Adaptive Synthesis of Indoor Scenes
via Activity-Associated Object Relation Graphs

QIANG FU, Beihang University
XIAOWU CHEN*, Beihang University
XIAOTIAN WANG, Beihang University
SIJIA WEN, Beihang University
BIN ZHOU, Beihang University
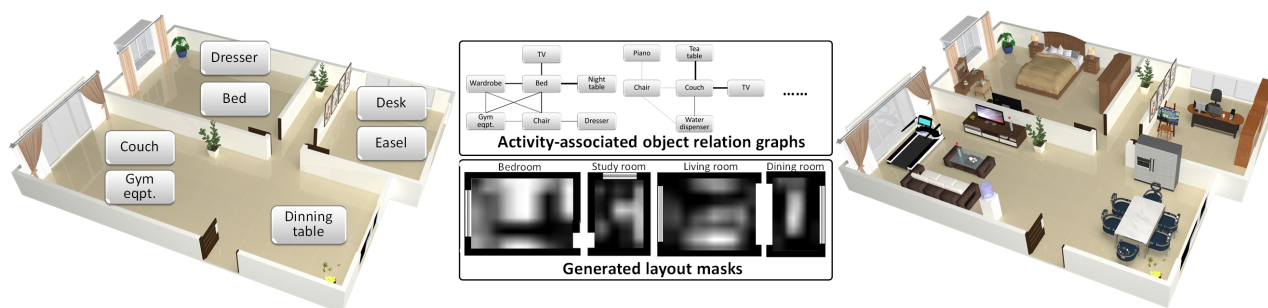HONGBO FU, City University of Hong Kong

Fig. 1. Given an empty room and only a few specified object categories (Left), our system leverages activity-associated object relation graphs learned from a 2D floor plan database to suggest object combinations and then generates their layout masks (Middle) to guide adaptive creation of 3D scenes (Right).

We present a system for adaptive synthesis of indoor scenes given an empty room and only a few object categories. Automatically suggesting indoor objects and proper layouts to convert an empty room to a 3D scene is challenging, since it requires interior design knowledge to balance the factors like space, path distance, illumination and object relations, in order to insure the functional plausibility of the synthesized scenes. We exploit a database of 2D floor plans to extract object relations and provide layout examples for scene synthesis. With the labeled human positions and directions in each plan, we detect the activity relations and compute the coexistence frequency of object pairs to construct activity-associated object relation graphs. Given the input room and user-specified object categories, our system first leverages the object relation graphs and the database floor plans to suggest more potential object categories beyond the specified ones to make resulting scenes functionally complete, and then uses the similar plan references to create the layout of synthesized scenes. We show various synthesis results to demonstrate the practicability of our system, and validate its usability via a user study. We also compare our system with the state-of-the-art furniture layout and activity-centric scene representation methods, in terms of functional plausibility and user friendliness.

CCS Concepts: • **Computing methodologies** → **Computer graphics**; Shape modeling;

Additional Key Words and Phrases: 3D scenes, adaptive scene synthesis, object relation graph

*Corresponding author: chen@buaa.edu.cn

1 INTRODUCTION

Indoor scene synthesis is an important problem in computer graphics and has received a great deal of attention lately. 3D virtual scenes that enable daily human activities are of high demand in various applications like interior design and 3D game production. Given a database of 3D scene models, to create such scenes, it generally requires solving two sub-problems: first select a proper subset of furniture objects, and then place them into a given room with a proper layout. Achieving these two steps manually is time consuming. Designing quality 3D scenes is even challenging for novice users, since this process requires interior design knowledge to fulfill the functional demands while satisfying the physical space

constraints. Such users can greatly benefit from systems that adaptively explore proper indoor objects and their layouts to synthesize 3D scenes.

The existing works on 3D scene synthesis have mainly focused on addressing either of the above two subproblems. To explore proper objects for an indoor scene, some works aim to suggest new objects based on the statistic object-object relations, with some existing furniture [Fisher and Hanrahan 2010] or a reference scene [Fisher et al. 2012]. Besides, some other works focus on the human-object relations to explore indoor objects based on certain actions (e.g., [Fisher et al. 2015; Ma et al. 2016; Savva et al. 2016]). On the other hand, some researches leverage interior design guidelines for the arrangement of *a given set of indoor objects* [Merrell et al. 2011; Yu et al. 2011], whose manual selection, however, is a tedious task especially for novice users.

Due to the lack of coordination mechanisms between the current solutions to such two subproblems, naively combining these methods might easily cause undesired results. For example, if the object exploration step returns too many furniture objects, the layout step might create a too crowded scene. Moreover, the current object exploration methods focus more on pairs of directly related objects, e.g., a PC on a desk or a human sitting on a chair and using a desk, while objects with activity relations like bookshelf-chair-desk groups are rarely concerned about. In addition, guidelines in the current layout methods only leverage partial interior design knowledge focused mainly on object-object relations, without considering function or activity related rules like anti-backlighting and path optimization and thus possibly resulting in scenes that are spatially well arranged but are not very suitable for certain activities (e.g., watching TV).

To address these problems, we present a novel indoor scene synthesis system that is able to adapt furniture exploration and layout creation with respect to a given empty room with the door and/or window specifications. To reduce the amount of user intervention, users only need to specify a small number of required object categories, with an optional number (one by default) of objects per category, and our system automatically suggests other object categories that are closely related to the existing categories. This design choice also enables our system to automatically determine the scene type of a whole or certain area of the given room, thus reducing the user intervention of selecting many scene type tags when creating large rooms with subareas of different scene types. This is achieved by exploiting the knowledge hidden in a database of 2D floor plan designed by professionals for different scenes. The scene examples in the floor plans also contain the latent activity relations of object categories. To suggest a proper set of objects, our system first determines the scene type of the input room according to the room size and the quantity of each specified object. To make the auto-completion of object categories more powerful, we introduce and use an activity-associated object relation graph for each scene, which captures not only the object-object relations used in the previous works [Liu et al. 2014; Xu et al. 2014] but also the

relations between objects that are not in close proximity but related by a certain activity (e.g., a bookshelf to a desk for reading). We also derive the layouts from the 2D floor plan database by considering the positions and sizes of windows and doors, which are seldom considered in the existing indoor scene synthesis works. The layouts of the database floor plans with the rooms of similar capacity to the input one are finally weighted and combined to guide the placement of the explored objects (see Figure 1).

We show the practicability of our system with various synthesis results. A user study was also conducted to evaluate our system compared to manual scene creation. We also compared our system with the state-of-the-art furniture layout method [Merrell et al. 2011] and activity-centric scene representation [Savva et al. 2016], in terms of functional plausibility of synthesized scenes and user friendliness of the systems.

We claim the following contributions in this work:

- An object relation graph representing objects with activity relations and coexistence frequencies, to enable adaptive object suggestion given a room and a few specified object categories.
- A 2D floor plan database that enables object relation graph construction and provides layout examples for layout creation of the synthesized scenes.
- An end-to-end system that adaptively creates 3D scenes, and support other applications like indoor scene editing and activity animation creation.

2 RELATED WORK

Contextual Scene Understanding. Contextual scene understanding has become an important technique for indoor object organization and scene synthesis. Several data-driven approaches attempted to model the context of indoor objects or their spatial relationships from existing scenes, e.g., to encode semantic scene structures [Fisher and Hanrahan 2010; Fisher et al. 2011]. Liu et al. [2014] presented an algorithm that uses a probabilistic grammar learned from examples, for hierarchical decomposition of a scene into semantic components. Xu et al. [2014] proposed to organize a heterogeneous scene collection by clustering the scenes based on a set of extracted focal points, which are used to represent substructures in a scene collection. The contextual information can also be used to turn a freehand sketch to semantically valid and well arranged 3D scenes [Xu et al. 2013], or for automatic semantic modeling of indoor scenes from a sparse set of low-quality RGB-D images [Chen et al. 2014]. The *Imagining the Unseen* system [Shao et al. 2014] was proposed to hallucinate geometry in the occluded regions of a scanned scene by globally analyzing the physical stability of the resultant object arrangements. In our work, we extract object relation graphs to represent the coexistence and activity relations of pair-wise objects, similar to the current scene structure representation methods [Liu et al. 2014; Xu et al. 2014]. However, our object

Table 1. A summary of differences between our work and the closely related works.

	[Fisher et al. 2012]	[Xu et al. 2013]	[Savva et al. 2016]	Ours
Object exploration	Scene example	Sketch	Posed agent	Few specified objects
	[Merrell et al. 2011]	[Chen et al. 2015a]	[Fisher et al. 2015]	Ours
Layout creation	Layout guidelines	Indoor image example	Placed agents	Floor plan examples
	[Merrell et al. 2011] [Chen et al. 2015a]	[Fisher et al. 2012] [Xu et al. 2013]	[Fisher et al. 2015] [Savva et al. 2016]	Ours
Relation prior	Object position/direction relation	Object co-existence relation	Object-activity relation	Activity-associated object relation

relation graphs are activity-associated and capture potential relations between objects even if they are not in close proximity or directly interacted by a posed human agent.

Activity-Centric Analysis. Activity-centric understanding of man-made objects and indoor environments has been a popular research topic in recent years. Based on the concept of affordance, several techniques have been proposed to use human-object interactions for various applications like pose estimation [Kim et al. 2014], shape analysis [Xie et al. 2014], object reshaping [Zheng et al. 2016], and scene reconstruction [Xu et al. 2015]. Understanding actions in 3D scenes has also been studied. For example, Grabner et al. [2011] proposed to learn an affordance detector and apply it to scene analysis. Savva et al. [2014] presented an action map over a scene to predict a likelihood of a given action taking place over every location in a 3D environment. Besides, Hu et al. [2015] presented an *interaction context* descriptor to explicitly represent the geometry of object-object interactions. The state-of-the-art works have been able to learn graphs encoding relations between human poses and object geometries, and represent these human-object relations in the form of *action + noun* descriptions [Savva et al. 2016], which is closely related to our work. Such descriptions could accurately determine what kind of objects should be interacted to, when an action occurs. However, such an accurate definition to describe the functional demands of a room might be tedious for users to give. In contrast our system only requires a small set of object categories to indicate the user's intension easily.

Scene Synthesis. Synthesizing indoor scenes by furniture arrangement has practical applications such as interior design and online games. Several systems like *Make it Home* [Yu et al. 2011] and the interactive furniture layout [Merrell et al. 2011] employed the pre-defined guidelines or the relations learned from a 3D scene database, to assist users in suggesting arrangements for a given set of furniture objects. Fisher et al. [2012] presented an example-based synthesis method to produce a diverse set of plausible new scenes given an existing 3D scene as input. Other systems like *Magic Decorator* [Chen et al. 2015b] and *Image2Scene* [Chen et al. 2015a] aimed to reveal the indoor scene styles like colors and layouts from indoor scene images. Besides, there exist some works on layout computation for large scale scenes like buildings and urban

faces [Peng et al. 2014; Yang et al. 2013]. Our work differs from these previous works in terms of both inputs (an empty room with user-specified object categories versus a given set of 3D furniture objects already or to be placed into a room) and methodologies (activity-associated object relations versus relative positions of object pairs).

Using human factors for scene synthesis has been getting more and more popular, since the nature of indoor scenes is to meet the demand of human activities. Jiang et al. [2012] proposed to use human context for object arrangements by learning how objects relate to human poses based on their affordances, ease of use, and reachability. Fisher et al. [2015] presented a method to generate 3D scenes that allow the same activities as real environments represented as noisy and incomplete 3D scans. Savva et al. [2016] proposed to learn a probabilistic model connecting human poses and arrangements of object geometry, for jointly generating 3D scenes and interaction poses. Ma et al. [2016] provided a framework for *action-driven evolution* of indoor scenes focusing on the object placement variations caused by a sequence of actions. Our system focuses on furniture-level layout of indoor scenes with the objects and their layouts adaptively determined. In our system, different sizes of input room or different quantities of a certain object category can lead to variations of synthesized scenes due to the adaptive scene type determination and object exploration. Besides, the 2D floor plan database in our system provides layout examples following interior design knowledge and thus being able to guide object arrangement for the given room and explored objects.

In Table 1, we have summarized the differences of our system from closely related works in terms of the conditions for object exploration, layout creation, and the required relation priors. Compared with the works based on activity relations [Fisher et al. 2015; Savva et al. 2016], our work performs the object-level inference for scene synthesis, instead of requiring user manipulation on human agents. Compared with the works based on object co-existence relations [Fisher et al. 2012; Xu et al. 2013], our work uses object co-existence frequency to enhance the object relations which are associated by meaningful activities. Compared with the works focusing object arrangement [Chen et al. 2015a; Merrell et al. 2011], our work can also implicitly describe the potential function/activity of
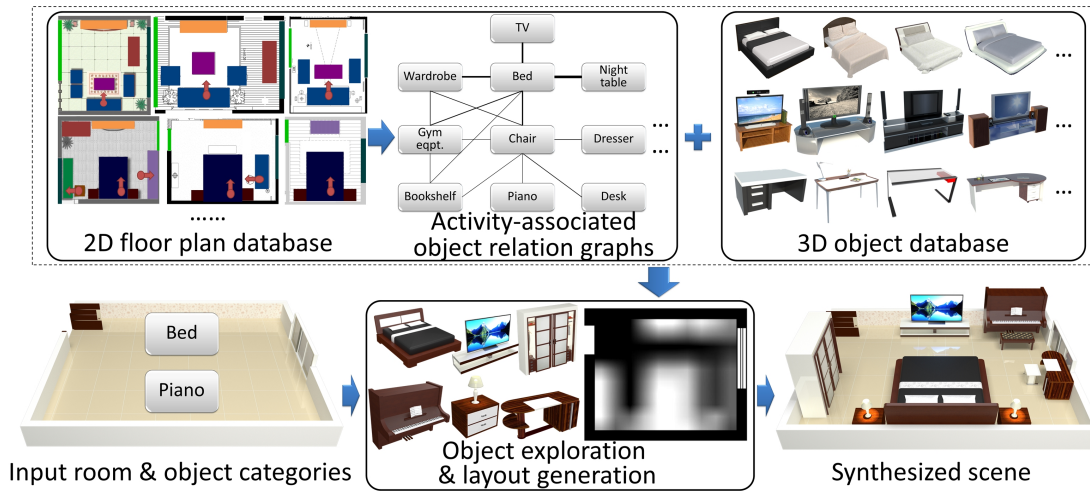
Fig. 2. Pipeline of our system. We use a 2D floor plan database to generate activity-associated object relation graphs for each type of scene. Given an empty room and a few user-specified object categories, our system employs the object relation graphs to explore more proper object categories and generate their layouts to guide scene synthesis.

the synthesized scenes thus to facilitate implementations like activity animation creation. Given these differences, our system requires only a very sparse specification of major object categories, and can automatically create scenes with enriched objects adapted to the given room.

3 OVERVIEW

As illustrated in Figure 2, our system consists of two stages: an off-line stage for extracting object relation graphs from a 2D floor plan database for each type of scene; an on-line stage for using the extracted object relation graphs to explore proper objects and similar floor plan examples to guide the synthesis of 3D scenes.

In the first stage, we collect a 2D floor plan database with layouts designed by professionals, including eight types of scenes. To extract the activity-related prior of indoor objects behind these floor plans, we first segment and label all the indoor objects in each plan. As a cue to reveal the interaction relations between objects, we also label the human positions and directions inside floor plans. For some objects such as wardrobe and chair, human-object relationship is rather fixed and can be determined after object annotation. However, for other objects like sofa, there might be multiple potential human positions, and their positions/directions thus might change across floor plans, depending on surrounding interactive objects. Hence we place some human agents with proper positions and front directions to ensure that every existing object in the floor plans can be interacted with a certain placed human agent (Section 4.1). We also have a 3D indoor object database for scene synthesis, with eighteen categories of furniture-level objects. With such a floor plan database, we count the statistics on the coexistence frequency of each pair of object categories, and also detect the activity relations of object pairs using the placed human agents in each floor

plan. Afterwards, we construct object relation graphs with the object categories as the nodes and their relations as the edges with different weights to describe their relation degrees (Section 4.2). Besides, we also extract the room feature and layout of each 2D database floor plan for the subsequent adaptive scene synthesis (Section 5.1).

In the second stage we use our 2D floor plan database, 3D object database and object relation graphs to generate scenes adapted to the given empty room and user-specified object categories, with one (by default) or multiple objects per category. Our system first determines the scene type of the given room based on the input room size and the specified object categories in terms of their quantity (Section 5.2). Then, an object exploration algorithm is employed to adaptively suggest more proper object categories beyond the user-specified ones via the object relation graphs and the floor plan database, to make the input room have more closely related objects and a reasonable rate of occupancy. Finally, we use the room feature of the input room to find several similar floor plan examples. The layouts of these examples are weighted and combined to create the layout of the input room, and all the explored objects are then placed into the room to generate the synthesized scene (Section 5.3). Our system naturally enables scene synthesis and editing. For example, the objects that are added into an existing room can be automatically re-arranged. We also show the object relation graphs can benefit animation creation of indoor activities by seeking for interactive related neighbors in the graph starting from a certain object (Section 6).

4 ACTIVITY-ASSOCIATED OBJECT RELATION GRAPHS

Activity plays an important role in revealing human-object relations. Our goal is to discover activity-associated objects
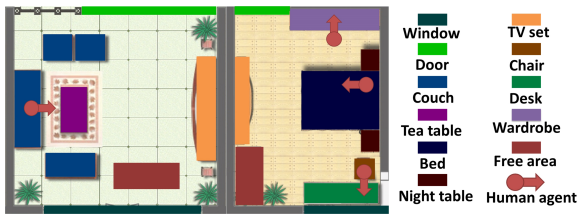
Fig. 3. In each room of our 2D floor plan database, we manually label the furniture objects, potential human positions (red dots) and their front directions (red arrows), and the free areas that can be potentially used for object placement.

and encode their relations as a graph. Each type of scene has its own object relation graph as a result of the functionality of the scene. Such a representation does not need to define any noun of action or activity, as done in [Savva et al. 2016]. Instead, we deem that activities are the latent relations embodied in the layouts of 2D floor plans. In this manner, given a certain object category, we can find its activity-related categories from the achieved relation graph to enrich the scene variations.

4.1 2D Floor Plan Database

We collected eight scene types of professionally designed floor plans to establish our 2D database, including bedrooms, living rooms, dining rooms, study rooms, offices, conference rooms, gyms and dining halls. These floor plans were collected from the websites of the house-builders and interior design companies. We denote these scene types as $\mathcal{S} = \{s_1, \cdots, s_8\}$. These floor plans have the associated scale information and well designed furniture layouts following interior design knowledge and fully considering the affordance of each room. We leverage these commonly seen home and public scenes to provide potentially related object groups and layout references in our system. To this end, as illustrated in Figure 3, we first manually labeled the categories, positions and sizes of the indoor objects, windows and doors in each plan. Next we manually placed human agents with proper positions and front directions into each floor plan. These agents are important for detecting activity-related objects. Since most of the floor plans only contain the basic furniture but we would also like to consider daily appliances like water dispenser and refrigerator, we invited a professional designer to assist in labeling some free areas (Figure 3) where some appliances might be placed without influencing the use of existing furniture in each floor plan.

4.2 Relation Extraction

Objects labeled in the 2D floor plans belong to some basic furniture categories. To enrich the variations of the synthesized scenes, we also collected a 3D object database (18 object categories in our system), denoted as $\mathcal{O} = \{o_1, \cdots, o_{18}\}$. It contains more object categories than the existing ones in the 2D database floor plans, and objects of these categories can be placed in certain labeled free areas. For each scene type,

Table 2. Some highlighted parameters.

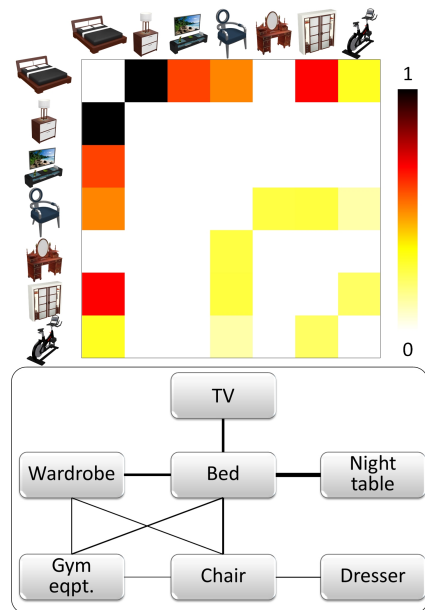|	Set	AVG. area	Graph weight		Feature	Area
Scene type	$\mathcal{S}$	θ_s	W_s	Scene	$\mathbf{f}_r$	θ_r
Object category	$\mathcal{O}$	θ_c	$\searrow$	Object	$\mathbf{f}_{obj}$	θ_o

Fig. 4. Part of an object relation graph (partial nodes) extracted from the type of bedroom floor plans. We show the pair-wise object relation weight matrix $\{W_s(o_i, o_j)\}$ (top) and the graph which only contains the edges whose weight is greater than 0 (bottom). Note the edges with greater weight are heavier.

a fully connected graph can be constructed, with each node representing one object category in $\mathcal{O}$. Our goal in this step is to determine the relations between each object category pair o_i and o_j as the weight of their edge $E(o_i, o_j)$. We take into account both the coexistence frequency and activity priors to describe such object relations. In Table 2, we highlight some parameters of scene type and object category as well as the scene and object, which will be used in the subsequent discussions.

Specifically, if two categories of objects o_i and o_j are often coexistent and simultaneously used by a human, or have the matched functional purposes, they would have a large edge weight. To compute the activity priors, denoted as a *binary* matrix $A(o_i, o_j)$, we use the placed human agents in each floor plan to first detect the interactive objects. For example, a lying agent in a bed has a touch area (i.e., within the arm's reach) covering a night table and a visual area (i.e., all regions in front of the agent) covering a TV set, $A(o_i, o_j) = 1$ for the bed-night table pair and bed-TV set

pair. Besides, we also define two kinds of *functional purposes*: to stay in a certain object for a long time (e.g., beds, chairs, couches, gym and play equipments); to fetch something from an object with a short stay (e.g., bookshelves, wardrobes, water dispensers, and refrigerators). We observe that objects with the first functional purpose can be related to switch one activity to the other one, while objects with the second functional purpose can be related to objects with the first purpose so as to fulfill the activity after fetching something (e.g., a bed or a chair can be related to a bookshelf for reading activity). In this way, $A(o_i, o_j) = 1$ if o_i and o_j both have the first functional purpose or one of them has the first purpose while the other has the second.

However, the object relations obtained by the functional purposes might not be always correct. For instance, according to the functional purposes alone, a refrigerator is related to a bed, but there is no natural activity associated with such an object pair. Fortunately, the 2D floor plan database can be used to address this problem by counting the objects' *coexistence frequency*. This is because activities are usually done in the same room. For example, a refrigerator is more likely to be related to a chair aside a dining table in a dinning room rather than a bed in a bedroom. Benefited from the labeled objects in each floor plan, the coexistence frequency is easy to calculate for most of the objects, except the free areas which are labeled to place certain appliances like piano and easel, which are not contained in the floor plans. We compare the sizes of the labeled free areas with the sizes of those appliances to give the existence probability of such appliance categories in each floor plan. Let o' denote a certain category of such appliances. We use the averaged short and long edge lengths $l_s^{o'}$ and $l_l^{o'}$ of floor projections of all the objects in this category to define the object size feature $\mathbf{f}_{obj}^{o'} = \{l_s^{o'}, l_l^{o'}\}$. Similarly, for a floor plan r with a free area we calculate its size feature $\mathbf{f}_{obj}$ and define $exist(o', r) = ||\mathbf{f}_{obj}^{o'} - \mathbf{f}_{obj}||_2^2$ to describe the existence probability of object category o' in r. Note that we also filter out the irrational existence relations in a certain scene type, e.g., we set all $exist(o', r) = 0$ for refrigerator in each floor plan of the conference room. For the other objects which appear in the floor plan database, we set $exist(o, r) = 1$ if object category o is labeled in a floor plan r and $exist(o, r) = 0$ otherwise. In this way, we obtain the coexistence frequency $P(o_i, o_j|s)$ of scene type s as follows:

$$P(o_i, o_j|s) = \frac{\sum_{r \in \nabla} exist(o_i, r) \cdot exist(o_j, r)}{|\nabla|}, \quad (1)$$

where $\nabla = \{r\}$ is a set of floor plans in scene type s. Then, the weight of their edge $E(o_i, o_j)$ for scene type s, denoted as $W_s(o_i, o_j)$, is the product of the coexistence frequency $P(o_i, o_j|s)$ and activity priors $A(o_i, o_j)$, i.e., $W_s(o_i, o_j) = P(o_i, o_j|s) \cdot A(o_i, o_j)$. For the fully connected graph with all object categories as the nodes, we cut the edges whose $W_s(o_i, o_j) = 0$ and remove the isolated nodes to obtain an object relation graph. See Figure 4 for an example of the
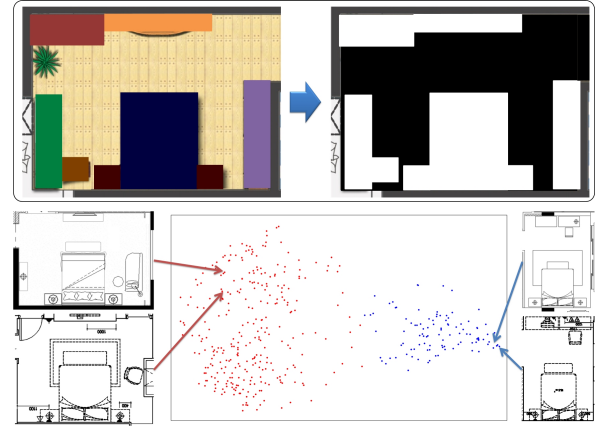

Fig. 5. **Top**: we use the labels in the floor plan to generate a layout mask. **Bottom**: 2D embedding of the layouts of the bedroom floor plans. We cluster them into two groups and show two cases in each group.

weight matrix and corresponding relation graph. The activity-associated relations contain both the direct interaction relations and the activity-associated contexts. For example, we can find a direct interaction relation between chair and dresser that can be interactive with a static human, and an activity-associated context between wardrobe and chair-dresser indicating a dynamic process of dressing-up activity from our relation graph.

5 ADAPTIVE SCENE SYNTHESIS

Similar to the recent furniture layout methods [Chen et al. 2015a; Merrell et al. 2011; Yu et al. 2011], our system focuses on furniture-level object arrangement. Since the layouts in our 2D floor plan database were designed by professionals, the design of such layouts has already followed not only arrangement guidelines but also the reasonable rate of occupancy and the activity-related priors such as anti-backlight when placing TV set, optimizing the indoor pathway, etc. This motivates us to leverage the 2D floor database to adaptively explore proper objects and their layouts for scene synthesis, using the activity-associated object relation graphs. To achieve this, our system first determines the scene type of a given room, then explores more proper object categories besides the user-specified object categories to enrich the scene, and finally uses the layout examples extracted from the 2D floor database to seek for the proper objects for the suggested object categories, and guide the arrangement of the explored objects to synthesize 3D scenes. This process is adaptive to the given room and user-specified objects. In other words, if the user already specifies enough object categories leading a proper ratio of occupancy for the given room, our system will not suggest more objects but focuses on generating a proper layout to place these specified objects.
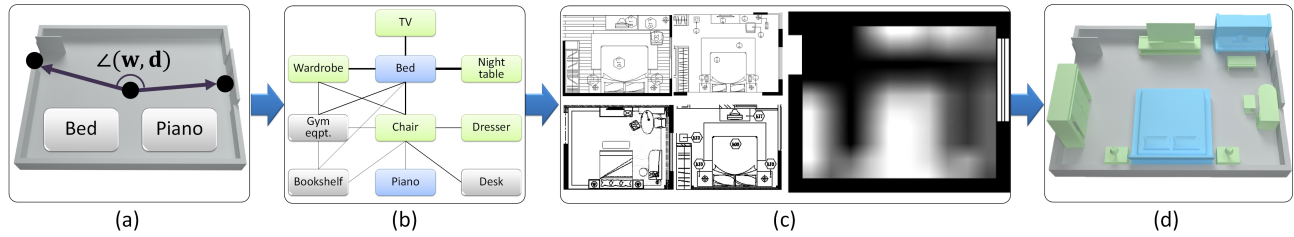
Fig. 6. Adaptive scene synthesis. (a) the input room and specified object categories. (b) based on the object relation graph, more proper object categories (green) which are related to the specified ones (blue) are explored. (c) a proper layout mask is generated by weighted amalgamating the database floor plans with the similar room features. We show the top-4 floor plans. Note that there might not exist a database floor plan that has the same room and layout as the generated result. (d) objects are placed into the room guided by the layout to synthesize a 3D scene.

5.1 Room and Layout Description

We observe that the layout of the scene depends at least on the room size, positions of windows and doors, and the areas of all furniture. For simplicity, let us assume all floor plans in our database are rectangular, with $\sum \theta_o$ and θ_r as the areas of all placed objects and the room, respectively. To describe the similarity of two rooms, we thus define a room feature $\mathbf{f}_r = \{\angle(\mathbf{w}, \mathbf{d}), r_s, r_e, \varphi\}$, where $\angle(\mathbf{w}, \mathbf{d})$ is the included angle between two vectors $\mathbf{w}$ and $\mathbf{d}$, which are from the home center to the average positions of the windows and doors, respectively (as shown in Figure 6 (a)), $r_s = \frac{\sum \theta_o}{\theta_r}$ is the ratio of occupancy, r_e is the ratio of the short edge to the long edge of a room, and $\varphi = \frac{\theta_r}{\theta_r + \theta_s}$ is a scale factor, where θ_s is the average area of all rooms in the same scene type to make the scale factor φ can describe the relative size of a room in its scene type.

The 2D floor plans also provide plausible layout examples for different rooms. For each database floor plan in the same type, we obtain a binary mask in which the areas placed with furniture are set to 1, while the other areas are set to 0 (see Figure 5 (top)). Then we align all these masks in the same type of scenes by the long edges of the rooms, and make the barycenter of the room to be in the bottom-right, by horizontally/vertically flipping the masks if needed. After that, each aligned mask is resized to a 10×10 matrix, denoted as M, such that layouts from different floor plans can be amalgamated to a new layout. We use a matrix with small size to ensure that the floor plan layouts can be amalgamated thoroughly, rather than a larger one that only generates a layout with all objects from the reference floor plans overlapped in a room. Besides, though the database floor plans have different sizes, we only use few floor plans with similar room features (including room size) to generate the layout of the given room. Thus it allows the masks of the database floor plans to be resized to the same size. Even a same room can have more than one plausible layout, and we thus also cluster these layout matrices in each type of scenes into groups (via *MeanShift* clustering) to reveal the layout similarity. In the bottom of Figure 5, we cluster the layout of all bedrooms in our floor plan database into two groups, and show some floor

plan examples in each group. Layout matrices in the same group are easier to amalgamate due to their similarities.

5.2 Scene Type Determination

Since different types of scenes have different layout distributions and object relations, we need to first determine which type of scene is more suitable for the input room. The input room can be a single scene or a composite scene consisting of multiple pieces. The output of this step is thus either a single scene type or a combination of multiple types. It depends on the size of input room and the quantity of the user-specified objects to determine which scene type(s) should the input room to be. Assume $\tilde{\mathcal{O}} \subset \mathcal{O}$ is a set of user-specified object categories and θ_s^i is the average room area of scene type s_i, calculated by using the 2D database floor plans in the same type. For an input room whose area is θ_r, we are to explore a set of scene types $\tilde{\mathcal{S}} = \{s_i\}$, such that the sum of the area of the explored scene types should be close to θ_r, and the explored scene types should be ready to accommodate all specified objects. Thus we determine the scene type(s) as follows:

$$
\arg\min_{\tilde{\mathcal{S}}} \frac{|\theta_r - \sum_{\tilde{\mathcal{S}}} \theta_s^i|}{\theta_r},
$$

$$
s.t. \max_{s_i \in \tilde{\mathcal{S}}} P(o_j|s_i) > 0 \text{ and } \sum_{s_i \in \tilde{\mathcal{S}}} N(o_j, s_i) \geq n_{o_j}, \forall o_j \in \tilde{\mathcal{O}}, \quad (2)
$$

where $P(o_j|s_i)$ is the occurrence frequency of object category o_j in scene type s_i, $N(o_j, s_i)$ is the maximum number of object o_j in any scene of type s_i, n_{o_j} is the user-specified quantity of o_j ($n_{o_j} = 1$ if it is not explicitly specified). To solve an approximate solution of this equation, we first use all scene types as $\tilde{\mathcal{S}}$ and remove the scene type s_i which has the largest average area one by one, until the sum of the areas of the remaining scene types is approximate to the given room. When removing a scene type will lead the constraints to be no longer satisfied, such a scene type will be preserved in $\tilde{\mathcal{S}}$.

We leverage Equation (2) to suggest a proper combination with one or multiple scene types for the given room (see examples in (Figure 11)). If the input room can be pieced together with multiple scenes (e.g., a large room divided into a bedroom and a study room), we use the $\max_{s_i \in \tilde{\mathcal{S}}} P(o_j|s_i)$ to specify each object category to the scene part whose type has

the greatest existence probability for such an object category. For example, given two scene types s_1 and s_2 being explored for the object categories $\tilde{\mathcal{O}}$, if $P(o|s_1) > P(o|s_2)$ for an object category $o \in \tilde{\mathcal{O}}$, then o belongs to the scene type s_1 when synthesizing the scene (Section 5.3). Such large rooms are then divided into sub-rooms, one for each single scene type, by adding virtual boundaries in the room to make the sub-rooms have the similar proportional relation to the average areas of their scene types. Hence the scene synthesis task is respectively tackled in each sub-room with a single scene type. In this way, given different sizes of input rooms or different quantities of specified objects, our system adaptively uses the layout examples from one or multiple types of scenes and suggests different categories of object to create 3D scenes, as shown in Section 7.

5.3 Object Exploration and Arrangement

To make our system easy to use, users only need to specify a small number of object categories. The quantity of the object in each specified category is set to 1 by default, and can be specified by the user. This gives a coarse description of the user's demands about the target scenes. We explore more proper object categories using the activity-associated object relation graphs. The key idea here is to ensure that the explored objects have close relations among themselves and to the specified ones, and all the explored objects can make the input room to have a reasonable rate of occupancy similar to some of the floor plan examples. Namely, if all potential object categories are used for synthesis, the scene might be too crowded. But by removing the weakly related object categories, the sum of object areas of the remaining categories decreases approaching to a desired rate of occupancy, which can be calculated by the floor plan examples with their rooms similar to the input one. Hence, we employ the following object exploration algorithm to achieve this goal:

ALGORITHM 1: Object Exploration

Input: Specified object category set $\tilde{\mathcal{O}}$, object relation graph $G_s = (\mathcal{O}, W_s)$, input room area θ_r
Output: Explored object categories $\mathcal{O}_{out}$
$\mathcal{O}_{out} = \mathcal{O}$;
$Cost = C(\mathcal{O}, \theta_r, W_s)$;
while $|\mathcal{O}| > |\tilde{\mathcal{O}}|$ **do**
    Remove object category $o \in \mathcal{O}$ and $o \notin \tilde{\mathcal{O}}$ with the minimum $E(W_s, o)$ from $\mathcal{O}$;
    Remove object categories $\{o_i\}$ from $\mathcal{O}$ which are isolated in graph G_s after o is removed;
    Update the edge weight W_s after graph G_s is changed;
    $NewCost = C(\mathcal{O}, \theta_r, W_s)$;
    if $NewCost < Cost$ **then**
        $\mathcal{O}_{out} = \mathcal{O}$;
        $Cost = NewCost$;
    end
end
return $\mathcal{O}_{out}$;

In this algorithm, $E(W_s, o_i)$ indicates the average weight of an object category o_i to its neighbors in the relation graph of scene type s used for removing the weakly related object category from the graph in each iteration, and $C(\mathcal{O}, \theta_r, W_s)$ is a cost function aiming at finding category group with proper rate of capacity and close relation. Assume category o_i has J neighbors in the group, we only compare top-K ($K = 10$ in our implementation) database floor plans who have the similar room features as the given room. $E(W_s, o_i)$ and $C(\mathcal{O}, \theta_r, W_s)$ are defined as follows:

$$E(W_s, o_i) = \frac{1}{J} \sum_{j \in \{1, \cdots, J\}} W_s(o_i, o_j),$$

$$C(\mathcal{O}, \theta_r, W_s) = \frac{1}{K} \sum_{k \in \{1, \cdots, K\}} \left| R(\mathcal{O}, \theta_r) - r_s^k \right| \qquad (3)$$

$$- \omega \cdot \sum_{\mathcal{O}} W_s(o_i, o_j),$$

where $\{o_j\}$ is the set of neighbors of o_i in the object relation graph, $R(\mathcal{O}, \theta_r) = \frac{\sum \theta_c}{\theta_r}$ is the ratio of capacity of the input room and r_s^k is the ratio of capacity of the k-th database floor plan which has a similar room feature as the input room by calculating their Euclidean distance with the uniform weighting for individual components, and we set the weight $\omega = 0.02$. Note that θ_c is the average area of the determined object categories, and if the user specifies multiple objects in the same category as input, we sum up all their projection areas for θ_c when calculating the ratio of occupancy $R(\mathcal{O}, \theta_r)$.

In this manner, we obtain a group of object categories that could be placed into the input room. As illustrated in Figure 6, these object categories will be arranged by the layouts extracted from the 2D floor database. We intend to seek for the floor plans with the similar room features to the input room and use their layout matrices to synthesize a new layout. By using the average area for each category of the suggested objects, we can calculate the room feature $\mathbf{f}_r$ of the input room. Then, top-K floor plans with the minimum Euclidean distances of the room features to $\mathbf{f}_r$ are explored as the references. These floor plan references have the similar rooms to the input one in terms of the room size, positions of windows and doors, and the ratios of capacity. The layout matrices of the floor plans in the same clustered group are amalgamated in a weighted manner as follows:

$$M = \frac{D(\mathbf{f}_r, \mathbf{f}_r^{\tilde{k}})}{\sum_{\tilde{k}} D(\mathbf{f}_r, \mathbf{f}_r^{\tilde{k}})} M_{\tilde{k}}, \qquad (4)$$

where $D(\mathbf{f}_r, \mathbf{f}_r^{\tilde{k}})$ is the feature distance between $\mathbf{f}_r$ and the feature of its $\tilde{k}$-th floor plan with the similar room whose layout matrix is $M_{\tilde{k}}$. Note that $\{\tilde{k}\} \subset \{1, \cdots, K\}$ is a set of floor plans whose members belong to the same clustered group in the top-K similar database examples. We use the set with most members if the top-K floor plans are clustered into different group to get the layout matrix M, and then resize it to the same size as the input room, as the layout mask for object arrangement. In this mask, the regions with the large
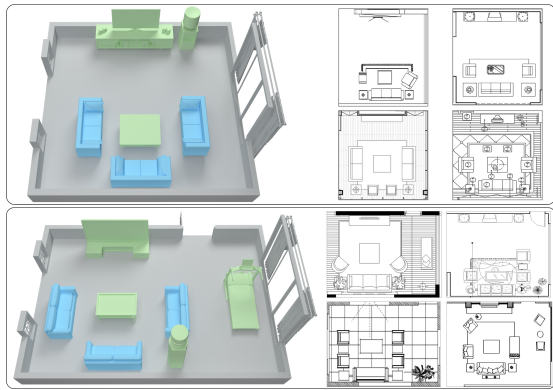
Fig. 7. Two synthesized scenes and some of their associated references. Our system explores objects (green) and creates layouts influenced by the reference floor plans. Note the two rooms have different configurations but are given the same category of object (blue).

values are more likely to be placed with furniture. Note that the reference floor plans influence both the object exploration and arrangement stages, to create scenes adapted to the given rooms. For an example in Figure 7, two rooms with different configurations (left) are suggested to have different floor plan references (right). The references in the top case only have the basic components of living room, while the configurations of the bottom ones provide the extra space to allow more objects. These differences actually lead to variations of the synthesized results, more specifically, in terms of different quantities of objects that a room can accommodate, and different layouts to arrange these objects. Since both the object exploration (Algorithm 1) and layout creation (Equation 4) rely on the floor plan references, the different reference floor plans thus lead to the different synthesized scenes. Hence, although the user specifies the same category of object (i.e., three couches), our system actually explores a gym equipment for the bottom case and places the couch-TV group slightly off centre compared to the top case.

Object arrangement. With the generated layout mask, we place the explored object categories into the room in turn. We choose the average length of the sides of the floor projection bounding box for each object category to represent a regular object in this category. Note we focus on the large furniture as bed, wardrobe, desk, etc, while their accessory objects like night table, TV, chair, etc, are placed after the associated large furniture arranged, and follow a fixed relative relations and directions as they are in the 2D database floor plans. The quantities of the accessory objects also rely on the associated large furniture and their sizes, e.g., large dinning tables or conference tables always have multiple chairs. The proportional relations of the quantities of the accessory objects and the size of their associated large furniture are determined by the examples in the 2D floor plan database. The sizes and relative positions of some common furniture combinations in the floor plans (e.g., three or four couches around a tea-table), are also extracted as the priors for object arrangement.
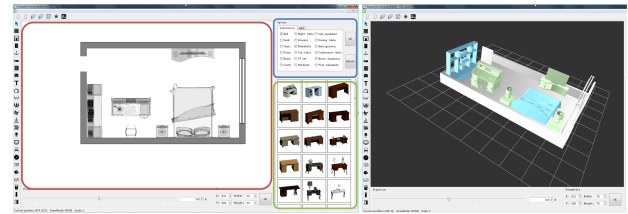

Fig. 8. The user interface of our system with the display panel (in red box), options panel (in blue box) and the suggestion list for object replacement (in green box). The created scene can be displayed as 2D floor plan (Left) or 3D scene (Right).

Benefited from the generated layout mask, the large furniture are placed in the mask with the positions and directions covering the region with maximal value. The furniture are placed one by one based on their sizes from the largest to the smallest to ensure that small furniture will not be placed in the region which is more suitable for the bigger ones. Each time an object is placed, we cover this region in the layout mask to 0 in this region to avoid object conflict. After all explored object categories are placed, we make a suggestion list of the associated 3D models in the database based on the covering regions of their floor projections in the mask.

6 USER INTERFACE

As illustrated in Figure 8, the user interface of our system provides two main panels: a display panel to assist the user to set up an empty room, and show the 2D floor plan or synthesized 3D scene; an options panel to specify object categories as the input demands or for adding new objects. Since our system is able to determine the scene type(s) of the given room, which might be divided into sub-rooms with different scene types, specifying object categories and their quantities as the system inputs can make the scene types of the given room be adaptive. For each object in the synthesized scene, we also provide a list of suggested objects for potential replacement to make scene variations (see the accompanying video for a live demo).

Scene synthesis. With our interface, the user first specifies a few object categories as constraints, and builds an empty room by controlling the room size, the positions and sizes of doors and windows. Afterwards, our system automatically explores more object categories related to the specified ones and creates a proper layout to place all the furniture into the given room. Since the synthesized scenes of our system are adaptive to the input room size and object categories, the same room with different specified object categories or the same specified objects in different sizes of rooms can lead to different synthesized scenes (see experimental results in Section 7).

Scene editing. Our system also allows the user to edit existing scenes, e.g., by adding new categories of objects or reusing existing layouts (Figure 9 (top)). Given a room with already arranged objects, our system provides a list of object categories for the user to choose new object categories to be
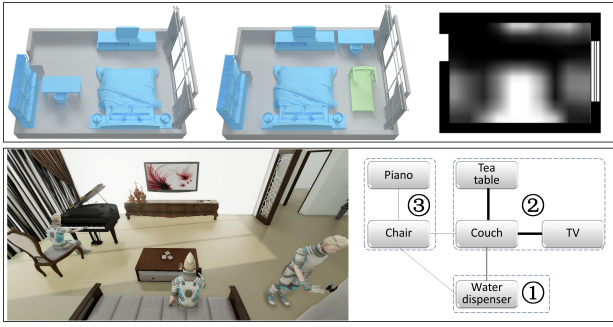
Fig. 9. **Top**: adding a new object (in green) into an existing scene (Left). Note the original layout has been adjusted following the generated layout mask (Right). **Bottom**: activity animation creation based on our activity-associated object relation graph (right). The human agents wander through three places to do different activities. This animation is shown in the accompanying video.

added into the room. With the ratio of occupancy changed due to the chosen objects, our system can re-generate new layouts by Equation (4) to place these objects into the room.

Activity animation creation. Using our activity-associated object relation graph to represent a scene with various furniture can also indicate the potential activities that can occur in the synthesized scene. Hence our system can generate activity scripts to drive a human agent to wander around all these objects according to the edge weights of the object pairs in the graph. As illustrated in Figure 9 (bottom), starting from any furniture which has the functional purpose for staying or fetching something, animations are performed with its related objects in the group for touching or watching, or getting something from the objects. Then the agent wanders to the next object which has the largest weight of the edge with the previous object in the object relation graph. Thus, all the potential activities about the existing objects can be performed, resulting in an activity animation. See the accompanying video for such animation.

7 RESULTS AND EVALUATION

In this section we first present various scene synthesis results generated by our system, and show how the synthesized scenes adapt to the inputs. Then we evaluate our system with a user study where our results are compared to manually created scenes in terms of both functional plausibility and visual naturalness. We also compare our method to the closely related works [Merrell et al. 2011; Savva et al. 2016] to demonstrate the practicability of our system in layout creation and activity representation.

Our 2D floor plan database consists of four types of home scenes including living rooms (195), bedrooms (419), study rooms (47), dining rooms (183), and another four types of public scenes including offices (36), conference rooms (23), gym (36), dining hall (115). These floor plans provide activity-based object relations and plausible layouts for our adaptive scene synthesis system. We also collect 18 categories of 3D

models including chairs (67), beds (87), couches (59), TV sets (59), desks (58), night tables (56), dining tables (54), dressers (53), tea tables (55), conference tables (51), pianos (40), easels (13), bookshelves (57), refrigerators (12), wardrobes (13), water dispensers (16), gym equipments (13) and play equipments like billiard tables and play tables (22) to enrich the variety of synthesized results. On average our system needs less than one minute to interactively set up an empty room with door(s) and window(s), and about 2 seconds to automatically generate a 3D scene, tested on a PC with Intel Core i7-4790 3.60GHz CPU with 16GB RAM.

Synthesis results. In Figure 10, we present 6 synthesized scenes to show how the scene results adapt to different specified object categories. Each column has a pair of scenes generated by the same empty rooms but differently category constraints. In Figure 11, we show 4 large scenes which are pieced together by multiple parts in different scene types. For each synthesized scene, the input empty room comes with doors, windows and some manually created ornaments like plants, and a few specified object categories, while our system adaptively explores more indoor objects and places them into the room, making the synthesized scenes visually pleasing and functionally plausible. In the top of Figure 12, we show that the synthesized results are adaptive to the room size. For each scene, a small room (Left) and a large one (Right) are given with the same object category constraints. Compare to the left example, the right case shows a large room that encouraged by our system to be pieced by multiple parts, i.e., a study room left and a bedroom right. Figure 12 (bottom) shows the adaptive scene synthesis by different quantities of a certain specified object category: different quantities of the specified gym equipments lead to a living room or a gym by our system given the same room.

User study. One of the important characteristics of our system is that the explored objects and layouts of the synthesized scenes are appropriate for indoor activities. A key evaluation of our system is thus to test whether the generated scenes are functionally plausible in terms of the objects and layouts. Like previous works [Fisher et al. 2015; Ma et al. 2016], we also leave such subjective judgments to human subjects. To prepare the user study, we collected 5 floor plans of different scene types (namely, bedroom, living room, office, conference room and dining hall). These new floor plans were again professionally designed but not included in our original 2D floor plan database (Section 4.1). We then used the room size and partial existing object categories to create scenes with our system. For further comparisons, we also invited 5 novice users with no interior design knowledge and asked them to manually select object categories, and place the objects into each room with the same inputs as our system. We choose a same object for each category in all those scenes to reduce the influence of furniture style on the subjective judgments.

Afterwards, the floor plans of the scenes generated by 3 different ways (in total, 5 sets of 3 scenes; see one set of scenes in Figure 13 (Top)) were blindly given to 20 professional participants engaged in interior design to evaluate the

☑ Couch × 3

☑ Desk
☑ Piano

☑ Desk × 2
☑ Tea table

☑ Bed

☑ Dining table

☑ Conference table

Fig. 10. Gallery of synthesized scenes given an empty room and one or multiple object categories.

☑ Couch × 3   ☑ Piano   ☑ Play eqpt.

Living room   Gym

☑ Bed   ☑ Gym eqpt.   ☑ Desk

Bedroom   Study room

☑ Conference table
☑ Desk

Conference room   Study room

☑ Dinning table × 3   ☑ Play eqpt. × 4

Dining hall   Gym
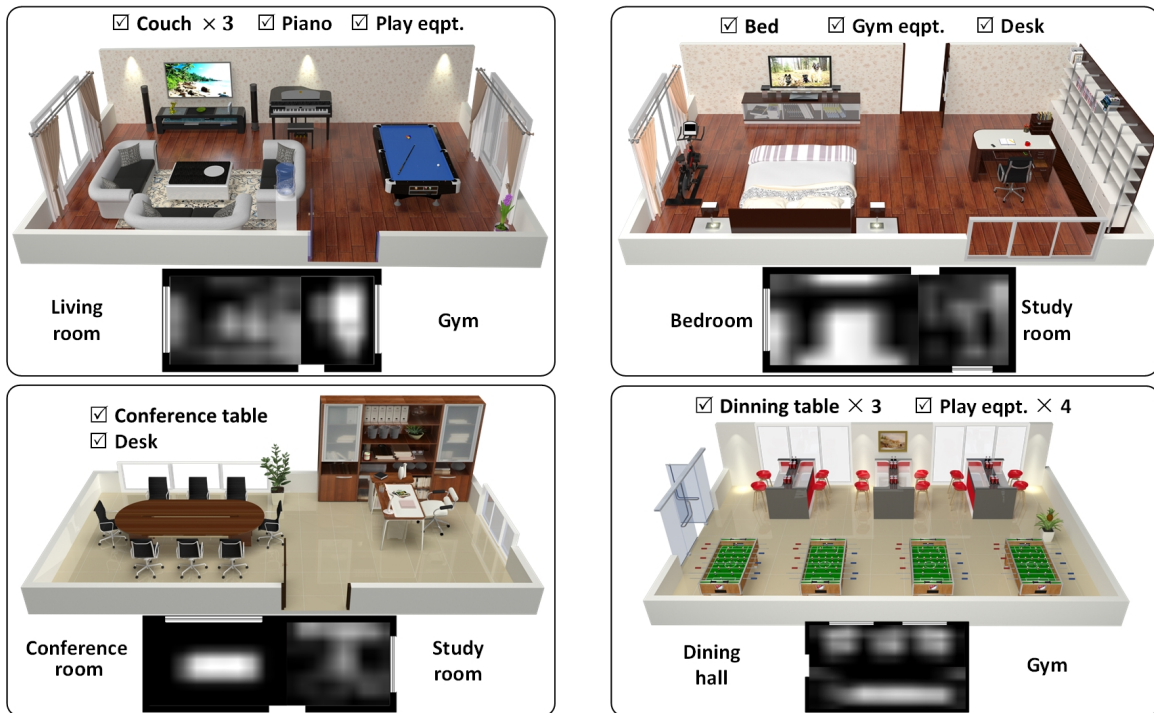
Fig. 11. Large scenes generated with the given room divided into multiple sub-rooms. We show the generated layout mask in the bottom of each case.

plausibility of both the layouts and the functionality. Each participant was asked to give a score for each scene in the discrete scale of 10 (best) to 0 (worst) based on the above two criterions. The statistics on the evaluation results are plotted in Figure 13 (Bottom). The scores of the novice users' design are low, mainly because that even though the novice users could choose proper objects for the scenes, the lack of interior design knowledge led their designs to low-quality

Fig. 12. **Top**: given the same specified object categories, the synthesized scene is adaptive to the size of the input room. **Bottom**: given the same room, the synthesized scene is also adaptive to the quantity of a certain specified object category.
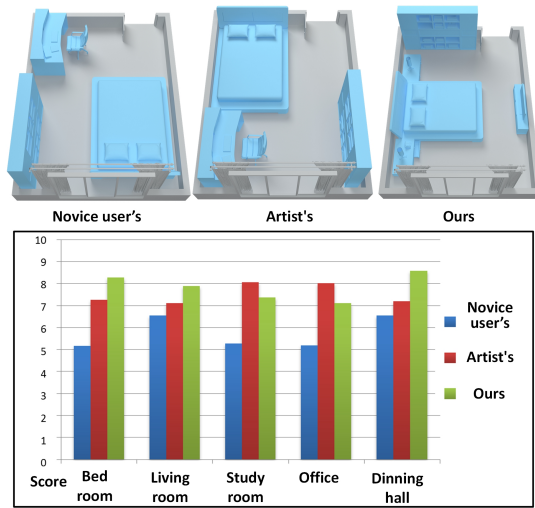

Fig. 13. Statistics of our user study. **Top**: the bedroom case with "bed" and "bookshelf" categories specified in our user study test data. **Bottom**: the average evaluation scores for the scenes created by novice users, professionals, and our method in terms of the functional plausibility and layout plausibility.

layouts. On the other hand, guided by the layouts in our 2D floor plan database, our results achieved similar or sometimes even better quality results compared to the scenes manually designed by professionals.

Comparisons to closely related works. In Figure 14 we show the furniture layout results (Left) suggested by the guidelines used [Merrell et al. 2011], and the synthesis results by our system (with the same furniture (Middle) and with our own suggested objects (Right)). In the two comparison cases, although the objects are well arranged using the furniture

layout guidelines, some drawbacks might influence certain activities in the scenes. For example, the position of the TV set in the top-left case is in a backlight status with the TV set facing the window, while this problem does not exist with our results (Middle and Right), since the reference layouts in our plan database have already implicitly considered anti-backlighting. In the bottom-left case, a pre-placed TV set plays a role as the emphasis to force the couches to face it, while such a layout increases the distance of the way from the door to sit on the couch (Left). Our work does not need such a pre-placed emphasis as constraint. Benefited from the layouts in the floor plan database, our system provides a better solution with a shorter path and enabling anti-backlighting for watching TV as well. Another significant advantage of our approach is automatic suggestion for proper objects, which have to be manually specified as input for [Merrell et al. 2011].

Our system also has the advantage over the current activity-centric scene synthesis methods in activity representation. In Figure 15, we compare the graphs of our system and the state-of-the-art interaction representation method *PiGraphs* [Savva et al. 2016], and visualize these graphs which represent the activities in the same room (Left). We can see that our activity-aware relation graph is a single graph connected all objects of the scene, while PiGraphs provides two isolated graphs representing certain groups of objects. It takes more efforts to represent a scene by their graph which is based on a posed single human, especially for scenes with objects where human does not stay for a long time, e.g., bookshelf, wardrobe, etc. For example, in this case, *PiGraphs* is able to represent the activities of "sit-bed+wathcing-TV", and "sit-chair+read-book". However, activities which involve dynamic processes like first getting a book from the bookshelf and then going to bed or chair to read the book can hardly be represented. In contrast, our activity-associated object relation graph can represent both the activities performed by a posed human,
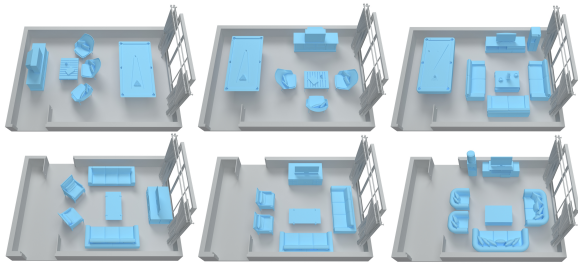
Fig. 14. Comparison with [Merrell et al. 2011] (Left). Our synthesis results with the same input objects as their (Middle) and our suggested objects (Right).
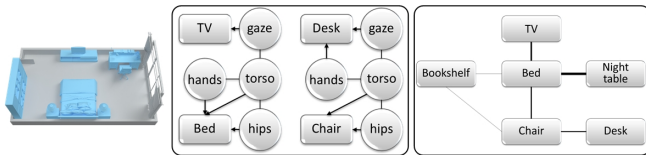

Fig. 15. For an indoor scene (Left), we show the descriptions of [Savva et al. 2016] (Middle) and ours (Right), which could be used for representing and guiding the synthesis of such a scene.

and the dynamic activities that need the human to walk from one place to the other one, facilitating the animation creation discussed in Section 6.

8 DISCUSSION, LIMITATION AND FUTURE WORK

We presented a novel technique for adaptive synthesis of indoor scenes. Compared with the guideline-based or activity-centric scene synthesis methods, our system is able to adaptively suggest the proper objects and their layouts to create 3D scenes, ensuring the functional plausibility of the scenes especially for the dynamic activities. Benefited from the object relation graphs and layouts extracted from a 2D floor plan database, our system has several applications like indoor scene synthesis, editing and even activity animation creation. We have validated the usability of our system via various scene creation results, subjective user study and comparisons to closely related works.

Discussions. Our current system uses the user-specified object categories as the constraints for scene creation. Some higher level prescription such as activity tag can be used as the alternative input. Since we have extracted the activity-associated object relation graphs, we can label certain relations with semantic activity tags and provide them as the options in the system UI. For example, if we label the relations of object pairs like bookshelf-bed, chair-desk, etc. with a "reading" tag, we can then extract some sub-graphs from the relation graph with certain objects associated with the reading activity. In these sub-graphs, some objects like bookshelf are necessary, while the others like desk or bed are optional. Once the user chooses the reading tag as input, it is equivalent to choosing the bookshelf as the user-specified object category, and our system can still explore the other
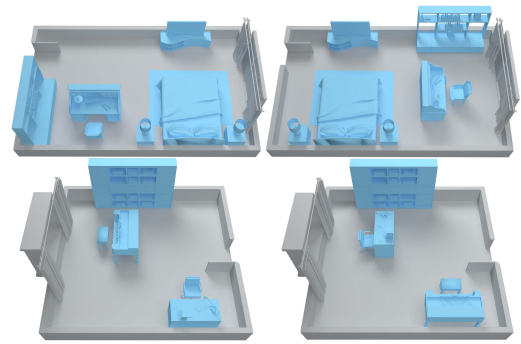

Fig. 16. **Top**: Two scenes with different layouts might lead to different subjective feelings when using desk or bed for reading. **Bottom**: the placing order based on the size of each object leads a piano to occupy the place which is more proper for the desk (Left), a slight user assistance is required to change their orders to get a more plausible result (Right).

associated objects adapted to the given room. Besides, the object exploration stage in our system can also be employed in the interactive scene synthesis frameworks (e.g., [Merrell et al. 2011]): once the user chooses an object, our algorithm automatically explores the related ones to accelerate the modeling process. Moreover, considering the color style of room and object might improve the quality and efficiency of scene creation. For example, using the color style metric from the existing works like *Magic Decorator* [Chen et al. 2015b] can enable our system to explore objects with consistent color styles from 3D database, thus constraining the object colors in the synthesized scene to the same color theme of the given room.

Limitations. Although our system often produces quality results given empty rooms and certain object demands, it still has several technical limitations. Since our system focuses on furniture-level object arrangement, the positions of some small objects like dishes and cups are not considered. In our 3D furniture database, such objects are attached to furniture objects like dining tables and tea tables with fixed positions. Hence our system can provide the initial scene input to existing methods focusing on the arrangement of such small objects (e.g., *Scene Evolution* [Ma et al. 2016]) to obtain more interesting variations closer to the natural indoor scenes. Besides, since our object exploration component relies on the ratios of occupancy of the database floor plans, a large input room will be suggested to have many explored indoor objects. If the user prefers a more empty room with fewer objects or does not like some of the explored objects, a slight user assistance is required to remove undesired object categories. Moreover, If a user specified too many object categories given a small room, our system will hold all the specified objects and lead the synthesized scenes to be too crowded.

Another limitation exists in the layout generation stage. When a large room is suggested to be divided into multiple sub-rooms with different scene types, their relative positions are not determined. In our current system, we always make

the largest room type in the middle of the room or nearest to the window. However, different ways to divide the room can certainly lead to different effects on human activities. For an example in Figure 16 (Top), a room has been divided into a study room and a bedroom with two different relative positions, and these rooms are respectively synthesized and pieced together to generate the whole scenes. The left result has a better lighting environment for the bed, which is closer to the window than the desk, to make the "reading" activity more appropriate to occur in the bed, while the right one is on the contrary. Since the impact of the environment on activities is very subjective, we need users to choose their favorite result when such scenarios happen. Besides, since the layouts of the sub-rooms are respectively created, when piecing them together to generate a whole scene, some objects (e.g., a bookshelf or wardrobe) might be placed in the border between two sub-rooms. Since some of them should better be in the places against the wall, these objects will be snapped against to the wall on the other side of the sub-room. Moreover, the placing order (from large to small) might not always be appropriate. In some cases, user assistance is required to change the placing order for a better result (Figure 16 (Bottom)). Lastly, our system has mainly focused on the rectangular rooms since the non-rectangular floor plans are less common and might not be suitable for layout transfer. However, our system allows users to set up some independent rectangular rooms and combine them to form a T- or L-type scene.

Future work. In the future, we plan to further consider living habits of human users to enable our system to produce more personalized generations. More kinds of data like videos of indoor human activities would also be further considered to achieve more information to enhance the ability of system to create more activity-associated 3D scenes. A final interesting direction to extend our system is to employ the natural language processing techniques to create accurate 3D scenes with text descriptions.

9 ACKNOWLEDGEMENTS

REFERENCES

Kang Chen, Yu-Kun Lai, Yu-Xin Wu, Ralph Martin, and Shi-Min Hu. 2014. Automatic Semantic Modeling of Indoor Scenes from Low-quality RGB-D Data Using Contextual Information. *ACM Transactions on Graphics* 33, 6 (2014), 208:1–208:12.

Kang Chen, Kun Xu, Yizhou Yu, Tian-Yi Wang, and Shi-Min Hu. 2015b. Magic Decorator: Automatic Material Suggestion for Indoor Digital Scenes. *ACM Transactions on Graphics* 34, 6 (2015), 232:1–232:11.

Xiaowu Chen, Jianwei Li, Qing Li, Bo Gao, Dongqing Zou, and Qinping Zhao. 2015a. Image2Scene: Transforming Style of 3D Room. In *Proceedings of the 23rd Annual ACM Conference on Multimedia Conference.* ACM, 321–330.

Matthew Fisher and Pat Hanrahan. 2010. Context-based Search for 3D Models. *ACM Transactions on Graphics* 29, 6 (2010), 182:1–182:10.

Matthew Fisher, Daniel Ritchie, Manolis Savva, Thomas Funkhouser, and Pat Hanrahan. 2012. Example-based Synthesis of 3D Object Arrangements. *ACM Transactions on Graphics* 31, 6 (2012), 135:1–135:11.

Matthew Fisher, Manolis Savva, and Pat Hanrahan. 2011. Characterizing Structural Relationships in Scenes Using Graph Kernels. *ACM Transactions on Graphics* 30, 4 (2011), 34:1–34:12.

Matthew Fisher, Manolis Savva, Yangyan Li, Pat Hanrahan, and Matthias Nießner. 2015. Activity-centric Scene Synthesis for Functional 3D Scene Modeling. *ACM Transactions on Graphics* 34, 6 (2015).

H. Grabner, J. Gall, and L. Van Gool. 2011. What makes a chair a chair?. In *IEEE Conference on Computer Vision and Pattern Recognition.* 1529–1536.

Ruizhen Hu, Chenyang Zhu, Oliver van Kaick, Ligang Liu, Ariel Shamir, and Hao Zhang. 2015. Interaction Context (ICON): Towards a Geometric Functionality Descriptor. *ACM Transactions on Graphics* 34, 4 (2015), 83:1–83:12.

Yun Jiang, Marcus Lim, and Ashutosh Saxena. 2012. Learning Object Arrangements in 3D Scenes using Human Context. In *International Conference on Machine Learning.*

Vladimir G. Kim, Siddhartha Chaudhuri, Leonidas Guibas, and Thomas Funkhouser. 2014. Shape2Pose: Human-centric Shape Analysis. *ACM Transactions on Graphics* 33, 4 (2014), 120:1–120:12.

Tianqiang Liu, Siddhartha Chaudhuri, Vladimir G. Kim, Qixing Huang, Niloy J. Mitra, and Thomas Funkhouser. 2014. Creating Consistent Scene Graphs Using a Probabilistic Grammar. *ACM Transactions on Graphics* 33, 6 (2014), 211:1–211:12.

Rui Ma, Honghua Li, Changqing Zou, Zicheng Liao, Xin Tong, and Hao Zhang. 2016. Action-Driven 3D Indoor Scene Evolution. *ACM Transactions on Graphics* 35, 6 (2016), 173:1–173:13.

Paul Merrell, Eric Schkufza, Zeyang Li, Maneesh Agrawala, and Vladlen Koltun. 2011. Interactive Furniture Layout Using Interior Design Guidelines. *ACM Transactions on Graphics* 30, 4 (2011), 87.

Chi-Han Peng, Yong-Liang Yang, and Peter Wonka. 2014. Computing Layouts with Deformable Templates. *ACM Transactions on Graphics* 33, 4 (2014), 99:1–99:11.

Manolis Savva, Angel X. Chang, Pat Hanrahan, Matthew Fisher, and Matthias Nießner. 2014. SceneGrok: Inferring Action Maps in 3D Environments. *ACM Transactions on Graphics* 33, 6 (2014), 212:1–212:10.

Manolis Savva, Angel X. Chang, Pat Hanrahan, Matthew Fisher, and Matthias Nießner. 2016. PiGraphs: Learning Interaction Snapshots from Observations. *ACM Transactions on Graphics* (2016), 5:1–5:2.

Tianjia Shao, Aron Monszpart, Youyi Zheng, Bongjin Koo, Weiwei Xu, Kun Zhou, and Niloy J. Mitra. 2014. Imagining the Unseen: Stability-based Cuboid Arrangements for Scene Understanding. *ACM Transactions on Graphics* 33, 6 (2014), 209:1–209:11.

Zhige Xie, Yueshan Xiong, and Kai Xu. 2014. AB3D: action-based 3D descriptor for shape analysis. *The Visual Computer* 30, 6-8 (2014), 591–601.

Kun Xu, Kang Chen, Hongbo Fu, Wei-Lun Sun, and Shi-Min Hu. 2013. Sketch2Scene: Sketch-based Co-retrieval and Co-placement of 3D Models. *ACM Transactions on Graphics* 32, 4 (2013), 123:1–123:15.

Kai Xu, Hui Huang, Yifei Shi, Hao Li, Pinxin Long, Jianong Caichen, Wei Sun, and Baoquan Chen. 2015. Autoscanning for Coupled Scene Reconstruction and Proactive Object Analysis. *ACM Transactions on Graphics* 34, 6 (2015), 177:1–177:14.

Kai Xu, Rui Ma, Hao Zhang, Chenyang Zhu, Ariel Shamir, Daniel Cohen-Or, and Hui Huang. 2014. Organizing Heterogeneous Scene Collections Through Contextual Focal Points. *ACM Transactions on Graphics* 33, 4 (2014), 35:1–35:12.

Yong-Liang Yang, Jun Wang, Etienne Vouga, and Peter Wonka. 2013. Urban Pattern: Layout Design by Hierarchical Domain Splitting. *ACM Transactions on Graphics* 32, 6 (2013), 181:1–181:12.

Lap-Fai Yu, Sai Kit Yeung, Chi-Keung Tang, Demetri Terzopoulos, Tony F. Chan, and Stanley Osher. 2011. Make it home: automatic optimization of furniture arrangement. *ACM Transactions on Graphics* 30, 4 (2011), 86.

Y. Zheng, H. Liu, J. Dorsey, and N. Mitra. 2016. Ergonomics-inspired Reshaping and Exploration of Collections of Models. *IEEE Transactions on Visualization and Computer Graphics* 22, 6 (2016), 1732–1744.