# Comparison of Low-Cost Odometry for Mobile Robot in Various Indoor Environments

Seungryeol Shin[1], Jaehoon Kim[2], Jaeho Shin[2], and Junghyuk Lee[2]

*Abstract*— **Indoor mobile robots, such as robot vacuum cleaners and guide robots, have been actively developed, and simultaneous localization and mapping (SLAM) is a key method for these robots to operate. Expensive sensors used in SLAM are major barriers to popularizing indoor robots. In this paper, trajectories from odometry of a mobile robot using various low-cost sensor fusions are estimated in the environments of the combinations of three environmental variables, and each result is evaluated. Odometry of Turtlebot3 is estimated through nine combinations of wheel encoders, IMU, 2D-LiDAR, monocular camera, and RGB-D camera, attached to the robot, and the results are compared to ground truth generated by motion capture equipment. The following algorithms are used: Differential Drive Kinematics; Extended Kalman Filter(EKF); Hector SLAM; Gmapping; ORB-SLAM2; Real-Time Appearance-Based Mapping (RTAB map). The experiments are conducted under eight environmental conditions, combinations of three factors that affect estimation. We find the combination of sensor fusion that shows high performance with low cost in two environments: for office environment with few features extracted: "Wheel + IMU + RGB-D" fusion; for living-room environment with many features extracted: "RGB-D Only". The sensor fusion combination that produces high performance versus price in various environments can be applied to developing inexpensive indoor mobile robots.**

## I. INTRODUCTION

Due to the appearance of autonomous driving and indoor guide robot, simultaneous localization and mapping(SLAM) technology has been developed rapidly. SLAM performs pose estimations of the robot using the real-time measurement from various sensors. In an outdoor environment, robots use GPS information to determine absolute coordinates. By contrast, it is more difficult to localize the robots position in the indoor environment because of the unavailability of GPS. Researchers have been using diverse ways of sensor fusing to overcome this constraint, for example, LiDAR-IMU fusion, VIO, camera-LiDAR fusion. Fusing such sensors guarantees a practicable performance as investigated in [1]; however, due to the issue of cost, there were trials to improve mapping and localization using low-cost sensor fusion [2]. In this study, ROS-based SLAM algorithms using 2D-LiDAR, monocular camera, and stereo camera (2D-LiDAR-based: GMapping, Hector SLAM, Cartographer; Monocular camera-based: LSD-SLAM, ORB-SLAM, DSO; Stereo camera-based: ZEDfu, RTAB-MAP, ORB-SLAM, S-PTAM) were compared with a rectangular trajectory in an indoor environment, which was typical office with monochrome walls.

The research [3] generated a ground truth with a FARO laser tracker and performed comparative analysis on the ROS-based 2D-LiDAR SLAM algorithms(Cartographer, Hector SLAM, GMapping). One study for comparison of ROS-compatible stereo visual SLAM algorithms (S-PTAM, RTAB-MAP, LIBVISO2, ORB-SLAM2, ZED-VO) was conducted [4]. The trajectory by Hector SLAM, 2D-LiDAR based algorithm, was set to a ground truth. In the paper [5], performances of overall visual SLAM algorithms, including monocular camera, stereo ZED camera, and Kinect depth sensor, were evaluated.

The above studies have the following limitations. (1) In the papers [1],[2],[4],[5], the actual ground truth coordinates of the trajectory for comparison were not known, so the result trajectory of the Hector SLAM was assumed to be ground truth. However, in our study, the precise global coordinate value of the robot is obtained with a motion capture system. Therefore, more accurate trajectory error of each algorithm can be measured. (2) From all studies mentioned above, the experiments were conducted in the same place, which meant a single environment. Therefore, researches conducted from simple trajectories cannot reflects the unpredictable features of the indoor environment in which the robot operates actually. To overcome this limitation, in our paper, three environmental factors were considered: the number of features, presence or absence of moving objects, and shape of the trajectory. (3) In mentioned papers, algorithms using only the single sensor without sensor fusion methods have been compared. In other words, investigation diversity is not sufficient due to the implementation of single sensor algorithms (Visual SLAM / LiDAR SLAM). Regarding vulnerability of low-cost sensors and complexity of indoor environment, fusing various sensor data methods must be implemented. Our research found the sensor combination and algorithm with the high performance, because conducted experiments reflected various situations of the indoor environments.

In this study, the experimental environments simulating the indoor environments were created using defined three environmental variables. First, the number of features extracted was adjusted by modifying the number of objects. Objects were excluded inside the experimental space surrounded by white paper simulating walls depicting office environment with fewer features, then various objects, such as boxes and chairs, were inserted to simulate spaces with many features like living rooms. Second, the effects of moving objects were verified by allowing people to move around the robot or not.

[1]S. Shin is with the Depart. Biosystems Engineering, SNU, Seoul, S. Korea dev.shin@snu.ac.kr

[2]J. Kim, J. Shin, and J. Lee are with the Depart. Mechanical Engineering, SNU, Seoul, S. Korea [poeq8283, leah100, jhl17]@snu.ac.kr

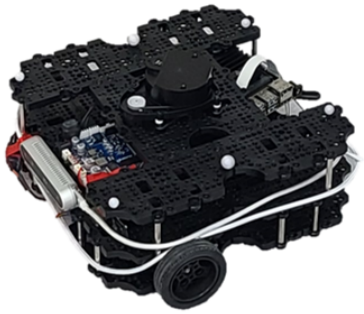

Fig. 1: Turtlebot3 Waffle Pi (ROBOTIS co.)

TABLE I: Sensor's Type & Price

| Type | Model | Price | Ref. |
|---|---|---|---|
| Wheel Encoder | AS5045 | $8.75 | [6] |
| IMU | MPU9250 | $14.57 | [7] |
| 2D-LiDAR | LDS-01 | $179.90 | [8] |
| Monocular Camera | Pi Camera | $8.99 | [9] |
| RGB-D Camera | Realsense D415 | $164.42 | [10] |

Finally, the movement of the robot was classified into straight and curved trajectories to investigate the effect of the shape of the trajectory on the odometry.

In this paper, we applied various combinations of low-cost sensors to the differential-drive mobile robot and found the proper sensor fusion that minimizes the trajectory error with respect to the ground truth data in multiple indoor environmental conditions. This study consists of the following: Section II describes the mobile robot used in this study, and Section III describes the experimental construction method, implementation of algorithm for sensor combinations and the techniques for analyzing the experimental results. Section IV performed the experimental results, and Section V performed the discussion on the results. Section VI concluded and summarized the results.

## II. ROBOT SETUP

We developed a mobile robot in Fig. 1 for the experiment. A single-board computer and various sensors ,including wheel encoders, IMU, 2D-LiDAR, and RGB-D camera, were attached to the two-wheeled differential drive mobile base. Detail of the robot hardware and software, including its URDF model, is described in this section.

### A. Robot Hardware Setup

In the paper, we remodeled a Turtlebot3 Waffle Pi of ROBOTIS. The robot base can move forward at speed up to 0.26 m/s or rotate at a speed of 104.27 deg/s, using two Dynamixel XM430-W210-T motors. The robot had an Intel Realsense D415 RGB-D camera in addition to its own 2D-LiDAR and IMU. 2D-LiDAR was on the top of the robot, and the RGB-D camera was attached in front of the robot with a laser-processed acrylic plate bracket. The single-board computer was replaced from Raspberry Pi to NVIDIA Jetson Nano to process data from the additional Realsense camera. The detailed explanation for the hardware specification of the

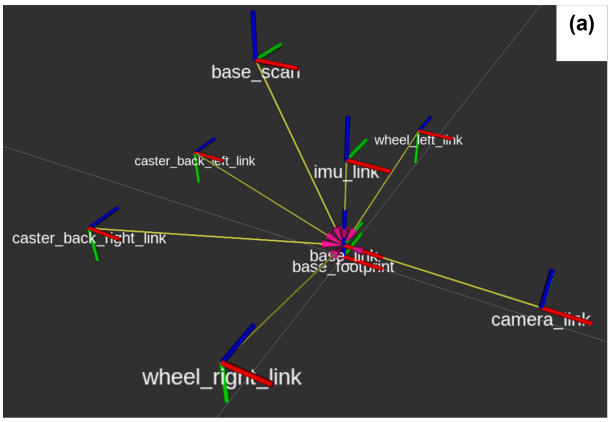


Fig. 2: The URDF model of the mobile robot (a) Visualization of URDF in rviz (b) tf tree of the model

mobile robot is in Table III. Information on the sensor types and prices is described in Table. I. In addition, the price is written based on the lowest online price.

### B. Software Setup and Robot Model

We installed Jetpack, Ubuntu OS, and Robot Operating System(ROS) on the Jetson Nano single board computer, and the version of the software is in Table III. All algorithms we employed in the research were provided as ROS packages. Therefore, we could use the tf package to calculate the relative coordinates between the base_link frame of the robot and the sensors, or simply represent the robot's odometry estimated by various algorithms. The URDF of the Turtlebot provided by ROBOTIS was modified and used for this purpose, and a result of visualization in rviz is shown in Fig. 2.

## III. METHODOLOGY

### A. Motion Capture System

We measured the ground truth pose trajectory using OptiTrack motion capture system. This ground truth data was used to evaluate the estimated trajectories. Twelve PrimeX13 [11] motion capture cameras were installed in the space where the experiment was conducted. The six spherical markers were attached to the top of the turtlebot3, and markers were arranged so that their geometric center coincided with the origin of the body-frame of the turtlebot3. OptiTrack Motive software was used for the motion capture process. In Motive, six markers were set to one rigid-body; the software stored the rigid-body pose at sampling rate of 120 Hz. The robot's position was exported in x, y, z values expressed in the global-frame, and the orientation was exported in the

TABLE II: Combinations for Low-cost Sensor Fusion

| Sensor Combinations | Algorithm | Ref. | Odom | Pose | Occupancy | Total price of sensor combinations |
|---|---|---|---|---|---|---|
| Wheel Only | Differential Drive Kinematics | [12] | ✓ | | | $8.75 |
| Wheel + IMU | EKF | [13] | ✓ | | | $23.32 |
| LiDAR Only | Hector SLAM | [14] | | ✓ | 2D | $179.90 |
| Wheel + IMU + LiDAR | GMapping | [15] | ✓ | ✓ | 2D | $203.22 |
| RGB-D Only | RTAB Map | [16] | ✓ | ✓ | 2D, 3D | $164.42 |
| LiDAR + RGB-D | RTAB Map | [16] | ✓ | ✓ | 2D, 3D | $344.32 |
| Wheel + IMU + RGB-D | RTAB Map | [16] | ✓ | ✓ | 2D, 3D | $187.74 |
| Wheel + IMU + LiDAR + RGB-D | RTAB Map | [16] | ✓ | ✓ | 2D, 3D | $367.64 |
| Monocular | ORB-SLAM2 | [17] | | | | $8.99 |

TABLE III: Hardware Specification of the Mobile Robot

| **Hardware Setup** | |
|---|---|
| Mobile Base | Turtlebot3 Waffle Pi |
| Motor | Dynamixel XM430-W210-T |
| Wheel Encoder | AMS AS5045 |
| SBC | Nvidia Jetson Nano |
| **Software Setup** | |
| Jetpack | 4.6 |
| OS | Ubuntu 18.04 |
| ROS | Melodic |

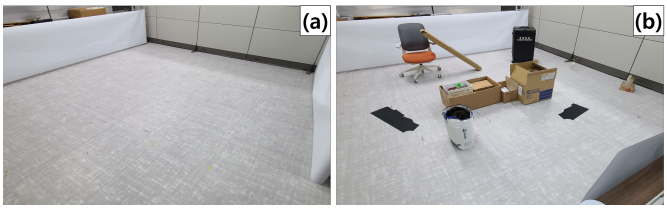


Fig. 3: (a) Office environment, where fewer features are extracted, (b) Living-room environment, where many features are extracted

form of a quaternion. These values were used for comparison with the estimated trajectories using various algorithms.

### B. Environmental Variables

We designed the experiment to verify the effect of environmental variables that affect the performance of trajectory estimation in the indoor place. Three key variables were selected: the number of features extracted, the presence of the moving objects, and the shape of the robot trajectory.

*1) The Number of Features Extracted:* We set two places as the environmental variable that could greatly influence algorithms using extracted features; we classified the indoor environment, where people live, into two main types. The first is a place composed of repeated and featureless walls and floors, such as offices and corridors, and there are not many objects. In these spaces, there are few extracted features and the performance of algorithms using features can decrease. In this paper, we decide to call this place "office" environment. The second is a place with various types of objects, such as a living room. In this place, many features can be extracted in the case of SLAM algorithms using features. In this paper, we decide to call this place "living-room" environment. In order to evaluate the effectiveness of environmental variables affecting the number of features, we organized the following two environments and conducted the experiment.

- Office(Environment with fewer extracted features): As shown in Fig. 3(a), The walls of the space, where the experiment was conducted, were covered with white paper to reduce the number of features extracted. We removed all objects in the space where the robot moved.
- Living-room(Environment with many extracted features): As shown in Fig. 3(b), in the same place as the office, we placed chairs, boxes, carriers, cloths, so that there were many objects that affected the RGB-D camera and the 2D-LiDAR.

*2) The Presence of The Moving Objects:* In the real indoor environment, there are people around the robot. Humans move dynamically and interfere with the robot's estimation process. We conducted experiments both with and without moving people to verify the effects of moving objects in the environment. There were two people moving in the environmental space when the moving-object-exist condition, and they continued to travel around the area included in the range of the robot's RGB-D camera and 2D-LiDAR at normal walking speed. There was no physical contact with the robot. The experimental environment without people is defined as "No Human," and the experimental environment with people moving is defined as "With Human."

*3) The Shape of The Robot Trajectory:* We also considered experimental conditions to compare the difference between the case where the path consists of straight lines and pure rotations and the case where the path consists of continuous curves. Path differences can affect wheel slip and the performance of the RGB-D sensor with a narrow viewing angle. The trajectory composed of straight lines and pure rotations is defined as "straight"(Fig. 4(a)). The trajectory composed of continuous curves is defined as "zig-zag"(Fig. 4(b)).

Eight experiments were conducted to verify the influences of the above three environmental variables. Eight experimental cases are defined as scenarios form S1 to S8. The environmental variables corresponding to each scenario are summarized in Table. IV.

All sensor data in each scenario was stored in the form of rosbag, and ground truth data of the robot was also stored by the motion capture system.

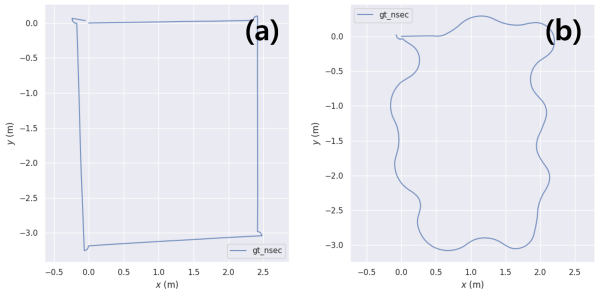


Fig. 4: The shape of the robot trajectory. (a) Straight, (b) zig-zag

### C. Implementation

The dataset of each scenario was used for trajectory estimation using nine SLAM algorithms. The nine "Trajectory" for each scenario are made through a combination of sensors mentioned in Table. I, and the algorithms used are shown in Table. II.

*1) Wheel Only:* The Differential Drive Kinematics[12] is derived from two pairs of wheels (the actual drive wheels and the encoder wheels that generate odometry measurements). A first order odometry error model has been derived to propagate the state error covariance following a motion. Because the two drive wheels are driven by two different motors, two separate optical shaft encoders are used to gather odometry information.

*2) Wheel + IMU:* The trajectory of this combination was obtained through Extended Kalman Filter[13]. Extended Kalman Filter is used to estimate the pose of a robot, based on partial pose measurements coming from different sensors. It uses a differential-drive motion model to project the state forward in time and corrects that projected estimate using perceived sensor data.

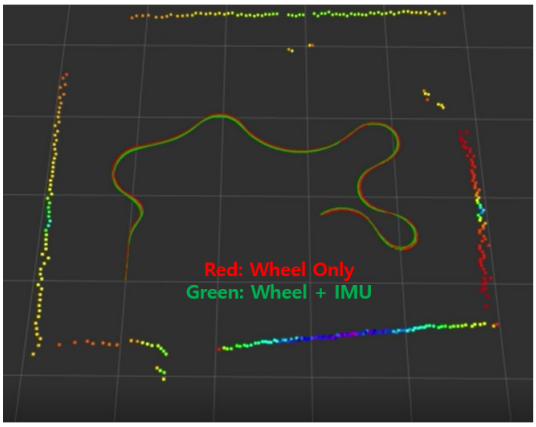


Fig. 5: "Wheel Only" and "Wheel + IMU" trajectory in Rviz

*3) LiDAR Only:* The trajectory of this combination was obtained through Hector SLAM[14]. Hector SLAM can create fast 2D occupancy grid maps from a 2D-LiDAR with low computation resources. It is not exactly a full SLAM approach as it does not detect loop closures, and thus the map cannot be corrected when visiting back a previous localization. Hector SLAM does not need external odometry, which can be an advantage when the robot does not have one, but can be a disadvantage when operating in an environment without a lot of geometry constraints, limiting laser scan matching performance.

*4) Wheel + IMU + LiDAR:* The trajectory of this combination was obtained through GMapping[15]. GMapping uses a particle filter to estimate the robot trajectory. As long as there are enough estimated particles and the real position error corresponds to the covariance of the input odometry, the particle filter converges to a solution that represents the environment well, particularly for GMapping when there are loop closures. GMapping has been widely used to derive a 2D occupancy grid map of the environment from 2D laser scans.

*5) RGB-D based sensor combinations:* Through Real-Time Appearance-Based Mapping(RTAB Map)[16], maps and trajectory can be obtained for various combinations such as "RGB-D Only", "LiDAR + RGB-D", "Wheel + IMU + RGB-D", and "Wheel + IMU + LiDAR + RGB-D". RTAB Map is an RGB-D, Stereo, and Lidar Graph-Based SLAM approach based on an incremental appearance-based loop closure detector. The loop closure detector uses a bag-of-words approach to determine how likely a new image comes from a previous location or a new location. When a loop closure hypothesis is accepted, a new constraint is added to the maps graph, and then a graph optimizer minimizes the errors in the map. Fig. 6 represents a block diagram for RTAB Map ROS node and can receive various sensor inputs. The results of RTAB Map for the experimental environment are shown in Fig. 7.

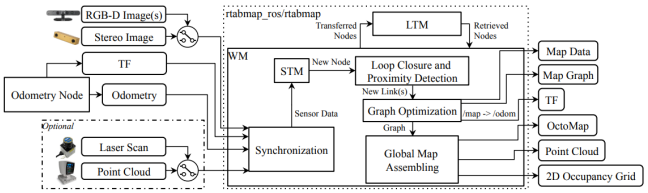


Fig. 6: Block diagram of RTAB Map ROS node including various sensors input

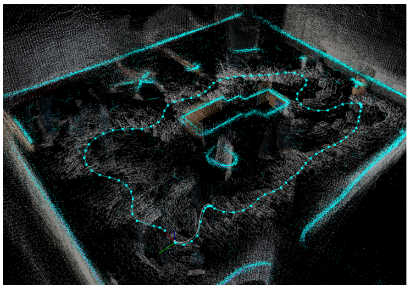

Fig. 7: RTAB Map reusult for sensor combinations (Wheel, IMU, LiDAR and RGB-D) in our dataset

*6) Monocular:* The trajectory of this combination was obtained through ORB-SLAM2[17]. ORB-SLAM2 is feature-based visual SLAM approach that can be used with a monocular camera. When a loop closure is detected, the map is optimized using bundle adjustment. Graph optimization after loop closure is done in a separate thread to avoid influencing camera tracking frame rate performance. Loop closure detection and graph optimization processing time increase as the map grows, which can make loop closure correction happen with a significant delay after being detected. The approaches maintain a sparse feature map. As a result of the experiment for eight scenarios, when monocular alone is used, it is very vulnerable to fast rotation and results as shown in the Fig. 8 with tracking loss. Since each scenario includes all fast rotations, tracking loss occurs for all scenarios, so the results for "Monocular" are excluded from this paper.

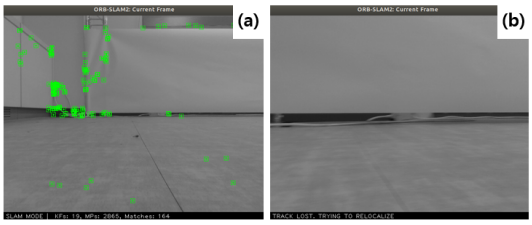


Fig. 8: Failure of ORB-SLAM2 (a) Proper features detected before rotation (b) Motion blur after rotation

### D. Evaluation Metric

A python package for the evaluation of odometry and SLAM, called evo, was used to compare the ground truth measured with the motion capture device and the error of trajectory obtained from sensor data. Through the package, it is possible to plot the robots trajectory from a given pose of several formats, e.g., TUM, KITTI, rosbag, EuRoC, and to calculate the absolute and relative errors of various trajectories compared to the reference. Also, mean, maximum, minimum, standard deviation, and median value of the coordinate difference can be obtained. Comparison of every data in this study is made with the absolute trajectory error after being aligned to the ground truth. That is, by measuring the nearest unique points $\hat{X}'_{gt}$ for each point $X_{gt}$ of ground truth with a predetermined distance metric, the RMSE value of the estimated coordinate is used. The formula of APE(absolute pose error) is as follows.

$$ATE_{pose} = \left(\frac{1}{N}\sum_{1}^{N-1} \left\|\Delta p_i^2\right\|\right)^{\frac{1}{2}} \quad (1)$$

$\Delta p_i$ is the distance between two corresponding points, and N is the number of points used for error calculation.

## IV. EXPERIMENTAL RESULTS & DISCUSSION

We applied eight algorithms to each of the eight scenarios to obtain 64 estimated trajectories. We compared 64 estimated trajectories and ground truth using evo and obtained the results of 64 RMSEs. It is impossible to analyze the effects of each environmental variable directly through these 64 results. Therefore, RMSE results were processed to show the influence of environmental variables well.

### A. Result of RMSE

Estimated trajectories from eight normally operated combinations of sensors were compared with ground truth using evo in eight scenarios, respectively. The RMSE value was calculated to analyze the results quantitatively. This is shown in Table V and the bar graphs of Fig. 10. It was confirmed that the larger the number of sensors used, the smaller the size of the RMSE and the smaller the deviation according to the scenario. Also, the result of using only 2D-LiDAR in scenarios S3 and S7 in which humans existed and the robot moved along the rectangular path also showed abnormally high RMSE value.

### B. Effects of rotation speed

As seen from Fig. 10, "LiDAR Only" SLAM showed a significant error in scenarios S3 and S7, the conditions in which moving objects existed and the robot went straight. However, in scenario S4, the environment where moving objects and not many features existed equally, the results of the overall pose estimation showed relatively low error. As shown in Fig. 11, by observing the robot's pose estimation process in scenario S3 with Rviz, the large distortion in the trajectory when rotating at the corner was found, which means the rapid rotational speed was the cause of the error. The point clouds were measured at the point farther than the previous frame due to a rapid change in the robot's pose. For the case of RTAB Map using an ICP(iterative close point) algorithm in registration, this would have made it difficult to estimate the exact pose. In this unstable situation, the point cloud by the human ,which changes its position frame by frame,had an inverse effect on the estimation process. As a result, the trajectory error of LiDAR Only in these scenarios was calculated as a large value. However, in scenario S4 the rotational speed was not large along the curved trajectory, so when the robot rotated, the point clouds of the next frame located at a near position from the measurement of the previous frame. Therefore, the registration of the point and pose of the robot were stably estimated.

### C. Effect of the number of features extracted

Scenarios from S1 to S4 are office environments where fewer features were extracted, and scenarios from S5 to S8 are living-room environments where many features were extracted. The RMSE values were processed as follows to well indicate the effect of the number of extracted features. For each algorithm, the average of RMSE in scenarios from S1 to S4 was used as the office result and the average of RMSE in scenarios from S5 to S8 as the living-room result. This data processing result is shown in Fig. 12(a).

In Fig. 12(a), the RMES results in the living-room environment with many features show better performance than in the

TABLE IV: Eight scenarios considering three environmental variables: the number of features extracted; the presence of the moving objects; the shape of the robot trajectory.

| Office (fewer features) | | | | Living-room (many features) | | | |
|---|---|---|---|---|---|---|---|
| No Human (static objects) | | Human (dynamic objects) | | No Human (static objects) | | Human (dynamic objects) | |
| Straight | Zig-zag | Straight | Zig-zag | Straight | Zig-zag | Straight | Zig-zag |
| S1 | S2 | S3 | S4 | S5 | S6 | S7 | S8 |

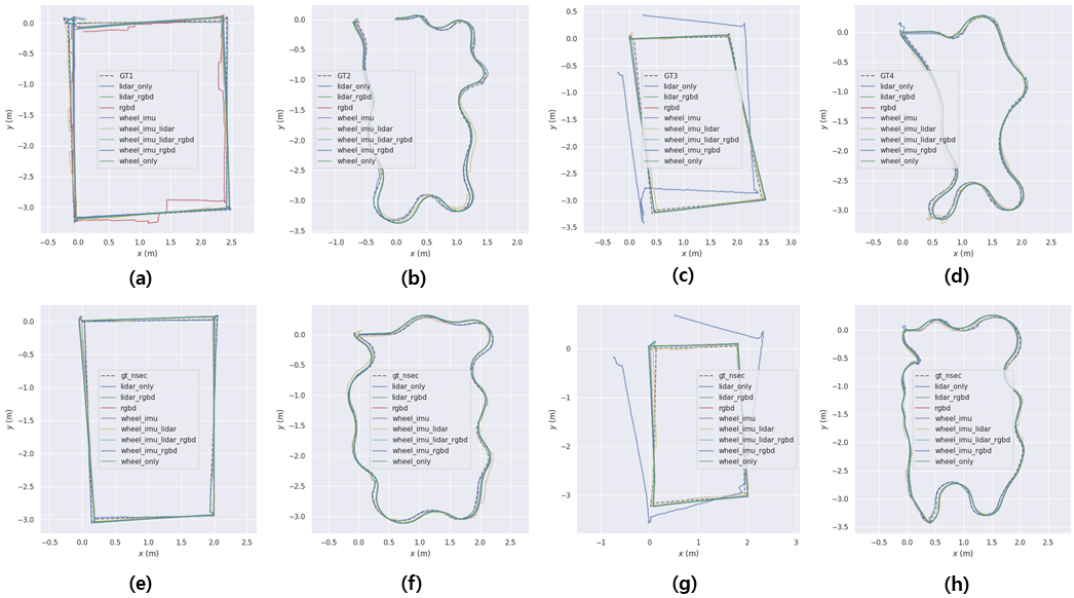


Fig. 9: Trajectory results for each scenario. (a) Scenario S1, (b) Scenario S2, (c) Scenario S3, (d) Scenario S4, (e) Scenario S5, (f) Scenario S6, (g) Scenario S7, (h) Scenario S8

TABLE V: RMSE(m) for sensor combinations

| | **S1** | **S2** | **S3** | **S4** | **S5** | **S6** | **S7** | **S8** |
|---|---|---|---|---|---|---|---|---|
| **Wheel Only** | 0.06954 | 0.05339 | 0.03855 | 0.05877 | 0.03806 | 0.04631 | 0.06250 | 0.04337 |
| **Wheel + IMU** | 0.08574 | 0.05648 | 0.03746 | 0.06107 | 0.03727 | 0.04837 | 0.05303 | 0.04591 |
| **LiDAR Only** | 0.01768 | 0.02259 | 0.46911 | 0.04344 | 0.01308 | 0.01783 | 0.51119 | 0.01890 |
| **Wheel + IMU + LiDAR** | 0.03993 | 0.06357 | 0.05397 | 0.06001 | 0.04155 | 0.06387 | 0.04754 | 0.04846 |
| **RGB-D Only** | 0.10658 | 0.06492 | 0.03787 | 0.05922 | 0.03696 | 0.04600 | 0.06087 | 0.04383 |
| **LiDAR + RGB-D** | 0.06105 | 0.05808 | 0.03760 | 0.06050 | 0.03697 | 0.04620 | 0.06134 | 0.04322 |
| **Wheel + IMU + RGB-D** | 0.06100 | 0.05459 | 0.03806 | 0.05935 | 0.03714 | 0.04631 | 0.06142 | 0.04343 |
| **Wheel + IMU + LiDAR + RGB-D** | 0.06305 | 0.05320 | 0.03774 | 0.05903 | 0.03699 | 0.04656 | 0.06126 | 0.04346 |

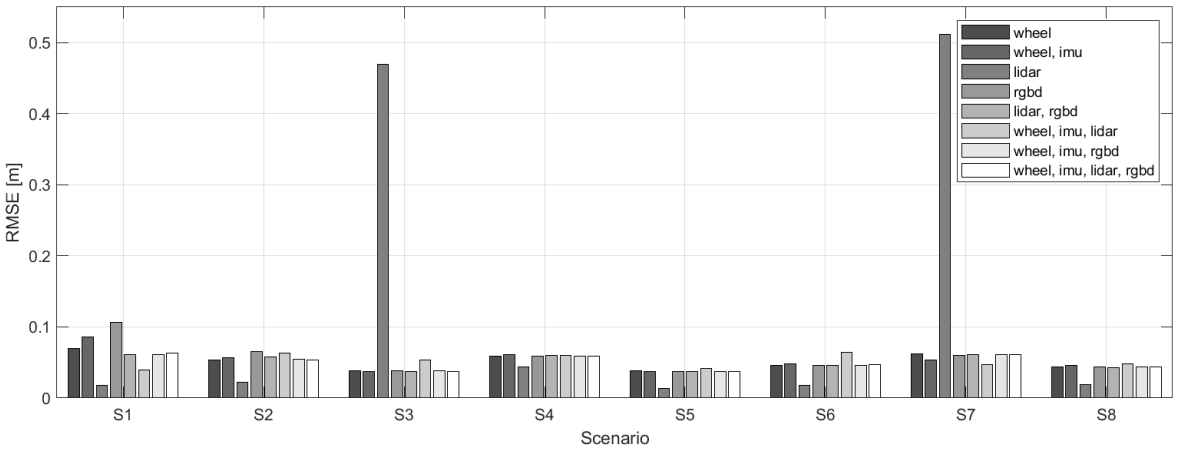


Fig. 10: Bar graph of RMSE result

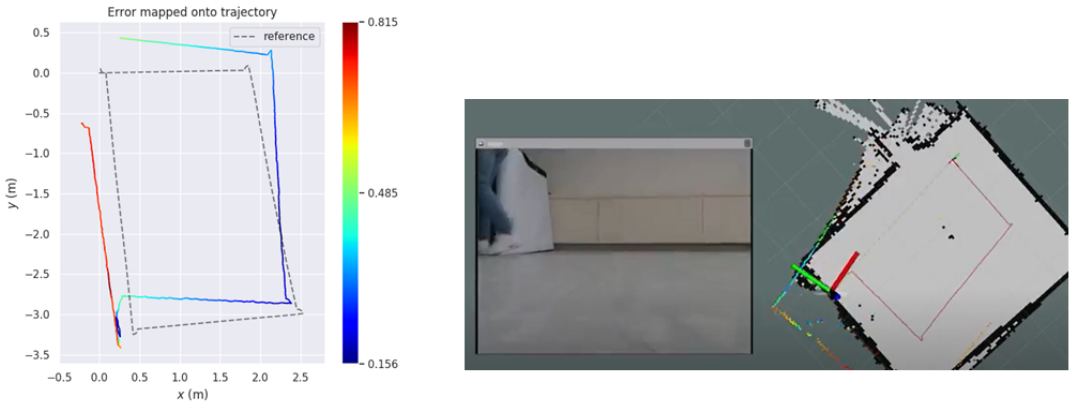


Fig. 11: (a) Trajectory of "LiDAR Only" SLAM in scenario S3, (b) Failure moment of "LiDAR Only" SLAM from Rviz

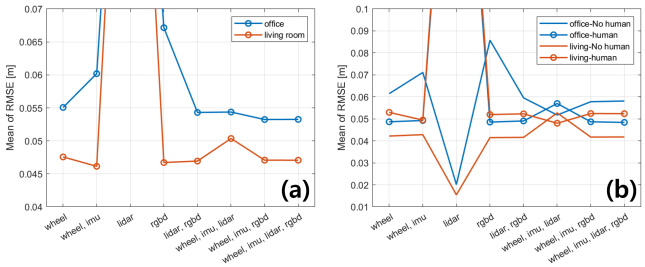


Fig. 12: For each algorithm, processed RMSE values for analysis: (a) Considering the effect of the number of features extracted. Office - average of scenarios S1∼S4, Living-room - average of scenarios S5∼S8; (b) Considering the effects of both the number of features extracted and Presence of the moving objects. Office-No human - average of scenarios S1&S2, Office-human - average of scenarios S3&S4, Living-No human - average of scenarios S5&S6, Living-human - average of scenarios S7&S8

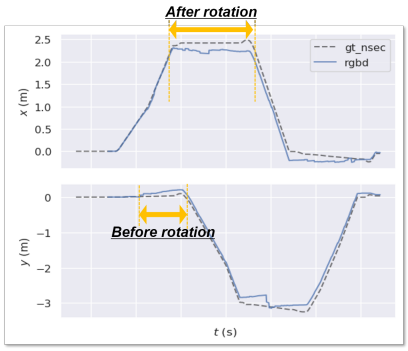


Fig. 13: Estimated trajectory using "RGB-D Only" algorithm in scenario S1 - office-no human-straight condition

office environment. Since the "Wheel Only" and "Wheel + IMU" algorithms are not affected by features, the difference of RMSE in two environments could be considered as an error due to the lack of the number of experiments.

### *D. Effects of both the number of features extracted and Presence of the moving objects*

Data processing was conducted to determine how the presence of moving objects in the office environment and living-room environment can affect the estimation results. For each algorithm, the average of scenarios S1 and S2 was used as office-no human result, the average of scenarios S3 and S4 was used as office-human result, the average of scenarios S5 and S6 was used as living-room-no human result, and the average of scenarios S7 and S8 was used as living-room-human result. These average results are shown in Fig. 12(b).

In Fig. 12(b), "RGB-D Only" data shows big error at office-no human condition. As we analyzed office-no human data of "RGB-D Only" SLAM in details from Fig. 10, we found out straight condition showed a large error while zig-zag course did not. This is because the depth camera has a narrow viewing angle. When the robot moves by straight course(in scenario S1), it could not correct the estimated position perpendicular to the direction of progress because there was no information of the wall next to the robot. However, in zig-zag, the RGB-D camera could see the wall next to it while keep rotating. This explanation is illustrated in Fig. 13. Fig. 13 shows the estimated trajectory of the "RGB-D Only" SLAM with ground truth in scenario S1, which is office-no human-straight condition. The robot's initial progress direction is the x-axis. During the initial straight line trajectory, it can be seen that the error occurs mainly in the y-axis direction perpendicular to the progress direction(x-axis) while the x-axis position almost coincides with the ground truth. After the robot's initial rotation of 90 degrees, it can be seen that the error of the x-axis perpendicular to the progress direction increases when the robot moved straight to the minus y-axis. Through these results, it is possible to explain the large RMSE of the "RGB-D Only" result under the "office-no human" condition.

From Fig. 12, we can select a sensor combination that works well in two indoor environments represented by office

and living-room. Considering the prices of sensors, the combination of sensors that perform high with minimal sensors was finally selected.

In the office environment with fewer features, "RGB-D Only" SLAM shows large RMSE value. Therefore, we need to compensate this by using wheel encoder or IMU. However, in the living-room condition with many features, "RGB-D Only" SLAM shows high performance. The methods using many sensors such as "Wheel + IMU + RGB-D" and "Wheel + IMU + LiDAR + RGB-D" have reasonable results with robustness. However, "Wheel + IMU + LiDAR" method has slightly low performance because of instability of 2D-LiDAR. "Wheel Only" and "Wheel + IMU" method shows slightly high performance compared to more expensive sensors, but it is not practicable because there are no method for pose correction if slip occurs.

In other words, in office environment with fewer features, "LiDAR + RGB-D", "Wheel + IMU + LiDAR", "Wheel + IMU + RGB-D" and "Wheel + IMU + LiDAR + RGB-D" combinations show good performances. In living-room environment with many features, "RGB-D Only", "LiDAR + RGB-D", "Wheel + IMU + RGB-D", and "Wheel + IMU + LiDAR + RGB-D" show good performances. Considering the price factor, the most expensive sensors are the 2D-LiDAR($179.90, [8]) and RGB-D camera($164.42, [10]). If there is no significant difference in performance, the sensor combination using only one of them should be selected. Therefore, we propose a combination of sensors that performs well in each indoor environment as follows. In the environment, such as an office where many features are not observed, the "Wheel + IMU + RGB-D" sensor combination is recommended. In the environment, such as the living room where many features are extracted, "RGB-D Only" is recommended for sufficient performance.

## V. CONCLUSION

In this research, we compared the performance of various sensor fusion methods from the indoor environment. The remodeled Turtlebot3 was used as a mobile robot platform within eight different scenarios reflecting various situations. The number of scenarios was resulted from three environmental variables: the number of features, the presence of the moving objects, the shape of the trajectory. The dataset from each sensor (2D-LiDAR, wheel encoder, IMU, RGB-D camera, and monocular camera) was saved as rosbag files and processed using the following SLAM algorithms: Differential drive kinematics, EKF, Hector SLAM, GMapping, RTAB map, ORB SLAM2. By using a python package for evaluating SLAM called evo, each result was compared.

- Comparison of RMSE: It was found that if larger number of sensors were used, absolute trajectory error was decreased and became robust across every scenario.
- "LiDAR Only" SLAM: Big trajectory error of Hector SLAM from scenario S3 and S7 was due to presence of moving objects and high rotational speed of the robot at the corner.
- "RGB-D Only" SLAM: "RGB-D Only" SLAM showed big positional error along the vertical direction of the movement. It was due to narrow sight angle of depth camera, which leaded to lack of depth information of the wall next to the robot.
- Optimal sensor fusion method for each environment: In the environment with many features, RGB-D Only would be enough for indoor SLAM considering cost issue, and from featureless environment, Wheel + IMU + RGB-D sensor combination showed best performance.

Since our experiment was conducted only a few times, our future work will improve statistical validity by repeating the process. Also, other sensor fusion methods including monocular camera (e.g. Wheel + IMU + Monocular) should be considered since monocular cameras are cheap.

## REFERENCES


[1] W. Peng, Y. Ao, J. He, and P. Wang, Vehicle Odometry with Camera-Lidar-IMU Information Fusion and Factor-Graph Optimization, *Journal of Intelligent & Robotic Systems*, vol. 101, no. 4, Apr. 2021.

[2] M. Filipenko and I. Afanasyev, Comparison of Various SLAM Systems for Mobile Robot in an Indoor Environment, in *2018 International Conference on Intelligent Systems (IS)*, 2018.

[3] Rauf Yagfarov, Mikhail Ivanou, and Ilya Afanasyev, Map Comparison of Lidar-based 2D SLAM Algorithms Using Precise Ground Truth, in *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2018.

[4] Riccardo Giubilato, Sebastiano Chiodini, Marco Pertile, and Stefano Debei, An Experimental Comparison of ROS-compatible Stereo Visual SLAM Methods for Planetary Rovers, in *2018 5th IEEE International Workshop on Metrology for AeroSpace (MetroAeroSpace)*, 2018.

[5] Ilmir Z. Ibragimov, Ilya M. Afanasyev, Comparison of ROS-based Visual SLAM methods in homogeneous indoor environment, in *2017 14th Workshop on Positioning, Navigation and Communications (WPNC)*, 2018.

[6] (2021, Dec.) AS5045 [Online]. Available: https://kr.mouser.com/ProductDetail/ams/AS5045-ASST?qs=jMXWnm70%252BC%2Fyg6ACAK2GtQ%3D%3D

[7] (2021, Dec.) MPU9250 [Online]. Available: https://www.amazon.com/-/ko/dp/B07QM2DYWL/ref=sr_1_1?keywords=mpu9250&qid=1639615201&sr=8-1

[8] (2021, Dec.) LDS-01 [Online]. Available: https://www.useabot.com/products/360-laser-distance-sensor-lds-01-lidar

[9] (2021, Dec.) Pi Camera [Online]. Available: https://www.amazon.com/-/ko/dp/B073RCXGQS/ref=sr_1_18?keywords=picam&qid=1639631207&sr=8-18

[10] (2021, Dec.) Intel Realsense D415 [Online]. Available: https://www.amazon.com/-/ko/dp/B07JVGRQZT/ref=sr_1_1?keywords=realsense+d415&qid=1639631292&sr=8-1

[11] (2021, Dec.) OptiTrack PrimeX13. [Online]. Available: https://www.optitrack.com/cameras/primex-13/

[12] K. Chong and L. Kleeman, Accurate odometry and error modelling for a mobile robot, in *Proceedings of International Conference on Robotics and Automation*, 1997

[13] R. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Fluids Engineering*, vol. 82, no. 1, 1960.

[14] S. Kohlbrecher, O. Von Stryk, J. Meyer, and U. Klingauf, A flexible and scalable SLAM system with full 3D motion estimation, in *9th IEEE International Symposium on Safety, Security, and Rescue Robotics*, pp. 155-160, 2011.

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

[16] M. Labbe, 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. 35, pp. 416-446, 2019.

[17] Raul Mur-Artal, J. M. M. Montiel and Juan D. Tardos, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147-1163, 2015.