Journal of Architectural Research and Development, 2025, Volume 9, Issue 3
https://ojs.bbwpublisher.com/index.php/JARD
Online ISSN: 2208-3537
Print ISSN: 2208-3529

Analysis of Path Planning and Navigation for Smart Plastering Robots Based on Indoor Construction

Yun Zeng*, Zhenzhou Ding, Daipeng Chen, Mi Xiao

School of Engineering and Architecture, Chwongqing Institute of Engineering, Chongqing 400056, China

***Corresponding author:** *Yun Zeng, zengyun666@126.com*

Copyright: © 2025 Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY 4.0), permitting distribution and reproduction in any medium, provided the original work is cited.

Abstract: Taking modern indoor building construction as an example, this study analyzes the path planning and navigation of a smart plastering robot. It includes a basic introduction to smart plastering robots, an analysis of multi-sensor fusion localization algorithms for smart plastering robots, and an analysis of path planning and navigation functions for smart plastering robots. It is hoped that through this analysis, a reference is provided for the path planning and navigation design of such robots to meet their practical application needs.

Keywords: Construction engineering; Indoor construction; Smart plastering robot; Path planning; Navigation function

Online publication: May 30, 2025

1. Introduction

In modern indoor construction projects, the smart plastering robot is a high-tech equipment. Through precise path planning and intelligent navigation, this robot can complete indoor plastering operations with high quality and efficiency, improving the plastering effect and reducing labor costs [1]. To achieve the rational application of such robots, researchers need to analyze their path planning and intelligent navigation based on their basic conditions and multi-sensor fusion localization algorithms, ensuring the effectiveness of their applied functions.

The smart plastering robot is an advanced intelligent robotic device used in wall plastering construction in the modern architectural engineering industry. Its basic working method involves acquiring environmental parameters through intelligent sensing technology and combining actual conditions with plastering task requirements to perform intelligent path planning and motion control, achieving good intelligent navigation results. With advantages such as high work efficiency, good plastering quality, high safety, low labor intensity, and low labor costs, smart plastering robots have been widely used in modern architectural engineering wall plastering work [2]. Especially in indoor plastering construction, the application of smart plastering robots has attracted much attention. Therefore, the intelligent path planning and navigation of such robots have become key research foci for relevant researchers in recent years.

2. Analysis of multi-sensor fusion localization algorithm for smart plastering robots

2.1. Introduction to the dataset

The dataset plays a critical supporting role when the smart plastering robot performs localization through a multi-sensor fusion algorithm. This algorithm aims to solve the position estimation problem of the robot under known scene conditions. Here, the known scene refers to the availability of RGB images and corresponding depth map information for each frame, satisfying the establishment of a 3D model of the scene before estimating the robot's position. Using the robot's position in the first frame as the origin of the camera coordinate system, the ICP algorithm is employed to estimate the camera trajectory and determine the coordinates of each frame in the RGB image within the camera coordinate system [3]. A point within the scene is selected as the origin in the world coordinate system, and the correspondence between the camera coordinate system and the world coordinate system is calculated. Based on the estimated camera motion trajectory, the corresponding world coordinate positions for each frame of the RGB image can be derived.

2.2. Localization network framework

During the localization process of the robot, the main implementation steps of its localization algorithm include the following:

(1) Scene coordinate regression

The RGB images acquired from the robot's scene are used as the training set, each frame of the images serves as the validation set, and the estimated robot motion trajectory is utilized as the test set, with a data configuration ratio of 7: 1: 2. Using a 640*480px RGB image training set as the basic data, a corresponding network model is established through a combination of BIM and 3D MAX software. The data from the RGB image training set is imported into this model, generating scene prediction coordinates with a specification of 80*60.

(2) Hypothesis sampling and selection

Hypothesis scoring is implemented for the predicted coordinate positions of the robot in the scene using the sigmoid function method. The following is the hypothesis scoring formula:

$$a(h) = \sum_i sig(\alpha - \beta(e_i(h, \omega))) \tag{1}$$

In the formula, $\alpha(h)$ represents the hypothesis score for the robot's position in the scene; h represents the hypothesis for the robot's position in the scene; α represents the inlier threshold; β represents a function control value; e_i represents the error function in the case of reprojection; and ω represents the learning parameter.

(3) Hypothesis extraction

Based on the actual requirements of the algorithm, initialize the variable ω and continuously optimize e_i during training. Grid optimization is then carried out using the nonlinear Gauss-Newton method until convergence to a specific grid form, followed by validation. To prevent overfitting due to excessive iterations, researchers reasonably determine the frequency of model result storage based on the original network. It is decided that after reaching 50,000 iterations, the model results will be saved every 5000 subsequent iterations. If the error and loss drop below a certain threshold, and the model begins to converge, model training can be stopped. This approach effectively reduces training time, enabling rapid acquisition of robot position information [4].

2.3. Position information correction

To achieve accurate acquisition of robot positioning data, researchers can adjust the laser information in the four directions (up, down, left, and right) of the robot based on the obtained positioning results. **Figure 1** shows a schematic diagram of the basic principle of robot laser positioning.

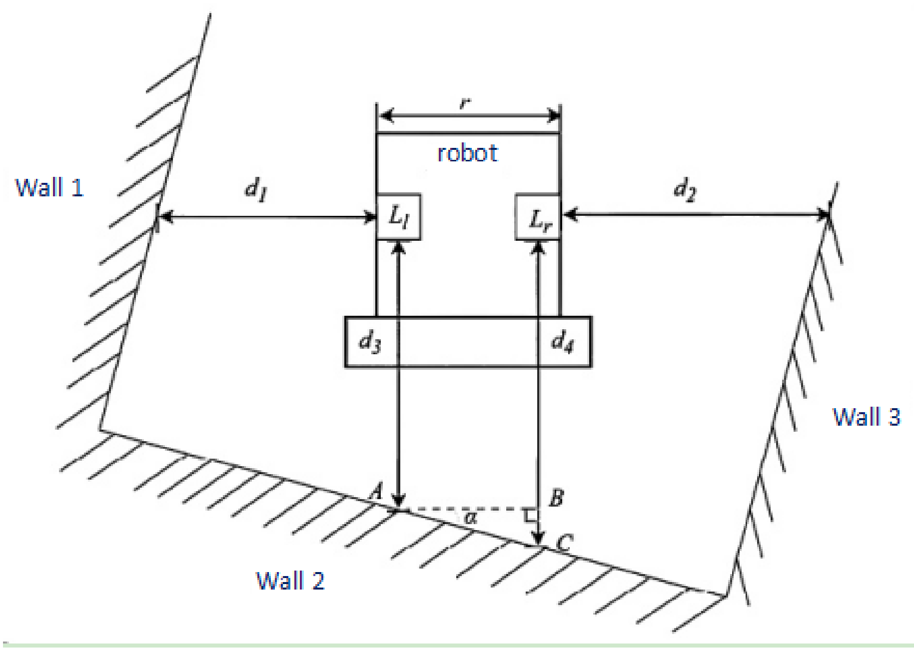

Figure 1. Schematic diagram of the basic principle of robot laser positioning

In this diagram, two single-line LiDARs, Laser1 and Laser3, are set at the LI position. The Laser1 radar is pointed towards Wall 1, with a ranging information of d1; the Laser3 radar is pointed towards Wall 2, with a ranging information of d3. At the Lr position, two single-line LiDARs, Laser2 and Laser4, are set. The Laser2 radar is pointed towards Wall 3, with a ranging information of d2; the Laser4 radar is pointed towards Wall 2, with a ranging information of d4. The distance between LI and Lr represents the width of the robot itself, denoted as r.

Based on the above information, researchers can perform positioning correction on the robot. Assuming that the estimated position of the robot obtained after learning the positioning network framework is *pn*, and the true position coordinate of the robot is *pr*. Researchers can correct its position in three steps:

(1) Angle correction

Based on the geometric relationship shown in **Figure 1**, the angle of the robot can be calculated using the following formula:

$$\alpha = \arctan\frac{d_4 - d_3}{r} \tag{2}$$

Its parameters can be calculated based on specific laser ranging values. If the angle β between pr and Wall 2 is not equal to *α*, then *pn* needs to be reasonably corrected based on the actual situation.

(2) Forward and backward correction

First, calculate the distance *b* between the robot and the wall using the following formula:

$$b = \frac{d_3 + d_4}{2} \tag{3}$$

If the value of *b* is not equal to the distance between pn and Wall 2, pn needs to be reasonably corrected based on the actual situation.

(3) Left and right correction

Assuming that both *d1* and *d2* data are valid, if the sum of their distances to Wall 1 from *pn* is not equal to the sum of their distances to Wall 2 from *pn*, *pn* needs to be corrected based on the actual situation. In the process of single-line LiDAR ranging, if the returned data value is very large, it indicates that there may be a window or door in the ranging direction, and such data should be excluded during correction to ensure accurate correction and improve the positioning accuracy of the robot [5].

3. Analysis of path planning and navigation function for smart plastering robots

To ensure effective path planning and navigation for smart plastering robots, researchers have proposed an improved algorithm based on the A* algorithm. This enhanced approach incorporates a directional adjustment step at the end of the A* algorithm, tailored to the specific requirements of the robot's application scenario. This additional step, implemented after the navigation process, further enhances the robot's navigation capabilities. The detailed implementation process for path planning and navigation is as follows [6]:

(1) Acquire current positioning information and determine the endpoint for path planning.

(2) Assess whether path planning has been successful. If not, the process ends; if successful, proceed to the next step.

(3) Plan the robot's path based on the actual situation.

(4) Check if the robot has reached the endpoint coordinates. If not, return to step 1; if yes, continue to the next step.

(5) Adjust the robot's own angle.

In this process, both the original A* algorithm and its improved version are implemented through software calculations. These calculations require only the input of map data and the robot's movement position data, with the specific calculation process not elaborated here.

3.1. Determination of the starting point

Assuming the presence of obstacles on a map, to implement the A* algorithm for searching obstacle areas, the map is divided using the length and width dimensions of the robot as the basic unit. The divided map can be marked using a matrix, denoted as Map. In this matrix, obstacles are represented by the number 2, the current position of the robot is represented by the number 1, and positions without obstacles are represented by the number 0. After image conversion, a corresponding grid diagram is obtained, where white grids represent positions without obstacles and yellow grids represent positions with obstacles. **Figure 2** illustrates the basic situation of the map used in this experiment.

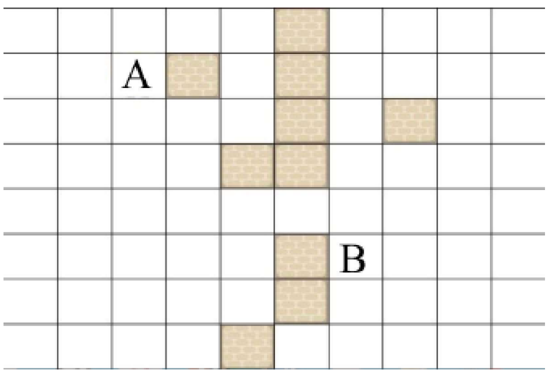

Figure 2. Schematic diagram of the basic situation of the experimental map

In this diagram, point A represents the current position of the robot obtained through the localization algorithm, while point B indicates the location where the robot needs to perform plastering work. Based on the matrix definition and related operational procedures mentioned earlier, researchers can denote point A as Map[1][2] and point B as Map[5][6].

3.2. Path planning

Based on the analysis of the improved A* algorithm, the robot's path planning map in the experiment is shown in **Figure 3**:

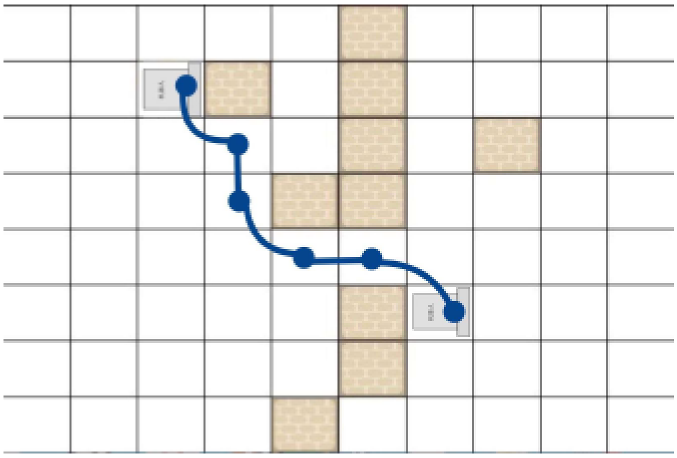

Figure 3. Schematic diagram of robot path planning obtained through improved A* algorithm analysis

It can be seen that for the static path of the robot, during the planning process, the improved algorithm can perform precise obstacle avoidance processing for all obstacles in the scene where the robot is located, to achieve reasonable planning of the robot's operating path. This allows the smart plastering robot to timely and effectively distinguish obstacles under unknown scene conditions, and to make reasonable planning for the next operating step based on the specific location of the obstacles, thereby achieving a good obstacle avoidance effect during operation.

3.3. Rotational navigation

After completing the operating path planning, the smart plastering robot can smoothly move from the starting point to the ending point. However, when the robot reaches the ending point (i.e., the plastering construction location), its actual orientation may not be the one needed for plastering. Based on this, in subsequent planning, researchers still need to continue implementing intelligent rotational navigation control for the robot [7]. To achieve this goal, researchers can use a multi-sensor fusion work mode to monitor the robot's rotational drive motor in real-time, enabling intelligent adjustment of its rotation angle and speed. In this process, multiple sensors use a wireless communication network to transmit environmental parameters and robot operating parameters obtained in real-time to the drive motor controller. The controller intelligently generates a motor drive strategy based on its path planning results and actual conditions, and promptly issues corresponding control commands to achieve real-time control of the motor's rotation direction and speed, driving the robot to rotate precisely [8]. This provides precise navigation for the robot's plastering actions, ensuring its plastering efficiency and construction quality.

3.4. Application experiment

Due to the time-sensitive characteristics of path planning and navigation for plastering robots in indoor construction scenes, researchers can create plastering construction scenes with different levels of complexity by setting different numbers of obstacles to verify the application effectiveness of established path planning and navigation methods. The path planning speed of the robot can be compared under different scene conditions. Specifically, during the experiment, it is assumed that the search area divisions of two rooms are completely different, with the matrix size of the first room controlled at 10*10 and the matrix size of the second room controlled at 50*50. The Visual Studio 2017 analysis software is installed on a Windows 11 64-bit computer system to randomly generate different obstacles and robot paths through coding. The open list and close list are stored in the form of a data structure queue. The feasibility of path planning and navigation effects is judged by calculating the average path planning speed of the robot under different conditions of the same room and varying numbers of obstacles.

Firstly, for the 10*10 room, because the amount of computation is relatively small, the computation time consumed is also relatively short as the number of obstacles increases. Simultaneously, the experiment found that as the number of obstacles in the room continues to increase, the planning time for the robot's reachable path shows a trend of first increasing and then decreasing. When the number of obstacles is 90, there is only one reachable path, so the computation time is significantly reduced. **Table 1** shows the variation in path planning computation time for the robot in a 10*10 room as the number of obstacles increases:

Table 1. Variation in path planning computation time for the robot in a 10*10 room with increasing number of obstacles

Serial number	Obstacle quantity	Path planning duration	Serial number	Obstacle quantity	Path planning duration
1	0	0ms	6	50	16ms
2	10	15ms	7	60	16ms
3	20	16ms	8	70	16ms
4	30	16ms	9	80	16ms
5	40	16ms	10	90	0ms

Next is the 50*50 room. Due to its relatively large computational requirements, researchers modified the source code during calculations. Specifically, the open list and close list, which were originally single-level queues, were improved to multi-level queue formats, and computational experiments continued. Through experimentation, it was found that as the number of obstacles in the room continues to increase, the planning time for the robot's reachable path exhibits a trend of first increasing and then decreasing. When the number of obstacles becomes excessive, the number of robot paths decreases significantly, leading to a noticeable reduction in subsequent path planning time. **Table 2** illustrates the variation in path planning computation time for the robot in a 50*50 room as the number of obstacles increases:

Table 2. Variation in path planning computation time for the robot in a 50*50 room with increasing number of obstacles

Serial number	Obstacle quantity	Path planning duration	Serial number	Obstacle quantity	Path planning duration
1	0	0ms	6	1250	9372ms
2	250	3583ms	7	1500	10848ms
3	500	4623ms	8	1750	9658ms
4	750	7396ms	9	2000	7482ms
5	1000	8239ms	10	2250	2672ms

Based on a comprehensive analysis of the above two experimental scenarios, it can be concluded that when the room matrix is the same, the path planning time for the plastering robot exhibits a trend of first increasing and then decreasing as the number of obstacles gradually increases. However, when the room matrix is different, a larger matrix results in longer path planning time. Nevertheless, considering the actual conditions of ordinary indoor architectural scenes, the path planning time is not excessively long, indicating strong feasibility for the overall path planning and intelligent navigation scheme.

4. Conclusion

In summary, for indoor plastering robots in construction engineering, researchers can conduct experimental calculations on path planning time by varying room matrices and the number of obstacles, provided that the path planning and navigation mechanisms are clearly understood. This approach effectively verifies the path planning and intelligent navigation effects, allowing for the assessment of the feasibility of robot applications.

Funding

Science and Technology Research Project of Chongqing Education Commission (Project No.: KJQN202401902), Chongqing Construction Science and Technology Plan Project (Project No.: Chinese Society For Urban Studies, 2024: 3-24) and Scientific Research Fund Project of Chongqing Institute of Engineering (Project No.: 2022gcky01), College Student Innovation and Entrepreneurship Training Program Project of Chongqing Institute of Engineering (Project No.: CXCY2024018)

References

[1] Ding Y, Xie L, Wang Y, et al, 2024, Trajectory Planning for Sideways Movement of Smart Wall Plastering Robots. Computer Simulation, 2024(7): 467–473, 478.

[2] Rao Q, 2023, Research on Structure and Performance Optimization of Indoor Wall Multi-angle Plastering Robots, thesis, Wuhan University of Science and Technology, Hubei.

[3] Shen H, 2020, Research on Robot Scanning Planning and Progress Monitoring Methods in Unknown Construction Scenes, thesis, Harbin Institute of Technology, Heilongjiang.

[4] Xiao W, Zhuang R, Tang W, 2023, Research on Construction of Plastering Robots. Construction Technology (Chinese and English), 2023(11): 22–26.

[5] Liu J, Zhou M, Xie L, et al, 2023, Performance Research on Indoor Wall Mortar Leveling Robots. Sensor and Microsystem, 42(7): 57–60.

[6] Liu L, 2023, Design Analysis of Interior Wall Leveling and Painting Robots. Engineering Construction and Design, 2023(22): 80–82.

[7] Shuai Z, 2022, Research on Intelligent Navigation of Indoor Construction Plastering Robots, thesis, University of Electronic Science and Technology of China, Sichuan.

[8] Wang W, Yuan Y, Yan H, 2024, Imaging Technology Research Based on High-frequency Radar Mounted on Wall-climbing Robots for Exterior Wall Insulation and Decoration Connections. Engineering Construction and Design, 2024(7): 18–20.

Publisher's note

Bio-Byword Scientific Publishing remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.