International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18 (2026) pp. 1-22

https://doi.org/10.70917/ijcisim-2026-0366
Article

Research on Virtual Reality Scene Construction and Space Layout Optimization in Interior Design Based on Collaborative Optimization Algorithm

Jiefang Jin [1,*]

[1] School of Art and Design, Shanghai University of Engineering Technology, Shanghai, 201620, China
* Correspondence author: JJF20251121@163.com

Abstract: With the wide application of virtual reality technology to the field of interior design, people pay more and more attention to the spatial layout of virtual interior scenes and the simulation effect of scene construction. In this study, we first design the indoor spatial point matching algorithm based on three-dimensional vision, use the similarity metric formula to solve the distance between the spatial points in the base image and the original image, and eliminate the wrong points to improve the spatial point matching accuracy, and then use the VR technology to carry out the virtual modeling of the interior scene. The energy function is introduced for optimization, spatial layout planning is carried out, and the collaborative particle swarm algorithm is used for optimization to realize the exploration of indoor virtual reality scene construction and spatial layout optimization methods. In the chi-square test p-value, for Q1 (the null hypothesis is that participants vote the same number of votes for each emotion), the p-value is much lower than 0.05, and the results indicate that there is a difference in voting for each emotion. For Q2 (the null hypothesis is that the participants vote the same number of votes for the answer to the question of whether the color distribution is similar to the common color distribution of that type of scene in other scenes except the real scene), the results are more diverse, which shows that the color distribution is similar to the common color distribution of that type of scene, thus verifying the validity and correctness of the method in this paper.

Keywords: collaborative optimization; PSO algorithm; VR; interior design; virtual reality scene construction

1. Introduction

Under the big data environment, information technology promotes the rapid development of the interior design industry, especially the application of virtual reality (VR) technology [1]. In traditional interior design, most of them are designed by the designer according to the user's needs, and then drawn by the designer, which will lead to poor decorative effect [2-3]. Applying VR to the design of indoor scene 3D layout planning scheme can satisfy the common needs of the designer and the user to a greater extent, and facilitate the designer's later modification, which in turn improves the comfort of indoor scene layout planning [4-7]. In the course of the development of the interior decoration and design industry so far, the application of VR and computer technology is essentially to meet the user's personalized needs, and the realization of this purpose requires more effective communication between the designer and the user, in order to achieve the best scene layout planning effect [8-11]. The medium that connects the designer and the user is based on design drawings and computer 3D effects, which have evolved many times to eventually restore the form of the work in different Chen Gueda, but the realization of the scene is very different from the user's feelings in the real scene [12-15]. The

application of VR can provide users with different scenes and perform different scene transformations at the same time, which greatly increases the communication efficiency between users and designers and improves the overall work efficiency [16-18]. Therefore, the construction of virtual reality scenes has a very significant meaning for contemporary indoor scene layout planning.

For the problem of interior space layout, it is usually required to create a layout or optimize a layout in a given area or space [19]. During the optimization process, in addition to maintaining the layout and structure of the indoor scene, it is necessary to minimize the distortion and distortion of the objects in the scene [20-21]. Therefore, taking the boundaries of a given building as well as the user's requirements as design constraints, the use of intelligent algorithms can automatically generate indoor space layouts and can be further extended to apply to space layouts of large-scale scenarios, which has great potential for application [22-24].

The article firstly reconstructs the 3D visual spatial points of the indoor environment, and then uses the Euclidean distance to calculate the vector relationship between the spatial points to realize the 3D visual spatial point matching. Then the architectural model of the VR scene is converted, combined with the VR rapid generation technology, the indoor model is converted into the components in the VR scene, and the virtual scene is established by utilizing the design of the three parts of the VR model library: the auxiliary scene, the cross-terminal collaboration and the roaming display. After that, the indoor scene layout template and energy function are designed to help the layout optimization process converge quickly, and the collaborative PSO is designed for solving. Finally, we enter the experimental part, respectively, simulation experiments on the collaborative particle swarm optimization algorithm designed in this paper, indoor floor plan layout generation experiments using the method of this paper, and evaluation of the effect of interior design based on the method of this paper.

2. Indoor virtual reality scene construction

2.1. Indoor spatial point matching algorithm based on 3D vision

2.1.1. Three-dimensional visual spatial point reconstruction of indoor environment

In order to restore the location of 3D visual space points in the indoor environment, let the 2 original images R_1, R_2 pairing point of $x_1 \leftrightarrow x_2, x_1$ have the image coordinates of (u_1, v_1), x_2 have the image coordinates of (u_2, v_2), x_1 and x_2 corresponding to the camera coordinates of f_1, f_2, respectively, and the basis matrix of the original image is represented by F, then all matching points should be satisfied:

$$x \mid Fx_2 = 0 \tag{1}$$

If Eq. (1) is satisfied, the original image is normalized to obtain the canonical 3D visuospatial points:

$$\begin{cases} P_1 = (x_{11}, y_{11}, 1)^{\gamma} = J^{-}(u_1, v_1, 1)^{\gamma} \\ P_2 = (x_{22}, y_{22}, 1)^{\gamma} = J^{-1}(u_2, v_2, 1)^{\gamma} \end{cases} \tag{2}$$

Where J denotes the image internal reference matrix [25]. After obtaining the canonical 3D visual spatial points, the relationship between the spatial points and the image points needs to be found with the following equation:

$$\lambda \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = \begin{bmatrix} P_{11} \\ P_{12} \\ P_{13} \end{bmatrix} X_i \tag{3}$$

Where: p denotes the row vector of the projection matrix. p_{11}, p_{12} and p_{13} denote the chi-square coordinates on the image corresponding to p. x_i denotes the spatial chi-square coordinates of the corresponding matching point. y_i denotes the spatial chi-square coordinates of the corresponding matching point. Derived from Eq. (2) and Eq. (3):

$$\begin{bmatrix} p_{13}x_i - p_{13} \\ p_{12}y_i - p_{12} \\ p_{23}x_i - p_{21} \\ p_{23}y_i - p_{22} \end{bmatrix} X_i = 0 \tag{4}$$

Apply Eq. to the space point x.

2.1.2. 3D visual spatial point matching for indoor environments

After completing the visual spatial point reconstruction of the indoor environment, it is necessary to deal with the mapping relationship between multiple images, which is also known as spatial point matching, to find similar spatial point pairs by searching for this correspondence, and then after obtaining the spatial point relationship of each image. The Euclidean minimum distance algorithm is applied to calculate the vector relationship between spatial points to realize the corresponding spatial point matching and establish the connection between multiple images. Let any spatial point of 2 images R_1, R_2 be h_1, h_2, then the similarity measure formula of h_1, h_2 is:

$$d(h_1, h_2) = \sqrt{\sum_{j=1}^{64} (h_{1\ddot{g}}, h_{2ij})^2} \tag{5}$$

Where: d denotes the similarity value. i denotes the next nearest neighbor coordinates. j denotes the nearest neighbor coordinates.

Indoor environment-aware spatial point matching process: first, extract the spatial point information of the image, which is expressed in vector form to simplify the calculation process, and then solve the distance between the spatial points in the original image and the baseline image by Eq. (5). Secondly, the distance between the nearest neighbor coordinate point and the next nearest neighbor coordinate point of the spatial point in the reference image in the original image is calculated. If the ratio of the nearest neighbor coordinate point to the second nearest neighbor coordinate point is less than a certain threshold, it means that the nearest neighbor space point corresponding to the point is a pair of matching points. If it is greater than a certain threshold, it means that the number of matching points obtained is large and the accuracy is low. When calculating, the threshold should be set reasonably to effectively reduce the number of false matches and appropriately increase the work of eliminating erroneous points.

2.2. Virtual Generation Algorithm for Indoor Scenes Based on VR Technology

2.2.1. Virtual Modeling Methods for Indoor Space

The indoor spatial point matching algorithm based on 3D vision can realize the accurate capture of indoor spatial points and eliminate erroneous points. On this basis, this section further proposes a virtual generation algorithm for indoor scenes based on VR technology to realize the construction of indoor virtual reality scenes.

(1) Constraint-based process-oriented modeling

The main task of indoor modeling is to collect modeling data, and the main data for indoor modeling are: 2D graphics, surface graphics and model surface texture. With the diverse development of indoor models, the difficulty of obtaining data has increased in recent years due to the different styles of interior design and other reasons, in order to reduce the low utilization of data in the process of indoor design modeling and other problems, the data attribute analysis model is shown in Figure 1.

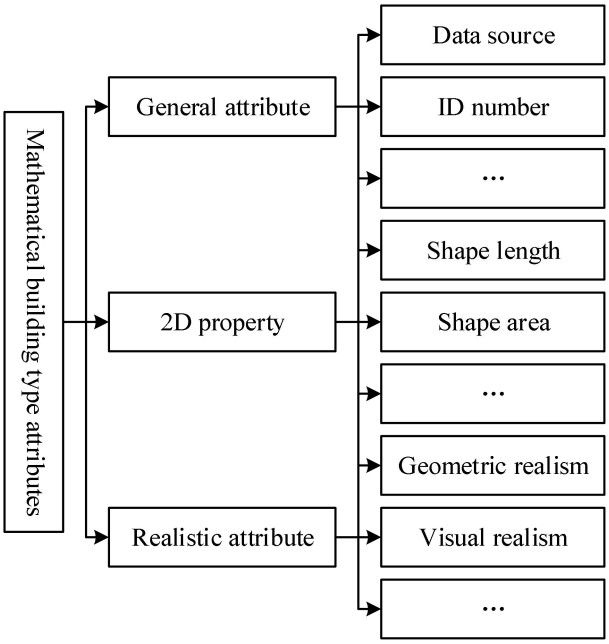


Figure 1. Data attribute analysis model

The model provides a comprehensive consideration of the data sources required for the indoor modeling process, etc., as different data play different roles in influencing the modeling. The realism factor of modeling is analyzed in terms of the characteristics of the input data to improve the realism of the model in terms of geometric and visual aspects respectively. The above rules are used in different algorithms when modeling interior architecture. The visual sense can be enhanced by mapping the architectural texture through geometric chunking and texture selection rules [26].

Since there is no height attribute in the input data, which results in the lack of 3D information of the model during modeling, a customized height generation rule is used to generate the height value of the building using a procedural method and two attributes, building type and building level, are added to the building through a calibration platform. The building type attribute can ensure the special characteristics of the building. The building level attribute not only ensures that there is no large height difference between buildings, but also ensures the rationality of texture mapping. According to the building attribute to generate the height of the building model, the building height range can be expressed by the formula:

$$\begin{cases} H_{\max} = \min(A_{cype}, A_{bree}) \\ H_{\min} = \max(A_{cype}, A_{bree}) \end{cases} \tag{6}$$

Where, $H_{\max}$ and $H_{\min}$ denote the maximum and minimum height of the building respectively. $A_{v\rho e}$ denotes the building type attribute. A_{bw} denotes the building level attribute. In addition to the building type and building level attributes, the building height is also affected by other constraints such as the environment, based on which the method of calculating the building height by randomized way is proposed, and the formula can be expressed as:

$$\begin{cases} H = \varepsilon[B(A_{ope}, A_{loce})] + \gamma C(A_{side}, A_{loce}) + \eta D(\overline{h}) + \lambda E(\cdot) \\ B(A_{ope}, A_{loce}) = \dfrac{X}{R_{\max}}(H_{\max} - H_{\max}) + H_{\min} \\ \varepsilon + \gamma + \eta + \lambda = 1 \end{cases} \tag{7}$$

Where, B denotes the influence function of building type and building level on the building. C denotes the influence function of neighborhood on the building. D denotes the influence function of the surrounding environment on the building. E represents the influence function of other factors. $\varepsilon, \gamma, \eta$ and λ denote the influence factors of the corresponding functions, respectively. A_{sidt} and A_{sing} denote the length and width of the neighborhood, respectively. $\overline{h}$ denotes the average height of the buildings in the neighborhood. X denotes the random number generated by the random function.

$R_{\max}$ denotes the maximum value generated by the random function. The height generation strategy allows the generation of the heights of all buildings and, due to the randomness, can enhance the realism of the model while conforming to the actual situation.

Considering the possibility of different textures for the four elevations of the building, the approach of combining the geometry module with texture mapping is adopted. The building façade is first segmented into block units through customized rules, and then the texture is mapped onto the block units according to the requirements to enhance the visual realism. Since neural network has the function of connecting individual units and forming a complex network, the neural network is applied to the learning of texture selection for indoor architectural models, and the formula can be expressed as:

$$y = F\left(\sum_{i=0}^{s} u_i \alpha_i\right) \tag{8}$$

where y denotes the appropriate texture image selected by the learning model. F denotes the excitation function. u_i denotes the input to the model. α_i denotes the size of the weights under different input attributes. When the input data attributes are different, the output of the neural network and the impact on the building model are also different.

Different texture patterns are selected for mapping in four different directions: front, back, left and right of the façade block to ensure that different types of buildings can get the corresponding texture patterns, which are named in $Atyp\_Alev\_Anum\_Adir.jpg$ manner. Where $Anum$ denotes the encoding of the texture in that building type. $Adir$ denotes the four directions of the façade. Combining the height of the building and the texture naming rule, the range of the texture library of the building is determined, and the formula can be expressed as:

$$G_{nam} = \frac{Y * (N-1)}{R_{max}} + 1 \tag{9}$$

where Y denotes the random value selected from the random number. $R_{\max}$ denotes the maximum value produced by the random function. N denotes the optional solids. Through the architectural generation rules and texture selection rules, not only the construction of indoor architectural models can be realized, but also the architectural models can be made to have a strong sense of realism, which provides the basis for complex digital indoor modeling.

(2) Indoor Layout Estimation

In order to generate a realistic indoor building 3D model as much as possible, it is necessary to recover the scene layout and other information. The object detection algorithm is used to recognize the objects in the building, and the layout structure of the objects is estimated by the target detection results. The objects in the 3D space are first projected into the 2D space, and the optimal layout solution is sought by adjusting the layout. In terms of a single object in the scene, the objective function is:

$$\min_H k = 1 - L(h, O(H, l, P)) \tag{10}$$

where $L(\cdot)$ denotes the probability of overlap of two regions in a two-dimensional space. h denotes the target detection result. $O(\cdot)B$ denotes the projection function. H and l denote the object information and object rotation information in 3D space, respectively. P denotes the parameters. For multiple objects in the scene, it is necessary to consider not only the target detection results of each object in 2D space, but also the overlapping problem of each object space. The multiple object spatial layout objective function is:

$$\min_H k = \sum_i (1 - L(h_i, O(H_i, l_i, P))) + \varpi \sum_{m,s} L(H_m, H_s) \tag{11}$$

Where the first term of Eq. (6) denotes the deviation of each object in the two-dimensional space from the detection result of the objective function. The second term represents the overlap probability of two objects in the three-dimensional space. By adjusting parameter ϖ, it can be ensured that the objects in the three-dimensional space do not overlap, which is in line with the physical principles satisfied in the real scene.

In order to make the layout of objects in the building more accurate, the optimization method is designed. With the information obtained from the two-dimensional space, the indoor space enveloping box can be made to generate more accurate data with the formula:

$$\frac{q_k^i}{r_k^i} = \frac{q_s^i}{r_s^i} \tag{12}$$

where q_k^i and r_k^i denote the length and width of the i rd object in 2D space, respectively. q_i^i and r_l^i denote the length and width of the enclosing box of the i th object scene space, respectively. The category and number of objects can be obtained by target detection, and the type initialization of objects can be expressed as:

$$\hat{W}_i \in \left\{V_{z_i}^1, V_{z_i}^2, \cdots, V_{z_i}^n\right\} \tag{13}$$

where $\hat{W}_i$ denotes the volume information of the i nd object. $V_{z_i}^m$ denotes the n th model when the category is z_i . By laying out the objects in 3D space and fine-tuning them on this basis, more accurate and realistic models can be derived.

2.2.2. VR design system

In order to scale up the VR design library, the conversion of architectural models through the VR scene is carried out, and on the basis of constructing indoor architectural models, the architectural models are converted into components in the VR scene by combining the VR build rapid generation technology. After converting the building components into the VR database, the VR model library is utilized to build the virtual scene in the VR design system, mainly for the design of the 3 parts of the auxiliary scene, cross-terminal collaboration and roaming display [27]. Using VR technology to display the 3D building model image to the designers and customers, so that they can fully feel the real interaction with the building. The block diagram of the design of VR technology indoor is shown in Figure 2.

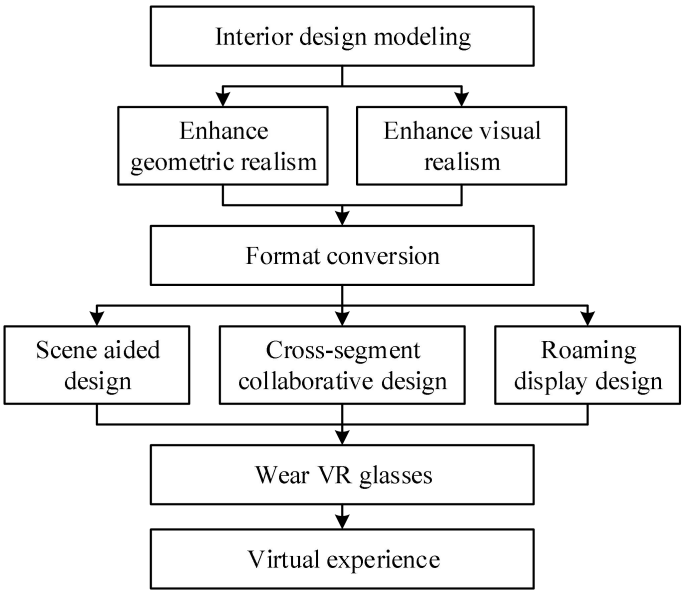


Figure 2. Design block diagram

3. Space layout optimization based on indoor virtual reality scenarios

3.1. Multi-objective planning model for indoor space layout

3.1.1. Initializing the layout

Layout object groups are assigned to each functional area, and in this paper we use a utilization-based filling method to traverse each functional area. The method flow is:

(a) Calculate the reference benchmark space utilization rate U_a and wall utilization rate U_w . Accordingly set the utilization rate range to $[U_i - \kappa_i, U_i + \kappa_i]$. Where κ_i is a small super parameter.

(b) Pre-assign a set of layout objects S_i to functional area V_i. Update space utilization U_a and wall utilization U_x.

(c) If U_a or U_a is out of range, reject this allocation and continue to examine the next set of layout objects, otherwise, accept this allocation. (d) When the allocation is complete, traverse the next functional area.

In the method flow, $U_\alpha = \sum_{s \in S} \frac{A(s)}{A_r}, U_\kappa = \sum_{s \in S} \frac{L(s)}{L_r}$. Where $A(\cdot)$ is the area operator, $L(\cdot)$ is the wall occupancy length operator, and A_r and L_r are the total area of the input contour and the length of the input contour, respectively. After the allocation is completed, the layout initialized according to the information of functional areas and groups recorded in the SG structure, the layout object group templates are randomly placed in the functional areas, and the objects against the walls are placed along the walls. During the placement process, overlapping of layout objects is avoided.

3.1.2. Energy function

After the existence of the layout information in the hierarchical graph structure (SG), it is necessary to create an efficient layout $SG^* = \arg\min_s \min \sum_{V_i \in V} E_r(S)$ where E_r is the weighted sum of the following energy terms, and the optimal layout solution is determined by minimizing this energy function.

(a) Space requirement. There needs to be open space around the objects in the group. Spacing also needs to be maintained between groups and between groups and walls as walking paths. Equation (1) shows the energy function for space constraint E_r:

$$E_s = \sum \varphi(\varepsilon - 2\varepsilon\sigma(\alpha(S_i) + \alpha(M_i) + \alpha(W_i))) \tag{14}$$

Where: $\alpha(S_i)$ is the area of spatial overlap between objects in the group. $\alpha(M_i)$ is the area of overlap between groups. $\alpha(W_i)$ is the area of spatial overlap between walls and groups. $\beta(S)$ is the total space requirement area. ε is a constant taken as 1.5. $\sigma(x)$ is the sigmoid function. $\varphi(x)$ is expressed as:

$$\varphi(x) = \frac{1}{x + \sqrt{x^2 + \tau^2}} \tag{15}$$

Where: τ is a small increment taken as 0.05.

(b) Filling degree requirement. The layout of most indoor scenarios requires full utilization of space, e.g., exhibits need to be evenly placed so that all parts of the exhibit have content. In shopping malls, expensive space has to be fully utilized to place the shelves, and so on. Eq. is the energy function for the padding constraint E_a.

$$E_a = \sum_{V_i \in V} \varphi(1 - \frac{\delta_{min}}{\delta_{V_i}}) \tag{16}$$

Where: δ_{ma} is the functional area V_i space utilization minimum threshold. δ_{v_i} is the actual space utilization. The area of space occupied by a group of layout objects is determined by the sum of their footprint and their space requirements.

(c) Accessibility requirements. There is a connectivity relationship between functional areas in the input contour, and the accessibility requirement is that the connected functional areas are accessible to each other, and the neighboring areas must share a sufficiently wide access. Eq. is the energy function of the accessibility constraint E_n:

$$E_a = \sum_{V_i, V_j \in V} \gamma(e^{\rho(d_{min} - d_{ij})} - 1) \tag{17}$$

Where: $d_{\min}$ is the minimum threshold for channel width. d_{ij} is the neighboring functional area channel width. $\rho(\cdot)$ is the ReLU function. γ is the 0-l parameter, and $\gamma = 1$ denotes functional area adjacency.

(d) Scaling constraints. In order to avoid one group occupying too much space and compressing the space of other layout objects, it is necessary to control the area share of the layout object group. Eq. is the energy function of the proportionality constraint E_p:

$$E_p = \sum_i \varphi(1 - \frac{\rho(\mu_i - \mu_{i\max})}{\mu_{i\max}}) \tag{18}$$

Where: μ_i is the distance between numbered neighboring objects in the group. $\mu_{i\max}$ is the maximum threshold of the distance parameter. $\mu_{\varepsilon max} = \iota_\varsigma + \iota_{\xi j} + Z, \iota_{si}$ and ι_{sj} are the space requirement widths of individual layout objects, Z is a constant, and the smaller Z is the stronger the constraint on the area share of the group. $\rho(\cdot)$ is the ReLU function.

The energy function is weighted to form a cost function as in Eq:

$$Er(S) = \omega_0 E_{,} + \omega_1 E_\alpha + \omega_2 E_\alpha + \omega_3 E_\rho \tag{19}$$

Where: ω_i is a positive weight and the value of the parameter in the experiment is $\omega_0 = \omega_1 = \omega_2 = \omega_3 = 1.0$.

3.2. Spatial layout optimization based on indoor virtual reality scenes

3.2.1. Particle Swarm Optimization Algorithm

Particle swarm optimization is an intelligent algorithm. Each individual in a particle swarm follows a simple behavior, i.e., a cumulative behavior that unfolds by emulating the successes of neighboring individuals. In the particle swarm optimization model, the individual optimal point represents the best position experienced by this individual from the beginning of the simulation, and the neighborhood optimal point is the best position experienced by all the neighbors of this individual, and these 2 optimal points are used as attractors. The individual has the memory of individual optimal point and neighborhood optimal point, and it uses the distance between the optimal point and the current position to adjust the position of the particle according to some simple rules in a certain proportion, so that the group gathers near the target within a certain number of iterations [28]. Its mathematical model is:

$$\begin{cases} v_{t+l} = wv_t + c_l rand()(p_{best} - x_t) + c_2 rand()(g_{best} - x_t) \\ x_{t+l} = x_t + v_{t+l} \end{cases} \tag{20}$$

Where: v_{t+1}, x_{t+1} denotes the velocity and position at the moment of $t+1$, respectively. w is the inertia factor. c_1 is the cognitive factor. c_2 is the social factor. p_{best} is the individual optimal position. g_{best} is the global optimal position.

3.2.2. Collaborative particle swarm optimization and spatial layout optimization process

(1) Co-PSO

Co-operative particle swarm (Co-PSO) is an optimization method after adopting the particle swarm into a cooperative architecture. In this paper, Co-PSO is applied to three-phase state estimation and the improved Co-PSO conceptual diagram is shown in Fig. 3. There are 2 groups of particle swarms in the figure, one is the main particle swarm and the other is the sub-particle swarm. The main particle swarm mainly controls the loop iteration and the sub-particle swarm mainly performs iterative optimization with the main particle swarm. The detailed interaction process between the main particle swarm and the sub-particle swarm is as follows:

a) The main particle swarm iteratively searches for a feasible solution, and the main particle swarm shares g_{best} to the sub-population for every N iterations.

b) The 1st sub-population gets g_{best} as its initial value and then searches the 1st dimension of the state variable. The search of the sub-population is performed on a one-dimensional space, and after the search is completed, the 1st sub-population sends the result of dimension 1 to the position of the corresponding dimension of g_{best}.

c) The main particle swarm sends new g_{best} vectors to the 2nd subpopulation, the 2nd

subpopulation takes the obtained g_{best} vectors as initial values and then performs the search in dimension 2, after the search is completed, sends the result of dimension 2 to the main particle swarm g_{best} corresponding dimensional location.

d) The sub-population sequentially obtains the latest g_{best} vectors, and after searching, sends the corresponding best dimensional solution to the main particle swarm.

e) After all the sub-populations are searched, the main particle swarm searches using the synergistic g_{best} values, and the main particle swarm searches in N_s dimensional space. Then re-enter to step (1) until the convergence condition is reached.

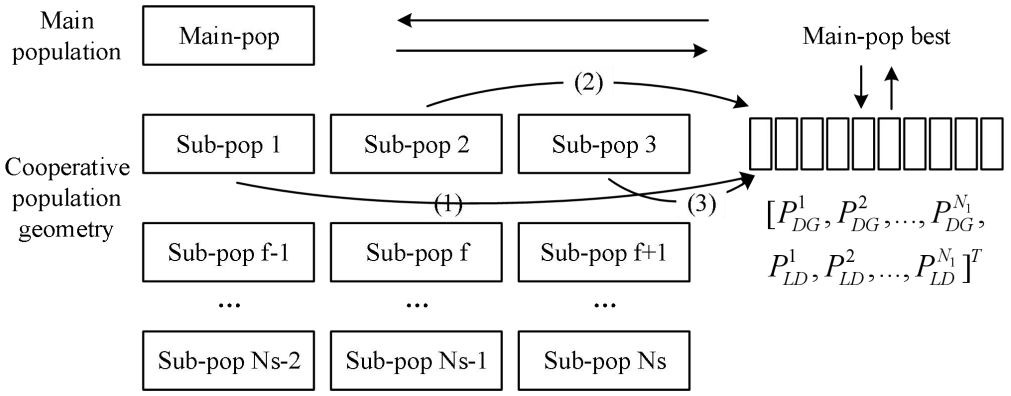


Figure 3. Improved Co-PSO concept diagram

(2) Initialization

X for all particles of the population, the initial state variable X_1 of the system particle population can be expressed as follows:

$$\begin{cases} X=[X_1,X_2,...,X_i,...,X_{N_s}], X_i=[x_j]_{l\times n}, i=1,2,3,...,N_s \\ x_j=rand()\cdot(P_{DG,\max}^j-P_{DG,\min}^j)+P_{DG,\min}^j, j=1,2,3,...,N_g \\ x_j=rand()\cdot(P_{LD,\max}^j-P_{LD,\min}^j)+P_{LD,\min}^j, j=N_g+1,N_g+2,N_g+3,...,N_g+N \end{cases} \tag{21}$$

Where: x_j is the location of the j nd state variable. $rand(\cdot)$ is a random number generator. N_s is the number of state variables. N_L is the number of loads for which estimation is to be performed. N_g is the number of distributed power sources for which estimation is to be performed.

(3) Fitness function

The equation constraints are satisfied during the particle swarm search process, and by penalizing the inequality constraints into the objective function, the fitness function of Co-PSO can be formed, which is expressed as follows:

$$\begin{cases} F(X_i)=f(X_i)+k_i\sum_{j=1}^{N_{\max}}(\max[0,-g_j(X_i)]) \\ g_j(X_i)<0, \quad j=1,2,3,...,N_{ucq} \end{cases} \tag{22}$$

where: $f(X_i)$ is the objective function of the state estimation problem. N_{ueq} is the number of inequality constraints. $g_j(X_i)$ is the set of all inequality constraints. k_i is the penalty factor for constraint overrun, which is taken as 1000 in this paper.

(4) Algorithm Flow

The flow of the proposed Co-PSO algorithm is shown in Fig. 4.

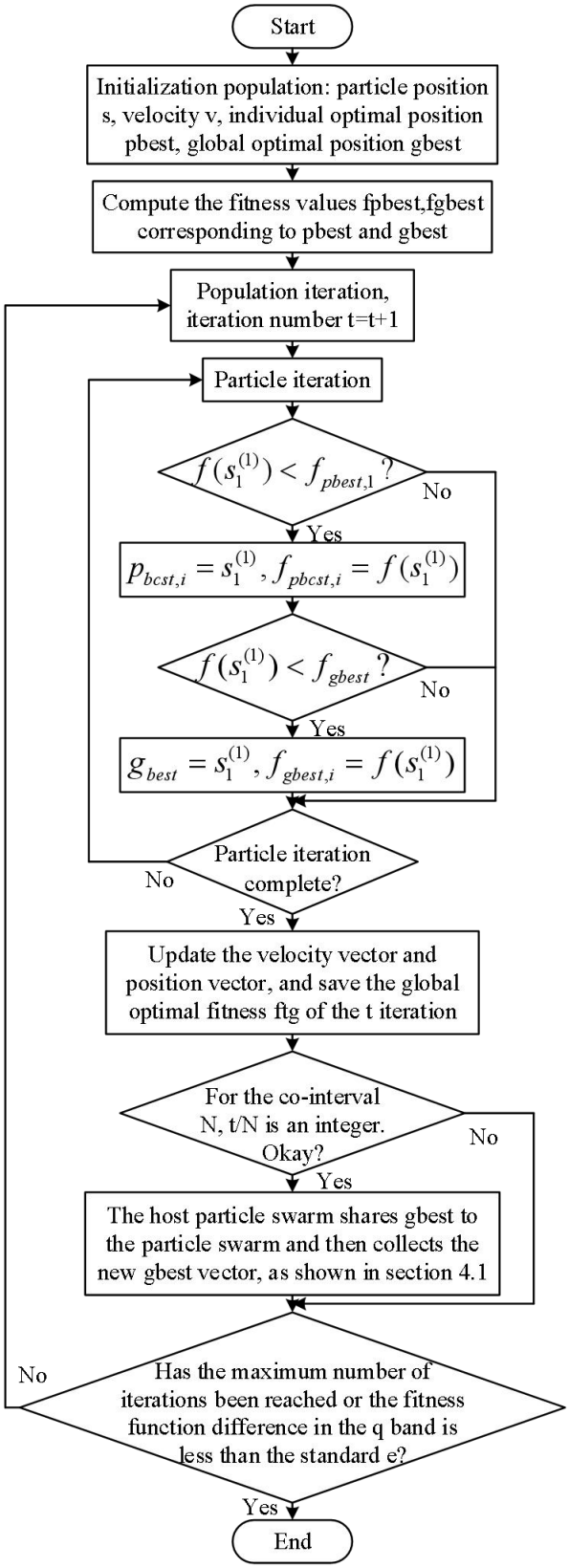


Figure 4. Co-PSO algorithm flow

4. Analysis of experimental results and applications

4.1. Simulation experiments based on synergistic PSO algorithm

4.1.1. Simulation experiments

In order to illustrate the performance of the COPSO algorithm, this paper uses it in the inertial right PSO and PSO algorithm with contraction factor, respectively, that is to say, it adds the downscaling collaborative optimization strategy proposed in this paper for the optimal position experienced by the particles in these two conventional PSO algorithms, and compares the performance of the algorithms before and after the addition of the experiments. In this paper, three commonly used standard nonlinear test functions are selected, which are: the Sphere function, the Rastrigrin function, and the Griewank function. The minimum value of all three standard functions is 0. The Sphere function is a single-peak function, and the others are multi-peak functions. In all experiments, the 3 functions were taken to be 30 dimensional and the number of microparticles in the cluster was set to be 30. An asymmetric selection of the initial values of the microparticles is used, and their limiting values for the range of initial values, target accuracy, microparticle velocity and position are shown in Table 1, with a maximum number of evolutionary generations of 5,000 for all experiments.

Table 1. The range of parameters of each test function

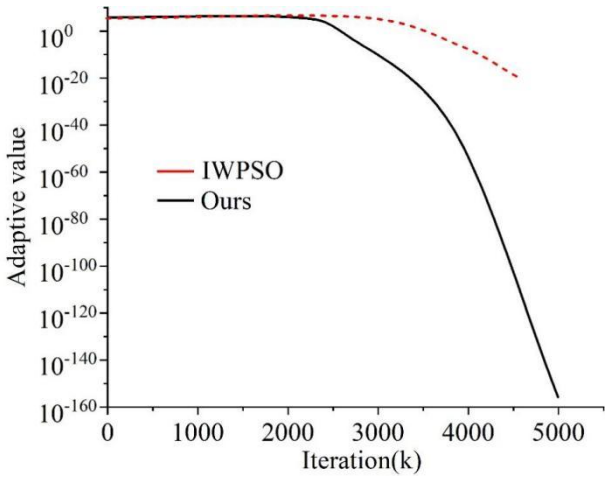
	Initial range	**Target accuracy**	$x_{\max} = v_{\max}$
f_1	$(50,100)^{30}$	0.00001	25
f_2	$(2.55,5.36)^{30}$	100	10
f_3	$(320,600)^{30}$	0.06	1300

All simulation experiments in this paper are done in MATLAB. In order to ensure the credibility of the experimental results, each algorithm is optimized for each test function for 20 independent simulation experiments, and the results of the comparative experiments are shown in Table 2, where IWPSO denotes the inertia right PSO algorithm. It_av, It_max, and It_min denote the average number of iterations, the maximum number of iterations, and the minimum number of iterations required to achieve the target accuracy in 20 independent experiments, respectively. F_av denotes the average function value (or adaptation value) of the solution that can be achieved after 5000 iterations. The 0 in the table indicates that the actual value is less than the minimum value displayed by MATLAB, and it can be assumed that the algorithm optimization has reached the global optimum. As can be seen from the table, the optimization performance of COPSO is improved for all three tested functions. For the Sphere function, COPSO first advanced the number of iterations to reach the target accuracy by 739 compared to the inertial right PSO, which is equivalent to increasing the convergence speed by almost 20%. For the Rastrigrin function, although COPSO improves the convergence speed significantly, the final convergence accuracy is not significantly improved, further indicating that COPSO is beneficial to improving the convergence speed of the PSO algorithm, but it cannot guarantee that the algorithm converges to the global optimum.The performance of COPSO is most prominent in the optimization of the Griewank function, which makes the algorithm converge more than four times faster and reaches the global optimum.

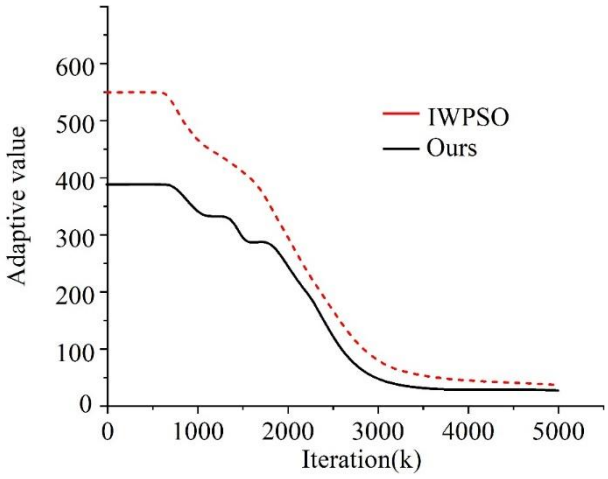
Table 2. Comparison experiment results

Test function		**It_av**	**It_max**	**It_min**	**F_av**
Sphere	Ours	2786	2855	2748	4.5269×10^{-155}
	IWPSO	3525	3667	3476	1.6355×10^{-32}
Rastrigrin	Ours	2321	2603	1861	27.6522
	IWPSO	2805	2929	2658	38.5122
Griewank	Ours	720	796	3361	0
	IWPSO	3198	716	3049	0.015

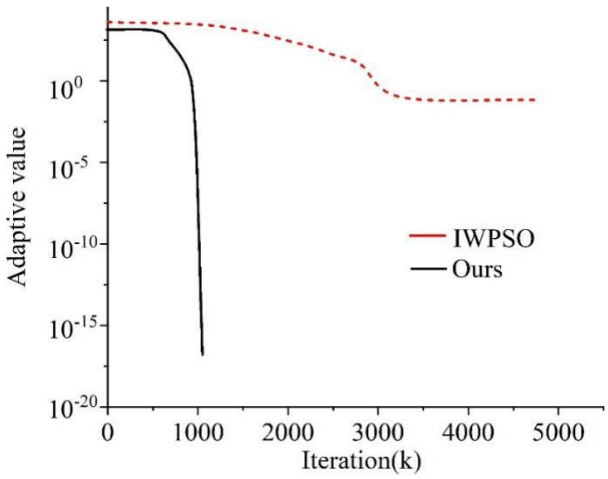
In order to show the experimental process more clearly, this paper plots the comparative experimental process in Table 2 as a curve as shown in Fig. The comparison curves are shown in Fig. 5 (Figs. a~c are Sphere function, Rastrigrin function and Griewank function, respectively). The two curves in each plot represent the change process of the average adaptation value of the solutions of the two algorithms in 20 independent experiments, respectively. As can be seen from the graphs, COPSO makes the algorithm convergence process significantly faster, and Figures (a) and (c) show that COPSO has significantly improved the algorithm convergence accuracy. Figure (b) shows that there is no significant difference between the two algorithms for the Rastrigrin function, but it can still be seen that COPSO converges significantly faster in the initial phase.



(a) Sphere function



(b) Rastrigrin function



(c) Griewank function

Figure 5. Comparison curve

The experimental results comparing this paper's algorithm with PSO with shrinkage factor are shown in Table 3. In the table CFPSO denotes the PSO algorithm with shrinkage factor. N/A indicates that the set target accuracy is not reached in all 20 independent experiments. From the table, it can be seen that for the Sphere and Griewank functions, COPSO shows a significant improvement in both

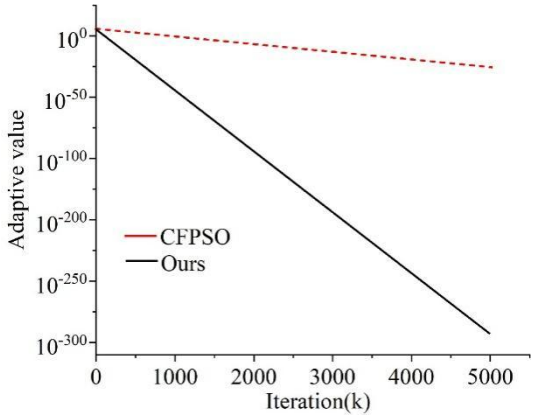
convergence speed and convergence accuracy, and the COPSO algorithm is more stable in terms of the maximum and minimum number of iterations to reach the target accuracy. However, for the Rastrigrin function, COPSO shows no advantage and even slightly inferior to CFPSO in terms of algorithmic accuracy.

Table 3. Comparison experiment results

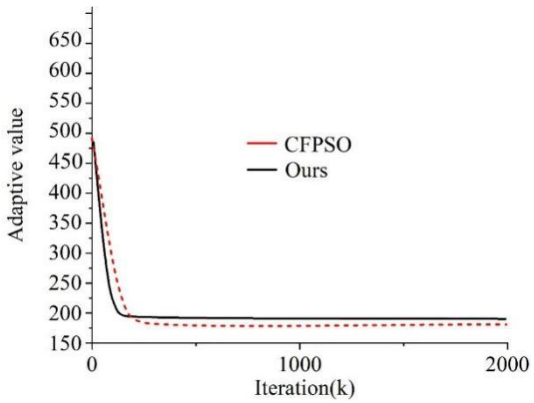
Test function		It_av	It_max	It_min	F_av
Sphere	Ours	172	182	162	7.9325×10^{-312}
	CFPSO	592	755	526	3.8966×10^{-75}
Rastrigrin	Ours	N/A	N/A	N/A	192.9322
	CFPSO	N/A	N/A	N/A	189.8421
Griewank	Ours	122	133	105	0
	CFPSO	391	439	366	0.0533

The results of the comparison experiments are shown in Fig. 6 (Figs. a~c are for the Sphere function, the Rastrigrin function, and the Griewank function, respectively). Since for the Rastrigrin function, the adaptation value of the solution no longer changes after 2000 iterations for both algorithms in all 40 independent experiments, only the first 2000 iterations are plotted in order to show the changes in the early stages of the algorithms. The significant advantage of the COPSO algorithm in optimizing the Sphere function and the Griewank function is evident from the plots. It is worth mentioning that, according to the analysis of the table data, for the Rastrigrin function, the COPSO algorithm is even inferior to the PSO with shrinkage factor, but it can be seen from Fig. (b) that COPSO still significantly improves the speed of convergence of the algorithm in the initial stage.

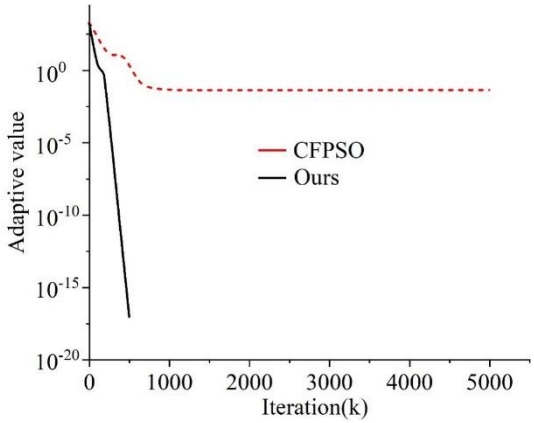
From the simulation results, it can be concluded that the COPSO algorithm can significantly improve the convergence speed and convergence accuracy, and even converge to the global optimum, which again shows that the COPSO algorithm does not guarantee convergence to the global optimum.



(a) Sphere function



(b) Rastrigrin function



(c) Griewank function

Figure 6. Comparison experiment results

4.1.2. Experiments in generating interior floor plan layouts

The genetic parameters, algorithm convergence age and time for the generation experiment are shown in Table 4. The initial population size is set to be 1000 and the number of iterations is 10000, i.e., the program terminates when the population evolves to the 10000th generation. The fitness curve of the in-set planar layout generation experiment is shown in Fig. 7. From the fitness curve, it can be seen that with the increase in the number of iterations, the average fitness of the population gradually decreases, the average environmental adaptability is gradually enhanced, and after the number of iterations exceeds 6750 generations, the average fitness maintains up and down in the neighborhood of 0.145, which indicates that even if the population continues to iterate, it is difficult to produce better individuals, and the algorithm has already converged. The better individuals of the population are extracted for decoding to generate a graphical representation of the planar layout.

Table 4. Genetic parameters, algorithm convergence times and time

Genetic parameter				**Algorithm convergence times and time**		
Initial population scale	Cross probability	Mutation probability	Evolutionary cycle(s)	Maximum evolutionary age	Mean age	Mean time(min)
1000	0.9	0.05	0.07	10000	6632	5.53

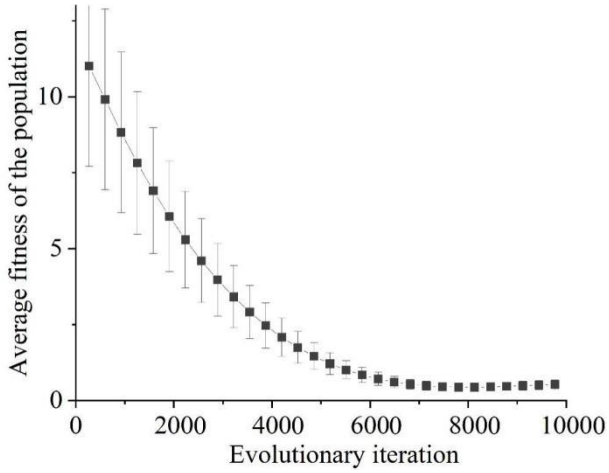


Figure 7. The fitness curve of the experiment in the inner plane layout

The results of the set plan layout generation are shown in Figure 8. The residential space is limited by a 600*600 grid, and the spatial contours of most of the results are relatively regular, and the area and size of the space basically meet the usage requirements. Individual generation results have shown high quality, the connection of functional space is in line with the functional topological relationship, and

presents spatial adaptability features such as spatial loop, LDK, and spatial centralization. However, there are still some defects in the generated results, such as the contour of individual spaces is not regular, and the width, depth and area are not optimized.

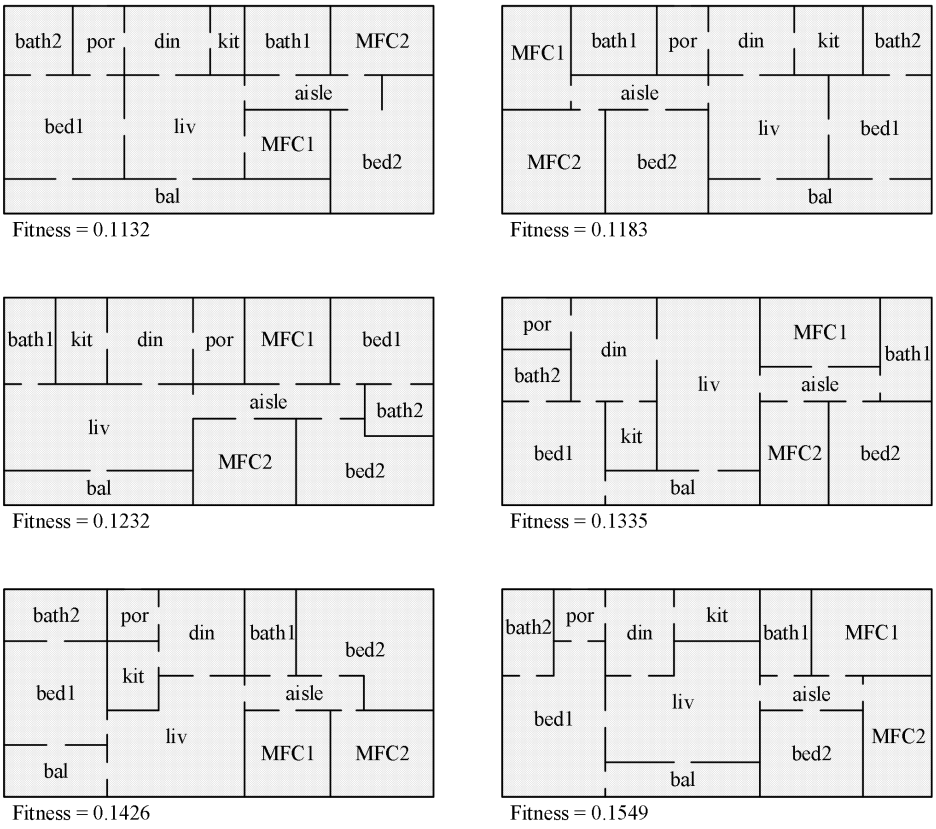
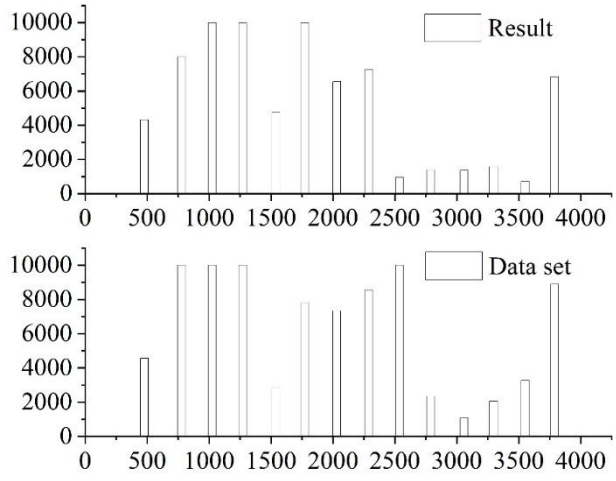


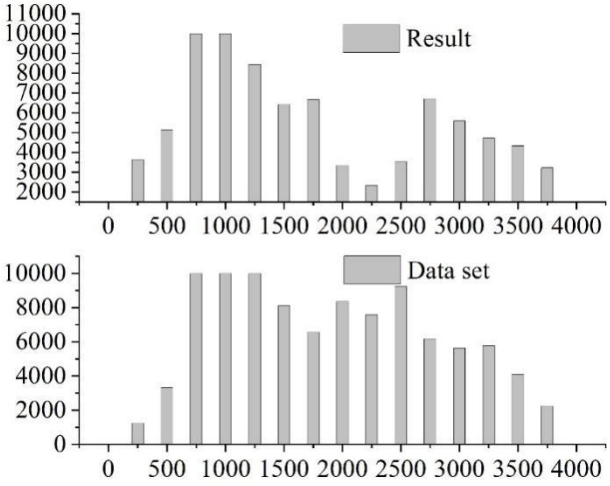
Figure 8. The layout of the inner plane is generated

4.2. Indoor scene layout effect display and analysis

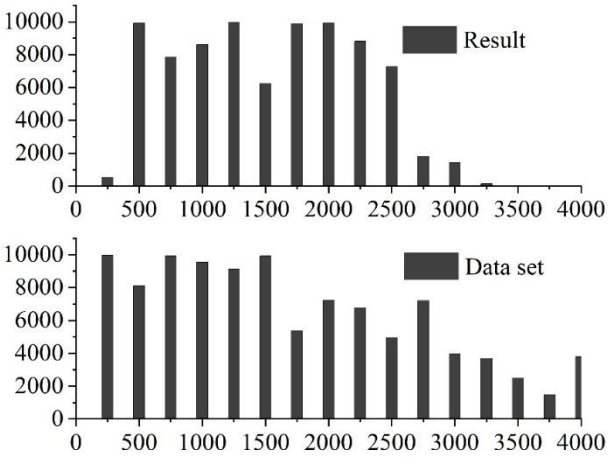
In order to evaluate the application effect of this paper's algorithm in interior design, this paper uses the histogram to compare the similarity between the optimization results and the target mood image dataset, and to judge the similarity between the optimization results' color themes and the color themes of the target mood image dataset, to verify the correctness and effectiveness of this paper's algorithm. The histogram of the optimization result and the target mood is shown in Figure 9 (a~e denote pleasure, horror, calm, melancholy and romance, respectively).



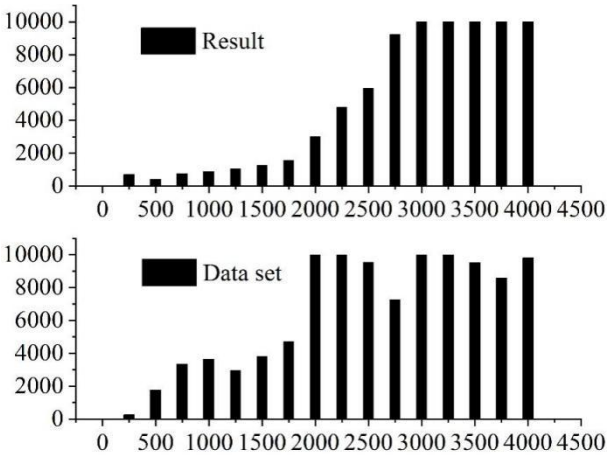
(a) Pleasure



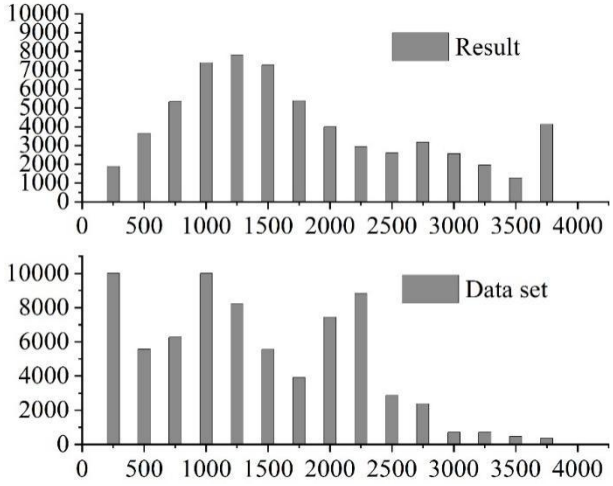
(b) Horror



(c) Peace



(d) Gloom



(e) Romance

Figure 9. Optimize the results and the target mood histogram

In this paper, histogram comparison is performed between the optimization results and its target emotion image dataset, and the results of histogram comparison are shown in Table 5. Less than 0.5 indicates a high degree of similarity when using the Bachmann distance for histogram similarity calculation, and greater than 0.5 indicates a high degree of similarity when using the correlation for histogram similarity calculation. As shown in the table, in the results of histogram comparison, all the values of Bachmann's distance are less than 0.5, and the correlation value is greater than 0.5, which indicates that the color theme of the optimization result of this paper's method has high similarity with the target emotional color theme, and proves that this paper's method is correct and effective for the creation of the emotional indoor space.

Table 5. Comparison of histogram

Emotions	Pleasure	Horror	Gloom	Peace	Romance
Barsmith distance	0.449	0.499	0.494	0.451	0.493
Correlation	0.569	0.533	0.565	0.745	0.547

In addition, related studies have developed a metric for matching colors with emotions. In the experimental results of this paper's method, the dominant color of the pleasure scene is bright colors such as yellow, the dominant color of the horror and melancholy scenes is dark colors such as black, the calm scene has white as the dominant color scheme, and the dominant color of the romantic scene is warm colors such as pink, which is in line with the corresponding emotional criteria. The color scheme of the experimental results of this paper has a good match with the target emotional color theme, which further indicates that this paper's method for rendering virtual indoor scenes emotionally is accurate and reliable.

In order to test the effectiveness of this paper's method in different aspects, 50 participants were invited to conduct two independent user studies in this section. The participants were 24 males and 26 females, ranging in age from 18-40 years old with an average age of 25 years old, and the participants consisted of 20 students and faculty members majoring in computer science, 20 students and faculty members from art colleges, and 10 members of the community.

User Study 1: Optimization results scoring and classification. Participants randomly viewed different indoor scenes optimized by the methods in this paper and completed the following three tasks.

Task 1: Scoring. Participants rated the five emotion optimization results generated by this paper's method using a 1-5 Likert scale. The ratings reflected participants' assessment of how well the optimization results of this paper's method matched the target emotions, with higher scores indicating that participants believed the optimization results matched the target emotions better. For consistency, each emotion was tested in a bedroom scenario throughout the task.

Task 2: Emotion Recognition. This task was designed to confirm whether participants were able to identify the target emotion in the scene optimization results. Five emotion optimization results were given, and participants were asked to select the emotion that most closely matched the scene from a list; the emotion list also contained an additional other option for situations where none of the five emotions

matched.

Task 3: Authenticity. Similar to Task 1, participants rated the scenes based on how realistic the colors of the optimized scenes were. Scenes were optimized with an authenticity weight of 1.0 and an emotion weight of 0.0 in both the dining room and living room scenes.

User Study 1: Optimization results scoring and classification. The average score for all scenes in Task 1 and Task 3 was 3.932 out of 5.000, indicating that participants generally agreed that the scene optimization results matched the target emotions.

Scene emotion recognition is shown in Figure 10. Of the five optimized scenarios used in Task 2, the matching of the scenarios to the emotions received the most votes in each case, with the exception of the scary restaurant. The user research data showed that the difference between melancholy and horror was not significant, and this paper compares the histograms of the datasets and optimization results for both melancholy and horror emotions.

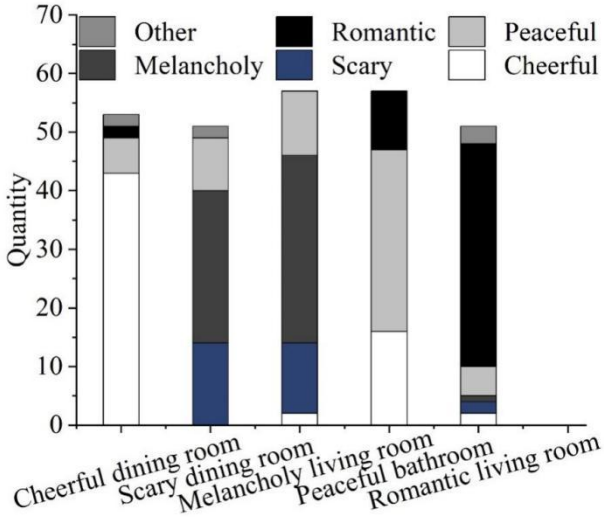


Figure 10. Scene emotion recognition

The results of the histogram comparison are shown in Table 6. As shown in the table, in the results of the histogram comparison, all the barometric distance values are less than 0.5 and the correlation values are greater than 0.5, which indicates that the datasets and optimization results of the two emotions are somewhat similar. Through the research and analysis, it is found that melancholy and horror have the same semantics to a certain extent, most of the scenes of melancholy and horror are dark, and the two emotion datasets used to train the classifier have a certain degree of similarity, so that the results have a certain degree of rationality and interpretability.

Table 6. Comparison of histogram

	Data set	**Results**
Barsmith distance	0.472	0.479
Correlation	0.511	0.546

In this paper, paired t-tests were conducted on the data from User Research 1. The p-values of the t-tests conducted on the quantitative task scenarios are shown in Table 7. As shown in the table, the p-values of the t-tests between the two scenarios are mostly greater than 0.05, indicating that there is no significant difference. Since the mean score was 3.933, it indicates that participants found the optimization results of this paper convincing. most of the t-tests with p-values lower than 0.05 came from comparing the romantic bedroom scenario, which scored significantly lower compared to the other scenarios with higher scores, and treating this scenario as an outlier. The analysis revealed that many of the furniture objects in the romantic mood image dataset were dark in color, and there were many dark scenes due to low light. In future studies, further research could be conducted to analyze finding a method for metrics of situations in which dark colors are more appropriate for romantic moods. Statistical analysis of User Study 1 showed that participants generally found the optimization results convincing. In the five scenes optimized using this paper's method, participants were able to accurately identify the target mood of the scenes and gave good ratings of this paper's optimization results in the user study.

Table 7. The p test of the quantitative task scenario is tested

Scene	Real restaurant	Real living room	Happy bedroom	Horror bedroom	Blue bedroom	Quiet bedroom	Romantic bedroom
Real restaurant	—	0.537	0.487	0.651	0.915	0.145	0.008
Real living room		—	0.19	0.184	0.605	0.398	0.075
Happy bedroom			—	0.778	0.439	0.027	0.001
Horror bedroom				—	0.601	0.082	0.003
Blue bedroom					—	0.173	0.024
Quiet bedroom						—	0.108
Romantic bedroom							—

User Study 2: Ablation Experiment. In this user study, two questions about the optimization results of the methods in this paper were tested, Q1: Which emotion does the scene best match? and Q2: Is the color distribution similar to the common color distribution of that type of scene? The results of User Study 2 are shown in Figure 11.

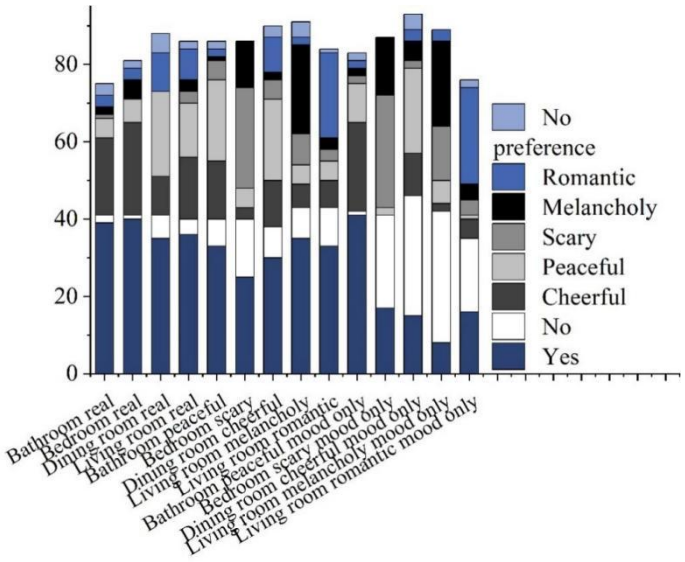


Figure 11. User study 2 results

For Q1, the null hypothesis is that participants vote the same number of votes for each emotion. For Q2, the null hypothesis is that for scenarios other than real scenarios, participants vote the same number of votes for the answer to the question of whether the color distribution is similar to the common color distribution for that type of scenario. For real scenarios, this paper assumes that participants will vote by an absolute majority in favor of this paper's optimization for realism that the texture colors of the scene's objects are similar to the objects' texture colors in the real world. The chi-square test p-values are shown in Table 8 (wM and wP are the emotional cost weights and real cost weights, respectively). As shown in the table, for Q1, the p-value is much lower than 0.05, and the results indicate that there are differences in voting for each emotion. For Q2, the results are more diverse, in real scenarios, participants voted in favor of the optimized color of the scenario to satisfy authenticity with absolute dominance, and in other cases, participants voted more evenly between the opposing option and the favoring option, indicating that the color distributions are similar to the common color distributions of this type of scenarios, which further verifies the validity and correctness of the authenticity metric algorithm proposed in this paper.

The user study2 research shows that this paper's method is not only able to generate scenarios that match the target emotion, but also generates optimized scenarios that satisfy the authenticity needs of

the scenarios, and this paper's method can satisfy different needs by adjusting the weights of the emotional cost and the authenticity cost.

Table 8. The card is tested for P

Scene	Weight	Q1 p-value	Q2 p-value
Real bathroom		<0.01	0.045
Real bedroom		<0.01	0.25
Real restaurant	wM=0.0	<0.01	<0.01
Real living room	w*P*=1.0	<0.01	0.045
Quiet bathroom		<0.01	<0.01
Horror bedroom	wM=0.7	<0.01	0.065
Pleasant restaurant	w*P*=0.3	<0.01	<0.01
Blue living room		<0.01	<0.01
Romantic living room		<0.01	<0.01
Quiet bathroom (emotion alone)		<0.01	<0.01
Horror bedroom (emotion alone)	wM=1.0	<0.01	0.12
Pleasant restaurant (emotion alone)	w*P*=0	<0.01	0.02
Blue living room (emotion alone)		<0.01	<0.01
Romantic living room (emotion alone)		<0.01	0.8

5. Conclusion

The study models the indoor virtual reality scene based on VR technology and introduces a collaborative particle swarm optimization algorithm on this basis to achieve spatial layout optimization. Finally, several types of experiments are selected for method evaluation, and the generated results prove the effectiveness of the method in this paper.

For the Sphere function, the number of iterations to reach the target accuracy is advanced by 739 for COPSO compared with inertial right PSO, and the results of multiple simulation comparison experiments show that the improved collaborative PSO algorithm proposed in this paper is conducive to improving the speed of convergence of the algorithm, and it can be converged to the global optimum in some problems.

Through the experiments on this paper's algorithm in four different indoor scenes, it is found that after the histogram similarity calculation using the Bachmann's distance, all the Bachmann's distance values are less than 0.5, and the correlation values are all greater than 0.5, which can be obtained that the color theme of the optimization results of this paper's method has a high degree of similarity with the target color theme, which verifies the correctness and effectiveness of this paper's algorithm.

References

1. Kaleja, P., & Kozlovska, M. (2017). Virtual reality as innovative approach to the interior designing. Selected Scientific Papers-Journal of Civil Engineering, 12(1), 109-116.
2. Kalantari, S., & Neo, J. R. J. (2020). Virtual environments for design research: Lessons learned from use of fully immersive virtual reality in interior design research. Journal of Interior Design, 45(3), 27-42.
3. Zhu, Z., & Du, Y. (2021). Research on interior design optimization based on virtual reality technology. In Journal of physics: Conference series (Vol. 1746, No. 1, p. 012063). IOP Publishing.
4. Pober, E., & Cook, M. (2019). Thinking in virtual spaces: impacts of virtual reality on the undergraduate interior design process. International Journal of Virtual and Augmented Reality (IJVAR), 3(2), 23-40.
5. Dai, Y. (2023). 3D Interior Design System Model Based on Computer Virtual Reality Technology. Journal of Electrical Systems, 19(4).
6. Guo, L. (2023). Simulation evaluation of virtual reality in interior design effect display and practice mode innovation. Soft Computing, 27(12), 8371-8380.
7. Tang, Y. M., Lau, Y. Y., & Ho, U. L. (2023). Empowering digital marketing with interactive virtual reality (IVR) in interior design: Effects on customer satisfaction and behaviour intention. Journal of Theoretical and Applied Electronic Commerce Research, 18(2), 889-907.

8. Joy, E., & Raja, C. (2024). Digital 3D modeling for preconstruction real-time visualization of home interior design through virtual reality. Construction Innovation, 24(2), 643-653.
9. Cho, J. Y., & Suh, J. (2023). Spatial ability performance in interior design and architecture: Comparison of static and virtual reality modes. Buildings, 13(12), 3128.
10. You, W., Shao, Y., Zheng, Z., Lu, Y., Yang, C., Zhou, Z., & Sun, L. (2023). Old house new home: Facilitating interior design with RedesignUS in virtual reality. Displays, 80, 102555.
11. Wang, R., & Huang, Y. (2022). Application of 3d software virtual reality in interior designing. Mobile information systems, 2022(1), 5315262.
12. Neves, L., & Pombo, F. (2020, May). Virtual Reality for interior design history. The Ofir house as experimental project. In Meeting of Research in Music, Arts and Design (pp. 461-471). Cham: Springer International Publishing.
13. Yu, D. (2020, April). Interior landscape design and research based on virtual reality technology. In Journal of Physics: Conference Series (Vol. 1533, No. 3, p. 032038). IOP Publishing.
14. Zhang, J., Piumsomboon, T., Dong, Z., Bai, X., Hoermann, S., & Lindeman, R. (2020, April). Exploring spatial scale perception in immersive virtual reality for risk assessment in interior design. In Extended abstracts of the 2020 CHI conference on human factors in computing systems (pp. 1-8).
15. Wu, P., Liu, Y., Chen, H., Li, X., & Wang, H. (2025). VR-empowered interior design: Enhancing efficiency and quality through immersive experiences. Displays, 86, 102887.
16. Gao, Z., & Huang, J. (2018). DESIGN AND APPLICATION OF VIRTUAL REALITY TECHNOLOGY IN 3D INTERIOR DESIGN AND SIMULATION SYSTEM. Academic Journal of Manufacturing Engineering, 16(3).
17. Iftekhar, M. W., & Prajapati, R. J. (2023). Impact of Virtual Reality and Eco-Friendly Interior Design Tool on Carbon Emission. Journal of Online Engineering Education, 14(1s), 32-40.
18. Han, A. R., & Park, T. (2017). Application for Furniture Arrangement based on Virtual Reality to Share Interior Design in Real-time. Journal of Digital Contents Society, 18(2), 249-256.
19. Sun, Y. (2022). Design and optimization of indoor space layout based on deep learning. Mobile Information Systems, 2022(1), 2114884.
20. Wu, W., Fan, L., Liu, L., & Wonka, P. (2018, May). Miqp‐based layout design for building interiors. In Computer Graphics Forum (Vol. 37, No. 2, pp. 511-521).
21. Wu, W., & Feng, Y. (2022). Interior space design and automatic layout method based on CNN. Mathematical Problems in Engineering, 2022(1), 8006069.
22. Chen, C., Li, S., & Huang, J. (2024). Research on Optimization of Interior Space Design Based on Machine Learning Algorithm. In International Symposium on World Ecological Design (pp. 362-369). IOS Press.
23. Wu, S., & Han, S. (2023). System evaluation of artificial intelligence and virtual reality technology in the interactive design of interior decoration. Applied Sciences, 13(10), 6272.
24. Juan, Y. K., Chi, H. Y., & Chen, H. H. (2021). Virtual reality-based decision support model for interior design and decoration of an office building. Engineering, Construction and Architectural Management, 28(1), 229-245.
25. Wenshan Lv,Haifeng Zhang,Weiren Chen,Xiaoming Li & Shengtian Sang. (2024). A Point Cloud Registration Algorithm Based on Weighting Strategy for 3D Indoor Spaces. Applied Sciences,14(12),5240-5240.
26. Minglei Li,Mingfan Li,Min Li & Leheng Xu. (2025). Building Lightweight 3D Indoor Models from Point Clouds with Enhanced Scene Understanding. Remote Sensing,17(4),596-596.
27. Yang Liu. (2025). Indoor thermal comfort and VR landscape design of low-energy buildings based on interactive design technology. Thermal Science and Engineering Progress,59,103349-103349.

28. Li Dong,Di-quan Li & Fei-bo Jiang. (2018). A two-stage CO-PSO minimum structure inversion using CUDA for extracting IP information from MT data. Journal of Central South University,25(5),1195-1212.