Analysis of an RGB-D Simultaneous Localization and Mapping Algorithm for Unmanned Aerial Vehicle

Muhammad Zamir Fathi Mohammad Effendi[1], Norhidayah Mohamad Yatim[2],
Zarina Mohd Noh[3], Nur Aqilah Othman[4]
G7 Aerospace Sdn. Bhd., Pusat Perindustrian Cipta Selatan, Bandar Serenia, 43900, Sepang, Selangor, Malaysia[1]
Fakulti Teknologi dan Kejuruteraan Elektronik dan Komputer (FTKEK), Universiti Teknikal Malaysia Melaka (UTeM),
Durian Tunggal, Melaka, Malaysia[2, 3]
Faculty of Electrical & Electronics Engineering Technology, Universiti Malaysia Pahang Sultan Abdullah,
Kuantan, Pahang, Malaysia[4]

***Abstract*—This study investigates the implementation of an RGB-D Simultaneous Localization and Mapping (SLAM) algorithm on an unmanned aerial vehicle (UAV) equipped with an Intel RealSense D435i camera. The study focuses on Real-Time Appearance-Based Mapping (RTAB-Map), a well-established RGB-D SLAM method capable of building 3D maps while simultaneously localizing a robot within its environment. Despite its advanced capabilities, deploying RTAB-Map on UAVs introduces specific challenges due to the dynamics of aerial navigation. This research evaluates the performance of RTAB-Map in terms of robustness, precision, and accuracy to optimize its application in UAV-based RGB-D SLAM. The findings reveal that the sequential frame-matching approach, combined with a minimum inliers threshold of 10, provides the most robust performance. In contrast, the global matching approach with a minimum inliers threshold of 20 offers better precision and accuracy. The results show that this implementation, utilizing an off-the-shelf hardware and software setup, has significant potential for advanced applications such as monitoring and surveillance in environments, where dense 3D mapping is critical.**

***Keywords*—Unmanned aerial vehicle; UAV; simultaneous localization and mapping; SLAM; RGB-D; real-time appearance based map; RTAB-map**

I. Introduction

Autonomous robotics has been a key area in the field of robotics. The necessity of determining a robot's position on a map in unknown environments, while simultaneously building the map representation itself, has led to the development of Simultaneous Localization and Mapping (SLAM) algorithms over the past three decades [1]-[4]. The demand for SLAM algorithms has grown significantly, as they are widely used for various applications in both ground-based and aerial robots. Computer vision technology has advanced over recent years, and the usage of vehicles with teleoperation capabilities is seen as a big potential to be commercialized. By integrating the SLAM algorithm with these vehicles, a technological milestone could be achieved. This study focused on the issue of SLAM with Unmanned Aerial Vehicle (UAV) which addresses a specific SLAM problem, involves the challenge of mapping the environment that has been explored by UAV in a 3D map representation, while determining its position within the gradually constructed 3D map [5].

Despite intensive research, implementing SLAM on UAV platforms remains challenging. One of the primary issues is that UAVs operate at higher speeds, which can reduce the accuracy of SLAM algorithms. Moreover, when applying SLAM in large or complex environments, UAVs may encounter difficulties such as rapid viewpoint changes, inconsistent loop closures, and accumulated drift errors [6]-[8]. A drawback often occurs when the UAV revisits previously mapped regions, where newly estimated landmark locations may not align with earlier ones. A robust and efficient algorithm is therefore required to achieve reliable SLAM performance for UAVs, especially given the trade-off between real-time operation and mapping accuracy. While robust methods exist for static and structured environments, mapping unstructured, dynamic, or large-scale settings continues to be an open research problem [2] [4].

Among existing RGB-D SLAM algorithms, RTAB-Map has proven effective as a graph-based approach in various robotic applications. However, its performance when applied to UAVs is inconsistent. UAV platforms introduce unique challenges such as high-speed motion, rapid viewpoint changes, and dynamic environments, which can significantly degrade SLAM accuracy and robustness. In particular, the performance of RTAB-Map depends heavily on parameter configurations (e.g., camera resolution, frame rate, odometry strategy, and loop closure thresholds) [9]-[12]. Yet, there is limited research systematically examining how these parameters influence UAV-based SLAM performance. Without such insights, it is difficult to identify optimal configurations that balance accuracy, robustness, and real-time feasibility for UAV operations.

Given this problem, this research study tries to address the following questions:

- How do state-of-the-art RGB-D SLAM algorithms such as RTAB-Map perform when applied to UAV platforms in terms of accuracy, robustness, and efficiency?
- Can modifications, parameter tuning, or integration strategies improve their performance for UAV-based applications?

To address the research questions outline the following objectives are outlined:

- To evaluate the impact of key parameters such as RGB and depth camera resolution, frame rate (fps), visual odometry strategies, and loop closure inlier thresholds on the performance of RTAB-Map for UAV-based SLAM.
- To investigate the optimum parameter configuration that maximizes accuracy, efficiency, and robustness of RTAB-Map in UAV navigation and mapping tasks.

This research is significant as it reduces the gap between conventional SLAM implementations and the unique requirements of UAV navigation. By evaluating and comparing RGB-D SLAM algorithms in UAV contexts, this study is expected to provide researchers and practitioners with some insights into algorithmic strengths and weaknesses, as well as practical guidance for deploying UAV-based SLAM in real-world environments. It is hoped that the outcomes can support the development of more reliable and efficient autonomous UAV systems for applications in surveillance, mapping, disaster response, and environmental monitoring.

The remainder of this study is structured as follows: Section II presents the literature review of the SLAM algorithm, focusing on two prominent RGB-D SLAM algorithms: ORB-SLAM2 and RTAB-Map. Section III outlines the research methodology applied in this study, while Section IV discusses the results and findings of the experiments conducted. Finally, Section V provides the conclusion of the study.

II. Literature Review

A. *Visual Simultaneous Localization and Mapping (VSLAM)*

In the Simultaneous Localization and Mapping (SLAM) algorithm, mapping is for obtaining a model of the robot environment, and localization is to estimate the position of the robot in the obtained map. Despite significant progress in this area, it still poses great challenges [2]-[4]. The accuracy of environment mapping depends highly on the effective parameters which need to be studied. The important parameters that affect SLAM performance include sensor uncertainty, correspondence issue, loop closing, time complexity and dynamic environment [1].

Visual sensor implementation of SLAM or also known as visual SLAM (VSLAM) systems, can be classified into monocular SLAM, binocular SLAM and RGB-D SLAM [9]. The input sensor for classic visual SLAM systems is a monocular or stereo camera. Using these two vision sensors to reconstruct a 3D map, sophisticated map initialization and map point triangulation operations are required, which are computationally challenging and can result in noisy observations. The introduction of RGB-D cameras, which deliver RGB images along with depth information, addresses the issue that comes with monocular and stereo cameras. Due to the fact that RGB-D cameras can provide both colored and depth images simultaneously, they are increasingly utilized for indoor scene reconstruction and can effectively address limitations such as reconstruction of low-textured areas [12]. Low-textured areas are surfaces with few distinguishing features. These areas are difficult to reconstruct because they lack sufficient feature points for effective tracking and mapping [10], [11].

B. *ORB-SLAM2 and RTAB-Map*

In this study, the RGB-D SLAM algorithms of interest that are to be researched are RTAB-Map and ORB-SLAM2. RTAB-Map was first developed in 2009 and released as an open source library in 2013. Since then, it has evolved into a complete graph-based SLAM approach that can be used in a variety of setups and applications [13]. RTAB-Map is a graph-based SLAM approach based on an incremental appearance-based loop closure detector. To determine whether a new image is likely to come from a prior or new place, the loop closure detector employs a bag-of-words approach. When a loop closure hypothesis is accepted, a new constraint is added to the map's graph, and the map's mistakes are minimised using a graph optimizer. To ensure that real-time limitations on large-scale environments are always maintained, a memory management strategy is employed to limit the number of locations used for loop closure detection and graph optimization. Visual odometry strategies play a crucial role in RTAB-Map’s operation. Frame-to-frame odometry compares each image to the previous one, offering speed and efficiency but suffering from drift over time. In contrast, frame-to-map odometry compares each image to a gradually built map, improving accuracy but increasing computational demands [11], [14].

ORB-SLAM2 is a comprehensive SLAM system with map reuse, loop closing, and re-localization capabilities. Proposed in 2016 by Raul Mur-Artal et al., ORB-SLAM2 provides support for stereo cameras and RGB-D cameras as an improvement from its predecessor, ORB-SLAM. The algorithm is able to operate swiftly in real-time [9]. ORB-SLAM2 selects ORB as the 2D features for all SLAM operations, including tracking, mapping, relocalization, and loop closing, in order to maintain speed and rotation invariance [10]. If tracking was successful in the previous frame, ORB-SLAM2 use a constant velocity motion model to project the camera pose and conduct a guided search of the map points encountered in the previous frame. If inadequate matches were discovered (i.e., due to an inadequate motion model), ORB-SLAM2 does a broader search of the map points surrounding the subject's position in the previous frame. The pose is then enhanced using bundle adjustment (BA) through the discovered 2D correspondences [10].

Although ORB-SLAM2 has significantly improved in both efficiency and precision, certain issues remain unresolved. One major challenge is system tracking failure in dynamic environments, which arises from the feature point extraction process being insufficiently robust in conditions such as sudden lighting changes, excessively strong or weak light intensity, or environments with minimal texture information [9]. Solving motion blur becomes a key focus in visual SLAM since common cameras frequently produce image blur when moving rapidly.

N. Ragot et al. (2019) conducted research on the benchmark of the two reviewed SLAM algorithms; ORB-SLAM2 and RTAB-Map [15]. The research utilized an Intel RealSense D435 camera mounted on a robotics-powered electric wheelchair to test and evaluate the performance of these algorithms across various configurations, such as straight-line motion, straight-line motion with a return path, and circular paths with loop closure. All experiments were performed in a controlled indoor environment.

The study highlighted that both RTAB-Map and ORB-SLAM2 have unique strengths in specific aspects. RTAB-Map, a graph-based SLAM system, integrates two key algorithms: loop closure detection and graph optimization. According to N. Ragot et al., RTAB-Map demonstrated more accurate trajectory estimation, better performance in translational movements, and more effective utilization of RGB data. However, it produced less accurate results during rotational movements. On the other hand, ORB-SLAM2 relies on consecutive image frames to construct a map and localize itself within it. It showed superior performance in distance measurement and odometry estimation compared to RTAB-Map. Nonetheless, its performance with RGB cameras was less effective than stereo cameras, particularly in outdoor scenarios [9].

While these benchmarking studies provide valuable insights, most evaluations of RTAB-Map have concentrated on ground robots or controlled indoor environments with fixed parameter settings. Consequently, there remains a limited understanding of how RTAB-Map behaves in UAV contexts, where challenges such as higher flight speeds, rapid viewpoint changes, and real-time processing demands are more critical. This gap shows the need for investigations into how key parameters such as camera resolution, frame rate (FPS), odometry strategy, and loop closure thresholds affect RTAB-Map's performance when deployed on UAVs. Addressing this gap is important for adapting RTAB-Map to aerial robotics and ensuring robust autonomous navigation in real-world scenarios.

Hence, this study focuses exclusively on RTAB-Map and conducts a parameter sensitivity analysis to investigate the optimum configuration for UAV-based SLAM.

III. METHODOLOGY

A. System Interface

In this study, a UAV modelled NXP KIT-HGDRONEK66 was used, where the devices for communication and flight control using PX4 [16] flight management unit were on board. Fig. 1 shows the front and top views of the assembled UAV. All devices are mounted on the UAV bar, and the camera is attached underneath the UAV at its centre of gravity to ensure the UAV's stability. The overall system interface is illustrated in Fig. 2.

On board the UAV, a Raspberry Pi 4B was mounted to transmit data from the Intel RealSense D435i RGB-D camera [17] to the remote PC. Both the Raspberry Pi 4 and the remote PC run Ubuntu and Robot Operating System (ROS) as the system employs a ROS-based communication architecture. The remote PC is configured as the ROS master. The remote PC also runs SLAM using RTAB-Map. The Raspberry Pi on the UAV operates as a ROS node, publishing data from the RealSense camera to the RTAB-Map node on the remote PC.

In addition to data transmission, telemetry communication between the UAV and the remote PC is established using the Holybro SiK Telemetry Radio V3 device. This communication link enables real-time monitoring of flight parameters and the ability to issue control commands through the QGroundControl software on the remote PC.

Fig. 1. Front view (left) and top view (right) of the assembled UAV.

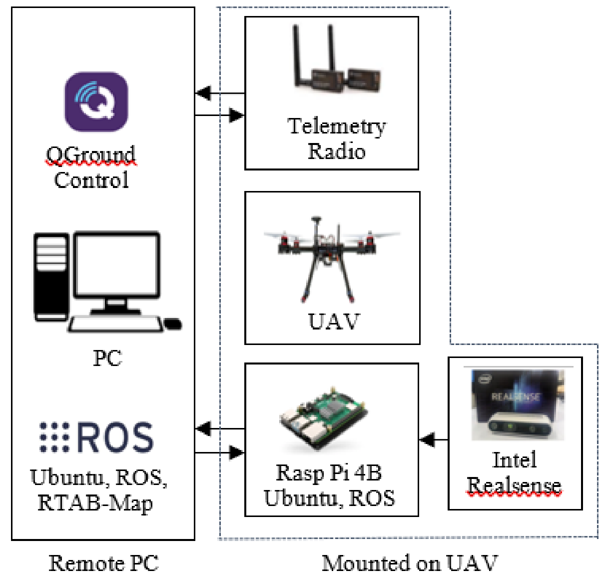

Fig. 2. Overall system interface of the RGB-D SLAM system.

B. Effective Parameters Tuning

The goal of this parameter study is to identify the optimum configuration of parameters that yield better performance of the RTAB-Map SLAM algorithm for UAVs. The parameters that were studied and tuned are as follows:

- RGB and depth camera resolution,
- RGB and depth camera number of fps,
- visual odometry strategy, and
- The minimum inliers threshold for loop closure detection

The resolution of RGB and depth cameras are parameters that study the correlation between the quality of images with transmission rate performance. A higher camera resolution yields a sharper image and gives more visual features in the surroundings. However, a higher quality image requires more processing and hence has an impact on the transmission rate or speed of the data sent to the ROS topics.

As for the RGB and depth camera fps, they are important parameters as they affect the frequency of data being published to the ROS topics. If there are more frames per second captured by both RGB and depth cameras, the data needs to be published to the RGB and depth topics more often. Hence, requires a higher transmission frequency.

In RTAB-Map, the RGB and depth topics are used to estimate the trajectory of the RGB-D sensor and to build a 3D map. Fig. 3 shows a redrawn rqt graph from ROS, considering only the relevant nodes and topics involved in this process. For better distinction, only the nodes are colored. The rqt graph shows that the /camera/realsense2_camera_manager node publishes all sensor data, including RGB, depth, and IMU information. This data is then processed by other nodes. The /rtabmap/rgbd_odometry node subscribes to the color and depth topics to perform visual odometry, estimating the localization of the RealSense camera. This localization is considered the UAV's position and is published to the /rtabmap/odom topic. The /rtabmap/rtabmap node subscribes to /rtabmap/odom, color, depth, and IMU data to build a 3D map and localize the UAV within the global map. It should be noted that the /rtabmap/rgbd_odometry node computes local odometry incrementally, the /rtabmap/rtabmap node performs global pose graph optimization and also loop closure detection. The mapping results, including the optimized trajectory and 3D map, are published on the /rtabmap/mapData topic.

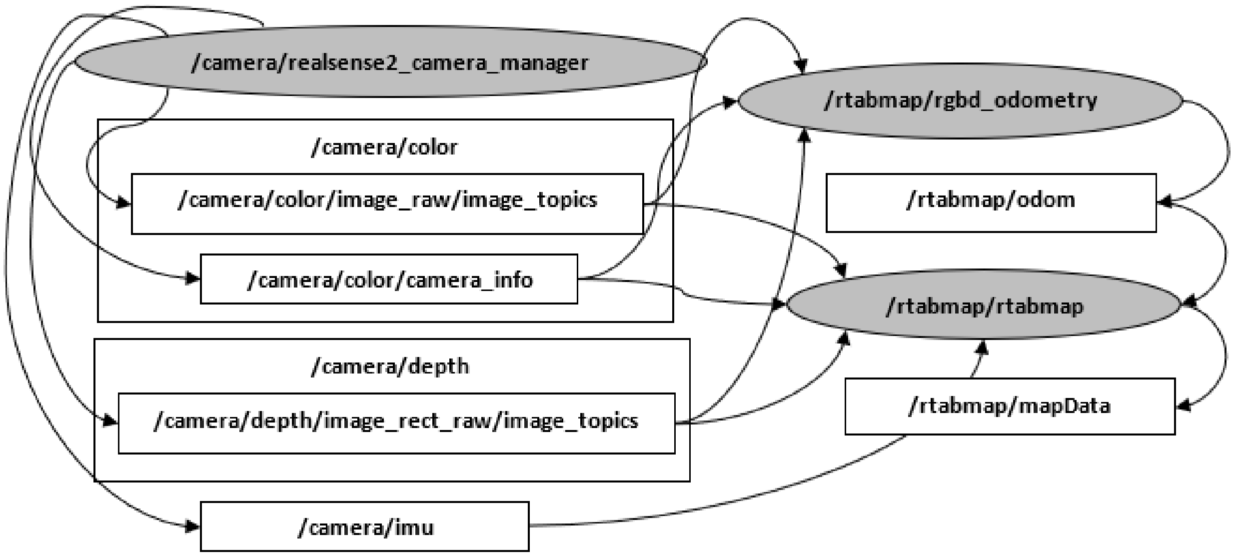

Fig. 3. A part of rqt graph when executing RTAB-Map.

The trajectory of the UAV is estimated using the visual odometry strategy implemented [14]. Visual odometry is the process of determining odometry information using sequential camera images to estimate the distance travelled by the RGB-D sensor on board, hence the UAV. The accuracy of visual odometry affects the overall performance of RTAB-Map. In this study, the visual odometry strategy was studied to determine its significance of impact. Visual odometry strategy that were tested includes frame-to-frame and frame-to-map.

In frame-to-frame odometry, the algorithm compares each image frame (or camera view) directly to the previous frame to estimate the trajectory. This is computed in the /rtabmap/rgbd_odometry node. Although this approach is efficient, as only a few consecutive images are compared, however, it is more prone to error accumulation over time. This strategy is suitable for real-time applications where some drift is acceptable. In frame-to-map odometry, the algorithm compares the current frame to an existing global map, which has been built gradually from past frames. This is computed in /rtabmap/rtabmap node. It is a more accurate approach as it references the entire map, but it can be slower and more computationally intensive. This approach is suitable for long-term mapping.

The last parameter studied is, minimum inliers threshold. It encapsulates how well the algorithm accepts input data as loop closure. Theoretically, a higher value allows the algorithm to only accept frames with more features to perform loop closure. This will result in lesser loop closures being detected as more matching visual features are required.

C. Performance Analysis

This section highlights the corresponding metrics that were observed and measured for performance evaluation purposes. The performance of the RGB-D SLAM algorithm was evaluated through a series of different categories, which are computational efficiency, robustness, precision and accuracy. Computational efficiency measures the amount of time required to process and transmit data. Metrics that were observed for this criterion are the publishing time of topics and the rate of data transmission. The ability of the algorithm to handle noise, outliers and other sources of errors defines the robustness criterion, where it is represented by the percentage of successful runs. A successful run is defined by a situation where a loop closure is successfully detected during a mapping sequence, and there are no odometry loss occurrences. The precision criterion measures the resolution and detail of the map produced by the algorithm. Keypoint detection per millisecond is observed to determine the precision of the algorithm. Last but not least, total distance travelled and distance between landmarks are the metrics measured to determine the accuracy of the algorithm, where the distance computed by the algorithm is compared with the ground truth.

IV. Results and Discussion

This section encapsulates the results obtained from experiments conducted based on the methodology mentioned in previous section. The results were split based on different environments to observe the performance of the algorithm. Analysis on the findings of the experiments were discussed in this section. The effects of parameter tuning can be demonstrated through the analysis conducted.

A. Camera Parameter Optimization

Through research and experiments, it is noted that the publishing rate of topics is dependent on the Wi-fi signal, resolution and fps of the RGB and depth camera. The initial configuration is shown in the middle column of Table I. In the initial configuration, no data was published to the camera topics, preventing the algorithm from running. After further investigation, these parameters need to be tuned in order to yield better performance in terms of topics' publishing rate. The Wi-fi signal band of 2.4 GHz is better for long-range applications but has a slower speed, while 5 GHz gives significant improvement in speed but is lacking in terms of range coverage.

The fps of the camera, on the other hand, affects the transmission performance. This is because higher fps results in the camera capturing more frames that need to be processed every second. This may lead to a bottleneck, where the rate of frames being processed to be published as ROS topics exceeds the system's computational capacity, causing frames to be dropped and not transmitted. Overloading of the processing performance causes frames to be dropped and not transmitted. After switching to the 5 GHz band and reducing the fps of both the RGB and depth camera to 15 fps, the publishing rate of the topics was improved to an average of 7 Hz.

Further tuning of the parameter was conducted to obtain the best publishing rate performance. By using a lower setting of resolution for both RGB and depth camera, an experiment was conducted to observe the publishing rate. The new configuration that was tested can be observed in the rightmost column of Table I. As a result, the publishing rate of both the RGB and depth topics was further improved to around 15 Hz, which could be observed in Fig. 4 and Fig. 5. It can be concluded that transmission performance is directly affected by the resolution of RGB and depth cameras. This shows that the RGB and depth topics require minimal delay or bottleneck when processing the topic to be able to publish at an acceptable and consistent rate.

B. Indoor Environment Mapping Experiment

The RTAB-Map algorithm was first run on a Raspberry Pi without integration to the UAV. The Intel RealSense D435i camera was connected to the Raspberry Pi 4B, and handheld mapping was conducted inside our research laboratory in the faculty. This experiment was conducted to analyze the effect of parameter tuning in an indoor environment in terms of precision and accuracy, as well as to identify the most robust parameter configuration. The experiment was successfully conducted, where a 3D map and an occupancy grid map of the laboratory were obtained. Occupancy grid map is a map representation that is commonly used by the SLAM algorithm that generate precise metric maps which are close to the detailed environmental representations [7]. Fig. 6 and Fig. 7 show both maps, respectively.

TABLE I. INITIAL AND UPDATED CONFIGURATION

Parameter	Initial Value	Updated Value
Wi-fi band	2.4 GHz	5 GHz
RGB camera fps	30	15
Depth camera fps	30	15
Resolution of RGB camera	640 x 480	424 x 240
Resolution of depth camera	640 x 480	480 x 270

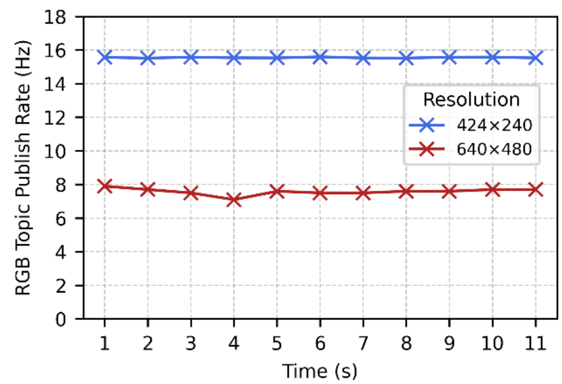

Fig. 4. Graph of publish rate of RGB topic vs. time.

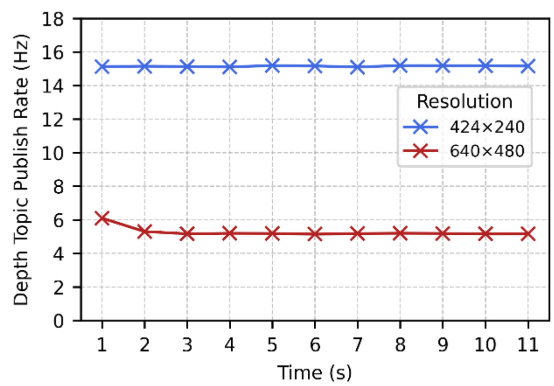

Fig. 5. Graph of publish rate of depth topic vs. time.

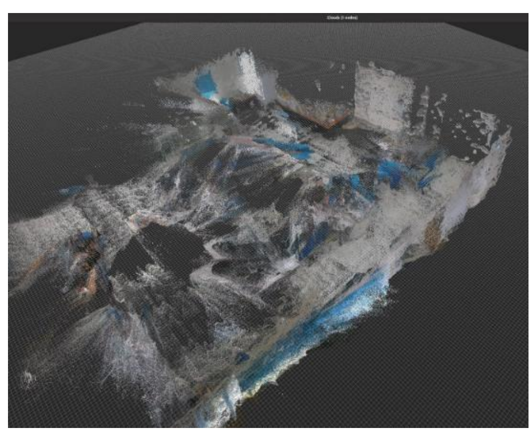

Fig. 6. 3D map of handheld mapping.

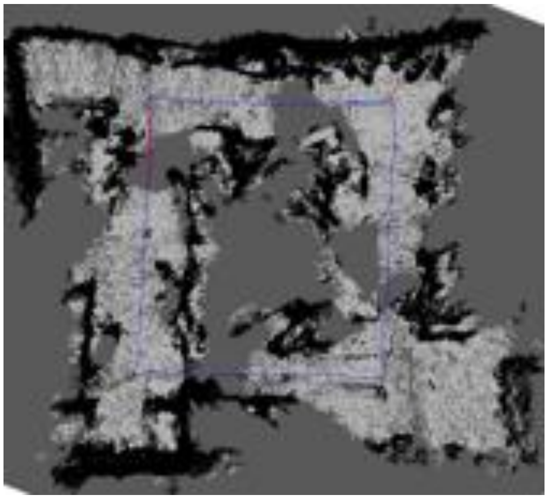

Fig. 7. Occupancy grid map of handheld mapping.

The experiment was conducted through a series of different parameter configurations. The parameters that are tuned were the visual odometry strategy and the minimum inliers threshold. As mentioned in Section III B, there are two visual odometry strategies, which are frame-to-map and frame-to-frame. On the other hand, the values used for the minimum inliers threshold were 10 and 20. Firstly, the robustness of the algorithm with these different parameter configurations was observed. In a total of 11 trials for each parameter configuration, the percentage of successful runs was computed. It was identified that the frame-to-frame odometry strategy with a minimum inliers threshold value of 10 showed the best performance in terms of robustness of the algorithm, with a 90.91% rate of successful runs.

We can observe that by changing the visual odometry strategy from frame-to-map to frame-to-frame, for each minimum inliers threshold, the value increases the robustness performance by at least 9.09%, as depicted in the bar chart in Fig. 8. The advantage of using a frame-to-frame setting over a frame-to-map is that the tracked features don't dip below a ratio of the previous frame the robot had seen. Hence, if the robot is not moving, the key frame would remain static and would remember the features in that frame. In this indoor environment experiment, the frame-to-frame odometry strategy provided better odometry compared to the frame-to-map odometry strategy, which indirectly increases the loop closure detection of the algorithm.

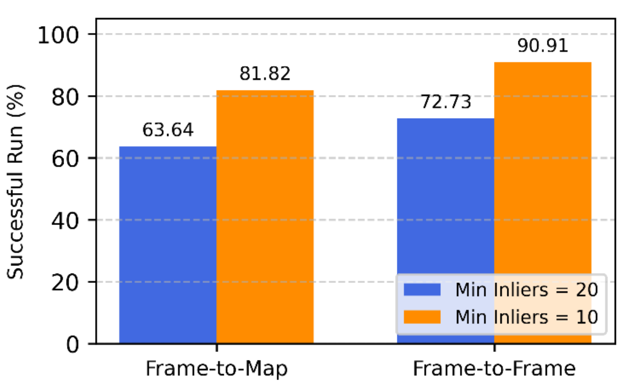

Fig. 8. Percentage of successful runs in the indoor environment.

Next, we can also observe from the bar chart in Fig. 8 that, by lowering the minimum inliers threshold from 20 to 10 for each visual odometry strategy boosts the robustness by at least 18.18%. A lower minimum inliers threshold results in frames with less features to be accepted as keyframe, which are then used for computation of localization and mapping by RTAB-Map algorithm. When the keyframe acceptance is more lenient, then the running performance of the RTAB-Map algorithm could be increased, as shown in the indoor environment mapping experiment.

The experiment was also conducted to evaluate the performance in terms of precision. For the frame-to-map visual odometry strategy, the keypoint detection per ms was observed and tabulated in the first two rows of Table II. The subsequent rows show the keypoint detection per ms for the frame-to-frame visual odometry strategy. For both strategies, a minimum inliers threshold value of 20 shows a higher keypoint detection per ms value compared to using a threshold value of 10. The observed results show that when the minimum inliers threshold is set to a higher value, the keypoint detection per ms will be higher. This is because a higher number of features need to be recognized within a frame for it to be accepted as a keyframe before proceeding to evaluate the next frame. Consequently, these selected frames can be considered of higher quality as they show to have more keypoints (refer to Table II). It can be concluded that applying a stricter threshold ensures the extraction of more keypoints, resulting in a more precise computation of the surrounding environment.

TABLE II. KEYPOINT DETECTION PER MS

Visual odometry	Min inliers threshold	Keypoint/ms
Frame-to-map	20	5.83
	10	5.20
Frame-to-frame	20	5.80
	10	4.86

Evaluation of performance in terms of accuracy was also conducted, where total distance travelled and the distance between two landmarks: Landmark 1 and Landmark 2, in meters, were observed and compared relative to the ground truth. Table III shows the estimated distance of the RTAB-Map algorithm. This shows that using RTAB-Map, similar to other common SLAM algorithms, the error is accumulated over the distance travelled. In this experiment, the error in the estimated trajectory is approximately 9.1 meters, which corresponds to about 26.6% error relative to the actual distance of 34.23 meters. In contrast, for a single measurement between Landmark 1 and 2, the error is only 0.15 meters, or approximately 7.5%, indicating that initial measurements are typically more accurate than long trajectory estimation.

TABLE III. DISTANCE ERROR FOR INDOOR ENVIRONMENT

Experiment	Actual (m)	RTAB-Map (m)	Error
Distance travelled	34.23	43.33	0.266
Landmark 1 and Landmark 2	2.00	2.15	0.075

Overall, it can be concluded that parameter configuration with the frame-to-frame visual odometry strategy and a minimum inliers threshold of 10 yields the most robust performance. This is due to the fact that it has higher trajectory accuracy and more lenient keyframe acceptance. However, there is a trade-off in terms of precision and accuracy where the generated 3D map is less dense and has fewer keypoints compared to the frame-to-map odometry strategy with a minimum inliers threshold of 20 parameter configuration.

C. Outdoor Environment Mapping Experiment

Before outdoor environment mapping is conducted using a UAV, an experiment with handheld mapping was first conducted to analyze the performance of the algorithm with a different layout. The purpose of this is to get the basis of the performance of the RTAB-Map algorithm in an outdoor environment. Since the nature of UAVs' flight is high-speed movement and maneuverable, a robust algorithm is desired to ensure a sustainable mapping process. Hence, the parameter configuration with the highest success rate was chosen, which was the frame-to-frame and minimum inliers threshold of 10. The experiments were executed at two different outdoor scenarios around the faculty: outdoor scenario 1 and outdoor scenario 2. As a result of the experiment, the 3D maps were successfully obtained for both of the outdoor scenarios, as depicted in Fig. 9 and Fig. 10.

Fig. 9. 3D Map for outdoor 1.

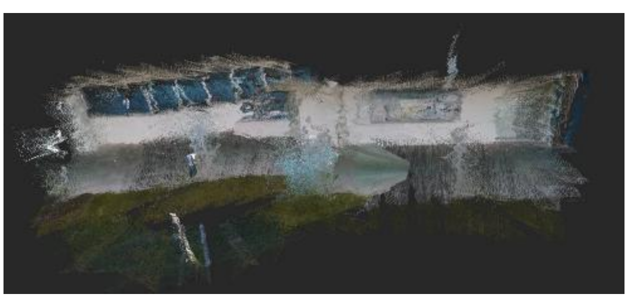

Fig. 10. 3D Map for outdoor 2.

Scenario of outdoor 1 and outdoor 2 yields different performance in terms of robustness and accuracy. The mapping sequence showed a 50% success rate for outdoor 1, while outdoor 2 demonstrates an 83.33% success rate out of a total of 6 trial runs, as shown in Fig. 11. In terms of accuracy, the total distance travelled, in meters, was measured and compared with the ground truth, as shown in Table IV. Outdoor 1 showcase a higher error (0.8283) compared with outdoor 2 (0.5676). Both these errors are still high and show that the accuracy of the trajectory travelled using RTAB-Map is less accurate. The longer the total distance travelled, increases the inaccuracy of the generated map due to accumulated error.

However, the computed error between landmarks remains relatively low for both outdoor 1 and outdoor 2, with values of 0.0259 and 0.0268, respectively (see Table V). This demonstrates that RTAB-Map is more reliable for short-term mapping that does not involve long distances travelled, as it can tolerate small drifts and still maintain a consistent map.

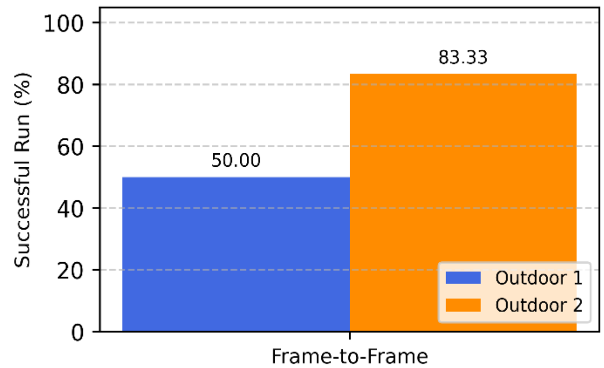

Fig. 11. Percentage of successful runs in outdoor environment.

TABLE IV. TRAVELLED DISTANCE ERROR FOR OUTDOOR ENVIRONMENT

Experiment	Actual (m)	RTAB-Map (m)	Error
Outdoor 1	16.53	30.22	0.8283
Outdoor 2	14.3	22.42	0.5676

TABLE V. DISTANCE BETWEEN LANDMARK ERROR FOR OUTDOOR ENVIRONMENT

Experiment	Actual (m)	RTAB-Map (m)	Error
Outdoor 1	3.18	3.26	0.0259
Outdoor 2	3.67	3.76	0.0268

V. LIMITATION

Although this study provides some insights into the effect of parameter configurations on RTAB-Map performance for UAV-based SLAM, several limitations remain. First, the experiments were conducted under controlled conditions with specific UAV hardware such as Intel RealSense D435i camera. The results may therefore not fully generalize to other UAV platforms, sensors, or outdoor environments subject to more dynamic conditions such as wind and lighting variations. Second, the evaluation was limited to a selected set of parameters; camera resolution, frame rate, visual odometry strategy, and loop closure inliers. While these parameters are critical, RTAB-Map contains additional configuration options (e.g., memory management strategies, feature extraction settings, and graph optimization techniques) that could further influence performance. Third, computational resources were constrained by the onboard UAV hardware, which may have affected the algorithm's scalability in larger or more complex environments.

VI. Conclusion and Future Works

RTAB-Map is an RGB-D SLAM algorithm with a loop closure detector that employs a bag-of-words approach to determine the possibility of a fresh image to be coming from a prior or new place. In this research, the loop closure detector of RTAB-Map utilizes both RGB and depth data from Intel RealSense D435i in making the decision whether to accept or deny loop closure hypotheses.

This project generally studies the applicability and relevance of SLAM algorithm implementation with UAVs by collecting all the important data and conducting a thorough assessment. Effective parameters and their relative effects on the performance of the SLAM algorithm were evaluated accordingly. The effective parameters were identified as RGB and depth camera resolution, RGB and depth camera fps, visual odometry strategy and minimum inliers threshold. The effect of tuning the parameters was then investigated in different environments (indoor and outdoor) to better understand the significance of the impact. An analysis was conducted to determine the best configuration of parameters to be applied for outdoor environment mapping by the UAV. Frame-to-frame odometry strategy minimum inliers threshold value of 10, proved to be the most robust setting to be implemented for UAV. However, there will be repercussions faced by the algorithm, where the performance will suffer in terms of precision and accuracy. Although the optimal parameters were identified for maximum robust performance of RTAB-Map, it still was not enough to cope with the movement of the UAV, as sharp turns would cause the algorithm to halt due to odometry data being lost.

Future research can extend this work in several directions. A broader exploration of RTAB-Map parameters and their interdependencies would provide deeper insights into optimizing UAV-based SLAM. Furthermore, integrating adaptive parameter tuning methods, such as machine learning-based auto-configuration, could allow UAVs to dynamically adjust SLAM parameters in response to environmental changes. Finally, long-term autonomous flight experiments, including large-scale outdoor mapping and multi-UAV collaborative SLAM, represent promising directions for advancing the robustness and applicability of RTAB-Map in real-world UAV operations.

Acknowledgment

The authors would like to thank the Centre for Research and Innovation Management (CRIM), Centre for Telecommunication Research & Innovation (CeTRI), Fakulti Teknologi dan Kejuruteraan Elektronik dan Komputer (FTKEK), Universiti Teknikal Malaysia Melaka (UTeM) and the Ministry of Higher Education Malaysia for the grant FRGS/1/2023/ICT02/UTEM02/2 which has made this research possible.

References

[1] M. Mahrami, M. N. Islam, and R. Karimi, “Simultaneous Localization and Mapping: Issues and Approaches,” 2013.

[2] S. Thrun and J. J. Leonard, “Simultaneous Localization and Mapping.”

[3] S. Wang, Z. Wu, and W. Zhang, “An overview of SLAM,” in Proc. 2018 Chinese Intelligent Systems Conference: Volume I, 2018, pp. 673–681.

[4] R. Azzam, T. Taha, S. Huang, and Y. Zweiri, “Feature-based visual simultaneous localization and mapping: a survey,” SN Applied Sciences, vol. 2, no. 2, pp. 1–20, Feb. 2020, doi: 10.1007/s42452-020-2001-3.

[5] T. T. O. Takleh, N. A. Bakar, S. A. Rahman, R. Hamzah, and Z. A. Aziz, “A brief survey on SLAM methods in autonomous vehicles,” Int. J. Eng. Technol. (UAE), vol. 7, no. 4, pp. 38–43, 2018, doi: 10.14419/ijet.v7i4.27.22477.

[6] Huijing Z., Masaki C., Ryosuke S., Xiaowei S., Jinshi C., and Hongbin Z., “SLAM in a Dynamic Large Outdoor Environment using a Laser Scanner,” IEEE Xplore, 2008.

[7] S. Saat, W. N. Abd Rashid, M. Z. M. Tumari, and M. S. Saealal, “Hectorslam 2D mapping for simultaneous localization and mapping (SLAM),” in Journal of Physics: Conf. Series, vol. 1529, no. 4, p. 042032, 2020, doi: 10.1088/1742-6596/1529/4/042032.

[8] M. Essaaidi, M. Nemiche, and IEEE Morocco Section, “SLAM algorithms implementation in a UAV, based on a heterogeneous system: A survey.”

[9] K. Luo, M. Lin, P. Wang, S. Zhou, D. Yin, and H. Zhang, “Improved ORB-SLAM2 Algorithm Based on Information Entropy and Image Sharpening Adjustment,” Math Probl Eng, vol. 2020, 2020, doi: 10.1155/2020/4724310.

[10] S. Zhang, L. Zheng, and W. Tao, “Survey and Evaluation of RGB-D SLAM,” IEEE Access, vol. 9, pp. 21367–21387, 2021, doi: 10.1109/ACCESS.2021.3053188.

[11] Z. Chen, W. Sheng, G. Yang, Z. Su, and B. Liang, “Comparison and analysis of feature method and direct method in visual SLAM technology for social robots,” in Proc. 2018 13th World Congress on Intelligent Control and Automation (WCICA), Jul. 2018, pp. 413–417, doi: 10.1109/WCICA.2018.8630714.

[12] B. Bacca, J. Salvi, and X. Cufí, “Appearance-based SLAM for mobile robots,” Artificial Intelligence Research and Development, pp. 55–64, 2009.

[13] M. Labbé and F. Michaud, “RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation,” J Field Robot, vol. 36, no. 2, pp. 416–446, Mar. 2019, doi: 10.1002/rob.21831.

[14] C. Audras, A. I. Comport, M. Meilland, and P. Rives, “Real-time dense RGB-D localisation and mapping,” in Proc. Australian Conf. on Robotics and Automation (ACRA), 2011.

[15] N. Ragot, R. Khemmar, A. Pokala, R. Rossi, and J.-Y. Ertaud, “Benchmark of visual SLAM algorithms: ORB-SLAM2 vs RTAB-Map,” in Proc. 2019 Eighth Int. Conf. on Emerging Security Technologies (EST), pp. 1–6, 2019.

[16] PX4, “PX4 robotic drone vehicle / flight management unit (FMU),” 2022. [Online]. Available: https://px4.io.

[17] Intel Corporation, “Intel® RealSense™ Camera D400 Series Product Family Datasheet Rev. 01/2019,” 2019. [Online]. Available: https://www.intel.com/content/dam/support/us/en/documents/emerging-technologies/intel-realsense-technology/Intel-RealSense-D400-SeriesDatasheet.pdf.