Web3D-based Automatic Furniture Layout System Using Recursive Case-based Reasoning and Floor Field

Peihua Song [1], Youyi Zheng [2], *Jinyuan Jia[1] and Yan Gao [3]

[1] School of Software Engineering, Tongji University, Shanghai 201804, P.R.China

sph2000@126.com and jiajytj@126.com

[2] State Key Lab of CAD & CG, Zhejiang University, Hangzhou, 310058, P.R.China

youyizheng@zju.edu.cn

[3]Electrical and Computer Engineering, Clarkson University, Potsdam, NY, 13676, U.S.

gaoy@clarkson.edu

Abstract: Furniture layout in a virtual 3D scene is an important and challenging task, as it is time-consuming and requires experience. To address this issue, we propose automatic furniture layout algorithms to help users to rapidly generate reasonable layout results. Specifically, our algorithms divide a scene layout into four layout modes, namely, coupled mode, enclosed mode, matrix mode, and circular mode. Then each model is solved independently. The coupled mode is solved using recursive techniques and case-based reasoning. The enclosed mode is solved using floor field. The distance and angle among the furniture are determined by ergonomics guidelines. Finally, the layout results of the scene can be obtained by combining the solutions from those layout modes. For a room with non-rectangular floor, our algorithms can also handle this case using shape standardization techniques. Based on our algorithms, an online 3D furniture layout system is developed. Many experiments are conducted on the system with the real interior design cases, and we compared our algorithms with other popular algorithms. The experimental results show that our algorithms are efficient, and can meet the real response requirements of online furniture layout.

Keywords: Furniture layout; Case-based reasoning; Floor field; Recursive techniques

1 Introduction

The scene design can be widely found in many domains such as city planning, architectural design and 3D scene modeling [11-14], which is a research focus in computer graphics and optimization techniques. As an important component of scene design, furniture layout has gained more attention in recent years, often existing in 3D interior design and game scene modeling. Traditional manual furniture layout is not an easy task because it is time consuming and tedious. Furniture layout often performed in a virtual 3D environment and each piece of furniture has six degrees of freedom, thus

it is inconvenient to manually manipulate the furniture with a mouse. Moreover, for ordinary users without interior design expertise and experience, they need many efforts to get a reasonable furniture layout result; for interior designers, repeating the similar operation of arranging furniture is also a tedious work.

To overcome these problems, researchers have proposed several efficient algorithms for automatic furniture layout, including algorithms based on optimization model [1-7], constraints [15-19] and case library [20-24]. However, most of existing algorithms of furniture layout have a limitation either in the number of furniture or in the shape of the layout room. For example, Yu et al. [1] defined a layout evaluation function, and then searched the suitable location to arrange furniture using simulated annealing. But for the layout problem of the large quantity of furniture and area of the room, the number of iterations increase and the convergence speed becomes slow as the large search range, resulting in a long run time [1, 5]. Algorithms based on case library or constraints rarely consider the impact of the doors and windows on the layouts [17, 23, 24], requiring the floor shape of the room to be rectangular. Meanwhile, with the rapid development of 3D technique in web environment [8-10], there emerge some online 3D interior design systems. Those systems are popular because the users can perform online design without the need of installing any offline software. This trend motivates us to extend existing offline furniture layout to online furniture layout, and online furniture layout requires that the furniture layout algorithms to have real time response and can generate reasonable layout results.

In this paper, we tackle the problem of automatic furniture layout in the web 3D environment. The general framework of our algorithms is as follows. The whole room layout divided into four furniture layout modes, namely, coupled mode, enclosed mode, matrix mode, and circular mode, solving those modes respectively. Then we combine the solutions from those modes to obtain the layout result of the room. The major contributions of this paper are listed below.

(1) We propose the respective shape standardization algorithms for the bedroom and living room, which is the preprocessing for a non-rectangular room.

(2) To reduce layout complexity, we introduce the four furniture layout modes mentioned previously and present the respective algorithms for them. All those algorithms utilize ergonomics guidelines to determine the distance and angle among the furniture. Coupled mode and enclosed mode are solved using recursive case-based reasoning and floor field, respectively.

(3) Our framework is highly extensive that we have added four furniture layout algorithms for living room, bedroom, classroom, and restaurant.

(4) A Web3D-based interactive system for automatic furniture layout is presented based on our algorithms. The system can give reasonable results of furniture layout when users press the specified button in the system. Our algorithms have been implemented with real interior design cases, and compared with other algorithms. The experimental results show that the algorithms are effective, and can meet the real response requirements of online layout.

The rest of this paper is organized as follows. Section 2 reviews some related works. In Section 3, we give a systematic overview and detailed description of the general framework of furniture layout algorithms. Section 4 presents the shape standardization algorithms for the room. Section 5, 6, 7, and 8 describe the four

layout modes in detail and the respective algorithms for them. Section 9 presents the respective furniture algorithms for various rooms based on the general framework. Section 10 reports the experimental results and gives a discussion, and finally, we conclude our paper in Section 11.

2 Related work

The virtual 3D scene design and synthesis have been widely used in the outdoor scene. For example, Merrell et al. [11] proposed an approach for automatically generation of building layouts using a Bayesian network. Esch et al. [14] presented a modeling framework for modeling large street networks. The user can create street network from scratch using the framework. In this section, we focus on the related work of the furniture layout methods in the indoor scene，and those methods can be broadly divided into three categories as follows.

2.1 Algorithms based on optimization model

Many intelligent optimization algorithms are used to solve the furniture layout problem, such as, simulated annealing algorithm, genetic algorithm and particle swarm optimization algorithm. The general framework of those algorithms is as follow. First those algorithms define an evaluation function by furniture layout rules which include the distance and angle among the furniture. Then, the layout results are obtained by searching the solutions using intelligent algorithms. Yu et al. [1] presented an algorithms based on the simulated annealing algorithm. The evaluation function of the algorithm mainly considers accessibility, visibility, pathway constraints and so on. Merrell et al. [2] presented an interactive furniture layout system based on interior design guidelines. The algorithm incorporates the layout guidelines as terms in a density function, and then generates layout suggestions. Chen et al. [3] presented a hierarchical optimization strategy, and the strategy combined with particle swarm to solve the furniture layout problem, improving computing efficiency. Akase et al. [4] introduced a furniture layout cost function based on ergonomic, and used interactive evolutionary to solve the cost function. Liu et al. [5] introduced a composite model for home furnishing generation and used the genetic algorithm to solve the model.

Although these algorithms can generate reasonable results, the running time of these will increase rapidly as a scene contains a lot of furniture. To reduce layout complexity and running time, in this paper we divide a sense layout into fours layout modes and solve the layout modes respectively.

2.2 Algorithms based on constraints

Bukowski et al. [15] presented a software framework to help designers manipulate 3D objects with a 2D cursor. The framework used the object associations to determine the location of each object. Smith et al. [16] also presented techniques to restrict object motion to assist users in 3D environment, which considered various constraints among the objects, such as dual constraints. Although [15] and [16] considered some

geometric and physical constraints in designing, their whole design process was user-driven and still time-consuming. Kjølaas [17] used a set of default layout templates to solve the furniture layout problem. The algorithm is limited to the rectangular room. Germer et al. [19] presented an approach to arrange furniture in large virtual indoor scenes using the agent-based interiors, but the furniture layout case of multiple doors in a room was not considered in this approach.

2.3 Algorithms based on case library

Fisher et al. [20] presented a method for synthesizing 3D object arrangements from examples. The method synthesizes a diverse set of plausible new scenes using machine learning method. Xu et al. [21] proposed a framework that automatically created a 3D scene from a sketch drawing. The framework obtains the corresponding 3D model by searching the model library. Xu et al. [22] presented a system for automatically synthesizing 3D interior scenes for a room shape entered by users. The algorithm arranges 3D models based on the relationships between the models and room. In the authors' previous work [24], an algorithm for furniture layout was proposed, but the algorithm also limited to the rectangular shape room.

Those algorithms rarely consider the layout of problem of a non-rectangular room with multiple doors. Indeed, there exist many living rooms which are non-rectangular rooms and have multiple doors. To deal with those problems, this paper adopts shape standardization techniques and case-based reasoning.

3 System Overview

The main function of the system is to automatically generate a furniture layout result to help user complete the work of interior design. The system developed with WebGL and C sharp language in Visual Studio 2015. As shown in Fig.1, the system consists of three modules: user input module, automatic furniture layout module and results output module.

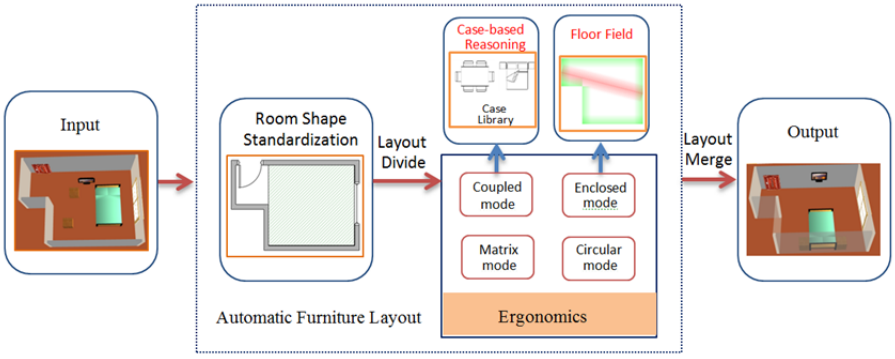


Fig. 1 The overview of the proposed system

3.1 The functionality of each module

The functions of the three modules are listed below. (1) The function of user input module is as follows. A user can input a 2D floor plan, and then the system generates a 3D room and interactive environment. Next, the user can add, move, delete, and rotate the 3D furniture in the room. (2) Automatic furniture layout module is the core module of the system, which implements the respective furniture layout algorithms for living room, bedroom, classroom, and restaurant. (3) In the results output module, the user can adjust the camera position and angle, obtaining a 2D picture of layout form the current 3D environment.

3.2 General framework of furniture layout algorithms

We have given a simplified description for the general framework of our algorithms in Section 1, and here we give a detailed description of the framework. The framework is a general process of furniture layout, thus we need to adjust those phases according to the function of the scene. The framework is composed of 4 steps as follows.

Step 1: If the floor shape of a room is non- rectangular, then we extract the rectangular layout region from the floor shape to reduce the complexity of the whole room layout.

Step 2: According to the type of room inputted by the user, the room layout is split into four furniture layout modes, namely, coupled mode, enclosed mode, matrix mode, and circular mode.

Step 3: Solve these selected modes respectively. The coupled mode is solved using recursive techniques and case-based reasoning. The enclosed mode is solved using floor field. The distance and angle among furniture can be set according to ergonomics.

Step 4: Merge the solutions from these layout modes into a solution, namely, the layout result of the room.

4 Room shape Standardization

Room shape standardization aims to extract a rectangular region from the floor of a living room or bedroom. The rectangular region is used to arrange the bed in the bedroom or sofa in the living room. The respective shape standardization algorithms for the bedroom and living room are given below.

4.1 Bedroom shape standardization

Let polygon $\mathrm{P} = p_1 p_2 \ldots p_n$ denote the floor shape of a bedroom. The polygon P contains two sets: a set of vertices $\{p_1, p_2 \ldots p_n\}$ and a set of line segments $\{p_1p_2, p_2p_3 \ldots p_{n-1}p_n, p_np_1\}$. The segments denote the walls in the bedroom. The point p_{door} and point p_{window} denote the position of a door and window, respectively. The detailed description of bedroom shape standardization algorithm is as follows.

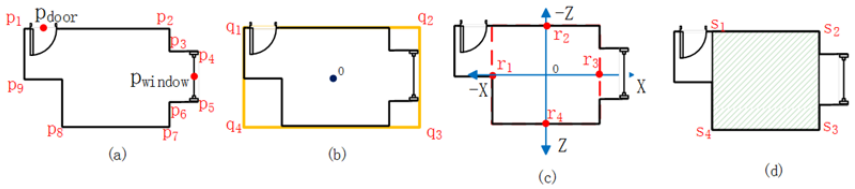


Fig. 2 (a) The shape of a bedroom (b) The bounding box of the room (c) The intersections of the line segments and the coordinate axis (d) The rectangular layout region of the bedroom

Algorithm 1 Bedroom shape standardization
Input: A bedroom floor shape $P = p_1 p_2 \ldots p_n$
Output: The rectangular layout region

Step 1: Find the polygon's bounding box $Q = q_1 q_2 q_3 q_4$ and its center point O.

Step 2: Establish a two-dimensional coordinates whose origin is the point O. Let the positive direction of the coordinate X axis be from the point O to the window position point p_{window}.

Step 3: Extend the length of the line segments. Find the intersections of the line segments and the coordinate axis.

Step 4: Find the point r_1 from those intersections. The point r_1 meets the following two constraints at the same time. (1) The x coordinate value of point r_1 is smaller than zero. (2) The x coordinate value of point r_1 is the largest in the intersections which meets the constraint (1). Similarly, find the point r_2, point r_3, and point r_4.

Step 5: Construct the rectangular region with the point r_1, point r_2, point r_3, and point r_4 and point O, and then return the layout region $s_1 s_2 s_3 s_4$.

Figure 2 illustrates the behavior of the algorithms for a non-rectangular bedroom. As shown in the figure, the rectangular region $s_1 s_2 s_3 s_4$ is the layout region. Then the furniture of the bedroom can be arranged in the rectangular region.

4.2 Living room shape standardization

As shown in Fig. 3a, let polygon $P = p_1 p_2 \ldots p_n$ denotes the floor shape of a living room. The living room shape standardization algorithm is given below.

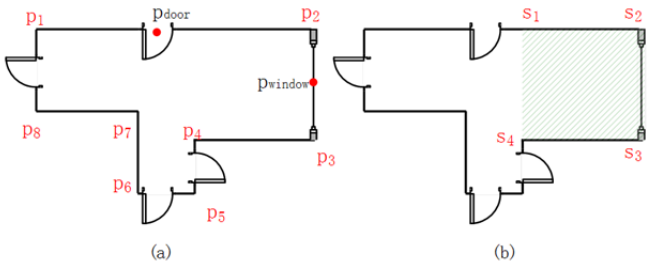


Fig. 3 (a) The shape of a living room (b) The rectangular layout region for the living room

Algorithm 2 Living room shape standardization
Input: A living room floor shape $\mathrm{P} = p_1 p_2 \ldots p_n$
Output: The rectangular layout region

Step 1: Find the three walls p_1p_2, p_2p_3, and p_3p_4 which are adjacent to the window.

Step 2: Find all the door position points on the wall p_1p_2 and p_3p_4; Put these position points into a set D_{door}. Then calculate the distance between the point p_{window} and the point p_1, point p_4 and the points in the set D_{door}, respectively. Find the point p_{min} to be of minimum distance. The point p_{min} is p_4 as shown in Fig. 3b.

Step 3: Construct the rectangular layout region $\mathrm{s_1s_2s_3s_4}$ using the point $\mathrm{p_2}$, point $\mathrm{p_3}$ and point p_{min}, and then return the region $s_1s_2s_3s_4$ as shown in Fig. 3b.

In this algorithm, if the length of the region $s_1s_2s_3s_4$ is smaller than the length of a sofa, resulting in the region cannot be arranged the sofa in the living room, and then we could expand the region by increasing the length of the region.

5 Furniture layout modes

A furniture layout mode refers to the layout relationship between the furniture and furniture or a room. We define four layout modes, namely, coupled mode, enclosed mode, matrix mode and circular mode. (1) Coupled mode requires that a certain distance and angle among furniture need to be maintained between the furniture, such as the bed and bedside cabinets (see Fig. 4a). It can be divided into two categories: furniture and furniture coupled mode, furniture and room coupled mode. (2) Enclosed mode requires that the furniture first is arranged at the corner, and then is arranged along the wall, such as a writing desk in the room (see Fig. 4b). (3) Matrix mode requires the furniture to be arranged in a rectangular region, such as the desks in the classroom (see Fig. 4c). (4) Circular mode requires the furniture to be arranged around circular furniture, such as the circular dining table and chairs (see Fig. 4d). The distance and angle between furniture in those layout models need to meet the ergonomic [25]. Common furniture layout modes presented in indoor scenes are show in Table 1.

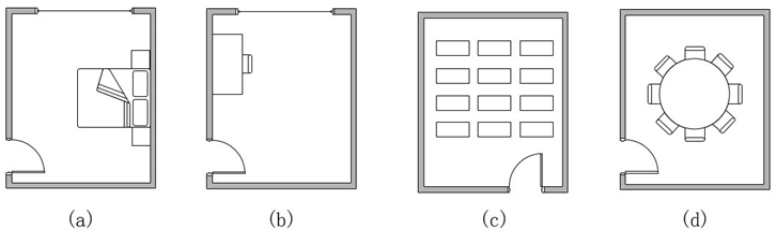


Fig. 4 (a) A example of the coupled mode (b) A example of the enclosed mode (c) A example of the matrix mode (d) A example of the circular mode

Table 1. Common layout modes presented in indoor scenes

Scene	Layout modes
Living room	Coupled mode and enclosed mode
Bedroom	Coupled mode and enclosed mode
Classroom	Coupled mode and matrix mode
Restaurant	Matrix mode, circular mode and coupled mode

6 Solution for Coupled mode

6.1 Mathematical model

We use a mathematical model to denote the layout relationship among the furniture in the coupled mode. A coupled mode consists of one piece of parent furniture and n pieces of child furniture. Each piece of child furniture is identical. As shown in Fig. 5, the parent furniture is the desk, and the child furniture is the stool. The parent furniture contains n layout points, and each piece of child furniture has one layout point. We use the axis-aligned bounding box to represent a piece of furniture, and use k to denote the child furniture number, $1 \leq k \leq n$. Let l, w and h denote the length, width and height of the furniture, respectively. Let $S = (l/2, h/2, w/2)^T$ be the size matrix, S_0 be the size matrix of the parent furniture, and S_k be the size matrix of the child furniture k. Let $P = (x, y, z)^T$ denote the furniture position, P_0 denote the parent furniture position, and P_k denote the position of the child furniture k. Let a be the furniture rotation angle around its Y axis with $a = \{0,90,180,270\}$, a_0 be the child furniture rotation angle, and a_k be the rotation angle of the child furniture k. The main idea of the coupled mode algorithm is as follows. We find the layout points of the parent furniture and that of the child furniture and then obtain the layout results through those layout points and matrix transformation. The position relationship between the child furniture k and the parent furniture can be modeled as follows:

$$\begin{aligned} P_k &= G_k + T_k + D_k \\ G_k &= M_k S_k \\ T_k &= P_0 + R_k S_0 \end{aligned} \tag{1}$$

where $G = (g_{ij})_{3\times1}$, and G_k denotes the layout point of the child furniture$k$;$T = (t_{ij})_{3\times1}$, and T_k denotes the parent furniture layout point relative to the child furniture k; $D = (d_{ij})_{3\times1}$, and D_k denotes the child furniture k distance matrix; $M = (m_{ij})_{3\times1}$, and M_k denotes the child furniture k transformation matrix; $R = (r_{ij})_{3\times1}$, and R_k denotes the parent furniture transformation matrix relative to the child furniture k; P_0, S_0 and S_k are known. a_k, M_k, R_k and D_k are set according to ergonomics and the real layout case.

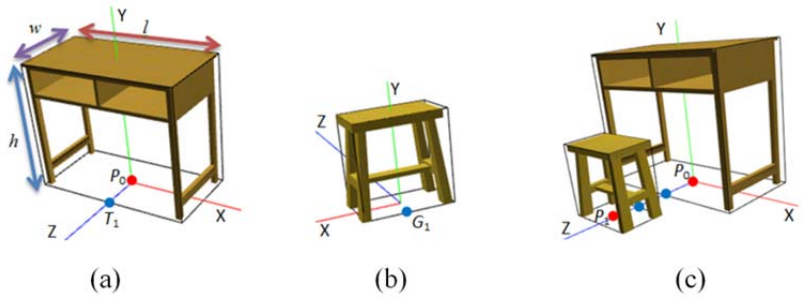


Fig. 5 (a) The information of the parent furniture table (b) The information of the child furniture stool (c) The information of a coupled mode

For example, the parameters of the mathematical model of the desk and stool in Fig.5 are as follow.

a_0=0, a_1=0

$$M_{01}=\begin{bmatrix}0&0&0\\0&0&0\\0&0&1\end{bmatrix}, M_1=\begin{bmatrix}0&0&0\\0&0&0\\0&0&1\end{bmatrix}, D_1=\begin{bmatrix}0\\0\\20\end{bmatrix}$$

6.2 Furniture-furniture coupled mode algorithm

Case-based reasoning has been widely used to solve various problems [26-28], and we use it to solve the coupled mode. The layout case consists of two parts: the layout problem description and the layout solution description. The layout problem is described by the vector $q=(l_0,w_0,h_0,l_1,w_1,h_1)^T$, where l_0,w_0,h_0,l_1,w_1, and h_1 denote the length, width, and height of the parent and child furniture, respectively. The layout solution is the parameters of mathematical model. The layout case name is composed of the parent furniture name, the child furniture name and the total amount of child furniture. Case similarity is defined as the Euclidean distance between the two layout cases. We use the nearest neighbor algorithm (K-Nearest Neighbor algorithm, K= 1) to retrieve cases.

$$\min_i \left\{ Dis(case_0, case_i) = \sqrt{\sum_{j=1}^{6} (q_{0,j} - q_{i,j})^2} \right\} \quad (2)$$

where $case_0$ denotes a await layout case, and $case_i$ denotes a layout case identical to the name of $case_0$ with $1 \leqslant i \leqslant m$. The problem description of $case_0$ and the problem description of $case_i$ are denoted by q_0 and q_i, respectively. The minimization returns the best case $case_{best}$ which is the most similar to $case_0$ in the case library. Substituting the solution of $case_{best}$ and known parameters P_0, S_0 and S_k of the $case_0$ into the mathematical model, we can obtain a layout result. The algorithm for one coupled mode is given below.

Algorithm 3 Furniture - furniture coupled mode
Input: An await layout $case_0$
Output: Layout results of the $case_0$

Step 1: Find a set of case whose name is identical to the $case_0$ from the case library.

Step 2: Find the $case_{best}$ using the nearest neighbor algorithm using formula (2).

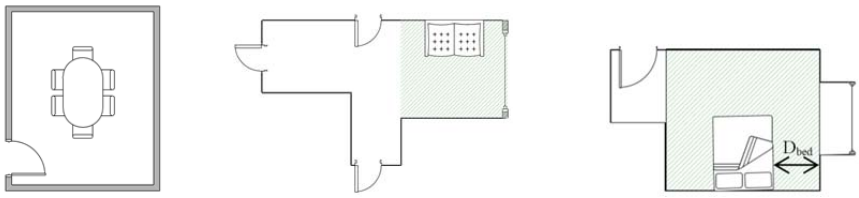
Step 3: Get the parameters form the mathematical model of the $case_{best}$.

Step 4: Assign the parameters to the mathematical model of $case_0$ and calculate using formula (1).

Step 5: Obtain and return the layout result of $case_0$.

6.3 Furniture-room coupled mode algorithm

When there is a coupled layout mode between furniture and a room, the room can be treated as furniture. Thus furniture-room coupled mode is converted into the furniture - furniture coupled mode. If the room shape is non-rectangular, and then it's rectangular region should be extracted with room shape standardization algorithm in Section 4. The common Furniture-room coupled modes are summarized as follows. (1) The tables are arranged in the middle of the conference room (see Fig. 6a). (2)The sofas arranged along the windows of the living room (see Fig. 6b). (3) The bed is arranged along the windows of the bedroom (see Fig. 6c), and the distance D_{bed} is determined by the room size.



(a) (b) (c)

Fig. 6 (a) Meeting room (b) Non-rectangular living room (3) Non-rectangular bedroom

6.4 Recursive case-based reasoning

Note that one coupled mode can be solved using Algorithm 5 in Section 6.1 and 6.2. However, there may have more than one coupled mode among the furniture in a room. As shown in Fig. 7, there exist four coupled modes, namely, the table and chair, the table and computers, the table and book, and the computer and mouse. For this case, we use both recursive techniques and case-based reasoning to solve. The algorithm is given below.

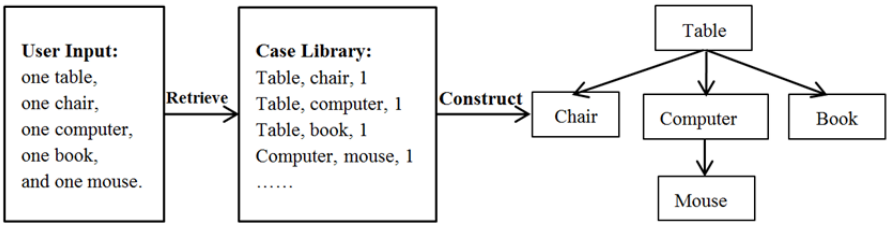


Fig. 7 Process of constructing a case tree

Algorithm 4 Multiple coupled modes
Input: A case tree
Output: Layout results of the furniture in the tree.

Step 1: Construct a case tree by retrieving the case library.
Step 2: Use an array A to save the furniture that has been arranged.
Step 3: Perform the following procedure starting from the root node of the tree.
Recursive Case-based Reasoning (root)

Procedure：Recursive Case-based Reasoning(node)

```
1   Let n denote the number of children of the node.
2   IF n>0 THEN
3       Sort children furniture of the node by their bounding box size
4       FOR i=1 TO n
5           child←children[i]
6           Try to arrange the child using formula (1)
7           IF no collision between child and the furniture in the array A
              THEN
8               Arrange the child using formula (1)
9               Put child into the array A
10              Recursive Case-based Reasoning (child )
11          ELSE
12              Arrange the child on the support surface. If arrange failed,
                and then the system advise the user to manual arrange the
                child furniture.
13          END IF
14      END FOR
15  END IF
```

In this procedure, in Step 7, we use the AABB bounding box of the furniture to perform collision detection. In Step 12, the support surface refers to the floor of the room or the surface of parent furniture. In Step 6, if we calculate the Y axis coordinate position of the child furniture equal to zero, and then the support surface of the child furniture is floor; otherwise, the support surface of the child furniture is the surface of parent furniture. For example, the layout tree is shown in Fig.7 and the

layout result is shown in Fig. 8b. The arranging furniture process using recursive case-based reasoning is as follows.

Step1: Arrange the table, and find the child nodes of the table; Sort those children by size, and then obtain the following sequence: chairs, computers and books. Put the table into the array A.

Step2: Arrange the chair and put it into the array A.

Stpe3: Arrange the computer. The computer has a child of mouse, and thus we should arrange the mouse. Similarly, put both the computer and mouse into the array A.

Step4: Arrange the book on the table. The book should be arranged in the same position of the computer according to the case library, which leads to the collision between the book and the computer (see Fig. 8a). Therefore we should find a position on the support surface of the table again to arrange the book, and finally the book is arranged in the upper left corner of the table (see Fig. 8b).



（a） (b)

Fig. 8 (a) A collision between the book and computer (b) The finally layout result

7 Solution for enclosed mode

7.1 Floor field

Many scholars used various floor fields to solve the problem of pedestrian evacuation [29-31]. This paper presents a floor field suitable for furniture layout to solve the enclosed mode. The floor field guides furniture arranged according to the energy distribution. Before constructing the floor field, the room**'s** floor should be discretized. We use 1 unit to represent 1cm in real world. For example, the length and width of a room are 4 meters and 3 meters, respectively, and then the size of the room's floor is expressed as 400*300. The room floor field consists of two parts: walls energy field and paths energy field. For example, a bedroom floor field is shown in Fig. 9a.

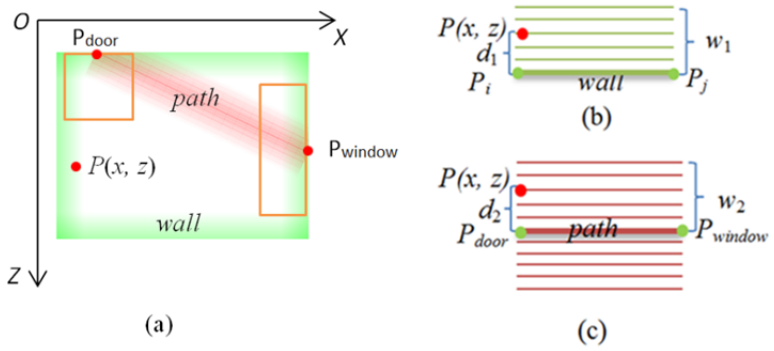


Fig. 9 (a) The floor field of a room (b) The energy field of a wall (c) The energy field of a path

A wall energy field is an attractive energy for the furniture. Let t denote the maximum energy intensity of a wall, t >0. Assume that the shortest distance from the point $P(x,z)$to the wall p_ip_i is d_1, and w_1 denotes the width of the wall energy field. The energy value of the point $P(x,z)$ inside the room generated by the wall $p_ip_i$is calculated as follows.

$$V(P(x,z),wall(p_ip_j))=\begin{cases}\dfrac{d_1}{w_1}\times t, d_1 \le w_1\\ 0, d_1 > w_1\end{cases} \tag{3}$$

A path energy field is a repulsive energy for the furniture. Let s denote the maximum energy intensity of a path, s<0. Assume that the shortest distance from the point $P(x,z)$ to the path $p_{door}p_{window}$ is d_2, and w_2 denotes the width of the path energy field. The energy value of the point $P(x,z)$ inside the room the path $p_{door}p_{window}$ is calculated as follows.

$$V(P(x,z),Path(i))=\begin{cases}\dfrac{d_2}{w_2}\times s, d_2 \le w_2\\ 0, d_2 > w_2\end{cases} \tag{4}$$

Given the positions of doors and that of window, any point's energy value in the room can be calculated as follows:

$$V(P(x,z))=\sum_{i=1}^{n}V(P(x,z),wall_i)+\sum_{j=1}^{m}V(P(x,z),path_j) \tag{5}$$

where n and m denote the number of walls and the number of paths, respectively. According to ergonomics guidelines, our parameter settings are as follows: t=50, s=−50, and w_1=w_2=50.

7.2 Furniture energy and furniture energy maximization

The furniture energy denotes the sum of the energy values of the points occupied by the furniture. As shown in Fig. 10, the rectangle is the bottom of the furniture bounding box, and the furniture energy is calculated as follows.

$$E(P(x,z),S(l,w)) = \sum_{i=x-l/2}^{x+l/2} \sum_{j=z-w/2}^{z+w/2} V(P(i,j)) \tag{6}$$

where point $P(x,z)$ denotes the furniture position, and l and w denote the length and width of the furniture, respectively.

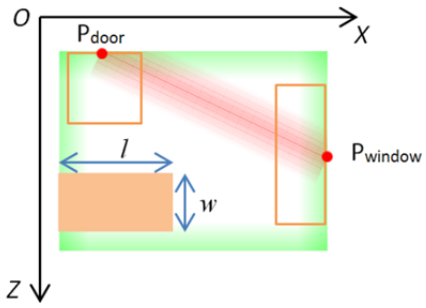


Fig. 10 Furniture energy

Furniture energy maximization refers to searching the position in the room where the furniture energy value is the largest, which is represented as follows.

$$\underset{x,z}{Max}\, E(P(x,z),S(l,w)) \tag{7}$$

where $E(P(x,z),S(l,w))$ denote the energy value of furniture in position $P(x,z)$. The furniture can be rotated with 0, 90,180 or 270 degrees in the search process of energy maximization. To improve the computational efficiency, the energy value of the furniture in current position can be calculated according to the energy value of the furniture in the previous position. Note that if the furniture collides with other furniture or the door's region (see Fig.10), and then the furniture cannot be arranged. In addition, in a bedroom, if the furniture collides with the window's region (see Fig.10) and the furniture's height is higher than the specified height, and then the furniture also cannot be arranged. A comparison is made between distance field in [24] and the floor field. Figure 11a shows the energy distribution using the distance field in [24], and Figure 11b shows the energy distribution using our floor field. Figure 11a and Figure 11b shows the layout results generated using distance field in [24] and our floor field. The result shows that our floor field is more suitable for furniture layout in a non- rectangular room.

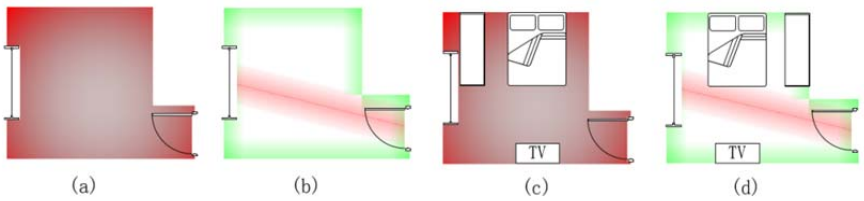


Fig.11 (a) The energy distribution using the distance field in [24] (b) The energy distribution using our floor field (c) The layout result using distance field in [24] (d) The layout result using our floor field

8 Solution for matrix mode and circular mode

8.1 Solution for matrix mode

The matrix mode is similar to the regular interval placement proposed in [23]. Furniture grouping is the emphasis in [23]. Here we focus on how to obtain the distance relationship among the furniture through the case library. Note that all the furniture is identical in the matrix mode. As shown in Fig. 12, let *l, w* and *h* denote the length, width and height of the furniture, respectively. Let d_v and d_h denote the vertical distance and horizontal distance among furniture, respectively. We store the parameters, namely, the real case, namely, the scene type, the furniture name, *l, w*, *h*, d_v and d_h, into the case library. The case name consists of the scene type and furniture name. The parameters *l, w*, *h*, d_v and d_h are used to describe case.

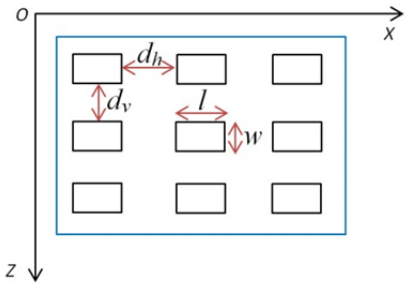


Fig.12 Distance parameters of the matrix mode

Consider that there exists furniture of matrix mode to be arranged, and find the furniture layout results. Assume that the furniture should be arranged in a rectangular region, and the location and size of the rectangular region have been determined. The layout process of the matrix mode is as follows.

Sept 1: Find the cases with identical name in case library.

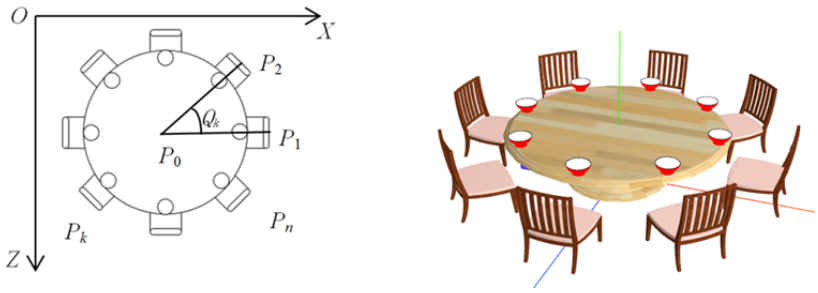
Sept 2: Similar to algorithm 3, retrieve the best case from those identical name cases using the nearest neighbor algorithm.

Sept 3: Find the parameters d_v and d_h from the best case.

Sept 4: According to the parameters d_v and d_h and the rectangular region, we can obtain the layout results using geometry calculation. Here we omit the process of the geometry calculation since they are simple.

8.2 Solution for circular mode

Here we use an example to illustrate the Circular mode algorithm. Given a round table and n chairs shown in Fig.13, find the layout results among the furniture. Let r denote the radius of the round table. The length and width of the chair are equal, denoted by l. Let $P_0(x,y,z)$ denote the position of the round table and $P_i(x,y,z)$ denote the position of the kth chair, $1 \leqslant k \leqslant n$. Assume that the n chairs can be arranged next to the table, and those chairs have no collision with each other.



(a) (b)

Fig.13 A example of the circular mode

The position of the kth chair can be obtained using the following formula.

$$\begin{aligned} P_k(x) &= P_0(x) + (r + l/2 + d) \times \cos(Q_k) \\ P_k(z) &= P_0(z) - (r + l/2 + d) \times \sin(Q_k) \\ P_k(y) &= P_0(y) \end{aligned} \tag{8}$$

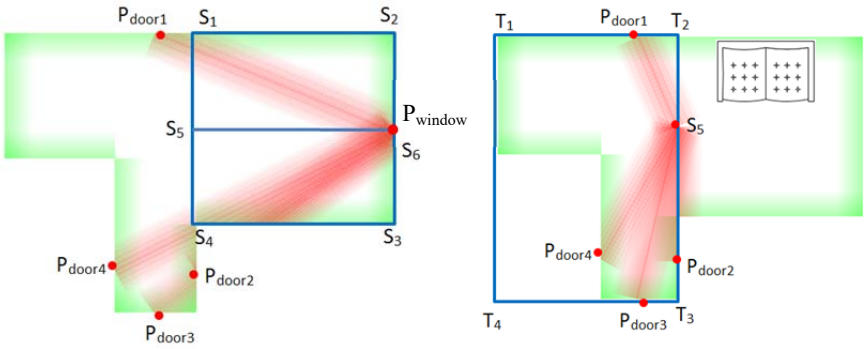
where Q_k denotes the angle of the kth chair rotating around Y axis of the round table, $Q_k = (k-1) \times 360/n$ (see Fig. 13a), d denotes the distance between the round table and the chair, and here d equals to zero. The distance d is required to meet the needs of ergonomics, and it also can be stored in the case library. The angle of the kth chair rotating around its Y axis equal to $\text{Angle}_k = Q_k - 90$. Similarly, we also can obtain the layout result between the round table and the bowls (see Fig. 13b).

9 Furniture layout algorithms for four scenes

In the previous sections, we proposed the general framework of furniture layout algorithms, and the solutions for the four layout modes. Based on the framework and those solutions, in this section we give the respective furniture layout algorithms for a living room, a bedroom, a classroom and a restaurant.

9.1 Living room furniture layout Algorithm

We consider the following problem. Given a living room which contains furniture such as sofas, televisions, dining chairs, dining tables and so on, generate a reasonable layout for the living room. The furniture layout algorithm for the living room is given as follows.



(a) (b)

Fig. 14 (a) Energy distribution of finding layout region for the sofa (b) Energy distribution of finding the position of the dining table

Algorithm 5 Living room furniture layout
Input: A living room and furniture
Output: Furniture layout results of the living room.

Step1: Arrange the furniture which belongs to the coupled mode using the coupled mode Algorithm in Section 6.

Step2: Extract the rectangular region $S_1S_2S_3S_4$ using the living room shape standardization algorithm in Section 4, and then divide the region $S_1S_2S_3S_4$ into two small regions: $S_1S_2S_6S_5$ and $S_5S_6S_3S_4$.

Step3: Generate floor field using formula (3), (4) and (5), which is shown in Fig. 14a. Calculate energy value for the two small regions using formula (6).

Step4: Arrange the television and sofa. The region with lower energy values is used to arrange the television, and the region with higher energy values is used to arrange the sofa. As is shown in Fig. 14a, the region $S_1S_2S_6S_5$ and $S_5S_6S_3S_4$ are used to arrange the sofa and television, respectively.

Step5: Regenerate floor field, which is shown in Fig. 14b.

Step6: Arrange the dining table and make its energy maximization using formula (7). The position of the dining table can be searched within the region $T_1T_2T_3T_4$.

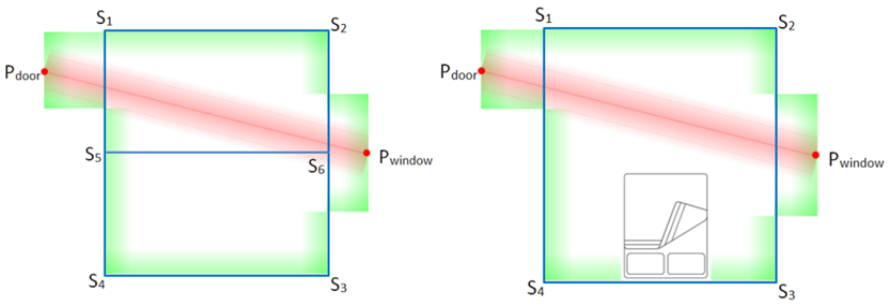
Step7: Sort the remaining furniture by size. Arrange the remainder furniture to make its energy maximization respectively. Return the layout results.

In this algorithm, in Step 2, if the length of the region $S_1S_2S_6S_5$ is smaller than the length of the sofa, and then the position of the sofa should be adjusted. In Step 5, as

the position of the sofa has been found, we change the current end point of paths, point P_{window}, to the point S_5. Thus, the floor field needs to regenerate.

9.2 Bedroom furniture layout Algorithm

We consider the following problem. Given a bedroom which contains furniture such as beds, televisions, cabinets and so on, generate a reasonable layout for the bedroom. The furniture layout algorithm for the bedroom is given as follows.



(a) (b)

Fig. 15 (a) Energy distribution of the bedroom (b) Finding the position of the bed

Algorithm 6 Bedroom furniture layout
Input: A bedroom and furniture
Output: Furniture layout results of the bed room.

Step1: Arrange the furniture which belongs to the coupled mode using the coupled mode Algorithm in Section 6.

Step2: Extract the rectangular region $S_1S_2S_3S_4$ using the bedroom room shape standardization algorithm in Section 4, and then divide the region $S_1S_2S_3S_4$ into two small regions: $S_1S_2S_6S_5$ and $S_5S_6S_3S_4$.

Step3: Generate floor field using formula (3), (4) and (5), which is shown in Fig. 15a. Calculate energy value for the two small regions using formula (6).

Step4: Arrange the television and bed. The region with lower energy values is used to arrange the television, and the region with higher energy values is used to arrange the bed. As is shown in Fig. 15a, the region $S_1S_2S_6S_5$ and $S_5S_6S_3S_4$ are used to arrange the television and bed, respectively.

Step5: Sort the remaining furniture by size. Respectively arrange the remainder furniture to make its energy maximization using formula (7). Return the layout results.

9.3 Classroom furniture layout Algorithm

We consider the following problem. Given a classroom which contains furniture such as a teacher's desk, students' desks, and students' chairs and so on, generate a reasonable layout for the classroom. Assume that the floor shape of the classroom is rectangular. The furniture layout algorithm for the bedroom is given as follows.

Algorithm 7 Classroom furniture layout
Input: A classroom and furniture
Output: Furniture layout results of the classroom

Step1: Arrange the students' desks and chairs using the coupled mode algorithm in Section 6.
Step 2. Arrange the students' desks using the matrix mode algorithm in Section 8.
Step 3. Arrange the teacher's desk using the coupled mode algorithm in Section 6.
Step 4: Return the layout results.

9.4 Restaurant furniture layout Algorithm

We consider the following problem. Given a restaurant which contains furniture such as a round table, chairs and so on, generate a reasonable layout for the restaurant. Assume that the floor shape of the restaurant is rectangular. The furniture layout algorithm for the restaurant is given as follows.

Algorithm 8 Restaurant furniture layout
Input: A restaurant and furniture
Output: Furniture layout results of the restaurant room.

Step1: Arrange the furniture which belongs to the coupled mode using the coupled mode Algorithm in Section 6.
Step2: Arrange the round tables and chairs using the circular mode algorithm in Section 8.
Step3: Arrange the round tables using the matrix mode algorithm in Section 8.
Step4: Return the layout results.

10 Experimental results and discussion

To validate efficiency of our algorithm, we conduct various experiments in our furniture layout system with the real interior design cases. We designed 16 common cases of the furniture layout to store in database of SQL Server 2012. The computation was performed on a computer with a processor of Dual-core 2.6GHz, main memory 4G, operation system Microsoft Windows 8 64bit and Chrome browser.

10.1 Comparisons with prior art

To evaluate our algorithms, we compared our algorithms with Liu's algorithm [5]. Since there is no size information for the two living rooms in [5], we manually extracted furniture and scene information from [5]. Then we conducted the two living room layout using our algorithms with the extracted furniture and scene information. Figure 16 and Table 2 shows the layout results and the detailed running time. The running time of our algorithms includes loading times of model files. The layout results of our algorithms are consistent with that of Liu's algorithms [5], but the computation efficiency of our algorithms is significantly higher than that of Liu's

algorithms [5]. The results show that the layouts of the two living rooms using our algorithms are reasonable and efficient.

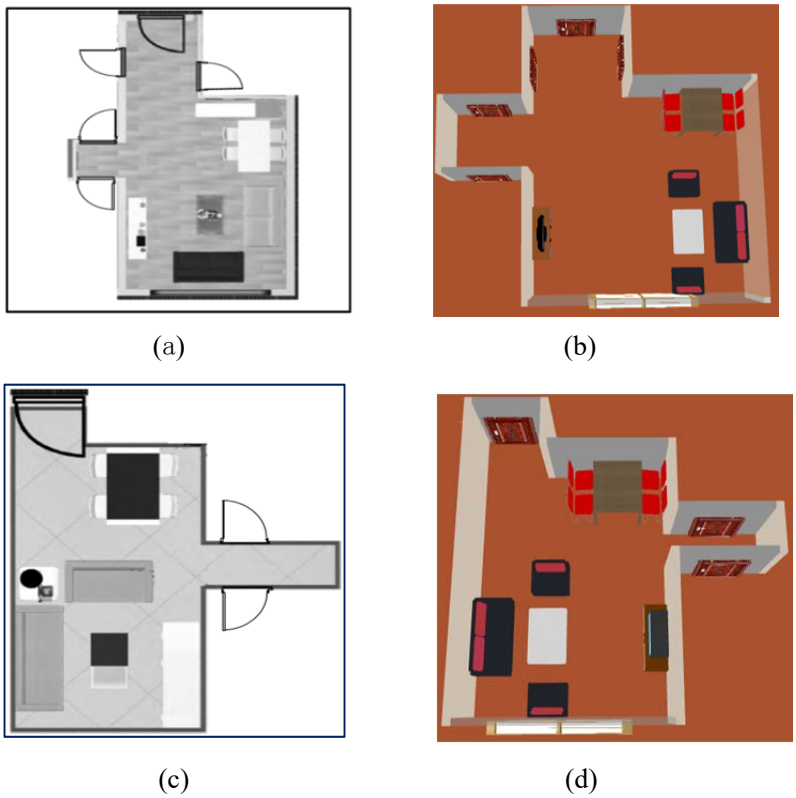


Fig. 16 (a) Living room 1 layout result using Liu's algorithms [5] (b) Living room 1 layout result using our algorithms (c) Living room 2 layout result using Liu's algorithms [5] (d) Living room 2 layout result using our algorithms

Table 2. Statistics of running time comparisons with [5]

Algorithms	Computer configuration	Programming tools	Average running time (s)
Our algorithms	RAM 4GB, and 2.6GHz CPU Intel Core i5-3230	WebGL and VS2015	1.5
Liu's algorithms [5]	RAM 4GB, and 3.4GHz CPU Intel Core i7-2600	Matlab and VS2008	50.3

10.2 Comparison with real designs

Given two real design cases shown in Fig. 17a and Fig. 17c, we also manually extracted furniture and scene information from the two cases. We invited an interior

designer to manually design two layout results according to the extracted information on our system. The designer was allowed to manually drag an object and move it along the floor plane, but the designer was not allowed to use any algorithmic inference. It took him almost an hour to finish the classroom and bedroom design. We conducted the bedroom and classroom layout using our layout algorithms on the system. The running time of the bedroom furniture layout is 1.7 seconds, and the running time of the classroom furniture layout is 0.6 seconds. The layout results of the bedroom and classroom using our algorithms are shown in Fig.17b and Fig.17d, respectively. The layout results of our algorithms are consistent with that of the two real design cases.

In addition, we conducted a restaurant layout experiment on our system. The restaurant contains 153 objects. The running time of the restaurant layout is 3.1 seconds. The running time also includes loading times of model files. The layout results of the restaurant using our algorithms are shown in Fig.18. The layout modes included in those experiments scene are given in Table 3.

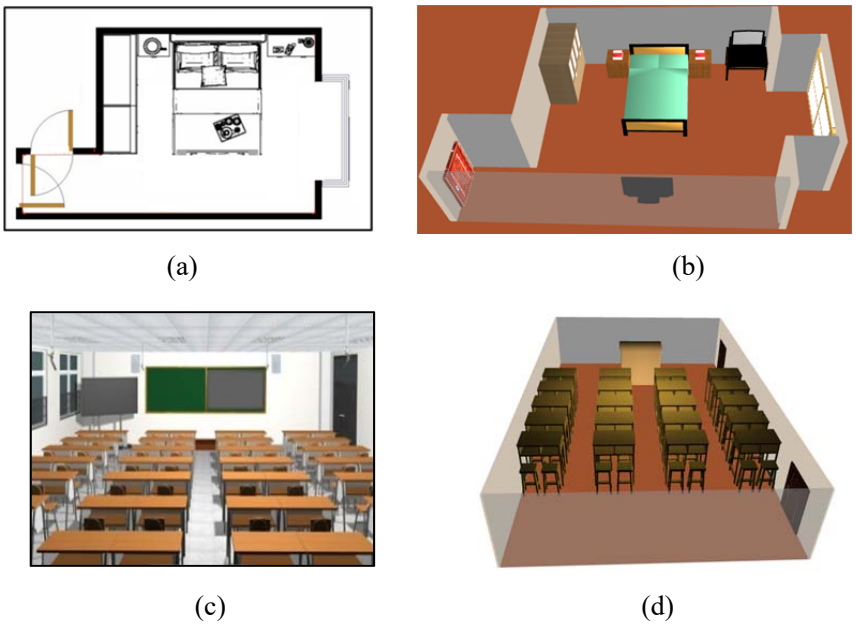


Fig.17 (a) A real design case of bedroom (b) The bedroom layout results using our algorithms (c) A real design case of bedroom (d) The classroom layouts result using our algorithms



Fig.18 The layout results of the restaurant using our algorithms

Table 3. The layout modes included in experiment scenes

Scene	Layout modes
Living room	Coupled mode: the sofas & coffee table, the dining table & chairs, the living room & sofas, the sofas & TV cabinet, the TV cabinet & television Enclosed mode: the sofas, the dining table
Bedroom	Coupled mode: the bed & bedside cabinets, the bedside cabinets & books, the bedroom & bed Enclosed mode: the bed, the cabinet, the dressing table
Classroom	Coupled mode: the students' desk & students' chair, the classroom & teacher's desk, the teacher's desk & students' desk Matrix mode: the students' desks
Restaurant	Matrix mode: the round table Circular mode: the round table & chairs, the round table & bowls

10.3 Discussion

In Section 10.2 and 10.3, our algorithms are performed in the living rooms, the bedroom, the classroom and the restaurant. The experimental results are analyzed as follow.

(1) Results show that our algorithms can meet the functional requirement of those scenes. On the one hand, the techniques of case-based reasoning make layout results learn from the real cases, thus the results are reasonable. On the other hand, the layout rule generated by floor field agrees with the real design rules. For non-rectangular living rooms and bedrooms, the paths in those rooms are found. The paths and walls in the room are given negative and positive energy, establishing a floor field, and then the furniture is arranged on the location with the largest energy value.

(2) The running time of our algorithms is suitable for real-time application. We divide the whole room layout into four furniture layout modes using the idea of divide and conquer, reducing complexity of the room layout, and thus our algorithms are fast. At the same time, we can see that the running time of Liu's algorithms [5] is 50.3 seconds, which is difficult to meet real-time requirements of the online layout. Liu et al [5] use the genetic algorithm to solve the furniture layout problem. The advantage of genetic algorithm is to be able to jump out from the local optimal. However, for a layout problem containing a lot of furniture, the layout relationship among the furniture becomes complicated, the search range of the algorithm becomes larger, and the number of iterations of the algorithm increases, and the running time being long. A similar problem is also found in [1]. Yu et al. [1] spend 219 seconds to deal with a restaurant layout problem using simulated annealing, and the restaurant contains 54 objects.

Based on the above analysis, we can conclude that our algorithms are effective, and can meet the needs of online real-time response. Our algorithms can be used in the online or off-line interior design platform to help users quickly create design drawing. In addition, another important application of our algorithm is modeling of 3D game scenes. In this scenario, our algorithm can be a function of the tool for automatically arranging 3D objects, reducing users' workload.

The limitations of this work should be acknowledged. For one thing, our algorithms are not suitable for the indoor scenes with multiple functions. For example, assume that a room is used as both living room and bedroom, containing furniture such as beds, televisions, cabinets, sofas, dining tables and so on. For this kind of rooms, our algorithm cannot obtain reasonable layout results. In practice, we found that this issue has little impact as most indoor room functions are independent for each other. For another, our algorithm requires that the walls of the room are parallel or vertical to each other. Except for in circular mode, our algorithms require that the furniture can only be rotated with 0, 90,180 or 270 degrees.

The future work will focus on the overall layout of the room which contains the light layout [32], color matching, and style matching [33]. Furthermore, this paper only considers a room furniture layout, and this work can be extended to a larger scene, such as the shopping malls, the whole floor, or even the whole building.

11 Conclusion

In this paper, we have proposed online automatic furniture layout algorithms and developed a Web 3D System based on those algorithms. A key component of our algorithms is that we solve the problem by dividing the whole scene layout into four layout modes, which reduce the layout complexity. To solve those layout modes and make the layout results reasonable, we utilized ergonomics guidelines, recursive techniques, case-based reasoning and floor field. We also presented the room shape standardization techniques to handle the layout problem of the non-rectangular room. Many experiments have been conducted in our layout system. The experimental results show that our algorithms are fast and effective, and the layout result of our algorithms can meet the functional requirement of the indoor scenes layout.

Acknowledgments

The authors appreciate the comments and suggestions of all anonymous reviewers, whose comments significantly improved this paper. This work is supported by The Key Research Projects of Central University of Basic Scientific Research Funds for Cross Cooperation (201510-02), Research Fund for the Doctoral Program of Higher Education of China (No.2013007211-0035), NSFC (No. 61502306), and Key project in scientific and technological of Jilin Province in China (No.20140204088GX).

References

1. Yu LF, Yeung SK, Tang CK, Terzopoulos D, Chan TF, Osher SJ (2011) Make it home: automatic optimization of furniture arrangement. ACM Transactions on Graphics 30(4): 86:1-86:11. doi:10.1145/2010324.1964981
2. Merrell P, Schkufza E, Li Z, Agrawala M, Koltun V (2011) Interactive furniture layout using interior design guidelines. ACM Transactions on Graphics 30(4):87:1-87:9. doi:10.1145/2010324.1964982
3. Chen G, Li G, Liu P, Ye T, Xian C (2014) Hierarchical constraints with particle swarm optimization for furniture arrangement. Journal of Computer-Aided Design and Computer Graphics 26(10):1603-1612
4. Akase R, Okada Y (2013) Automatic 3D Furniture Layout Based on Interactive Evolutionary Computation. In Proc. Conf. on 2013 Seventh International Conference on Complex, Intelligent, and Software Intensive Systems, pp. 726-731
5. LIU M, JIANG H, MAO TL, WANG Z (2016) Composite Model for Home Furnishing Generation. Chinese Journal of Computers 39(2):2:1-2:14
6. Sanchez S, Roux O L (2003) Gaildrat V, et al. Constraint-based 3d-object layout using a genetic algorithm. In Proc. Conf. on Computer Graphics and Artificial Intelligence
7. Roux M L O L, Gaildrat V (2004) Using Meta-Heuristics for Constraint-Based 3D Objects Layout. In Proc. Conf. on Computer Graphics & Artificial Intelligence
8. Guimarães MDP, Dias DRC, Mota JH et al (2016) Immersive and interactive virtual reality applications based on 3d web browsers. Multimedia Tools and Applications 75: 1-15. doi: 10.1007/s11042-016-4256-7
9. Zorrilla M , Martin A, Sanchez JR et al (2014) HTML5-based System for Interoperable 3D Digital Home Applications. Multimedia Tools and Applications 71(2):533–553. doi: 10.1007/s11042-013-1516-7
10. Quax P, Liesenborgs J, Barzan A et al (2016) Remote rendering solutions using web technologies. Multimedia Tools and Applications 75(8):4383-4410.doi: 10.1007/s11042-015-2481-0
11. Merrell P, Schkufza E, KoltunV (2010) Computer-generated residential building layouts. Acm Transactions on Graphics 29(6): 1-12.doi: 10.1145/1866158.1866203
12. Ulmer A, Müller P, Gool LV et al (2006) Procedural modeling of buildings. ACM Transactions on Graphics 25(3): 614-623. doi: 10.1145/1141911.1141931
13. Muller, P (2006) Procedural modeling of cities. In Proc. Conf. on SIGGRAPH 2006, ACM, pp.139-184.doi: 10.1145/1185657.1185716
14. Esch G, Wonka P, Zhang E (2007) Interactive procedural street modeling. ACM Transactions on Graphics 27(3): 35:1-35:8. doi: 10.1145/1278780.1278822

15. Bukowski RW (1995) Object associations: a simple and practical approach to virtual 3D manipulation. In proceedings of the Symposium on Interactive 3d Graphics, ACM, pp 131-138. doi: 10.1145/199404.199427
16. Smith G, Salzman T, Stuerzlinger W (2001) 3D scene manipulation with 2D devices and constraints. In Proc. Conf. on Graphics Interface ,pp.135-142
17. Kjølaas K.A.H (2000) Automatic Furniture Population of Large Architectural Models. Master's thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, Cambridge, MA
18. Xu K, Stewart J, Fiume E (2002) Constraint-based automatic placement for scene composition. In Proc. Conf. on Graphics Interface, pp 25-34
19. Germer T, Schwarz M (2009) Procedural Arrangement of Furniture for Real‐Time Walkthroughs. Computer Graphics Forum 28(8):2068-2078. doi: 10.1111/j.1467-8659.2009.01351.x
20. Fisher M, Ritchie D, Savva M et al (2012) Example-based synthesis of 3D object arrangements. ACM Transactions on Graphics 31(6):135:1-135:15.doi: 10.1145/2366145.2366154
21. Xu K, Chen K, Fu H, Sun, W L, Hu SM (2013) Sketch2Scene:sketch-based co-retrieval and co-placement of 3D models. ACM Transactions on Graphics 32(4):123: 1-123:15. doi:10.1145/2461912.2461968
22. Xu W, Wang B, Yan DM (2015) Wall grid structure for interior scene synthesis. Computers & Graphics 46(1): 231-243.doi: 10.1016/j.cag.2014.09.032
23. Akazawa Y, Okada Y, Niijima K (2002) Automatic 3d scene generation based on contact constraints. In Proc. Conf. on Eighth International Conference on Computer Graphics and Artificial Intelligence, pp 593-598
24. SONG PH, JIA JY(2016) Online Furniture Layout Method Based on Case-based Reasoning and Distance Fields. Journal of System Simulation 28(10):2438-2447
25. Liu Y, Ch Z (2013) Ergonomics and interior design. China Electric Power Press ,Beijing
26. Kolodner J L (1992) An introduction to case-based reasoning. Artificial Intelligence Review 6(1): 3-34.doi: 10.1007/BF00155578
27. Jin B, Xu S, Geng W (2016) Learning to Sketch human facial portraits using personal styles by case-based reasoning. Multimedia Tools and Applications 1-25.doi: 10.1007/s11042-017-4457-8
28. Besbes G, Baazaoui-Zghal H (2015) Modular ontologies and CRB-based hybrid system for web information retrieval. Multimedia Tools and Applications 74(18): 8053-8077. doi: 10.1007/s11042-014-2041-z
29. Fu Z, Xia L, Yang H, et al (2017) Simulation study of overtaking in pedestrian flow using floor field cellular automaton model. International Journal of Modern Physics C 28(5): 1750059:1- 1750059:17. doi: 10.1142/S0129183117500590
30. Nishinari K, Kirchner A, Namazi A, Schadschneider A (2004) Extended floor field CA model for evacuation dynamics. Ieice Transactions on Information and Systems 87(3): 726-732
31. Huang HJ, Guo RY (2008) Static floor field and exit choice for pedestrian evacuation in rooms with internal obstacles and multiple exits. Physical Review E Statistical Nonlinear & Soft Matter Physics 78(1):021131:1-021131:6. doi: 10.1103/PhysRevE.78.021131
32. Schwarz M, Wonka P (2014) Procedural Design of Exterior Lighting for Buildings with Complex Constraints. ACM Transactions on Graphics 33(5): 166:1-166:16. doi:10.1145/2629573
33. Li, W, Li, W, Li, W, Funkhouser T (2015) Style compatibility for 3d furniture models. ACM Transactions on Graphics 34(4) : 85:1-85:9. doi:10.1145/2766898