

Visual analytics of 3D LiDAR point clouds in robotics operating systems

Alia Mohd Azri¹, Shuzlina Abdul-Rahman², Raseeda Hamzah³,
Zalilah Abd Aziz⁴, Nordin Abu Bakar⁵

^{1,2,3,4,5}Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, Malaysia

^{2,4}Research Initiative Group of Intelligent Systems, Universiti Teknologi MARA, 40450 Shah Alam, Selangor, Malaysia

Article Info

Article history:

Received Sep 17, 2019

Revised Dec 28, 2019

Accepted Feb 3, 2020

Keywords:

Autonomous vehicle

LiDAR

Localization

Mobile robot

RTAB-map

Simulation

ABSTRACT

This paper presents visual analytics of 3D LiDAR point clouds in robotics operating system. In this study, experiment on simultaneous localization and mapping (SLAM) using point cloud data derived from the light detection and ranging (LiDAR) technology is conducted. We argue that one of the weaknesses of the SLAM algorithm is in the localization process of the landmarks. Existing algorithms such as grid mapping and monte carlo have limitations in dealing with 3D environment data that have led to less accurate estimation. Therefore, this research proposes the SLAM algorithm based on real-time appearance-based (RTAB) and makes use of the red green blue (RGB) camera for visualisation. The algorithm was tested by using the map data that was collected and simulated on the robot operating system (ROS) in Linux environment. We present the results and demonstrates that the map produced by RTAB is better compared to its counterparts. In addition, the probability for the estimated location is improved which allows for better vehicle maneuverability.

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



Corresponding Author:

Shuzlina Abdul-Rahman,
Research Initiative Group of Intelligent Systems,
Faculty of Computer and Mathematical Sciences,
Universiti Teknologi MARA, 40450 Shah Alam, Selangor, Malaysia.
Email: shuzlina@fskm.uitm.edu.my

1. INTRODUCTION

Simultaneous localization and mapping (SLAM) of an autonomous vehicle is a process of locating and mapping the location of a moving vehicle in a dynamic environment. Localization and mapping process work together similar to the “chicken and egg” analogy. Localization of an autonomous vehicle (AV) needs a map of the environment in order to locate itself. The same goes to map process to produce a map of the surroundings where it requires the current state and location of the vehicle to build the map of the environment. Technically, AV is categorized on a scale from zero to five, with zero meaning no autonomy and five signifying complete autonomy [1].

The AV technology is important, gives many benefits such as lower the crashing probability, and would be very useful for smart city communities. Conceptually, autonomous navigation of a driverless vehicle consists of several questions such as “where am I going?”, “where am I currently?” and “how do I get there?” [2]. Self-localization is an important feature in the navigation of an autonomous vehicle. Navigation is known as a process of determining a suitable and safe path between the start point and the destination point where the mobile robot will move [3]. Besides that, it also can be defined as the ability to move in any particular environment [3]. The question of, “where am I currently?” is very important as it determines

the precise current state of the autonomous vehicle which can be achieved via a sensor. Light detection and ranging (LiDAR) sensor such as Velodyne VLP-16 PUCK was designed to scan the surrounding environment to construct the map and locate its current location. Based on [4], it is a stable sensor and reliable to be used because it can replicate the contour information of the surrounding obstacles. Velodyne VLP-16 PUCK emits its laser to scan the surroundings and collect data called point cloud which is in a 3-dimension state. LiDAR sensor can be integrated with the inertial measurement unit (IMU) for a precise detection and localization. It provides attitude or orientation, and velocity to an AV [5].

SLAM is the key feature in making a vehicle autonomous and navigates in an unknown environment. In SLAM, there are many algorithms that have been used such as the Kalman filter and Particle filter. Kalman filter algorithm is applied based on the calculation of the mean and variance to estimate a vehicle's pose. As for the Particle filter, it converts the SLAM problem into path and landmark estimation [6]. Particle filter algorithm consists of two popular methods namely the Monte Carlo and Rao-Blackwellized. Monte Carlo uses the Bayesian filtering approximation by drawing samples of particles as explained in [7]. It is one of the algorithms that is widely used for robot localization. According to [8], Monte Carlo algorithm is specialized for target tracking, statistics and computer vision literature as well as dynamic probabilistic networks and it works well based on fast sampling technology. As the robot moves and sense, Monte Carlo uses resampling method to estimate the distribution. When comparing Monte Carlo to other filter algorithm such as Kalman filter based, the advantage of Monte Carlo algorithm is that Monte Carlo able to express multi-modal distributions which resulting the robot can be localized globally. However, Monte Carlo only can deal with 2D data. Rao-Blackwellized method uses the extended Kalman filter and the unscented Kalman filter algorithm to improve further the estimation accuracy.

According to [9], it is a real complex problem for an autonomous vehicle to move from one place to another in urban areas like cities because of too many dynamic objects like pedestrians. In terms of the system algorithm for an autonomous car to work, the SLAM of a mobile robot has problems to locate itself. It also has problems to move based on the estimated prior location and to build map simultaneously as they move. SLAM algorithm that has been presently researched such as the Kalman filter algorithm can only be applied to the non-linear Gaussian distribution model, which can cause high computational time, linearization error and which can lead to non-accurate estimation. In addition, according to [10], extended Kalman filter algorithm can lead to poor representations of nonlinear function. This can cause the filter to be divergent. Some related work has shown that Kalman filter is not good in terms of accuracy for state vector estimation and also to measure noises in environments. As supported in [11], it is obviously known that Particle filter is better than Kalman filter when it comes to produce more precise and robust estimation because Kalman filter requires more extensive calculations. As for the graph-based SLAM, the main idea is to organize information in a graph and give solutions to SLAM problems as a global graph optimization. In addition, there is also the implementation of graph-based SLAM using visual methods such as using red green blue-depth (RGB-D) type camera. This visual SLAM algorithm is called the RTAB-Map algorithm. It uses graph optimization technique together with 3D map data that is achieved using the RGB-D camera to further improve the localization estimation accuracy. For localization using RTAB-Map algorithm, it uses Bayesian filtering that records closure-loop assumptions by estimating probability of forming closure-loop. The process of localization by using Bayesian filtering is really important, as Bayesian has the strongest foundation and the oldest approach [12]. Information fusing using Bayesian fusion is found to be more straightforward to be used for indoor and outdoor environments.

This paper presents a simulation of visual analytics of 3D LiDAR point clouds in robotics operating system. The aim of this paper is to experiment a few standard SLAM algorithms for mapping and localization using the robotic operating system (ROS) during the localization and mapping processes. The weakness of the SLAM algorithm is in the localization process of the landmarks. There are limitations in existing algorithms such as grid mapping and Monte Carlo when the data is in 3D environment which will lead to less accurate estimations. Thus, this paper proposes the SLAM algorithm based on real-time appearance-based (RTAB) and makes use of red green blue (RGB) camera for visualization. The remainder of this paper is organized as follows: Section 2 describes the past studies and presents the commonly used algorithm in SLAM. Section 3 presents the methodology while section 4 discusses the results and findings of the study. Finally, section 5 concludes the paper.

2. BACKGROUND STUDY

Traditionally, the advanced driver-assistance systems (ADAS) have been implemented to navigate an autonomous vehicle. However, over several past years, SLAM has shown its improvement over ADAS, particularly in detecting and tracking the moving objects, as described in [13]. For SLAM to navigate autonomously, an AV needs to know the precise location of its current state. This is known as localization or

in other words as a local state, which refers to the current location of an autonomous car as it navigates to designated places. According to Wolcott *et al.*, [14], localization has to be within a priori map in order to get a precise location. They further added that metadata is embedded into a priori map to transform the complication perception task into localization problems rather than using vehicle's sensors to detect lane, markings and traffic sign clearly. According to Bresson *et al.*, [15], the global navigation satellite system (GNSS) that was introduced as an alternative, cannot provide sufficient data for the autonomous vehicle to detect the location. This arises when the autonomous vehicle goes through an urban area with high buildings, where the infrastructure can block direct signals and generate multi-paths interference. Furthermore, in an urban area, there will be traffic congestion where these can also be an issue [16]. These two situations worsen the autonomous navigation in the urban and packed areas.

2.1. Real-time appearance-based algorithm

Real-time appearance-based (RTAB) SLAM is one of the algorithms that uses the basics of graph SLAM in carrying out the process of mapping and localization. According to [17], the advantage of graph SLAM is it lies within the structure of the matrix that stores the pose data of the vehicle together with the landmarks within the map. This information can visualize the entire trajectory of the vehicle. As explained in [18], the graph SLAM works by generating landmarks from raw radar signals and constructing a map once it achieves high accuracy in both localization and mapping. RTAB approach is one of the efficient methods to do SLAM because it creates a 2D occupancy grid and 3D octo-map that and has been applied in many past studies [19, 20].

In some researches, occupancy grid maps are built in Bayesian framework and it detected the mobile objects by comparing the accumulated occupancy grid map and local map built by current data [21]. The advantage of using graph SLAM is that the map is optimized and stored in an R-tree data structure in order to maintain its flexibility [18]. According to [19], RTAB-Map algorithm is designed to give solutions that are based on localization and graph composition. RTAB algorithm basically combines two algorithms which are loop closure detection and graph optimization. At the same time, it provides memory management to satisfy real-time requirements and allowing multi-session mapping to associate different maps. This model uses RGB images in order to generate locations related to those images. The images will then be compared to the previously visited locations for loop-closure detection for localization.

Based on one of the research on image processing for paddy in [22], pictures of paddy leaves taken from a UAV and with lens correction technique contained RGB images, successfully produced weighted accuracy of about 95.04%, which shows that detection using RGB image can generate accurate and detailed pictures as applied in this paper. This image will be processed by using image processing algorithm to enable its features to be extracted. According to Mutalib *et al* [22], image pre-processing that is frequently used is the conversion from RGB to Grayscale image. In addition, Kahar *et al.* [23], in their research on detection for paddy diseases, also included reading and extracting the RGB colour from images of paddy leaves to be converted into grayscale. At this stage, it is important to read the right colour as it determines the stages of the paddy diseases. The characteristics of this data are high dimensional, semistructured and also spatial patterns [24]. The use of RGB colour in RTAB algorithm provides a richer representation of the surrounding during the localization process.

As a loop-closure detection can be detected inside a map, detection can also be found between different maps due to the multi-session properties of the system. After a successful loop-closure detection, 3D transformation is applied to match the images. When the loop closure has been stored, the Bayesian filter will be updated by estimating the probability of performing loop closure. According to [25], Bayes rule is the best tool to perform the estimation because in solving SLAM problem, it is necessary to apply estimation theory to develop maps from noisy information. RTAB SLAM is a graph-based SLAM system that includes locations as nodes. The location will be applied during loop closure detection and then the map graph is optimized in order to decrease odometry errors. The steps in the RTAB algorithm is described as in [19].

2.2. 3D and 2D mappings

In the SLAM algorithm, the common problem is the localization of the landmark. For example, the existing algorithms such as the grid mapping and Monte Carlo localizations have limitations in dealing with 3D environment data which could lead to non-accurate estimations. These algorithms can only provide 2D maps for localization. The problem when using a 2D map is that the data is insufficient for the localization process as the vehicle moves around various places. Image data for a 2D map might drift away as the vehicle turns around and creates redundancy of data. Compared to a 2D map, a 3D map data is more preferable for localization. As in this research, the method used is the RTAB algorithm which includes both 2D and 3D maps. Obviously, with the help of both 2D and 3D maps, localizations made can be more precise compared to using only a 2D map.

As explained in [9], a 3D map is purposed to give a solution for the localization to be combined with a 2D map. A 3D map can help improve localizations as it contains many features such as point cloud data, noise, depth information of the image and also the colour intensity. A 2D map does not possess these features except that it requires smaller storage. Moreover, 3D maps have more flexible views and good dimensions compared to 2D maps. This information is important and contributes to better localizations. In addition, RTAB algorithm produces a 3D map by using RGB-Depth camera to collect data images from the surroundings. Generally, RTAB algorithm is suitable for SLAM since it generates 2D and 3D maps in multiple scenarios. With the help of 2D and 3D maps, it will give more accurate data for the localization process of the mobile robot since the localization will depend on the surrounding map to estimate its current and next state. As RTAB algorithm has richer information it is better than grid mapping and Hector SLAM, they can only generate 2D maps and process data from 2D maps in order to do localization.

Table 1 shows the comparison of features in RTAB-Map, grid mapping and Hector SLAM. As can be seen in Table 1, RTAB-Map has richer features compared to grid mapping and Hector SLAM. RTAB is one of the SLAMS visual and it uses RGB-Depth camera to collect data images from the surroundings. By having an RGB-Depth camera, RTAB can extract the depth element from the images. The depth data is so helpful to provide accurate information. Compared to grid mapping and Hector SLAM, only RTAB-Map algorithm uses the depth data. The depth data is collected by the Kinect sensor. For odometry; refers to the input that can be used to help the SLAM approach compute the motion estimation which is known as localization. Moreover, RTAB-Map can produce both 3D octo-map and 2D occupancy grid map which use point cloud data to map the surroundings. Unlike grid mapping, and Hector SLAM, they can only produce 2D occupancy grid map.

Table 1. Comparison between RTAB-Map, grid mapping, and Hector SLAM

Features	RTAB-Map	Grid Mapping	Hector SLAM
Type of Sensor	Kinect depth	Laser Scanner	2D-LiDAR
Degree of Freedom	6	3	6
Class	Stereo	Laser	Laser
Data	RGB-D	Scan	Scan
Depth	√		
2D-LiDAR	√	√	√
3D-LiDAR	√		
Odometry	√	√	
2D Map	√	√	√
3D Map	√		
Point Cloud	√		

3. METHODOLOGY

The data collection was carried out by using a mobile robot equipped with Velodyne VLP-16, high quality sensor designed for obstacle detection and navigation of autonomous vehicles. It operates by measuring distance with 16 layers of lasers, which spins around providing a complete 360° horizontal field of view. Figure 1 shows the system architecture for the RTAB-Map algorithm. This architecture comprises of 3 sections which are input, process and output. It starts with the input section where the data of the environment are scanned and collected using laser sensor VLP-16.

These data comprise of two types that are namely from Kinect sensor and laser sensor which will be extracted and synchronized. Next, the data is then matched and stored into database by using the concept of memory management. Memory management is considered as a database that consists of two types of memory, short-term memory (STM) and long-term memory (LTM). Short Term memory is like an entry point that works by adding new images when the data images received during the scanning process. While for long-term memory, it stores all of the data images of the surrounding that have been scanned. The experiments have been conducted using robotic operating systems (ROS) simulations. The output for this process is the map and the localization position for navigation.

4. RESULTS AND FINDINGS

This section explains the results and findings from creating maps and localizing the robot position. The results of mapping using RTAB-Map are discussed based on two versions, which are in 2D and 3D maps. The 2D map is based on the occupancy grid map while the 3D map is based on point cloud.

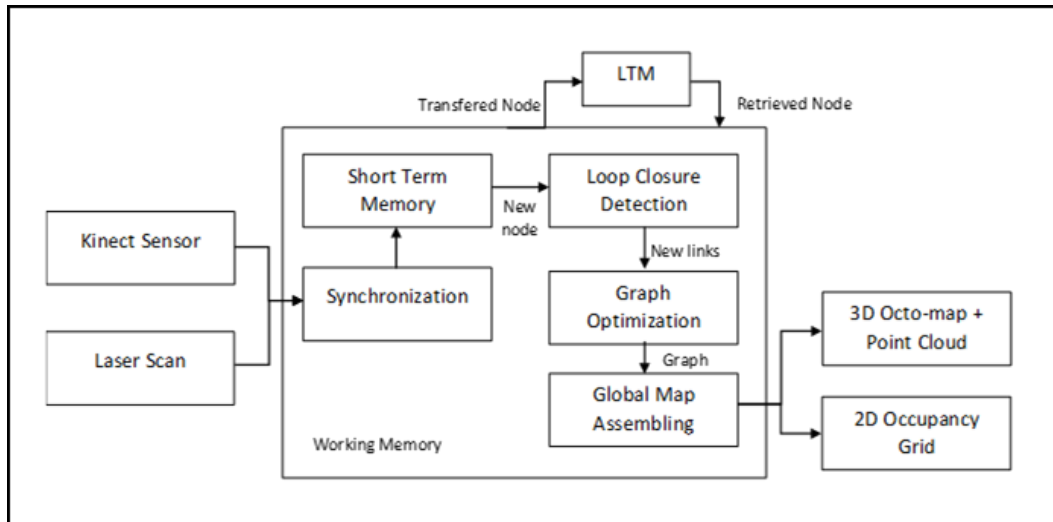


Figure 1. System architecture

4.1. Mapping results

The results of mapping using RTAB algorithm in simulation generated 3D map consist of depth data from kinect sensor and known as depth graphical view of the environment. Figure 2 and Figure 3 show the 3D map that has been created in simulation for both environments namely living room and tv room. As can be seen in the figure, mapping results for the 3D map seems so detailed and real akin to the real environment. These 3D maps consist of many point cloud data. The generated 3D map can be seen not just of the top view of the plan, but from any angle. In addition, the 3D map also has colours of the environment.

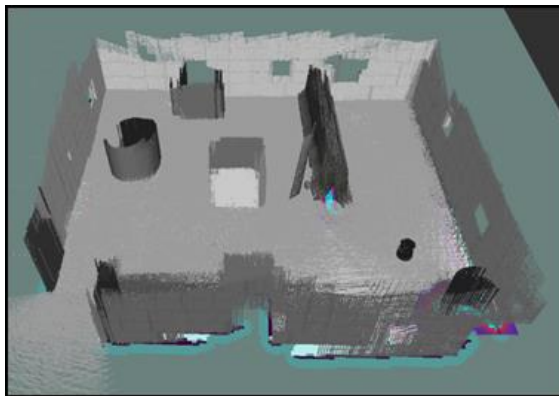


Figure 2. 3D map of living room

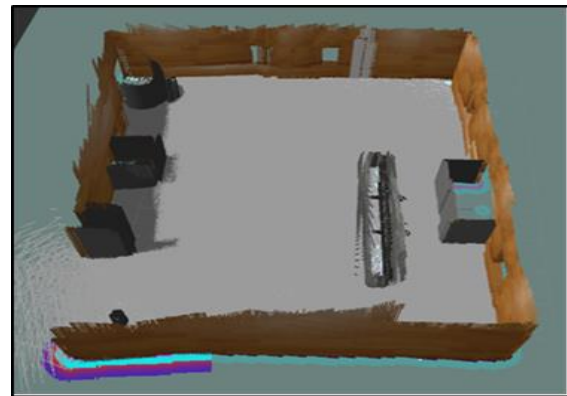


Figure 3. 3D map of tv room

Based on the created map, tv room mapping generated good results. For the first environment, living room, the map generated was poorer because of less features in the environment. The robot was not able to generate a good 3D map due to the lack of features which are needed for loop closure detection. When the robot scanned the environment in living room, less features made the robot failed to do loop closure to match the pictures of environment since it scanned an empty surrounding. However, in the second environment, a few more features were added such as cylinders, cones and tables in order to get better results.

4.2. RTAB SLAM performance

For the first room, living room in Figure 2, the turtlebot has been pointed to one goal. In this room, the turtlebot will pass through the obstacles. There were 5 experiments conducted as listed in the following table. The distance for each experiment was fixed at 3 meters each. Figure 4 illustrates the movement

of the turtlebot from point A to point B, with a 3 meters distance. Navigation goal is point B. As shown in Figure 4, the yellow dotted line represents the movements of the robot from point A to point B.

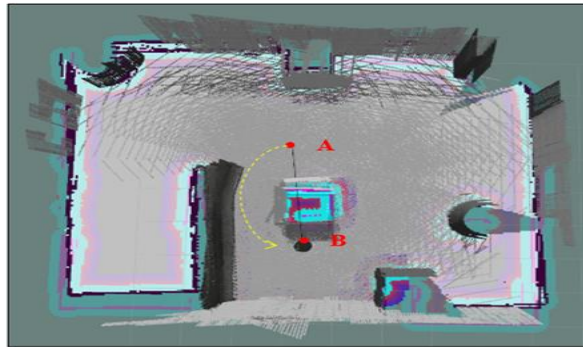


Figure 4. Experiment with RTAB algorithm

From the dotted line, it showed that the robot did not move in a straight line because of the obstacles between point A and point B. The robot was able to avoid the obstacles and move as indicated by the yellow dotted line. As for the time taken for the robot to reach its destination, the terminal output the time in Unix time stamp. This value was converted into ISO format to note the real time taken. The conversion of time from Unix time stamps to ISO format was done online. The time taken for experiment 1 was 15.636 seconds. Next, the second experiment was also carried out by passing through the obstacles similar to the first experiment but at different points. The distance between points C to point D was set at 3 meters. From the experiment, when the robot detected an obstacle, it tried to find another path and tried to locate itself while scanning the environment. The experiment results have been tabulated in the following Table 2 that shows the time taken from the tests conducted for both rooms, which have been gathered and represented as RTAB SLAM performance.

Table 2. RTAB SLAM performance

Experiment	Time taken to reach the goal		
	Points	Living room (seconds)	TV room (seconds)
1	A to B	15.636	10.107
2	C to D	36.890	7.458
3	E to F	17.085	7.548
4	G to H	63.240	8.447
5	I to J	14.179	7.160
Total		147.030s	40.720

Room with less total time taken is considered as a good environment for the RTAB. As shown in the table, total time taken for the turtlebot to reach its goal in living room was 147.030 seconds, while for tv room it was 40.720 seconds. There was a slightly different time recorded between the two above. In living room, the turtlebot passed through an obstacle from one point to another point, while in tv room, there were not many obstacles and the turtlebot did not pass through any obstacle, allowing it to move in a straight line.

From the results, the conclusion is tv room is the best environment compared to living room as the robot took the least time to navigate to the location at approximately 40 seconds. The RTAB algorithm performed its best to localize the robot when there were no obstacles around it as it moved to the goal. When there were no obstacles, the robot could locate itself precisely, and this helped the robot to process the information it collected from its surroundings in lesser time as to where to navigate within the map. Thus, it reached its goal faster without the need to avoid any obstacles. As we can see, the way the robot navigated itself in living room, it had to pass through the obstacles in front of it. As the robot reached the obstacle, it slowed down to 'decide' as to which way it needed to go, either to the left or right. In order to decide which way to turn, it needed to scan the environment for others possible path to go through. During the scanning process, the robot rotated a few times. Here, obviously we could see that when there were obstacles, it took time for the robot to detect and decide which path to use so as to avoid the obstacles in front of it.

4.3. Comparison between RTAB and AMCL localization

This subsection presents the comparison between RTAB and AMCL algorithms, in which five experiments were conducted, involving 5 different start-end points, experiment using AMCL algorithm shown in Figure 5. To be fair, a distance of 3 meters was fixed for each experiment. Table 3 shows the time taken for the turtlebot to reach the goal for both experiments using RTAB and AMCL algorithms. The experiments were conducted in tv room. As can be seen in this table, there are big differences in the time taken for the robot to reach the goal. For RTAB, the results showed it took approximately 41 seconds for the three experiments that were carried out. While for the AMCL algorithm, it took almost 204 seconds for the robot to reach its goal. These results prove that the RTAB algorithm is more successful in doing SLAM process as compared to the AMCL algorithm. It also shows that with the help of 3D maps, a more precise robot and autonomous vehicle mapping and localization can be produced. To support these findings, as explained in [26] that the use of Graph optimization method, which was applied partly in the RTAB algorithm, the localization improved with the map being constantly kept up-to-date. This is also due to feature selection and data size per map area remaining constant.

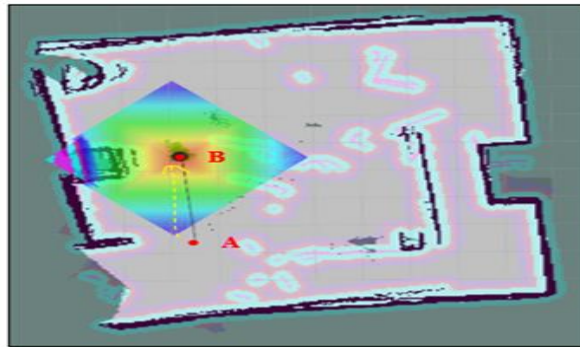


Figure 5. Experiment using AMCL algorithm

Table 3. Performance comparison between RTAB and AMCL in TV room

Experiment	Time taken to reach the goal		
	Points	Living room (seconds)	TV room (seconds)
1	A to B	10.107	88.214
2	C to D	7.458	9.218
3	E to F	7.548	60.275
4	G to H	8.447	30.169
5	I to J	7.160	15.956
Total		40.720	203.832

5. CONCLUSION

This paper demonstrated the simulation study of SLAM by using a low-cost service robot platform which was capable of mapping and localizing in an unknown environment. We have shown that the turtlebot was able to localize itself in the map created using Bayes filter in RTAB algorithm and was able to navigate to its goal better than the Monte Carlo and Gmapping algorithms. The entire process of navigation was visualized using Rviz and was found to be satisfactory since the robot was able to reach the goal point successfully. Further studies in this field with other SLAM algorithm and by using real mobile robot would be beneficial to many parties as the robotic field is currently highly demanded.

ACKNOWLEDGEMENTS

The authors would like to thank the Institute of Quality & Knowledge Advancement (InQKA) for the publication fee and the Faculty of Computer & Mathematical Sciences for all the given support.

REFERENCES

- [1] Bagloee, S. A., Tavana, M., Asadi, M., & Oliver, T., "Autonomous vehicles: challenges, opportunities, and future implications for transportation policies," *Journal of Modern Transportation*, vol. 24, no. 4, pp. 284-303, 2016.

- [2] S. Zhao, J. Gu, Y. Ou, W. Zhang, J. Pu and H. Peng, "IRobot self-localization using EKF," *2016 IEEE International Conference on Information and Automation (ICIA)*, Ningbo, 2016, pp. 801-806.
- [3] B. Rahmani, A. Eko Putra, A. Harjoko, and T. K. Priyambodo, "Review of vision-based robot navigation method," *International Journal of Robotics and Automation (IJRA)*, vol. 4, no. 4, pp. 254-261, Dec 2015.
- [4] Y. Xiao, Y. Ou and W. Feng, "Localization of indoor robot based on particle filter with EKF proposal distribution," *2017 IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, Ningbo, 2017, pp. 568-571.
- [5] R. Li, J. Liu, L. Zhang, and Y. Hang, "LIDAR/MEMS IMU integrated navigation (SLAM) method for a small UAV in indoor environments," *DGON Inertial Sensors and Systems (ISS)*, pp. 1-15, 2014.
- [6] F. Zhang, S. Li, S. Yuan, E. Sun, and L. Zhao, "Algorithms analysis of mobile robot SLAM based on Kalman and particle filter," *9th International Conference on Modelling, Identification and Control (ICMIC)*, pp. 1050-1055, 2017.
- [7] M. Murata, H. Nagano, and K. Kashino, "Monte Carlo filter particle filter," *European Control Conference (ECC)*, pp. 2836-2841, 2015.
- [8] W. Wei and K. Curran, "Indoor robot localisation with active RFID," *IAES International Journal of Robotics and Automation (IJRA)*, vol. 1, no. 3, pp. 137-144, 2012.
- [9] R. Altman, D. Mulligan, and Y. Shoham, "Artificial intelligence and life in 2030: One hundred year study on artificial intelligence," *Report of the 2015 Study Panel, Stanford University*, pp. 1-27, 2016. Available: <https://ai100.stanford.edu>
- [10] S. M. Malagon-Soldara, M. Toledano-Ayala, G. Soto-Zarazua, R. V. Carrillo-Serrano, and E. A. Rivas-Araiza, "Mobile robot localization: A review of probabilistic map-based techniques," *International Journal of Robotics and Automation (IJRA)*, vol. 4, no. 1, pp. 73-81, 2015.
- [11] N. Y. Ko and T. G. Kim, "Filtering method for location estimation of an underwater robot," *IAES International Journal of Robotics and Automation (IJRA)*, vol. 3, no. 3, pp. 168-183, 2014.
- [12] K. Nagla, M. Uddin, and D. Singh, "Multisensordata fusion and integration for mobile robots: A review," *IAES International Journal of Robotics and Automation (IJRA)*, vol. 3, no. 2, pp. 131-138, 2014.
- [13] A. S. A. Yeon, K. Kamarudin, R. Visvanathan, S. M. M. Syed Zakaria, L. M. Kamarudin, A. Zakaria, and A. Y. M. Shakaff, "Feasibility analysis Of 2D-slam using combination of kinect and laser scanner," *Jurnal Teknologi*, vol. 76, no. 12, pp. 9-15, 2015.
- [14] R. W. Wolcott and R. M. Eustice, "Visual localization within LIDAR maps for automated urban driving," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 176-183, 2014.
- [15] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser, "Simultaneous localization and mapping: A survey of current trends in autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 2, no. 3, pp. 194-220, 2017.
- [16] N. Isa, A. Mohamed, and M. Yusoff, "Implementation of dynamic traffic routing for traffic congestion: A review," *Communications in Computer and Information Science Soft Computing in Data Science*, pp. 174-186, 2015.
- [17] T. T. O. Takleh, N. A. Bakar, S. A. Rahman, R. Hamzah, and Z. A. Aziz, "A brief survey on SLAM methods in autonomous vehicle," *International Journal of Engineering & Technology*, vol. 7, no. 4, pp. 38-43, 2018.
- [18] F. Schuster, C. G. Keller, M. Rapp, M. Haueis, and C. Curio, "Landmark based radar SLAM using graph optimization," *IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2559-2564, 2016.
- [19] A. Abramov, C. Bayer, C. Heller, and C. Loy, "Multi-lane perception using feature fusion based on GraphSLAM," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3108-3115, 2016.
- [20] Z. Kong and Q. Lu, "A brief review of simultaneous localization and mapping" *IECON 2017-43rd Annual Conference of the IEEE Industrial Electronics Society*, pp. 5517-5522, 2017.
- [21] J. Zhou, D. Jianmin, and Y. Guangzu, "Occupancy grid mapping based on DSMT for dynamic environment perception," *IAES International Journal of Robotics and Automation (IJRA)*, vol. 2, no. 4, pp. 129-139, 2013.
- [22] S. Motalib, M. H. Abdullah, S. Abdul-Rahman, and Z. A. Aziz, "A brief study on paddy applications with image processing and proposed architecture," *IEEE Conference on Systems, Process and Control (ICSPC)*, pp. 124-129, 2016.
- [23] M. A. Abdul Kahar, S. Motalib, and S. Abdul Rahman, "Early detection and classification of paddy diseases with neural networks and fuzzy logic," *Recent Advances in Mathematical and Computational Methods*, pp. 248-257, 2015.
- [24] S. Abdul-Rahman, A. A. Bakar, and Z-A. Mohamed-Hussein, "An intelligent data pre-processing of complex datasets," *Intelligent Data Analysis*, vol. 16, no. 2, pp. 305-325, 2012.
- [25] N. M. Yatim and N. Buniyamin, "Particle filter in simultaneous localization and mapping (SLAM) using differential drive mobile robot," *Jurnal Teknologi*, vol. 77, no. 20, pp. 91-97, 2015.
- [26] F. Schuster, W. Zhang, C. G. Keller, M. Haueis, and C. Curio, "Joint graph optimization towards crowd based mapping," *IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1-6, 2017.