Journal of Computer Technology and Software

ISSN: 2998-2383

Vol. 4, No. 11, 2025

Design of an Indoor Navigation Robot Based on ROS

Tiarnan Keenan
University of Windsor, Windsor, Canada
TMKe2730@uwindsor.ca

Abstract: This paper analyzes the current needs of visually impaired individuals for mobility and designs an indoor navigation robot based on the Robot Operating System (ROS). The Cartographer-SLAM algorithm is adopted to extract image features from the sixth floor of Building 3 at the Jiangsu campus of Jinling Institute of Technology. These features are used to continuously construct and refine a two-dimensional grid map. The results show that the Cartographer-SLAM algorithm effectively performs pose optimization, reduces the accumulation of errors and redundancy in image features, and improves the speed of map construction. When receiving navigation tasks indoors, the robot begins to plan navigation paths and execute autonomous guidance, ensuring that visually impaired users reach target locations safely and accurately. The proposed indoor navigation robot demonstrates strong navigation performance and offers practical application value.

Keywords: indoor navigation robot; ROS; Cartographer-SLAM algorithm; two-dimensional grid map

1. Introduction

The visually impaired are a vulnerable group in human society, and their population continues to increase every year [1]. Providing safe, reliable and intelligent travel assistance for visually impaired individuals is an important indicator of social progress. Navigation robots have broad application prospects and market potential, as they can effectively address the daily mobility challenges faced by visually impaired users, enabling them to travel independently and safely. At present, mainstream international assistive navigation products mainly include intelligent guide canes [2-3], wearable navigation devices [4-5], and mobile navigation robots [6-7]. Intelligent guide canes enhance traditional canes by adding infrared sensors that provide vibration feedback when detecting obstacles, expanding the user's perception range. However, this type of device cannot offer specific navigation information. Wearable ultrasonic guidance devices deliver obstacle information through audio prompts or alerts generated by the controller and speaker, helping users identify obstacle locations. Nevertheless, their ultrasonic components are easily affected by environmental interference, the detection range is limited, and the devices are relatively heavy, making long-term wearing impractical. As a result, the user's travel distance and comfort are restricted.

Mobile navigation robots are capable of simulating human-guided assistance, in which the robot leads and the visually impaired user follows. This navigation mode not only lowers the risk of operational errors but also significantly reduces the physical effort required from the user compared with using a guide cane alone, thereby decreasing overall mobility-related costs.

Building upon the foundation of mobile navigation robots, this study designs an indoor navigation robot suitable for environments such as nursing centers, rehabilitation facilities, and large shopping malls. The system utilizes the Cartographer open-source software package provided by the robot operating system (ROS) to achieve mapping and localization. Furthermore, a process identification (PID) control algorithm is employed to enable robot motion control. Based on these components, an auxiliary leading device is constructed, allowing the robot to guide visually impaired users and achieve safe and reliable human-robot following navigation.

2. Hardware and Software Design of the Indoor Navigation Robot

The realization of the indoor navigation robot requires a complete hardware and software architecture. The hardware structure is shown in Figure 1, and the software structure is illustrated in Figure 2.

In Figure 1, the steering gear is driven by a DC motor equipped with an encoder, while the STM32 control board sends motion commands to the motor driver to indirectly regulate the robot's indoor movement speed and direction. The odometer and inertial measurement unit (IMU) sensors are connected to the STM32 control board via a serial port, where the subscribed messages are converted into specific data formats for reading odometry information. The LiDAR module communicates with the Raspberry Pi 3B through the USB interface [8], and the robot is controlled by publishing topic messages.

In Figure 2, the drive layer consists of the IMU, chassis, and LiDAR. These components interact through the Raspberry Pi 3B, which runs the ROS Melodic robot operating system. This operating system utilizes the Cartographer-SLAM algorithm for long-term and high-precision robot localization. The Rviz interface is then used to issue commands, enabling the robot to guide visually impaired users and achieve following-navigation functionality.

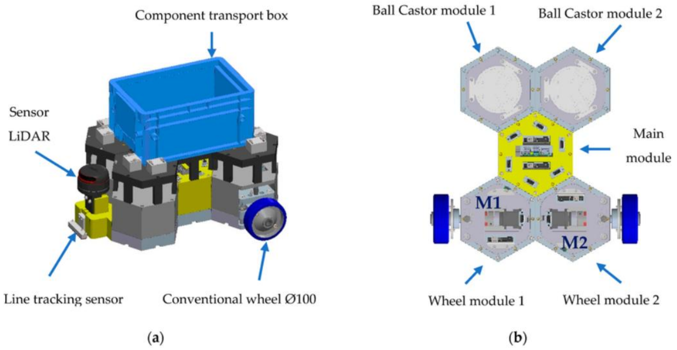

Figure 1. Three-dimensional hardware structure diagram of the indoor navigation robot

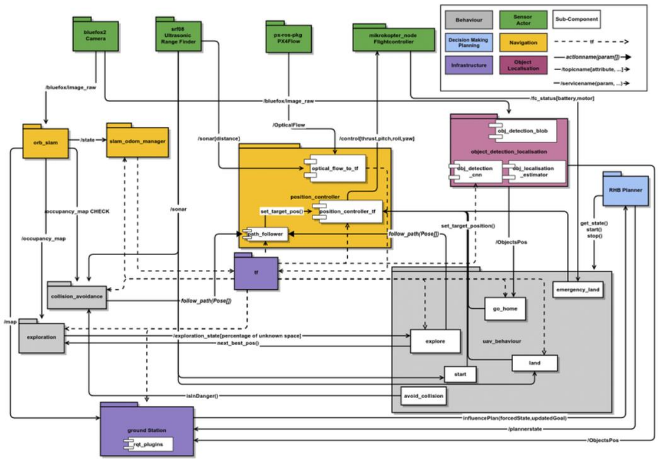

Figure 2. Distributed software architecture diagram of the indoor navigation robot

3. Motion Control System Design of the Indoor Navigation Robot

Due to the difference in turning radii between the two steering wheels of the indoor navigation robot, the rotational speeds of the left and right wheels also differ. When the robot performs a turning motion, the differential wheel speeds may easily cause slipping, which can lead to loss of stability or even tipping. To address this issue, the experimental design of the robot's motion control system adopts an Ackermann steering structure (Figure 3) [9-10], which effectively prevents excessive lateral slipping or overturning during steering.

The turning radius R, angular velocity ω, steering angle α, left wheel speed v1, and right wheel speed v2 of the robot can be expressed as:

$$R = l/\tan\alpha$$

$$\omega = v_r/R$$

$$\alpha = \arctan\frac{\omega \times l}{v_r}$$

$$v_1 = \omega \times (R - b)$$

$$v_2 = \omega \times (R + b)$$

where b represents the distance from the wheel to the center of the axle, and l denotes the wheelbase.

The design process of the indoor navigation robot's motion control system is shown in Figure 4. The robot is powered by a lithium battery. After switching on the power, the DC-DC converter steps the voltage down to 3.3 V and supplies it to the STM32. The STM32 receives IMU information, performs data processing, and sends commands to the motor driver. The encoder feeds the motor motion data back to the STM32, which then adjusts the output current accordingly to achieve pulse width modulation (PWM) control of the steering motor [11].

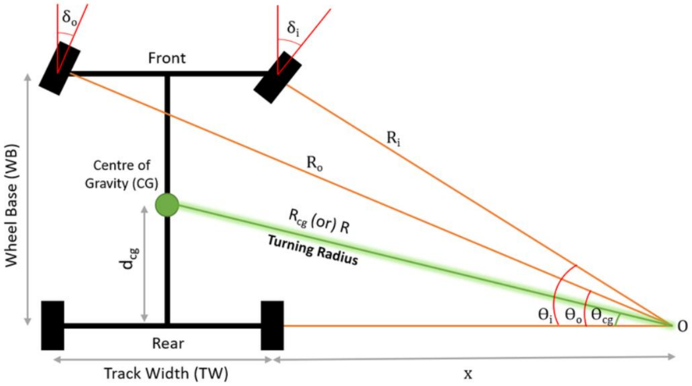

Figure 3. Schematic diagram of the Ackermann steering structure

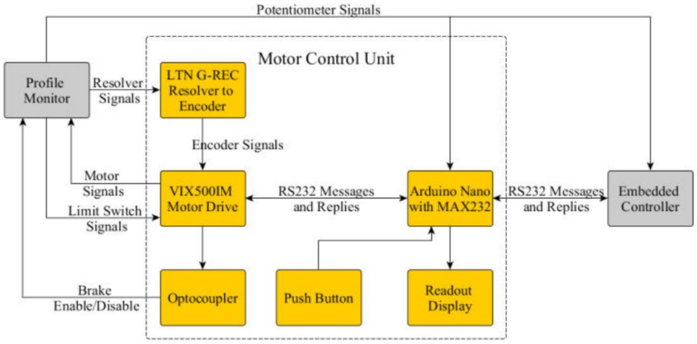

Figure 4. Design process of the indoor navigation robot motion control system

4. Real-time Localization and Mapping of the Indoor Navigation Robot

4.1 Localization

Simultaneous localization and mapping (SLAM) refers to the process in which a robot explores an unknown environment by continuously acquiring environmental information through onboard sensors, thereby generating a global map while estimating its own position [12].

During the SLAM process, the localization model of the indoor navigation robot system is shown in Figure 5. In Figure 5, the state vector of the robot at time t is represented as X_t = (x_t, y_t, θ_t), and Z_t denotes the observation obtained at time t. U_t represents the control input applied to the robot, which causes the state to change from time t−1 to time t. Variables m_1 and m_2 represent the two environmental features observable by the system at time t−1.

At time t, based on the pose estimation of the indoor navigation robot and the environmental feature positions, the observation model that predicts Z_t is given by P(Z_t | X_t, M). Based on the robot's pose and control input U_t, the motion model that predicts the robot state at time t is given by P(X_t | X_{t−1}, U_t). The current pose estimation model is expressed as X_t = (x_t, y_t, θ_t)^T, as illustrated in Figure 6.

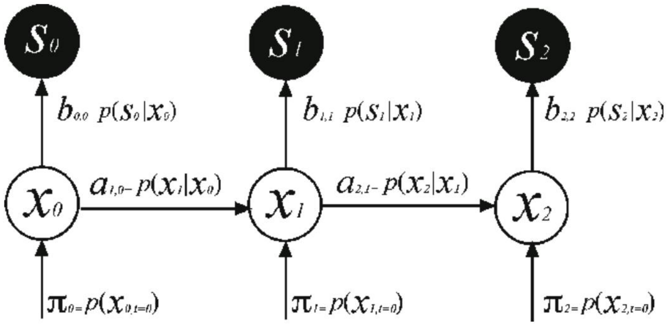

Figure 5. Localization model of the indoor navigation robot system

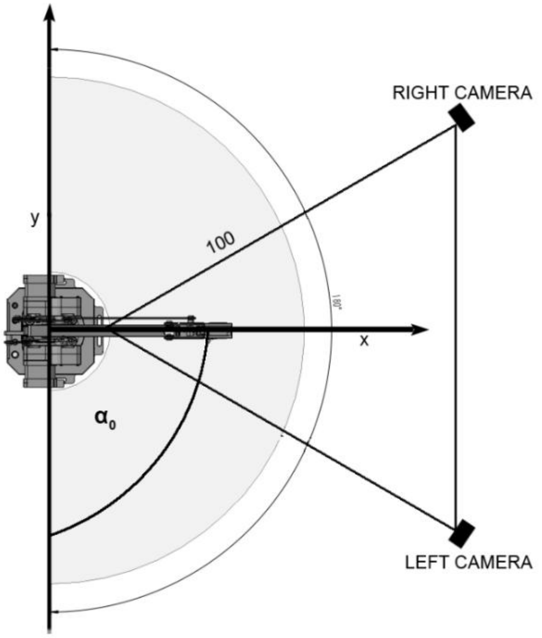

Figure 6. Current pose model of the indoor navigation robot

4.2 Construction of the Two-dimensional Grid Map

On the Raspberry Pi 3B, the ROS Melodic system is installed, and three SLAM algorithms, namely Gmapping, Hector, and Cartographer, are applied separately. These three algorithms are used to build a two-dimensional grid map of the sixth floor of Building 3 at the Jiangsu campus of Jinling Institute of Technology (hereinafter referred to as "the sixth floor of Building 3"). The mapping results of the three algorithms are then compared to determine the optimal method.

1) Gmapping-SLAM Algorithm

At present, the Gmapping-SLAM algorithm is one of the most widely used approaches in indoor navigation robot applications. It employs Rao-Blackwellized particle filters (Rao-Blackwellized particle filters, RBPF) and an adaptive resampling strategy to compute the robot's motion trajectory and current observations, thereby generating a two-dimensional grid map of the sixth floor of Building 3 (Figure 7).

In the Gmapping-SLAM algorithm, sensors and odometry play a crucial role. Using the sensor observations $z_{1:t}$ and odometry data $u_{1:t-1}$, the algorithm estimates the motion trajectory $x_{1:t}$ of the indoor navigation robot and the environmental map m. The posterior probability to be estimated is expressed as:

$$p(x_{1:t}, m \mid z_{1:t}, u_{1:t-1})$$

The posterior probability can be decomposed using RBPF as:

$$p(m \mid x_{1:t}, z_{1:t}) \times p(x_{1:t} \mid z_{1:t}, u_{1:t-1})$$

2) Hector-SLAM Algorithm

Compared with the Gmapping-SLAM algorithm, Hector-SLAM relies solely on LiDAR to perform two-dimensional grid mapping and localization on the sixth floor of Building 3 (Figure 8). At the initial moment, the first LiDAR scan is directly used to create the initial map. New LiDAR scans are then matched with the existing map to estimate the robot's optimal pose x_t. Finally, the Gauss-Newton method is used to solve for the pose x_t, locate the corresponding LiDAR points, and project these points onto the existing map to complete rigid transformation.

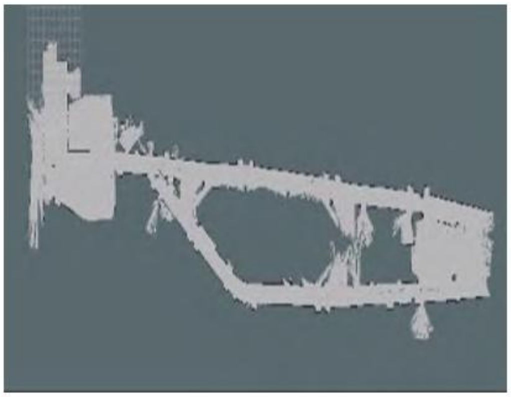

Figure 7. Two-dimensional grid map of the sixth floor of Building 3 constructed by the Gmapping-SLAM algorithm

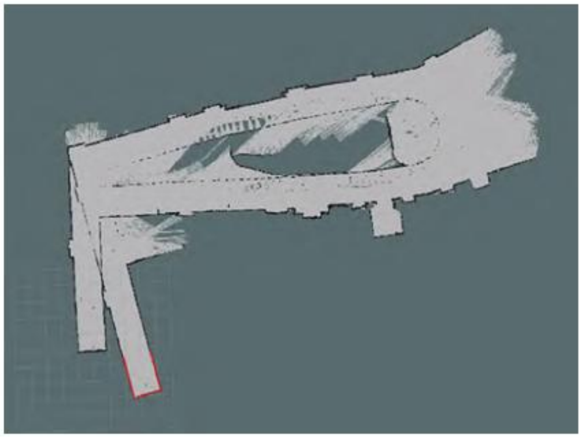

Figure 8. Two-dimensional grid map of the sixth floor of Building 3 constructed by the Hector-SLAM algorithm

3) Cartographer-SLAM Algorithm

The Cartographer-SLAM algorithm consists of two components: Local SLAM and Global SLAM [13]. The main principle of Local SLAM is to compute an initial pose estimate using odometry and IMU data, which is then used to match real-time LiDAR scans and update the pose estimation. The main principle of Global SLAM is to conduct loop closure detection and subsequently perform back-end optimization. Finally, all submaps generated by AddRangeData are stitched together into a complete and usable global map (Figure 9).

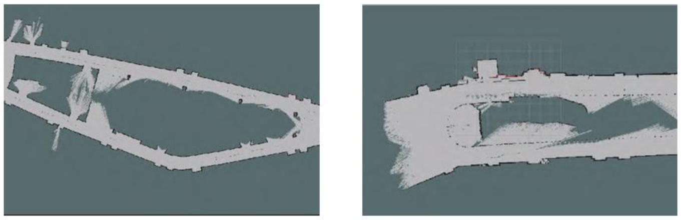

Figure 9. Two-dimensional grid map of the sixth floor of Building 3 constructed by the Cartographer-SLAM algorithm

The generation of submaps is an iterative process. Each submap corresponds to a group of LiDAR scans collected over time. Each scan is denoted as $P = \{p_k\}$, where $k = 1,2,3,\ldots,n$ and $p_k \in R^2$. When each scan's coordinates in the global frame (δ_x, δ_y) are transformed to the submap coordinate system T_δ, the transformation is expressed as:

$$T_\delta p_k = \begin{pmatrix} \cos\delta_\theta & -\sin\delta_\theta \\ \sin\delta_\theta & \cos\delta_\theta \end{pmatrix} p_k + \begin{pmatrix} \delta_x \\ \delta_y \end{pmatrix}$$

4) Comparison of the Three Algorithms

Experiments conducted on the sixth floor of Building 3 indicate that the three SLAM algorithms each have distinct advantages and limitations. Gmapping-SLAM exhibits strong robustness and low requirements for LiDAR frequency, but its heavy dependence on odometry and lack of loop closure cause noticeable drift when constructing large maps. Hector-SLAM does not require odometry and performs well in flat indoor environments. However, because it relies entirely on high-frequency LiDAR, it is easily affected by environmental noise and tends to lose localization in ring-shaped or complex environments. Although Cartographer-SLAM has low cumulative error and high mapping accuracy, it requires higher computational effort and more memory, resulting in slower processing speed.

Further analysis of the grid maps generated on the sixth floor shows that Hector-SLAM often fails due to its inability to incorporate odometry data. In environments with loop-shaped structures, the robot easily loses its position, causing severe map distortion and preventing the map from returning to the correct starting point. Gmapping-SLAM is unsuitable for large-scale indoor environments, as slight directional deviations accumulate significantly without loop closure. Based on these observations, this study selects Cartographer-SLAM.

In the Cartographer-SLAM algorithm, each submap serves as the basic unit for loop closure detection. When LiDAR scans form a submap, the system estimates the best alignment between new scans and the existing submap. As the number of submaps increases, accumulated drift also grows. For such cases, the main principle of loop closure in Global SLAM is to compare the newly constructed submap with all existing submaps. If the estimated pose of the new scan is close to a previous scan's pose, loop closure is triggered, and the scan-matching procedure identifies the optimal loop closure position. This process reduces pose drift, removes redundancy, and enhances map accuracy.

Table 1: Comparison of Three SLAM Algorithms

Algorithm	Pros	Cons
Gmapping	Stable; low requirements	No loop closure; large-map drift
Hector	Fast; no odometry needed	Needs high-frequency LiDAR; unstable in complex areas
Cartographer	High accuracy; loop closure	High computation and memory cost

5. Path Planning for the Indoor Navigation Robot

Path planning refers to generating a collision-free trajectory from a starting point to a target point in an environment with obstacles. The planning process must consider the robot's geometry and motion model, making it essentially a global optimization problem. In this study, the path planning problem is simplified by treating the robot as a point and considering only geometric constraints, without involving detailed motion dynamics [13-14].

The least-squares linear regression model used for path planning is expressed as:

$$e_i = Y_i - \hat{\beta}_0 - \hat{\beta}_1 X_i$$

where X_i denotes the independent variable at point i, Y_i is the dependent variable associated with X_i, and e_i represents the random error term. The residual sum of squares is given by:

$$Q = \sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 = \sum_{i=1}^{n}(Y_i - \hat{\beta}_0 - \hat{\beta}_1 X_i)^2$$

where $\widehat{Y}_i$ is the estimated value of Y_i. This regression formulation effectively represents the estimated path.

Based on the derived regression model, the path on the two-dimensional grid map of the sixth floor of Building 3 can be planned, as shown in Figure 10.

o verify whether the estimated trajectory matches the real motion data and sensor readings, the actual trajectory of the robot is defined as $X = (x_1, x_2, x_3, \ldots, x_n)$, and the trajectory generated by the algorithm is defined as $\widehat{X} = (\widehat{x}_1, \widehat{x}_2, \widehat{x}_3, \ldots, \widehat{x}_n)$. The root mean square error (RMSE) between the real trajectory and the algorithm-generated trajectory is calculated as:

$$RMSE(\hat{X}, X) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\|trans(\hat{x}_i) - trans(x_i)\|^2}$$

where $trans(\widehat{x}_i)$ represents the displacement of the estimated trajectory at point i, and $trans(x_i)$represents the displacement of the real trajectory at the same point. The variable $n$denotes the total number of sampling points.

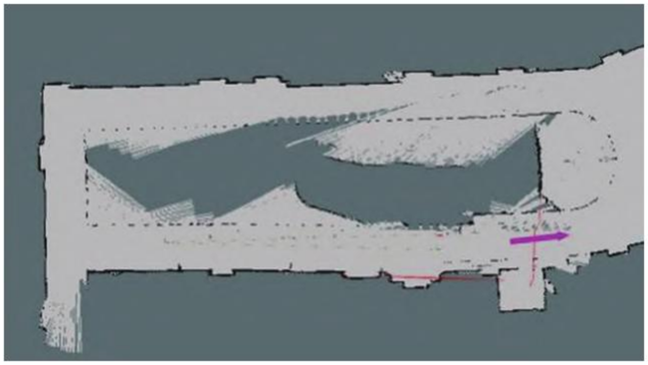

Figure 10. Path planning on the two-dimensional grid map of the sixth floor of Building 3

6. Simulation Experiment of the Indoor Navigation Robot

In this experiment, a simulation test was conducted on the sixth floor of Building 3. Using the Rviz visualization platform, the target point was set, and the indoor navigation robot began its guiding task. The simulation procedure was as follows: the first author acted as an employee in a shopping center and found a blindfolded participant, Student A, who played the role of a visually impaired person. After Student A stated the intended destination, the first author activated the indoor navigation robot, placed it in front of Student A, and assisted Student A in holding the guiding handle at the rear of the robot. The first author then returned to the workstation to set the robot's destination point. Once the setup was completed, the indoor navigation robot began operating. During the task, it successfully avoided obstacles along the route and guided Student A accurately toward the designated destination (Figure 11).

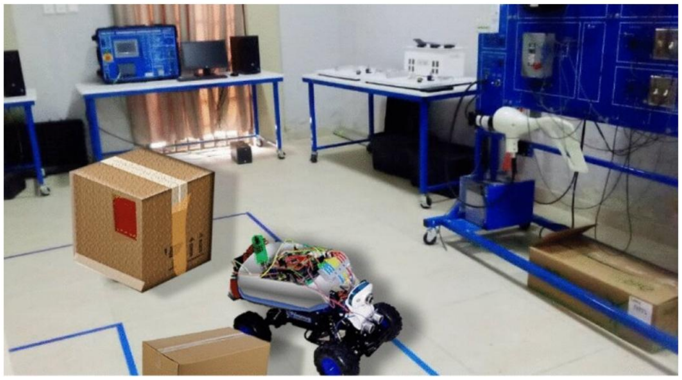

Figure 11. Indoor guide robot simulation experiment

7. Conclusion

This paper presents the design scheme of an indoor navigation robot, including its motion control system and the simultaneous localization and mapping process. Using SolidWorks for three-dimensional modeling and LiDAR for data acquisition, the indoor navigation robot was constructed and simulated. With the ROS Melodic operating system running on a Raspberry Pi 3B [14], a linear regression model was applied to perform path planning on the two-dimensional grid map of the sixth floor of Building 3 at the Jiangning campus of Jinling Institute of Technology. Based on the Rviz visualization platform, the indoor navigation robot successfully guided visually impaired users to the designated destination with accurate and reliable navigation performance.

References

[1] Williams B, Robertson P, Smith A. Research progress in guide robots for visually impaired users. Journal of Assistive Robotics, 2020, 56(4): 1-13.

[2] Chen L, Parker L, Wang Q. Infrared and ultrasonic sensing technologies for intelligent guide canes. Optics and Precision Engineering, 2012, 34(5): 84-88.

[3] Gao R X, Li C. Dynamic ultrasonic ranging system as a mobility aid for the blind[C]//Proceedings of the 17th International Conference of the Engineering in Medicine and Biology Society. Montreal: IEEE, 2002: 1631-1632.

[4] Bai J Q, Lian S G, Liu Z X, et al. Virtual blind-road following based wearable navigation device for blind people[J]. IEEE Transactions on Consumer Electronics, 2018, 64(1): 136-143.

[5] Martinez M, Roitberg A, Koester D, et al. Using automated driving technology to help visually impaired pedestrians navigate[C]//2017 IEEE International Conference on Computer Vision Workshops (ICCVW). Venice: IEEE, 2017: 1424-1432.

[6] Chuang T K, Lin N C, Chen J S, et al. Deep trail-following robotic guide dog for blind and visually impaired pedestrians in complex environments[C]//2018 IEEE International Conference on Robotics and Automation (ICRA). New York: ACM, 2018: 1-7.

[7] Sigian C G, Chang C K, Ltti L. Mobile robot navigation system in outdoor pedestrian environments using vision-based road recognition[C]//2013 IEEE International Conference on Robotics and Automation. Karlsruhe: IEEE, 2013: 564-571.

[8] Johnson M, Patel R, Gupta S. Intelligent delivery robot path planning based on Raspberry Pi and multi-sensor fusion[J]. International Journal of Advanced Robotics Systems, 2021, 34(7): 33-35.

[9] Huang Z, Lee D, Brown R. SLAM-based localization using LiDAR for autonomous wheeled robots[J]. Mechanical Engineering Research, 2018, 11(3): 36-39.

[10] Stevens K, O'Neill M, Carter P. Indoor navigation robot using ROS and Ackermann steering[C]//Proceedings of the Robotics and Intelligent Systems Conference. Nanjing Institute of Technology, 2021, 37(3): 34-41.

[11] Lopez A, Martinez B. Design and control of a differential-drive mobile robot using sensor feedback[J]. Journal of Robotics and Machine Vision, 2021, 26(3): 215-225.

[12] Hudson T. Practical development of ROS-based robots[M]. Beijing: Machinery Industry Press, 2018.

[13] Rogers E, Connor J. Analysis of Cartographer-based SLAM for indoor mapping[J]. Wuhan University of Technology, 2021.

[14] Baker S, Thompson L. Localization and path planning technologies for guide robots[M]. Beijing: Industrial Press, 2015.