Automatic Furniture Arrangement Using Greedy Cost Minimization

Peter Kán* Hannes Kaufmann

Institute of Visual Computing and Human-Centered Technology, TU Wien, Vienna, Austria.

Figure 1: Interior scene generated by our method in 380 milliseconds. The selection and arrangement of furniture objects is optimized by greedy cost minimization and scene details are locally enhanced by procedural methods.

ABSTRACT

In this paper, we present a novel method for fast generation of furniture arrangements in interior scenes. Our method exploits the benefits of optimization-based approaches for global aesthetic rules and the advantages of procedural approaches for local arrangement of small objects. We generate the furniture arrangements for a given room in two steps: We first optimize the selection and arrangement of furniture objects in a room with respect to aesthetic and functional rules. The infinite trans-dimensional space of furniture layouts is rapidly explored by greedy cost minimization. In the second step, the procedural methods are locally applied in a stochastic fashion to generate important scene details. We demonstrate that our method achieves comparable results to a recent method for automatic interior design in terms of user preferences and that local procedural design enhances the result of optimization-based interior design. Additionally, our method is one order of magnitude faster than the compared method. Finally, the execution times of up to one second show that our method is suitable for generating large-scale indoor virtual environments during runtime.

Index Terms: Computing methodologies—Graphics systems and interfaces—Virtual reality

1 INTRODUCTION

Some virtual reality application areas would greatly benefit from methods for fast and automatized content generation. The 3D content generation is particularly important in the fields of architectural and interior design in which virtual reality plays an extremely important role. Both designers and clients can virtually visit, explore, and interact with the architectural scenes even before or during their construction. However, the process of manual placement and arrangement of 3D models in the scene makes the creation of such 3D interior scenes prohibitively expensive especially in case of large-scale indoor virtual environments.

Previous methods addressed this problem by semi-automatic [13, 22] or fully automatic [11, 21] stochastic methods. However, these methods are not capable of generating interior scenes fully automatically with the runtime below one second. High computational complexity of automatic interior design problem is caused by its infinite trans-dimensional search space of possible solutions. We address this problem by exploiting the property of interior design that it does not require to reach a global optimum and multiple diverse (locally optimal) solutions are preferable. Therefore, the interior design search space can be explored by much faster algorithm, in our case by greedy cost minimization. As a result, our algorithm can automatically arrange a room in a time below one second, it achieves diversity of multiple subsequent solutions and it can reach quality comparable to the recent offline methods.

While recent stochastic methods can arrange furniture objects in indoor scenes, they fail to generate important scene details (e.g. books on the shelf, doors, windows, table accessories, etc.). The main problem with these detail objects is that they often obey very strict rules which are difficult to fulfill with stochastic optimization-based approaches. We address this problem by utilizing procedural methods for scene detail enhancement. Procedural methods are fast and viable solution to this highly constrained problem because they directly generate the content according to the given constraints. We also enable stochastic behavior in these procedural methods if necessary (e.g. the position of laptop on the table can slightly vary).

Several procedural methods for automatic interior design were presented in previous research which achieve interactive speed [8, 19]. However, the drawback of these methods is that they do not take into account the optimization of aesthetic, ergonomic and functional rules. Therefore, they are prone to generation of unlivable scenes. Our method uses fast optimization of these rules in combination with procedural decoration, thus achieving high quality of generated scenes in an interactive time.

Manual furniture arrangement, on a professional level, requires the knowledge of common aesthetic, functional, and ergonomic rules (also called design guidelines) used in interior design practice. These rules were also employed in recent methods for automatic generation of indoor scenes [11, 13, 22]. Typically, the interior design rules are used as mathematical expressions which form a cost function to be optimized. We also utilize these rules in our greedy cost optimization. Additionally, we extended the set of rules with a composition rule inspired by golden section principle [15].

Our method for automatic interior design generates furniture arrangements in two main steps. In the first step, global furniture layout (and selection of objects) is optimized with respect to a cost function, formed by interior design rules. During this optimization the furniture arrangement is randomly altered by specific moves

*peterkan@peterkan.com

2018 IEEE Conference on Virtual Reality and 3D User Interfaces
18-22 March, Reutlingen, Germany
978-1-5386-3365-6/18/$31.00 ©2018 IEEE

(Section 3.1) and a new configuration is accepted only if its cost is lower than the cost of original interior design. In the second step, the resulting design is locally augmented by smaller objects to generate important scene details. For this purpose, we use the set of procedures, each targeting specific local arrangement (e.g. plates on table, windows in the openings, etc.).

We compare the results of our system to a recent method for automatic furniture arrangement [11] and the statistical analysis suggests that both methods achieve qualitatively comparable results. We also show that local procedural enhancements of interior increases the user preferences of design. Additionally, our method generates interior scene one order of magnitude faster than the compared method. Finally, we demonstrate that our method can successfully arrange objects in large-scale virtual environments. The main contributions of this paper can be summarized as follows:

- We propose a novel method for automatic furniture arrangement which combines optimization and procedural methods.
- We use greedy cost minimization to rapidly explore the space of possible furniture arrangements.
- Our system is 44 times faster than the compared method for automatic interior design.

2 Related Work

Interactive methods. Several methods were presented in past research which can populate virtual scenes with furniture in interactive time. These methods are typically based on fast procedural generation of furniture configurations using predefined rules. A real-time method for automatic furniture arrangement in large-scale virtual environments was proposed by Germer and Schwarz [8]. Their method uses agent-based solution to automatically generate object layouts. A fast method for automatic furniture layout was proposed by Tutenel et al. [19]. The authors developed a rule-based layout solver which can automatically arrange the selected objects given the rules for positioning of these objects. More recently an interactive furniture layout system was proposed [13] which assists users by suggesting new furniture arrangements. Given the initial layout and user constraints, this system generates new layout suggestions by Markov chain Monte Carlo sampler. Interior design rules are used to form the cost function which is explored by this sampler. This method requires a user assistance to generate the final interior design. Another method for user-assisted interior design was presented by Akase and Okada [1]. Their method uses evolutionary computation to optimize the furniture configuration. The user assistance is needed in this method to evaluate each layout candidate within the optimization process. The problem of objects layout in augmented reality was addressed by Gal et al. [7]. The authors proposed a rule-based framework for generating object layouts by solving a constraint-satisfaction problem with given rules.

Offline methods. Due to the high complexity and infinite search space in automatic interior design, numerous methods do not achieve interactive speed. Three main categories of offline methods can be recognized in previous research: Optimization-based methods, example-based methods, and procedural methods.

Optimization-based methods generate realistic furniture arrangements by optimizing furniture layout with respect to a cost function. Typically, this cost function includes ergonomic, aesthetic and functional terms. Several methods which optimize the cost function by evolutionary computing were proposed in past [11, 18]. While the former method uses the set of predefined constraints to form the cost function, the later method uses the cost function which mimics the principles used in professional interior design practice. Interior design principles were also utilized in the work of Yu et al. [22]. The authors optimize furniture arrangement by simulated annealing. Their method requires the set of furniture objects to be determined by the user. In contrast to that, our method is fully automatic, including the selection of furniture objects and optimization of their arrangement. The problem of automatic furniture selection and layout was addressed in past by Yeh et al. [21]. The authors proposed to use a stochastic Markov chain Monte Carlo sampling to explore the space of possible furniture configurations. In their system, the constraints of furniture objects are defined in an imperative programming language. In our system the furniture constraints are specified in a user-friendly interface as parameters.

Example-based methods address the interior design problem by utilizing information from existing layouts [5, 9, 23]. These approaches require a set of user-created layout examples to generate new plausible object arrangements. Recently, two methods were proposed which model 3D interior scenes based on the human activities performed in these scenes [6, 12]. The first method requires a 3D scan of the real environment to populate the virtual space with objects and the second one needs the initial scene layout to augment this layout by additional objects.

Several procedural methods for interior design were presented which require offline calculation due to the high complexity of placement rules or necessary user interaction. Some of the procedural methods were based on constraints [2, 20]. These constraints typically include placement constraints, proximity constraints, support constraints, contact constraints, etc. Another solution to automatic furniture arrangement is to address it from the perspective of recommender systems [4]. This solution firstly generates the set of layouts procedurally and then it uses a scoring function to rate the layouts and select the most profitable candidate.

Interior design rules. Aesthetic, ergonomic and functional rules are one of the key principles used by interior designers when creating a new design for a specific room. In order to form the cost function in our optimization, we used mathematical expressions representing the rules used in professional interior design [3, 14, 16] as well as the ones previously presented in the area of computer graphics [11, 13]. Additionally, we added the golden section principle into the cost function [15].

3 Automatic Interior Design

Our method for automatic interior design uses the combination of optimization and procedural methods to generate livable interior designs for a given virtual room. First, the optimal layout of big furniture objects is generated by greedy cost optimization. In the second step a procedural method is used for local arrangement of small objects (i.e. room decoration). The benefit of greedy cost optimization and procedural decoration is that our method achieves interactive speed of fully automatic interior design (Figure 1).

3.1 Furniture Arrangement Optimization

We use greedy cost minimization to populate a virtual room with furniture objects from the database in an optimization process. The main goal of this optimization is to find a furniture arrangement for a given room with minimal cost function. The minimization of the cost function works as follows: Initially, the furniture layout is randomly generated using the appropriate objects for a given room. Then, the cost function of this layout is calculated. In the next step the furniture layout is altered by the set of moves (mutations). Finally, the new configuration is accepted if its cost is lower than the previous one. The evaluation of the cost function and generation of new design by the set of moves is iteratively executed to reach the optimal furniture layout.

Despite the poor results of greedy cost minimization in search for global optimum, showed in previous research, our algorithm achieves the results comparable to other optimization methods. The reason for these surprisingly superior results of greedy cost minimization is that the problem of furniture arrangement optimization does not require to find a globally optimal solution. Rather multiple diverse solutions are preferable which are locally optimal and which fulfill specific expectations (i.e. different interior designers might like distinct furniture configurations for a room as soon as these configurations obey the expected aesthetic, ergonomic, and functional rules). Therefore, a simpler optimization method (e.g. greedy cost minimization) can be used to address this problem.

Objects categories. Our algorithm requires the information about properties and relationships of objects which are used for automatic layout. In order to simplify the addition of new objects into our system, we use the concept of object categories and store the required properties per category. Then, each object needs to only indicate to which category does it belong. These categories correspond to the types of objects, e.g. *"wardrobe"* or *"table"*. All objects from one category share the same properties which are used in the calculation of the cost function. These properties correspond to the measurements and relations used in the professional design practice [11, 13, 16, 17]. The following properties are used in our method for each category of objects:

- **Clearance values: front, back, left, right.** Clearance values specify the empty space around the object required for its comfortable use. In our experiments, we used values suggested in previous research [13].
- **Probability of standing against a wall.** This value specifies how important is it for an object to snap to a wall.
- **Possible parents.** This property contains the list of object categories which can be parents of a current object. Additionally, the minimum and maximum distance to a parent is used and the orientation towards the parent is set (front or side).
- **Probability of having a parent.** It represents importance for the object of being in a group relationship with the other objects.
- **Room importance.** This property is related to the functional needs in specific rooms. For example a table has to be in a dining room, thus having importance 1.0 for this room.
- **Desired count.** Each category contains a minimum required and maximum allowed number of objects of this category in a specific room.

Moves. The furniture layout optimization progresses in each iteration by altering current furniture layout. For this purpose we use the set of moves which can change the furniture layout [11]. The following moves are used in our method:

1. Randomly change the position of furniture object.
2. Randomly change the orientation of furniture object.
3. Align furniture object with the closest object.
4. Align furniture object with the closest wall.
5. Snap furniture object to the closest object.
6. Snap furniture object to the closest wall.
7. Connect furniture object with one of possible parents.
8. Add new children to the parent object.
9. Add new furniture object to the layout.
10. Remove random object from the furniture layout.

Each of these moves is executed in one iteration with a specific probability. In our experiments we empirically set these probabilities using following values. The moves 1 to 5 are applied for each furniture object with the probability 0.3. Move 6 is performed on each object with the probability given as the property of a specific object category. Move 7 is executed for each furniture object with the probability of having a parent, defined for the specific object category. Move 8 is performed on each object with the probability 0.6. Finally, the probabilities of adding and removing the objects to/from the design are set to 0.5 and 0.1 respectively. If the moves 8 and 9 are adding a new child object to a design, we use a special heuristic to achieve the alignment of children around parent object (e.g. the chairs around the table). This heuristic positions the child objects to the opposite sides of the parent object and aligns them based on their count. We accept only moves which do not cause intersections of objects.

3.2 Cost Function

Our algorithm optimizes furniture layout with respect to the cost function to reach the most desirable furniture arrangement for a given room. Therefore, our cost function reflects the rules which are used by professional designers in their practice. We use the mathematical representation of these rules presented in [11]. Additionally, we extended the set of rules by a term which expresses a golden section principle used in design [15]. The following rules and their mathematical expressions are used in our cost function:

Clearance. Furniture objects require an empty space from specific directions to be used by humans. We model the violation of this requirement as the amount of overlap between objects' bounding boxes extended by clearance constraints:

$$g_c = \frac{1}{|\mathscr{A}|\,(|\mathscr{A}|-1)} \sum_{b_1,b_2 \in \mathscr{A}: b_1 \neq b_2} \frac{V(b_1 \cap b_2)}{V(b_1)} \tag{1}$$

b_1 and b_2 are extended bounding boxes from the set $\mathscr{A}$ which contains extended bounding boxes of all furniture objects in a furniture layout in union with bounding boxes of walls, windows and doors. Function V returns the volume of the 3D geometric shape.

Circulation. Furniture objects have to be physically accessible by people. No part of the room should be blocked to be livable and usable. We express the violation of circulation by the number of objects which are not accessible from the entrance of the room. We implement the path finding by utilizing backtracking algorithm. The algorithm returns the number of accessible objects n_a. The total number of objects in the layout is denoted by n_t. The expression for the circulation can be then written as:

$$g_r = 1 - n_a/n_t \tag{2}$$

Group Relationships. We express the relationships between objects in terms of average distance of objects of the same type:

$$g_g = \frac{1}{|\mathscr{C}|\,(|\mathscr{C}|-1)d_r} \sum_{\vec{c}_1,\vec{c}_2 \in \mathscr{C}: \vec{c}_1 \neq \vec{c}_2} G(\vec{c}_1,\vec{c}_2)|\vec{c}_1 - \vec{c}_2| \tag{3}$$

where d_r is the diagonal size of the room in the ground plane. $\mathscr{C}$ is the set of all centers of furniture objects in the room. Function $G(\vec{c}_1,\vec{c}_2)$ returns 1 if the centers $\vec{c}_1,\vec{c}_2$ belong to the objects of the same category and 0 otherwise.

Alignment. In interior design, the objects should be properly oriented and aligned to their supporting surfaces and to other objects. We model the alignment by the variance of the angles between front vectors of objects in combination with a probabilistic distance measure between objects and their nearest wall:

$$g_a = \frac{1}{|\mathscr{V}|\,(|\mathscr{V}|-1)} \sum_{\vec{v}_1,\vec{v}_2 \in \mathscr{V}:\vec{v}_1 \neq \vec{v}_2} (\alpha - \overline{\alpha})^2 + g_w \tag{4}$$

$$\alpha = 1 - 0.5(\vec{v}_1 \cdot \vec{v}_2 + 1) \tag{5}$$

α is proportional to the angle between two front vectors $\vec{v}_1$ and $\vec{v}_2$. $\mathscr{V}$ is the set of front vectors of all furniture objects. We also include the front vectors of room walls into $\mathscr{V}$. $\overline{\alpha}$ is the mean value of α for all vectors in $\mathscr{V}$.

We add the wall distance term g_w to the alignment cost to support the requirement of some objects to stand near a wall. This term is using the probability of standing against a wall p_w defined for each object category.

$$g_w = \frac{1}{|\mathscr{P}|\, d_r} \sum_{\vec{p}_b \in \mathscr{P}} p_w |\vec{p}_b - \mathrm{proj}_w(\vec{p}_b)| \tag{6}$$

$\vec{p}_b$ is the point on the back side of the furniture object. The set $\mathscr{P}$ represents the set of these back points from all furniture objects present in current interior design. The function $\mathrm{proj}_w(\vec{p}_b)$ projects back point $\vec{p}_b$ to its closest wall and returns this projected point.

Distribution and Rhythm. The furniture objects should be properly distributed in space and the frequency of this distribution should follow a rhythm. We model this rule as a variance of the relative distances between the pairs of objects:

$$g_d = \frac{1}{|\mathscr{C}|\,(|\mathscr{C}|-1)} \sum_{\vec{c}_1,\vec{c}_2 \in \mathscr{C}:\vec{c}_1 \neq \vec{c}_2} (d - \overline{d})^2 \tag{7}$$

$$d = \frac{|\vec{c}_1 - \vec{c}_2|}{d_m} \tag{8}$$

d stands for relative distance between two points which is the Euclidean distance divided by the maximum distance d_m between two objects in the scene. $\overline{d}$ is the mean relative distance between all pairs of objects in the interior design.

Viewing Frustum. In an optimized layout, some objects should be visible from the others for their primary function. We express the violation of this rule using the number of occluded parent-child pairs n_i:

$$g_v = n_i / n_t \tag{9}$$

Functional Needs. Different rooms require specific objects for their primary function and for activities performed in these rooms. The expression for this requirement consists of two terms:

$$g_f = \frac{1}{2} \frac{\sum_{i_o \in \mathscr{I}} 1 - i_o}{|\mathscr{I}|} + \frac{1}{2} \frac{\sum_{o_c \in \mathscr{O}} \Delta(o_c)}{|\mathscr{O}|} \tag{10}$$

The first term is proportional to the importance of objects for a specific room and it sums up the importance values i_o of objects present in the furniture layout. $\mathscr{I}$ is the set of these importance values. The second term is related to the desired count of objects of a specific category in a room (see Objects categories above). Function $\Delta(o_c)$ calculates the difference between objects count of category o_c and the desired objects count of this category. The set $\mathscr{O}$ represents all categories present in the current interior design.

Proportion. We model the proportion of furniture size with respect to the room as the ratio of the volume covered by the objects to the room volume:

$$g_p = \frac{\max(r_v - V_o/V_r, 0)}{r_v} \tag{11}$$

V_o is the volume of all furniture objects together and V_r is the total volume of the room. r_v is the required ratio of the volume covered by furniture. We empirically set the value of r_v to 0.45 in our experiments.

Golden Section. We added a new term to the cost function which is based on a golden section principle, commonly used in art and design [15]. This principle suggests the eye-pleasing subdivision of the space. We subdivide the room rectangle by the lines which lie at the golden section ratio from each side. We approximate the golden section ratio by the value of 0.618. These four subdividing lines are used in design as locations of main objects. Therefore, we evaluate the golden section cost in terms of the distance of furniture objects to their closest golden section line:

$$g_s = \frac{1}{|\mathscr{C}|\, d_r} \sum_{\vec{c} \in \mathscr{C}} |\vec{c} - \mathrm{proj}_{gs}(\vec{c})| \tag{12}$$

$\vec{c}$ represents the center of furniture object and $\mathrm{proj}_{gs}(\vec{c})$ returns the projection of this center to its closest golden section line.

Cost Function. The cost function for furniture layout optimization is defined as a weighted sum of the above defined terms:

$$f_c = \sum_{i=1}^{n} w_i g_i \tag{13}$$

The weight for each guideline was set empirically in our experiments. We used value 1.0 for majority of weights. The weight for circulation was set to 1.1 and the proportion weight was 2.5. The weight for proportion was increased because this rule is essential for inserting objects to the scene. Additionally, we set the weight for functional needs to 3.0 because this rule is very important to add suitable furniture to a specific room. Finally, the weight for golden section was set to 0.5. We found these weights to work the best for our system and we used them across all our experiments.

3.3 Procedural Decoration

Decoration elements and small objects of everyday use make the interior space feel more cozy and inhabited. Therefore, we augment our method for automatic furniture layout by procedural decoration with small objects. The procedural decoration locally invokes procedures for arrangement of decorative objects into the room, furnished by our greedy cost minimization. These procedures typically position objects onto the supporting surfaces (e.g. laptop on the office table) based on given positioning rules. We also allow stochastic placement of objects within the given bounds in spatial and angular dimensions. Each decoration procedure has a set of objects assigned which can be positioned by the rules given for this procedure. Before placing the object into the scene, collisions with the existing objects are evaluated and if the collision occurs, the object is not inserted. We integrated the following procedures into our system:

Windows decoration. Typically, the reconstructed geometry of the room contains openings for the windows, but there are no window models inserted in these openings. Therefore, our procedure selects the window model from the database which fits the most appropriately the aspect ratio of the window opening to minimize the deformation of the window model. Finally, the window model is scaled to fit the opening and it is correctly positioned.

Curtains are often used as decorative elements in interior spaces. Therefore, our procedural decoration positions curtains in front of the windows if the surrounding walls and objects allow it. The curtains are scaled to fit the height of the room and to surround the window space. In our experiments we scaled the width of curtains to cover 140% of the window width.

Window blinds are useful for controlling the amount of light in indoor spaces and therefore are frequently part of the interior or exterior. Our procedure for positioning of window blinds first select the 3D model from the database which fits the window opening the most and then it scales it and positions in front of the window.

Doors decoration. This procedure is positioning the door objects into the door openings in the room geometry. The procedure for selection of appropriate door model and its placement is the same as the one for windows. The difference is that it uses different set of 3D models to be positioned.

Tables decoration. Tables in interior spaces serve for different purposes and they are always used as supporting surfaces for elements of everyday use, e.g. dishes, flowers, or office equipment. Due to the differences of table usage in various rooms, our procedure decorates tables depending on which room they are used in. We implemented the table decoration for kitchen, living room, and office. Each room has its specific set of objects to decorate the table with. The decoration of kitchen table uses the parent-child relationships of table and chairs to add plates and cutlery. These are positioned on the table in front of each chair. For the office and living room table, the objects are positioned stochastically with higher probability to be in the center of the table. The rotation of objects on the office table is calculated in a way to point the front vector of each object towards the chair. This orientation is stochastically perturbed with the allowed yaw angle variation of 15 degrees. Finally, the floor under the table is decorated by carpet in kitchen and living room if there is enough space around the table for inserting this carpet. The carpet model is also randomly selected from the database.

Corner decoration. In interior design the empty corners can be decorated by plants, lamps, or other objects. Therefore, we integrated procedure for corner decoration into our system. Our algorithm first finds the corners formed by walls and furniture. Then, random decoration objects from the specific set of 3D models are placed to these corners until the number of placed decoration objects reach the desired number. We used the empirically set number 3 as the desired number of corner decoration objects in our experiments.

Decoration of horizontal surfaces. The inhabited interior spaces tend to use horizontal surfaces (e.g. shelf or top desk of cabinet) for storing decorative objects or objects of everyday use (e.g. books, vases, candles). Thus, we integrated the procedure for positioning decorative objects onto the empty horizontal surfaces. In our implementation we used cabinets and shelfs as the base horizontal surfaces. The decoration procedure randomly selects a decorative object from the database and places it onto the random location on the base surface. This placement uses bounding boxes of objects to prevent object overlaying over the edge of the base surface.

Our procedural decoration covers the most common spaces for placement of small and decorative objects. Nevertheless, it is possible to extend the procedural decoration by adding new procedures into the system. Each new procedure should use its own set of decoration objects and its own rules for positioning of objects with respect to the existing furniture configuration. We implemented our algorithm in Unreal Engine 4 which is based on C++. Therefore, the decoration procedures are also implemented in C++ language. A comparison of the furnished room without and with the procedural decoration can be seen in Figure 4.

4 RESULTS

We evaluated our method for automatic furniture arrangement both quantitatively and qualitatively. Four different scenes were used in our evaluation to judge the speed and quality of the generated interior designs: Bedroom, kitchen, living room and office. We first measured the speed of our method and then we evaluated the quality of the generated results in a user study. The results of our method were also compared to the recent method for automatic interior design [11]. Finally, we evaluated the capability of our method to generate livable interior design in an expert study. The furniture arrangements, generated by our method, can be seen in Figure 1, Figure 4 and in supplementary materials.

4.1 Time Measurements

In order to evaluate the execution time of our method we generated an interior design for each test scene five times in a row and we calculated the mean. We used 500 iterations in greedy cost minimization in all our experiments. After 500 iterations the reduction of cost and the improvement of interior design were not significant. Also, the fixed number of iterations was employed to preserve certain level of diversity in subsequent results (i.e. to not always find the same global optimum). We used 48 3D models of furniture in our layout optimization experiments. Moreover, 30 3D models subdivided into 10 different sets were used for procedural decoration. All performance measurements were done on a laptop PC with a quad-core CPU with 2.4 GHz, 8GB of RAM and two GPUs NVIDIA GeForce GT 750M. The resulting execution times of our algorithms can be seen in Table 1. The results indicate that our method for automatic furniture arrangement runs in interactive frame rates for all tested scenes.

Additionally, we compared the speed of our method to the recent method for automatic interior design based on a genetic algorithm [11] using the same hardware. The results of this comparison are shown in Table 2. Our method achieves the execution times 44 times faster in average than genetic algorithm.

Table 1: Execution times of our method for each tested scene. Layout time expresses the time of furniture arrangement by greedy cost minimization and decoration time denotes the time of procedural decoration. The time is shown in milliseconds.

Scene	Layout time (ms)	Decoration time (ms)
Bedroom	265	18
Kitchen	509	15
Living room	751	15
Office	378	17

Table 2: Comparison of our method to the automatic interior design based on genetic algorithm [11]. The time measurements are in seconds. The speedup indicates how many times faster our method runs in comparison to genetic algorithm. Our method used 500 iterations and we set genetic algorithm to use 50 iterations in all scenes.

Scene	Genetic algorithm (s)	Ours (s)	Speedup
Bedroom	18.17	0.28	65x
Kitchen	20.49	0.52	39x
Living room	21.62	0.77	28x
Office	17.31	0.40	43x

4.2 User Study

We evaluated the quality of the furniture arrangements generated by our method by conducting a user study. The main goal of this study was to investigate if our method can achieve similar or better quality of the generated interior design than a recent method based on a genetic algorithm [11]. We used four scenes for this comparison: Bedroom, kitchen, living room and office (Figure 4). Additionally, two conditions, without decoration and with decoration, were evaluated for each test scene. The procedural decoration was only enabled for our method. In total 8 scenes had to be evaluated. 44 users participated in the user study, including 32 males and 12 females, in age from 17 to 44.

Study Design. Our null hypothesis H_0 was that there are no significant differences of user preference between interior design generated by genetic algorithm [11] and the one generated by our method. The alternative hypothesis H_1 was that there are significant differences of preference between these two conditions. If we find the scenes with no statistically significant differences between genetic algorithm and our method (H_0) or if our results are preferred by the users (H_1 in favor of our method), our method may be considered successful. In order to validate our hypotheses, we conducted an experiment using a subjective, two-alternative, forced-choice preference approach similar to [10, 11].

Procedure. Our user study was conducted in the form of an online questionnaire. Each question of the questionnaire showed two interior designs for side-by-side comparison by the user, one created by our system and one created by genetic algorithm [11]. Each interior design was represented by three images: Two images from the first-person perspective and one image from the top. For each question the user had to select which interior design, from two alternatives, would he/she prefer to live in. The questionnaire consisted of 8 questions, four for furniture arrangements without procedural decoration and four for furniture arrangements with procedural decoration. The procedural decoration was only enabled for greedy cost minimization. The order of the questions was randomized. The sides (left, right) of two compared interior designs were also randomized.

Outcome and Analysis. We used a Chi-square nonparametric analysis to determine any statistical significance in each of our 8 conditions (4 rooms, without and with decoration). The Chi-square analysis in one-dimension was used for each of these conditions separately. Each condition contained 44 preference answers, thus the frequencies of preferences were compared to an expected 22/22 result. The Chi-square values were computed and tested for significance. The results of this analysis are shown in Table 3. Moreover, the measured frequencies of user preferences are shown in Figure 2.

The results of the Chi-square analysis suggest that there is no clear winner, amongst our method and genetic algorithm, which would significantly outperform the other in all tested scenarios. In a layout task without the decoration, our method outperformed the genetic algorithm in bedroom scene and office scene while in the latter, the difference was statistically significant ($p = 0.003$). In the kitchen scene and living room scene, the genetic algorithm achieved higher preferences than our method. In all tested scenes, the procedural decoration was able to improve the preference of our method (the procedural decoration was only applied on the result of our method). Additionally, in all tested scenes our algorithm with procedural decoration achieved higher preference than genetic algorithm without the procedural decoration. This trend was significant for bedroom, living room, and office scenarios.

In summary, our method achieved higher frequency of user preference than genetic algorithm in 6 out of 8 conditions while 4 of them were statistically significant. On the other hand genetic algorithm was preferred by the users in 2 out of 8 conditions while 1 of them was significant. Based on these results we conclude that both evaluated methods achieve comparable quality of automatic furniture arrangement in terms of user preference. Finally, the results of the user study suggest that procedural decoration by small objects is beneficial for improving the user preference of interior design.

Table 3: The results of the Chi-square analysis. Values in boldface indicate significant difference (level of significance = 0.05). The left column shows the results of furniture layouts without procedural decoration and the right column shows the results for layouts which used procedural decoration for enhancement of greedy cost minimization.

	Layout only		**Layout + decoration**	
Scene	χ^2-value	p-value	χ^2-value	p-value
Bedroom	0.818	0.366	**7.364**	**0.007**
Kitchen	**7.364**	**0.007**	0.364	0.546
Living room	0.818	0.366	**13.091**	$<$ **0.001**
Office	**9.091**	**0.003**	**23.273**	$<$ **0.001**

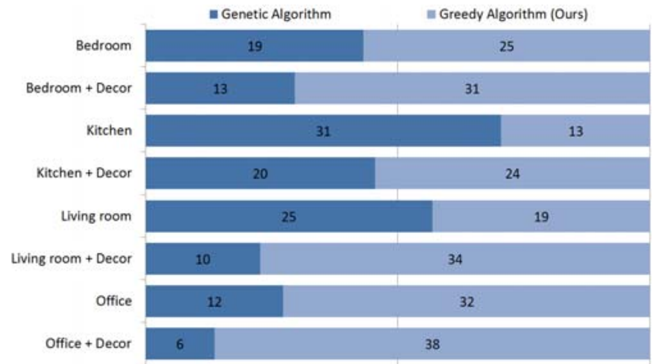

Figure 2: Frequencies of user preferences in our study. Each of the 8 displayed conditions represents the preferences amongst answers of 44 participants. Decor indicates that procedural decoration was enabled for greedy cost optimization.

4.3 Expert Study

In addition to the quality of generated interior design we evaluated the probability that the algorithm generates livable interior design at the first chance. Moreover, we used this probability for comparison of our method with genetic algorithm [11]. In order to calculate the probability of generating livable design, we generated 50 furniture arrangements in a row by each algorithm for each of two used scenes: Bedroom and office. Then, we asked two professional interior design artists to judge all generated scenes if they are livable or not (i.e. if they satisfy the aesthetic, ergonomic, and functional requirements which are necessary for living in these scenes). The count of interior designs marked as livable for each scene and each algorithm can be seen in Figure 3. From these numbers we can calculate the probabilities of generation of interior design: Genetic algorithm generated 40% livable designs for bedroom scene and 52% livable designs for office scene. Our method generated 50% of livable designs for bedroom scene and 40% of livable designs for office scenes. In summary both methods achieve comparable rates for generation of livable furniture arrangements.

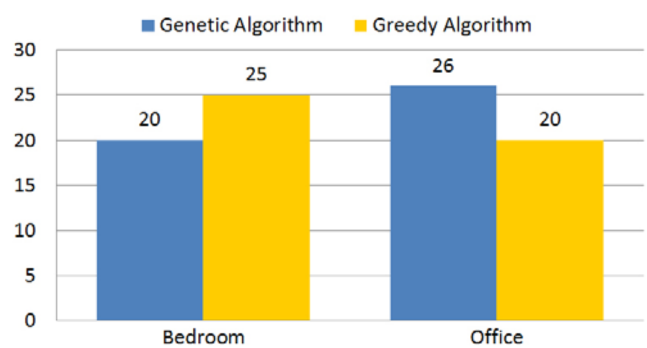

Figure 3: The numbers of interior designs which were judged by artists as livable. In total 50 designs were judged for each column.

4.4 Complex Virtual Environments

We integrated the presented methods into a complex system for content generation in virtual environments. This system enables fast generation of large-scale virtual environments which can be utilized in many application areas including architecture, real estate, training, and others. We tested our system in a scenario where automatic content generation in complex virtual environment is required: Generation of interior design in a large flat for VR walkthrough.

Our system was able to successfully generate the target scene. The resulting interior design can be seen in supplementary materials. We implemented our system in Unreal Engine 4 which enables realistic rendering, VR integration and content editing in Unreal Editor.

5 Discussion

The results of user study and expert study indicate that our method is capable of producing sensible and livable interior designs which are comparable with the recent method for automatic interior design [11] in terms of user preference. Moreover, the performance measurements show that our method is one order of magnitude faster than the compared method. The fast speed of our algorithm enables interactive generation of VR environments also during runtime.

Based on the varying significance values and trends in user preferences, we can hypothesize that the quality of the generated interior design depends on the type of the room. Our algorithm performed very well in the bedroom and office scenes while it was less preferred by participants in the kitchen and living room scenes. Therefore, future investigations can help to identify the source of this variability and to consequently extend the cost function. Finally, the results of the user study indicate that procedural decoration is vital for increasing the user preference of virtual scenes.

5.1 Limitations and Future Work

Currently, our system uses predefined 3D models of furniture from model database to generate the furniture arrangement. However, in interior design practice, custom furniture is often built to exactly fit the dimensions of interior space and to optimize ergonomics and aesthetics. Therefore, it will be beneficial to focus on the custom procedural furniture construction in future work.

The limitation of greedy cost minimization is that it can be trapped in a local minimum. The result of local minimum is acceptable for interior design if the cost function reaches certain threshold and if the aesthetic, ergonomic, and functional requirements are satisfied. However, our expert study indicates that in some cases these requirements are not satisfied. Therefore, future investigation of automatic restart strategies and more tightly coupled procedural methods with optimization would be helpful for further improvement of our method.

The cost function used by our algorithm takes into account many interior design rules used in practice. However, the judgment of furniture arrangement by interior designers is much more complex function which includes not just the design rules but also artistic feeling, natural sense of livability and imagination. Such a function is very far in terms of complexity from the defined cost function and it is impractical to model it analytically. Therefore, the training of deep neural network for judgment of interior designs based on artist-provided data will be a vital task for future work.

6 Conclusion

This paper presents a novel method for automatic furniture arrangement in interior scenes. The presented method utilizes the combination of optimization-based and procedural generation of interior in virtual environments. The benefit of this combination is that the resulting furniture arrangement obeys the aesthetic, ergonomic, and functional rules used in interior design practice and in the same time it creates the feeling of inhabited space thanks to the procedural decoration. We evaluated the presented methods qualitatively and quantitatively. The results of our studies suggest that our method can generate sensible and livable interior designs which are comparable to the recent research [11] while our method achieves the runtimes one order of magnitude faster than the compared method. Finally, we demonstrated the suitability of the presented method for the content generation in complex virtual environments.

Acknowledgments

We thank anonymous reviewers for their valuable suggestions and comments. Our thanks also goes to professional artists Anna Cséfalvay and Matej Hoppan for their help with the livability evaluation in our expert study. This research was funded by the Austrian research project FFG-Bridge under grant no. 850706 and by the research project WWTF ICT15-015.

References

[1] R. Akase and Y. Okada. Automatic 3D furniture layout based on interactive evolutionary computation. In *CISIS '13*, pp. 726–731. IEEE Computer Society, 2013.

[2] Y. Akazawa, Y. Okada, and K. Niijima. Automatic 3D scene generation based on contact constraints. In *Proc. Conf. on Computer Graphics and Artificial Intelligence*, pp. 593–598, 2005.

[3] D. K. Ballast. *Interior Design Reference Manual: Everything You Need to Know to Pass the NCIDQ Exam*. Professional Publications Incorporated, 2013.

[4] S. Chojnacki. Scoring functions for automatic arrangement of business interiors. In *SIGGRAPH Asia 2012 Technical Briefs*, SA '12, pp. 27:1–27:4. ACM, New York, NY, USA, 2012.

[5] M. Fisher, D. Ritchie, M. Savva, T. Funkhouser, and P. Hanrahan. Example-based synthesis of 3D object arrangements. *ACM Trans. Graph.*, 31(6):135:1–135:11, Nov. 2012.

[6] M. Fisher, M. Savva, Y. Li, P. Hanrahan, and M. Nießner. Activity-centric scene synthesis for functional 3D scene modeling. *ACM Trans. Graph.*, 34(6):179:1–179:13, Oct. 2015.

[7] R. Gal, L. Shapira, E. Ofek, and P. Kohli. Flare: Fast layout for augmented reality applications. In *IEEE ISMAR*, pp. 207–212, 2014.

[8] T. Germer and M. Schwarz. Procedural arrangement of furniture for real-time walkthroughs. *Comp. Graph. Forum*, 28(8):2068–2078, 2009.

[9] P. Guerrero, S. Jeschke, M. Wimmer, and P. Wonka. Learning shape placements by example. *ACM Trans. Graph.*, 34(4):1–13, 2015.

[10] J. Jimenez, V. Sundstedt, and D. Gutierrez. Screen-space perceptual rendering of human skin. *ACM Trans. Appl. Percept.*, 6(4):1–15, 2009.

[11] P. Kán and H. Kaufmann. Automated interior design using a genetic algorithm. In *Proceedings of VRST 2017*. ACM, 2017.

[12] R. Ma, H. Li, C. Zou, Z. Liao, X. Tong, and H. Zhang. Action-driven 3D indoor scene evolution. *ACM Trans. Graph.*, 35(6):1–13, 2016.

[13] P. Merrell, E. Schkufza, Z. Li, M. Agrawala, and V. Koltun. Interactive furniture layout using interior design guidelines. *ACM Trans. Graph.*, 30(4):87:1–87:10, July 2011.

[14] M. Mitton and C. Nystuen. *Residential Interior Design: A Guide To Planning Spaces*. Wiley, 2011.

[15] E. Neufert, P. Neufert, and J. Kister. *Architects' Data*. Wiley, 2012.

[16] L. O'Shea, C. Grimley, and M. Love. *The Interior Design Reference & Specification Book: Everything Interior Designers Need to Know Every Day*. Rockport Publishers, 2013.

[17] J. Panero and N. Repetto. *Anatomy for Interior Designers. Third Edition*. Whitney Library of Design, 1975.

[18] S. Sanchez, O. Le Roux, H. Luga, and V. Gaildrat. Constraint-based 3D-object layout using a genetic algorithm. In *International Conference on Computer Graphics and Artificial Intelligence*, 2003.

[19] T. Tutenel, R. Bidarra, R. M. Smelik, and K. J. d. Kraker. Rule-based layout solving and its application to procedural interior generation. In *CASA'09 Workshop on 3D Advanced Media in Gaming and Simulation*, pp. 15–24, 2009.

[20] K. Xu, A. J. Stewart, and E. Fiume. Constraint-based automatic placement for scene composition. In *Graphics Interface*, pp. 25–34, 2002.

[21] Y.-T. Yeh, L. Yang, M. Watson, N. D. Goodman, and P. Hanrahan. Synthesizing open worlds with constraints using locally annealed reversible jump MCMC. *ACM Trans. Graph.*, 31(4):56:1–56:11, 2012.

[22] L.-F. Yu, S.-K. Yeung, C.-K. Tang, D. Terzopoulos, T. F. Chan, and S. J. Osher. Make it home: Automatic optimization of furniture arrangement. *ACM Trans. Graph.*, 30(4):86:1–86:12, July 2011.

[23] X. Zhao, R. Hu, P. Guerrero, N. Mitra, and T. Komura. Relationship templates for creating scene variations. *ACM Trans. Graph.*, 35(6):207:1–207:13, Nov. 2016.

Figure 4: The resulting furniture arrangements generated by our method without (left) and with (right) procedural decoration. The following scenes are shown from top to bottom: Bedroom, kitchen, living room and office. These scenes were also used in our user study.