ORB-SLAM, IMU and Wheel Odometry Fusion for Indoor Mobile Robot Localization and Navigation

Jian Lin[1], Jierui Peng[2], Zhichao Hu[3], Xiaofeng Xie[4] and Rui Peng[*]

[1] School of Electronic Engineering, University of Electronic Science and Technology of China, Chengdu, China
[2] Brandeis University, Computer Science Department, Waltham, USA
[3] Research and Education Center High IT School, National Research Tomsk State University, Russia
[4] College of Information Engineering, Wuchang Institute of Technology, Wuhan, China
[*] Department of Mechanical Engineering, The Hong Kong Polytechnic University, Hong Kong

ABSTRACT. *In this paper, we propose a light-weight multi-sensor fusion method containing ORB-SLAM, IMU and wheel odometry for localization and navigation of an indoor mobile robot in GPS-denied environment. Known as an accepted generally visual simultaneous localization and mapping (SLAM) system, ORB-SLAM based on feature matching computes real-time camera pose. The Inertial Measurement Unit (IMU) measures the angular velocity of the robot by one of its gyroscopes. The wheel odometry provides linear motion velocity for the robot and records distance the robot has moved. Through leveraging both rotation characteristic of IMU and linear characteristic of wheel odometry, the rough localization estimation for the robot is obtained. During every navigation of the robot, the rough localization estimation provides relatively accurate mapping scale of the real world for ORB-SLAM. And the mapping scale revises the monocular camera pose of ORB-SLAM to obtain global robot pose estimation in the real world. In the experiment, the robot can locate itself with tolerable error and perform great navigation ability in a specific scene.*

KEYWORDS: *mobile robot, ORB-SLAM, multi-sensor fusion, indoor localization*

1. Introduction

Due to the low cost and rich texture information of the camera, many researchers prefer to explore vision-based method for indoor localization, especially monocular visual localization.

ISSN 2616-5775 Vol. 3, Issue 1: 131-141, DOI: 10.25236/AJCIS.2020.030114

Raúl Mur-Artal proposed the ORB-SLAM algorithm [1], and it taking advantages of the ORB features runs well in real time even without GPU. The system can work in the indoor environment, and also is robust to violent movement. Some of ORB-SLAM based works are applied for the mobile robot. A localization and navigation method with ORB-SLAM is presented in [2]. An improved robot's localization method based on ORB-SLAM is presented in [3]. Even the ORB-SLAM can be employed on an aerial robot for localization [4]. A quadrotor realizes autonomous flight by monocular SLAM [5].

However, depending on an ORB-SLAM system alone is not perfectly able to deal with mobile robot localization in the complex conditions like light changing, less texture and high-speed motion. Multi-sensor fusion is an effectively robust approach to handle these intractable problems. VINS-Mono [6] fuses monocular visual SLAM and IMU (Inertial Measurement Unit) to perform an excellent state estimator. More and more sensor fusion based localization methods have been proposed. A kind of indoor positioning approach using visual and inertial sensors is presented in [7]. A mobile robotic platform with real-time localization via fusion of inertial and visual data is presented in [8].

Furthermore, employed to compute wheel odometry, the wheel encoder as a unique sensor of the mobile robot is added into sensor fusion for the indoor robot localization and navigation. A method fusing visual odometry, IMU and wheel encoders is presented by [9]. Based on the work in [9], we design an indoor mobile robot system with ORB-SLAM, IMU and wheel odometry fusion, a light-weight method proposed. This robot system can locate itself with the tolerate error and autonomously navigate on the basis of a grip map with the motion planning and control algorithm. In order to perform the fusion method well, We establish a specific scene where this robot system can show the outstanding localization ability with low complexity. The whole software system of the robot runs in the ROS open-source platform. The demo video is online: https://youtu.be/56y17I5wLtI.

2. Methodology

2.1 IMU and wheel odometry

To show good performance of the fusion method, we take advantages of all of the robot sensors plenarily and design an integrated software system. Fig. 1 shows the pipeline of the whole software system.

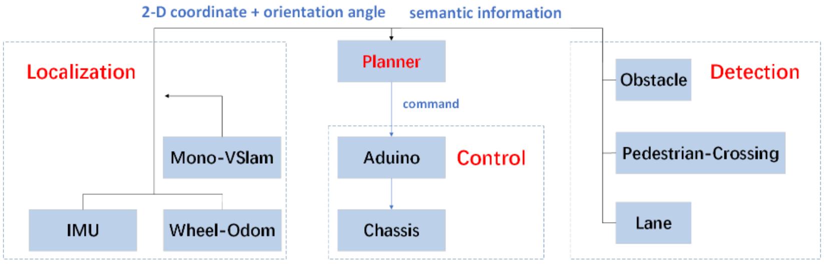

Figure. 1 Pipeline of the robot software system

To describe the fusion method numerically, O_r is defined as the robot body frame. In experimental scene, ORB-SLAM, IMU and wheel odometry are fused to obtain robot localization with respect to the world frame, here called O_w.

The robot with two driving wheels abides by the kinematic unicycle model [12]. It moves in the horizontal plane spanning the X-axis and Y-axis of O_r and O_w. It rotates about the Z-axis of O_r with the angular velocity ω_{robot} and moves forward or backward along the Y-axis with the linear velocity v_{robot} with respect to O_r. It has only two degrees of freedom according to its mechanical structure. The location of the robot in O_w is the origin of $O_r(t)$, given by $[x_t, y_t, \theta_t]^T$. The orientation of the robot is θ_t measuring by the angle between the Y-axis of O_r and X-axis of O_w.

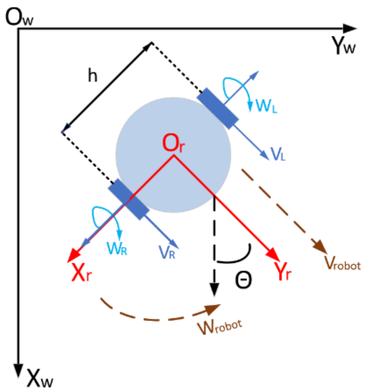

Figure. 2 Kinematic unicycle model of robot

The kinematic model of the robot is shown in Fig. 2. Wheel encoders can measure the angular velocity of the corresponding wheels and are used to compute wheel odometry by providing motion information of the mobile robot. And the wheel odometry is an indispensable component of the mobile robot localization. The angular velocities of the left wheel and right wheel are given by ω_l and ω_r

ISSN 2616-5775 Vol. 3, Issue 1: 131-141, DOI: 10.25236/AJCIS.2020.030114

respectively, as shown in Fig. 2. These measurements are converted into the linear velocity of the wheels by:

$$v_l = \omega_l * r \tag{1}$$

$$v_r = \omega_r * r \tag{2}$$

Where $r \in R^+$ is the constant radius of the wheels. v_l is linear velocity of the left wheel and v_r is linear velocity of the right wheel. By measurement, the constant distance between two wheels is given by $h \in R^+$. The linear and rotational velocity of the robot are given by:

$$v_{robot} = (v_l + v_r)/2 \tag{3}$$

$$\omega_{robot} = (v_l - v_r)/2 \tag{4}$$

The rough location estimation of the robot in O_w is obtained, and the formula is as follows:

$$[x_t \quad y_t \quad \theta_t] = \sum [-\sin\theta_k \Delta t \quad \cos\theta_t \Delta t \quad \Delta t] \tag{5}$$

Where $\Delta t \in R^+$ is a very short interval between two time stamps. The IMU measures the linear acceleration through its accelerometers and the angular velocities through its gyroscopes. On account of the restricted motion of the robot, only one of the gyroscopes is leveraged to measure the angular velocity ω_{robot}. The robot rotation angle within the Δt is given by:

$$\theta_k = \sum \omega_{robot} * \Delta t \tag{6}$$

The selection of Δt affects accuracy of the rough location estimation directly. Empirically, the shorter Δt is, the more accurate robot location estimation obtained by IMU and wheel odometry is. However, no matter how short Δt is, long-term error accumulation is unavoidable. Thus the robot system needs other methods to efficaciously reduce location error.

2.2 Monocular Visual SLAM

ORB-SLAM, a powerful and integral visual SLAM system is a crucial component of our software system based on the ROS platform. It mainly consists of tracking, mapping, relocation, and loop closure. Its estimation process is based on feature matching. In this sense, its name indicates that the visual feature to be leveraged is a binary descriptor called ORB descriptor [10]. The ORB descriptor robust to rotation changes effectively estimates the direction of key points according to the intensity centroid of the image patch. Also, the visual odometry of ORB-SLAM is obtained by the accumulation of camera pose.

In our system, a monocular camera is applied for ORB-SLAM, instead of RGB-D camera [3], as illustrated in Fig. 3. Tracking thread initializes camera pose by global relocalization and is leveraged to estimate the camera pose according to the

ISSN 2616-5775 Vol. 3, Issue 1: 131-141, DOI: 10.25236/AJCIS.2020.030114

last frame after extracting ORB feature from the current frame. Local mapping thread establishes local maps, including the insertion of the key frame and generating the new map points. Loop closing thread contains two components: loop detection and loop correction. In loop detection, the bag-of-words model is leveraged to accelerate screening out the matching frame. Loop correction consists of loop fusion and graph optimization based on the essential graph.

2.3 Fusion Approach

In the experiment, ORB-SLAM can computes robust pose for once robot navigation. But the mapping scale from monocular camera pose to the world frame is different after every initialization of ORB-SLAM, due to the feature matching affected by the environmental conditions easily. It is impractical to set different mapping scales for many times of robot navigation experiments. However, IMU and wheel odometry can provide the relatively constant scale of the robot motion in the world frame O_w.

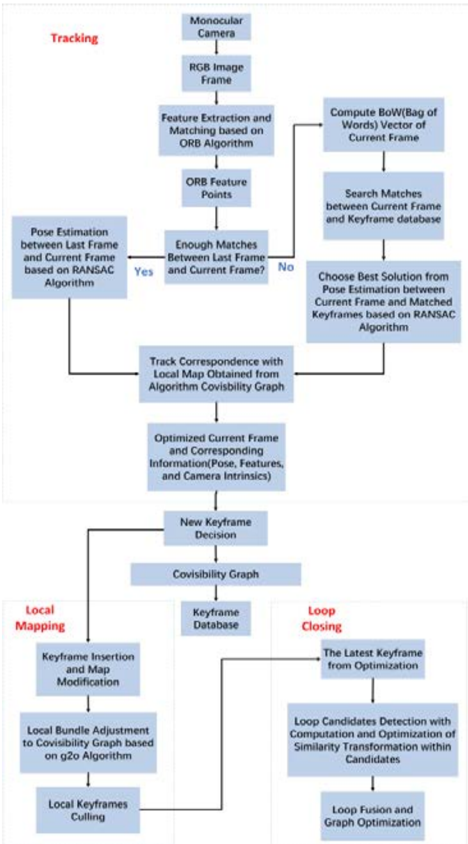

Figure. 3 ORB-SLAM system framework.

ISSN 2616-5775 Vol. 3, Issue 1: 131-141, DOI: 10.25236/AJCIS.2020.030114

Thus, the rough estimation obtained by IMU and wheel odometry is leveraged to offer scale compensation to the visual odometry of the ORB-SLAM. According to subsection II-A, the robot location change of the rough estimation is considered accurate in a very short interval, defined as $\Delta\delta$. $\Delta\delta$ is the scale computing interval.

Then $10\Delta\delta$ is defined as the fixed moving interval, here called Δk. The relation between $\Delta\delta$ and Δk is shown in Fig. 4.

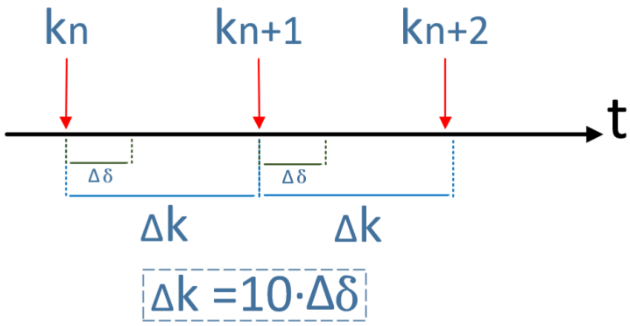

Figure. 4 Time-Axis of Robot Navigation

In Fig. 4, k_n, k_{n+1}, and k_{n+2} are arbitrary and consecutive time stamps of the robot navigation time axis. We take the Δk between k_n and k_{n+1} as an example. From the start of the k_n, the location of the rough estimation is given by (ω_x^a, ω_y^a) corresponding to k_n, and the location of the camera pose is given by (c_x^a, c_y^a) corresponding to k_n. After the $\Delta\delta$, the location of the rough estimation is given by (ω_x^b, ω_y^b) corresponding to $k_n+\Delta\delta$, and the location of the camera pose is given by (c_x^b, c_y^b) corresponding to $k_n+\Delta\delta$. Then the location change of the rough estimation within the $\Delta\delta$ is given by:

$$\Delta\omega_x = \omega_x^b - \omega_x^a \tag{7}$$

$$\Delta\omega_y = \omega_y^b - \omega_y^a \tag{8}$$

And the location change of the camera pose within the $\Delta\delta$ is given by:

$$\Delta c_x = c_x^b - c_x^a \tag{9}$$

$$\Delta c_y = c_y^b - c_y^a \tag{10}$$

The mapping scale from the camera pose to the world frame within this Δk is given by:

$$m_x = \Delta\omega_x / \Delta c_x \tag{11}$$

$$m_y = \Delta\omega_y / \Delta c_y \tag{12}$$

ISSN 2616-5775 Vol. 3, Issue 1: 131-141, DOI: 10.25236/AJCIS.2020.030114

During the whole navigation of the robot, the total time is divided into plenty of Δk. Within every Δk, both m_x and m_y are computed by equations above to map the camera pose to the world frame O_w to obtain a robot pose belonging to this Δk. Ultimately, so many of these short poses are integrated as the final estimated robot motion trajectory of navigation under the world frame O_w.

2.4 Motion Planning and Navigation

The robot system in this paper is based on a known indoor map to navigate autonomously. The map is designed as a grid map. The motion planning route consisting of many joint lines is determined before the navigation. During the navigation of the robot, we set the current robot pose and the planning route as the input of the P-D control algorithm. The controller is given by:

$$u(n) = k_p E(n) + k_d e(n) \tag{13}$$

Where k_p, k_d are adjustable P-D controller coefficients. The E(n) shown in Fig. 5 is the error between the robot current orientation θ_r^n and target orientation, given by:

$$E(n) = \arctan((y_n - y_{n-1})/(x_n - x_{n-1})) - \theta_r^n \tag{14}$$

In consideration of the kinematic model, the output of the controller is employed to control the robot moving along the planning route, leading to autonomous robot navigation.

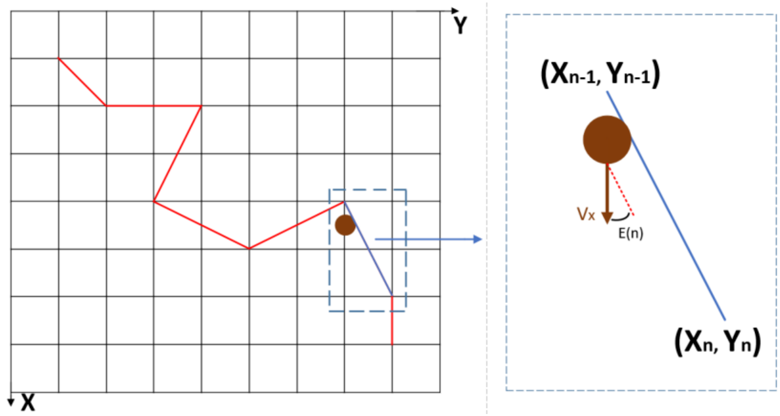

Figure. 5 The robot navigates in the grid map

3. Experimental Results

The robot which we use is modified from TurtleBot3-Burger, which is a low-cost, flexible individual robot kit with a laser sensor and open-source software. The

ISSN 2616-5775 Vol. 3, Issue 1: 131-141, DOI: 10.25236/AJCIS.2020.030114

modification is that we change the laser sensor as two cameras, one of which is for monocular ORB-SLAM. The robot hardware platform is shown in Fig. 6.

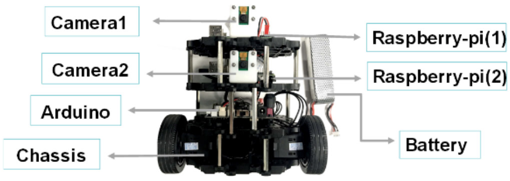

Figure. 6 Robot hardware platform. (1) The robot base is a two-wheel chassis with three layers of separated space vertically. The wheel is mounted with an encoder measuring its angular velocity precisely. (2) The controller unit is Arduino MEGA2560-R3 with an IMU (Inertial measurement unit) in the bottom layer of the robot. (3) In the top and the middle layer of the robot is a Raspberry-pi 3 Model B+ with an RPi camera-B

To validate the capability of our indoor multi-sensor fusion method without lidar, we design a simulated scene for our robot system. The experimental scene shows as Fig. 7. Besides, the robot needs to complete many times of autonomous navigation tests in this scene to show great robustness of the fusion method. The top view is shown in Fig. 8.

Figure. 7 The experimental scene

ISSN 2616-5775 Vol. 3, Issue 1: 131-141, DOI: 10.25236/AJCIS.2020.030114

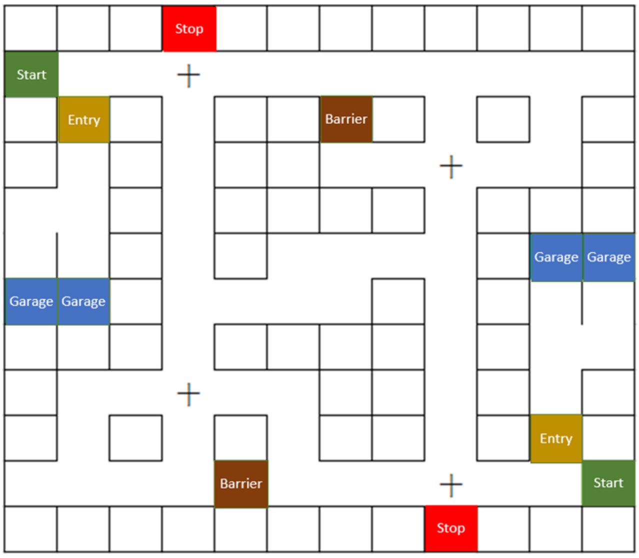

Figure. 8 Top view of experimental scene

During the robot navigation, we record both location data of the rough estimation and the final robot pose estimation with the robot passing every 36cm in the experiment field.

Then these all discrete points are connected to generate two trajectories, respectively. By comparing two trajectories to the planning route, the performance of our multi-sensor fusion method for robot localization and navigation is explicitly demonstrated. Fig. 9 shows the results of robot navigation in two different planning routes.

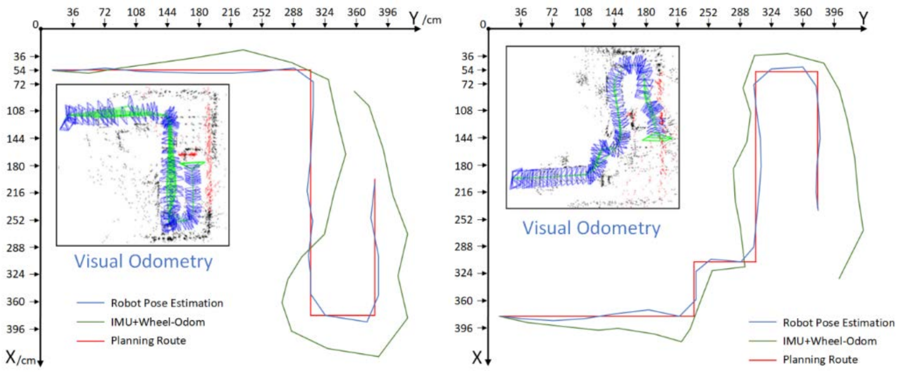

Figure. 9 Robot localization and navigation results

ISSN 2616-5775 Vol. 3, Issue 1: 131-141, DOI: 10.25236/AJCIS.2020.030114

4. Conclusion

In this paper, we propose a light-weight multi-sensor fusion method containing ORB-SLAM, IMU and wheel odometry for localization and navigation of an indoor mobile robot. Then we perform the fusion method on a low-cost robot platform with low complexity and the robot can locate itself accurately and autonomously navigate in indoor environment. Simulated self-driving experiments demonstrate great performance of the robot system with the fusion method. Besides, it can be conveniently expanded for specific indoor applications.

References

[1] Ral Mur-Artal, Jose Maria Martinez Montiel, Juan D. Tardos. ORB-SLAM: a versatile and accurate monocular SLAM system, IEEE Transactions on Robotics, vol. 31, no. 5, pp. 1147-1163, 2015.

[2] Hailan Kuang, Xiaodan Wang, Xinhua Liu, Xiaolin Ma, Ruifang Li, An Improved Robots Localization and Mapping Method Based on ORB-SLAM," in 2017 10th International Symposium on Computational Intelligence and Design (ISCID), vol. 2, pp. 400-403, 2017.

[3] Shirong Wang, Yuan Li, Yue Sun, Xiaobin Li, Ning Sun, Xuebo Zhang, Ningbo Yu, A localization and navigation method with ORB-SLAM for indoor service mobile robots," in 2016 IEEE International Conference on Real-time Computing and Robotics (RCAR), pp. 443-447, 2016.

[4] Garca, Walterio Mayol-Cuevas, Towards autonomous flight of micro aerial vehicles using ORB-SLAM," in 2015 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS), pp. 241-248, 2015.

[5] Omid Esrafilian, Hamid D. Taghirad, Autonomous flight and obstacle avoidance of a quadrotor by monocular SLAM," in 2016 4th International Conference on Robotics and Mechatronics (ICROM), pp. 240-245, 2016.

[6] Tong Qin, Peiliang Li, Shaojie Shen, VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator," IEEE Transactions on Robotics, 34 (4), pp. 1004-1020, 2018.

[7] Wen Fu, Ao Peng, Biyu Tang, Lingxiang Zheng, Inertial sensor aided visual indoor positioning," in 2018 International Conference on Electronics Technology (ICET), pp. 106-110, 2018.M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 198.

[8] Seyed Jamalaldin Haddadi, Eugenio B. Castelan, Visual-Inertial Fusion for Indoor Autonomous Navigation of a Quadrotor Using ORB-SLAM," in 2018 Latin American Robotic Symposium, 2018 Brazilian Symposium on Robotics (SBR) and 2018 Workshop on Robotics in Education (WRE), pp.106-111, 2018.Kass, R. E. and A. E. Raftery (1995). Bayes Factors. Journal of the American Statistical Association 90, 773–794.

[9] Jinglin Shen, David Tick, Nicholas Gans, Localization through fusion of discrete and continuous epipolar geometry with wheel and IMU odometry," in Proceedings of the 2011 American Control Conference, pp. 1292-1298, 2011.

ISSN 2616-5775 Vol. 3, Issue 1: 131-141, DOI: 10.25236/AJCIS.2020.030114

[10] Ethan Rublee, Vincent Rabaud, Kurt Konolige, Gary Bradski, Orb: An efficient alternative to sift or surf, in International Conference on Computer Vision, pp. 2564-2571, 2011.
[11] Joseph Redmon, Ali Farhadi, YOLO9000: Better, Faster, Stronger," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6517-6525, July 2017.
[12] Jean-Paul Laumond, Jean-Jacques Risler, Nonholonomic systems: controllability and complexity, Theoretical Computer Science, vol. 157, pp. 101114, 1996.