SceneDirector: Interactive Scene Synthesis by Simultaneously Editing Object Groups in Real-Time

Journal:	*Transactions on Visualization and Computer Graphics*
Manuscript ID	Draft
Manuscript Type:	Regular
Keywords:	3D Scene Editing, Interactive 3D Modeling, 3D Scene Synthesis
Note: The following files were submitted by the author for peer review, but cannot be converted to PDF. You must view these files (e.g. movies) online.	
videoTVCG.mp4 supp.zip	

SceneDirector: Interactive Scene Synthesis by Simultaneously Editing Object Groups in Real-Time

Shao-Kui Zhang, Hou Tam, Yi-Ke Li, Ke-Xin Ren, Hongbo Fu, Song-Hai Zhang, *Member, IEEE*

Abstract—Intelligent tools for creating synthetic scenes have been developed in great progress in recent years. Existing techniques on interactive scene synthesis only incorporate a single object at every interaction, i.e., crafting a scene through a sequence of single-object insertions with user preferences. These techniques suggest objects by considering existent objects in the scene instead of fully picturing the eventual result, which is inherently difficult since the sets of objects to be inserted are seldom fixed during interactive processes. In this paper, we introduce SceneDirector, a novel interactive scene synthesis tool to help users quickly picture various potential synthesis results by simultaneously editing groups of objects. Specifically, groups of objects are rearranged in real-time with respect to a position of an object specified by a mouse cursor or gesture, i.e., a movement of a single object would trigger the rearrangement of the existing object group, the insertions of potential appropriate objects, and the removal of redundant objects. To achieve this, we first propose an idea of coherent group set which expresses various concepts of layout strategies. Subsequently, we present layout attributes, where users can simply adjust how objects are arranged by tuning the weights of the attributes. Thus, our method gives users intuitive control of both how to arrange groups of objects and where to place them. Through extensive experiments and two applications, we demonstrate the potentiality of our framework and how it enables effective and efficient interactions of editing groups of objects concurrently.

Index Terms—3D Scene Synthesis, 3D Scene Editing, Interactive 3D Modeling.

1 Introduction

3D scene synthesis benefits various applications, including metaverse [1], virtual reality [2], [3], computer vision [4], interior designs [5], etc. Many attempts (e.g., [6], [7], [8], [9]) have been made to automatically synthesize 3D scenes. However, as verified in [10], [11], automatic layout generations often do not guarantee preferences of users. Typical interior designers usually have to listen to their customers and manually craft scenes according to the needs of customers and interior rules [12], [13], [14]. Thus, an intelligent interactive tool is more practical, and has also been investigated in recent years.

To allow interactive control of scene synthesis, existing literature considers "objects" as the targets of manipulation [10], [11], [15], [16], [17], i.e., as the control units, objects are successively inserted into scenes. While users have a full control of the synthesis process, these techniques do not offer quick overviews of synthesized results during the synthetic process. For example, a user might add several objects in the beginning but halfway through the synthesis she/he finds that some of objects are functionally/aesthetically/stylistically unsuitable, thus consuming more time due to a longer interactive session. Thus, in practice, existing interactive solutions still leave blanks in foreseeing variations on desired scenes and have gaps in simultaneously manipulating groups of objects.

To address these issues, we present SceneDirector, which provides a novel tool for interactive scene synthesis by editing object groups. With our tool, when a user changes a position of a single object, our method automatically rearranges the rest of related objects as shown in Figure 1. This is achieved by considering the concepts of layouts, which consist of style-compatible sets of objects, express strategies of arranging objects, and respond for human affordance [12], [13], [14], [18]. However, layout concepts are sophisticated [12], [13], [14], [18], especially when we want to acquire a specific one and explicitly apply it under various contexts. Thus, we try learning layout concepts through a data-driven process, but instead of preparing a large amount of data and training unified models such as [6], [9], [19] we treat groups of objects with a compatible style and function as a unique set, which we name *Coherent Groups Set* (CGS). Each CGS stands for a particular layout concept, which is learned using only its corresponding CGS. The more groups in a CGS, the clearer the description of the corresponding concept, where one of our contributions is to formulate a layout concept given a CGS. To support the CGS, existing datasets are insufficient due to their low densities of objects

- *Shao-Kui Zhang and Hou Tam are with the Department of Computer Science and Technology, Tsinghua University, Beijing, China. E-mail: zhangsk18@mails.tsinghua.edu.cn, th21@mails.tsinghua.edu.cn*
- *Yi-Ke Li and Ke-Xin Ren is with the Academy of Arts & Design, Tsinghua University, Beijing, China. E-mail: lyk20@mails.tsinghua.edu.cn, rkx20@mails.tsinghua.edu.cn*
- *Prof.Hongbo Fu is with the School of Creative Media, City University of Hong Kong, Hong Kong. E-mail: hongbofu@cityu.edu.hk*
- *Song-Hai Zhang is with the Department of Computer Science and Technology, Tsinghua University, Beijing, China and Beijing National Research Center for Information Science and Technology (BNRist), Tsinghua University, Beijing, China. E-mail: shz@tsinghua.edu.cn*

Manuscript received - -, 2022; revised - -, -.

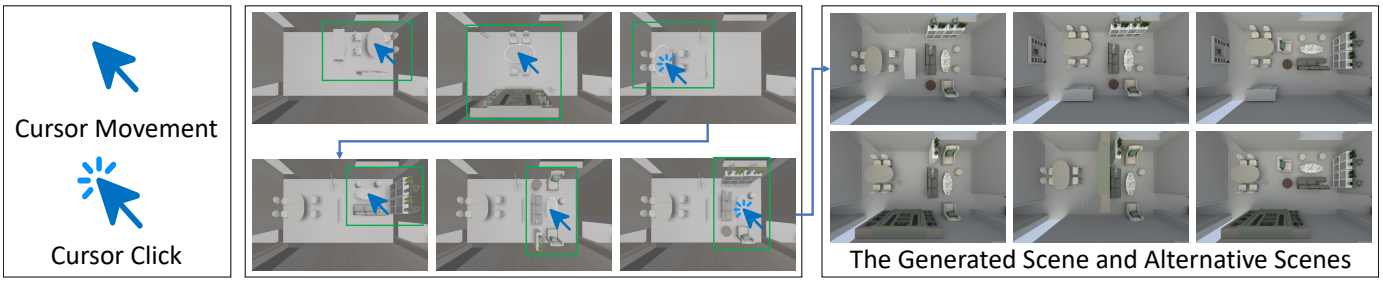


Fig. 1: We present SceneDirector for interactively synthesizing 3D indoor scenes by simultaneously editing groups of objects based on cursor movements and clicks (Left). Given a cursor movement at any moment, our framework automatically inserts, removes, translates, and rotates a group of objects (in green boxes) plausibly into a new scene in real-time, w.r.t the cursor's current position (Middle). Mouse click would end an iteration, thus directly achieving a layout of a group of objects. After a few iterations, a plausible 3D indoor scene is synthesized while alternative results are available depending on user preferences (Right).

relative to layouts, so we also contribute a 3D scene dataset, where each CGS has more than 10 coherent groups to express its concept.

Additionally, we present layout attributes, which are adjustable with respect to the learned concepts. For example, a concept could derive a 3D scene requiring one or two walls surrounding it, while it could also derive a 3D scene as compact as possible, given the same location and room shape. Such ambiguities are resolved by user-specified weights of the layout attributes. For example, a user can strongly demand a scene to be synthesized without a dependent wall by reducing the attribute "dependency" (e.g., the top row in Figure 2). He/she could also increase the attribute "space utilization" to further make the scene compact (e.g., the bottom row in Figure 2). Subsequently, our method gives a convenient control over groups of objects, while it enables the exploration of various layouts through tuning attributes.

The evaluation shows that our work significantly reduces the interaction time in various aspects, e.g., searching objects or transforming objects. We also conduct a usability study to evaluate how users feel more comfortable when interacting with our system, compared with the existing solutions [20], [21]. We also conduct a user study to quantitatively demonstrate a higher plausibility and aesthetics of the resulting scenes of ours than the scenes by the existing solutions. Finally, we present two fully implemented applications based on cursor movements and gestures in VR to show the potentiality of the proposed technique.

In this paper, we make the following contributions:

- We present a new technique for interactively editing groups of objects concurrently in 3D scenes, using minimum inputs such as cursor movements on a desktop environment or hand gestures in VR.
- We quantify the attributes of layouts. By continuously adjusting the attributes, users can easily change how layouts are generated under different contexts.
- We propose the idea of "coherent groups set", which helps restore layout concepts as much as possible. To achieve this, a dataset and an annotation platform are also presented.

2 Related Works

Automatic 3D scene synthesis focuses on the generation of entire layouts. Data-driven techniques range from mathematical formulations of interior rules [6], [23] to neural networks [9], [24]. They fit a series of models to examples in scene datasets and use the fitted or learned models to derive new layouts. For example, [6], [8], [25], [26], [27], [28] formulate a set of models and optionally fit them with datasets of scenes if being data-driven and then synthesize scenes based on MCMC. [9], [19], [24], [29], [30] directly yield 3D scenes by feeding random numbers or top views into neural networks. [23], [31] propose to synthesize scenes based on the geometries of objects and rooms. Although our method could potentially be automated, we focus on developing an interactive system since desired scenes are not unique and often need user inputs to guide the synthesis process. Additionally, compared with data-driven approaches, ours learns a concept with a very coherent set of object groups with functional and aesthetic compatibility instead of learning unified models.

Various scene synthesis techniques have been proposed to generate layouts by considering additional inputs. For example, He et al. [32] propose to generate 3D scenes by taking into account real-world blocks. Hand-drawn sketches have been utilized for deriving scenes [33]. The techniques in [34], [35] interpret texts into scene graphs for generating scenes. RGB-D scans [36], [37], [38] or RGB images [39] have also been used for the physical and visual guidance of scene synthesis. Xiong et al. [40] propose to transfer reference layouts to 3D scenes with motion plans of objects and the technique by Fisher et al. [7] synthesizes scenes with learned models and given example scenes. We propose the use of attribute weights as an additional input, which allows users to tune weights to adjust 3D scenes with respect to the learned concepts.

Interactive 3D scene synthesis is the main focus of our paper and this topic has not been explored extensively. Savva et al. [16] present a tool for enabling object selections given mouse clicks. [10], [11], [15] suggest detailed and small objects so that existing scenes are enriched. [41] generates a series of scenes and gives users choices on them, thus conversely optimizing scenes presented to users. [42] generates scenes given user-specified constraints and

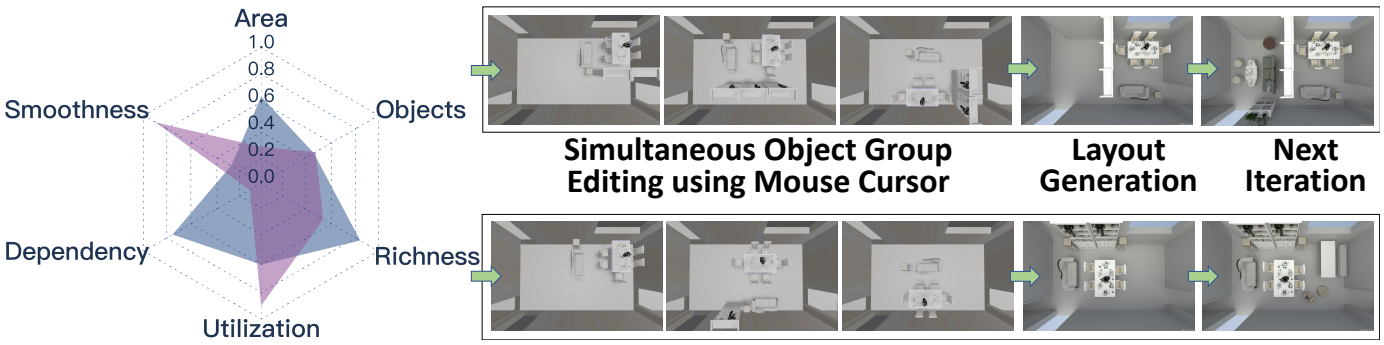


Fig. 2: The pipeline of our method. After a user changes the weights of the attributes, the strategy for object rearrangement changes accordingly. For example, the top row tries arranging object as private as possible (Blue in the attribute weight diagram) while the bottom row tries arranging them as transparent as possible (Purple in the diagram). This entire process is considered a single interaction and is executed twice or more for exploration towards a final scene.

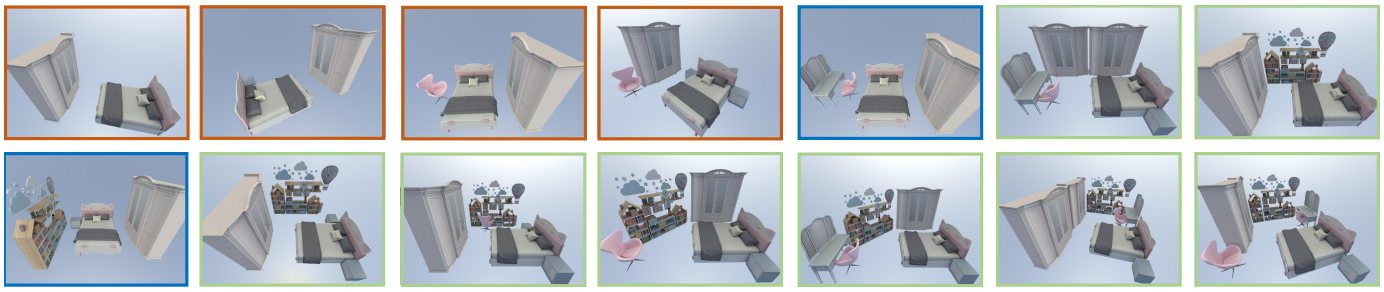


Fig. 3: A typical example of coherent group set with the gentle light style. All groups share the same object pool, which contains all alternative objects to express the concepts of layouts. Depending on contexts such as positions suggested by users and constraints such as attributes and room shapes, a concept could derive an open-plan layout (in orange boxes), a parallel layout (in blue boxes), or a semi-enclosed layout (in green boxes) [22]. The set is crafted from using a very few objects within a pool of objects. In this paper, all the CGSs are crafted by professional interior designers, in order to fully exploit our method.

examples. Some works also investigate passive interactions [43], [44], where layouts are optimized to make workspaces more efficient given subject behaviours. Nevertheless, existing literature focuses on interactions incorporating only a single object each time instead of incorporating a group of objects each time. To the best of our knowledge, we are the first to investigate simultaneous editing of object groups.

3 Methodology

Figure 1 shows the pipeline of our method. A user first selects an existing object or inserts an object into the scene. When she/he suggests a single movement of the selected object using the mouse cursor, groups of objects related to the selected one are simultaneously re-arranged or inserted into the scene (Section 3.1). She/he can also specify different weights of the layout attributes so that objects are arranged further according to the user preference as shown in Figure 2. In our current implementation, we consider the following attributes: area, number of objects, richness, utilization, dependency, and smoothness (Section 3.2). Overall, the aforementioned process is considered a single interactive session.

3.1 Coherent Group Set

A coherent group set is defined as a set of groups $\mathcal{S} = \{G_i | i \in [1, N]\}$. Each group G_i has a finite number of objects, where groups of the same set $\mathcal{S}$ all hold a same and unique dominant object (e.g., the double bed in Figure 3), which is typically the object to be interacted with w.r.t mouse cursor movements or hand gestures. Subsequently, other objects would be adjusted according to the CGS. The unique dominant object in $\mathcal{S}$ is the one leading others and each G_i is constructed according to its transformations and styles. Common dominant objects could be coffee tables, double beds, dinning tables, etc. Other objects are subordinate and they could be sofas, dinning chairs, nightstands, etc. Note that dominant objects are interactively selected by users, so an object can be dominant in one CGS but not in another CGS. Each object in G_i has an index to a particular object instance such as a CAD model. It also has a relative transformation w.r.t the dominant object. Figure 3 shows an original CGS designed by a professional interior designer, in which the dominant object is the double bed.

As illustrated in Figure 1, our system calculates optimal coherent groups G^t in real-time with respect to the movements of casted locations $\vec{x}$ and the room shape R. Specifically, this problem is formulated in Equation 1, such that two more conditions are satisified as shown in equation 2:

$$G^t = \arg\max \vec{\eta}^T g_i(\vec{x}, R, G^{t-1}), \quad (1)$$

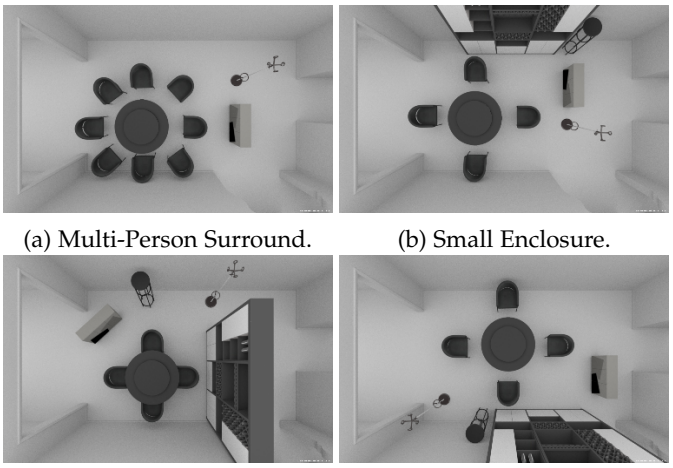
$$\ni \begin{cases} G_i \in \mathcal{S} \cup \mathcal{S}' \\ d(\vec{x}, R) \geq D(G_i), \end{cases} . \quad (2)$$

where G_i is a potential object group to be synthesized, G^{t-1} is the currently calculated object group before the cursor movement, and the position $\vec{x}$ could be a location casted by the mouse cursor or a hand gesture (Section 4). $g_i(\vec{x}, R, G^{t-1})$ returns the attributes of a coherent group G_i and $\vec{\eta}$ contains the user-adjustable weights of the attributes. Some attributes such as area and space utilization are proprietary values of G_i, while others require a context, e.g., R.

Each $g_i(\cdot)$ corresponds to a $G_i \in \mathcal{S} \cup \mathcal{S}'$, where $\mathcal{S}'$ is the expanded version of $\mathcal{S}$. Since too few instances in $\mathcal{S}$ are insufficient to express a layout concept, we multiplex $\mathcal{S}$ as far as possible. First, we take the power set $\{G' \cup \{o_i^{dom}\} | G' \subseteq (G_i \setminus \{o_i^{dom}\})\} \setminus \{\emptyset\}$ for each G_i, where o_i^{dom} denotes the user-specified dominant object. Since the empty set is meaningless, it is removed from the power set. An interactive session should always contain the dominant object, so the power set operation does not include o_i^{dom}, which is added to all generated subsets. Second, for each G' in the power set, we further derive three groups by flipping it vertically, horizontally, and diagonally. Since for a large G_i, taking its power set could have excessive calculation pressure, we sample M ($M = 75$ in our implementation) G' to prevent $\mathcal{S}'$ being too large.

Fig. 4: Illustration of expansion spaces.

The $d(\vec{x}, R) \geq D(G_i)$ is a rigid requirement, since we never want furniture being outside or embedded into walls. $d(\cdot)$ and $D(\cdot)$ respectively return the expansion space w.r.t $\vec{x}$ and G_i. The illustration of expansion spaces is given in Figure 4. An expansion (blue dashed rectangles) of a coherent group G_i is conducted by the anchor μ_a, left μ_l, right μ_r, and depth μ_d (purple arrows) w.r.t the dominant object (blue circle), where the distance from the nearest wall (gray solid rectangle) to it corresponds to μ_a. μ_d refers to the distance from the wall towards the nearest wall. μ_l and μ_r are respectively the left-side and right-side expanded distance w.r.t μ_a. An expansion (green solid rectangle) of $\vec{x}$ and R is conducted by further expanding $D(G_i)$ so that a larger rectangle is derived. For G_i, we are finding a maximal rectangular area given objects in G_i. For $\vec{x}$, we are finding a minimal rectangular area in the room shape R. Therefore, this condition requires that $(\mu_a, \mu_l, \mu_r, \mu_d)$ are all smaller than $d(\vec{x}, R)$.



(a) Multi-Person Surround. (b) Small Enclosure.

(c) Private Enclosure. (d) Transparency.

Fig. 5: Four example scenes with the grey modern style for illustrating layout attributes.

A tiny collision may destructively influence results because the results of 3D scene synthesis are subjective to humans. To avoid collisions, we consider object-object detection and object-door/window detection. Each object is decomposed into a set of constituent components, e.g., desktop and desk legs. Any component of an object intersecting with a component of another object is considered a collision. Windows and doors are considered as single components that are cuboids elevated in the normal direction of their depending walls. If there is a collision between $o' \in G_i$ and an existing entity in the scene, G_i would remove o as $G_i \setminus \{o'\}$. After calculating G^t, the layout of objects at $t-1$ time is re-organized accordingly, which could be messy since objects might drastically move together, thus reducing the visual experience. Section C of the supplementary document addresses this issue.

3.2 Layout Attributes

3D scenes are complicated [45]. In some cases, attributes may have respective effects on how objects are arranged. They may also contradict each other. Thus, we present a typical usage of the attributes based on our observations, and give the balance control of the trade-offs to users. All the attributes are pre-computed before the pipeline described by Equation 1 operates. Note that all attributes are eventually normalized in $[0, 1]$ in order to have the same magnitude.

Area. In most cases, we want a coherent group G_i to be as large as possible, in order to fill up a given room. The area of G_i refers to the area of the expansion space of G_i (i.e., dashed blue rectangles in Figure 4), where the gap spaces among objects are considered due to aesthetic and affordance. In practice, this attribute frequently contradicts with space utilization $A_u(G_i)$ (to be introduced soon). For example, the layout in Figure 5a is more compact but the layout in figure 5b is larger.

Space Utilization. This attribute $A_u(G_i)$ describes how efficient G_i exploits the given space measured by $A_a(G_i)$. The space utilization is calculated by the following equation:

$$A_u(G_i) = A_p(\mathcal{P}(o_1) \cup ... \cup \mathcal{P}(o_N))/A_a(G_i), o_i \in G_i. \quad (3)$$

where $\mathcal{P}(\cdot)$ projects each object $o_i \in G_i$ into the ground using the mesh of o_i. By taking the union of each projected mesh, a polygon is unified and its area is calculated by $A_p(\cdot)$. Therefore, $A_u(G_i)$ is equal to the ratio of the unified polygon to the area expanded by $D(G_i)$.

Number of Objects. This attribute $A_n(G_i) = |G_i| - 1$ simply counts the number of object in group G_i, excluding the dominant object. This attribute is used with the other attributes. For example, to acquire a compact scene with as many objects as possible, a user may tune both $A_n(G_i)$ and $A_u(G_i)$ up. Otherwise, being used individually, this attribute may not yield expected results for users.

Richness. This attribute $A_r(G_i)$ measures the potential functionality of G_i, by counting the number of sub-groups inside G_i. For example, in a bedroom, a double bed with two nightstands is considered as a sub-group, and a dressing table with an ottoman is considered as another sub-group. Thus, dividing G_i into sub-groups is equivalent to finding potential relations in G_i, where the relations can involve three or more objects and we employ an existing method [31] to determine such relations.

Dependency. For home decorations, many objects are designed to be strictly placed against walls, e.g., the cabinet in Figure 5b. In contrast, objects such as cabinets and wardrobes could also be assembled for zoning spaces at the early stage of the residential design [12], [14], e.g., the scene in Figure 5c. Thus, formulated in Equation 4, dependency $A_d(G_i)$ measures how G_i is inclined to require walls to support it. A higher value of A_d results in "wall supporting" while a lower value results in "space zoning". Entries in $\vec{\xi}$ are either 0 or 1, denoting the dependencies w.r.t the anchor, left, right, and depth. Calculating the entry follows the same rule of calculating $D(G_i)$ for coherent groups. For example, if a wardrobe is against the left side of a coherent group, the "anchor" of the wardrobe results in the "left" of the group. The function $\phi(\cdot)$ truncates the distances to 0 or 1, i.e., if the anchor is sufficiently close to the wall the dependency is considered satisfied, and vice versa. $\circ$ denotes the element-wise multiplication, where the L1 norm finally sums the entries of the result vector together. Therefore, we first calculate the differences between the two expansions and truncate them. As long as $\vec{\xi}$ denotes that dependencies in some (or all) boundaries are required, the corresponding exponents are valid.

$$A_d(G_i, R) = ||\phi((d(\vec{x}, R) - D(\mathcal{G}_i(\vec{x}))) \circ \vec{\xi})||_1. \quad (4)$$

Smoothness. To interact with coherent groups, we realize that the transition from one G^t to another G^{t+1} could be overdramatic, e.g., objects in G^t are totally swapped instead of modifying the transformations of merely one or two objects. As a result, smoothness is calculated between groups to measure their differences. $A_s(G_i, G_j)$ takes in two groups. We follow the method of Fisher et al. [46], which uses graph kernels [47] to measure how two scenes are different. For node kernels, we simply follow their solutions, but for edge kernels groups in CGS may have totally identical relationships (Enclosure, Horizontal Support, Vertical Contact, Oblique Contact in [46]) between objects. The Kronecker delta kernel used in [46] returns whether the two edges being compared are tagged with the same contact type. It does not apply in our cases where objects are distanced and the distance may vary due to the size of the room area, e.g., the scenes in Figure 5b and 5d. Therefore, measurements based on transformations in the 3D space are required. We thus modify the method in [46]. See the detail in Section D of the supplementary document.

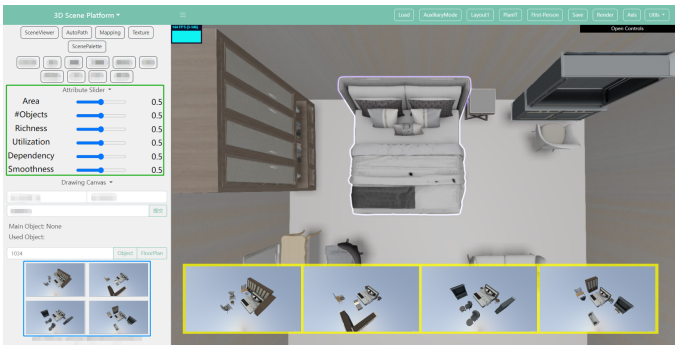
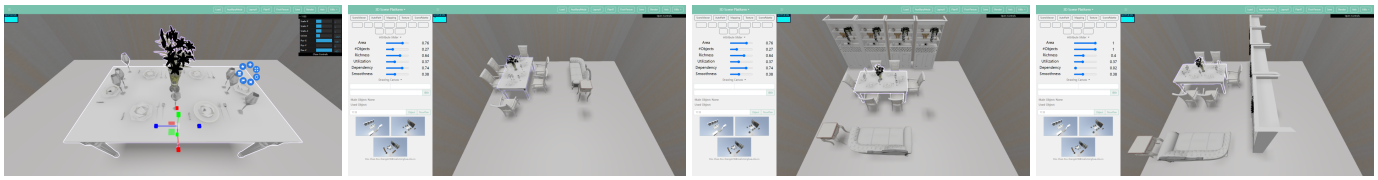


Fig. 6: The prototype system with the proposed CGS-based method. On the left, the attribute panel (in green box) allows users to tune the weights of the attributes, and the search panel (in blue box) allows users to swap alternative CGSs (highlighted in the bottom yellow box) w.r.t the dominant object (highlighted in purple). On the right, users could select a dominant object by clicking it. With the movement of the cursor, the double bed moves accordingly and the layout is adjusted in real-time.

4 System and Dataset

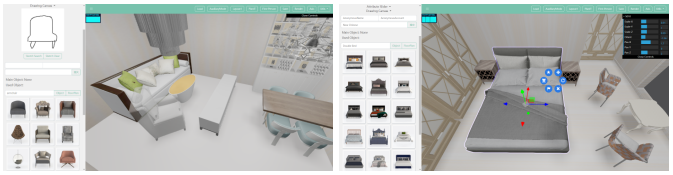
In this section, we design a prototype system to show how we utilize CGS for interactive scene synthesis/editing by object group editing. As shown in Figure 6, the UI of our system consists of three parts. First, in a 3D scene, a user could select a dominant object and enter the CGS mode. Theoretically, a dominant object could be any entity. For simplicity, this paper recognizes entities such as coffee tables or double beds as dominant objects, which typically represent the functionality of coherent groups. Second, it has a panel for users to tune the weights on layout attributes. Since the attributes are normalized, the weights are in the range of $[0, 1]$, where 0 denotes totally ignoring an attribute and 1 denotes fully engaging an attribute. Third, users could select various CGSs in a search panel, where each CGS stands for a particular concept and is represented by its final form, which is a layout derived by the concept with the greatest extent without any constraint. In Figure 6, the four presented scenes stand for four different CGSs with the dominant double bed in the 3D scene. When a user clicks a CGS in that panel, the layout concept is swapped accordingly and our method operates on the newly selected concept.

Figure 7 shows a typical interactive process of our system, where a user sets a proper scale of the dominant object w.r.t the room and starts to edit object groups. Different layouts are established, given different layout attributes and casted positions of the cursor. Clicking the cursor would add the suggested layout into the scene and end an interaction. A supplementary video shows more demonstrations on object group editing in real-time.



(a) (b) (c) (d)

Fig. 7: An interactive editing process. The interacted object is highlighted in purple. a) in the beginning, users could re-scale the dominant object, and thus rescale the entire CGS. b) when the cursor is close to the corner, the space constraint (see Equation 1) is hard to satisfy, so a small scene is compromised. c) given sufficient space, the CGS generates a scene as large as possible. d) by changing the attribute weights, e.g., reducing the dependency, a more private scene is presented.

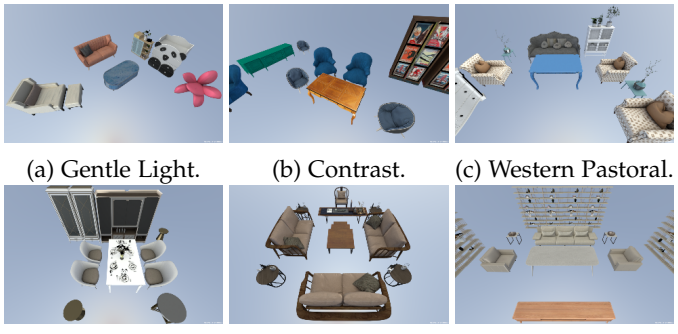


(a) Search and Insert. (b) Transform and Submit.

Fig. 8: The system for annotations and user studies. It supports industrial operations on 3D scenes. a) designers could search for a style-compatible object and insert it into the scene. b) the object can be translated, rotated, and re-scaled through three panels: the blue panel using cursors, the local coordinate system, and the value setting panel in the upper left corner. After finishing a coherent group, she/he could submit it with a CGS name (red box).

To implement our ideas, we need a CGS dataset. Initially, we tried 3D-Front [48], which, however, has the following problems: (1) The usage of dominant objects w.r.t subordinate objects is over sparse, i.e., the size $|\mathcal{S}|$ of each CGS could be very small for each dominant object, thus being insufficient for deriving layout concepts[1]. (2) Layout styles of 3D-Front are limited, e.g., the four variations in Figure 5 are hard to be derived from 3D-Front. (3) It is also hard to decompose different object groups from rooms, e.g., a large bay may contain both a bed set and a coffee table set but

1. See Section B in the supplementary document for the statistics.



(a) Gentle Light. (b) Contrast. (c) Western Pastoral.
(d) Luxury. (e) New Chinese. (f) Minimalism.

Fig. 9: Examples in our dataset. Each example has an evolutionary chain similar to Figure 3. More details are shown in Section B of the supplementary document.

(a) (b) (c)

Fig. 10: Interacting with objects using "Force". Our method allows mouse-free interactions in VR using hand gestures. The time axis shows the movement of the hand. More demonstrations are also shown in the video.

they are too close to separate.

Therefore, we invited 25 professional interior designers to create more layouts to address the above issues, leading to a new dataset focusing on density and layout variations. We use the same system in Figure 6 for annotating layouts: designers could search, insert, translate, and rotate objects as shown in Figure 8. Designers could also edit room shapes if they are too small to hold a coherent group. To ensure style compatibility, we partition CAD models into different style categories and require objects in the same CGS to be from a unique style category. Figure 9 shows representative samples of the dataset. Due to our limited budget, the dataset contains $1,719$ layouts, which will be released to the public and expanded in the future.

We also develop a VR-oriented client to exploit more potentialities of CGS. In immersive virtual environments, no more mouse click or cursor is available. Instead, we typically use controllers with hand gestures for interacting with 3D scenes. Existing literature researching 3D scene interactions in VR based on hand gestures has explored one-to-one gesture-command mapping without much intelligent response by environments, where each interaction involves up to a single object [49], [50], [51], [52]. Leveraging the proposed method, we present a natural hand gesture interaction with groups of objects. As a response to the gestures of users, our system suggests different layouts into a VR scene in real-time. Figure 10 illustrates the process of how users create new 3D scenes in VR with our system, where the dominant object is detected by eye-tracking instead of using the cursor, and the 3D scene is synthesized according to hand gestures instead of cursor hovering. More technical details can be found in Section E of the supplementary document.

The SceneDirector was implemented in Three.js (Web), Unity (VR) and Flask. Our implementation follows a client-

server environment. Our server runs on a Ryzen 2700x machine with 16GB of RAM. Our method runs in real-time, thus being suitable for nearly all modern devices including those with integrated graphics cards. Extracting layout attributes and boundaries requires less than 10 seconds for a CGS with 15 groups and a maximum of 10 objects.

5 Evaluation

To evaluate the utility of SceneDirector, we conduct a user study to measure the efficiency, usability, and result quality of the system compared with existing industrial approaches. We recruited 35 subjects to interactively craft indoor scenes using our system. Invited subjects were composed of university students, office workers, freelancers, designers, etc. All subjects frequently used computers and reported experiences on using tools for 3D modeling or for editing office documents. Given an empty room, subjects were asked to generate two satisfied layouts as quickly as possible, until they felt satisfied with the crafted layouts. The two satisfied layouts were crafted using the following two settings:

In Setting 1, similar to Kujiale[2] and Planner5D[3], the system provides the typical industrial solutions for manipulating objects, as shown in Figure 8. For searching, users could search objects with keywords, sketches [53], or styles. Styles of objects are manually classified by several professional interior designers, e.g., new Chinese style or grey modern style. Users could also directly click a frequent keyword in a recommended list. Users could manipulate (translate, rotate, and/or scale) objects by a transform controller, a cursor-based approach that transforms a selected object with the movement of the cursor, or a form that directly accepts values (coordinates), where the object alignment is enabled, e.g., for object surrounding layout. For adding objects, users could click a searched result and the object would follow the movement of the cursor until object insertion, or they could duplicate existing objects with a single click.

In Setting 2, the system provides the proposed method. The subjects could alternatively adjust the positions and orientations of objects. They could also optionally search and add more objects if they want.

Two settings were randomly assigned to the participants. We made sure that half of the subjects conduct Setting 1 first and the rest of them conduct Setting 2 first. For each subject, we randomly assigned them a scene type including bedroom, hotel room, living room, dinning room or kids room. They were shown several well-designed layouts crafted by interior designers as a standard in advance to avoid generating low-quality layouts.

We prepared a manual to guide the participants through the operations, and each participant was taught how to use our system to manipulate objects and to use the proposed method. They were free to play with the system until they were comfortable with it before they conducted the study under Settings 1 & 2. All the subjects reported being familiar with the functionalities with less than 15 minutes. During the experiments, a technical staff was available in case of any technical questions.

2. https://b.kujiale.com/
3. https://planner5d.com/

5.1 Time Consumption

This section measures how our method saves time consumed by interactive 3D scene synthesis. Table 1 shows the average time spent on different user interactions including: (1) navigating the scene where users adjust views to interact with scenes from different perspectives. (2) searching objects where users search favourite objects through keywords, styles and sketches. (3) adding objects where users drag searched results into scenes or duplicate objects. (4) removing objects from scenes. (5) translating objects through the three control panels. (6) rotating objects. (7) rescaling objects. (8) the proposed method. (9) other time consumption including elaborations, considerations, misoperations, etc, which are not trackable by our system. We add timers to every unit operation to make sure that each consumed time is correctly recorded by our system, so the systematic errors are negligible. Engineering details of the timers are included in Section F of the supplementary document. Note that we still allow participants to use operations from Setting 1 to give further preferences to them, and counting on this time our method still shows significant interactive time savings.

Each cell in Table 1 contains an average time and a standard deviation in brackets. Time is recorded in seconds. According to Table 1, our method significantly reduces the overall time required to craft a 3D scene. A Kruskal-Wallis H-Test shows that there are significant statistical differences between the total time for Setting 1 and the total time for Setting 2, with the p-value of 0.0452. Because our method directly operates on groups of objects, the typical routine for searching and adding objects is much less conducted. Although most participants wished to customize their rooms, the time required for transforming (translating, rotating, and rescaling) objects is still improved by ours, since groups of objects are plausibly arranged by ours. Another overwhelmingly convenient feature is it prevents users from "making mistakes", simply because humans could not pay attention all the time, e.g., mis-deleting an object and backtracking this instruction (Ctrl+Z in our system). As claimed that our method gives users precognition during the real-time cursor movements in the scene, it also helps the participants with designing scenes. To request fewer objects from our system, we observe that some users lowered the "#Objects" weight while others simply manually removed the unwanted objects, so the removing time of ours is slightly higher.

We observe that the standard deviation values in Table 1 are relatively large. This is possibly because of the following two reasons. First, the room types randomly assigned to users were different, and living-dinning rooms were more complicated than the bedrooms in most cases. Therefore, the subjects assigned with the former typically consumed more time on thinking and arranging. Second, the background skills of them also vary: some subjects reported strong skills on interacting with virtual world (e.g., video games) and some subjects even majored in arts, but other subjects were only familiar with document editing.

5.2 User Satisfaction

This section measures how our system is satisfied by users. Table 2 shows the statistics of user satisfactions on our method, where each cell includes an average value

TABLE 1: Time Consumptions on Setting 1 (S1) & Setting 2 (S2).

	Navigating	Searching	Adding	Removing	Translating	Rotating	Rescaling	Ours	Others	Total
S1	12.5 (10.3)	54.0 (45.1)	25.3 (16.3)	0.4 (0.9)	11.4 (8.3)	11.5 (9.7)	2.8 (4.0)	0.0 (0.0)	67.4 (38.4)	185.3 (78.0)
S2	7.8 (9.4)	21.0 (20.5)	6.8 (5.5)	1.5 (4.5)	8.3 (10.4)	2.4 (3.3)	2.3 (2.8)	24.0 (14.7)	30.5 (25.1)	104.7 (53.4)

TABLE 2: User Satisfaction.

Measurement	Setting 1	Setting 2
Interactive Satisfaction	2.9 (1.0)	4.0 (0.8)
Result Satisfaction	3.6 (0.9)	3.8 (0.7)

and a standard deviation in brackets. Once each subject finished with Settings 1 & 2, we asked them to mark their overall satisfaction with the results. Likert-scale is adopted from 0 "results are totally inaesthetic and implausible" to 5 "results are very aesthetic and plausible". We also asked them to mark how they were comfortable and convenient during the entire process of interactions. The Likert-scale ranges from 0 "the interaction is annoying and inconvenient" to 5 "the interaction is delightful and convenient". Our method of simultaneously editing groups of objects was favored by most of the participants according to the "interactive satisfaction", and a Kruskal-Wallis H-Test also shows a significant difference, with the p-value extremely close to 0.0. By interviewing the subjects after their user studies, we found most of them really enjoyed exploring and foreseeing potential layouts in real-time. As for the "result satisfaction", since we asked the participants to customize a plausible scene until they were satisfied, the results did not show a big difference. However, ours is still competitive with Setting 1 because our real-time method gives users more information during their crafting scenes.

5.3 Aesthetic and Plausibility

This section further measures the aesthetics and plausibility of our method. We conduct another user study to quantitatively measure the results of our method. Another 48 participants were invited to evaluate the generated scenes by Settings 1 and setting 2. The newly recruited participants were also composed of university students, office workers, freelancers, designers, etc. They were only presented a series of questions, without being told about the previous study. Each question contained two rendered scenes respectively generated from Settings 1 and 2. The participants were asked to select a favored scene from the presented images. Questions were shuffled for each questionnaire and rendered scenes presented to the subjects in each question were also shuffled. See Section G of the supplementary document for more details of the online platform for conducting this user study. The result shows in 46.181% of the comparisons, where scenes generated by ours were favored, the scenes from Setting 1 were preferred for 38.044% of the comparisons, and there was no preference for the rest of the comparisons. Since the scenes generated by ours were favored by nearly half of the subjects, we could conclude that our method generated competitive results compared with the traditional solution.

5.4 Attribute Tuning

As a work on attributing layouts, we have to evaluate how meaningful the proposed attributes are to users and whether users with different preferences could benefit from tuning attributes. Thus, we conduct an independent user study to collect how users select a comfortable setting. The participants were the same group from Section 5.1. They were initially told about the meaning of the six attributes. Subsequently, a technical staff majoring in interior design controlled the cursor and showed them the synthetic process of three layouts: a coffee table set, a dinning table set, and a double bed set. During each layout adjustment w.r.t the cursor movements, the technical staff constantly asked questions about the arrangements of objects, e.g., "Do you want the scene to become larger or more compact?", "Do you want the scene to change gradually or variously?", etc. According to the perceptions of the participants, the staff would iteratively tune the weights until they captured a favourite arrangement.

The kernel density estimation of the weights is visualized in Figure 11, where we choose a kernel as $(1 - |(y - y')/\delta|^2)/\delta$ for all $|y - y'| \leq \delta$, where y' denotes the original data points. The bandwidth δ is empirically set as 0.2 since we want the original distribution instead of a smooth curve. From the curves, the distributions of the weights spread across the x-axis, meaning all the attributes give rational plays to their roles. For example, to enable a private layout, users typically reduced the weight for dependency and raised the weight for richness, while some users still wanted the layout concept to expand fewer objects, thus reducing the weights of #objects. Some attributes have their own frequently chosen values such as 0.5 for utilization or 0.8 for the area, which makes sense since the layout attributes are meaningful to users, e.g., most users want the layout concept to expand as large as possible, so they raise the weight of area.

6 Conclusion

We have proposed a method for simultaneously editing groups of objects with an idea of coherent group set and layout attributes. To the best of our knowledge, our work is the first interactive scene synthesis system supporting interactive control of multiple objects. We hope our work will be inspiring for the follow-up works.

The user preferences are acceptable in two forms in our implementation: user-specified positions and user-specified weights. The former is used for suggesting where to put a

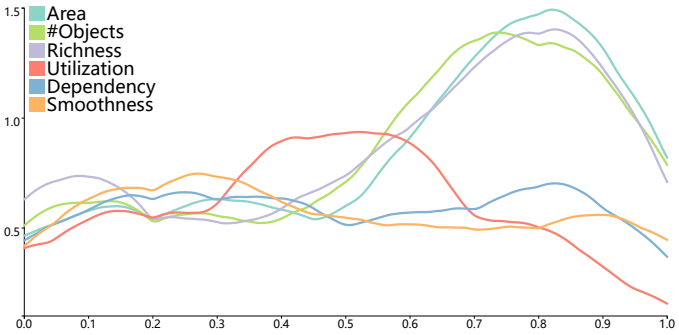


Fig. 11: Distributions of the weights selected by the subjects.

layout and the latter is used for suggesting how to put a layout. We have also explored hand gestures as an alternative to the former. In future works, we are interested in exploring more potential forms of user input to strengthen user preferences in scene synthesis. To express a specific layout concept, although this paper presents a way to approximate it using 10-20 examples, examples are never sufficient. We should either create a larger set of examples or multiplex existing examples more efficiently and creatively. The former consumes more expenditure and the latter is considered a methodological improvement. The collision strategy of this paper is simply removing all collided objects in CGS, if the objects are collided with doors, windows or the objects in other groups. Avoiding the collision only makes a scene become more valid, but this could break the strategy of how objects are arranged, e.g., if the cabinets are removed in Figure 5c due to a collision w.r.t a door, the entire layout will be no longer private. Thus, a smarter strategy is to readjust both involved entities, possibly requiring learning relations at a "group" level.

Acknowledgments

This work was supported by the Natural Science Foundation of China (Project Number 62132012) and Tsinghua-Tencent Joint Laboratory for Internet Innovation Technology.

References

[1] J. D. N. Dionisio, W. G. B. III, and R. Gilbert, "3d virtual worlds and the metaverse: Current status and future possibilities," *ACM Computing Surveys (CSUR)*, vol. 45, no. 3, pp. 1–38, 2013.

[2] W. Li, J. Talavera, A. G. Samayoa, J.-M. Lien, and L.-F. Yu, "Automatic synthesis of virtual wheelchair training scenarios," in *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 2020, pp. 539–547.

[3] S.-H. Zhang, C.-H. Chen, Z. Fu, Y. Yang, and S.-M. Hu, "Adaptive optimization algorithm for resetting techniques in obstacle-ridden environments," *IEEE Transactions on Visualization and Computer Graphics*, 2022.

[4] A. Handa, V. Patraucean, V. Badrinarayanan, S. Stent, and R. Cipolla, "Understanding real world indoor scenes with synthetic data," in *Proceedings of the IEEE CVPR*, 2016, pp. 4077–4085.

[5] F. D. Ching and C. Binggeli, *Interior design illustrated*. John Wiley & Sons, 2018.

[6] L.-F. Yu, S. K. Yeung, C.-K. Tang, D. Terzopoulos, T. F. Chan, and S. Osher, "Make it home: automatic optimization of furniture arrangement." *ACM Trans. Graph.*, vol. 30, no. 4, p. 86, 2011.

[7] M. Fisher, D. Ritchie, M. Savva, T. Funkhouser, and P. Hanrahan, "Example-based synthesis of 3d object arrangements," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 6, p. 135, 2012.

[8] S. Qi, Y. Zhu, S. Huang, C. Jiang, and S.-C. Zhu, "Human-centric indoor scene synthesis using stochastic grammar," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5899–5908.

[9] K. Wang, Y.-A. Lin, B. Weissmann, M. Savva, A. X. Chang, and D. Ritchie, "Planit: Planning and instantiating indoor scenes with relation graph and spatial prior networks," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, p. 132, 2019.

[10] L.-F. Yu, S.-K. Yeung, and D. Terzopoulos, "The clutterpalette: An interactive tool for detailing indoor scenes," *IEEE transactions on visualization and computer graphics*, vol. 22, no. 2, pp. 1138–1148, 2015.

[11] S. Zhang, Z. Han, Y.-K. Lai, M. Zwicker, and H. Zhang, "Active arrangement of small objects in 3d indoor scenes," *IEEE transactions on visualization and computer graphics*, vol. 27, no. 4, pp. 2250–2264, 2019.

[12] H.-L. Lu, *Residential Interior Design*. Liaoning Science and Technology Publishing House, 2010.

[13] T.-K. Zheng, *Interior Design For Home*. Huazhong University of Science & Technology Press, 2011.

[14] A. B. Vasiliki Asaroglou, *Furniture arrangement: in Residential spaces*. CreateSpace Independent Publishing Platform, 2013.

[15] S. Zhang, Z. Han, and H. Zhang, "User guided 3d scene enrichment." in *VRCAI*, 2016, pp. 353–362.

[16] M. Savva, A. X. Chang, and M. Agrawala, "Scenesuggest: Context-driven 3d scene design," *arXiv preprint arXiv:1703.00061*, 2017.

[17] S.-K. Zhang, Y.-X. Li, Y. He, Y.-L. Yang, and S.-H. Zhang, "Mageadd: Real-time interaction simulation for scene synthesis," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 965–973.

[18] M. Mitton and C. Nystuen, *Residential interior design: A guide to planning spaces*. John Wiley & Sons, 2016.

[19] K. Wang, M. Savva, A. X. Chang, and D. Ritchie, "Deep convolutional priors for indoor scene synthesis," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, p. 70, 2018.

[20] kujiale.com, "Kujiale," dec 2020. [Online]. Available: https://www.kujiale.com/

[21] planner5d.com, "Planner5d," dec 2020. [Online]. Available: https://planner5d.com/

[22] Y.-G. Peng, *Architectural space combination theory*. China Architecture & Building Press, 2008.

[23] Q. Fu, X. Chen, X. Wang, S. Wen, B. Zhou, and H. Fu, "Adaptive synthesis of indoor scenes via activity-associated object relation graphs," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 6, pp. 1–13, 2017.

[24] Z. Zhang, Z. Yang, C. Ma, L. Luo, A. Huth, E. Vouga, and Q. Huang, "Deep generative modeling for scene synthesis via hybrid representations," *ACM Transactions on Graphics (TOG)*, vol. 39, no. 2, pp. 1–21, 2020.

[25] 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 Transactions on Graphics (TOG)*, vol. 31, no. 4, p. 56, 2012.

[26] Y. Liang, S.-H. Zhang, and R. R. Martin, "Automatic data-driven room design generation," in *International Workshop on Next Generation Computer Animation Techniques*. Springer, 2017, pp. 133–148.

[27] T. Weiss, A. Litteneker, N. Duncan, M. Nakada, C. Jiang, L.-F. Yu, and D. Terzopoulos, "Fast and scalable position-based layout synthesis," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 12, pp. 3231–3243, 2018.

[28] S.-H. Zhang, S.-K. Zhang, W.-Y. Xie, C.-Y. Luo, Y.-L. Yang, and H. Fu, "Fast 3d indoor scene synthesis by learning spatial relation priors of objects," *IEEE Transactions on Visualization and Computer Graphics*, 2021.

[29] M. Li, A. G. Patil, K. Xu, S. Chaudhuri, O. Khan, A. Shamir, C. Tu, B. Chen, D. Cohen-Or, and H. Zhang, "Grains: Generative recursive autoencoders for indoor scenes," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 2, pp. 1–16, 2019.

[30] D. Ritchie, K. Wang, and Y.-a. Lin, "Fast and flexible indoor scene synthesis via deep convolutional generative models," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6182–6190.

[31] S.-K. Zhang, W.-Y. Xie, and S.-H. Zhang, "Geometry-based layout generation with hyper-relations among objects," *Graphical Models*, vol. 116, p. 101104, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1524070321000096

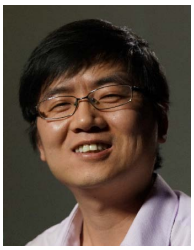
[32] Y. He, Y. Liu, Y. Jin, S.-H. Zhang, Y.-K. Lai, and S.-M. Hu, "Context-consistent generation of indoor virtual environments based on geometry constraints," *IEEE Transactions on Visualization & Computer Graphics*, no. 01, pp. 1–1, 2021.
[33] K. Xu, K. Chen, H. Fu, W.-L. Sun, and S.-M. Hu, "Sketch2scene: sketch-based co-retrieval and co-placement of 3d models," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 4, p. 123, 2013.
[34] A. Chang, W. Monroe, M. Savva, C. Potts, and C. D. Manning, "Text to 3d scene generation with rich lexical grounding," *arXiv preprint arXiv:1505.06289*, 2015.
[35] R. Ma, A. G. Patil, M. Fisher, M. Li, S. Pirk, B.-S. Hua, S.-K. Yeung, X. Tong, L. Guibas, and H. Zhang, "Language-driven synthesis of 3d scenes from scene databases," in *SIGGRAPH Asia 2018 Technical Papers*. ACM, 2018, p. 212.
[36] K. Chen, Y. Lai, Y.-X. Wu, R. R. Martin, and S.-M. Hu, "Automatic semantic modeling of indoor scenes from low-quality rgb-d data using contextual information," *ACM Transactions on Graphics*, vol. 33, no. 6, 2014.
[37] M. Fisher, M. Savva, Y. Li, P. Hanrahan, and M. Nießner, "Activity-centric scene synthesis for functional 3d scene modeling," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 6, p. 179, 2015.
[38] K. Chen, Y.-K. Lai, and S.-M. Hu, "3d indoor scene modeling from rgb-d data: a survey," *Computational Visual Media*, vol. 1, no. 4, pp. 267–278, 2015.
[39] A. Luo, Z. Zhang, J. Wu, and J. B. Tenenbaum, "End-to-end optimization of scene layout," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3754–3763.
[40] G. Xiong, Q. Fu, H. Fu, B. Zhou, G. Luo, and Z. Deng, "Motion planning for convertible indoor scene layout design," *IEEE Transactions on Visualization and Computer Graphics*, 2020.
[41] P. Merrell, E. Schkufza, Z. Li, M. Agrawala, and V. Koltun, "Interactive furniture layout using interior design guidelines," in *ACM transactions on graphics (TOG)*, vol. 30, no. 4. ACM, 2011, p. 87.
[42] M. Yan, X. Chen, and J. Zhou, "An interactive system for efficient 3d furniture arrangement," in *Proceedings of the Computer Graphics International Conference*, 2017, pp. 1–6.
[43] W. Liang, J. Liu, Y. Lang, B. Ning, and L.-F. Yu, "Functional workspace optimization via learning personal preferences from virtual experiences," *IEEE transactions on visualization and computer graphics*, vol. 25, no. 5, pp. 1836–1845, 2019.
[44] Y. Zhang, H. Huang, E. Plaku, and L.-F. Yu, "Joint computational design of workspaces and workplans," *ACM Transactions on Graphics (TOG)*, vol. 40, no. 6, pp. 1–16, 2021.
[45] Y. Li, J. Zhang, Y. Cheng, K. Huang, and T. Tan, "Df 2 net: Discriminative feature learning and fusion network for rgb-d indoor scene classification," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.
[46] M. Fisher, M. Savva, and P. Hanrahan, "Characterizing structural relationships in scenes using graph kernels," in *ACM SIGGRAPH 2011 papers*, 2011, pp. 1–12.
[47] S. V. N. Vishwanathan, N. N. Schraudolph, R. Kondor, and K. M. Borgwardt, "Graph kernels," *Journal of Machine Learning Research*, vol. 11, pp. 1201–1242, 2010.
[48] H. Fu, B. Cai, L. Gao, L.-X. Zhang, J. Wang, C. Li, Q. Zeng, C. Sun, R. Jia, B. Zhao, and H. Zhang, "3d-front: 3d furnished rooms with layouts and semantics," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 10 933–10 942.
[49] H. Kim, G. Albuquerque, S. Havemann, and D. W. Fellner, "Tangible 3d: Hand gesture interaction for immersive 3d modeling." *IPT/EGVE*, vol. 2005, pp. 191–9, 2005.
[50] J. Kela, P. Korpipää, J. Mäntyjärvi, S. Kallio, G. Savino, L. Jozzo, and S. D. Marca, "Accelerometer-based gesture control for a design environment," *Personal and Ubiquitous Computing*, vol. 10, no. 5, pp. 285–299, 2006.
[51] N. H. Dardas and M. Alhaj, "Hand gesture interaction with a 3d virtual environment," *The Research Bulletin of Jordan ACM*, vol. 2, no. 3, pp. 86–94, 2011.
[52] Y. Li, X. Wang, Z. Wu, G. Li, S. Liu, and M. Zhou, "Flexible indoor scene synthesis based on multi-object particle swarm intelligence optimization and user intentions with 3d gesture," *Computers & Graphics*, vol. 93, pp. 1–12, 2020.
[53] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3d shape recognition," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 945–953.

Shao-Kui Zhang is a Ph.D. candidate in the Department of Computer Science and Technology at Tsinghua University, Beijing, China. His research interests include computer graphics, 3D scene synthesis, intelligent 3D scene interaction. He received a B.S. degree of software engineering from Northeastern University, Shenyang, in 2018.

Hou Tam received his bachelor's degree in computer science from Tsinghua University in 2021. He is currently a master student in the Department of Computer Science and Technology, Tsinghua University.

Yi-Ke Li received her bachelor's degree in computer science from Wuhan University of Science and Technology, Wuhan, in 2020. She is currently pursuing her master's degree in the Academy of Arts & Design, Tsinghua University.

Ke-Xin Ren received her bachelor's degree in landscape architecture from China Central Academy of Fine Art, Beijing, in 2020. She is currently pursuing her master's degree in the Academy of Arts & Design,Tsinghua University.



Hongbo Fu is a Professor with the School of Creative Media, City University of Hong Kong. He received the B.S. degree in information sciences from Peking University, and the Ph.D. degree in computer science from Hong Kong University of Science and Technology. He has served as an Associate Editor of The Visual Computer, Computers & Graphics, and Computer Graphics Forum. His primary research interests include computer graphics and human computer interaction.

Song-Hai Zhang received the PhD degree of Computer Science and Technology from Tsinghua University, Beijing, in 2007. He is currently an associate professor in the Department of Computer Science and Technology at Tsinghua University. His research interests include computer graphics and virtual reality.

SceneDirector: Supplementary Materials

Shao-Kui Zhang, Hou Tam, Yi-Ke Li, Ke-Xin Ren, Hongbo Fu, Song-Hai Zhang, *Member, IEEE*

APPENDIX A
OVERVIEW

The supplementary materials contain the following contents that strengthen our contributions:

- **Section B**: The details of the proposed dataset, including styles, statistics and examples.
- **Section C**: The details of the group transitioning from G^{t-1} to G^t corresponding to Section 3.1 of the main text.
- **Section D**: Further illustrations on the attribute smoothness in Section 3.2 of the main text.
- **Section E**: The technical details of the application "SceneForce".
- **Section F**: Engineering details of the timers for the user study in Section 5.1 of the main text.
- **Section G**: The user study platform for measuring the aesthetic and plausibility in Section 5.3 of the main text.
- A demonstration video is attached separately through the online submission system, to demonstrate the proposed method.
- Samples of our dataset are also attached in the supplementary files [1].

APPENDIX B
THE PROPOSED DATASET

In sum, our dataset contains ten styles elaborated by professional interior designers, including grey modern, gentle light, modern contrast, western pastoral, luxury, new Chinese, minimalism, Wabi-Sabi, American country style and Mediterranean style. Since this dataset is initially proposed to support the coherent group set (CGS) in the main text, so we name it **CGS-0**. Figure 1 shows the ten styles of the dataset. Compared with 3D-Front [1] where the averaged occurrence of objects is 24.865, ours is 28.515. It seems to be a minor improvement, but the standard deviation of occurrence of objects is 98.505 for 3D-Front and is 31.286 for ours. This is because, in 3D-Front, most objects are only used once or twice among all the layouts, which refers to the imbalance problem for 3D scene datasets [2]. In contrast, an object is thoroughly used in at least one entire CGS. Besides, since this dataset is created for functionally and stylistically coherent groups of objects, we classify them with leader objects representing functions and styles. This dataset will be publicly available. Designing 3D content is expensive. Due to our budget, the dataset contains 1719 layouts before the submission of this paper. We consider creating this dataset a long-term task. Though the dataset in this paper is sufficient for our method, more layouts will be created in the future. We will also make the entire annotation system open-source. As far as we investigated, we are the first to release an annotation tool for research on 3D scenes.

We first invite three professional interior designers to select a list of furniture categorized in the ten styles as shown in figure 1, with an average of 200 objects in each style. All those common and essential furniture in the living room, dining room and bedroom are provided, such as sofa, bed, nightstand, TV stand, etc. We also ensure that there are multiple choices of objects for each category in each style, e.g., the dinning table category of the gentle light style should have multiple instances (CAD models) to support it. After that, We invited more than 20 designers with interior design and art-related backgrounds to create the CGS-0.

The designers first choose a style from the provided ten styles. Then they select a topic such as living room, dining room or bedroom. Next, one dominant object expressing the topic to the greatest extent is selected with several subordinate objects from the chosen style, in order to design 10-20 different coherent group layouts. The dominant object appears exactly once in each layout while the subordinate objects can be placed repeatedly according to the needs of designers and layout strategies. The 10-20 layouts show how the scene changes gradually and variously w.r.t the contexts and constraints. The designers use the same platform as in our user study to craft the layouts. They can translate, rotate and re-scale furniture objects. After they finish designing a CGS, 10-20 layouts consisting of the same object pool with a consistent style are constructed. More evolutionary chains of CGS are shown in the supplementary files.

- *Shao-Kui Zhang and Hou Tam are with the Department of Computer Science and Technology, Tsinghua University, Beijing, China. E-mail: zhangsk18@mails.tsinghua.edu.cn, th21@mails.tsinghua.edu.cn*
- *Yi-Ke Li and Ke-Xin Ren is with the Academy of Arts & Design, Tsinghua University, Beijing, China. E-mail: lyk20@mails.tsinghua.edu.cn, rkx20@mails.tsinghua.edu.cn*
- *Prof.Hongbo Fu is with the School of Creative Media, City University of Hong Kong, Hong Kong. E-mail: hongbofu@cityu.edu.hk*
- *Song-Hai Zhang is with the Department of Computer Science and Technology, Tsinghua University, Beijing, China and Beijing National Research Center for Information Science and Technology (BNRist), Tsinghua University, Beijing, China. E-mail: shz@tsinghua.edu.cn*

Manuscript received - -, 2022; revised - -, -.

1. Due to the size limit, the demonstration video and the samples are compressed.

APPENDIX C
GROUP TRANSITIONING

This section formulates how we re-organize object from G^{t-1} to G^t. Assuming an set containing all objects is $\tilde{O}$ as shown in equation 1, which is also the set that containing the actual objects being placed to 3D scenes.

$$\tilde{O} = \{o_i = (x_i, y_i, z_i, u_i, \gamma_i)\} \tag{1}$$

The tuple (x_i, y_i, z_i) stands for the current position of o_i relative to the dominant object in 3D space, and variable u_i stands for whether o_i is already used in the scene. γ_i refers to a specific instance, e.g., a pink chair demanded by G^t. Assuming the target coherent group generated by a layout concept is $G^t = \{o'_j = (x'_j, y'_j, z'_j, \gamma_j)\}$, where each target object o'_j will be selected from $\tilde{O}$. The tuple (x'_j, y'_j, z'_j) stands for the target position of o'_j in the next transient movement. Thus, the problem is how to select a subset $\hat{O} \subset \tilde{O}$, so that the re-organization from G^{t-1} to G^t is as tidy as possible. The re-organization process is formulated in algorithm 1. For each transient time, this algorithm is executed once, so positions of objects in $\tilde{O}$ constantly change.

ALGORITHM 1: Finding the appropriate objects set for the next generated layout G^t.

Input: Entire object set $\tilde{O}$ and target layout G^t.
Output: An optimal subset $\hat{O} \subset \tilde{O}$ with the overall minimal moving distances of objects.
for *Each object* $o_i \in \tilde{O}$ **do**
  $u_i = False$;
end
$\hat{O} = \emptyset$;
for *Each* $o'_j \in G^t$ **do**
  $D = \infty$;
  for *Each* $o_i \in \tilde{O}$ **do**
    if *not* u_i *and* $\gamma_i == \gamma_j$ **then**
      $d = ||(x_i, y_i, z_i) - (x'_j, y'_j, z'_j)||_2$;
      if $d \leq D$ **then**
        $I = o_i$;
        $D = d$;
      end
    end
  end
  $\hat{O} = \hat{O} \cup \{I\}$;
  $o_i = (x'_j, y'_j, z'_j, True)$;
end

Algorithm 1 follows a greedy approach. The outer loop iterates through the entire G^t to find a best matched $o_i \in \tilde{O}$. Each o_i can be used up to once. If o_i is already occupied by a $o'_j \in G^t$, the u_i of o_i is marked as "true". The inner loop iterates through the $\tilde{O}$. The basic idea is to greedily find a nearest $o_i \in \tilde{O}$ based on the Euclidean distance. After that the position (x_i, y_i, z_i) of o_i will be set to the target position (x'_j, y'_j, z'_j) suggested by G^t. Subsequently, G^{t+1} would run algorithm 1 based on the positions set by G^t. Note that the proposed coherent group set guarantee that all target objects $o'_j \in G^t$ can always find an object from the object set $\tilde{O}$.

TABLE 1: Smoothness on the scenes in figure 2.

	2a	2b	2c	2d
2a	0	0.274	0.698	0.979
2b	0.274	0	0.833	0.735
2c	0.698	0.833	0	0.655
2d	0.978	0.735	0.655	0

APPENDIX D
SMOOTHNESS

We exploit the relative direction between the dominant object and subordinate objects. Edges are defined between each subordinate object o_i and the dominant object as unit vectors of o_i's relative direction with respect to the dominant object. The kernel between two edges is the inner product of their unit vectors and is clamped to 0 if less than 0. This metric shows how consistent the directions of two subordinate objects are with the dominant object. The distance is excluded since it might devalue the similarity metric as the room area goes larger or smaller. Table 1 is an example of smoothness on the four scenes in figure 2, where values are normalized in $[0, 1]$. First of all, a scene has no difference from itself, so values in the diagonal are all zero.

First the difference between the scene in figure 2a and the scene in figure 2b is very small, because we simply add several subordinate objects in figure 2b. However, compared with the scene in figure 2c w.r.t 2a, the layout strategy changes, so the value increases from 0.272 to 0.698. Compared with the scene in figure 2d w.r.t 2a, the layout strategies further diverge and the value is close to 1.0. From the scene in figure 2b to the scene in figure 2c, objects are not only added but also removed a bit, where we removed the three small corner-side tables. Thus, the value between them is even higher, though the two scenes in figure 2a and figure 2b are simialr. In contrast, two scenes in figure 2b and figure 2d share the similar groups of objects, so the value between them results in 0.735. Finally, the two scenes in figure 2c and 2d have a medium difference on both layout strategies and object set, so their value remains 0.655. Note that each result value of smoothness is shared by groups $G_i \in \mathcal{S}'$ derived from the same coherent group $\mathcal{S}$, to alleviate the precomputing cost.

APPENDIX E
SCENE FORCE

There have been plenty of works researching 3D hand gestures in the context of smart environments. [3] designs semaphoric gestures to trigger discrete commands in smart environments. [4] proposes 3D gestures in indoor synthesis. But gestures are only used to change the position, orientation and scale of individual objects. [5] proposes hand gestures that generate commands to directly control objects in a game. [6] builds a 3D modelling system that could manipulate and deform the meshes with hand gestures. Essentially, these works propose one-to-one gesture-command mapping without much intelligent response by environments. Additionally, only a single object is involved in each

interaction. In contrast, our application modifies the entire environment intelligently according to users' interactions, i.e., groups of objects are involved according to the proposed method. As shown in figure 10 in the main text, with users moving their hand from left to right, the positions of groups of objects change and the layout are established accordingly.

We use an HTC Vive Pro Eye HMD for eye-tracking. We use one HTC Vive controller and two HTC Vive trackers for hand-tracking. The application runs at the minimum of the HMD's framerate (90 Hz). We use equation 2 to calculate $\vec{x}$ in the virtual environment depending on hand movements in each frame and the scale ratio. The scale ratio is a positive constant to amplify the movement of objects in a scene. Δs denotes scene syntheses movement per frame. Δh denotes hand movement per frame, and a denotes axis in x and z.

$$\Delta s_a = r \times \Delta h_a \tag{2}$$

APPENDIX F
TIMERS

This section illustrates how we add timers to correctly count time in the evaluation of the main text. First, the navigating time includes users using the mouse to rotate the perspective camera by holding the left click, users using the mouse to translate the perspective camera by holding the right click, users using UP/DOWN/LEFT/RIGHT to translate the camera and users using the wheel to zoom in or zoom out. For the former two, the timer records time from starting to hold the trigger until releasing the trigger. For the latter two, since the keyboard and wheel events are discrete, the timer first caches a series of events with intervals less than t'. Subsequently, the timer records the time from the start of the series until the end of it. In this paper, we empirically set $t' = 0.2s$. Second, the research time includes users entering keywords such as "nightstand" or "pink fairy tale, users drawing sketches. The time records from their start interacting with the search panel until they click a result item. For adding objects, the time records from that a user clicks a searched item until the object is inserted. If an operation of searching or adding is interrupted, the time is recorded to "others".

As illustrated in the main text, transformations, including translation, rotation, and scale, are adjustable through three panels. The timer starts to record the time when an object is clicked. If a user uses the blue panel, the timer simply ends recording the time when the object is released. If a user uses the local coordinate system, the timer also keeps recording time, but it ends recording time if the user releases the coordinate system for more than t'. For example, a user uses the arrows representing the axes to translate objects by holding and dragging the axes. If she/he releases the axes for a while, we consider it is the end of a single interaction. If a user uses the right-top panel, the timer ends recording time after the user inputs the values. Note that if a user selects an object and does nothing, the time is recorded to "others". For removing objects, the timer records from an object being clicked until the object are removed. Our method of group editing is used through the blue panel, so the timer records the time similar to recording transformation time.

APPENDIX G
AESTHETIC AND PLAUSIBILITY

Figure 3 shows how we conduct a user study in Section 5.3 in the main text, where users could select a favored scene or select no preference, and they can jump to other questions. Subjects can also zoom in on each scene for better perceptions during the study.

REFERENCES

[1] H. Fu, B. Cai, L. Gao, L.-X. Zhang, J. Wang, C. Li, Q. Zeng, C. Sun, R. Jia, B. Zhao, and H. Zhang, "3d-front: 3d furnished rooms with layouts and semantics," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 10 933–10 942.

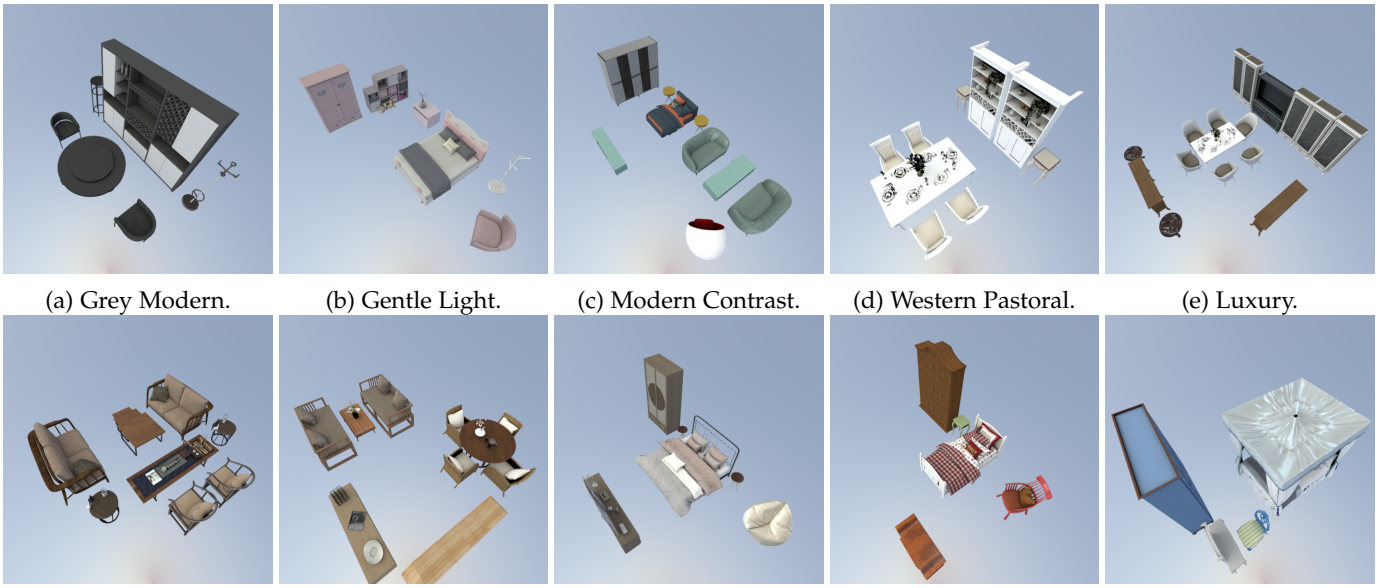
[2] S.-H. Zhang, S.-K. Zhang, W.-Y. Xie, C.-Y. Luo, Y.-L. Yang, and H. Fu, "Fast 3d indoor scene synthesis by learning spatial relation priors of objects," *IEEE Transactions on Visualization and Computer Graphics*, 2021.

[3] J. Kela, P. Korpipää, J. Mäntyjärvi, S. Kallio, G. Savino, L. Jozzo, and S. D. Marca, "Accelerometer-based gesture control for a design environment," *Personal and Ubiquitous Computing*, vol. 10, no. 5, pp. 285–299, 2006.

[4] Y. Li, X. Wang, Z. Wu, G. Li, S. Liu, and M. Zhou, "Flexible indoor scene synthesis based on multi-object particle swarm intelligence optimization and user intentions with 3d gesture," *Computers & Graphics*, vol. 93, pp. 1–12, 2020.

[5] N. H. Dardas and M. Alhaj, "Hand gesture interaction with a 3d virtual environment," *The Research Bulletin of Jordan ACM*, vol. 2, no. 3, pp. 86–94, 2011.

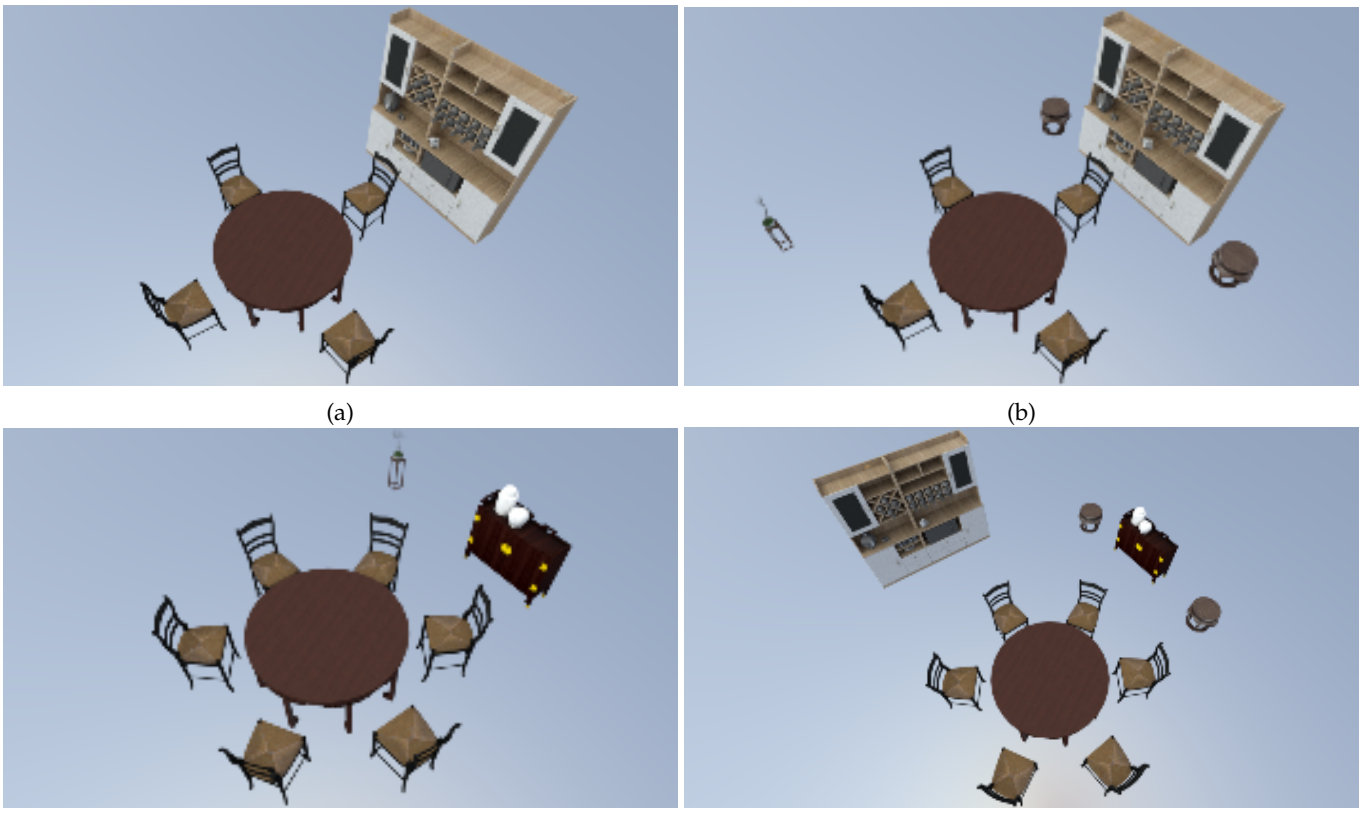
[6] H. Kim, G. Albuquerque, S. Havemann, and D. W. Fellner, "Tangible 3d: Hand gesture interaction for immersive 3d modeling." *IPT/EGVE*, vol. 2005, pp. 191–9, 2005.



(a) Grey Modern. (b) Gentle Light. (c) Modern Contrast. (d) Western Pastoral. (e) Luxury.

(f) New Chinese. (g) Minimalism. (h) Wabi-Sabi. (i) American Country. (j) Mediterranean.

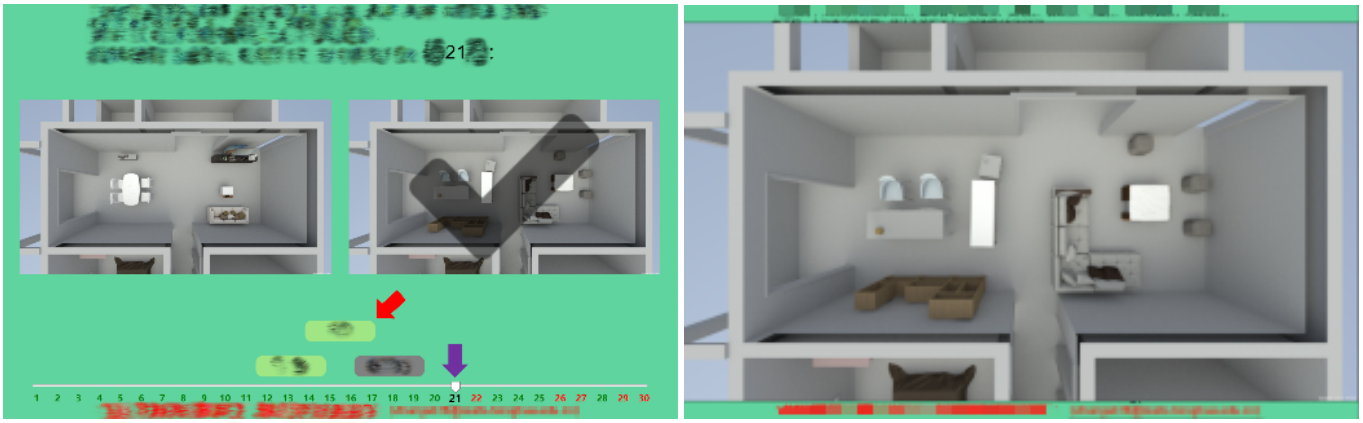
Fig. 1: Examples of the 10 styles in our datasets.



(a) (b)

(c) (d)

Fig. 2: Four example scenes for illustrating differences between scenes.



(a) (b)

Fig. 3: The user study platform for measuring the aesthetic and plausibility. 3a: users could select a favored scene or select no preference (see the red arrow), and they can also jump to other questions (see the purple arrow) if some questions require a further consideration. 3b: zooming in each scene for better perceptions.