

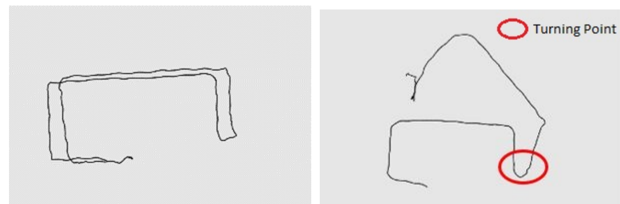
Nonetheless, while HectorSLAM lost its map and tracking after the fourth turn (red circled area in Fig. 13b) in the second walking mode, Cartographer's was able to follow very closely to the floor plan. It is also interesting to see that the spikes did not appear in this mode. Higher walking speed seems to reduce the effect of user's hand movement. As can be seen, Cartographer is the sole viable option, and it runs admirably in this walking mode.

In the long corridor, Gmapping was once again unable to finish its trajectory tracking. This demonstrates that our conversion of laser scan data to odometry information is insufficient for Gmapping in a handheld system. HectorSLAM and Cartographer, on the other hand, show comparable results. Both techniques track the walking path extremely effectively and close to the actual path in the first walking mode. It



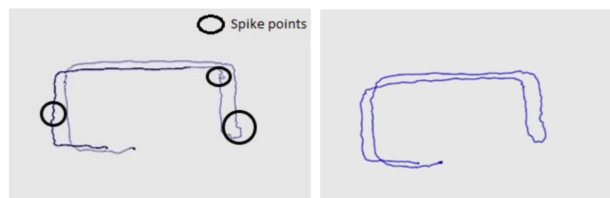
(a) (b)

Fig. 12: Gmapping trajectory in the lab office: (a) Walking mode 1: standard speed, smooth rotation. (b) Walking mode 2: fast speed, sharp rotation.



(a) (b)

Fig. 13: HectorSLAM trajectory in the lab office: (a) Walking mode 1: standard speed, smooth rotation. (b) Walking mode 2: fast speed, sharp rotation.



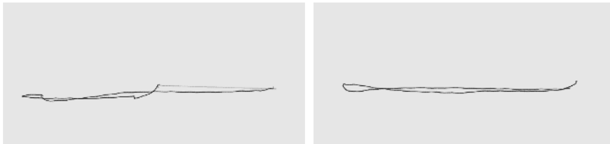
(a) (b)

Fig. 14: Cartographer trajectory in the lab office: (a) Walking mode 1: standard speed, smooth rotation. (b) Walking mode 2: fast speed, sharp rotation.

(a) (b)

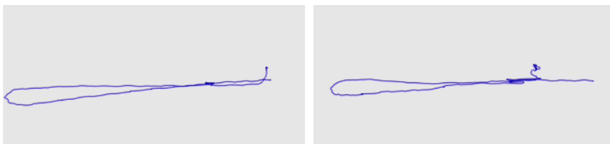
Fig. 15: Gmapping trajectory in long corridor: (a) Walking mode 1: standard speed, smooth rotation. (b) Walking mode 2: fast speed, sharp rotation.

is worth noting that Cartographer lost the tracking near the completion of the walking path in walking mode 2. The tracking ability was impacted by a sharp rotation when the user turned around. Surprisingly, even if HectorSLAM's map is inaccurate, its tracking capabilities remain intact. Overall, in this experiment, HectorSLAM and Cartographer performed comparably and provide sufficiently accurate results, especially the Cartographer when we take into consideration of trajectory tracking within the generated map.



(a) (b)

Fig. 16: HectorSLAM trajectory in long corridor: (a) Walking mode 1: standard speed, smooth rotation. (b) Walking mode 2: fast speed, sharp rotation.



(a) (b)

Fig. 17: Cartographer trajectory in long corridor: (a) Walking mode 1: standard speed, smooth rotation. (b) Walking mode 2: fast speed, sharp rotation.

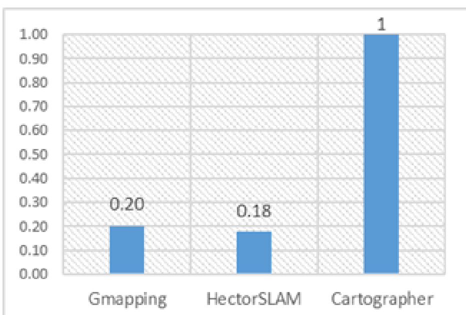


Fig. 18: CPU Usage - normalized to Cartographer's value.

C. CPU Usage

CPU usage was recorded while the experiments were conducted. This record is shown in Fig. 18. Cartographer was used to normalize the results. Cartographer, as can be seen, consumed the most CPU power. When compared to the other two procedures, this value is five times higher. HectorSLAM and Gmapping consume relatively little CPU power, and HectorSLAM has been proved to be the most efficient approach in terms of computing resource utilization. Given how each strategy works [6-8], this is not surprising.

VI. Conclusion

In this paper, we have presented an experimental evaluation of three types of SLAM techniques (Gmapping, HectorSLAM and Cartographer) for mapping and tracking of first responders within unprepared indoor environment where GPS or Internet is not available. The conditions considered are in-line with that of experienced by the first responders, which is the focus of our research.

Purpose built hand-held sensor system prototype was made in-house at CAL International for the evaluation. Experimentations were carried out in two different indoor environments for comparison purposes. A thorough discussion of the experimental results is presented above, and the summary is presented below.

Out of the three SLAM techniques, Cartographer excelled in all scenarios.This approach is also robust to sudden movements and changes in direction. In our test, the Cartographer's RMSE was as low as 0.017m, and the largest RMSE was just around 0.35m. This error is within the acceptable margin for majority of circumstances faced by the first responders, in hazardous indoor environments. However, the Cartographer technique has not been optimized for CPU performance.

HectorSLAM on the other hand, provides the second-best mapping results, but its performance suffers when dealing with sudden shifts and long corridor areas. HectorSLAM's CPU utilization however, was the best of the three choices, which is a major selling point. Due to the lack of a reliable odometry source, Gmapping was unable to complete the mapping in the office setting.

These results may be influenced by the parameter choices. The optimum settings can only be determined by the actual environment. Our experimental study demonstrated that Cartographer is a solid choice for producing 2D maps using low-cost LiDAR in odometry-free handheld systems. As a result of this finding, we have decided to adopt Cartographer as the primary technology for the next stage of our research, which will be a 2D mapping system with multiple agents.

References

[1] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part I," *IEEE Robotics Automation Magazine*, vol. 13, no. 2, pp. 99–110, Jun. 2006.

[2] B. Jang and H. Kim, "Indoor Positioning Technologies Without Offline Fingerprinting Map: A Survey," *IEEE Communications Surveys Tutorials*, vol. 21, no. 1, pp. 508-525, Firstquarter 2019, DOI: 10.1109/COMST.2018.2867935.

[3] G. Dedes and A. G. Dempster, "Indoor GPS positioning - challenges and opportunities," *VTC-2005-Fall. 2005 IEEE 62nd Vehicular Technology Conference*, 2005., 2005, pp. 412-415, DOI: 10.1109/VETECF.2005.1557943.

[4] K. Chintalapudi, A. P. Iyer, and V. N. Padmanabhan, "Indoor localization without the pain," *In Proceedings of the sixteenth annual international conference on Mobile computing and networking (MobiCom '10)*, 2010, New York, NY, USA, pp. 173–184.

[5] A. Yassin *et al.*, "Recent Advances in Indoor Localization: A Survey on Theoretical Approaches and Applications," *IEEE Communications Surveys Tutorials*, vol. 19, no. 2, pp. 1327-1346, Secondquarter 2017, DOI: 10.1109/COMST.2016.2632427.

[6] L. Huang, "Review on LiDAR-based SLAM Techniques," *2021 International Conference on Signal Processing and Machine Learning (CONF-SPML)*, 2021, pp. 163-168, DOI: 10.1109/CONF-SPML54095.2021.00040.

[7] J. Rantakokko *et al.*, "Accurate and reliable soldier and first responder indoor positioning: multisensor systems and cooperative localization," *IEEE Wireless Communications*, vol. 18, no. 2, pp. 10-18, April 2011, DOI: 10.1109/MWC.2011.5751291.

[8] L. Faramondi *et al.*, "An enhanced indoor positioning system for first responders," *International Conference on Indoor Positioning and Indoor Navigation*, 2013, pp. 1-8, DOI: 10.1109/IPIN.2013.6817921.

[9] Ros.org — Powering the world's robots [Online]. Available: https://www.ros.org, Accessed on: Feb. 5, 2022

[10] B. Kehoe, S. Patil, P. Abbeel and K. Goldberg, "A Survey of Research on Cloud Robotics and Automation," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 2, pp. 398-409, April 2015, DOI: 10.1109/TASE.2014.2376492.

[11] A. Banos *et al.*, "Hexapod robotic system for indoor neutron and gamma radiation mapping and inspection," *2022 25th International Symposium on Design and Diagnostics of Electronic Circuits and Systems (DDECS)*, 2022, pp. 48-53, DOI: 10.1109/DDECS54261.2022.9770114.

[12] L. Su, G. Qiu, W. Tang and M. Chen, "A ROS Based Open Source Simulation Environment for Robotics Beginners," *2021 6th International Conference on Robotics and Automation Engineering (ICRAE)*, 2021, pp. 286-291, DOI: 10.1109/ICRAE53653.2021.9657761.

[13] G. Grisetti, C. Stachniss and W. Burgard, “Improved techniques for grid mapping with Rao-Blackwellized particle filters,” *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 34–46, Feb. 2007.

[14] S. Kohlbrecher *et al.*, “Hector open source modules for autonomous mapping and navigation with rescue robots,” *RoboCup 2013: Robot World Cup: Robot World Cup XVII.*, vol. 8371, pp.624–631, 2014.

[15] W. Hess *et al.*, “Real-time loop closure in 2D LIDAR SLAM,” *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1271–1278, 2016. DOI: 10.1109/ICRA.2016.7487258.

[16] J. M. Santos, D. Portugal and R. P. Rocha, “An evaluation of 2D SLAM techniques available in Robot Operating System,” *2013 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pp. 1–6, 2013. DOI: 10.1109/SSRR.2013.6719348.

[17] D. M. Turnage, “Simulation results for localization and mapping algorithms,” *2016 Winter Simulation Conference (WSC)*, pp. 3040–3051, 2016. DOI: 10.1109/WSC.2016.7822338.

[18] Q. Zou, Q. Sun, L. Chen, B. Nie and Q. Li, “A comparative analysis of LiDAR SLAM-Based indoor navigation for autonomous vehicles,” *IEEE Transactions on Intelligent Transportation Systems*, 2021, pp. 1-15, DOI: 10.1109/TITS.2021.3063477.

[19] K. Konolige *et al.*, “Efficient sparse pose adjustment for 2D mapping,” *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 22-29, DOI: 10.1109/IROS.2010.5649043.

[20] B. Steux and O. E. Hamzaoui, “tinySLAM: A SLAM algorithm in less than 200 lines C-language program,” *2010 11th International Conference on Control Automation Robotics Vision*, 2010, pp. 1975-1979, DOI: 10.1109/ICARCV.2010.5707402.

[21] H. Durrant-Whyte; N. Roy; P. Abbeel, “A linear approximation for graph-based simultaneous localization and mapping,” *Robotics: Science and Systems VII*, MIT Press, 2012, pp.41-48.

[22] X. Lei *et al.*, “A novel FastSLAM framework based on 2D LiDAR for autonomous mobile robot,” *Electronics* , vol. 9, no. 4, p. 695, Apr. 2020, DOI: 10.3390/electronics9040695.

[23] H. Zhang, L. Yu and S. Fei, “Design of Dual-LiDAR High Precision Natural Navigation System,” *IEEE Sensors Journal*, 2022, DOI: 10.1109/JSEN.2022.3153900.

[24] W. Ali, P. Liu, R. Ying and Z. Gong, “A Life-Long SLAM Approach Using Adaptable Local Maps Based on Rasterized LIDAR Images,” *IEEE Sensors Journal*, vol. 21, no. 19, pp. 21740-21749, Oct 2021, DOI: 10.1109/JSEN.2021.3100882.

[25] R. Kuramachi *et al.*, “G-ICP SLAM: An odometry-free 3D mapping system with robust 6DoF pose estimation,” *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 176–181, 2015. DOI: 10.1109/ROBIO.2015.7418763.

[26] M. Bosse, R. Zlot and P. Flick, “Zebedee: Design of a spring-mounted 3-D range sensor with application to mobile mapping,” *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1104–1119, Oct. 2012.

[27] H. Wang, C. Wang and L. Xie, “Lightweight 3-D localization and mapping for solid-state LiDAR,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1801-1807, April 2021, DOI: 10.1109/LRA.2021.3060392.

[28] L. Zhou, D. Koppel and M. Kaess, “LiDAR SLAM with plane adjustment for indoor environment,” *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7073-7080, Oct. 2021, DOI: 10.1109/LRA.2021.3092274.

[29] V. Kachurka *et al.*, “WeCo-SLAM: Wearable cooperative SLAM system for real-time indoor localization under challenging conditions,” *IEEE Sensors Journal*, vol. 22, no. 6, pp. 5122-5132, March 2022, DOI: 10.1109/JSEN.2021.3101121.

[30] H. Zhou, Z. Yao, Z. Zhang, P. Liu and M. Lu, “An Online Multi-Robot SLAM System Based on Lidar/UWB Fusion,” *IEEE Sensors Journal*, vol. 22, no. 3, pp. 2530-2542, Feb 2022, DOI: 10.1109/JSEN.2021.3136929.

[31] Z.X. Li, G.H. Cui, C.L. Li and Z.S. Zhang, “Comparative Study of Slam Algorithms for Mobile Robots in Complex Environment,” *2021 6th International Conference on Control, Robotics and Cybernetics (CRC)*, 2021, pp. 74-79, DOI: 10.1109/CRC52766.2021.9620122.

[32] Y. K. Tee and Y. C. Han, “Lidar-Based 2D SLAM for mobile robot in an indoor environment: A review,” *2021 International Conference on Green Energy, Computing and Sustainable Technology (GECOST)*, 2021, pp. 1-7, DOI: 10.1109/GECOST52368.2021.9538731.

[33] RPLIDAR-A2 Laser range scanner [Online]. Available: https://www.slamtec.com/en/Lidar/A2, Accessed on: Feb. 5, 2022.

[34] Jetson Xavier NX Developer Kit[Online]. Available: https://developer.nvidia.com/, Accessed on: Feb. 5, 2022.

[35] A. Censi, “An ICP variant using a point-to-line metric ,” *2008 IEEE International Conference on Robotics and Automation*, 2008, pp. 19-25, DOI: 10.1109/ROBOT.2008.4543181.

Dr Quang Huy Nguyen received the BSc (Hons.) and MSc (Hons.) degree from University of Surrey in 2013 and 2014 respectively and the PhD degree from Liverpool John Moore University in 2020. He is currently a researcher for Liverpool John Moores University. From 2019, he has been working with CAL International and Innovate UK in a collaborated project . His research interests include robotic, indoor positioning, SLAM etc. Dr Nguyen is the lead project engineer for this KTP project.

Dr Princy Johnson is the academic lead on this KTP project. Dr Johnson is a Reader in Sensors and Data in the School of Engineering at Liverpool John Moores University. She received her PhD from King’s College London. During her tenure at Nortel Networks UK Ltd. she had a patent issued for a novel re-configurable OADM. Her research interests include energy-efficient protocols for wireless sensor and mobile networks, community clustering in big data, and intelligent algorithms. Her pedagogy in leaderships skills is the result of her passion for developing great leaders. An avid collaborator and a registered PRINCE2 ® practitioner, she has built an extensive collaborative network across Europe, Canada and USA.

David Latham is the Head of Development for NeedleSmart as well as the Engineering Director for CAL International. David is the industry supervisor for this KTP project. David is responsible for managing processes from concept through to product development. He has extensive knowledge of project management in multidisciplined roles across a variety of engineering sectors.