Localization and Navigation Analysis of Mobile Robot Based on SLAM

Juan An, Hairong Mou, Rui Lu and Youbing Li

Guangzhou City Construction College, Guangzhou, China, 510925
Corresponding author e-mail: 525831524@qq.com

Abstract. In recent years, many advanced computer technologies have appeared on the market, for example, artificial intelligence. With the maturity of artificial intelligence technology, the transformation of intelligent robots are also more perfect. The difficulty that we have always wanted to break through is to install positioning navigation on mobile robots. Through continuous efforts, the navigation and positioning technology on the robot can now be used proficiently in a known environment, but we need further research when using it in an unknown environment. In the navigation and positioning technology, it introduced simultaneous positioning and map construction algorithms (SLAM, simultaneous localization and mapping). The algorithm can learn about the surrounding environment information through the sensors carried by the robot. If this algorithm can be skillfully applied to navigation and positioning technology, I believe it will be helpful for the application of positioning and navigation technology in unknown environments.

1. Introduction
This paper mainly introduces the color image and depth image provided by the RGB-D sensor, which is based on ORB-SLAM. It expands the sparse image of the original system sealing structure into a dense point cloud image. At the same time, we recommend using an octree structure to construct eight robots. Finally, the algorithm was successfully applied to autonomous mobile robots. Experimental results show that the algorithm can achieve autonomous obstacle avoidance under challenging conditions and long-distance trajectories.

2. Summarize
In a real environment, autonomously moving robots must have the ability to navigate in large, unstructured, dynamic and unknown spaces. Therefore, mobile robots must be able to construct environmental maps and realize self-positioning, that is, simultaneous positioning and map construction. SLAM has received extensive attention and research in the past 30 years. Scholars have proposed many SLAM methods, including various sensors, optimization techniques and map description methods. For this reason, Lidar was developed, which can provide accurate environmental geometric measurement at high frequencies, and its main disadvantages are high price, high energy consumption, large size and weight [1].

The visual SLAM method uses the camera as a sensor to estimate the robot's pose and environment map. Klien proposed to divide tracking and map construction into two parallel threads in the monocular VSLAM system, and realized a parallel tracking and map construction system based on package adjustment. RGB-D cameras can provide color images and depth images. Compared with monocular cameras, they can obtain absolute proportions, they avoid the problems of depth calculation and scale.
blur. Compared with binocular cameras, they can directly obtain depth information, so the amount of calculation is small, it reduces the difficulty of implementation.

Kerl proposed to optimize the optical measurement consistency error term to register continuous RGB-D frames to construct the RGB-D odometer, and then proposed a robust method combining the optical measurement "uniformity error and the error term based on dense depth data". Estimating and ignoring map construction Henry et al. proposed that the SLAM system uses an RGB-D camera combined with an iterative closest point algorithm and visual features to calculate and optimize the camera pose, thereby constructing a dense point cloud map, but not in real time.

The RGB-D SLAM algorithm proposed by Et et al., it runs on the CPU like the Kerl algorithm, and it uses the same hardware for evaluation experiments, which has better robustness and accuracy. These algorithms use random sampling to estimate related visual features. The closed loop between points, and the use of nonlinear methods to optimize the posture map and the final generated three-dimensional voxel map of the environment can be applied to robot obstacle avoidance, path planning and navigation [2]. Because this algorithm uses ICP for pose graph optimization, it relies too much on a good initial estimate and lacks a measure of overall matching quality.

Since visual aids are usually located on the edge of the object, the depth measurement data is susceptible to noise interference. In order to reduce the estimation error, ElasticFusion proposed an RGB-DSLAM algorithm, which is based on plane features. The fixed-point function helps to improve the accuracy and robustness of traditional ICP. At the same time, it can maintain a reasonable amount of calculation to ensure real-time performance. ElasticFusion performs visual ranging by combining ICP with a large number of measured reprojection errors to create a point cloud image. The point cloud image is very accurate, but it requires CPU acceleration. Point cloud imaging consumes a lot of resources, and it is not conducive to maintenance. This question does not apply to robot applications. Mur-Arta I and others proposed ORB-SLAM. It estimates poses and textures in real time based on image sequences. It is more accurate and reliable than the above method, and it supports conventional monocular, binocular and RGB-D cameras. ORB-SLAM can complete real-time attitude estimation in indoor and outdoor environments of different sizes, it can create sparse resource point maps, provide closed-loop detection, extensive baseline redistribution, and fully automatic initialization. Compared with RGB-D SLAM, its disadvantage is that it limits the scope of application of the constructed sparse feature point map. It focuses on the positioning function and cannot be used for obstacle avoidance, route planning or autonomous robot navigation [3].

3. Octree map construction

VSLAM generally has front-end and back-end modules. The front-end solves sensor data association and attitude estimation, and the back-end solves local graph optimization, closed-loop detection and global optimization. In this paper, an octree map is constructed based on the optimized data of the ORB-SLAM algorithm, and it has been successfully applied to map construction and obstacle avoidance of mobile robots, as well as robot path planning, navigation and interactive operations. The key frames in ORB-SLAM store the corresponding camera poses, color images and depth images, build a point cloud map based on the data in the current key frame, and then create an octree map for the robot to avoid obstacles autonomously.

In the point cloud map, the 3D environment is described as a series of points: \( P = \{ P_1, P_2, \ldots, P_n \} \), Where: \( P_i = (x_i^C, y_i^C, z_i^C)^T \), The position information of the point in the 3D space, \( I \) represents the \( i \) th point; \( C \) is in camera coordinates [4].

According to the camera pinhole model, 3D points in space: \( (x_i^C, y_i^C, z_i^C)^T \), 2D pixels projected by the camera as 2D pixels on the image plane: \( (u, v)^T \). According to the geometric relationship \( u = x_i^C \cdot f / z_i^C \), \( v = y_i^C \cdot f / z_i^C \), the coordinates of the 3D points can be reversed according to the pixel points and the internal parameters of the camera, using a matrix The form is expressed as:
Among them: $f_u, f_v, C_u, C_v$ is the internal parameters of the camera, obtained by calibrating the camera. For RGB-D cameras, $z_i$ it can be obtained through depth image data. So far, the pixel point can be calculated from the formula (1) $(u, v)^T$ the corresponding spatial point $(x_i^C, y_i^C, z_i^C)^T$ the coordinate in the camera coordinate system. For all pixels in the key frame, the coordinates in the camera coordinate system are calculated by formula (1). Point in 3D space, coordinates in the world coordinate system $P_i^W$ the transformation between the coordinates in the camera coordinate system and the coordinates in the camera coordinate system can be calculated by equation (2):

$$P_i^W = R \cdot P_i + t$$  \hspace{1cm} (2)$$

In this formula, $R$ represents the rotation matrix of the camera, and $t$ represents the displacement vector. The coordinates in the world coordinate system can be obtained by changing the formula (2), and then the point cloud image corresponding to the key frame can be obtained[5]. As the scale of key frames becomes larger and larger, maintenance operations become more difficult. For robotic applications, such large-scale spatial points are not practical and cause a lot of waste. The octree map has a very convenient update mode, and can also be used to describe all 3D environments. Compared with the point cloud map, the octree structure hierarchical mode is used for storage, which can effectively compress and update the map, and can also intelligently map the resolution, thereby solving the problem of not being compact and avoiding obstacles. And navigation. The form of the use probability of the leaf nodes in the octree structure indicates the occupied, empty or unknown state of the space, that is, whether the point in the 3D space has obstacles or can pass through. In the process of constructing the map, for leaf node $n$, the observation data at $1, 2, ..., k$ can be known from Bayesian theory as $z_1, z_2, ..., z_k$, and through this node is:

$$Q(n) = \frac{1}{2} \left[ 1 + \ln \left( \frac{1 - Q(n \mid z_k)}{Q(n \mid z_k)} \right) \right]$$  \hspace{1cm} (3)$$

It can be seen from equation (4) that for any point in the three-dimensional space, each new observation is directly added to the previous observation result, and the update method is flexible and easy to implement. The occupancy probability of the parent node in the octree map can be calculated according to the value of the child node. There are mainly two methods: $I(n) = \sum_{i=1}^{8} I(n_i)$ taking the average value: $\hat{I}(n) = \max_{i} I(n_i)$ and taking the maximum value.
4. Autonomous mobile robot platform

In order to apply the octree graph to the autonomous obstacle avoidance of mobile robots on the ground, it established an autonomous mobile robot platform [6]. The body structure is Axial AX10, the camera is Kinect 2.0, and the controller is Pixhawk [12-13]. The unit is a microcomputer with a 4-core i7 processor, 16 GB memory and 256 GB solid-state drive. The structure of the system and the built-in mobile robot platform are shown in Figure 1.

![Figure 1. Main mobile robot platform](image)

The microcomputer reads the color image and depth image of Kinect2.0 through the USB3.0 interface, and publishes it in the form of ROS theme. Through algorithms, posture estimation and tracking, feature map construction and optimization, closed-loop detection and eighth tree graph construction are completed [7]. The pose estimation of the camera is used as the state estimation of the mobile robot, and the created octree map is used for autonomous obstacle avoidance of the mobile robot. Among them, the motion control instructions are issued in the form of ROS topics, and the Pixhawk controller subscribes to the topic data, and outputs signals to the electronic governor to control the motor, which realizes the motion control of the mobile robot platform [8].

5. Autonomous obstacle avoidance strategy

In order to verify that the algorithm can be successfully applied to the self-navigation ability of mobile robots in unknown and complex environments, its motion control part mainly adopts the roaming autonomous obstacle avoidance strategy. According to the constructed octree graph, the distance of obstacles in front of the mobile robot is detected [9,10]. When there is no obstacle in front of the robot or the obstacle distance is greater than the safe distance, it will issue a straight forward command. When the distance of the front obstacle is less than or equal to the safety distance, a left turn command is given. If the obstacle is still less than the safe distance after turning, issue the command to turn right and backward until there is no obstacle in front, and then issue the command to continue to explore the environment. The obstacle avoidance strategy code is as follows:
6. Conclusion
The ORB-SLAM-based system is a new system researched by RGB-D sensors---VSLAM system. Through key frame data, the new system can make the original system's map construction more perfect. It allows the robot to better avoid obstacles effectively, which is conducive to the maintenance of the robot.

Acknowledgment
This paper is funded by 2020 Guangdong Provincial Universities Characteristic Innovation Project “Research on VR Navigation System in High-risk Indoor Environment Based on Mobile Robot SLAM Technology”(2020KTSCX387)

References
[1] Wang Hongjian, Fu Guixia, BIAN Xinqian, LI Juan. Synchronous Positioning and Map Construction of Mobile Robots based on SRCKF [J]. Robotics, 2013: 74-81.
[2] Wang Quan, Hu Yue-li, Zhang He. Simultaneous positioning and map construction of mobile robots based on ROS and Kinect [J]. Research in Computer Applications, 2017: 310-313.
[3] Zhao Xiyu, Shen Yiming. Research on Simultaneous Positioning and Map Construction of Mobile Robot based on ICP [J]. Mechanical Engineering and Automation, 2018: 73-75.
[4] Lin Hui-can, LU Qiang, WANG Guo-sheng, ZHANG Yang, LIANG Bing. Simultaneous 3D localization and map construction of autonomous mobile robot based on VSLAM [J]. Computer Applications, 2017.
[5] Zhang Yi, ZHAO Liming, LUO Yuan. Research on Simultaneous Positioning and Map Construction of Mobile Robot based on MIEKF [J]. Computer Application Research, 2011: 108-110.
[6] Chen Qunying. Research on Simultaneous Positioning and Map Construction of Mobile Robots based on adaptive Method [J]. Microcomputer Applications, 2018.
[7] Peng Weizhi, Yuan Fengwei, Zhou Zhiwei. Analysis and Implementation of Mobile Robot Localization Based on RGB-D Camera [J]. Intelligent Computer and Applications, 2019: 175-177+182.
[8] ZHANG Yidong. Analysis of Navigation and Positioning Technology for Autonomous Mobile Robot [J]. Integrated Circuit Applications, 2017
[9] Hu Chunxu, Xiong Xiao, Ren Wei, He Dingxin. Localization and Navigation of Indoor Mobile Robot Based on Embedded System [J]. Journal of Huazhong University of Science and Technology (Natural Science Edition), 2013: 265-268+277.
[10] Wang Quan, Hu Yue-li, Zhang He. Mobile Robot Localization and Map Construction Based on ROS and Kinect [J]. Computer Application Research, 2017: 310-313.