The Indoor Localization of a Vision-Based Unmanned Aerial Vehicle

Jian-Xun Wu, Yuan-Pao Hsu, *Member, RST*

***Abstract*—This work implements the Real-Time Appearance Map (RTAB-MAP) algorithm on a unmanned aerial vehicle (UAV) to perform indoor localization task. The RTAB-MAP, based on a RGB-D camera, estimates the camera moving trajectory, mileage and local map according to the feature points between adjacent images to obtain globally consistent map information and camera locations. However, when using a camera for simultaneous localization and mapping (SLAM), images are prone to blurred by the fast motion of the vehicle, on which the camera is mounted, or the overlapping area of two consecutive image frames is too small, causing the feature matching failed. Therefore, this study combines an inertial measurement unit (IMU) to provide odometry data to solve the problem. In this study, when the communication between the experimental drone and ground station is well established, the ground station collects the sensory data from the drone and builds the map of the indoor environment through the RTAB-MAP method. According to the map, the system plans a path and sends waypoints to test the localization of the drone. Simulation and experimental results reveal that average trajectory errors are within ± 5 cm.**

***Index Terms*—Indoor Localization, RTAB-MA, SLAMP, UAV**

I. INTRODUCTION

IN order to make UAVs fly smoothly in indoor environments and carry out tasks, they must be equipped with sensors (such as laser range finder (LRF), infrared sensor, RGB camera, RGB-D camera, etc.) to obtain environment information (such as obstacles, walls, turns, etc.). The process, using algorithms and sensors to build the map of the environment and utilizing the map for localization at the same time, is called simultaneous localization and mapping (SLAM) [1]. If the established map is consistent with the environment, the system can use the map and features of the environment to perform further tasks.

In the early days, LRFs were used for indoor localization for UAVs [2][3]. In recent years, some scholars have begun to use monoculars, binoculars, depth cameras for visual localization [4][5][6].

Ragot *et al*. compared two common methods for visual SLAM, ORB-SLAM and RTAB-MAP. In terms of mapping and trajectory estimation, the RTAB-MAP is quite superior and it has memory management function, which is suitable for real-time mapping in long-term or large-scale environment. On the contrary, the ORB-SLAM is faster and more accurate than RTAB-MAP in feature extraction and matching [7]. Hence, in this paper, we employ the RTAB-MAP as the main SLAM algorithm and use the ORB-SLAM for feature extraction and feature matching to realize an indoor localization system on a quadrotor.

This article is organized as follows. The RTAB-MAP algorithm is described in section II. A UAV localization simulation is conducted in Section III. System architecture is in section IV. Section V depicts the UAV localization experiments. Finally, some concluding remarks and future perspectives are given in section VI.

II. THE RTAB-MAP ALGORITHM

The RTAB-MAP was originally regarded as a loop-closure detection method based on memory management, where long-term or large-scale mapping can be achieved by limiting the size of the map. After that, an open source library of RTAB-MAP has been developed and has become a complete SLAM algorithm. RTAB-MAP has the advantages of real-time mapping, stable odometer and stable positioning, and supports monocular camera, binocular camera, depth camera and laser. Fig. 1 is a block diagram of the visual based RTAB-MAP. Each block is described below.

A. Sensor Data

The main task of this step in RAB-MAP is to read and preprocess the camera image in visual SLAM. If it is a robot, it can read other sensor information, such as IMU, and synchronize the information with the camera. In addition to the visual SLAM, the laser can also be used as the main sensor. The RTAB-MAP can also use a camera and a laser at the same time.

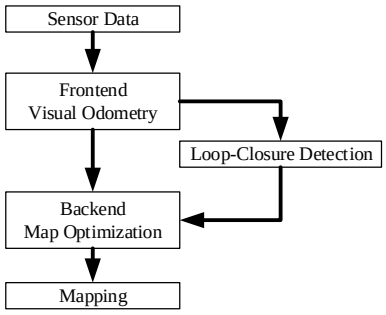

Fig. 1. RTAB-MAP block diagram

Fig. 2. RealSense R200 (RGB-D camera)

This paper was submitted on February 27, 2021.

Jian-Xun Wu was graduate student of the Department of Computer Science and Information Engineering, National Formosa University, Yunlin, Taiwan (e-mail: 10763113@gm.nfu.edu.tw).

Yuan-Pao Hsu is with the Department of Computer Science and Information Engineering, National Formosa University, Yunlin, Taiwan. (e-mail: hsuyp@nfu.edu.tw).

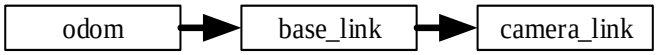

Fig. 3. Visual odometer and TF frame of robot

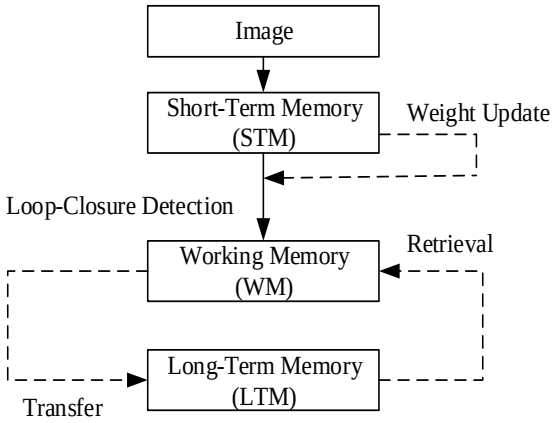

Fig. 4. Memory management block diagram

In this study, a RGB-D camera (Fig. 2) is selected as the main sensor. After camera calibration, the system reads the IMU data, and synchronizes the data with camera data to become the input message for the RTAB-MAP.

B. Visual Odometry

The purpose of the visual odometry is to estimate the motion between two consecutive images of the camera and gives a rough local map. RGB-D camera or stereo camera can be applied as image input source. Before the visual odometer works, we need to know the relative pose of the camera to the robot in order for the TF in ROS to convert coordinate frames between visual odometer and robot frame. Fig. 3 shows the TF framework after executing the visual odometer node.

For estimating the motion of the camera from two consecutive images, the RTAB-MAP extracts features from current RGB image. Then, after doing feature matching and optimization with the previous image, the camera posture is obtained by running the motion estimation algorithm. And finally, the optimized camera posture is attained

We choose ORB [9] to be the feature extraction and matching algorithm [10], and EPNP (3D-2D) to be the motion estimation method [11]. The RANSAC [12] and Bundle Adjustment [13] algorithms are for the optimization of matching points and motion estimation, respectively.

C. Loop-Closure Detection

Posture optimization is important in SLAM. In the process of mapping, when the path continues to extend, errors will inevitably be accumulated and may result in wrong posture and map. Therefore, RTAB-MAP uses the loop-closure detection method to identify the scene that it has previously reached to optimize the pose. However, after a long time of mapping, the required processing time will increase with the size of the internal map, causing the real-time mapping becomes more difficult. Therefore, RTAB-MAP adopts a memory management to limit the number of locating points during loop-closure detection. This solves the problem that it can't do mapping in a long time or a large environment.

Fig. 4 is the block diagram of the memory management. STM is the entry point when receiving a new image. All images in WM are used to participate in the operation of loop-closure detection. When the WM size exceeds a threshold, the image that is most unlikely to form a loop-closure will be transferred to LTM. If the loop-closure detection of an image is detected, WM retrieves the images related to the detected image from LTM to speed up the detection process. This makes sense. Imaging, when we see a familiar scene, it is likely we will soon see scenes nearby the familiar scene.

D. Map Optimization

Although RTAB-MAP has good performance in loop-closure detection, it still has errors. In the visual loop detection, when two postures are different but the similarity is very high or even the same, this may produce false loop-closure detection and add more errors into the whole map.

For this, in RTAB-MAP, users can choose g2o [9], gtsam [10] or Toro [11] to detect the error loop-closure detection. When loop-closure detection or some localization points are transferred to LTM, a graphic optimization will be carried out. According to [12], the above three algorithms have been tested in an environment with eight corridors. The results show that there is no obvious overlap in the optimization results by using g2o. Therefore, we choose g2o as the algorithm for map optimization in this study.

III. THE UAV LOCALIZATION SIMULATION

The quadrotor indoor localization is simulated in the simulation environment of the gazebo [13] in Ubuntu operating system. Using robot localization node, which combines visual odometer and IMU data, and adjusting TF, a stable SLAM can be achieved. Finally, the UAV is controlled to complete the indoor localization.

A. Indoor Localization Simulation

The purpose of this simulation is to control a UAV to fly accurately according to the planned waypoints in a known indoor environment. This article carries out two flight modes, vertical takeoff-landing and square path flight. During each flight, the error between the trajectory estimated by RTAB-MAP and the real trajectory are calculated per second.

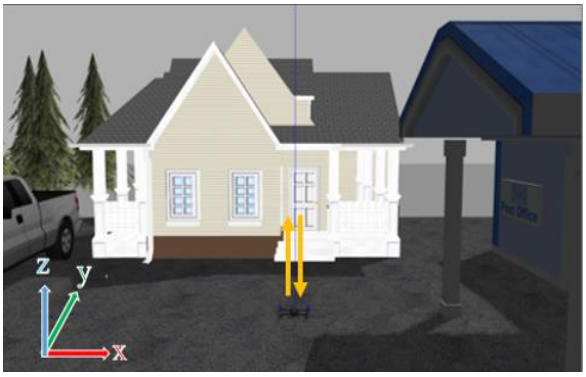

Fig. 5. Simulation environment for vertical takeoff-landing

TABLE I
ERRORS OF TAKEOFF-LANDING SIMULSTION

Error of trajectory		
Trial \ Error	Average(m)	Max(m)
1st	0.017	0.079
2nd	0.014	0.054
3rd	0.019	0.137
4th	0.015	0.111
5th	0.012	0.035
Average	0.015	0.083
Error between return point and starting point		
Trial \ Error	x(m)	y(m)
1st	0.035	-0.008
2nd	-0.043	-0.005
3rd	0.019	-0.040
4th	0.047	-0.014
5th	-0.024	-0.002
Average	0.007	-0.014

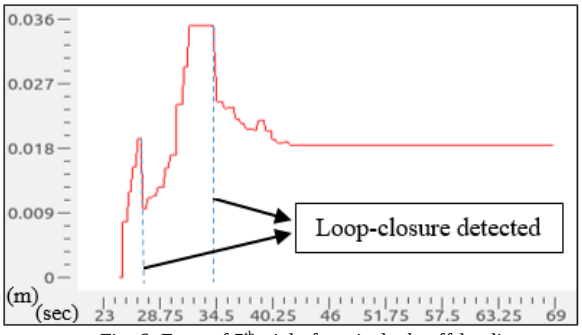

Fig. 6. Error of 5th trial of vertical takeoff-landing

1) Simulation of Vertical Takeoff-Landing

In the simulation environment in Fig. 5, the drone flies vertically to altitude of 2 meters and descends along the same path. The yellow arrow is the flight path. The results are shown in TABLE I. The average error, the maximum trajectory displacement and errors between starting point and return point are calculated respectively. Fig. 6 shows the trajectory error of the 5th trial of vertical takeoff-landing. The x-axis is in second and the y-axis is in meter. The loop-closure detection occurs at 25 seconds and 34.5 seconds causing the errors rapidly decreased.

2) Simulation of Square Path Flight

In Fig. 7, the UAV flies around the tree along a square path. The yellow arrow is the flight path. The errors are shown in Table 2.

In TABLE II, it has about 130 position errors being calculated, in which each error is the displacement between the RTAB-MAP visual estimation and the real trajectory during flying along the square path.

Although the maximum displacement error is about 46 cm, thanks to the correction of loop-closure detection, the average displacement error is only about 4.9 cm, and the average errors of x and y between the return point and starting point are -1.1cm and -2.1cm, respetively.

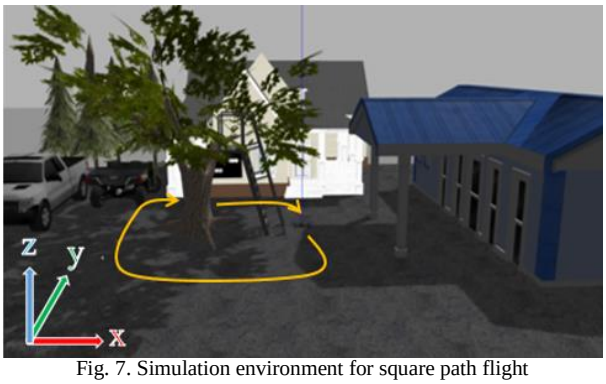

Fig. 7. Simulation environment for square path flight

TABLE II
ERROR OF SQUARE PATH SIMULATION

Error of trajectory		
Trial \ Error	Average(m)	Max(m)
1st	0.053	0.467
2nd	0.046	0.291
3rd	0.053	0.428
4th	0.048	0.394
5th	0.043	0.267
Average	0.049	0.369
Error between return point and starting point		
Trial \ Error	x(m)	y(m)
1st	0.031	-0.044
2nd	-0.140	0.036
3rd	-0.004	-0.004
4th	0.045	-0.041
5th	0.014	-0.050
Average	-0.011	-0.021

IV. SYSTEM ARCHITECTURE

Fig. 8 shows the architecture of the experimental system. The system has a computer with Ubuntu operating system, a UAV, a ground station and a router. All of them need to be connected under the same network domain. The computer controls the UAV through TCP/IP [14] communication protocol and secure shell(SSH) [15] transmission protocol for sharing topics. Fig. 9 is the block diagram of steps for the UAV to perform indoor localization experiments.

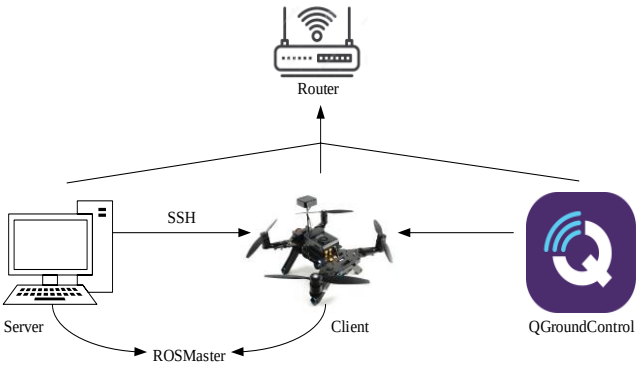

Fig. 8. Environment architecture

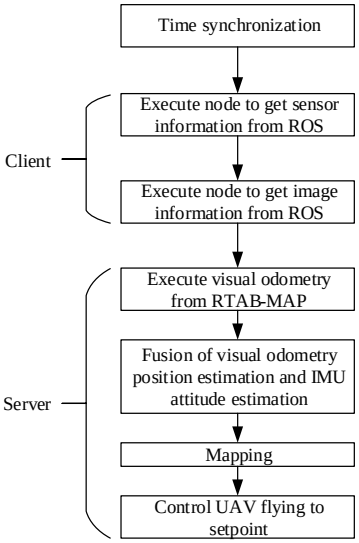

Fig. 9. Indoor positioning block diagram of UAV

TABLE III
ERROR OF TAKEOFF-LANDING EXPERIMENTS

Error of trajectory		
Trial \ Error	Average(m)	max(m)
1st	0.053	0.467
2nd	0.046	0.291
3rd	0.053	0.428
4th	0.048	0.394
5th	0.043	0.267
Average	0.049	0.369
Estimated error between return point and starting point		
Trial \ Error	x(m)	y(m)
1st	0.031	-0.044
2nd	-0.140	0.036
3rd	-0.004	-0.004
4th	0.045	-0.041
5th	0.014	-0.050
Average	-0.011	-0.021

TABLE IV
ERROR OF TAKEOFF-LANDING BETWEEN THE RETURN POINT AND THE STARTING POINT

Actual error between return point and starting point		
Trial \ Error	x(m)	y(m)
1st	0.034	0.010
2nd	0.019	0.004
3rd	0.051	0.002
4th	0.020	0.008
5th	0.039	0.018
Average	0.032	0.008

V. THE UAV LOCALIZATION EXPERIMENTS

This section presents the experimental results of indoor localization for the quadrotor. The two flight modes as the simulation are performed in the experiments. Because GPS can't be used as the reference of real trajectory indoors, and no other sensors can accurately calculate the posture of the UAV. Therefore, in the ROS, setpoints are taken as the real trajectory in our experiments.

We use the Intel aero quadrotor for the experiments [21]. The accelerometer, gyroscope, compass and barometer sensors are calibrated in the ground station, and the PID control parameters are also adjusted in advance.

A. Experimental Results of Takeoff-Landing

In this experiment, the flight altitude is 40 cm. As shown in TABLE III, the average error, the maximum displacement and errors of return point to the starting point are calculated respectively. TABLE IV shows the actual errors between the return point and the starting point.

B. Experimental Results of Square Path

The experimental results in TABLE V and VI show that the actual position (x, y) error (TABLE VI) and estimated value (x, y) error (the second half of TABLE V) of return point to starting point are all not more than 7 cm. And the average error between estimated postures and setpoints of each waypoint is within 4.2 cm. These prove that the system can accurately complete the indoor positioning task.

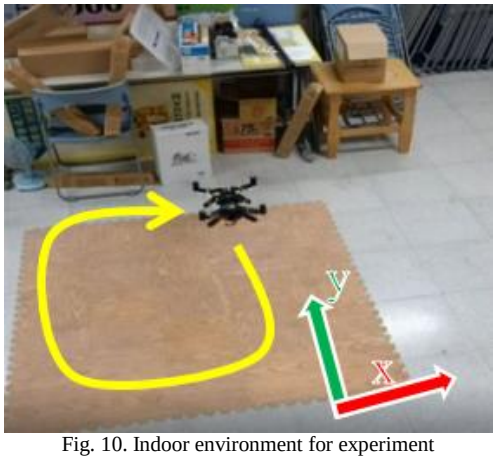

Fig. 10. Indoor environment for experiment

TABLE V
ERROR OF SQUARE PATH EXPERIMENTS

Error of trajectory		
Trial \ Error	Average(m)	max(m)
1st	0.039	0.111
2nd	0.043	0.104
3rd	0.038	0.108
4th	0.047	0.119
5th	0.041	0.097
Average	0.042	0.108
Estimated error between return point and starting point		
Trial \ Error	x(m)	y(m)
1st	-0.076	-0.040
2nd	-0.051	-0.024
3rd	-0.070	-0.046
4th	-0.067	-0.037
5th	-0.062	-0.031
Average	-0.065	-0.036

TABLE VI
ERROR OF SQUARE PATH FLIGHT BETWEEN THE RETURN POINT AND THE STARTING POINT

Actual error between return point and starting point		
Trial \ Error	x(m)	y(m)
1st	0.071	0.035
2nd	0.066	0.041
3rd	0.052	0.032
4th	0.070	0.039
5th	0.064	0.021
Average	0.065	0.034

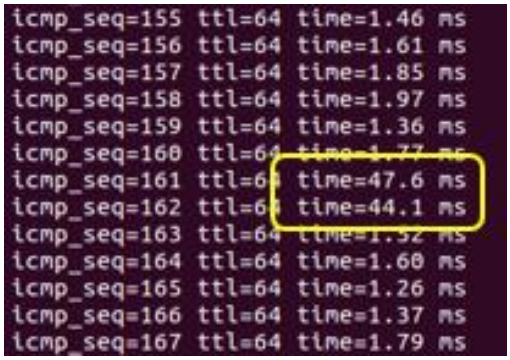

Fig. 11. Diagram of network jitter

VI. CONCLUSION AND FUTURE PERSPECTIVE

A. Discussion and Conclusion

The system connects the drone and the server under the same domain through Wi-Fi. Due to the network jitter, as shown in Fig. 11, it causes high delays at certain times randomly. Therefore, if the drone encounters network jitter during flight, it will not be able to communicate with the server and lost its current location information. In severe cases, the jitter may last for 2-3 seconds, during this period of time the drone could fly along any direction. Until the end of the jitter, the UAV recognizes the current position again through the map and returns to the setting waypoint. This makes the drone could not fly stably for a long period of time.

The system calculates the errors between the RTAB-MAP estimation trajectory and the real trajectory per second in the simulation. However, the real trajectory could not be obtained in our experiments. Therefore, in experiments the errors are the differences between the estimated postures and the waypoints. Anyhow, the simulation and experimental results show that the average errors of trajectory are less than 5 cm. In terms of distance, the purpose is to verify the indoor localization effect of a UAV when using visual-based RTAB-MAP. So we restrict the drone flying only along a 50 cm x 50 cm square path in the experiment.

B. Contribution of this Study

RTAB-MAP is a very mature method. Most related papers have used mobile robots with camera or laser to go with RTAB-MAP in indoor localization. There are few or even no related papers choosing UAVs as targets to run RTAB-MAP in indoor localization. Therefore, this paper implements an indoor localization system on a quadrotor, using RTAB-MAP algorithm, combined with RGB-D camera and IMU sensor, for self-localization. Simulation and experimental results show that the integrated system can accurately perform indoor localization task.

C. Future Work

The integrated indoor positioning system in this study can be improved in the following aspects in the future.

1) In this paper, due to the network jitter, all actions will be stopped for a certain period of time randomly, which deeply affects the mapping. In the future, we should choose a router with better performance to reduce the delay and network jitter. So the UAV and server can communicate with each other stably.
2) We choose the Intel aero UAV, whose performance (such as CPU and GPU) is not good enough to perform SLAM tasks. So the images have to be sent to the server through wireless network for image processing. But this will encounter the problem of 1). Therefore, UAVs with better equipment should be selected in the future. Such that the server only needs to send commands to control the well-equipped UAV to get rid of the network jitter problem.
3) The UAV performs only translational flight in this work. In the future, rotational flight should be added to make the system more flexible.
4) We use only RGB images to complete the indoor positioning. In the future, we should add depth information to the system. So the system has more functions like obstacle avoidance and path planning to carry out the complete indoor navigation task.

REFERENCES

[1] D. Whyte and T. Bailey, “Simultaneous Localization and Mapping (SLAM): Part I The Essential Algorithms,” *IEEE Robotics & Automation Magazine*, Vol. 13, Iss. 2, Jun. 2006.
[2] A. Bachrach, A. d. Winter, R. He, G. Hemann, S. Prentice and N. Roy, “RANGE - Robust Autonomous Navigation in GPS-Denied Environments,” *2010 IEEE International Conference on Robotics and Automation*, May 2010.
[3] G. Chowdhary, D. M. Sobers, C. Pravitra, H. C. Christmann and A. Wu, “Self-Contained Autonomous Indoor Flight with Ranging Sensor Navigation,” *AIAA J. Guid. Control Dyn*, Vol. 35, No. 6, Oct. 2012.
[4] M. Achtelik, M. Achtelik, S. Weiss and R. Siegwart, "Onboard IMU and Monocular Vision Based Control for MAVs in Unknown In- and Outdoor Environments," *2011 IEEE International Conference on Robotics and Automation*, May 2011.
[5] S. J. Haddadi and E. B. Castelan, "Visual-Inertial Fusion for Indoor Autonomous Navigation of a Quadrotor Using ORB-SLAM," *2018 Latin American Robotic Symposium, 2018 Brazilian Symposium on Robotics (SBR) and 2018 Workshop on Robotics in Education (WRE)*, Nov. 2018.
[6] W. G. Aguilar, G. A. Rodríguez, L. Á. and S. Sandoval, "Visual SLAM with a RGB-D Camera on a Quadrotor UAV Using on-Board Processing," *International Work-Conference on Artificial Neural Networks*, May 2017.
[7] N. Ragot, R. Khemmar, A. Pokala, R. Rossi and J. Y. Ertaud, "Benchmark of Visual SLAM Algorithms: ORB-SLAM2 vs RTAB-Map*," *2019 Eighth International Conference on Emerging Security Technologies* (EST), Jul. 2019.
[8] RealSense R200，https://reurl.cc/R60ND6
[9] ORB，https://reurl.cc/7ovzY1
[10] Ebrahim, S. Prasad and M. shehata, “Image Matching Using SIFT, SURF, BRIEF and ORB: Performance Comparison for Distorted Images,” *2015 Newfoundland Electrical and Computer Engineering Conference*, Nov. 2015.
[11] PNP algorithm，https://reurl.cc/od2Gr3
[12] RANSAC，https://reurl.cc/j5Zp71
[13] Bundle adjustment，https://reurl.cc/3LE87L
[14] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige and W. Burgard, "G2o: A general framework for graph optimization," *2011 IEEE International Conference on Robotics and Automation*, May 2011.
[15] F. Dellaert, "Factor graphs and GTSAM: A hands-on introduction," Tech. Rep. GT-RIM-CP&R-2012-002, Georgia Institute of Technology, Sep. 2012.
[16] G. Grisetti, S. Grzonka, C. Stachniss, P. Pfaff and W. Burgard, "Efficient estimation of accurate maximum likelihood maps in 3d," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov. 2007.
[17] J. M. M Ferrão, "Study of SLAM algorithms for autonomous navigation in unstructured environments," Mar. 2018. [Online] Available: https://ria.ua.pt/handle/10773/25126?locale=en
[18] Gazebo，http://wiki.ros.org/gazebo
[19] TCP/IP，https://reurl.cc/WL6R85

[20] SSH，https://reurl.cc/Z7K92V
[21] Intel Aero Ready to Fly Drone，https://reurl.cc/e9mbgx

Jian-Xun Wu received the B.S. degree in the Department of Computer Science and Information Engineering from National Formosa University, Yunlin, Taiwan, R.O.C., in 2018. He received the M.S. degree in the Department of Computer Science and Information Engineering from National Formosa University, Yunlin, Taiwan, R.O.C., in 2021. His major research interests include UAV applications and Robot Operating System.

Yuan-Pao Hsu received the B.S. and M.S. degrees in the Department of Electronic Engineering from National Taiwan University of Science and Technology, Taipei, Taiwan, R.O.C., in 1987 and 1989. He received the Ph.D. degree in Electrical Engineering department of National Chung Cheng University, Chiayi, Taiwan, R.O.C., in 2004. His research areas are mobile robot applications, applications of SW/HW co-design, machine learning, image processing, and embedded systems.