Research on Key Technologies of Smart City Building Interior Decoration Construction based on In-Depth Learning

Li Zhang[1], Aimin Qin[2*]
Information Technology and Creative Design College, Qingyuan Polytechnic, Qingyuan, China[1]
College of Architecture and Civil Engineering, West Anhui University, Lu'an, China[2]

***Abstract*—The intelligentization of building interior decoration construction is of great significance to the construction of smart city, and robot automation has brought an opportunity for this. Robot self-decoration is the development trend in the future. One of the key issues involved, is the self-planning of mobile path. In this regard, the research adopts the proximal policy optimization algorithms (PPO) to improve the self-planning path ability of the decoration robot. For the information of lidar and robot status, the Full Connect Neural Network (FCNN) is used to process it. In addition, the reward function and the corresponding Credit Assignment Problem (CAP) model are designed, to accelerate the learning process of path planning. Aiming at the dynamic uncertainty in the actual environment, the adaptive loss function is used to build an auxiliary model to predict the environmental change. The simulation results show that the research and design strategy significantly improves the learning efficiency and path planning success rate of the decoration robot, and shows good adaptability to the dynamic environment, which has important reference significance for the practical application of the decoration robot.**

Keywords*—Interior decoration; path planning; deep reinforcement learning; reward function; credit allocation

I. Introduction

Smart city is the universal vision of urban development in various countries. It uses modern technologies such as the Internet, big data and the Internet of things to solve urban governance problems, involving many fields such as architecture, transportation, medical treatment and smart home [1]. Intelligent robot is indispensable in the comprehensive environment of smart city [2]. In smart city architecture, the building scale is gradually expanding, but the number of labor is decreasing. Building construction automation has become an inevitable trend to improve production efficiency and quality. Robot technology provides new conditions and foundation for this [3]. On the whole, the development of construction robots is still at an early stage. The main robots that have been put into use at home and abroad are wall construction robots, decoration robots, facade maintenance robots, 3D printing robots, etc. [4-6]. The decoration robot has realized the plastering, painting, door and window installation and other processes of interior decoration, which has promoted the automation of interior decoration. The decoration robot mainly includes sensor, positioning system, control system and other parts. The sensor combined with the positioning system enables the decoration robot to realize the movement path planning, positioning, obstacle avoidance and other actions. Traditional robots use simultaneous localization and mapping (SLAM) to construct prior knowledge to avoid obstacles [7]. Therefore, the degree of intelligence is not high, and automatic identification cannot be carried out. Faced with the uncertainty of the actual indoor environment, the adaptability to the dynamic environment is not high, and the ability of path planning and obstacle avoidance needs to be improved. Therefore, a path planning strategy based on PPO deep reinforcement learning algorithm is proposed, which enables the decoration robot to obtain external information through the sensor system, make path selection decisions through autonomous learning, and improve the intelligent degree of the robot. Through the auxiliary model based on adaptive loss function, the adaptive degree of the robot is improved, providing a new way for the practical application of the indoor decoration robot.

II. Related Work

Interior decoration is an important part of urban construction, and its construction automation is an important direction of future development. In this regard, robot technology is a key technology, in which the research on robot path planning is an important issue and research hotspot. Using voxel space modeling and vector field path optimization, Min JK et al. proposed a new method for planning the robot's three-dimensional obstacle avoidance path to realize the robot's three-dimensional obstacle free path planning in the factory environment [8]. Guo X et al. proposed a robot intelligent assembly path planning scheme based on the improved Q-learning algorithm by adding a dynamic reward function to accelerate the convergence speed and design a trap avoidance scheme. Experiments show that the six joint robot UR10 has a good three-dimensional path optimization ability under this algorithm [9]. Zhang S et al. used information entropy to describe the population diversity of ant colony algorithm, thus improved an adaptive ant colony algorithm and improved the optimization ability of the algorithm. Through evaluation and significance test, it is proved that the algorithm is feasible in mobile robot path planning and has good performance [10]. Zhao W et al. proposed a heuristic search path planning method to solve the defects of traditional path planning methods in high demand environments. The experimental results show that more robots in narrow and long channels can run in a coordinated and

*Corresponding Author

orderly manner and have better path planning ability [11]. Jing et al. designed an intelligent interactive system for home robots, so that the robot has a perception and positioning system, and can realize autonomous navigation and obstacle avoidance. Experiments show that the robot using this system can better adapt to the family dynamic environment, and the human-computer interaction effect is good [12]. Islam M R et al. reset and introduced a new basic operator of chemical reaction optimization algorithm, and proposed an improved meta heuristic path planning algorithm. The experimental results show that the algorithm has good performance and can effectively improve the robot's path optimization ability in complex tasks [13].

Sharma K and Doriya R use the intelligent distance measurement method to improve the path planning ability of multi robot systems when performing tasks in large warehouses. This method has achieved good results in the coordinated operation and path optimization of multi robot systems [14]. Kang J G et al. designed a rerouting method using trigonometric inequality to improve the Rapid Exploring Random Tree (RRT). The simulation results show that, compared with RRT and RRT connection algorithm, the planning ability and path optimization ability of this algorithm are improved [15]. Zhang Z et al. proposed a path planning method based on the improved A* algorithm according to the special kinematic characteristics of the spherical motion robot, which improved the search efficiency of the robot and could find the optimal path in a short time [16]. Zhang TW et al. proposed an improved firefly algorithm (FA) combined with genetic algorithm (GA) to solve the defect of local optimal solution of FA algorithm. The experimental results show that the performance and accuracy of the improved algorithm are improved, and good results can be achieved when applied to robot path planning [17]. Chi W et al. proposed a heuristic path search method by using the feature extraction algorithm of Generalized Voronoi Diagram (GVD) to improve the path planning ability of fast robots in obstacle environments. The results show that this method is highly efficient, the results of one-time feature extraction can be reused, and the robot can efficiently search the obstacle free path [18]. Yang Y et al., based on the global and local levels, used the two-level path planning method to enable the robot to effectively avoid obstacles in a multi obstacle mobile environment. The results show that the actual application effect is good [19]. Xu T et al. perceived the shape of obstacles and planned the obstacle avoidance path of the robot arm under specific circumstances by improving the artificial potential field method. The results show that the method is highly adaptable to the environment and can effectively avoid obstacles [20]. In spherical tank inspection, Li J et al. planned the path of their inspection robot by improving Fleury algorithm, and recognized the welding line through depth learning network. The results show that the application effect of the path planning method is good [21].

From many researches on robot path planning, it can be seen that the current common path planning methods mainly include graph planning algorithm, spatial sampling, deep reinforcement learning and so on. The deep reinforcement learning method combines the deep learning and reinforcement learning, so that the robot can realize the autonomous learning of path planning in the interaction with the external environment. Compared with the traditional methods, it has obvious advantages, and is more suitable for the path planning of indoor construction of decoration robots. However, previous studies have mostly used the existing algorithm framework to build models, neglecting the design of the algorithm itself, especially the shaping of the reward function, and the quality of the reward function is directly related to the effect of robot learning. The novelty of this paper is to optimize the algorithm through the construction of reward function, aiming to improve the ability of robot autonomous planning. In this regard, this research focuses on the shaping and use of reward function, and deeply discusses the deep reinforcement learning method of robot path planning, hoping to propose an effective autonomous planning learning scheme.

III. Design of Self Planning Learning Scheme for Mobile Path of Decoration Robot

A. *Learning Method of PPO Algorithm and FCNN Neural Network*

With the development of smart cities, intelligent interior decoration has become a new direction of interior decoration development. Robots replace manual workers to carry out relevant decoration operations. During this period, the robot path planning problem needs to be solved. In this paper, PPO algorithm is used as the basic algorithm of path planning learning for decoration robot. This algorithm is a gradient optimization algorithm based on policy garden and off policy. It combines the advantages of strategy and value function, and performs well in tasks with continuous control and continuous scenarios. Generally, a reinforcement learning process can be described as a Markov Decision Process (MDP) [22]. This process can be expressed as a four tuple $M=(S,A,R,P)$, S is the environment state set, A is the action set, R is the reward function, and P is the state transition function. In the interactive environment, at each time step t, the agent's status is s_t and the action is a_t. It gets a reward r_{t+1} and moves to the next state s_{t+1}. When the cumulative reward reaches the maximum G_t, as shown in equation (1), the agent enters the termination state.

$$G_t=\sum\nolimits_{n=0}^{T-t-1}\gamma^n R_{t+n+1} \qquad (1)$$

In equation (1), $\gamma\in[0,1]$ is the attenuation coefficient. PPO algorithm is based on Actor-Critic algorithm. The Actor-Critic algorithm combines two reinforcement learning methods, value based and policy based, and is divided into two different network structures, Actor and Critic [23]. The idea of the algorithm is to use Actor network to generate actions to interact with the environment, and Critic network to evaluate actions. In the strategy π_φ, the state estimation of Critic output is shown in equation (2):

$$Q_\rho(s,a)\approx Q^{\pi_\varphi}(s,a) \qquad (2)$$

In equation (2), ρ is the parameters of Actor network, φ is the parameters of Critic network. Therefore, as a Policy Gradient (PG), the calculation of Actor-Critic algorithm is shown in formula (3):

$$\nabla_\varphi J(\varphi) \approx E_{\pi_\varphi}\left[\nabla_\varphi \log \pi_\varphi(s,a) Q_\rho(s,a)\right] \tag{3}$$

The Policy Gradient has a high square difference, which can be solved by subtracting a Baseline [12]. During actual operation, you can select a Baseline function and add an Advantage function in the form of $A^{\pi_\varphi}(s,a) = Q^{\pi_\varphi}(s,a) - B^{\pi_\varphi}(s)$, so as to obtain a new calculation formula of Actor-Critic algorithm, as shown in (4):

$$\nabla_\varphi J(\varphi) \approx E_{\pi_\varphi}\left[\nabla_\varphi \log \pi_\varphi(s,a) A^{\pi_\varphi}(s,a)\right] \tag{4}$$

According to formula (4), the Actor-Critic algorithm adopts online update, that is, it needs to collect data again after completing an update, which is inefficient. In this regard, PPO algorithm introduces Importance Sampling into the existing framework, so that samples can be reused and offline updates can be realized.In this method, when it is impossible to obtain the expected $e(x)$ of a variable x subject to a continuous random distribution m, a distribution n with close values is set, and the expected $f(x)$ of the distribution is calculated to obtain the expected score distribution m, as shown in formula (5):

$$\begin{aligned} E_{x\sim m}\left[e(x)\right] &= \int f(x) m(x) dx = \int f(x)\frac{m(x)}{n(x)} n(x) dx \\ &= E_{x\sim m}\left[e(x)\frac{m(x)}{n(x)}\right] \end{aligned} \tag{5}$$

Therefore, the Actor-Critic gradient calculation formula of Importance Sampling is introduced, that is, the gradient calculation formula of PPO algorithm is shown in (6):

$$J^{\bar{\varphi}}(\varphi) \approx E_{(s_t,a_t)-\pi_\varphi}\left[\frac{m_\varphi(a_t \mid s_t)}{m_{\bar{\varphi}}(a_t \mid s_t)} A^{\bar{\varphi}}(s_t,a_t)\right] \tag{6}$$

The path planning training flow chart of PPO algorithm is shown in Fig. 1.

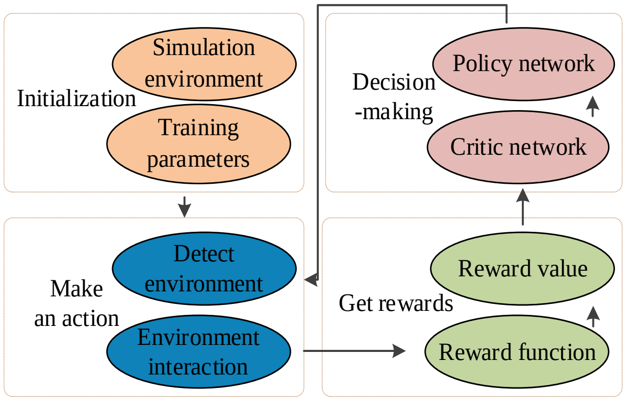


Fig. 1. PPO training process

FCNN neural network is used to process the state information of laser radar and robot. The algorithm has good fitting effect on nonlinear data. The full connection layer is converted into a convolution layer, which can be classified at the pixel level. However, it is not precise enough in the sampling process and is insensitive to some details. FCNN consists of input layer, hidden layer and output layer. The basic unit is neurons. There is no correlation between neurons in each layer, but all neurons in the other layer are associated. Information propagates unidirectionally from input layer to input layer. Its structure is shown in Fig. 2.

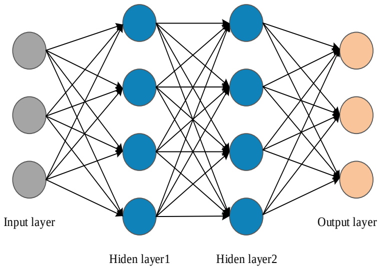


Fig. 2. Structure of FCNN

The propagation mode of FCNN can be expressed as formula (7):

$$\begin{cases} L_i = K_{i-1}H_i + C_i \\ K_i = F_i(L_i) \end{cases} \tag{7}$$

In equation (7), L_i represents the linear input of layer, K_i represents the output after layer activation, H_i represents the weight matrix from layer to layer, C_i represents the offset vector of layer, and F_i represents the activation function of layer. In deep learning, a reverse algorithm is usually used to make FCNN propagate from the output layer to the input layer to update the neuron weight matrix and bias vector. The principle is to bring the data set $I = \{(x^p, y^p)\}_{p=1}^{p}$ into FCNN training so that the output layer get $\hat{y}^p$ as shown in formula (8):

$$\hat{y}^p = FCN\left[\left(x^p, \varphi\right)\right] \tag{8}$$

In Eq. (8), $FCN[\ ]$ represents a FCNN neural network and φ represents the training parameters of the FCNN network. After we get $\hat{y}^p$, we can use the cost function to calculate $Loss(y^p, \hat{y}^p)$, and the gradient of the network can be obtained according to the partial derivative and learning rate σ of the loss function to the weight and offset. Finally, the inverse algorithm updates the weight H and offset C of the network, as shown in equation (9):

$$\begin{cases} H - \sigma \dfrac{\partial Loss(H,C)}{\partial H} \Rightarrow H \\ C - \sigma \dfrac{\partial Loss(H,C)}{\partial C} \Rightarrow C \end{cases} \tag{9}$$

B. Design and Optimization of Reward Function

In the process of deep reinforcement learning, the learning goal is to achieve the maximum cumulative reward, so the setting of reward function is directly related to the realization effect of learning goals [24]. The research aims at the problem of autonomous planning of the robot's indoor moving path, which involves two key abilities, namely, the ability of path optimization and the ability of obstacle avoidance. Therefore, two potential energy reward functions can be created respectively, and an initial reward function can be combined to get a better training effect. The initial reward function $R(s,a,\vec{s})$ is shown in equation (10):

$$R(s,a,\vec{s})=\begin{cases} 40 & if\ \delta_d < 0.25m \\ -20 & if\ \delta_0 < 0.2m \\ -0.15 & p(every\ step) \end{cases} \tag{10}$$

In Eq. (10), $\delta_d < 0.25m$ means that the robot successfully reaches the target within 0.25m, successfully completes the task, and the reward value is 60, and starts the next round of training; $\delta_0 < 0.2m$ means that if the robot reaches within 0.2m of the obstacle and the task fails, the reward value is -30 and the next round of training starts; p is punish, motivating robots to perform training tasks. For the path optimization capability, it involves the distance from the end of the task and the angle of the travel direction. First, set the potential energy function F_r for the target distance as shown in equation (11):

$$\begin{cases} F_r(s,a,\vec{s}) = \delta_r(\vec{s}) - \delta_r(s) \\ \delta_r = F_\varepsilon\left(r_{ax}-r_{tx}, r_{ay}-r_{ty}\right) \\ F_\varepsilon = \sqrt{(\nabla rx)^2 + (\nabla ry)^2} \end{cases} \tag{11}$$

In Eq. (11), r_{ax}, r_{ay} indicates the position of the target position on the x-axis and y-axis; r_{tx}, r_{ty} indicates the position of the trolley in the x-axis and y-axis; δ_r represents the distance between the robot and the target; ∇rx , ∇ry represents the difference between the target position and the trolley position in the x-axis and y-axis; δ_r represents Euclidean distance; s indicates the current status; a indicates the action taken in the current state; $\vec{s}$ indicates the next state to move to. Set the potential energy function F_a for the robot travel direction angle δ_g as shown in equation (12):

$$\begin{cases} F_g(s,a,\vec{s}) = \delta_g(\vec{s}) - \delta_g(s) \\ \delta_g = \begin{cases} \pi/4 & if\ g \le \pi/4 \\ abs(g) & if\ g \le \pi/4 \end{cases} \end{cases} \tag{12}$$

In Eq. (12)， g represents the orientation angle between the robot and the target position, and $abs(g)$ represents the absolute value. Within the range of 2 meters in diameter centered on the robot, the distance δ_d between the obstacle and the robot sets the potential energy reward function F_d as shown in Eq. (13):

$$\begin{cases} F_d(s,a,\vec{s}) = \delta_d(\vec{s}) - \delta_d(s) \\ \delta_d = \begin{cases} otr & if\ otr \le 1 \\ 1 & if\ otr \ge 1 \end{cases} \end{cases} \tag{13}$$

In Eq. (13) ， otr is the distance between the current robot and the obstacle. Adding the three potential energy reward functions set to the initial reward function $R(s,a,\vec{s})$ is the new reinforcement learning reward function, as shown in Eq. (14):

$$\dot{R}(s,a,\vec{s}) = R(s,a,\vec{s}) + F_r + F_g + F_d \tag{14}$$

In the training scheme adopted in the study, every time the robot reaches the target point or fails to avoid obstacles, it will re-enter the new training process, and each round will obtain a training path t_s including reward sequence r_s, as shown in Eq. (15):

$$t_s = s_1, a_1, r_1, s_2, a_{2,} r_2, \ldots, s_n, a_n, r_n \tag{15}$$

This training process is carried out in rounds until the cumulative reward reaches the maximum G_t . The global training in this way involves a credit allocation problem [25]. Generally speaking, in the whole process of robot path selection, the final selection step is relatively more Critic. However, as the number of training steps increases, the discount coefficient decreases exponentially. Therefore, the correlation between the reward return performance of the previous steps and the number of subsequent steps is very low, so that the variance of the final cumulative return is too high. This has a great impact on the update of the strategy gradient, and the sample utilization and training efficiency are therefore low. In order to solve this problem, we can build a credit allocation model, reallocate the reward value at the end of the decision sequence, and obtain the cumulative return value with smaller variance, so as to make the gradient update of reinforcement learning algorithm more stable and accelerate the training speed. The training structure of adding credit allocation model is shown in Fig. 3.

In Fig. 3, S_{ld} is the lidar information, S_s is the self-status information, $V_\eta(s)$ is the status output value of the critical network, $\pi_\iota(a)$ is the action distribution of the actor network, and $V_\varphi(s)$ is the status output value after credit allocation.After a complete path planning training, the robot can obtain three different reward values, namely, the positive reward value 60 for completing the training, the negative reward value -30 for collision obstacles, and the common reward value [-2.0,2.0] for reaching the maximum number of training steps. Each reward is shown in Fig. 4.

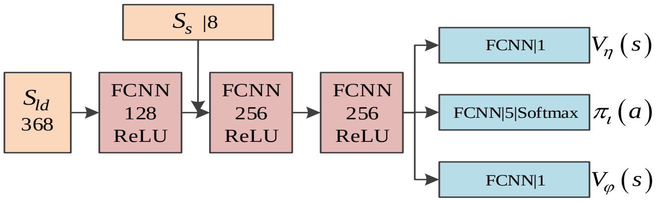
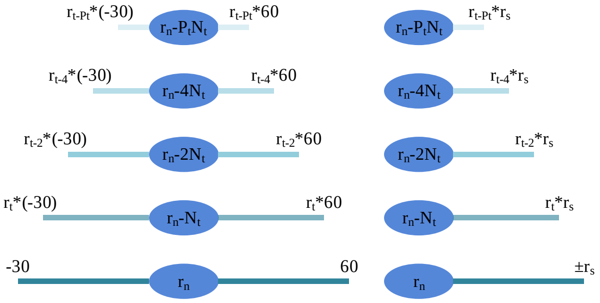


Fig. 3. Training model after adding credit allocation



（a）Positive and negative rewards （b）Normal rewards

Fig. 4. Credit allocation models

The core of the problem to be solved is to allocate the reward impact of the later stage to the reward of the previous stage. First, the whole sequence needs to be segmented. The operation requirements are shown in formula (16):

$$P_t = round\left(S_t / N_t\right) \tag{16}$$

In Eq. (16), S_t is the reward sequence length, N_t is the credit allocation indicator, that is, the sequence length divided, N_t is the number of divided sequences, and $round(\ )$ refers to rounding. On the basis of the new segmented sequence, the reward of the later segment of the original sequence is redistributed in the direction from the back to the front and the discount coefficient γ is taken as the criterion, so as to update the reward sequence r_s and recalculate the cumulative reward G_t. Then, the advantage functions AF_{old} and AF_{new} of the original sequence and the updated sequence are introduced respectively, and their values are the respective maximum cumulative reward G_t and the state output value of the Critic network. The two advantage functions are multiplied by the coefficients κ and λ respectively, so obtain the final form AF of the reward function, as shown in formula (17):

$$AF = \kappa AF_{old} + \lambda AF_{new} \tag{17}$$

In addition, in order to make the credit allocation perceived by the algorithm, a CACritic network is introduced to output additional credit allocation status. Finally, the loss functions of CACritic network, Critic network and Actor network are calculated by PPO algorithm respectively, so as to complete the shaping of the reward function.

C. Auxiliary Model Construction of Adaptive Loss Function

In the static environment, because the elements of the environment are fixed, the “State-Action-Reward” of machine training is fully responsive. As long as the time and strategy required for training are met, the robot can tend to repeat training on a successful path or finally retain the training state related to all environmental elements, so it is easy to obtain the optimal path and achieve the maximum reward. However, in the real decoration scene, the environmental state will change constantly, and new structures and elements will be generated. Taking the same action at different times of the same path may result in different reward situations, which requires more adaptation to costs and has a negative impact on machine learning. The change of this environment is accidental. Therefore, the study assumes that the uncertain s-a-r condition obeys the Gaussian distribution, and then constructs an auxiliary model to predict this distribution. In the training process, the uncertainty of s-a-r is transformed into the uncertainty of obtaining rewards to affect the robot's decision-making. Therefore, the core of the auxiliary model is to connect two FCNN on the basis of the training model to predict the mean and variance of s-a-r distribution, so as to represent the mean of the reward and measure the uncertainty of the reward, and then use the adaptive regression loss function to compare the predicted value with the actual reward value. The reward uncertainty perceived by the robot with the help of the auxiliary model can be used as a reference for the next path planning decision-making, so as to improve the adaptability of the robot and speed up the speed of completing the learning task. The training structure with auxiliary model is shown in Fig. 5.

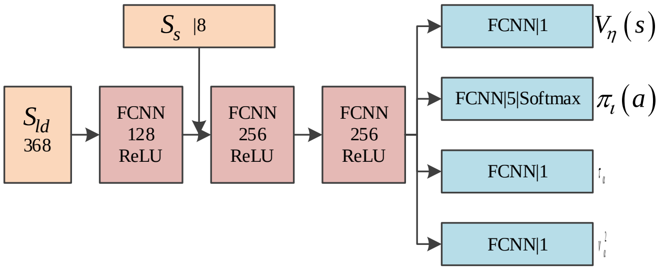


Fig. 5. Training model after adding credit allocation auxiliary task

In Fig. 5, τ_α is the s-a-r distribution mean predicted by the auxiliary model, and v_α^2 is the distribution variance.The specific method to build the model is to set the S-A-R data set of the dynamic environment, which follows the Gaussian distribution, described as $Trend(s,a,r)$,as shown in Eq. (18):

$$Trend\left(s,a,r\right) \in N\left(\tau_\alpha, \upsilon_\alpha^2\right) \tag{18}$$

In Eq. (18), τ_α is value of distribution, υ_α^2 is variance of distribution, used to measure the uncertainty of reward. FCNN is used to predict the mean and variance of S-A-R distribution. The prediction results are compared with the actual reward situation for regression training, and then the adaptive loss function W_α is used to calculate the difference between the two, as shown in equation (19):

$$W_\alpha = \frac{1}{n}\sum_{j=1}^{n}\frac{1}{2\upsilon\left(s_j\right)^2}\left\|h\left(s_j\right)-r_j\right\|^2 + \frac{1}{2}\log\upsilon\left(s_j\right)^2 \tag{19}$$

In Eq. (19), s_j represents the state of the path, r_j is the reward obtained in this state, $h\left(s_j\right)$ is the predicted mean value obtained by the auxiliary model, $\upsilon\left(s_j\right)^2$ is the variance obtained by the prediction model, and $\log\upsilon\left(s_j\right)^2$ is the constraint added to prevent the gradient turbulence caused by too high variance. Finally, the loss functions of CACritic network, Critic network and Actor network are calculated by PPO algorithm respectively, so as to complete the addition of auxiliary model.

IV. Effect Analysis of Self Planning Learning Strategy for Mobile Path of Decoration Robot

A. Simulation Experiment Setup

The common robot operating system (ROS) is used as the programming framework of the robot running program, and the simulation environment is built in combination with the Gazebo simulation platform. Gazebo can perform simulation independently of ROS, or install ROS related function packs to build an environment model as a node of ROS. In addition, the turtlebot3 of the turtlebot series robots is selected as the simulation robot. It has high support for ROS, strong modularity and flexibility, and has a large number of relevant resources in the ROS community, which brings great convenience to the research. The ROS simulation training framework thus established is shown in Fig. 6.

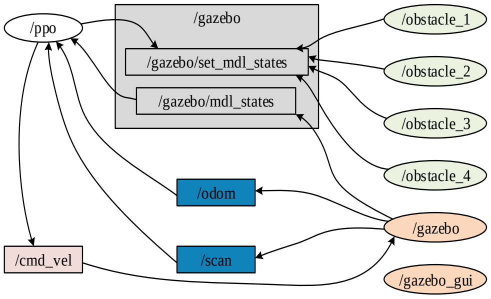
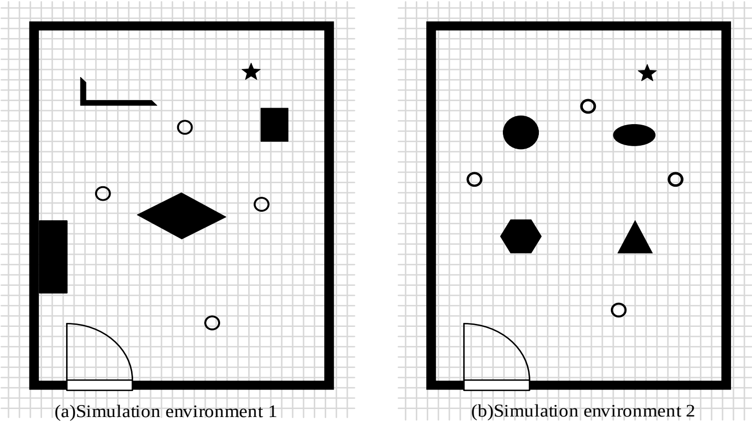


Fig. 6. Framework of ROS simulation training

In Fig. 6, /ppo refers to the algorithm used in the experiment/ cmd_ Vel is the moving speed command sent to the robot/ gazebo and /gazebo_ GI refers to the Gazebo simulation program/ gazebo/set_ mdl_ States and /gazebo/mel_ States means setting obstacles and robot states/ Odom and /scan refer to mobile state sensor and laser ranging sensor/ obstacle_ 1~/obstacle_ N means moving obstacles.Firstly, the experiment simulation environment is built by using Gazebo software. The research is aimed at the indoor movement of the decoration robot. Therefore, considering the actual situation and the needs of experimental training, two kinds of square enclosed spaces including static obstacles and dynamic obstacles are built. The plan is shown in Fig. 7.


(a)Simulation environment 1

(b)Simulation environment 2

Fig. 7. Gazebo simulation environment

In Fig. 7, the closed block represents the static obstacle, the circular graph represents the dynamic obstacle, and the five pointed star represents the target position. Secondly, the linear velocity and angular velocity at different levels are set for the action parameters of the robot; For the running environment of the experimental program, the operating system is Ubuntu20.04 LTS, the processor is Intel i5 12600K, the graphics card is NVIDIA RTX2070Ti, python version 3.9, and pytoch version 1.9.1; The machine status includes real-time distance to the target, travel direction angle, travel speed information and completed action information; The ranging system adopts light detection and ranging (LIDAR).

B. Path Planning Ability Test and Performance Analysis of Decoration Robot

Firstly, the training effect of path planning for decoration robot is compared between the initial reward function training model PPO and the improved reward function training model. The improved reward function training model is divided into improved reward function PPO-S and added credit allocation model PPO-SC. In static simulation environment 1, the path planning training effect of each model is shown in Fig. 8.

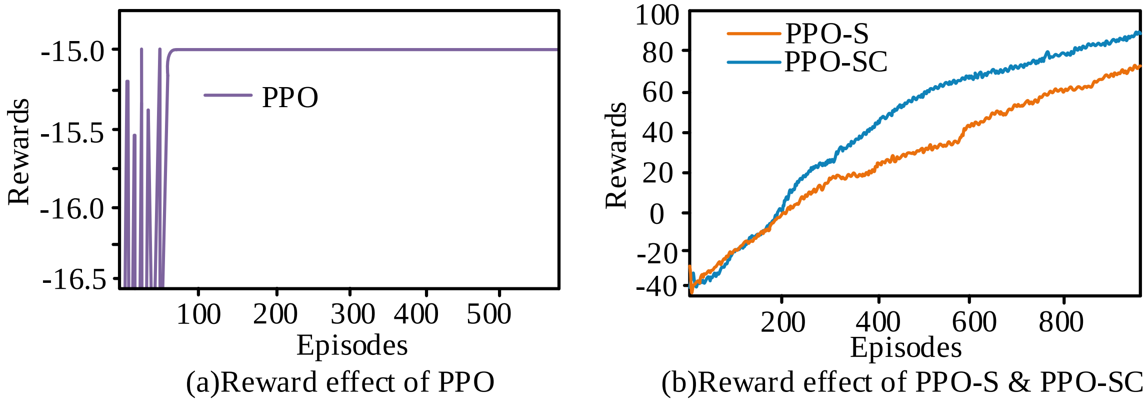


Fig. 8. Simulation environment 1(Static state)

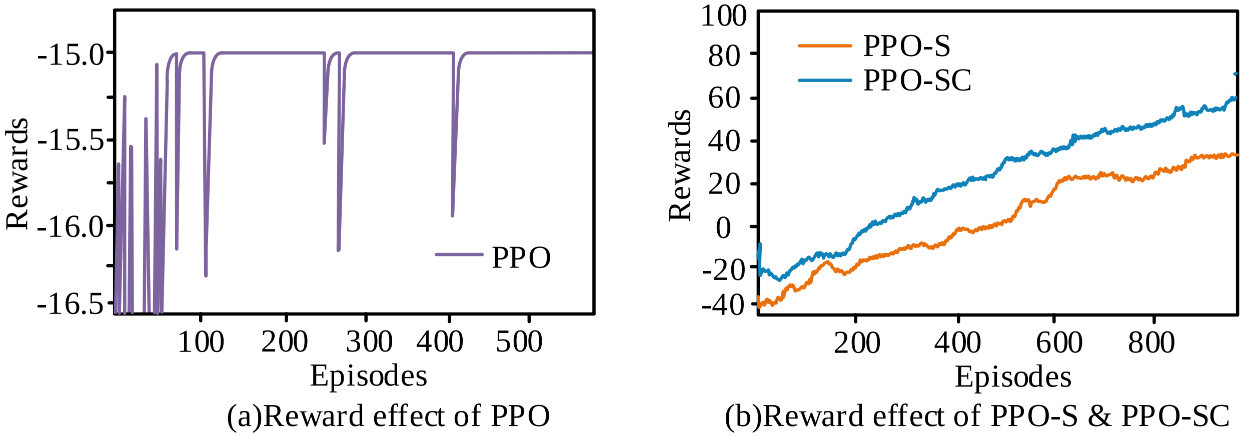
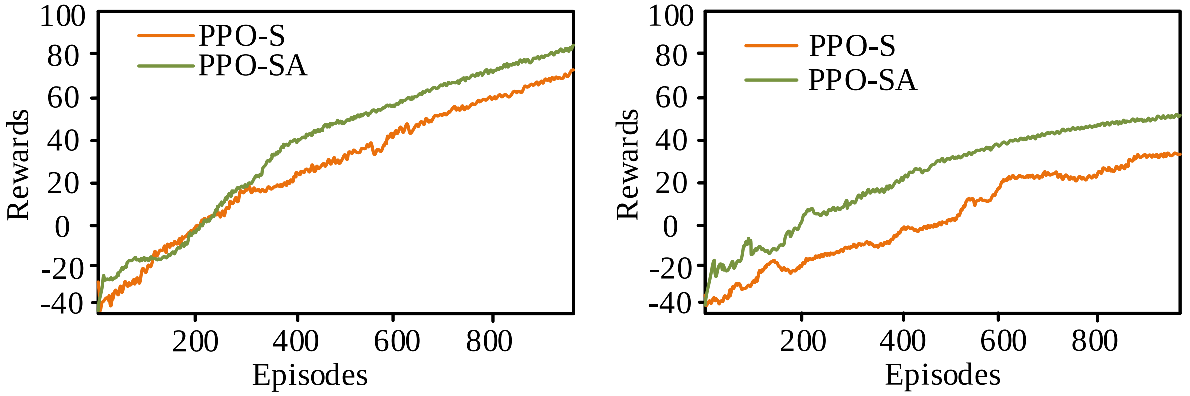


Fig. 9. Simulation environment 1(Dynamic state)

The results show that the PPO model robot has more collisions with its random exploration route, and gets some negative rewards. Finally, in order to avoid collision, it stays in place, and the reward remains at -15, and the task fails. PPO cannot guide the robot to complete the training task. In the PPO-S and PPO-SC models, the robot can complete the learning of path planning, and the PPO-SC model has faster rising speed and higher success rate. Therefore, in static environment 1, the new reward function and credit allocation model improve the robot's path planning ability. In dynamic environment 1, the training effect of each model is shown in Fig. 9.

The results show that compared with the static environment, the robot collision situation of PPO model increases and gets more negative rewards. In the simulation environment, the robot will still collide with the dynamic obstacles when it is stationary. In PPO-S and PPO-SC models, the training efficiency of the robot is significantly lower than that of the static model, and the stability is poor, but the training effect of PPO-SC is still better. To sum up, PPO-S with potential energy function can effectively guide the robot to achieve the goal and complete the training task, and the learning ability is improved and the training effect is better after credit distribution and reward sequence redistribution. However, the dynamic environment adaptability is poor and the training effect is average. In order to verify the effect of auxiliary tasks on improving the adaptability to dynamic environment, the training effect of PPO-S model and PPO-SA model combined with auxiliary tasks is compared. The results are shown in Fig. 10.

The results show that in both static and dynamic environments, PPO-SA model has better reward performance than PPO-S model. The reward rises faster and the learning efficiency is higher. In the dynamic environment, the robot training effect under PPO-S model is significantly worse than that in the static environment, the number of collisions has increased significantly, the training effect is poor, and the reward curve fluctuates greatly. In the PPO-SA model, through early learning, the auxiliary model in the middle and late stages can effectively predict the reward distribution, so that the robot can avoid obstacles in advance in route selection, the training effect has been significantly improved, and the reward curve is relatively stable. In order to further verify the effect of the research and design strategy in robot path planning, the deep deterministic policy gradient (DDPG) algorithm, A-star algorithm and RRT algorithm are used to design the robot indoor path planning modell. Conduct training in static environment 1, and compare the percentage of the difference between the completed path and the optimal path. The results are shown in Fig. 11.



(a)Simulation environment 1 (Static） (b)Simulation environment 1 (Dynamic)

Fig. 10. Reward effect of PPO-S & amp; PPO-SA

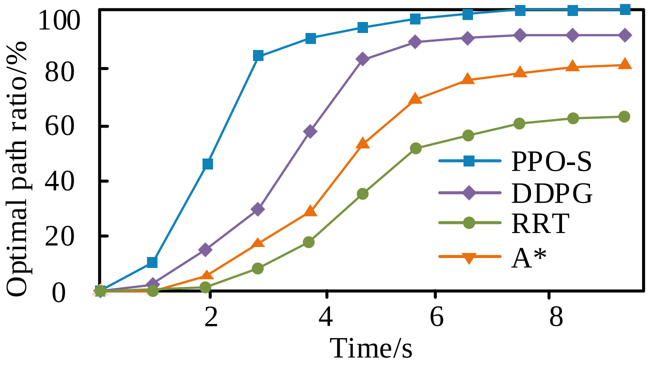


Fig. 11. Ratio of path difference in simulation environment1

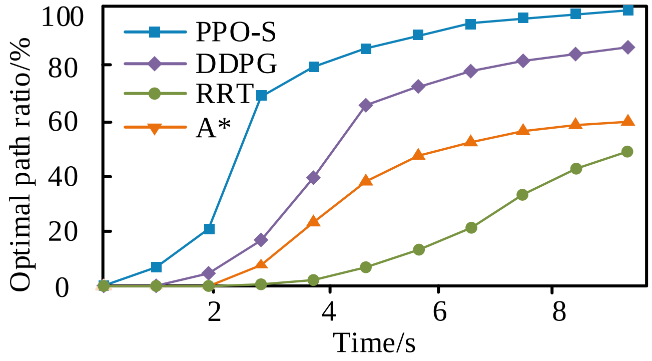


Fig. 12. Ratio of path difference in simulation environment2

It can be seen from the figure that in the static simulation environment 1, the path optimization ability of the PPO-S model studied and designed is better, the curve rises fastest, and the time to find the optimal path is the shortest. Conduct training in the static simulation environment 2, and compare the proportion of the difference between the completed path and the optimal path. The results are shown in Fig. 12.

It can be seen from the figure that in the static simulation environment 2, the path optimization ability of the PPO-S model studied and designed is better, the curve rises fastest, and the time to find the optimal path is the shortest. Each model is used to train robots in dynamic environment 1 and dynamic environment 2. Compared with PPO-SA model, the results show that the designed training model has achieved good training effect, while other algorithms used for comparison are not applicable in dynamic environment. It shows that the traditional common path planning algorithms are usually placed in static environment, and it is difficult to deal with dynamic environment without adding improvement measures. In the static simulation environment 2, 50 simulation experiments were conducted, and the statistics of the success times and success rates of different models are shown in Fig. 13.

TABLE I. SUCCESS TIMES AND SUCCESS RATE OF THE MODE

Model	Success times (times)	Success rate (%)
PPO-S	47	94
DDPG	40	80
RRT	21	42
A*	33	66

In Table I, from the success times and success rates of different models, PPO-S model has the highest success rate of 94%, significantly higher than other models, while DDPG model takes the second place, with a success rate of 80%. The success rate of A * model is the lowest, with only 21 successful times. It can be seen that the adaptability of PPO-S model is relatively optimal.

V. Conclusion

In order to promote the application of decoration robot in automatic decoration construction, it is a key problem to solve the autonomous planning of mobile path. In this paper, FCNN neural network is introduced into PPO algorithm and applied to interior decoration path planning. Aiming at the two core problems of path planning, a potential reward function is established. The reward function is redesigned based on the initial reward function. In order to solve the problem that the variance of cumulative reward returns is too large due to the difference between the front and rear reward values, a credit allocation model is constructed. By predicting the distribution of rewards through the auxiliary task model, the robot can predict the distribution of dynamic obstacles through training, and avoid obstacles in advance. The simulation results show that compared with the training model of the initial reward function, the learning efficiency of the robot is significantly improved and there are fewer negative rewards in the training of the new reward function after adding credit allocation; In a dynamic environment, the reward curve is more stable, the learning efficiency is higher, and the adaptability is better. Compared with other common path planning algorithms, PPO-S model has better performance in path planning, with the highest success rate of 94%, 14% higher than DDPG. Therefore, the research and design of deep reinforcement learning algorithm for indoor mobile path planning of decoration robot is feasible, and the training effect is also very good, which has important practical significance. Applying this scheme to the actual environment is the next research direction.

ACKNOWLEDGMENT

The research is supported by: 2018 Guangdong Young Innovative Talents Project “Research on the application of soft decoration materials in interior decoration design from the perspective of Internet” (Project No.: 2018gwqncx141).

REFERENCE

[1] Y. Chen, D. Han. “Water quality monitoring in smart city: A pilot project,” Automation in Construction, vol. 89, pp. 307-316, 2018.

[2] H. Liu, Y. Deng, D. Guo, et al. “An Interactive Perception Method for Warehouse Automation in Smart Cities.,” IEEE Transactions on Industrial Informatics, PP(99):1-1, 2020.

[3] A. J. Wit, L. Vasey, V. Parlac, et al. “Artificial intelligence and robotics in architecture: Autonomy, agency, and indeterminacy.” International Journal of Architectural Computing, vol. 16, no. 4, pp. 245-247, 2018.

[4] S. Seriani, A. Cortellessa, S. Belfio, et al. “Automatic path-planning algorithm for realistic decorative robotic painting.” Automation in Construction, vol. 56(aug.), pp. 67-75, 2015.

[5] Y. S. Lee, S. H. Kim, M. S.Gil , et al. “The study on the integrated control system for curtain wall building faade cleaning robot,” Automation in Construction, vol. 94(OCT.), pp. 39-46, 2018.

[6] T. Kim, H. Lim, K. Cho, “Conceptual robot design for the automated layout of building structures by integrating QFD and TRIZ.” The International Journal of Advanced Manufacturing Technology, vol. 120, no. 3, pp. 1793-1804, 2022.

[7] H. Chen, H. Huang, Y. Qin, et al. “Vision and laser fused SLAM in indoor environments with multi-robot system.” Assembly Automation, vol. 39, no. 2, pp. 297-307, 2019.

[8] J. K. Min, P. Kang, “Generation of a 3D robot path for collision avoidance of a workpiece based on voxel and vector field.” Journal of Mechanical Science and Technology, vol. 36, no. 1, pp. 385-394, 2022.

[9] X. Guo, G. Peng, Y. Meng, “A modified Q-learning algorithm for robot path planning in a digital twin assembly system.” The International Journal of Advanced Manufacturing Technology, vol. 119, no. 5-6, pp. 3951-3961, 2022.

[10] S. Zhang, J. Pu, Y. Si, “An Adaptive Improved Ant Colony System Based on Population Information Entropy for Path Planning of Mobile Robot.” IEEE Access, PP(99):1-1, 2021.

[11] W. Zhao, R. Lin, S. Dong, et al. “Dynamic node allocation based multirobot path planning.” IEEE Access, PP(99):1-1, 2021.

[12] Q. Li, J. Wu, et al. “Autonomous Tracking Control for Four-Wheel Independent Steering Robot Based on Improved Pure Pursuit.” Journal of Beijing Institute of Technology, vol. 29, no.106(04), pp. 35-42, 2020.

[13] M. R. Islam, P. Protik, S. Das, et al. “Mobile robot path planning with obstacle avoidance using chemical reaction optimization.” Soft Computing, pp. 1-28, 2021.

[14] K. Sharma, R. Doriya, “Coordination of multi-robot path planning for warehouse application using smart approach for identifying destinations.” Intelligent Service Robotics, vol. 14, no.2, pp. 313-325, 2021.

[15] J. G. Kang, D. W. Lim, Y. S. Choi, et al. “Improved RRT-Connect Algorithm Based on Triangular Inequality for Robot Path Planning.” Sensors, vol. 21, no. 2, pp. 1-34, 2021.

[16] Z. Zhang, Y.Wan, Y. Wang, et al. “Improved hybrid A* path planning method for spherical mobile robot based on pendulum.” International Journal of Advanced Robotic Systems, vol. 18, no. 1, pp. 671-680, 2021.

[17] T. W. Zhang, G. H. Xu, X. S. Zhan, et al. “A new hybrid algorithm for path planning of mobile robot.” The Journal of Supercomputing,: pp. 4158–4181, 2021.

[18] W. Chi, J. Wang, Z. Ding, et al. “A Reusable Generalized Voronoi Diagram Based Feature Tree for Fast Robot Motion Planning in Trapped Environments.” IEEE Sensors Journal, PP(99):1-1, 2021.

[19] Y. Yang, Z. Lin, M. Yue, et al. “Path planning of mobile robot with PSO-based APF and fuzzy-based DWA subject to moving obstacles.” Transactions of the Institute of Measurement and Control, vol. 44, no. 1, pp. 121-132, 2022.

[20] T. Xu, H. Zhou, S. Tan, et al. “Mechanical arm obstacle avoidance path planning based on improved artificial potential field method.” Industrial Robot,vol. 49, no. 2, pp. 271-279, 2022.

[21] J. Li, S. Jin, C. Wang, et al. “Weld line recognition and path planning with spherical tank inspection robots.” Journal of Field Robotics, vol. 39, no. 2, pp. 131-152, 2022.

[22] Q.V. Do, I.Koo, “Actor-critic deep learning for efficient user association and bandwidth allocation in dense mobile networks with green base stations.” Wireless Networks, vol. 25, no. 5, pp. 5057-5068, 2019.

[23] J. Guo, *L. I.* Mengtian, L. I. Quewei, et al. “GradNet: Unsupervised Deep Screened Poisson Reconstruction for Gradient-Domain Rendering.” ACM Transactions on Graphics, vol. 38, no. 6, pp. 223.1-223.13, 2019.

[24] X. Ning, H. Wang, et al. “Multi-Level Policy and Reward-Based Deep Reinforcement Learning Framework for Image Captioning.” IEEE Transactions on Multimedia, vol. 22, no. 5, pp. 1372-1383, 2019.

[25] B. A. Richards, T. P. Lillicrap, “Dendritic solutions to the credit assignment problem.” Current opinion in neurobiology, vol. 54, pp. 28-36, 2018.