



Development and Verification of an Autonomous and Controllable Mobile Robot Platform

Tianyu Jiang^{1,2} , Shaolin Zhang^{1*} , Rui Wang¹ , Shuo Wang¹

¹ State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences, 100190 Beijing, China

² Institute of Advanced Engineers, University of Science and Technology Beijing, 100083 Beijing, China

* Correspondence: Shaolin Zhang (zhangshaolin2015@ia.ac.cn)

Received: 02-16-2023

Revised: 02-28-2023

Accepted: 03-05-2023

Citation: T. Y. Jiang, S. L. Zhang, R. Wang, and S. Wang, “Development and verification of an autonomous and controllable mobile robot platform,” *Mechatron. Intell Transp. Syst.*, vol. 2, no. 1, pp. 11-19, 2023. <https://doi.org/10.56578/mits020102>.



© 2023 by the authors. Licensee Acadlore Publishing Services Limited, Hong Kong. This article can be downloaded for free, and reused and quoted with a citation of the original published version, under the CC BY 4.0 license.

Abstract: In this paper, we design a mobile robot platform, which employs a fully autonomous mechanical structure and electrical control system. Two driving wheels realize flexible steering movement with four universal wheels. A variety of sensors are built on the mobile robot platform, including the Inertial Measurement Unit (IMU) used to establish the inertial navigation coordinate system and the Velodyne’s Puck lidar sensor (VLP-16) used to obtain the three-dimensional (3D) point cloud information of the environment. Then, we build a software control architecture based on the Robot Operating System (ROS), using multi-node communication to perform positioning, environment perception, dynamic obstacle avoidance, path planning and motion control. Furthermore, a method of actively exploring the environment and constructing a map is proposed, using multi-path evaluation for real-time path planning and obstacle avoidance. In the end, we conduct autonomous exploration experiments to verify the performance of the designed mobile robot platform in indoor multi-obstacle scenes.

Keywords: Mobile robot platform; Autonomous exploration; Relative positioning; Obstacle avoidance; Map construction

1. Introduction

Mobile robot is an important branch of robotics field, which has the characteristics of simple mechanical structure, strong scalability, stable motion control, and easy operation. It can replace humans to perform specific tasks in complex and harsh environments. Mobile robots are widely used in industrial production, home service and other scenarios, such as building patrol, ground cleaning, warehousing, and freight transportation. Autonomous positioning and navigation, path planning, and map construction constitute the core of the mobile robot system, which is the premise for mobile robots to realize intelligence and autonomy [1-3].

So far, many researchers and institutions have built experimental platforms for mobile robots. National Key Lab of Intelligent Technology and System in Tsinghua University developed an effective and robust high-speed system called THMR-V (Tsinghua Mobile Robot V), which realized automatic control by analyzing lane lines in grayscale images [4]. Nguyen et al. [5] made Segway Robotic Mobility Platform, which was based on the self-balancing Segway Human Transporter (HT) and capable of about 13km/h indoor. Amsters and Slaets [6] built a robotics education platform using turtlebot3, which realized autonomous navigation by equipped with a 360° laser scanner. Oltean [7] made a mobile robot platform commended by Raspberry Pi and Arduino Uno interfaces, which had the ability to move as line follower robot with mapping, navigation, and obstacle avoidance features. Wu et al. [8] designed an omnidirectional mobile robot platform named Savvy, which was equipped with a variety of unified I/O interfaces and used the ROS framework. Existing mobile robot platforms have poor scalability, which are difficult to meet the various interfaces of sensors on the market. In addition, they are often oriented to fixed scenarios without the advantage of expanding other functions.

Autonomous exploration is an important application of robotics, where robot platforms explore unknown environments actively. Batinovic et al. [9] proposed a planner based on the detection of a boundary between the

explored and unknown part of the environment, which used the properties of the Octree environment representation to cluster the frontier points. Bircher et al. [10] employed a receding horizon “next-best-view” scheme in the proposed planner, where it found the best branch in an online computed random tree determined by the amount of unmapped space that can be explored and only the first edge of this branch was executed at every planning step. Selin et al. [11] used Frontier Exploration planning (FEP) as a global exploration planner to explore a large environment consisting of separate regions with ease and applied Receding Horizon Next-Best-View planning (RH-NBVP) for local exploration to explore individual regions. Dai et al. [12] presented an exploration strategy for the reduction of map entropy regarding occupancy probabilities, which exploited the implicit grouping of frontier voxels in the underlying octree map representation. Chen et al. [13] proposed an approach, which combined graph neural networks and deep reinforcement learning, enabling decision-making over graphs containing exploration information to predict a robot’s optimal sensing action in belief space. The above algorithms have high computational complexity and rarely build 3D maps of the indoor environment.

Mobile robot positioning refers to the judgment of its own pose and the position relative to the world coordinate system when the robot is running in the environment. Accurate positioning helps the robot to better perceive the surrounding environment information. Zheng et al. [14] used nonlinear graph optimization to achieve precise position estimation of the robot by combining lidar, IMU and Global Positioning System (GPS). Based on the Markov positioning algorithm, Wu and David [15] used sensor data and kinematic models to complete the positioning of mobile robots. Li et al. [16] proposed a fusion positioning algorithm based on ultra-wideband and lidar, which used a particle filter algorithm to perform complementary fusion calculations on the positioning data of the sensor, achieving good accuracy. At present, most researchers mainly use single-line lidar and cameras for environment perception and robot localization, while there are relatively few research works on multi-line lidar for active exploration in indoor environments.

The main contributions of this paper are summarized as follows:

(1) A mobile robot platform is designed with an autonomously controllable mechanical structure and control system, which is made of iron. The differential speed is controlled by the two driving wheels in the middle, and four universal wheels around are used as supporting weights. We write a set of basic controller algorithms that can achieve more flexible steering movements than other mobile robot platforms.

(2) A map construction method for active exploration of the environment is proposed. Under the premise that the environment is unknown, the robot reaches the artificially set global target point through the multi-path evaluation system. The environment map is constructed from the 3D point cloud data collected during the movement. Effective environment exploration and autonomous obstacle avoidance can be achieved without prior maps.

(3) Based on the designed mobile robot platform, an active exploration physical experiment is carried out to verify the stability of the mechanical structure and control system.

In the rest of this paper, the second chapter introduces the mechanical structure and electrical control system design of the mobile robot platform, the third chapter introduces the software architecture design of the mobile robot platform, and the fourth chapter explains the map construction method for the active exploration of the environment. The fifth chapter shows the active exploration of physical experiments, and the sixth chapter summarizes this paper and proposes prospects.

2. System Design of Mobile Robot Platform

2.1 Mechanical Design

The overall structure of the designed mobile robot platform is shown in Figure 1. The platform consists of two parts, the mobile chassis, and the bracket. The VLP-16 is installed on top of the bracket with the ability to detect a wide range of surrounding environmental areas for obstacle detection and map building. Laptop is placed in the hollow position in the middle of the bracket, which has a good fixing effect and is convenient for connecting various devices. The four corners of the mobile chassis are equipped with universal wheels, which play a supporting role and carry large loads. The two differential drive wheels are installed in the middle of the mobile chassis and pressed against the ground by springs, which enhances the adaptability to the ground and makes the driving more stable. At the same time, the spring structure effectively avoids idling. The mobile chassis shell is a 3D printed part that houses the electrical equipment. The fan is used for ventilation and heat dissipation. Some key parameters of the mobile robot platform are shown in Table 1.

The mobile robot platform has a stronger load bearing capacity. The combination of universal wheels and driving wheels provides the possibility for flexible steering functions. In addition, the setting of the emergency stop switch fully considers the safety factor, allowing people to cut off the power supply in time. Easy expansion of peripherals is another important advantage of the mobile robot platform. It provides a wealth of commonly used interfaces such as Universal Serial Bus (USB), Controller Area Network (CAN), Transistor-transistor logic (TTL), Recommended Standard 232 (RS232), Recommended Standard 485 (RS485) and Ethernet, which are used to add other peripheral devices that may be needed.

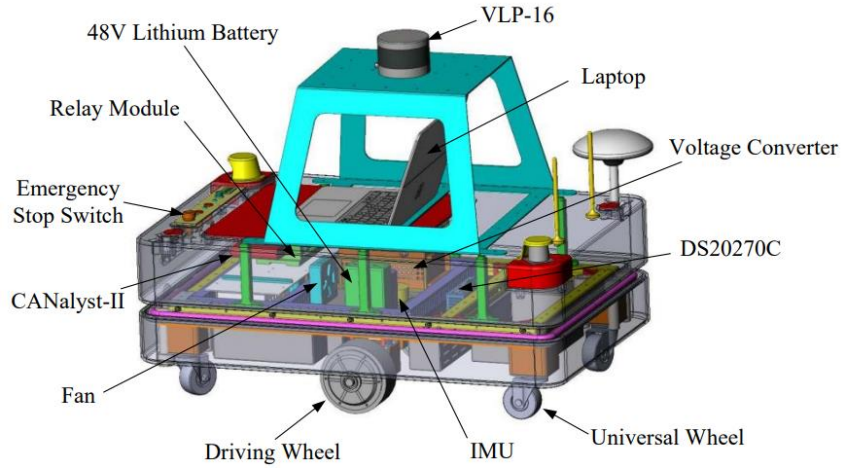


Figure 1. Structural design of mobile robot platform

Table 1. Key parameters of mobile robot platform

Name	Parameter
Installation height of VLP-16	0.75m
Body width	0.53m
Body length	0.86m
Maximum travel thrust of drive wheels	429N
Maximum travel speed	4m/s
Maximum load	248N

2.2 Mechanical Design

In order to facilitate timely debugging, the laptop is directly used to control in the research and development stage. The electrical control system is shown in Figure 2.

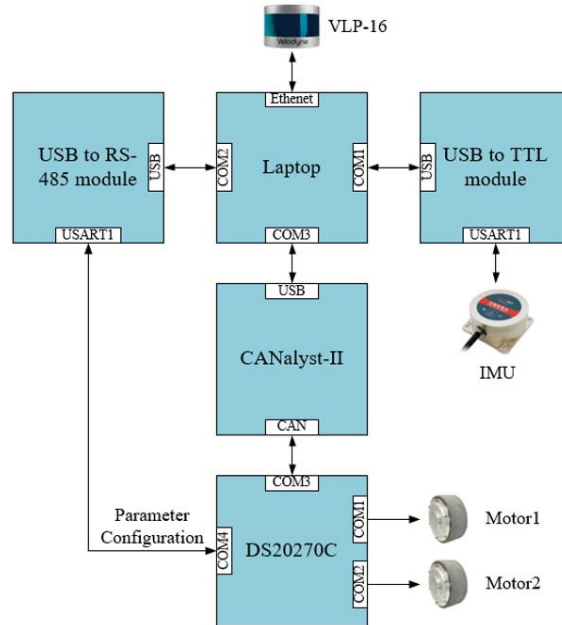


Figure 2. Control system configuration of mobile robot platform

The laptop is connected to the motor driver through the RS-485 communication protocol. The motor control mode selection and control parameter configuration are performed through the host computer software. After the configuration is completed, the speed command is sent through the CAN bus to control the two driving wheels, realizing functions such as forward, backward, and differential speed turning. The laptop is connected to VLP-16

through the Ethernet interface, and obtains the original point cloud information, which is used to describe the surrounding environment after digital filtering and coordinate transformation. In addition, the relay module is connected to the laptop through the Ethernet interface, and directly controls the power supply of the motor driver in the form of a switch. When encountering an unexpected situation, the mobile robot platform performs emergency braking by the relay module, avoiding the control delay of the motor driver. The IMU is connected to the laptop through the USB interface to provide the heading angle information of the mobile robot platform. The main control system device parameters used by the mobile robot platform are shown in Table 2.

Table 2. Device parameters of mobile robot platform control system

Name	Parameter
Laptop	Lenovo Legion Y7000P 2019 PG0
16-line lidar	VLP-16
Motor Controller	DS20270C
IMU	TL740D
Relay module	SRND-WM-RY1
USB to CAN module	CANalyst-II

The electrical control system uses a simplified communication protocol conversion module to unify the data transmission protocols required by different peripherals to the USB port, which means that it is friendly to the choice of the main controller. The modular design makes equipment replacement more convenient without affecting the normal operation of other functions.

3. Software Architecture Design of Mobile Robot Platform

The software framework of the system is shown in Figure 3. The laptop sits at the heart, managing all inputs and outputs with extended USB ports and switches. The ROS runs on the laptop to divide the program into different modules and transmit data by publishing and receiving topics, realizing the active exploration and real-time obstacle avoidance of the mobile robot platform. In the algorithm, the environment detection results are obtained periodically to judge the rationality of multiple planned paths. Then, by giving the motion sequence points of the mobile robot platform, the left and right driving wheels are differentially controlled.

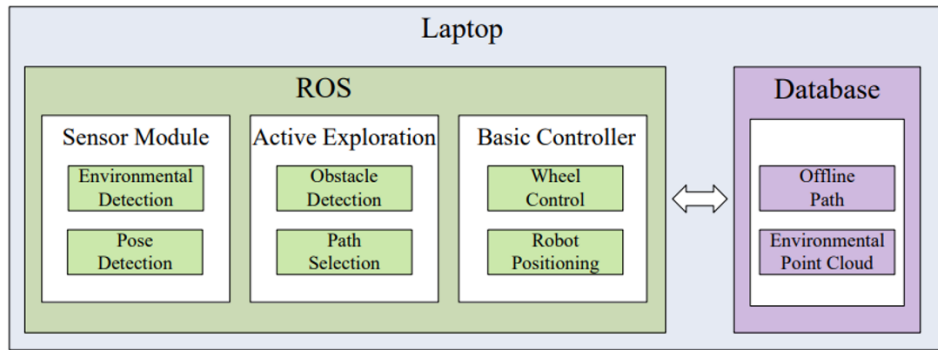


Figure 3. Software framework of mobile robot platform

The sensing module is divided into two parts, which are used to process the data of IMU and VLP-16 respectively. The base controller module is used for positioning and motion control of the mobile robot platform. The active exploration module sends speed commands to the basic controller module through the related topic, which is used for obstacle detection and path selection. A data packet of the IMU is 14 bytes, and the data transmission is set at a baud rate of 115200, which means that 822 valid data can be transmitted per second. The data of IMU is published to other nodes through its topic. The basic controller module obtains the attitude angle information of the mobile robot platform by subscribing to the topic of the IMU. Then, velocity information is published by the topic of the basic controller module. The dead reckoning method is used to estimate the position of the mobile robot platform, and the attitude information of the robot is released to other nodes through the related topic. The measurement frequency of VLP-16 is 5-20Hz, and the original point cloud information is obtained through the official driver, which is relative to the base coordinate system of VLP-16. We set the starting position of the mobile robot platform as the origin of the world coordinate system. By combining the real-time estimated pose of the mobile robot platform, the original point cloud is transformed into the world coordinate system as the basis for obstacle detection and path selection. The point cloud data during the exploration process is stored for offline mapping.

4. Active Exploration Map Construction Method

4.1 Mobile Robot Positioning

The basic principle of relative positioning is to calculate the pose change between adjacent moments through the internal sensor information on the premise that the initial pose of the mobile robot is given, realizing the real-time estimation of the pose. This method is also called dead reckoning, which predicts the trajectory of the robot based on the kinematics model [17]. The internal sensors commonly used in relative positioning methods mainly include odometers, gyroscopes, and inertial sensors. In this paper, our mobile robot platform adopts the IMU. We wrote a set of basic controller algorithms to drive the operation of the mobile robot platform. The code environment adopts C++ and ROS. The algorithm receives velocity and corner information through topics. In the information, the linear velocity in the x direction is always kept as the direction of the vehicle head, and the unit is m/s. The z-axis angular velocity is expressed as the spin angular velocity in rad/s, assuming that the counterclockwise direction is positive. That is, we control the forward, backward and turning of the robot. Assume that the traveling speed is v , the steering angular velocity is ω , and the robot's turning radius is r . Then the left wheel speed v_l and the right wheel speed v_r on the robot chassis can be expressed as Eq. (1).

$$\begin{aligned} v_l &= v - \omega * r \\ v_r &= v + \omega * r \end{aligned} \quad (1)$$

After that, the speed command is sent to the motor driver DS20270C through the CAN bus communication protocol, realizing the movement of the mobile robot platform.

In terms of dead reckoning, we use the moment obtained by the ROS timestamp as the integration time reference. Assume that the surrounding environment is in the same horizontal plane, the current pose of the robot is $P(x, y, z, r_x, r_y, r_z)$, the initial pose is $P_0(x_0, y_0, z_0, r_{x0}, r_{y0}, r_{z0})$, the current moment provided by the ROS is t_{now} , the previous moment is t_{before} , the heading angle provided by the IMU in real time is ψ , and the value range is $[-\pi, \pi]$. Then in the algorithm cycle, the robot pose can be estimated Expressed as Eq. (2).

$$\begin{aligned} x+ &= v * \cos(\psi) * (t_{now} - t_{before}) * c_x \\ y+ &= v * \sin(\psi) * (t_{now} - t_{before}) * c_y \\ r_z &= \psi \end{aligned} \quad (2)$$

Among them, c_x and c_y are calibration coefficients respectively, which are generated by system error and measurement error.

The positioning model above is streamlined and has the accuracy to meet the needs of task scenarios. It is undeniable that there will be cumulative errors in the dead reckoning method. However, when the moving speed is relatively fixed and the speed change frequency is low, the cumulative error is also relatively slow. Under the condition of short running time, there is an approximate linear relationship between the cumulative error and the actual distance, so the cumulative error is reduced to a certain extent by the calibration coefficient. We make the mobile robot move arbitrarily within a certain range for a period by means of remote control, and make it return to the initial position to obtain the cumulative error of dead reckoning. During the running process, the total distance change of the mobile robot platform in the x and y directions is recorded at the same time. By comparing the cumulative error with the total distance, the calibration coefficient is obtained.

4.2 Active Exploration Method

The proposed method uses multi-path evaluation for active exploration, and the multiple paths generated offline are shown in Figure 4.

The path points contained in it are stored in the form of data packets, which are directly called in the algorithm, saving computing resources greatly. The mobile robot platform obtains the environment point cloud information in real time through the VLP-16. Combined with its own length, width, and height, the mobile robot platform judges whether the planned path to be selected collides with environmental obstacles and obtains one or more effective paths. Then, according to the position of the end point, a path closer to the target is selected. Compared with reference [18] using intensive small-scale search and reference [19] using a search scheme based on likelihood estimation, the path adopted in this paper is sparser, which effectively reduce the amount of calculation. In addition, we add multiple paths on the left side of $y=x$ and $y=-x$, which prevent the mobile robot from getting stuck at local points. The algorithm process is as follows.

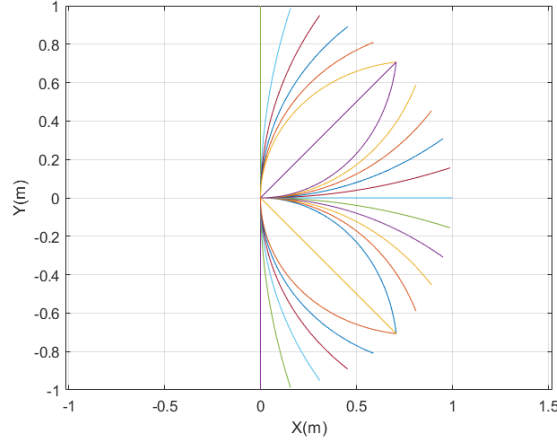


Figure 4. Multiple planning paths

Algorithm 1: Active Exploration Method Based on Multi-path Evaluation

1. Initialize multiple communication protocol configurations, mobile robot posture, and parameters of each node
 2. Obtain the 3D point cloud, and transform to the world coordinate system after performing voxel grid filtering, obtaining the point cloud set P . The number of points in it is set to N_p
 3. Read the offline path file and get N pieces of path information, including path number, path point coordinates and tangent slope of the point
IMU
 4. Obtain artificially set target point information
 5. **for** $i = 1 \dots N$ **do**
 6. **for** $j = 1 \dots N_p$ **do**
 7. According to the position of the path point, the slope of the tangent line at this point, and the length, width, and height of the mobile robot platform, calculate whether the i -th path collides with the j -th environment point, and determine the feasibility of the path
 8. **end for**
 9. If the i -th path is feasible, obtain the distance D_i between the end position of the path and the target point; if it is not feasible, set D_i to a larger number
 10. **end for**
 11. According to $D_1 \dots D_n$, select the best path to drive the mobile robot platform to explore
-

The point cloud set P is the 3D point cloud of the environment after filtering and coordinate transformation, including obstacle information. N represents 25 paths generated offline, and the path information is stored in other arrays. The set D represents the distance between the end point and the end position of the selected path, where we use the Euclidean distance.

In the process of active exploration, whether the environment point cloud accurately reflect the environmental information directly determines the accuracy of path selection in active exploration. Since VLP-16 has a blind area of one meter, the use of real-time environment point cloud for multipath evaluation often ignores important obstacle information. Therefore, in this paper, we unify the coordinates of the environment point cloud into the world coordinate system, and store a frame of environment point cloud at intervals to ensure that the scope of multipath evaluation is the environment information within the previous 3-5 seconds.

4.3 Offline Map Creation Method

During the active exploration process of the mobile robot platform, the 3D point cloud information of the environment is stored to generate an offline data package. Afterwards, the algorithm [20] is used for offline map building. It combines lidar and IMU data for mapping. By optimizing the ground points, high-precision 3D reconstruction of the environment is achieved. Since lidar data and IMU data are loosely coupled, it is also possible to use only lidar data for mapping. Constructing the environment map offline saves computing resources in the active exploration process of the mobile robot. At the same time, the offline method has the advantage of reviewing the exploration path selected by the mobile robot during operation and providing data support for the subsequent improvement of the algorithm. After the offline map is obtained, the effectiveness of the offline map is verified by measuring the positional relationship between obstacles and comparing it with the obstacle relationship in the real world. In addition, we draw a visual path for easy observation by subscribing to the odometer information.

5. Experiment

We conduct an indoor active exploration experiment, as shown in Figure 5.



Figure 5. Description diagram of autonomous exploration experiment for mobile robot platform

The starting point of the mobile robot platform is taken as the origin of the world coordinate system called /map. By artificially setting the target point, the mobile robot platform is driven to explore autonomously. The experiment is carried out in a $6m*6m$ space with multiple obstacles, including chairs and cardboard boxes. They are scattered throughout the environment to ensure that multiple paths are valid, avoiding artificially blocked and unsolvable situations. The $6m*6m$ space belongs to the small environment experiment scene. Compared with the real environment, the placement of four chairs and two cardboard boxes increases the complexity of the environment, which is more challenging for autonomous exploration.

At the beginning of the experiment, set an exploration target point through artificial estimation. After receiving the coordinates of the target point, the mobile robot platform starts active exploration. The experimental diagram is shown in Figure 6.

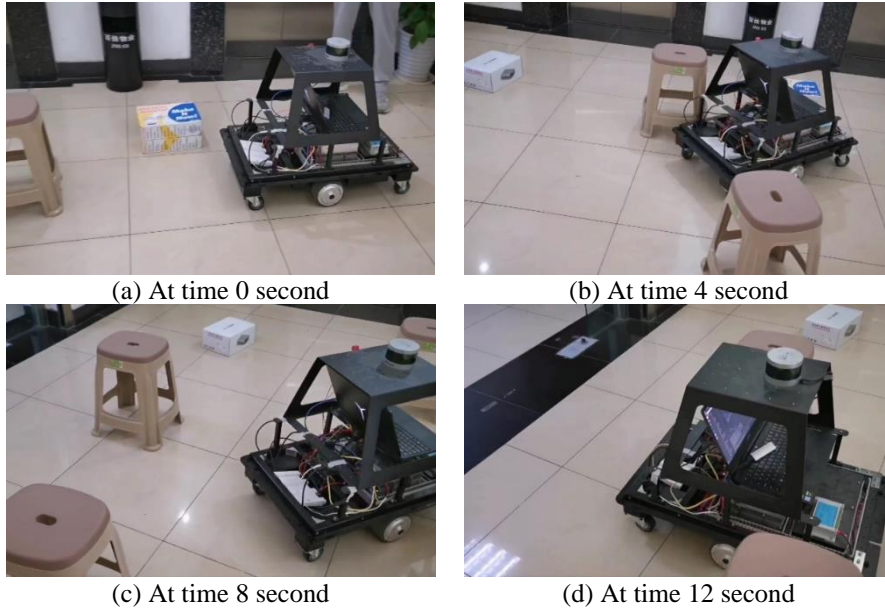


Figure 6. Autonomous exploration experiment of mobile robot platform

After that, we save the point cloud data during the autonomous exploration process and use it to build the environment map. The picture is shown in Figure 7.

After the mobile robot platform starts along the positive direction of the red x-axis, it judges the position of obstacles according to the VLP-16. When it encounters Carton 1 at time 0 second, the mobile robot platform adjusts its course to the left and chooses a path that avoids Chair 1. At time 4 second, the mobile robot platform is in the middle of Chair 1 and Chair 2, choosing to move on. Encountering the obstacle Chair 3 at time 8 second, the mobile robot platform adjusts its course to the right and chooses to avoid the obstacle Chair 4. At time 12 second, it reaches the artificially set target point.

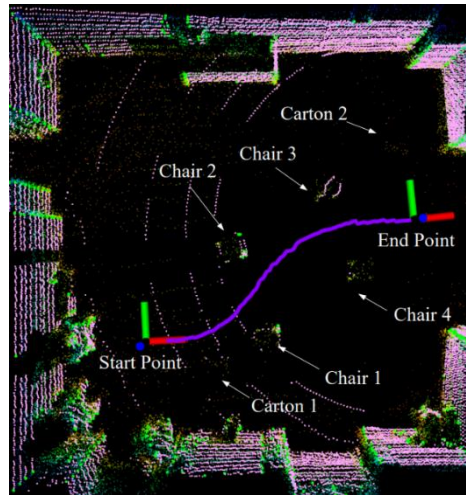


Figure 7. The environment map and exploration path obtained from the experiment

The experimental effect is quantitatively evaluated from two aspects. One is the exploration efficiency, which is evaluated by the number of explorations and the length of the exploration path. The number of explorations indicates how many times the mobile robot needs to choose a path to reach the target position, and the length of the exploration path indicates how much distance the mobile robot needs to travel to reach the target position. In the above experiment, the mobile robot selects a total of 4 paths. The second is information gain, which is mainly manifested in the quantity and quality of point cloud modeled on environmental objects. From the experimental picture, we see that this method has performed high-quality modeling on the surrounding walls and tables, but due to the influence of the sensor field of view, the modeling effect on chairs and cardboard boxes is poor. Only a small number of point cloud is obtained. Therefore, setting the starting point of the mobile robot is an important thing. There should be as few obstacles as possible within a certain range around the starting point, so that the final effect will be better.

6. Conclusions

In this paper, we design a mobile robot platform with autonomous controllable mechanical structure and control system, which has good mobility and scalability. In addition, we build a software control framework based on the ROS, using multi-node communication to realize the cooperation of various sensor data and control instructions. Finally, we propose an environmental active exploration map construction method, completing an active exploration experiment in the indoor environment by using multi-path evaluation. In the future, we will study active exploration methods that adapt to various complex environments and improve the operating efficiency of the algorithm.

Funding

This research was funded by the National Key Research and Development Program (Grant No.: 2020YFC1512102); the 2035 Innovation Task of the Institute of Artificial Intelligence Innovation, Chinese Academy of Sciences.

Data Availability

The data supporting our research results are included within the article or supplementary material.

Conflicts of Interest

The authors declare no conflict of interest.

References

- [1] C. Zhang, L. Zhou, and Y. Li, "Overview of multi-robot collaborative navigation technology," *Unmanned Syst. Technol.*, vol. 3, no. 2, pp. 1-8, 2020. <https://doi.org/10.19942/j.issn.2096-5915.2020.02.001>.
- [2] S. Zhao, C. Wang, J. Wang, and C. Li, "Design and implementation of low-cost autonomous navigation

- system for mobile robot platform,” *Unmanned Syst. Technol.*, vol. 5, no. 4, pp. 50-62, 2022. <https://doi.org/10.19942/j.issn.2096-5915.2022.4.038>.
- [3] Y. Wang and X. Wang, “Research on SLAM road sign observation based on particle filter,” *Comput Intel. Neurosci.*, vol. 2022, Article ID: 4478978, 2022. <https://doi.org/10.1155/2022/4478978>.
 - [4] K. Ni and K. He, “THMR-V: An effective and robust high-speed system in structured road,” In *SMC'03 Conference Proceedings, 2003 IEEE International Conference on Systems, Man and Cybernetics, Conference Theme - System Security and Assurance*, Washington, DC, USA, 2003, IEEE, pp. 4370-4374. <https://doi.org/10.1109/ICSMC.2003.1245672>.
 - [5] H. G. Nguyen, J. Morrell, K. D. Mullens, A. B. Burmeister, S. Miles, N. Farrington, K. M. Thomas, and D. W. Gage, “Segway robotic mobility platform,” *Mobile Robots XVII*, vol. 5609, pp. 207-220, 2004. <https://doi.org/10.1117/12.571750>.
 - [6] R. Amsters and P. Slaets, “Turtlebot 3 as a robotics education platform,” In *International Conference on Robotics in Education*, Vienna, Austria, pp. 170-181, 2019. https://doi.org/10.1007/978-3-030-26945-6_16.
 - [7] S. Oltean, “Mobile robot platform with arduino uno and raspberry pi for autonomous navigation,” *Procedia Manuf.*, vol. 32, pp. 572-577, 2019. <https://doi.org/10.1016/j.promfg.2019.02.254>.
 - [8] J. Wu, C. Lv, L. Zhao, R. Li, and G. Wang, “Design and implementation of an omnidirectional mobile robot platform with unified I/O interfaces,” In *2017 IEEE International Conference on Mechatronics and Automation*, Takamatsu, Japan, 2017, IEEE, pp. 410-415. <https://doi.org/10.1109/ICMA.2017.8015852>.
 - [9] A. Batinovic, T. Petrovic, A. Ivanovic, F. Petric, and S. Bogdan, “A multi-resolution frontier-based planner for autonomous 3D exploration,” *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 4528-4535, 2021. <https://doi.org/10.1109/LRA.2021.3068923>.
 - [10] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, “Receding horizon ‘next-best-view’ planner for 3d exploration,” In *2016 IEEE International Conference on Robotics and Automation*, Stockholm, Sweden, 2016, IEEE, pp. 1462-1468. <https://doi.org/10.1109/ICRA.2016.7487281>.
 - [11] M. Selin, M. Tiger, D. Duberg, F. Heintz, and P. Jensfelt, “Efficient autonomous exploration planning of large-scale 3D environments,” *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1699-1706, 2019. <https://doi.org/10.1109/LRA.2019.2897343>.
 - [12] A. Dai, S. Papatheodorou, N. Funk, D. Tzoumanikas, and S. Leutenegger, “Fast frontier-based information-driven autonomous exploration with an MAV,” In *P 2020 IEEE International Conference on Robotics and Automation*, Paris, France, 2020, IEEE, pp. 9570-9576. <https://doi.org/10.1109/ICRA40945.2020.9196707>.
 - [13] F. Chen, J. D. Martin, Y. Huang, J. Wang, and B. Englot, “Autonomous exploration under uncertainty via deep reinforcement learning on graphs,” In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Las Vegas, NV, USA, 2020, IEEE, pp. 6140-6147. <https://doi.org/10.1109/IROS45743.2020.9341657>.
 - [14] L. Zheng, Y. Zhu, B. Xue, M. Liu, and R. Fan, “Low-cost GPS-aided lidar state estimation and map building,” In *2019 IEEE International Conference on Imaging Systems and Techniques*, Abu Dhabi, United Arab Emirates, 2019, IEEE, pp. 1-6. <https://doi.org/10.1109/IST48021.2019.9010530>.
 - [15] Q. Wu and B. David, “A study on Markov localization for mobile robot,” *Acta Automatica Sinica*, vol. 29, no. 1, pp. 154-160, 2003. <https://doi.org/10.16383/j.aas.2003.01.021>.
 - [16] Z. Li, Y. Liu, F. Chang, J. Zhang, and M. Lu, “Research on UWB and LiDAR fusion positioning algorithm in indoor environment,” *Comput. Eng. Appl.*, vol. 57, no. 6, pp. 260-266, 2021. <https://doi.org/10.3778/j.issn.1002-8331.2005-0435>.
 - [17] D. Lu, Y. Zhang, Z. Gong, and T. Wu, “A SLAM method based on multi-robot cooperation for pipeline environments underground,” *Sustainability*, vol. 14, no. 20, Article ID: 12995, 2022. <https://doi.org/10.3390/su142012995>.
 - [18] C. Cao, H. Zhu, F. Yang, Y. Xia, H. Choset, J. Oh, and J. Zhang, “Autonomous exploration development environment and the planning algorithms,” In *2022 International Conference on Robotics and Automation*, Philadelphia, PA, USA, 2022, IEEE, pp. 8921-8928. <https://doi.org/10.1109/ICRA46639.2022.9812330>.
 - [19] J. Zhang, C. Hu, R. G. Chadha, and S. Singh, “Falco: Fast likelihood-based collision avoidance with extension to human-guided navigation,” *J. Field Robot.*, vol. 37, no. 8, pp. 1300-1313, 2020. <https://doi.org/10.1002/rob.21952>.
 - [20] T. Shan and B. Englot, “LeGO-LOAM: Lightweight and ground-optimized lidar odometry and mapping on variable terrain,” In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Madrid, Spain, 2018, IEEE, pp. 4758-4765. <https://doi.org/10.1109/IROS.2018.8594299>.