

International Journal of Internet Protocol Technology

ISSN online: 1743-8217 - ISSN print: 1743-8209

<https://www.inderscience.com/ijipt>

Intelligent robot path planning based on multimodal deep learning algorithm

Linyan Pan

DOI: [10.1504/IJIPT.2025.10071615](https://doi.org/10.1504/IJIPT.2025.10071615)

Article History:

Received:	14 February 2025
Last revised:	24 March 2025
Accepted:	30 March 2025
Published online:	10 July 2025

Intelligent robot path planning based on multimodal deep learning algorithm

Linyan Pan

Intelligent Manufacturing College,
Anhui Wenda University of Information Engineering,
Hefei, Anhui, China
Email: panjiaojiao99@126.com

Abstract: In order to overcome the problems of poor performance, low obstacle avoidance success rate, and long time in traditional intelligent robot path planning methods, this paper proposes an intelligent robot path planning method based on multimodal deep learning algorithm. By using depth cameras, LiDAR, and IMU to collect multimodal data, and utilising multimodal deep learning algorithms to determine the characteristics of the collected signal and image data, intelligent robot obstacle target detection can be achieved. By combining the gravity and repulsion functions of the artificial potential field method to improve the RRT Connect algorithm, a smooth path trajectory is obtained by smoothing the inflection point of the path through a cubic B-spline curve. The experimental results show that the proposed method has high smoothness and low complexity of intelligent robot paths, shorter paths, an average obstacle avoidance success rate of 97.08%, and an average path planning time of 74.25 ms.

Keywords: multimodal deep learning algorithm; intelligent robots; path planning; multimodal data; improved RRT-connect algorithm.

Reference to this paper should be made as follows: Pan, L. (2025) 'Intelligent robot path planning based on multimodal deep learning algorithm', *Int. J. Internet Protocol Technology*, Vol. 18, No. 1, pp.1–14.

Biographical notes: Linyan Pan received a Master's degree in Control Engineering from Hefei University of Technology in 2019 and is now an Associate Professor in the School of Intelligent Manufacturing of Anhui Wenda University of Information Engineering. The main research interests include intelligent manufacturing, automatic control, artificial intelligence and so on.

1 Introduction

With the rapid development of artificial intelligence technology, the application of intelligent robots in various fields is becoming increasingly widespread, and therefore they have gradually become an indispensable part of modern society (Shen et al., 2024; Xing et al., 2024). As one of the core issues in intelligent robot technology, path planning is of great importance. In real environments, robots need to face various obstacles, dynamically changing environmental conditions and uncertain external disturbances, all of which pose severe challenges to path planning algorithms (Kumar and Sikander, 2024; Dhouib, 2024). By implementing precise path planning strategies, robots can optimise their travel routes, reduce energy consumption, improve task completion efficiency and ensure safer and more stable operation processes. Therefore, research on intelligent robot path planning not only has theoretical value, but also has broad practical value.

Ma and Xue (2024) proposed a robot path planning method based on deep attention Q-network. In order to address the limitations of traditional Deep Q-Networks (DQN) in processing Partially Observable Markov Decision Processes (POMDP) due to the lack of memory components, an improved DQN algorithm that integrates attention mechanisms is introduced. This algorithm can effectively utilise and deeply mine perceptual information in historical data. Based on the Artificial Potential Field (APF) theory, a novel reward mechanism has been designed to guide the robot's movement distance and direction, thereby further improving the accuracy of path planning. However, after testing, it was found that the intelligent robot path planning effect of this method is not good, and there is a significant gap between it and the research objectives. Wang et al. (2024) proposed a robot path planning method based on the beetle algorithm and DWA algorithm. Using chaotic mapping to initialise the beetle algorithm population, adaptive t -distribution perturbation strategy and spiral search strategy are integrated into the algorithm after initialisation to

significantly enhance the overall performance of the algorithm and determine the global optimal path. In order to track the global optimal path and achieve dynamic obstacle avoidance, an Improved Dynamic Window method (IDWA) was adopted. Three improvement measures have been implemented to address the collision risk and linear velocity fluctuations that may occur during the operation of traditional dynamic window methods: Firstly, by combining global path planning algorithms, the algorithm effectively avoids getting stuck in local optima; Secondly, set the initial heading angle of the robot reasonably to reduce path redundancy; Finally, a new linear velocity change evaluation function is added to ensure smooth changes in linear velocity. We combined these two optimised algorithms and designed a path planning algorithm that includes both dynamic and static environments. However, after testing, it was found that the obstacle avoidance success rate of this method is not high. Xue et al. (2024) proposed a robot path planning method based on an improved ant colony algorithm. Guiding candidate nodes closer to the line connecting the starting and ending points through the designed guidance heuristic function helps to avoid local optima. By incorporating the Cauchy distribution function, the influence of the heuristic function is gradually weakened, thereby enhancing the algorithm's global exploration ability in the later stages. The distance heuristic function has been optimised to accelerate the convergence process of the algorithm. After the initial determination of the robot's movement path, the path is smoothed to achieve robot path planning. However, the complexity of this algorithm is relatively high, resulting in an increase in robot path planning time.

In order to solve the problems of poor performance in intelligent robot path planning, low obstacle avoidance success rate and long path planning time in the above methods, an intelligent robot path planning method based on multimodal deep learning algorithm is proposed. The technical route of this article is as follows:

- 1) Firstly, by utilising various high-precision sensors such as depth cameras, LiDAR, IMU, etc., multi-dimensional and high-precision data collection of the surrounding environment was achieved, providing rich environmental information for subsequent path planning. On this basis, data from different sensors are effectively fused through multi-mode deep learning algorithm, which improves the accuracy and robustness of obstacle target detection and lays a solid foundation for path planning.
- 2) Based on the obstacle target detection results, the RRT Connect algorithm is improved by integrating the concept of artificial potential field method. The attractive function is used to improve the efficiency of path search, while the repulsive function avoids local minima. At the same time, the pruning function is used to remove redundant nodes, and the cubic B-spline curve is used to smooth the path, thereby generating efficient and smooth path trajectories.
- 3) The effectiveness of this method was validated by selecting the effectiveness of intelligent robot path

planning, obstacle avoidance success rate and path planning time as indicators.

2 Design of path planning method for intelligent robots

2.1 Multimodal data acquisition

- (1) *Depth camera model*: Visual SLAM captures image information through the use of cameras, which can map three-dimensional scenes in space onto a two-dimensional image plane, thereby inputting perceived information in the form of images. Compared to other sensors, cameras have more accurate image information acquisition capabilities, which can provide more reliable data support for intelligent robot path planning (Wang et al., 2022; Mu et al., 2021).

- a) *World coordinate system to camera coordinate system*: Given that point P_w has coordinates of $[X_w Y_w Z_w]^T$ in the world coordinate system and coordinates of $P_c = [X_c Y_c Z_c]^T$ in the camera coordinate system. The matrix relationship between the two is:

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = T \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (1)$$

In the above formula, R represents the rotation matrix, T represents the camera extrinsic matrix and t represents the translation vector.

- b) *Camera coordinate system to image coordinate system*: Given that coordinate $P_c = [X_c Y_c Z_c]^T$ in the camera coordinate system corresponds to the image coordinate system, its coordinates are $P_i = [x y]^T$. According to the similarity triangle theorem, the matrix relationship between the two is obtained as follows:

$$Z_c \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \quad (2)$$

- c) *Image coordinate system to pixel coordinate system*: Point $P_i = [x y]^T$ on the image coordinate system can be transformed into the pixel coordinate system by scaling up and translating, denoted as $P = [u v]^T$. Assuming that it has been scaled up $[\alpha, \beta]$ times and translated a distance of $[a b]^T$ on the basis of the image coordinate system, the following relationship holds:

$$\begin{cases} u = \alpha x + a \\ v = \beta y + b \end{cases} \quad (3)$$

Organise formula (3) into homogeneous coordinate form, and the result is as follows:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha & 0 & a \\ 0 & \beta & b \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (4)$$

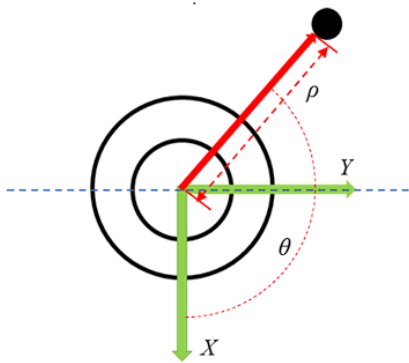
- d) *World coordinate system to pixel coordinate system*: By comprehensively organising the above transformations, the transformation relationship between the world and the pixel coordinate system is obtained (Rath et al., 2021), and the following relationship holds:

$$Z_c \cdot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & a & 0 \\ 0 & f_y & b & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = KT \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (5)$$

In the above formula, f_x and f_y , respectively represent the number of pixels of focal length f in the image coordinate system and K represents the camera internal parameter matrix.

- (2) *Lidar model*: Lidar ranging equipment emits laser, which reflects off the surface of an object and forms an image that falls onto a CCD sensor. When an object moves, the position of the light spot on the CCD changes accordingly. By measuring the displacement of the light spot, the distance between the object and the baseline can be accurately calculated. Owing to the different coordinate systems of each sensor, it is necessary to first clarify the radar coordinate system and then perform coordinate transformation to unify it (Qiu and Li, 2024), as shown in Figure 1.

Figure 1 Coordinate system of Lidar (see online version for colours)



Single line LiDAR typically outputs laser data in the form of vector sets, with each vector represented as (ρ, θ) . ρ represents the distance from the radar coordinate origin to the target point, and θ represents the orientation of the target

object. The observation model of the laser radar can be obtained by using the data obtained by the laser radar in the intelligent robot system. This model can be expressed in the form of conditional probability distribution. The specific calculation formula is as follows:

$$P(Z_k | X_k, M) = \prod_i P(z_k^i | X_k, M) \quad (6)$$

In the above formula, Z_k represents the laser observation value at time k , X_k represents the pose of the intelligent robot at time k and M represents the map.

- (3) *IMU measurement model*: IMU is usually affected by bias and Gaussian white noise, which can cause drift over time, resulting in inaccurate pose calculations (Chen et al., 2023). To solve this problem, a random walk model can be used to model Bias. Based on the knowledge of intelligent robot dynamics, the IMU measurement model is represented by the following formula:

$$\tilde{\omega}^B = \omega^B + b_\omega + n_\omega \quad (7)$$

$$\tilde{a}^B = (R_B^W)^T (a^W + g^W) + b_a + n_a \quad (8)$$

In the above formula, B represents the robot and IMU coordinate system, W represents the world coordinate system, $\tilde{\omega}^B$ and $\tilde{a}^B$ represent the angular velocity and linear acceleration measurements at B , and ω^B represents the instantaneous angular velocity at B . a^W and g^W represents the linear acceleration and gravitational acceleration under W , n_ω and n_a represent the Gaussian white noise of gyroscope and accelerometer, b_ω and b_a represent the bias of gyroscope and acceleration and R_B^W represents the coordinate transformation matrix.

Integrating data from various sensors such as depth cameras, LiDAR and Inertial Measurement Units (IMUs) to provide robots with more comprehensive environmental perception capabilities. At the data collection level, these sensors capture data from their respective fields simultaneously or almost simultaneously. To ensure effective fusion of these data, timestamp matching and coordinate system transformation become key steps. The matching of timestamps is achieved through hardware synchronisation and software synchronisation techniques. Hardware synchronisation uses specialised synchronisation devices to ensure that the timestamps of each sensor are completely consistent during data collection, while software synchronisation relies on algorithms to calibrate and interpolate the captured data timestamps to eliminate time differences caused by different sampling rates. In addition, because different sensors may collect data based on different coordinate systems, coordinate system conversion is also required before data fusion, that is, data from all sensors are converted to the same global coordinate system to ensure spatial consistency of data. Typically, complex mathematical

transformations such as rotation and translation are used to ensure that data from different modalities can be fused within a unified framework. Before inputting data into deep learning models, a series of pre-processing steps are required, including data cleaning, format conversion, normalisation, etc., to improve the quality and accuracy of the data. The normalisation formula is:

$$y = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad (9)$$

In the above formula, x represents the original data, $x_{\min}$ represents the minimum value in the data column, $x_{\max}$ represents the maximum value in the data column and y represents the normalised data.

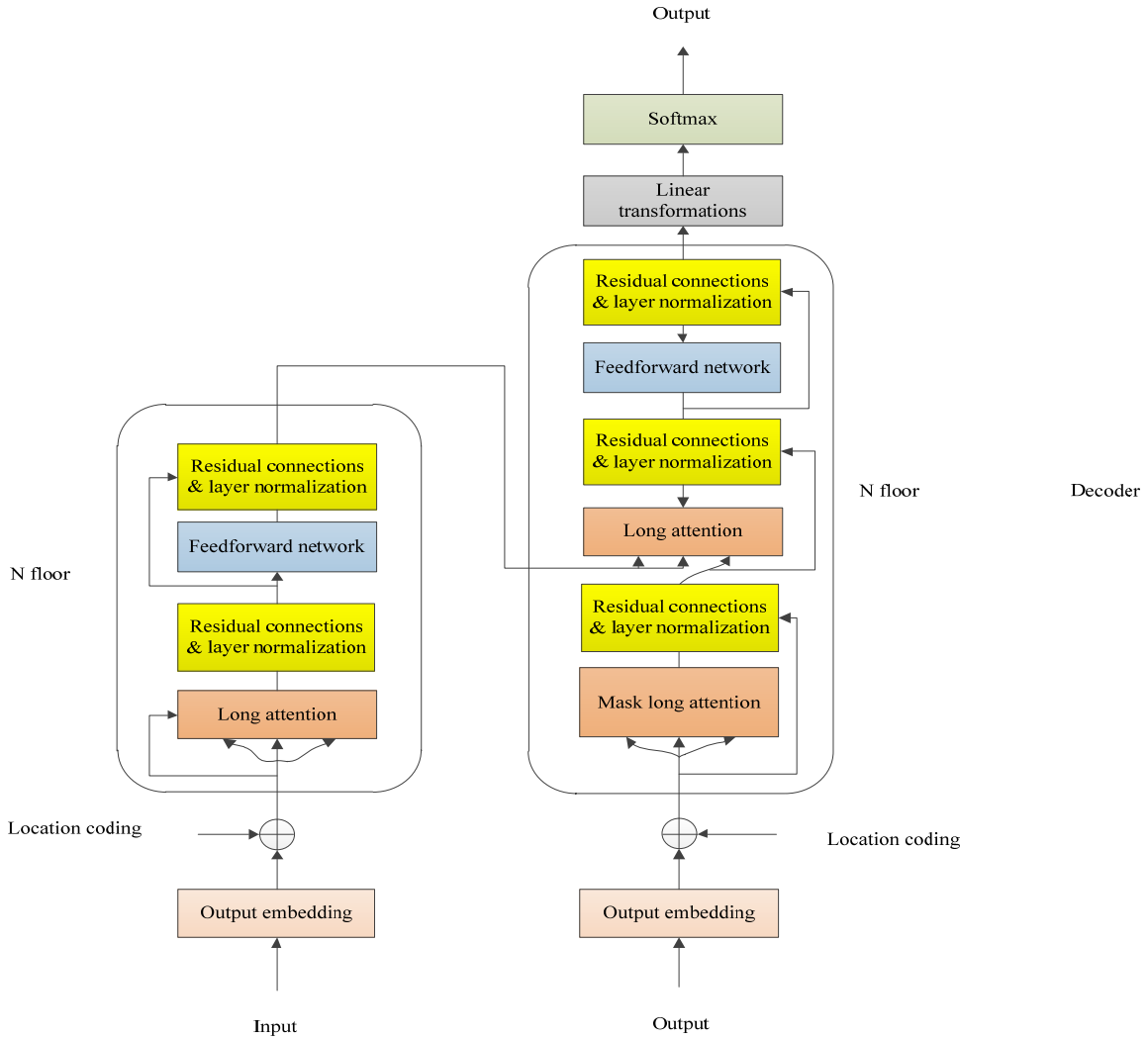
Through pre-processing, multimodal data fusion can better support intelligent robots in obstacle target detection, improving the accuracy and robustness of path planning.

2.2 Intelligent robot obstacle target detection based on multimodal deep learning

Multimodal depth learning can fuse data from different sensors, such as depth camera, laser radar, IMU, etc. These data have unique advantages and limitations (Huang et al., 2024). Through multimodal data fusion, intelligent robots can perceive the environment more comprehensively, improving the accuracy and robustness of obstacle target detection. Accurate obstacle target detection is the foundation of path planning. Only by correctly identifying and locating obstacles can intelligent robots plan safe and efficient paths. Obstacle target detection based on multimodal deep learning can provide more accurate environmental information, thereby improving the accuracy of path planning.

- 1) *Transformer module*: Transformer is mainly composed of encoder and decoder, and its network structure is shown in Figure 2.

Figure 2 Transformer module structure (see online version for colours)



Observing Figure 2, it can be observed that the Transformer module adopts a multi head self-attention strategy, which is an extension of the traditional self-attention mechanism. It distributes the original attention mechanism to multiple different sub-spaces for processing. In these subspaces, each independently learns its own attention weight to capture and parse semantic content from different dimensions in the input data. Specifically, the multi head self-attention mechanism first maps the input sequence data to multiple parallel sub-spaces through linear transformation, and then performs self-attention calculations separately in each sub-space. After completing these independent calculations, merge the results of each sub-space and obtain the final output result through a linear transformation. The entire calculation process is as follows:

$$\begin{cases} MultiHead(Q, K, V) = Concat(head_1, ..., head_h) W^O \\ head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \\ Attention(Q, K, V) = soft\ max\left(\frac{QK^T}{\sqrt{d_k}}\right)V \end{cases} \quad (10)$$

In the above formula, Q, K, V represents the result of projecting the input vector onto three different vector spaces:

Query, Key and Value. $head_1, ..., head_h$ represents different headers, and d_k represents the dimension of the Key vector.

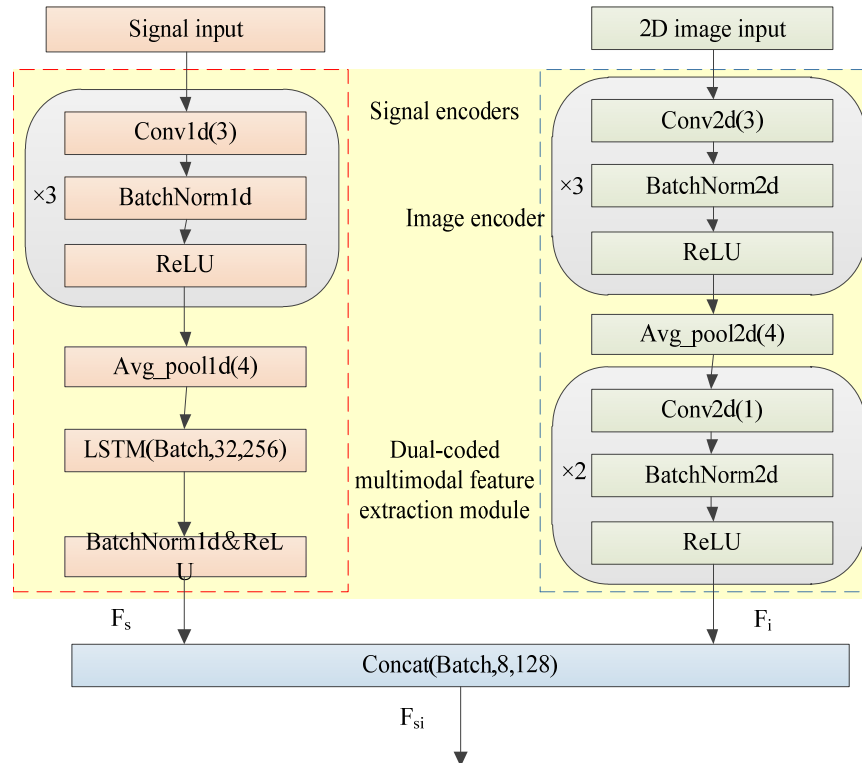
2) *Dual encoding multimodal feature extraction module:*
The dual encoding multimodal feature extraction module is shown in Figure 3.

As shown in Figure 3, the dual encoding multimodal feature extraction module consists of two sub-networks: a signal encoder and an image encoder. Input the signal data from the enhanced training set to the Signal Encoder (SE). Considering that the key features of the signal can be captured in local variations, CNN is chosen for feature encoding. In addition, Long-Short-Term Memory (LSTM) networks can extract long-term dependency features when processing time-series data. Therefore, a LSTM layer is connected after a convolutional structure for feature encoding, and an average pooling layer is set between the convolutional block and the LSTM layer to reduce the feature dimension. Finally, output signal feature F_s , whose calculation formula is as follows:

$$F_s = SE(S; \Theta_1) \quad (11)$$

In the above formula, SE represents the signal encoder, Θ_1 represents the parameters and S represents the one-dimensional signal.

Figure 3 Dual encoding multimodal feature extraction module (see online version for colours)



Similarly, the image data from the enhanced training set is fed into an Image Encoder (IE). When deciding to use CNN for feature encoding, considering that image data often has higher dimensions, compared to signal data, two 2D convolutional layers with 1×1 convolution kernels were selected to replace the LSTM layer, aiming to effectively reduce the feature dimension. This processing step can output image feature F_i , and its specific calculation method can be described by the following formula:

$$F_i = IE(I; \Theta_2) \quad (12)$$

In the above formula, $IE(\cdot)$ represents the image encoder, Θ_2 represents the parameters and I represents the two-dimensional image.

Adjust the signal feature F_s and image feature F_i to the same size, and obtain the multimodal fusion feature $F_{si} = Cat(F_s + F_i)$ through concatenation processing $Cat(\cdot)$. Combining classifier F_c with multimodal fusion feature F_{si} for robot obstacle target detection, the calculation formula is as follows:

$$\hat{y} = F_c(F_{si}; \Theta_c) \quad (13)$$

In the above formula, Θ_c represents the parameter of classifier F_c .

The dual encoding multimodal feature extraction module is a comprehensive processing framework that combines signal encoders and image encoders to achieve accurate detection of robot obstacle targets. In this module, signal data (such as sound, vibration, etc.) is first input into the signal encoder, which captures local features through Convolutional Neural Network (CNN), and extracts long-term dependent features in time series data with the help of Long-Short-Term Memory network (LSTM). In order to reduce the feature dimension, an average pooling layer is set between the convolutional block and the LSTM layer. After these processes, the signal encoder finally outputs refined signal features. At the same time, the image data is input into an image encoder, which utilises two two-dimensional convolutional layers with 1×1 convolution kernels to effectively reduce the dimensionality of image features while preserving the key information of the image. In the feature alignment stage, signal features and image features are adjusted to the same dimension, and then through concatenation processing, the features of these two modalities are fused into a unified multimodal feature vector. This vector contains both the temporal and local features of the signal, as well as the visual features of the image, providing rich information for subsequent robot obstacle target detection. Finally, the multimodal fusion features are input into a classifier, which is a fully connected neural network classification model that outputs the detection results of obstacle targets based on the input feature vectors. This

process not only improves the accuracy of robot obstacle detection, but also enhances its adaptability to complex environments.

2.3 Implementation of intelligent robot path planning based on improved RRT-Connect algorithm

In the process of intelligent robot path planning (Tran and Lin, 2023; Li et al., 2023), in order to accelerate the search process of RRT-Connect algorithm in path planning tasks, the concept of human potential field method is innovatively integrated, especially by integrating the functions of gravitational and repulsive effects, to enhance the algorithm performance. Here, the gravity function intelligently guides the random tree to rapidly and effectively expand towards the target point. Improved the purposefulness of path search, thereby significantly reducing planning time; introduce a repulsive function and dynamically adjust the step size to further accelerate the search speed.

- 1) *Gravity function guides node direction*: By incorporating the gravity function from the artificial potential field method into the RRT-Connect algorithm, it provides a goal-oriented effect on nodes to guide them towards target point x_{goal} for expansion. This method does not require global sampling and can shorten planning time. The expansion direction of nodes has been modified from random expansion, ensuring the optimality of the planned path and avoiding falling into local minima, greatly enhancing the algorithm's ability for intelligent robot path planning. The extension function of the n -th node extended from starting point x_{init} is represented by the following formula:

$$F(n) = R(n) + G(n) \quad (14)$$

In the above formula, $R(n)$ represents the random growth function of the node, and $G(n)$ represents the target gravity function. The formula for calculating the attraction between the target node and the nearest node is as follows:

$$G = k_a \cdot x_{goal} - x_{near} \quad (15)$$

In the above formula, k_a represents the gravitational coefficient, and x_{goal} and x_{near} , respectively represent the random sampling point and the target point.

Therefore, the calculation formula for the target gravity function at the n -th node is as follows:

$$G(n) = \lambda \cdot k_a \cdot \frac{x_{goal} - x_{near}}{x_{goal} - x_{near}} \quad (16)$$

In addition, the following equation is derived from the construction equation of new node x_{new} in the RRT algorithm:

$$R(n) = \lambda \cdot \frac{x_{rand} - x_{near}}{x_{rand} - x_{near}} \quad (17)$$

The result obtained by substituting formulas (16) and (17) into formula (14) is as follows:

$$F(n) = \lambda \cdot \frac{x_{rand} - x_{near}}{x_{rand} - x_{near}} + \lambda \cdot k_a \cdot \frac{x_{goal} - x_{near}}{x_{goal} - x_{near}} \quad (18)$$

After adding the gravity function, the calculation formula for the new node is as follows:

$$x_{new} = x_{near} + \lambda \cdot \frac{x_{rand} - x_{near}}{x_{rand} - x_{near}} + \lambda \cdot k_a \cdot \frac{x_{goal} - x_{near}}{x_{goal} - x_{near}} \quad (19)$$

- 2) *Improvement strategy for dynamic adjustment of step size*: Add an improved repulsive function at each obstacle in the workspace, with the maximum repulsive force at the centre of the obstacle. The improved repulsive function can be expressed as:

$$U_{rep} = \begin{cases} \frac{1}{2} k_r \left(\frac{1}{d} - \frac{1}{d_0} \right)^2 (x_i - x_g)^2, & d \leq d_0 \\ 0, & d > d_0 \end{cases} \quad (20)$$

In the above formula, k_r represents the repulsive constant, x_i represents the coordinates of the intelligent robot in Cartesian space, x_g represents the target point, d represents the shortest distance between the obstacle and the intelligent robot and d_0 represents the influence range of the repulsive potential field.

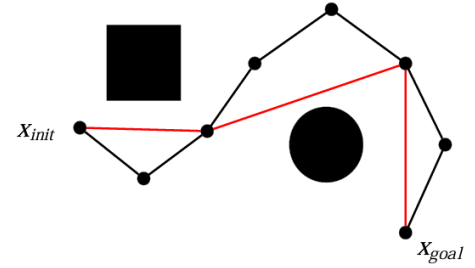
When the improved RRT-Connect algorithm generates a new node x_{new} , it first verifies whether it is within the established repulsive field range. If it is within the repulsive field range d_0 , the repulsive function is activated, the step size is reduced, and the improved step size is used for expansion; If it is not within the scope of the repulsive field, the step size with added target attraction is used for expansion, followed by collision detection under the condition of $x_{new} - x_{goal} < d$. If no collision occurs, the generated new node x_{new} is added to the random expansion tree, and then the next iteration of expansion is carried out. The specific changes are as follows:

$$s = \begin{cases} k_r \frac{\lambda}{|F_r|}, & d < d_0 \\ \lambda_1, & d \geq d_0 \end{cases} \quad (21)$$

In the above formula, s represents the improved step size, and $|F_r|$ is a scalar for repulsive force F_r .

- 3) *Implementation of path planning*: Owing to the generation of many redundant nodes in the path planning of intelligent robots, pruning functions are used to delete the redundant nodes generated in the path. The specific principle of pruning functions is as follows: firstly, using the RRT-Connect algorithm improved by the artificial potential field method proposed in this paper, a Path is planned from the initial point x_{init} to the target point x_{goal} . In this path, some redundant and useless path nodes will be generated. Starting from the starting node of the path, the subsequent path nodes will be connected in sequence one by one, while ensuring that the connecting lines between adjacent nodes can avoid all obstacles and avoid any collisions. Then, other path nodes between these two path nodes can be deleted. Repeat this process until a collision occurs. At this point, the path node where the collision point is located is considered the new starting point. Repeat the above operation until the target point is reached. The schematic diagram of pruning processing is shown in Figure 4.

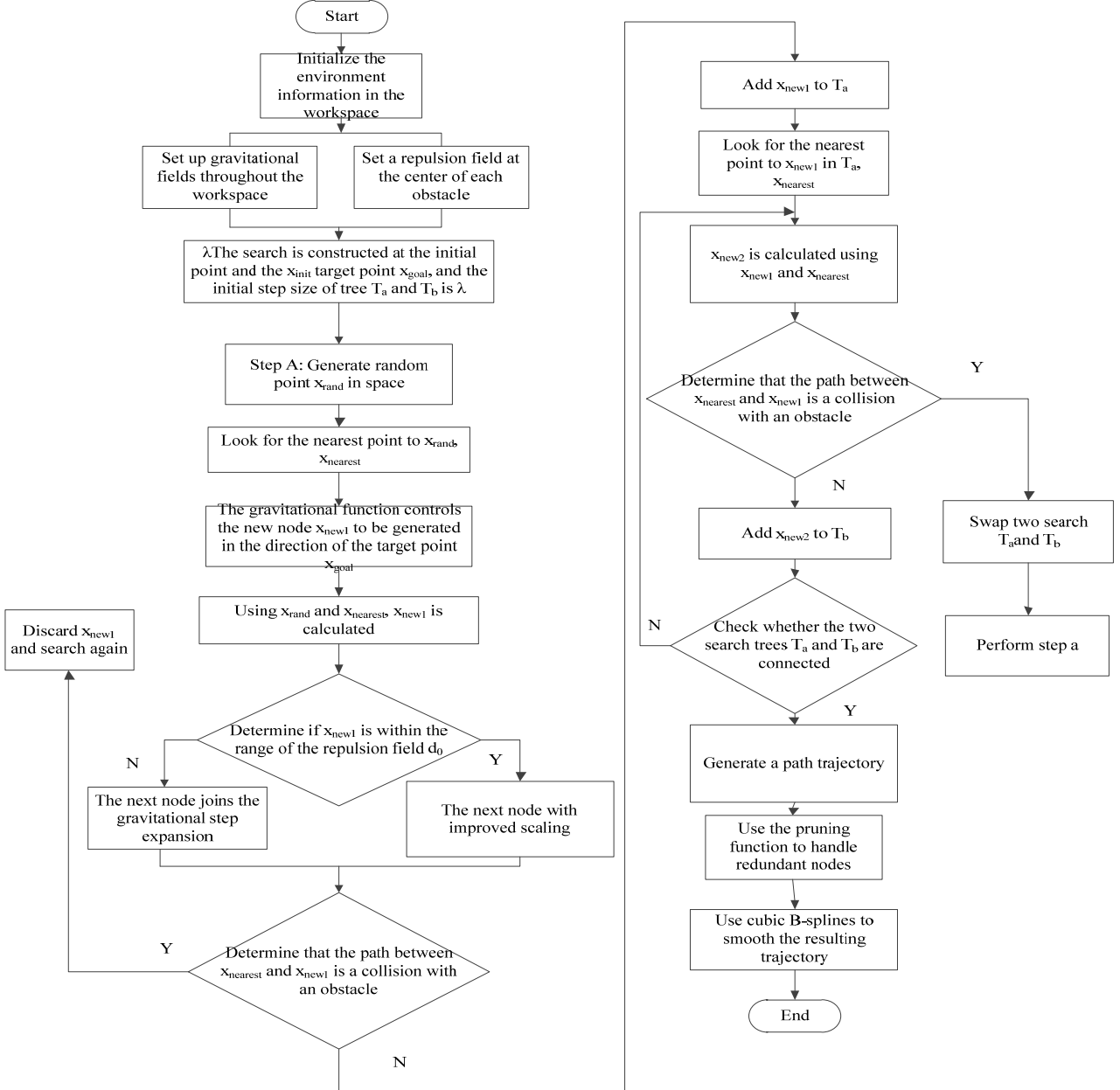
Figure 4 Schematic diagram of pruning processing (see online version for colours)



By extracting the coordinates of the intelligent robot path nodes processed by the pruning function, and using cubic B-spline curves to smooth the various turning points in the generated intelligent robot path, a smooth path trajectory is finally obtained. When the control point d_i is given, that is, the turning points in the path are known, the cubic B-spline curve equation is expressed by the following formula:

$$P_{0.3}(u) = d_0 N_{0.3}(u) + d_1 N_{1.3}(u) + d_2 N_{2.3}(u) + d_3 N_{3.3}(u) \quad (22)$$

In the above formula, $N_{0.3}(u)$, $N_{1.3}(u)$, $N_{2.3}(u)$ and $N_{3.3}(u)$ represent different B-spline basis functions, while d_0 , d_1 , d_2 and d_3 represent different control points. The path planning process for intelligent robots is shown in Figure 5.

Figure 5 Intelligent robot path planning process

3 Experiment

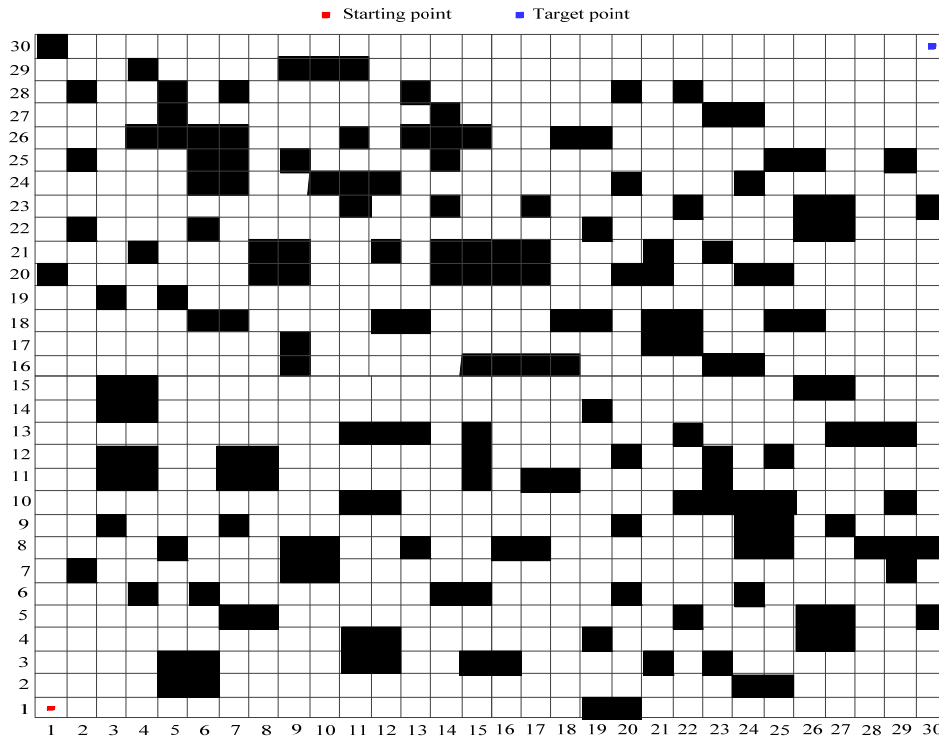
In order to verify the practical application effect of intelligent robot path planning based on multimodal deep learning algorithm, experimental tests were conducted. The specific experimental plan is as follows:

- 1) *Experimental subjects and equipment:* The ICR15005-SV intelligent robot model was selected as the research object, and its technical parameters are shown in Table 1.

During this experiment, the grid map of the target points and obstacles is shown in Figure 6.

Table 1 Technical parameters of intelligent robots

Technical parameter	Parameter description
Dead weight	120 kg
Chassis load	150 kg
Mechanical arm load	3–12 kg
Dimensions (excluding robotic arm)	800 mm×560 mm×450 mm
Navigation accuracy	±5 mm
Repetitive positioning accuracy	±0.02 mm– ±0.05 mm
Drive mode	Two wheel differential
Climbing ability	≤5°
Maximum speed	≤1.4m/s
Battery capacity	48v/35Ah
Battery life	8 h

Figure 6 Grid map (see online version for colours)

In the experiment, the robot is equipped with a depth camera, LiDAR and inertial measurement unit. Among them, the depth camera model is Intel RealSense D435, with a resolution of 1280×720 pixels, a frame rate of 30 fps and a depth range of 0.3 to 8 m, used to capture depth and colour image information of the environment; The laser radar model is Velodyne VLP-16, with a scanning frequency of 10 Hz, a horizontal scanning range of 360° and a vertical field of view angle selected according to requirements, providing three-dimensional point cloud data; The IMU model of the inertial measurement unit is MPU6050, and the range of the accelerometer and gyroscope is selected based on the robot's movement speed. The data update rate is at least 100 Hz, and it is responsible for real-time monitoring of the robot's attitude and acceleration. In order to support the efficient operation and experimental verification of deep learning

models, high-performance computers and embedded computing units were selected as the computing platforms. The high-performance computer is equipped with Intel Core i7 multi-core CPU, which has powerful computing power of 8 cores and 16 threads, as well as NVIDIA GeForce RTX series high-performance GPU. At the same time, the computer is also equipped with 64 GB DDR4 memory and 1TB SSD solid state drive. The embedded computing unit has chosen NVIDIA Jetson AGX Xavier, which integrates an 8-core processor based on NVIDIA Carmel ARM architecture and a 512 core GPU based on Volta architecture, providing high-performance computing capabilities at low power consumption. It is also equipped with 16 GB LPDDR4x memory and 32 GB eMMC flash memory, as well as multiple interfaces such as USB, Ethernet, HDMI, etc., making it easy to connect sensors, displays and other external devices. Both

computing platforms run the Ubuntu Linux operating system and support the installation and operation of deep learning frameworks such as TensorFlow and PyTorch, as well as the Robot Operating System (ROS). This enables the collection, processing, feature extraction and testing of multimodal data and path planning algorithms.

- 2) *Experimental process*: Firstly, use depth cameras, LiDAR and IMU sensors to collect data in diverse environments, and perform synchronisation, denoising, calibration and manual annotation. In the model training and validation phase, a deep learning model was constructed that includes a Transformer module and a dual encoding multimodal feature extraction module. Cross-validation strategy was adopted, and hyperparameters were adjusted through grid search or random search methods to optimise model performance. Meanwhile, the improved RRT Connect path planning algorithm was implemented in ROS environment and its performance was tested in a simulated environment.

In the process of training deep learning models, a multimodal data set is first constructed, which includes depth camera images, LiDAR point clouds and IMU data. This ensures that different lighting conditions, obstacle types and layouts are covered to improve the model's generalisation ability. The data is manually annotated accurately, completely, and consistently. Next, Convolutional Neural Networks (CNN) are used to extract features from deep camera images, and CNN or point cloud neural networks (PointNet) are used to process LiDAR point cloud data, while considering point cloud sparsity and irregularity for pre-processing and enhancement. For IMU data, models such as Recurrent Neural Networks (RNN) or Long Short-Term Memory networks (LSTM) are used for feature extraction and temporal modelling. In addition, the model training process is optimised by gradually reducing the learning rate, selecting appropriate batch sizes and iteration times, and adjusting hyperparameters such as network layers, number of neurons and regularisation coefficients. The

entire training process includes data pre-processing and partitioning, model construction and initialisation, model training, validation, testing and application, ensuring the effectiveness and practicality of the model in different scenarios and conditions. Finally, the trained model is integrated into the intelligent robot system for practical application testing.

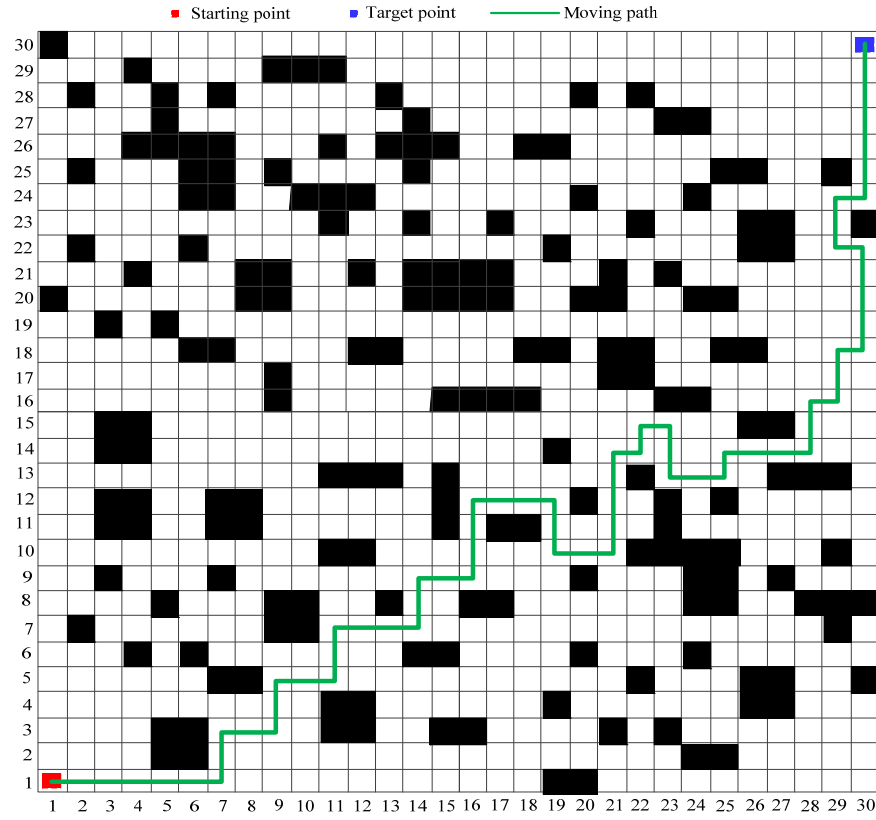
- 3) *Experimental data*: In the process of intelligent robot path planning experiments, various types of data will be involved. These data are crucial for robots to understand the environment and make path decisions. The specific types of experimental data are shown in Table 2.
- 4) *Evaluation indicators*: Select Ma et al. (2024) method, Wang et al. (2024) method and the proposed method as experimental comparison methods, and verify the practical application effects of different methods by comparing their intelligent robot path planning effects, obstacle avoidance success rates, and intelligent robot path planning time. The performance index of intelligent robot path planning refers to the ability to successfully reach the target point after using different methods for robot path planning. The success rate of obstacle avoidance for intelligent robots is a key indicator of their ability to avoid obstacles in dynamic or static environments. It is crucial for evaluating the safety, reliability and overall performance of robots. The obstacle avoidance success rate refers to the ratio of the number of times an intelligent robot successfully avoids obstacles and completes tasks (such as reaching the target location) to the total number of attempts in a series of experimental or practical scenarios. This indicator intuitively reflects the effective response capability of the robot obstacle avoidance system in the face of various potential collision risks.

3.2 Experimental result

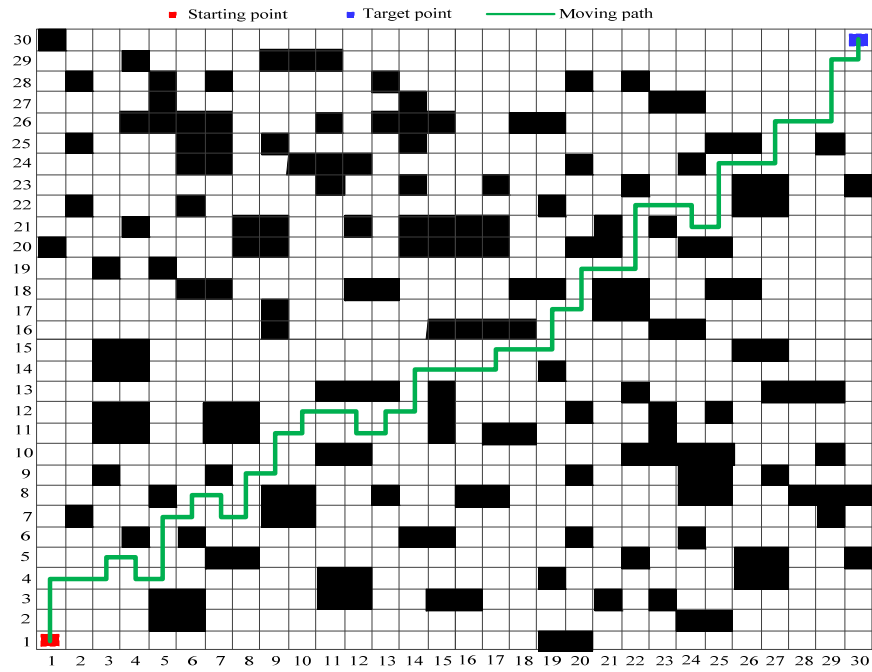
The path planning effects of the three methods for intelligent robots are shown in Figure 7.

Table 2 Experimental data types

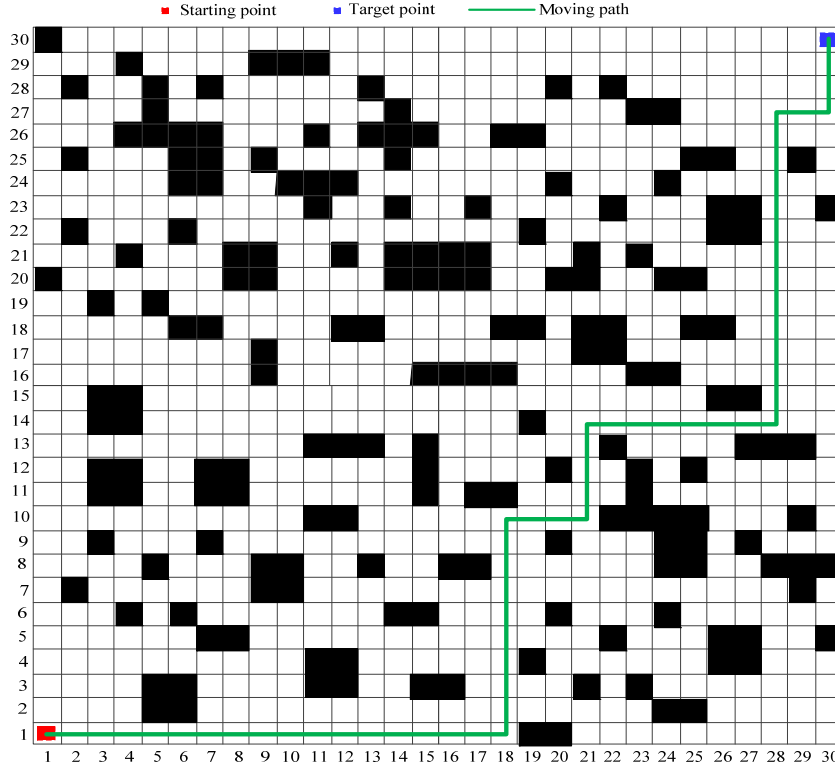
<i>Data type</i>	<i>Data composition</i>	<i>Descriptives</i>
Environmental data	Map data	The environmental map of the robot typically includes information on the location, shape and size of obstacles
	Obstacle data	Specific location, shape, size and other detailed information of obstacles on the map
Robot state data	Location data	The current position of the robot on the map is usually represented in coordinate form
	Speed data	The current movement speed of the robot, including linear velocity and angular velocity, etc.
	Acceleration data	Acceleration information of robots during movement
	Attitude data	The current orientation or posture of the robot
Sensor data	Distance sensor data	Data provided by laser radar, ultrasonic sensors, etc.
	Visual sensor data	Image data provided by visual sensors such as cameras
	Other sensor data	Data provided by inertial sensors, gyroscopes, etc.
Path planning data	Path point data	The path points generated by the robot during the path planning process, including the starting point, ending point, and intermediate path points
	Path cost data	Cost information related to each path point or segment, such as distance cost, time cost or energy cost.

Figure 7 The path planning effectiveness of three methods for intelligent robots (see online version for colours)

(a) Ma et al. (2024) method



(b) Wang et al. (2024) method

Figure 7 The path planning effectiveness of three methods for intelligent robots (see online version for colours) (continued)

(c) Proposed method

According to the analysis of the results in Figure 7, it can be seen that after using Ma et al. (2024) method and Wang et al. (2024) method for intelligent robot path planning, the smoothness of the robot's movement path is not high, and the path is relatively long and complex. Therefore, the path planning effect of these two methods is not good. The proposed method has high smoothness and low complexity in the path planning of intelligent robots, with no collision problems and short paths. This indicates that the intelligent robot path planning effect of this method is good and can be further applied in practice.

The success rate test results of intelligent robot obstacle avoidance are shown in Table 3.

Table 3 Test results of intelligent robot obstacle avoidance success rate

Number of experiments	Success rate of obstacle avoidance/%		
	Ma et al. (2024) method	Wang et al. (2024) method	Proposed method
10	79.66	80.14	98.74
20	78.47	87.66	95.48
30	71.23	85.23	96.37
40	80.63	84.75	97.52
50	81.36	81.26	96.44
60	80.69	84.78	96.18
70	78.55	86.33	97.52
80	79.31	84.65	98.36
Average value	78.74	84.35	97.08

According to the analysis of the data in Table 3, the maximum obstacle avoidance success rate of the intelligent robot using Ma et al. (2024) method is 81.36%, the average is 78.74%, and the minimum is 71.23%. The maximum success rate of obstacle avoidance for intelligent robots using Wang et al. (2024) method is 87.66%, with an average of 84.35% and a minimum of 80.14%. The maximum success rate of obstacle avoidance for the proposed method's intelligent robot is 98.74%, the average is 97.08% and the minimum is 95.48%. After comparison, it can be seen that the proposed method has the highest success rate in obstacle avoidance for intelligent robots, indicating that this method can ensure that intelligent robots can bypass obstacles and thus ensure their own safety.

The test results of intelligent robot path planning time are shown in Table 4.

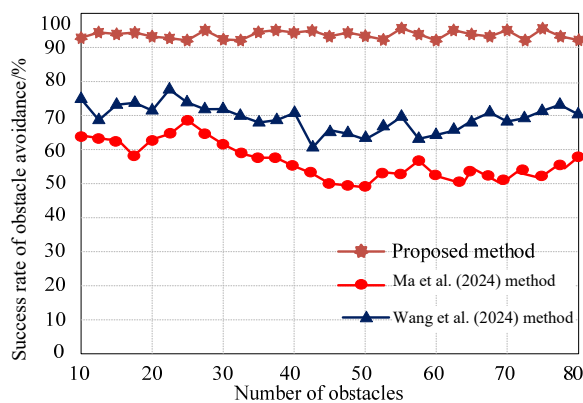
Table 4 Test results of intelligent robot path planning time

Number of experiments	Planning time/ms		
	Ma et al. (2024) method	Wang et al. (2024) method	Proposed method
10	296.35	125.89	68.97
20	214.78	126.91	84.74
30	255.63	147.08	75.63
40	247.18	125.84	78.15
50	256.33	116.32	79.63
60	247.82	166.51	80.36
70	255.64	164.78	56.89
80	262.21	113.47	69.62
Average value	254.49	135.85	74.25

According to the analysis of the data in Table 4, the maximum path planning time for the intelligent robot using Ma et al. (2024) method is 296.35 ms, the average is 254.49 ms and the minimum is 214.78 ms; The maximum path planning time for the intelligent robot using Wang et al. (2024) method is 166.51 ms, with an average of 135.85 ms and a minimum of 113.47 ms; The maximum path planning time of the proposed method for intelligent robots is 80.36 ms, the average is 74.25 ms and the minimum is 56.89 ms. After comparison, it can be seen that the proposed method has shorter path planning time and higher execution efficiency for intelligent robots.

To further validate the scalability of the proposed method, the obstacle avoidance success rates of different methods were compared under different numbers of obstacles. The experimental results are shown in Figure 8.

Figure 8 The success rate of obstacle avoidance using different methods with varying numbers of obstacles (see online version for colours)



As shown in Figure 8, with the increase of obstacles, the obstacle avoidance success rate of the comparative method decreases, but the obstacle avoidance success rate of the proposed method remains above 90%, indicating that the proposed method can adapt to different scenarios and has practical applicability.

4 Conclusion

In order to improve the quality of intelligent robot path planning, an intelligent robot path planning method based on multimodal deep learning algorithm is studied. The experimental results show that the proposed method has good performance in intelligent robot path planning, with an average obstacle avoidance success rate of 97.08% and an average path planning time of 74.25 ms. The actual application effect is good. This article studies the ability of robots to autonomously decide their travel paths, improving their ability to perform tasks in complex environments such as material handling and warehouse cargo sorting in industrial production. Meanwhile, by selecting the optimal path and adapting to environmental changes, the robot's work efficiency and safety have been significantly optimised, which can be further applied in practice. Although the

proposed method has achieved certain research results, there are also certain shortcomings. For example, although multimodal deep learning improves the accuracy and robustness of obstacle target detection, the detection accuracy may be affected in complex environments, especially when there are occlusions or similar objects. Therefore, developing more efficient obstacle target detection algorithms to improve detection accuracy and robustness in complex environments. We can consider utilising techniques such as 3D reconstruction and attention mechanisms in deep learning to better handle occlusion and similar objects.

Acknowledgements

This work was supported by the Key Natural Science Research Project of Anhui Wenda University of Information Engineering in 2023: Research on Path Planning and Control Strategy of Intelligent Robot Based on Deep Learning Algorithm (No. XZR2023A06) and the Domestic Visiting and Training Project for Young Backbone Teachers in Anhui Province in 2023 (No. JNFX2023092).

Declarations

All authors declare that they have no conflicts of interest.

References

- Chen, Z., Wang, X. and Cheng, Y. et al. (2023) 'Virtual reality combination experimental platform of path planning for autonomous mobile robot based on ROS', *Experimental Technology and Management*, Vol. 40, No. 1, pp.77–82,100.
- Dhouib, S. (2024) 'Intelligent path planning for cognitive mobile robot based on Dhouib-Matrix-SPP method', *Cognitive Robotics*, Vol. 4, No. 1, pp.62–73.
- Huang, Y., Wang, H. and Han, L. et al. (2024) 'Robot path planning in narrow passages based on improved PRM method', *Intelligent Service Robotics*, Vol. 17, No. 3, pp.609–620.
- Kumar, S. and Sikander, A. (2024) 'A novel hybrid framework for single and multi-robot path planning in a complex industrial environment', *Journal of Intelligent Manufacturing*, Vol. 35, No. 2, pp.587–612.
- Li, J, Zhao, B, Xu, F. (2023) 'Dynamic path planning of intelligent robot in power equipment maintenance environment', *Energy Reports*, Vol. 9, No. S8, pp.784–791.
- Ma, H.J. and Xue, A.H. (2024) 'Research on robot path planning based on deep attention Q-networks', *Transducer and Microsystem Technologies*, Vol. 43, No. 12, pp.66–70,75.
- Mu, X., Liu, Y. and Guo, L. et al. (2021) 'Intelligent reflecting surface enhanced indoor robot path planning: a radio map based approach', *IEEE Transactions on Wireless Communications*, pp.1–10.
- Qiu, G. and Li, J. (2024) 'Path planning for unified scheduling of multi-robot based on BSO algorithm', *Journal of Circuits, Systems and Computers*, Vol. 33, No. 7, pp.1–10.
- Rath, A.K., Parhi, D.R. and Das, H.C. et al. (2021) 'Design of a hybrid controller using genetic algorithm and neural network for path planning of a humanoid robot', *International Journal of Intelligent Unmanned Systems*, Vol. 9, No. 3, pp.169–177.

- Shen, G.C., Liu, J. and Ding, Y.L., et al. (2024) 'Continuous path planning for multi-robot in intelligent warehouse', *International Journal of Simulation Modelling (IJSIMM)*, Vol. 23, No. 2, pp.15–26.
- Tran, C.C. and Lin, C.Y. (2023) 'An intelligent path planning of welding robot based on multisensor interaction', *IEEE Sensors Journal*, Vol. 23, No. 8, pp.8591–8604.
- Wang, H.Q., Song, G.Z. and Chao, S. et al. (2024) 'Dynamic path planning of robot based on dung beetle algorithm and DWA algorithm', *Manufacturing Technology and Machine Tool*, Vol. 12, No. 1, pp.21–29.
- Wang, Y., Mai, Y. and Wen, W. (2022) 'Research on path planning algorithm of intelligent inspection robot', *Journal of Physics: Conference Series*, Vol. 2181, No. 1, pp.012006–012026.
- Xing, S., Fan, P. and Ma, X. et al. (2024) 'Research on robot path planning by integrating state-based decision-making A* algorithm and inertial dynamic window approach', *Intelligent Service Robotics*, Vol. 17, No. 4, pp.901–914.
- Xue, X., Zhu, Q.X. and Zhu, Y.H. (2024) 'Robot path planning based on improved ant colony optimization', *Journal of Xi'an Polytechnic University*, Vol. 38, No. 6, pp.59–66.