International Journal for Housing Science and Its Applications
Publish August 10, 2025. Volume 46, Issue 4 Pages 2171-2183

https://doi.org/10.70517/ijhsa464185

Hybrid Topological Map Construction and Application for Robot Path Planning and Obstacle Avoidance in Complex Local Spaces

Shushu Wang[1,*], Jianmin Zhang[1], Shengqiang Fan[1], Rui Wang[1] and Jin Xu[1]
[1] Marketing Service Center, State Grid Shanxi Electric Power Company, Taiyuan, Shanxi, 030000, China
Corresponding authors: (e-mail: WangShushu1991@163.com).

Abstract As robotics technology continues to mature, it has made people's lives more convenient. However, robot path planning and obstacle avoidance have become key research issues. To address these challenges, this study proposes a research framework for the construction and application of hybrid topological maps for robot path planning and obstacle avoidance in complex scenarios. Under the support of mobile robot SLAM theory, a topology-grid hybrid map is constructed. Since the topology-grid hybrid map exists in both static and dynamic scenarios, this study designs a static path planning and obstacle avoidance algorithm based on an improved A* algorithm and a dynamic planning and obstacle avoidance algorithm based on an improved TEB algorithm, and conducts validation instance analysis on both algorithms. In dynamic scenarios, the navigation success rate, average path length, and average time consumption of the proposed algorithms are superior to those of traditional algorithms, with values of 99.00%, 3.415m, and 16.35s, respectively. In static scenarios, similar phenomena are observed, demonstrating the effectiveness of the two dynamic path planning and obstacle avoidance algorithms. This research provides guiding value for the development of robot path planning and obstacle avoidance.

Index Terms SLAM, topological-grid hybrid map, A*, TEB, path planning and obstacle avoidance

I. Introduction

Robots were first proposed in the 1920s and 1930s as a class of mechanical devices and systems capable of performing specific tasks autonomously or with minimal human intervention [1]. With the continuous advancement of human science and technology, various types of robots have been invented and applied to various aspects of social production and daily life [2], [3]. Since the 21st century, with the sustained development of artificial intelligence and internet technologies, robots have begun to evolve toward intelligentization. Among them, mobile robots, as the earliest and fastest-developing type of robot, have achieved significant progress and are now widely applied in fields such as logistics warehousing, smart homes, intelligent living, and scientific research [4]-[7].

Mobile robot technology involves multiple disciplines such as computer science, sensors, human-machine interaction, and bionics [8]. Among these, environmental perception, autonomous localization, and path planning are the three key challenges in mobile robot technology, with path planning being the core and most critical component [9], [10]. Path planning for mobile robots refers to finding a relatively optimal and safe path from the starting point to the destination in an environment with static or dynamic obstacles [11]. As the application scenarios for mobile robots increase, the demand for path planning algorithms will also grow. For example, when mobile robots are applied in complex outdoor environments such as military reconnaissance, emergency rescue, geological exploration, and security inspections, considerations must extend beyond path length to include safety, smoothness, and multi-robot coordination and scheduling. This places higher demands on the algorithm’s interference resistance, response speed, execution efficiency, and code structure [12]-[15]. Therefore, path planning remains a critical research area within the field of mobile robots and is the core task for mobile robots to perform their intended functions and achieve autonomous navigation.

To meet the growing operational demands of robots, it is essential to conduct more in-depth research on path planning algorithms for mobile robots. Zhang, H., et al. addressed the issues of low planning efficiency and susceptibility to local minima in traditional Extended Random Tree (RRT) algorithms by introducing regression mechanisms and adaptive expansion mechanisms to improve the RRT algorithm, enabling robots to quickly plan paths to target points in complex environments [16]. Janis, A., and Bade, A. demonstrated that a reliable robot navigation system must possess localization and identification capabilities as well as the ability to generate smooth path trajectories. To this end, they explored a series of path planning algorithms for virtual robots in virtual complex environments, providing valuable insights into navigation problems in complex environments [17]. de Almeida, J. P. L. S., et al. investigated path planning methods for multi-robot systems using genetic algorithms to solve multi-traveling salesman problems, effectively enhancing the robots' path planning capabilities under uncertain localization conditions [18]. Ichter, B. and Pavone, M. proposed a learning sub-space motion planning (L-SBMP) algorithm for planning the motion paths of complex robotic systems. This algorithm enables robots to generalize to complex new environments by performing global exploration of the constructed learning sub-space [19]. Dang, T. et al. studied path planning strategies for robots in complex underground environments, proposing a local planner for optimization exploration in local subspaces and a global planner for exploration across the entire spatial range [20]. Guzzi, J. et al. designed and evaluated a long-range path planning method that utilizes partial knowledge to predict the local motion of robots, applying it to complex quadruped robots and simple differential robots, demonstrating excellent path planning and obstacle avoidance performance [21]. Feng, T. et al. integrated task hazard levels and road factors into a robot path planning model and employed a hybrid adaptive genetic algorithm (HAGA) to generate efficient path planning results that align with multi-dimensional interests, ensuring the safety and efficiency of robot operation under various tasks and complex road conditions [22].

From the above analysis, it can be seen that robot path planning and obstacle avoidance behavior involve the processing and analysis of obstacle information, which is essentially the process of establishing and analyzing environmental map models. To enhance the robot's understanding of environmental information during path planning in complex environments, topological maps can be utilized as models to represent local environmental obstacle maps in space, capable of expressing both metric information and positional relationship information [23]-[26]. Topological map models employ topological graphs based on the relationships between nodes and edges connecting them for path planning, making them suitable for large-scale spatial environments and providing crucial support for robot path planning and obstacle avoidance movements [27], [28].

This paper employs a visual SLAM algorithm to construct a topological-grid hybrid map that meets the research standards. Considering that robot path planning and obstacle avoidance can be divided into static and dynamic scenarios, two algorithms were designed based on the topological-grid hybrid map: an improved A* static path planning and obstacle avoidance algorithm and a dynamic planning and obstacle avoidance algorithm based on improved TEB. In the case study analysis, the topological-grid hybrid map was simulated and analyzed, confirming that it meets the research standards. The performance of the static path planning and obstacle avoidance algorithm based on A* and the dynamic path planning and obstacle avoidance algorithm based on the improved TEB were then investigated separately. Finally, an algorithm application analysis was provided to enhance the persuasiveness of the research results.

II. Robot SLAM and Hybrid Topology Map Construction

II. A. Mobile Robot SLAM

Robots must know their position in the working environment to perform navigation tasks, which led to the development of the Simultaneous Localization and Mapping (SLAM) theory. The mathematical model of SLAM is shown in Figure 1. SLAM technology enables robots to move in the working environment and use information recorded by their onboard laser radar sensors and odometers to map the working space while determining their position in the working space [29], [30]. In most cases, robots do not have prior knowledge of their working environment map, so they must use certain algorithms to establish a reasonably reliable map and determine their own position.

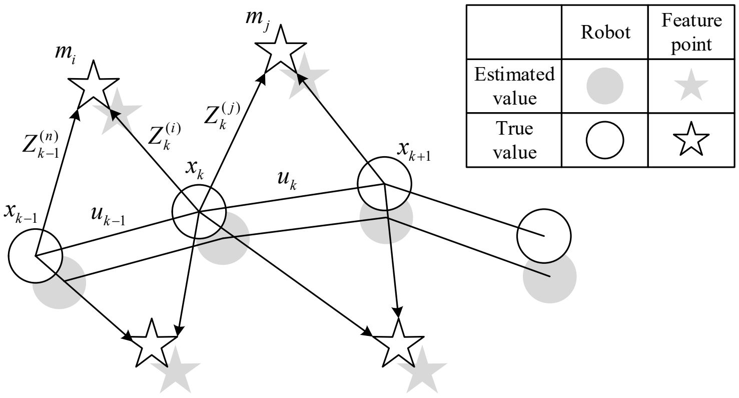

Figure 1: The mathematical model of SLAM

In the figure, $x_k = (x_k^{'}, y_k^{'}, \theta_k)^T$ represents the pose of the mobile robot at time k. The position of the mobile robot at time k is denoted by $x_k^{'}$ and $y_k^{'}$ at time k, and the pose angle at time k is denoted by θ_k. The pose of the mobile robot from time 1 to time k is denoted by $x_{1:k} = (x_1, x_2, ..., x_k)^T$ represents the pose of the mobile robot from time 1 to k. u_{k-1} represents the input control quantity at time $k-1$. The i th feature point in the environment is denoted by m_i, where $m = (m_1, m_2, ..., m_N$ represents the environment map m, and $z_k - 1^{(i)}$ and $z_k^{(i)}$ represent the observations of feature point m_i at time $k-1$ and time k, respectively. Through mathematical model transformation, the joint posterior probability of the mobile robot's pose x_k at time t and all feature points m in the environment is calculated, which can be specifically expressed as $p(x_{1:k}, m \mid z_{1:k}, u_{tk-1})$.

$$p(x_{1:k}, m \mid z_{1:k}, u_{1:k-1}) = \frac{p(z_k \mid x_k, m) p(x_k, m \mid z_{1:k-1}, u_{1:k-1})}{p(z_k \mid z_{1:k-1}, u_{1:k-1})} \tag{1}$$

II. A. 1) Vision-based SLAM

Mobile robot SLAM can use cameras to capture the surrounding environment of the mobile robot. Monocular cameras, stereo cameras, and depth cameras are three commonly used types of visual cameras. Mobile robots equipped with cameras as visual sensors require both image information and depth information when constructing their working environment. However, monocular cameras cannot directly obtain depth information and must instead use certain methods to indirectly derive it. A stereo camera system composed of two monocular cameras can utilize the differences between the images captured by the two cameras to obtain depth information. Depth cameras can directly use their internal ranging modules to quickly obtain depth information. Due to their low cost and ability to collect a large amount of data from the working environment, visual SLAM has gradually become widely adopted. However, visual SLAM also has several issues, such as being significantly affected by ambient light, particularly being unable to operate in dark environments, unstable dynamic performance, and high computational requirements.

II. A. 2) SLAM based on laser radar

When using lidar for SLAM in mobile robots, there are two main mathematical models: the beam model and the likelihood field model. Specific descriptions are as follows:

(1) Beam model

The beam model measures along each beam. Since measurements are affected by four types of error, the beam model is a mixture of four probability densities. The first type of error, represented by the Gaussian distribution p_{hit}, is noise generated during measurement due to defects in the sensor itself and the sensor's susceptibility to environmental influences. The second type of error, represented by the exponential distribution p_{short}, occurs when moving obstacles appear in the working environment during measurement, causing blockage and resulting in measured values that are smaller than the actual values. The third type of error can be represented as a point cloud distribution $p_{\max}$ centered at the maximum measurement distance. When reflections caused by surfaces like mirrors appear in the working environment, obstacles may be easily overlooked, leading to detection, and thus resulting in measurement values that are greater than the actual values. The fourth type of error can be represented as a uniform distribution p_{rand} within the sensor's measurement range, resulting from random measurement values caused by interference from different sensors. $z_{hit}, z_{short}, z_{\max}, z_{rand}$ represent the proportions of the four types of errors, respectively. Therefore, the beam model can be expressed as formula (2):

$$p(z_k \mid x_k, m) = \begin{pmatrix} z_{hit} \\ z_{short} \\ z_{\max} \\ z_{\max} \end{pmatrix}^T g \begin{pmatrix} p_{hit} \\ p_{short} \\ p_{\max} \\ p_{rand} \end{pmatrix} \tag{2}$$

(2) Likelihood field model

The likelihood field model applies Gaussian blurring to obstacles in space, then transforms the radar coordinate system to the world coordinate system using coordinate transformation formulas, and finally maps the measured endpoints to the global coordinate system according to certain rules. During this process, three types of errors may occur. The first type can be represented by a Gaussian distribution p_{hit} to describe the measurement error between the projected point of the measured point in the global coordinate system and the obstacle on the map. The second type can be represented by a point cloud distribution $p_{\max}$ to describe the absence of obstacles in the working environment that were not scanned during the measurement process. The third type can be represented by a uniform distribution p_{rand} within the sensor range, representing random measurement errors. $z_{hit}, z_{\max}, z_{rand}$

represent the weights of the three types of errors, respectively, so the likelihood field model can be expressed as formula (3):

$$p(z_k \mid x_k, m) = z_{hit} \, g \, p_{hit} + z_{\max} \, g \, p_{\max} + z_{rand} \, g \, p_{rand} \tag{3}$$

II. B. Overview of Mobile Robot Environment Mapping

II. B. 1) The Significance of Environmental Maps

An environmental map is a simplified and abstracted description of the working environment model for mobile robots, recording only key details such as the boundaries of the environment and the spatial locations of obstacles. This simplified model has achieved good results in practical applications [31], [32]. In many cases, the working environment of mobile robots often has uncertainty, making the environmental map an indispensable component for mobile robots. The importance of the environmental map lies in two main aspects: first, the environmental map serves as the foundation for the mobile robot's localization and navigation. The mobile robot collects environmental information through its onboard sensors, matches the acquired environmental information with the pre-stored map, and thereby determines its position within the environmental map. Second, the mobile robot can utilize the provided environmental map to determine obstacle-related information, enabling it to plan a feasible path from the starting point to the destination.

II. B. 2) Environmental map construction methods

Common methods for constructing environmental maps include topological mapping, visual mapping, grid mapping, and topological-grid hybrid mapping. Given the scope of this study, this paper adopts the topological-grid hybrid mapping method. The construction of the topological-grid hybrid map is not only practical but also enables the robot to utilize the environmental features it can recognize to generate topological nodes, thereby ensuring that the robot can extract the corresponding features during actual localization and avoid localization failures. Hybrid localization based on geometric-topological maps is a combination of local geometric localization and global topological localization. For specific experimental environments, when the robot enters a corridor environment from a room, it should switch to global topological localization, as corridor environments are conducive to topological localization and autonomous navigation. Conversely, when the robot moves from a corridor environment into a room, it should switch to local geometric localization, enabling precise localization to facilitate the execution of specific tasks.

III. Robot path planning and obstacle avoidance algorithm design

*III. A. Static path planning and obstacle avoidance algorithm based on improved A~**

III. A. 1) Improvements to the Distance Function

Improving the distance function can reduce this unnecessary node expansion and improve the efficiency of the algorithm. Euclidean distance is the most commonly used distance measurement method, defined as the straight-line distance between two points. In two-dimensional space, the Euclidean distance formula between two points is:

$$d(p,q) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2} \tag{4}$$

In this context, $p = (p_1, p_2)$ and $q = (q_1, q_2)$ are two points in two-dimensional space.

The advantage of the Euclidean distance lies in its intuitive accuracy and isotropy; it does not depend on the choice of coordinates or direction, and the distance is consistent in any direction. Its formula is:

$$d(p,q) = \mid q_1 - p_1 \mid + \mid q_2 - p_2 \mid \tag{5}$$

This method is simpler than Euclidean distance calculation, more computationally efficient, and suitable for grid layouts. However, in freely movable spaces, it does not reflect the shortest distance. Another common method is diagonal distance, also known as Chebyshev distance. Its formula is:

$$d(p,q) = \max(\mid q_1 - p_1 \mid, \mid q_2 - p_2 \mid) \tag{6}$$

This method can reflect the greatest distance in all possible directions, which is suitable for scenarios that need to consider extreme conditions. The calculation formula is as follows:

$$\begin{aligned} D(p,q) = D_d \times \min(\mid q_1 - p_1 \mid, \mid q_2 - p_2 \mid) + D_s \\ \times(\mid q_1 - p_1 \mid + \mid q_2 - p_2 \mid - 2 \times \min(\mid q_1 - p_1 \mid, \mid q_2 - p_2 \mid)) \end{aligned} \tag{7}$$

Among them, D_d is the cost of diagonal movement, and D_s is the cost of vertical or horizontal movement. To avoid square root operations, this paper uses the approximate values $D_d = 1.4, D_s = 1$.

III. A. 2) Dynamic Weighting Improvement

The classical form of the traditional A^* algorithm is based on $f(n)=g(n)+h(n)$, but the improved version of A^* adjusts the original $h(n)$ to $w(n)\times h(n)$, where $w(n)$ is a controllable parameter that can influence the search strategy. In the original algorithm, when $w(n)$ equals 1, the ratio of $g(n)$ to $h(n)$ remains consistent. When $w(n)$ is increased to 2, this ratio becomes 1:2, which may lead to an increase in the estimated path components. Although the search speed is significantly accelerated, it does not guarantee finding the global optimal solution, as it may deviate from the actual shortest path. As $w(n)$ decreases, the algorithm tends to seek a more precise optimal path, but search efficiency decreases accordingly. For

$$f(n)=g(n)+w(n)\times h(n) \tag{8}$$

In path planning, a balance must be struck between speed and quality. Therefore, $w(n)$ cannot be fixed but should be dynamically adjusted based on the proximity to the target node. At locations far from the target point, the weight of $w(n)$ can be moderately increased, sacrificing some optimality to reduce exploration of distant nodes, thereby shortening the overall search time. Conversely, when approaching the target, $w(n)$ should be reduced to ensure the best path is found. This dynamic adjustment aligns with the characteristics of the exponential decay function, which is used to set the weights, as expressed below:

$$w(n)=(w_{\max}-1)\cdot e^{-k\cdot\frac{d_{start}}{d_{start}+d_{goal}}}+1 \tag{9}$$

In this optimization process, $w_{\max}=3$ is selected as the initial parameter, K=5 is selected as the parameter controlling the weight decay rate, and d_{start} and d_{goal} represent the distance from the starting point to the current node and the distance from the current node to the target point, respectively, calculated using the distance function method described above. This dynamic weight adjustment strategy allows the decay rate of the weights to be controlled by adjusting the parameter k.

III. A. 3) Path reconstruction and smoothing

Bezier curves are used to smooth out broken lines through a set of control points. This article focuses on methods for smoothing paths using Bezier curves. The general formula for high-order Bezier curves is as follows:

$$B(t)=\sum_{i=0}^{n}\binom{n}{i}(1-t)^{n-i}t^{i}P_i \tag{10}$$

where t is a parameter with a value range between [0, 1]. $\frac{n}{l}$ is a combination number, representing the number of ways to select i elements from n elements. The formula means that the points on the curve are composed of control points $P_0,P_1,P_2,...,P_n$ according to the binomial coefficients.

High-order Bézier curves may exhibit poor stability when handling complex paths, causing the path to cross obstacles and resulting in suboptimal curve effects. To address this issue, multiple sets of third-order Bézier curves are used to optimize the path. The third-order Bézier curve formula is:

$$B(t)=(1-t)^3\cdot P_0+3\cdot(1-t)^2\cdot t\cdot P_1+3\cdot(1-t)\cdot t^2\cdot P_2+t^3\cdot P_3 \tag{11}$$

III. A. 4) Static path planning and obstacle avoidance implementation process

The algorithm's specific steps are as follows: Initialization phase: Set the starting node (G) and target node (E) of the graph. Initialize two dynamic sets: an open list (OPEN) to store nodes to be evaluated, and a closed list (CLOSED) to store nodes that have been evaluated. Add the starting node (G) to the open list and set it as the current node, while the closed list is initially empty. Node expansion process: Add all reachable child nodes of the current node to the open list and move the current node itself to the closed list. Search feasibility determination: Check whether the open list is empty to determine if there are any unevaluated feasible child nodes. If it is empty, it indicates that the search has no solution, and the algorithm terminates. Evaluation function application: Apply evaluation function $f(n)=g(n)+h(n)$ to calculate the evaluation values of each node in the open list. Goal Achievement Evaluation: If the current node is the goal node (E), proceed to the final stage; if not, return to the node expansion process. Path Confirmation: The current node, i.e., the goal node, is added to the closed list. Path Reconstruction and Output: Reconstruct the nodes in the closed list to reveal the complete path from the starting point to the goal point, and output it. At this point, the algorithm has achieved its search goal and terminates immediately.

III. B. Dynamic Planning and Obstacle Avoidance Algorithm Based on Improved TEB

III. B. 1) Homotopy Path Optimization

(1) Constructing a topology-grid hybrid map: As the core of real-time path planning, the method focuses on the local environment of the mobile robot's current position and the target point. A concise local map is constructed by selecting key points, both sides of obstacles, and straight lines connecting these points as edges. The specific operations include:

$$\frac{(Z_1-Z_3)(P_s-P_g)}{\|(Z_1-Z_3)\|\cdot\|(P_s-P_g)\|}>\delta \quad \delta\in[0,1] \tag{12}$$

$$\begin{cases} L=Z_1+t(Z_3-Z_1) \\ L\cap O=\varnothing \end{cases} t\in[0,1] \tag{13}$$

(2) Filtering of homotopy paths

The formula for calculating the H-signature is as follows, where p_k represents the vertices of the candidate path, o_l represents the obstacles in the local topology graph R, and $\grave{o}_l\subseteq o_l$. For

$$H(\tau)=\sum_{k=1}^{N-1}H_z(p_k,p_{k+1}) \tag{14}$$

$$H_z(p_k,p_{k+1})=\sum_{l=1}^{R}A_i(\ln(p_{k+1}-o_l)-\ln(o_l-p_k)) \tag{15}$$

$$A_i=f_0(o_i)/\left[\prod_{j=1,j\neq l}^{R}(o_i-o_j)\right] \tag{16}$$

$$\begin{aligned} H_s(p_k,p_{k+1}) &= \sum_{i=1}^{R}A_i[\ln(p_{k+1}-o_l)-\ln(o_l-p_k) \\ &+\underset{\alpha e\square}{imin}(arg(p_{k+1}-\grave{o}_l)-arg(p_k-\grave{o}_l)+2\alpha\pi\,|)] \end{aligned} \tag{17}$$

In the formula, $\arg(p_{k+1}-\grave{o}_l)$ represents the angle difference between adjacent sides. To minimize $\arg(p_{k+1}-\grave{o}_l)$, the α in the formula is typically set to a constant close to zero.

(3) Selection of the optimal path

The optimizer is fed into the parallel TEB, yielding M optimized candidate trajectory clusters $B_1^*,...,B_M^*$. Finally, the total cost of each optimized trajectory is calculated based on the cost function, and the one with the lowest cost is selected as the optimized trajectory. The formula is:

$$\begin{aligned} V(B)=w^T f(B)=\sum_{k=1}^{n-1}\{&\sigma_p\|\min(0,p_k)\|_2^2+\sigma_o\|\min(0,o_k)\|_2^2 \\ &+\sigma_v\|\min(0,v_k)\|_2^2+\sigma_\phi\|\min(0,\omega_k)\|_2^2+\sigma_\alpha\|\min(0,\alpha_k)\|_2^2 \\ &+\sigma_h\|h_k\|_2^2+\sigma_\omega\|\min(0,\omega_k)\|_2^2+\sigma_j\|\min(0,j_k)\|_2^2+\sigma_k\Delta T_k^2\} \end{aligned} \tag{18}$$

Among these, ΔT is the time cost, h_k is the kinematic cost, v_k is the velocity cost, α_k is the acceleration cost, and j_k is the cost of jerk, ω_k is the cost of yaw angular velocity, ω_k is the cost of yaw angular acceleration, o_k is the cost term for proximity to an obstacle, p_k is the cost of path length, and all σ are the normalized weighting coefficients for each cost.

III. B. 2) Shortest Distance Constraint Improvement

To solve the problem of local detours when turning during TEB path planning, a shortest distance constraint improvement is introduced. For:

$$h(n)=\sqrt{(x_G-x_n)^2+(y_G-y_n)^2} \tag{19}$$

Inspired by the above equation, in terms of calculation, a distance function is used to measure the shortest distance between adjacent pose points s_i and s_{i+n}, which is used as a constraint condition and incorporated into the hypergraph construction. The shortest distance penalty function f_{dis} forms a new hypergraph structure by connecting these constraint edges. The form of the shortest distance constraint function is as follows:

$$\begin{aligned}&f_{dis}(S_i,S_{i+n})=D_d\times\min(|x_{i+n}-x_i|,|y_{i+n}-y_i|)\\&+D_s\times(|x_{i+n}-x_i|+|y_{i+n}-y_i|\\&-2\times\min(|x_{i+n}-x_i|,|y_{i+n}-y_i|))\end{aligned} \tag{20}$$

III. B. 3) Algorithm Flow

The static planning and obstacle avoidance algorithm based on the improved TEB follows the steps below to ensure that ground unmanned vehicles operate efficiently and safely in complex environments. The process of the static planning and obstacle avoidance algorithm based on the improved TEB is shown in Figure 2.

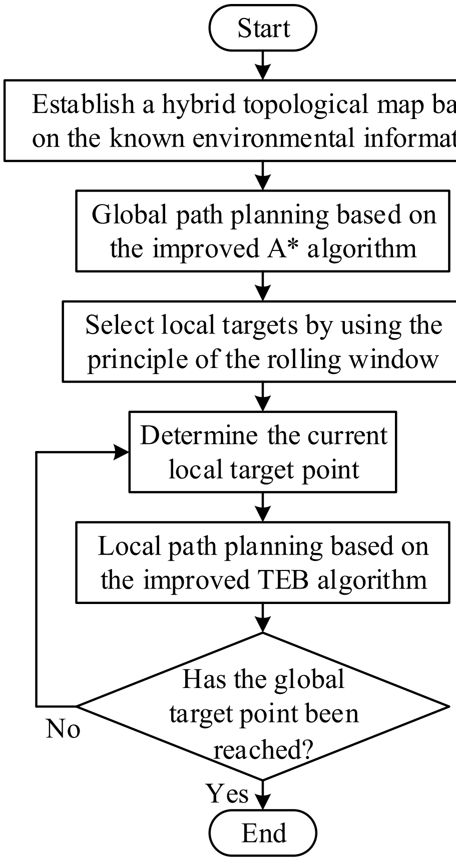
Start
Establish a hybrid topological map based on the known environmental information
Global path planning based on the improved A* algorithm
Select local targets by using the principle of the rolling window
Determine the current local target point
Local path planning based on the improved TEB algorithm
Has the global target point been reached?
No
Yes
End

Figure 2: Algorithm flowchart

IV. Robot Path Planning and Obstacle Avoidance Case Study Analysis

IV. A. Mobile Robots and Hybrid Topology Map Construction

IV. A. 1) Mobile Robots

The subject of this study is the Rocky One mobile robot, with an operational efficiency of 600 cycles per hour*1 in a 90° layout for palletizing and depalletizing. The pallet height is 2m, and the applicable vehicle dimensions are an inner height of 2.3–2.7m, an inner width greater than 2.3m, and unlimited length. The material specifications are a weight of 25kg*3 and side lengths of 200–600mm*4. The material feeding conveyor belt is a telescopic roller conveyor or folding belt conveyor, with a weight of 1,300 kg to 2,000 kg, dimensions of 1.4 m × 1 m or 1.83 m × 1.35 m, a maximum speed of 1 m/s to 2 m/s, omnidirectional mobility, and a charging voltage of three-phase 380 VAC. Battery runtime is 8 hours under maximum load and full-speed typical tasks, with a standard charging time of 2.5 hours (optional fast charging). The power supply voltage is three-phase 220VAC*6, and the operating temperature is 0–50°C.

IV. A. 2) Mixed Topology Map Construction Analysis

This section uses the SLAM algorithm to create a topology-grid hybrid map of the simulation environment. During the map creation process, no adjustments were made to the topology nodes, and no subsequent dilation-erosion processing was performed on the grid map. The topology-grid hybrid map is shown in Figure 3.

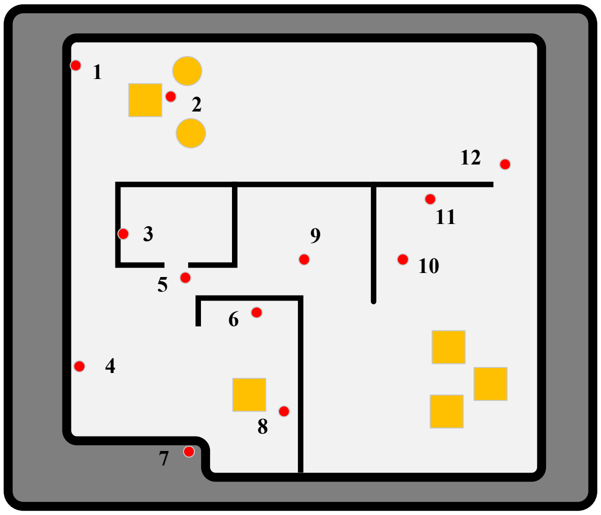

Figure 3: Unprocessed topology-raster hybrid map

As shown in Figure 3, before adjusting the topology node positions, some topology nodes were located on the walls of the grid map or on block-shaped obstacles. These topology node positions were inaccessible, preventing the robot from planning a global path in subsequent path planning tasks. Thin walls and walls with significant thickness differences reduced the robot's actual safety distance, thereby increasing the risk of the robot colliding with walls. The final map established after adjustment is shown in Figures 4–6.

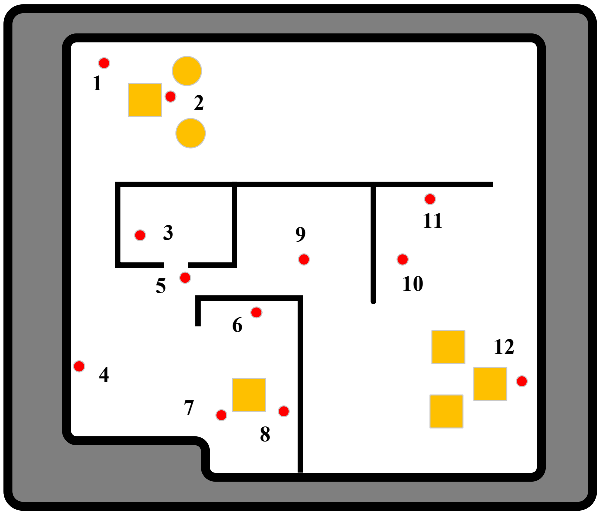

Figure 4: A 10* 10-square-meter topology-raster hybrid map

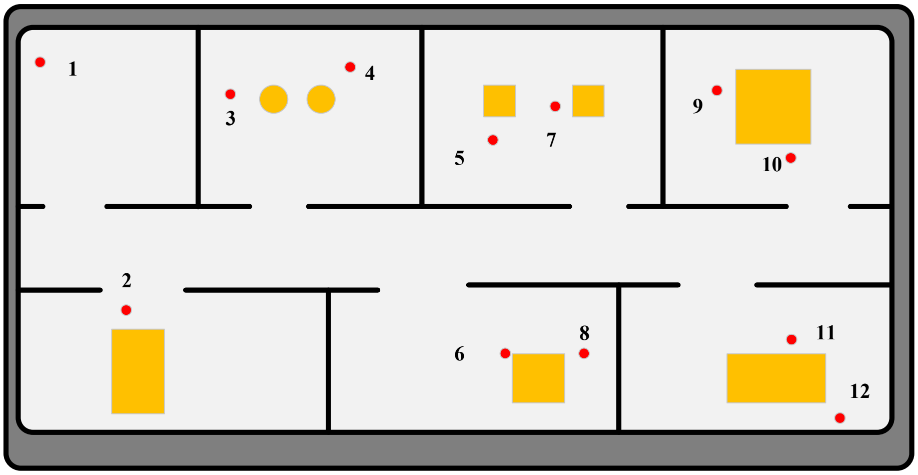

Figure 5: A 20* 10-square-meter topology-raster hybrid map

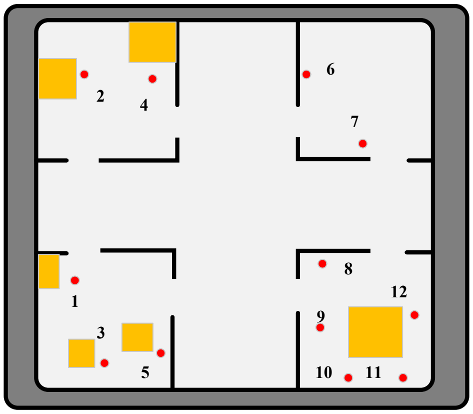

Figure 6: A 16* 16-square-meter topology-raster hybrid map

As shown in Figures 4–6, the location of each topological node in the map is near an obstacle, which represents equipment, furniture, appliances, etc. in the environment. This study believes that the more obstacles there are in the map, the more complex the environment is, and the greater the probability of uncertainty. From the simulation experiment results, it can be seen that all key areas in the map contain one or more topological nodes. These topological nodes provide the robot with an initial target point for navigation tasks. When the robot approaches a topological node, if a leak occurs in that area, the robot will promptly detect the fault and make reasonable path planning. Therefore, the topological-grid hybrid map established using the SLAM algorithm can meet the requirements of subsequent tasks.

After the topological-grid hybrid map is created, each topological node is assigned an initial position danger coefficient, which represents the danger level of the environment surrounding the topological node. In this section, the danger level of a topological node is determined by the area of obstacles surrounding it. The formula for the initial position danger coefficient is given in Equation (21):

$$\text{in}\,it_{score}\left(p_i\right)=s_{obstaclae}\left(p_i,r\right) \tag{21}$$

In the equation, p_i represents a topological node, and $S_{obstacle}(p_i,r)$ denotes the area of the circular region centered at the topological node p_i with radius r that is occupied by obstacles. In a topology-grid hybrid map, each topological node has a fixed initial position score, which does not change over time and is inherent to each topological node. When an initial position score is relatively high, it indicates that the environment surrounding the topological node is relatively complex, and the robot should first move to the vicinity of this topological node to explore whether there are any obstacles and eliminate the danger associated with this topological node.

In addition to calculating the initial position score of topological nodes to determine which topological node to move toward, the robot must also consider time factors. The unexplored time danger coefficient is used to account for the impact of time on topological node scores, indicating that the score of each topological node increases as time progresses during the exploration process. This algorithm uses two functions to calculate the unexplored time score, as shown in Formula (22):

$$\begin{cases} time_{score1} = 2^{t} \\ time_{score2} = t^{2} \end{cases} \tag{22}$$

In the formula, t represents time. The exponential function is used because if a topological node is not traversed for a long time, the score for the time not traversed for that topological node will increase rapidly, and at this point, the initial position score will no longer dominate the robot's traversal target. This avoids the problem of not being able to reach a topological node with a relatively small initial position score for a long time.

When the robot first enters a pre-existing map, it first calculates the initial position score for each topological node using formula (21), To prevent the robot from determining the first topological node to reach based solely on the initial position score, which could result in prematurely heading to a distant area, the robot first obtains its current position upon starting operation, then calculates the Euclidean distance to each topological node using formula (23). After obtaining all distances, they are sorted from largest to smallest. For:

$$\cos t_{i} = \sqrt{\left(robot_{\text{in}\,itx} - p_{ix}\right)^{2} + \left(robot_{\text{in}\,ity} - p_{iy}\right)^{2}} \tag{23}$$

In the equation, $robot_{\text{in}\,itx}$ and $robot_{\text{in}\,ity}$ are the initial position coordinates x、y of the robot, respectively, and p_{ix} and p_{iy} are the x、y coordinates of the i th topological node.

After sorting the robots, the distance score for each topological node is calculated using formula (24), and then the distance score and initial position score are added together to obtain the danger level of the topological node. It is:

$$\cos tcore_{i} = \frac{i}{topo_{number}} * \text{in}\, it_{score}\left(p_{i}\right) \tag{24}$$

In the formula, i is the order of topological nodes after sorting from smallest to largest, $topo_{number}$ is the total number of topological nodes, nit_{score} calculates $costcore_{i}$ and then uses formula (25) to calculate the danger level of the topological node. It is:

$$topo\,\text{int}_{score}\left(i\right) = \text{in}\, it_{score}\left(p_{i}\right) + \cos tcore_{i} \tag{25}$$

After calculating the danger level, the coordinates of the top node with the highest danger level are sent to the robot. The robot then uses the algorithm described in this paper to plan a global path and moves toward this top node. During movement, the robot continuously calculates the distance between its current position and each topological node. The algorithm specifies that if, while the robot is moving toward a topological node, another topological node is less than 0.5m away from the robot, the robot will first reach that topological node. After reaching that topological node, the robot continues to move toward the original highest-risk topological node. After numbering the map, the danger level of each topological node is calculated using formulas (21) to (25). The results of the topological node danger levels are shown in Table 1. From Table 1, it can be seen that the initial position score of topological node 6 is the highest, indicating that the complexity of the environment surrounding this topological node and the combined score of the distance between the robot and the topological node are the highest. This topological node is the first navigation target point for the robot.

Table 1: Topological node risk

Topological node	Topological node risk	Topological node	Topological node risk
1	153.29	7	0
2	157.25	8	152.12
3	217.25	9	126.33
4	155.25	10	157.08
5	167.51	11	91.54
6	230.33	12	49.42

IV. B. Robot path planning and obstacle avoidance algorithm verification analysis

As can be seen from the above analysis, this paper uses visual SLAM algorithms to construct a topology-grid hybrid map that meets the research requirements. On this basis, this section will verify and analyze the static path planning and obstacle avoidance algorithm based on the improved A* algorithm and the dynamic planning and obstacle avoidance algorithm based on the improved TEB algorithm. The details are as follows:

IV. B. 1) Static path planning and obstacle avoidance

Based on the topology-grid hybrid map constructed in Section 4.1.1, the mobile robot is placed at the starting position. Using the "2D Pose Estimate" function in the Rviz visualization interface, the initial coordinates and initial heading angle of the mobile robot are set. Then, using the "2D Nav Goal" function, the terminal coordinates and terminal heading angle of the mobile robot are set, and the mobile robot will perform path planning. To ensure the accuracy of the experimental data, 100 experiments were conducted. The static path planning experimental data are shown in Table 2. Based on the data size in the table, it can be seen that in the same static environment, the improved algorithm proposed in this paper (success rate: 99.00%, average path length: 3.415 m, average time: 16.35 s) outperforms the traditional algorithm (success rate: 87.00%, average path length: 3.735 m, average time: 20.05 s) in terms of navigation success rate, average path length, and average time, demonstrating the effectiveness of the improved A* static path planning and obstacle avoidance algorithm.

Table 2: Static path planning experimental data

Experimental environment	Number of experiments	Number of successes	Success rate	Average path length/m	Average time consumption/s
Traditional algorithm	100	87	87.00%	3.735	20.05
Ours algorithm	100	99	99.00%	3.415	16.35

IV. B. 2) Dynamic path planning and obstacle avoidance

To validate the feasibility of the improved TEB dynamic planning and obstacle avoidance algorithm proposed in Section 3.2 of this paper, the improved algorithm was tested in a static environment with moving pedestrians. During the mobile robot's movement toward the target point, moving obstacles appear at a certain speed along the pre-planned path. The mobile robot uses its onboard laser radar sensors to obtain information about the moving obstacles and re-performs local path planning to avoid them. To ensure the accuracy of the experimental data, 100 experiments were conducted. The results of the dynamic path planning and obstacle avoidance analysis are shown in Table 3. Based on the data in Table 3, it can be concluded that the improved TEB dynamic planning and obstacle avoidance algorithm achieves a 14% higher navigation success rate and shorter execution time when handling dynamic obstacles compared to traditional algorithms, thereby demonstrating the effectiveness of the improved TEB dynamic planning and obstacle avoidance algorithm.

Table 3: Experimental data of dynamic path planning

Experimental environment	Number of experiments	Number of successes	Success rate	Average path length/m	Average time consumption/s
Traditional algorithm	100	82	82.00%	4.305	30.22
Ours algorithm	100	96	96.00%	4.005	21.15

IV. B. 3) Application Analysis

As analyzed earlier, the static path planning and obstacle avoidance algorithm based on the improved A* algorithm and the dynamic path planning and obstacle avoidance algorithm based on the improved TEB algorithm demonstrate excellent performance. To validate the practical application effectiveness of the static path planning and obstacle avoidance algorithm based on the improved A* algorithm and the dynamic path planning and obstacle avoidance algorithm based on the improved TEB algorithm, a map containing a narrow channel was selected for testing. The dimensions of the narrow channel environment are 2000m × 1000m × 600m, with the starting and ending coordinates set to (100, 300, 100) and (1900, 560, 520), respectively. The average data obtained from 100 path planning runs in the narrow environment using the algorithms in this paper are shown in Table 4. The results indicate that the static path planning and obstacle avoidance algorithm based on the improved A* and the dynamic path planning and obstacle avoidance algorithm based on the improved TEB are comparable in terms of navigation success rate, average path length, and average time, and both meet the experimental standards, confirming the practical effectiveness of the two algorithms in this paper.

Table 4: Apply the test analysis results

Experimental environment	Number of experiments	Number of successes	Success rate	Average path length/m	Average time consumption /s
Improve the static path planning and obstacle avoidance algorithm of A~*	100	97	97.00%	100.21	232.41
Improve the dynamic programming and obstacle avoidance algorithm of TEB	100	98	98.00%	110.33	236.54

V. Conclusion

Robot path planning and obstacle avoidance ensure that mobile robots can operate safely and smoothly while improving their efficiency. This paper addresses the path planning and obstacle avoidance issues of mobile robots by proposing a study on the construction and application of hybrid topological maps for robot path planning and obstacle avoidance. First, a hybrid topological-grid map is constructed using visual SLAM algorithms. Based on this, two scenarios—static and dynamic—are considered. Accordingly, two algorithms are designed: a static path planning and obstacle avoidance algorithm based on an improved A* algorithm and a dynamic path planning and obstacle avoidance algorithm based on an improved TEB algorithm. Under the support of research equipment and data, both algorithms are validated and analyzed. In static scenarios, compared to traditional algorithms (success rate: 87.00%, average path length: 3.735m, average time: 20.05s), the proposed algorithm demonstrates higher performance (success rate: 99.00%, average path length: 3.415m, average time: 16.35s). Additionally, in dynamic scenarios, similar results were observed, fully validating the practical application effectiveness of the two path planning and obstacle avoidance algorithms.

Funding

This work was supported by State Grid Shanxi Electric Power Company Marketing Service Center (No: 52051L250008).

References

[1] Tzafestas, S. G. (2018). Mobile robot control and navigation: A global overview. Journal of Intelligent & Robotic Systems, 91, 35-58.
[2] Gao, X., Li, J., Fan, L., Zhou, Q., Yin, K., Wang, J., ... & Wang, Z. (2018). Review of wheeled mobile robots' navigation problems and application prospects in agriculture. Ieee Access, 6, 49248-49268.
[3] Cognominal, M., Patronymic, K., & Wańkowicz, A. (2021). Evolving field of autonomous mobile robotics: Technological advances and applications. Fusion of Multidisciplinary Research, An International Journal, 2(2), 189-200.
[4] Chen, X., Gan, Y., & Xiong, S. (2024). Optimization of Mobile Robot Delivery System Based on Deep Learning. Journal of Computer Science Research, 6(4), 51-65.
[5] Lee, J., Park, G., Cho, I., Kang, K., Pyo, D., Cho, S., ... & Chung, W. (2022). Ods-bot: Mobile robot navigation for outdoor delivery services. IEEE Access, 10, 107250-107258.
[6] Ciabattoni, L., Ferracuti, F., Foresi, G., Freddi, A., Monteriù, A., & Pagnotta, D. P. (2017, September). Real-time fall detection system by using mobile robots in smart homes. In 2017 IEEE 7th International Conference on Consumer Electronics-Berlin (ICCE-Berlin) (pp. 15-16). IEEE.
[7] Vanus, J., Hercik, R., & Bilik, P. (2023). Using interoperability between mobile robot and KNX technology for occupancy monitoring in smart home care. Sensors, 23(21), 8953.
[8] Liu, L., Wang, X., Yang, X., Liu, H., Li, J., & Wang, P. (2023). Path planning techniques for mobile robots: Review and prospect. Expert Systems with Applications, 227, 120254.
[9] Tazir, M. L., Azouaoui, O., Hazerchi, M., & Brahimi, M. (2015, July). Mobile robot path planning for complex dynamic environments. In 2015 International Conference on Advanced Robotics (ICAR) (pp. 200-206). IEEE.
[10] Wang, X., Ma, X., & Li, Z. (2023). Research on SLAM and path planning method of inspection robot in complex scenarios. Electronics, 12(10), 2178.
[11] Lu, Y., & Da, C. (2025). Global and local path planning of robots combining ACO and dynamic window algorithm. Scientific Reports, 15(1), 9452.
[12] Hitz, G., Galceran, E., Garneau, M. È., Pomerleau, F., & Siegwart, R. (2017). Adaptive continuous-space informative path planning for online environmental monitoring. Journal of Field Robotics, 34(8), 1427-1449.
[13] Galceran, E., & Carreras, M. (2013). A survey on coverage path planning for robotics. Robotics and Autonomous systems, 61(12), 1258-1276.
[14] Han, J., & Seo, Y. (2017). Mobile robot path planning with surrounding point set and path improvement. Applied Soft Computing, 57, 35-47.
[15] Chen, B., Gong, L., Yu, C., Du, X., Chen, J., Xie, S., ... & Liu, C. (2023). Workspace decomposition based path planning for fruit-picking robot in complex greenhouse environment. Computers and Electronics in Agriculture, 215, 108353.
[16] Zhang, H., Wang, Y., Zheng, J., & Yu, J. (2018). Path planning of industrial robot based on improved RRT algorithm in complex environments. IEEE Access, 6, 53296-53306.
[17] Janis, A., & Bade, A. (2016). Path planning algorithm in complex environment: a survey. Transactions on Science and Technology, 3(1), 31-40.

[18] de Almeida, J. P. L. S., Nakashima, R. T., Neves-Jr, F., & de Arruda, L. V. R. (2020). A global/local path planner for multi-robot systems with uncertain robot localization. Journal of Intelligent & Robotic Systems, 100, 311-333.

[19] Ichter, B., & Pavone, M. (2019). Robot motion planning in learned latent spaces. IEEE Robotics and Automation Letters, 4(3), 2407-2414.

[20] Dang, T., Khattak, S., Mascarich, F., & Alexis, K. (2019, December). Explore locally, plan globally: A path planning framework for autonomous robotic exploration in subterranean environments. In 2019 19th International Conference on Advanced Robotics (ICAR) (pp. 9-16). IEEE.

[21] Guzzi, J., Chavez-Garcia, R. O., Nava, M., Gambardella, L. M., & Giusti, A. (2020). Path planning with local motion estimations. IEEE Robotics and Automation Letters, 5(2), 2586-2593.

[22] Feng, T., Li, J., Jiang, H., Yang, S. X., Wang, P., Teng, Y., ... & Luo, B. (2024). The optimal global path planning of mobile robot based on improved hybrid adaptive genetic algorithm in different tasks and complex road environments. IEEE Access, 12, 18400-18415.

[23] Luo, R. C., & Shih, W. (2018, June). Autonomous mobile robot intrinsic navigation based on visual topological map. In 2018 IEEE 27th International Symposium on Industrial Electronics (ISIE) (pp. 541-546). IEEE.

[24] Kuric, I., Bulej, V., Saga, M., & Pokorny, P. (2017). Development of simulation software for mobile robot path planning within multilayer map system based on metric and topological maps. International journal of advanced robotic systems, 14(6), 1729881417743029.

[25] Zhang, Y., Tian, G., Shao, X., Liu, S., Zhang, M., & Duan, P. (2021). Building metric-topological map to efficient object search for mobile robot. IEEE Transactions on Industrial Electronics, 69(7), 7076-7087.

[26] Niijima, S., Umeyama, R., Sasaki, Y., & Mizoguchi, H. (2020, October). City-scale grid-topological hybrid maps for autonomous mobile robot navigation in urban area. In 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 2065-2071). IEEE.

[27] Tang, L., Wang, Y., Ding, X., Yin, H., Xiong, R., & Huang, S. (2019). Topological local-metric framework for mobile robots navigation: a long term perspective. Autonomous Robots, 43, 197-211.

[28] Gunathillake, A., Huang, H., & Savkin, A. V. (2019). Sensor-network-based navigation of a mobile robot for extremum seeking using a topology map. IEEE Transactions on industrial informatics, 15(7), 3962-3972.

[29] Shiyu Zheng. (2025). A Review of Navigation and SLAM Technologies in Orchard Environments. Asian Research Journal of Agriculture,18(2),13-21.

[30] Shaokai Shen & Jintao Meng. (2025). A Review of Autonomous Navigation Technology for Orchard Robots Based on Visual SLAM. Asian Research Journal of Agriculture,18(1),261-271.

[31] Zirun Li,Chunjia Zhu,Hao Tao,Qingfang Zhang,Chenyu Pu & Jinsheng Xiao. (2025). Robot map construction based on topology and hierarchical fusion in unknown environment. Journal of Physics: Conference Series,2999(1),012022-012022.

[32] Valeria Sarno,Elisa Stefanini,Giorgio Grioli & Lucia Pallottino. (2025). Mobile Robots for Environment-Aware Navigation: A Code-Free Approach with Topometric Maps for Non-Expert Users. Robotics,14(2),19-19.