

Design of mapping system for domestic service robot using light detection and ranging

Muhammad Attamimi¹, Felix Gunawan², Djoko Purwanto¹, Rudy Dikairono¹, Astria Nur Irfansyah¹

¹Department of Electrical Engineering, Faculty of Intelligent Electrical and Informatics Technology, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia

²Department of Electrical Engineering, College of Electrical Engineering and Computer Science, National Taiwan University of Science and Technology, Taipei, Taiwan

Article Info

Article history:

Received Dec 9, 2023

Revised Jul 26, 2024

Accepted Aug 7, 2024

Keywords:

Domestic service robot

Light detection and ranging

Mapping

Microcontroller

Mini-PC

ABSTRACT

Service robots are becoming increasingly essential in offices or domestic environments, usually called domestic service robots (DSR). They must navigate and interact seamlessly with their surroundings, including humans and objects, which relies on effective mapping and localization. This study focuses on mapping, employing the light detection and ranging (LiDAR) sensor. The sensor, tested at proximity, gathers distance data to generate two-dimensional maps on a mini-PC. Additionally, it provides rotational positioning and robot odometry, broadening coverage through robot movement. A microcontroller with wireless smartphone connectivity facilitates control via Bluetooth. The robot is also equipped with ultrasonic sensors serving as a bumper. Testing in rooms of varying sizes using three methods (i.e., Hector simultaneous localization and mapping (SLAM), Google Cartographer, and real-time appearance-based mapping (RTAB-Map)) yielded good quality maps. The best F1-measure value was 96.88% achieved by Google Cartographer. All the results demonstrated the feasibility of this approach for DSR development across diverse applications.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Muhammad Attamimi

Department of Electrical Engineering, Faculty of Intelligent Electrical and Informatics Technology

Institut Teknologi Sepuluh Nopember

Surabaya, Indonesia

Email: attamimi@ee.its.ac.id

1. INTRODUCTION

These days, people have less time to do their domestic load because of their increased socioeconomic behaviors. Finding a reliable and skilled assistant to care for daily needs is not easy [1]. Because of that, assistant robots have a promising potential to be a big part of people's lives and help our everyday lives [2]. This also applies to service robots. Some assistant robots can be programmed to work in a conditioned environment; thus, the kinematic movements could be programmed in a fixed way to do their job. However, domestic tasks often require direct interaction between the robots and the users who are not experts in the robotics field [1]. Thus, service robots cannot move without observing their environment since the environment always changes because of interactions with humans and other factors. To work in such conditions, the robots must work based on the target environment and learn the defined behavior in the robots and the environment, especially the complex ones such as the domestic environment [3]–[11].

Considering the environment modeling, the problem usually falls into simultaneous localization and mapping (SLAM) problems. In SLAM, robots process their trajectory (localization) while creating a map of

the environment (mapping) in an unknown area [12]. However, creating a map while processing movement kinematics is not easy. Hence, this problem can be divided into two problems, i.e., mapping and localization.

This study focuses on the mapping problem implemented on domestic service robots (DSR). Mapping has multiple types. According to the output (i.e., map), there are two-dimensional and three dimensional maps. According to how the data is obtained, it is laser-based using light detection and ranging (LiDAR), and visual-based using a depth camera. For this research, the mapping is a two-dimensional map using LiDAR. As for kinematics, kinematics is controlled by a smartphone as the remote control for the microcontroller that controls three omni-wheels. The data obtained from these sensors in the robot is then processed to be a map ready to be used on service robots. This mapping robot is expected to be able to map an area with multiple rooms nicely and accurately. Hence, this mapping robot could work optimally for DSR.

Several studies are related to this work. First, the pieces of literature [6]–[12] discussed DSR. Furthermore, this research is related to robot navigation. Robot navigation is related to mapping, localization, and path planning. Costa *et al.* [13] used a perfect match algorithm to calculate the position and orientation of the robot calculated from the data collected from the laser range finder. The instantaneous center of rotation (ICR) method is discussed by Shi *et al.* [14], where ICR location mapping allows the robot to separate linear and angular motions under Cartesian space. The use of uniform cubic B-Spline curves was carried out in research [15] to avoid large-scale matrix calculations, effectively reducing space and time complexity. Discussions about mapping and localization exist in the literature [16], [17]. Basically, the navigation system depends on the size of the robot and the type of sensors used. In this research, the focus is on engineering methods that have been developed with reference to the use of several types of sensors, such as LiDAR and ultrasound.

2. METHOD

This section explains the overall design of the system that has been created, including hardware design, software design, and data collection and testing processes. The hardware design includes mechanical and electronic systems from the robot's base. In this study, we develop a robot base, as shown in Figure 1(a), which consists of a three-wheeled omnidirectional structure, several sensors, including ultrasonic sensors, and LiDAR. The software design is divided into android software program design, microcontroller program design, and a mapping system that modified and compared several methods, such as Hector SLAM [18], Google Cartographer [19], and real-time appearance-based mapping (RTAB-Map) [20].

2.1. Overview of the proposed system

Figure 1(b) shows the block diagram of the mapping system where input data from the environment is obtained using a distance sensor in the form of LiDAR. The distance data from each reading angle of the LiDAR is computed with the map builder from the robot operating system (ROS) [21] package in the form of Hector SLAM, Google Cartographer, and RTAB-Map with an output of a map. Figure 1(c) shows a block diagram of the overall hardware. The microcontroller in the robot's base has three inputs: ultrasonic distance data, encoder data, and serial data from Bluetooth. The Bluetooth serial data only works one way to receive direction data from android smartphones from the Bluetooth module. There are four ultrasonic data to measure the distance from the front, right, back, and left of the robot that functions as a bumper so the robot will stop before it crashes into something while controlled. There are three encoder data to measure the motors' movement as feedback. The outputs from the microcontroller are three signals to control each motor's speed using the motor driver. In the mapping system, a mini-PC requests data from the LiDAR, which is sent to the mini-PC. This data is processed into a map and displayed on an external laptop monitor. The external laptop communicates with the mini-PC using virtual network computing (VNC) protocol connected to a wireless network; thus, the external laptop could function as a monitor and keyboard.

2.2. Base robot program design

The base robot program is used to create the robot's movement control system using an android smartphone as the controller. Hence, programming is needed to create the microcontroller's android application and controller program. The android program design was done with the help of the MIT App Inventor 2 platform [22]. The state diagram of this android program is shown in Figure 2(a).

When the program starts, it will wait for input from the user to connect the Bluetooth connection between the android and Bluetooth module. When the connect button is pressed, the program will request a pairing, and the program waits for a response that the connection has succeeded and then starts a timer. The timer is set with a period of ten milliseconds. If the timer runs out, the program will send a message of the button pressed and then restart the timer. In the waiting state or movement input, if the disconnect button is pressed, the Bluetooth connection will disconnect, and the program will return to the waiting-for-input state.

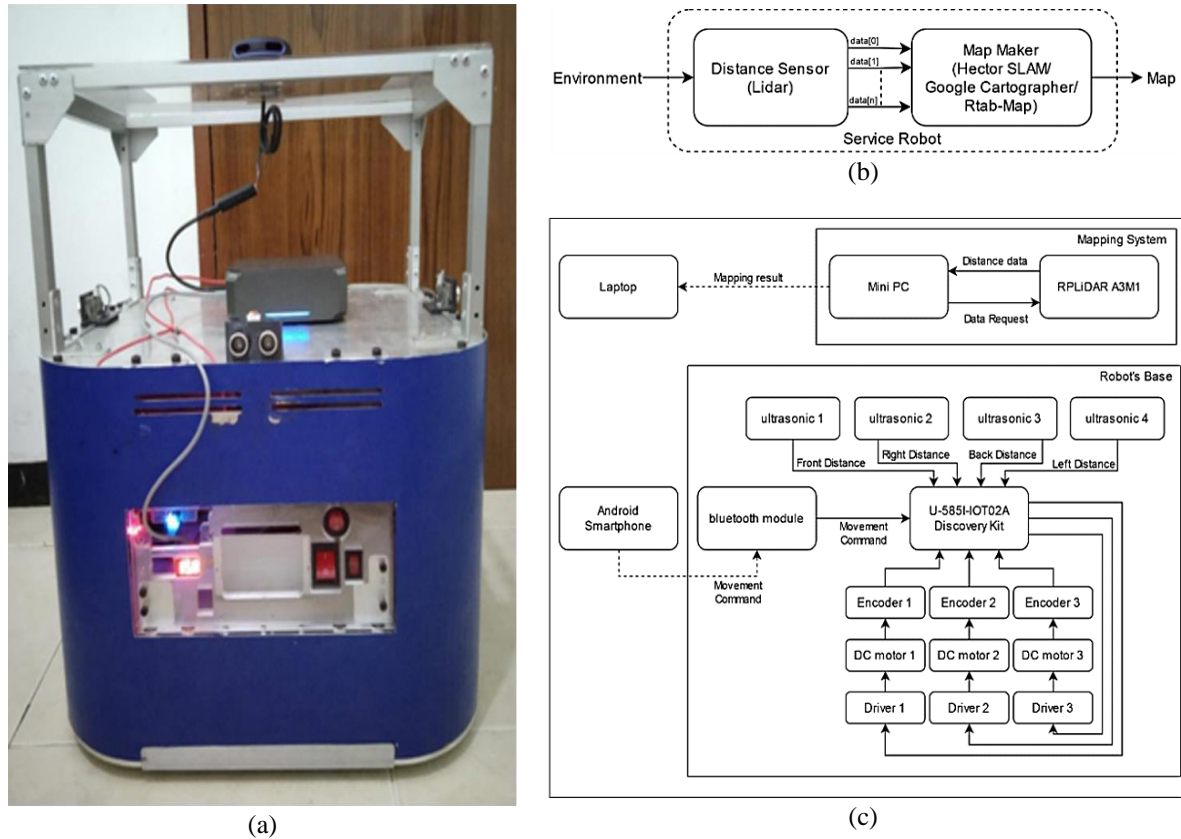


Figure 1. The robot used in this paper with system's block diagram: (a) robot base, (b) block diagram of a mapping system, and (c) block diagram of hardware

The microcontroller program design was done using the STM32CubeIDE 1.7.0 application [23]. The program was created based on the state diagram in Figure 2(b). This program works by first waiting for data from the Bluetooth module. If data is received, the data will be inserted into an array according to its order based on the parity value that has been defined. Ultrasonic sensors then measure the distance on all four sides of the robot to compute the input value into linear speed value using (1):

$$V_{linear} = V_{input} \times \frac{S_{ultrasound} - S_{stop}}{V_{inputmax} + S_{stop}}, S_{stop} \leq S_{ultrasound} \leq V_{inputmax} + 2 \times S_{stop} \quad (1)$$

After the linear speed is obtained, linear and rotational speed values are recalculated into the set point speed of each motor. Since we know that every motor has an angle of 120 degrees, with the first motor defined as 0 degrees, then the set point value of each motor s_1, s_2, s_3 is obtained using (2)-(4):

$$s_1 = v_x \cdot \cos(0) + v_y \cdot \sin(0) + v_w \quad (2)$$

$$s_2 = v_x \cdot \cos(240) + v_y \cdot \sin(240) + v_w \quad (3)$$

$$s_3 = v_x \cdot \cos(120) + v_y \cdot \sin(120) + v_w \quad (4)$$

These set points are then used on the proportional integral derivative (PID) control function, where the feedback from the rotary encoder connected to the motor is calculated as the error value.

2.3. Mapping program design

The mapping program was run using a mini-PC with Ubuntu 18.04 operating system with ROS Melodic. The design of this mapping program started with preparing the *rplidar_ros* package [24], and then we can design the Hector SLAM, Google Cartographer, and RTAB-Map program. Hector SLAM is a method that can make a 2D grid map quickly using 2D LiDAR with a low computation resource. This method also

can use external sensors such as an inertial measurement unit (IMU) to estimate its position in 3D. The downside of this method is no loop closure detection. Thus, the map cannot correct itself when visiting the previous location. Besides that, Hector SLAM does not require external odometry, which can have pros and cons depending on the environment. In this study, the Hector SLAM program is designed according to Kohlbrecher *et al.* [25].

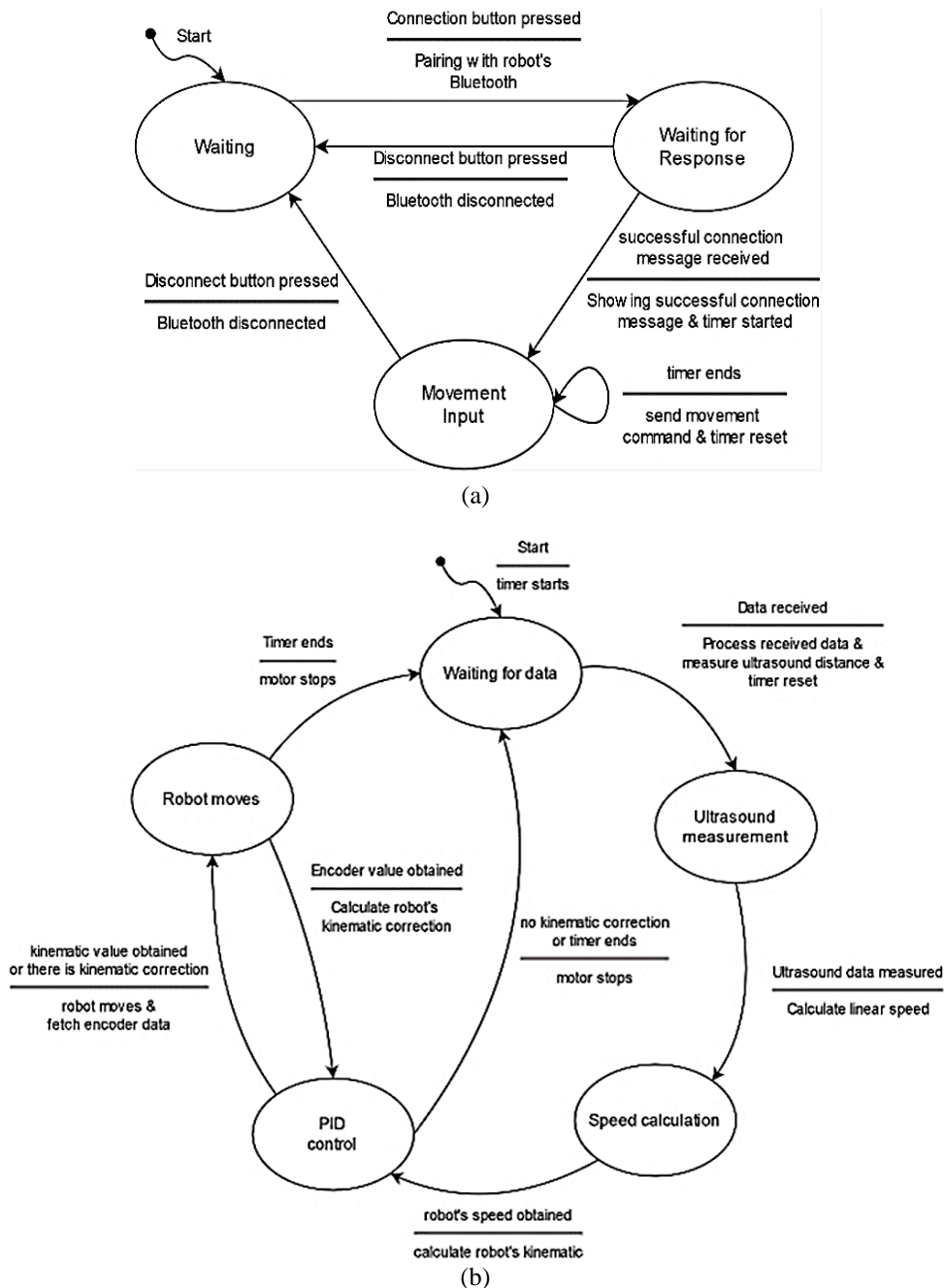


Figure 2. State diagrams used in this study: (a) state diagram of controller's system and (b) microcontroller's state diagram

Google Cartographer is a LiDAR graphic-based method that produces a 2D grid map from the graphical representation. While mapping, this method creates a sub-map connected by the borders in the graph. When a loop closure is detected, the sub-map position will be re-optimized to correct the error from

the sensor's noise and scanning accuracy. In this paper, the Google Cartographer program is designed according to Hess *et al.* [19].

RTAB-Map is a graphic-based SLAM method integrated with ROS as *rtabmap_ros* package [26]. This method can use a laser-based sensor, RGB-D camera, and stereo camera. This method also has an odometry configuration in the ROS node as an external input, which means the SLAM can use any odometry. The map structure is a graph with nodes and links. After sensor synchronization, the short-term memory module creates a node that memorizes the odometry pose, the sensor's raw data, and additional helpful information for the next module. Nodes are produced constantly in milliseconds based on the data amount created from the nodes that must overlap. For example, when the robot moves fast, and the sensor's range is small, the detection rate must be increased to ensure the data from the consecutive nodes are overlapping. However, if it is adjusted too high, it would increase unnecessary memory usage and computation time.

2.4. Data collection

Two data types are used in the data collection: real-time and recorded. For real-time data, we just need to use the command of each mapping method, and the robot will make the maps in real time. For recorded data, the *rplidar_ros* node has to be run first. When the node is already run, indicated by the rotating LiDAR, run the *roscat* command in a new terminal to record/*scan* node published by *rplidar_ros*. After recording, stop the command by pressing *ctrl+c* on each terminal. After the map is produced, use the *map_saver* command to save the map. The map will be saved in *pgm* and *yaml* format that can be published again as a map ready to be combined with an independent localization system.

According to Guimarães *et al.* [27], the parameters used in this map saver method are *-occ 65* and *-free 20*. Occ is an abbreviation of occupied threshold, meaning that the occupancy probability above the set value that is 0.65 will be considered as fully occupied. On the other hand, free means the occupancy probability lower than the set value that is 0.20 will be considered empty. In this study, we take four sets of recorded data for evaluation from Laboratory B-202 in the Department of Electrical Engineering, Institut Teknologi Sepuluh Nopember (ITS), Surabaya, Indonesia. The first data (hereafter we call scene 1) was taken in the main room of the laboratory in a clean condition (see Figure 3(a)), the second data (hereafter we call scene 2) was taken in the main room with some obstacles (see Figure 3(b)), the third data (hereafter we call scene 3) was taken in whole room of the laboratory in a clean condition, and the fourth data (hereafter we call scene 4) taken in whole room with some obstacles.



Figure 3. Experimental conditions: (a) clean condition and (b) with some obstacles

2.5. Map evaluation

The maps were evaluated by comparing each mapping result from each method in every dataset. The accuracy is measured based on a map area that missed the ground truth (GT). GT is created using the AutoCAD application, with the values obtained manually by measuring the room using a gauge. The map's distance is measured using the measure feature in RVIZ and adjusted to match the size in AutoCAD. The area was then obtained in AutoCAD using the area measurement feature. The obtained area values were then separated into true positive (TP), false positive (FP), and false negative (FN). TP is the intersection between mapping results and GT. FP is the area in the mapping result outside the GT area. FN is the area inside the

GT but outside the mapping result. These values are used to calculate recall (R) and precision (P). The blueprints of Laboratory B-202 used as the GT are shown in Figures 4(a) and (b). The blueprint's total area shown in Figures 4(a) and (b) are 467575 cm² and 873773 cm².

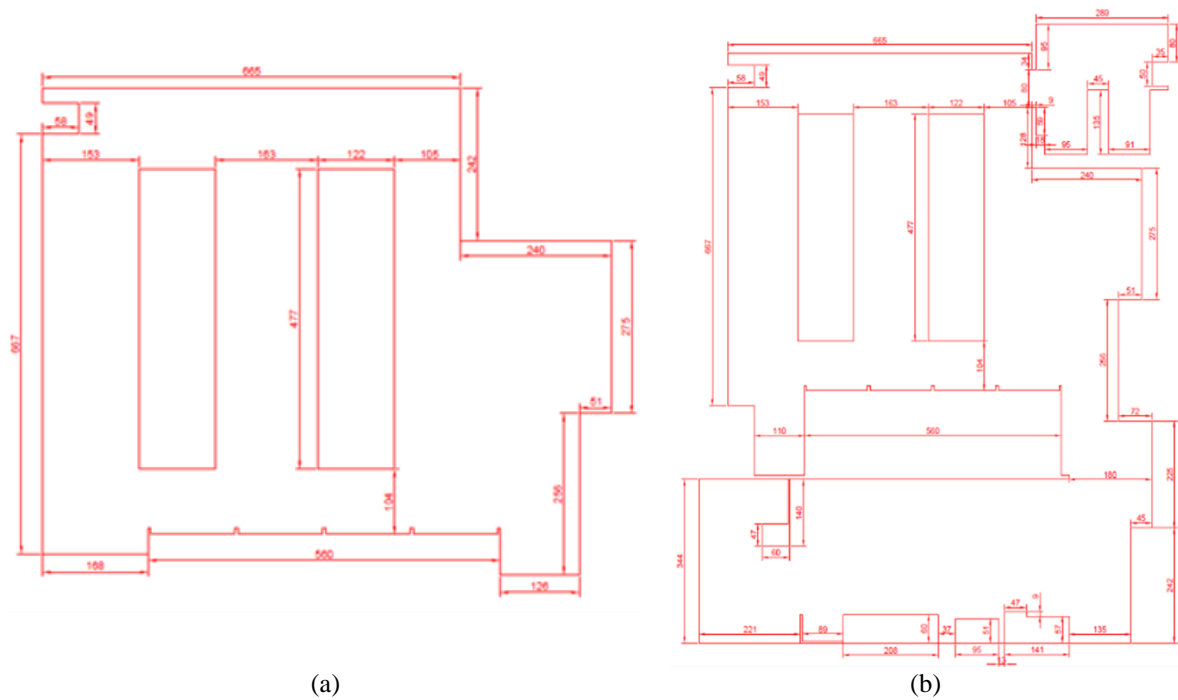


Figure 4. Lab. B-202 blueprint: (a) main room and (b) whole room

3. RESULTS AND DISCUSSION

From the mapping that has been done, the results and analysis are as follows. First, the maps generated from each method are depicted in Figure 5. Next, each scene's corresponding evaluation metrics are summarized in Tables 1 and 2. Assuming that the Hector SLAM results were notified with N/A with zero, we calculated the average value of recall, precision, and F1-measure for each method; and summarized them in Figure 6.

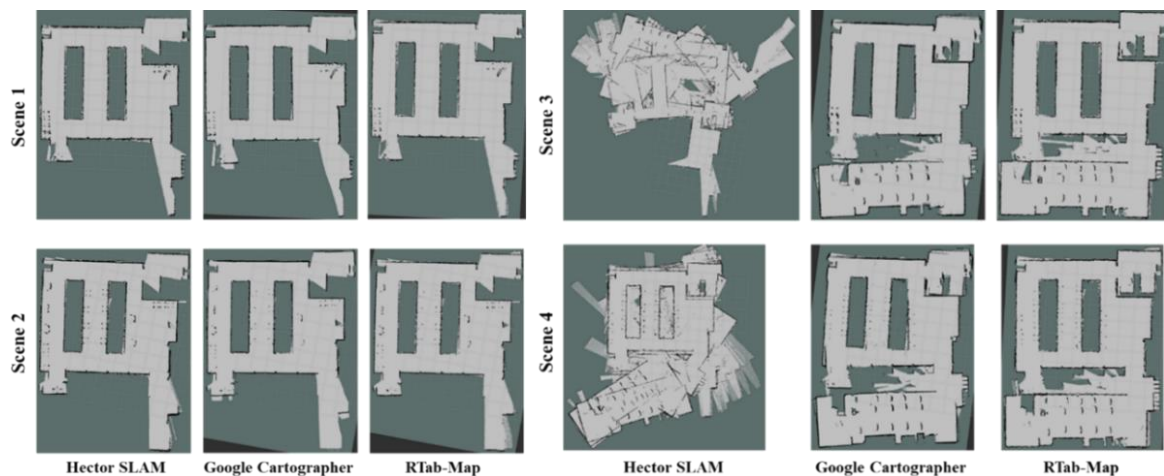


Figure 5. Maps of scene 1, 2, 3, and 4 generated by Hector SLAM, Google Cartographer, and RTAB-Map

Table 1. Evaluation results of maps of scene 1 and scene 2

Methods	Results of scene-1			Results of scene-2		
	Hector SLAM	Google Carto	RTAB-Map	Hector SLAM	Google Carto	RTAB-Map
GT	467575	467575	467575	467575	467575	467575
TP	440098.05	466497.74	465215.02	465942.3	467468.4	467502.1
FN	27476.95	1077.26	2359.98	1632.74	106.65	72.93
FP	6210.04	15543.74	117756.61	18545.33	14514.95	15176.68
R	94.12%	99.77%	99.50%	99.65%	99.98%	99.98%
P	98.61%	96.78%	79.80%	96.17%	96.99%	96.86%

Table 2. Evaluation results of maps of scene 3 and scene 4

Methods	Results of scene-1			Results of scene-2		
	Hector SLAM	Google Carto	RTAB-Map	Hector SLAM	Google Carto	RTAB-Map
GT	873773	873773	873773	873773	873773	873773
TP	N/A	838809.35	854709.55	N/A	822556.9	859461.8
FN	N/A	34963.65	19063.45	N/A	51216.09	14311.18
FP	N/A	36092.13	10400.68	N/A	37449.46	12510.54
R	N/A	96.00%	97.82%	N/A	94.14%	98.36%
P	N/A	95.87%	98.80%	N/A	95.65%	98.57%

From the obtained values, the RTAB-Map method has the best recall score. However, the Hector SLAM method has the best precision score. Besides recall and precision, we can observe that all three methods could produce visually good maps from data 1 and 2, as shown in Figure 5 (scene 1 and scene 2). On the other hand, maps from data 3 and 4 using the Hector SLAM method were deformed to the point they could not be measured (illustrated in scene 3 and scene 4 in Figure 5). However, the map from Hector SLAM in data 2 shows all small obstacles clearly without missing anything. The reason was suspected because this method was more sensitive to sudden and fast movement than the other methods. On the Google Cartographer method, maps from data 3 and 4 had a slight deformation, but the overall data was still usable. In this method, some small obstacles on data 2 and 4 were missing or too small to observe. On the RTAB-Map method, maps from all four data resulted in decent maps with no significant deformation. This method also did not detect some small obstacles or showed them in a very small size in data 2 and data 4. From maps produced by all three methods, they could filter moving objects around the robot.

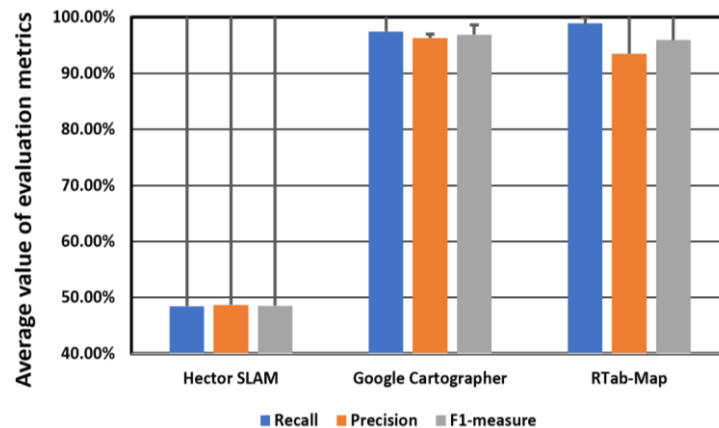


Figure 6. Average values of evaluation metrics (i.e., recall, precision, and F1-measure) generated by Hector SLAM, Google Cartographer, and RTAB-Map

4. CONCLUSION

From mapping result analysis using data 1 to data 4, we can conclude that this study successfully produced maps using LiDAR. Based on the calculation, the map quality produced by the Hector SLAM method, if we do not include data 3 and data 4 in the calculation, has a recall of 96.88% with a precision of 97.39%, the Google Cartographer method has a recall value of 97.47% with a precision of 96.32%, and RTAB-Map has a recall value of 98.14% with a precision of 93.51%. Mapping using the RTAB-Map method results in maps with the best recall value and stable without notable distortion with data 1 to 4. While the Hector SLAM method could produce maps with the highest precision and accurately show small obstacles in




data 2, this method is unstable and easily distorted in data 3 and 4. Google Cartographer method produces maps with quality and stability between RTAB-Map and Hector SLAM with slight distortion. The maps from all three methods could also differentiate moving objects and still objects that wanted to be mapped. In the future, we are planning to add a visual sensor to our DSR so that it enables mapping using visual information. Moreover, an integration of LiDAR and visual sensors will also be considered to make it more robust.

REFERENCES




- [1] S. M. Bhagya *et al.*, "Proxemics and approach evaluation by service robot based on user behavior in domestic environment," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 8192–8199, doi: 10.1109/IROS.2018.8593713.
- [2] T. Bandara, H. M. R., B. M. S. S. Basnayake, P. Jayasekara, A. G. B., and D. P. Chandima, "Enhancing conceptual spatial map by amalgamating spatial and virtual cognitive maps for domestic service robots," in *2018 2nd International Conference on Electrical Engineering (EECon)*, 2018, pp. 150–155, doi: 10.1109/EECon.2018.8541010.
- [3] C. Jayawardena *et al.*, "Deployment of a service robot to help older people," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, 2010, pp. 5990–5995, doi: 10.1109/IROS.2010.5649910.
- [4] M. Negrete, J. Savage, and L. A. Contreras-Toledo, "A motion-planning system for a domestic service robot," *Trudy SPIIRAN*, vol. 60, pp. 5–38, 2018, doi: 10.15622/sp.60.1.
- [5] M. A. V. J. Muthugala and A. G. B. P. Jayasekara, "MIRob: an intelligent service robot that learns from interactive discussions while handling uncertain information in user instructions," in *2016 Moratuwa Engineering Research Conference (MERCon)*, Moratuwa, 2016, pp. 397–402, doi: 10.1109/MERCon.2016.7480174.
- [6] M. Attamimi *et al.*, "Learning word meanings and grammar for verbalization of daily life activities using multilayered multimodal latent Dirichlet allocation and Bayesian hidden Markov models," *Advanced Robotics*, vol. 30, no. 11–12, pp. 806–824, 2016, doi: 10.1080/01691864.2016.1172507.
- [7] M. Nakano *et al.*, "Grounding new words on the physical world in multi-domain human-robot dialogues," in *2010 AAAI Fall Symposium*, 2010.
- [8] M. Attamimi, T. Nakamura, and T. Nagai, "Hierarchical multilevel object recognition using Markov model," in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, Tsukuba, Japan, 2012, pp. 2963–2966.
- [9] M. Attamimi, T. Araki, T. Nakamura, and T. Nagai, "Visual recognition system for cleaning tasks by humanoid robots," *International Journal of Advanced Robotic Systems*, vol. 10, no. 11, p. 384, 2013, doi: 10.5772/56629.
- [10] W. Y. Yagui and P. T. Aquino-Junior, "Mechanical development of modules for a modular service robot," in *III Brazilian Humanoid Robot Workshop (BRAHUR) and IV Brazilian Workshop on Service Robotics (BRASERO)*, Online, 2020.
- [11] H. P. C. Sirithunge, A. G. B. P. Jayasekara, and D. P. Chandima, "Human attention estimation by a domestic service robot using upper body skeletal information," in *2017 IEEE International Conference on Industrial and Information Systems (ICIIS)*, 2017, pp. 1–6, doi: 10.1109/ICIINFS.2017.8300421.
- [12] D. Wu, Y. Meng, K. Zhan, and F. Ma, "A LIDAR SLAM based on point-line features for underground mining vehicle," in *2018 Chinese Automation Congress (CAC)*, 2018, pp. 2879–2883, doi: 10.1109/CAC.2018.8623075.
- [13] P. J. Costa, N. Moreira, D. Campos, J. Gonçalves, J. Lima, and P. L. Costa, "Localization and navigation of an omnidirectional mobile robot: the Robot@Factory case study," *IEEE Revista Iberoamericana de Tecnologías del Aprendizaje*, vol. 11, no. 1, pp. 1–9, 2016, doi: 10.1109/RITA.2016.2518420.
- [14] Y. Shi, M. R. Elara, A. V. Le, V. Prabakaran, and K. L. Wood, "Path tracking control of self-reconfigurable robot hTetro with four differential drive units," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 3998–4005, 2020, doi: 10.1109/LRA.2020.2983683.
- [15] D. Cong, C. Liang, Q. Gong, X. Yang, and J. Liu, "Path planning and following of omnidirectional mobile robot based on B-spline," in *2018 Chinese Control and Decision Conference (CCDC)*, Shenyang, China, 2018, pp. 4931–4936, doi: 10.1109/CCDC.2018.8407985.
- [16] C. Cadena *et al.*, "Past, present, and future of simultaneous localization and mapping: toward the robust-perception age," in *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, Dec. 2016, doi: 10.1109/TRO.2016.2624754.
- [17] Z. Danping and P. Tan, "CoSLAM: collaborative visual SLAM in dynamic environments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 2, pp. 354–366, Feb. 2013, doi: 10.1109/TPAMI.2012.104.
- [18] S. Kohlbrecher, J. Meyer, O. von Stryk, and U. Klingauf, "A flexible and scalable SLAM system with full 3D motion estimation," in *Proc. IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*, 2011, pp. 155–160, doi: 10.1109/SSRR.2011.6106777.
- [19] 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)*, 2016, pp. 1271–1278, doi: 10.1109/ICRA.2016.7487258.
- [20] M. Labbé and F. Michaud, "RTAB-Map as an open-source Lidar and visual SLAM library for large-scale and long-term online operation," *Journal of Field Robotics*, vol. 36, no. 2, pp. 416–446, 2018, doi: 10.1002/rob.21831.
- [21] M. Quigley *et al.*, "ROS: an open-source robot operating system," in *ICRA Workshop on Open Source Software*, vol. 3, no. 3.2, p. 5, 2009.
- [22] S.-C. Kong and H. Abelson, "MIT app inventor: objectives, design, and development," in *Computational Thinking Education*, Eds., ch. 3, 2019, pp. 31–49, doi: 10.1007/978-981-13-6528-7_3.
- [23] STMicroelectronics, "STM32CubeMX," Datasheet, [Online]. Available: <https://www.mouser.tw/pdfDocs/stm32cubemx-2.pdf>. (Accessed: Jul. 25, 2024).
- [24] M. A. Markom, A. H. Adom, E. S. M. M. Tan, S. A. A. Shukor, N. A. Rahim, and A. Y. M. Shakaff, "A mapping mobile robot using RP Lidar scanner," in *2015 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS)*, 2015, pp. 85–92, doi: 10.1109/IRIS.2015.7451592.
- [25] S. Kohlbrecher, J. Meyer, T. Graber, K. Petersen, U. Klingauf, and O. von Stryk, "Hector open source modules for autonomous mapping and navigation with rescue robots," *Robot Soccer World Cup*, 2014, pp. 624–631, doi: 10.1007/978-3-662-44468-9_58.
- [26] M. Labbé and F. Michaud, "Multi-session visual SLAM for illumination-invariant re-localization in indoor environments," *Frontiers in Robotics and AI*, vol. 9, pp. 1–20, 2022, doi: 10.3389/frobt.2022.801886.
- [27] R. L. Guimarães, A. S. de Oliveira, J. A. Fabro, T. Becker, and V. A. Brenner, "ROS navigation: concepts and tutorial," in *Robot Operating System (ROS)*, 2016, pp. 121–160, doi: 10.1007/978-3-319-26054-9_6.

BIOGRAPHIES OF AUTHORS






Muhammad Attamimi    received his B.E., M.E., and D.E. degrees from the University of Electro-Communications in 2010, 2012, and 2015, respectively. He received scholarship from Ministry of Education, Culture, Sports, Science and Technology Japan (MEXT) for his BE and ME courses. From 2012 to 2015, he was also a Research Fellow (DC1) of Japan Society for the Promotion of Science (JSPS). He was with a postdoctoral researcher at the Department of Mechanical Engineering and Intelligent Systems, The University of Electro-communications (UEC) from April 2015 to December 2015. Since January 2016, he was with Tamagawa University Brain Science Institute as a commission researcher for six months. Currently, he is a lecturer at Department of Electrical Engineering, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia. His research interests are computer vision, visual recognition, machine learning, multimodal categorization, artificial intelligence, probability robotics, intelligent systems, and intelligent robotics. He can be contacted at email: attamimi@ee.its.ac.id.






Felix Gunawan    earned his bachelor of technology (bachelor of engineering) with cumlaude honors in electrical engineering from Institut Teknologi Sepuluh Nopember (ITS), Surabaya, Indonesia, in 2022. He is currently pursuing a master's degree in electrical engineering at the College of Electrical Engineering, National Taiwan University of Science and Technology (NTUST), Taipei, Taiwan. His research interests are focused on robotics, artificial intelligence, and the internet of things (IoT). As part of his academic engagement, he actively participates in projects with the Intelligent Robotics Laboratory at NTUST and has served as a speaker at the International Conference on Advanced Robotics and Intelligent Systems (ARIS 2023). He holds AWS Academy Graduate certifications in Cloud Foundations and Cloud Security. He can be contacted at email: felix.iniemail@gmail.com.






Djoko Purwanto    was born in Indonesia, in 1965. He received the B.E. degree in electrical engineering from Institut Teknologi Sepuluh Nopember (ITS), Surabaya, Indonesia, in 1989. The M.Eng. and Ph.D. degree in electrical engineering from the Keio University, Tokyo, Japan, in 1999, and 2002, respectively. In 1990, he joined the Department of Electrical Engineering, Institut Teknologi Sepuluh Nopember (ITS), Surabaya, Indonesia, as a lecturer. His research interests are industrial electronic system, robotics and automation, computer vision, and artificial intelligence applications. He can be contacted at email: djoko@its.ac.id.



Rudy Dikairono    is a distinguished professional in the field of electrical engineering. He earned his bachelor's degree in electrical engineering (EE) from Institut Teknologi Sepuluh Nopember (ITS) in 2004, followed by a master's degree from EE-ITS and Fachhochschule Darmstadt, Germany, in 2009, and a doctoral degree from EE-ITS in 2021. Throughout his career, he has been a dedicated member of the academic community, serving as a lecturer at EE-ITS since 2005. He also held the position of Head of the Information and Communication Technology and Robotics Laboratory at ITS from 2012 to 2016 and currently leads as the Head of the Center of Excellence in artificial intelligence for healthcare and society at ITS. He can be contacted at email: rudydikairono@ee.its.ac.id.



Astria Nur Irfansyah    received his Ph.D. and M.E. degree in electrical engineering from the University of New South Wales, Australia, in 2016 and 2008, respectively. He graduated from Universitas Gadjah Mada with a Bachelor of Engineering degree in 2005, where he then worked as an Embedded Systems Engineer at Gamatechno Indonesia IT company. He has been an assistant professor at Institut Teknologi Sepuluh Nopember, Surabaya since 2010. His Ph.D. dissertation was on sigma-delta modulation circuits with digital assistance, where several CMOS integrated circuits containing ADC and DAC were designed as part of his Ph.D. research. His research interest includes microelectronics, mixed-signal circuit design, and more recently neuromorphic circuits for low power smart applications. He can be contacted at email: irfansyah@ee.its.ac.id.