Computers & Graphics 46 (2015) 231–243

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

SMI 2014

Wall grid structure for interior scene synthesis

Wenzhuo Xu [a,b], Bin Wang [a,b,*], Dong-Ming Yan [c,d]

[a] School of Software, Tsinghua University, Beijing 100084, PR China
[b] Tsinghua National Laboratory for Information Science and Technology, Beijing 100084, PR China
[c] Visual Computing Center, King Abdullah University of Science and Technology, Thuwal 23955-6900, Saudi Arabia
[d] NLPR-LIAMA, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, PR China

ARTICLE INFO

Article history:
Received 8 July 2014
Received in revised form
24 September 2014
Accepted 27 September 2014
Available online 6 October 2014

Keywords:
Wall grid structure
Interior scene synthesis
Front extraction

ABSTRACT

We present a system for automatically synthesizing a diverse set of semantically valid, and well-arranged 3D interior scenes for a given empty room shape. Unlike existing work on layout synthesis, that typically knows potentially needed 3D models and optimizes their location through cost functions, our technique performs the retrieval and placement of 3D models by discovering the relationships between the room space and the models' categories. This is enabled by a new analytical structure, called *Wall Grid Structure*, which jointly considers the categories and locations of 3D models. Our technique greatly reduces the amount of user intervention and provides users with suggestions and inspirations. We demonstrate the applicability of our approach on three types of scenarios: conference rooms, living rooms and bedrooms.

1. Introduction

Efficient tools for the interior design of 3D indoor scenes remain in high demand in many applications. Since users may have only a rough idea about their desired design, it is hard for them to draw sketches or search for desired examples. Designers cannot come up with a pleasant solution without fully understanding users' demands. A promising solution to this problem is to provide users with diverse examples and discuss requirements based on these examples. As the building structures differ from one to another, it is a complicated and time-consuming task to build such examples. Examples on the internet (e.g., Google warehouse) may contain several drawbacks: (1) the number of plausible and well-arranged interior scenes is small. (2) Some of them have the problem of content missing. (3) Some even violate the principles of interior design. Thus, a tool which can generate a diverse set of plausible interior scenes for a certain room structure is desired.

Several criteria should be met for such an intuitive design tool. First, as space enclosed by building elements (floors, walls, ceilings, roofs, etc.) is essential raw materials of interior design [1], the design tool should take building structures and available space into consideration. Second, it should generate plausible scenes that look reasonable to a casual observer. Third, the generated scenes should be diverse enough to cover users' demands. Fourth, the amount of user intervention should be reduced to meet the requirements for efficient communication.

In this paper, we present a framework for interior scene synthesis that meets the above criteria. We encode the probability of occurrence of shapes on the surfaces of the scenes (as opposed to volumes) and this is sufficient in many cases to produce good layouts. A novel structure, called *Wall Grid Structure* (WGS) is used for our purpose. The WGS is an analytical structure which consists of a set of grid cells generated from the input room shape. The WGS can effectively characterize the arrangement information using probabilities by analyzing the relationships between grid cells and models. Our system consists of two stages: a learning stage and a synthesis stage. In the learning stage, our system establishes WGS to capture the arrangement information from a small database collected from the internet. In the synthesis stage, given the user-specified room shape (including windows, doors, vertical walls, floors, ceilings, etc.), our approach uses the WGS of the room shape to find out available space and generates a diverse set of well-arranged interior scenes.

We demonstrate the utility of our system by synthesizing a variety of interior scenes from the given room structures. The main contributions can be summarized as follows:

- A novel structure, WGS, that captures the structure of a room shape and learns the arrangement and symmetry information of the room.
- An interior scene synthesis method based on WGS, that generates plausible interior scenes conforming to the principles of interior design.
- A context-based algorithm that automatically extracts the front directions of models.

* Corresponding author.

http://dx.doi.org/10.1016/j.cag.2014.09.032
0097-8493/

- Several meaningful requirements for the assessment of interior scenes: functional requirements, illumination requirements, aesthetic requirements and greening requirements.

2. Related work

Our work is closely related to component-based modeling, indoor scene analysis and synthesis. In this section, we briefly review these approaches and discuss how they relate to our system.

Component-base modeling: In recent years, many techniques have been proposed for synthesizing new models from existing components. Kalogerakis et al. [2] and Chaudhuri et al. [3] use probabilistic graphical models to encode the cardinality, style, and the adjacency information of shape parts, respectively. The new models are then generated by recombining the sub-components based on probabilistic reasoning. Other geometric data structures such as hierarchical structures [4] and symmetry functional arrangements (SFARR) [5] are also used to guide the recombination of shape parts. Our work is inspired by these approaches, that give us the intuition to treat 3D interior scenes as combinations of 3D models and to synthesize new scenes through recombination. However, the sub-components of objects are often functionally related and have connectivity information, which can rarely be found in 3D interior scenes. Moreover, the structures, i.e., slots [2] and contacts [4] used for object modeling are not appropriate for interior scenes, because the relationships and constraints differ greatly from models to scenes. The reader is referred to the recent survey by Mitra et al. [6] for more details of component-based modeling.

Indoor scene analysis and synthesis: Indoor scene analysis has attracted more attention recently. The concurrent work of Nan et al. [7], Shao et al [8] and Kim et al. [9] focuses on detecting and recognizing repeated indoor objects from laser scanned data or RGBD images. Once the contents of scenes are detected, they can be easily used for further manipulation and reorganization [10].

Furniture layout synthesis aims to compute a feasible arrangement for a given set of models inside a room. Some specific design guidelines are used to define energy functions that evaluate the "goodness" of the layouts. Since the solution space is always very large, stochastic optimization methods are used for finding solutions [11,12]. While these approaches can generate plausible interior scenes, the set of furniture objects have to be specified beforehand.

Gaussian mixture models (GMM) are used to learn the pairwise spatial placement [13,14]. However, in interior scenes, many models are loosely related and some are even related by symmetry which can be hard to describe using GMMs. Fisher and coauthors use graph structures to characterize 3D scenes for comparison [15] and context-based searching [16]. But the lack of the arrangement information makes it difficult for scene synthesis. Bayesian networks together with arrangement models and GMM [14] are used to synthesize scenes. However, it is hard to learn a reliable Bayesian network from imperfect input datasets like ours. Factor graphs [17–19] are widely used to characterize relationships using constraints. But the prior knowledge of models and their potential locations are required. For example, a desk put in the center of the room and the same desk placed against a wall have different constraints with wall model. Reshuffle based method [20] generates new scenes by replacing models in the original scene according to some relationships, while in our case these relationships cannot totally characterize our arrangement information.

Sketch-based modeling is an alternative way for indoor scene synthesis. Eitz et al. [21] propose to use sketches for model retrieval. While Xu et al. [13] present an interactive modeling system by co-placement of indoor models from user-input sketches. However, these systems are not suitable for users without specific training. In contrast, our work aims at providing an easy-to-use tool for a larger range of users.

Pioneer work uses the artificial intelligence algorithms for layout generation and visualization, e.g., space reduction [22], genetic algorithms [23] and meta heuristics [24]. These works do not learn the existing database, and they need to describe the scene with some specific declarative rules. Then a solver is used to solve the constraint-based system. While our approach generates new scenes simply using a function evaluation based on the wall grid structure.

3. System overview

Our system consists of two main stages, a learning stage and a synthesis stage. In the learning stage, we learn the WGS and functional groups which contain arrangement information and group information from a database of existing interior scenes. In the synthesis stage, a user specifies a room shape, which is

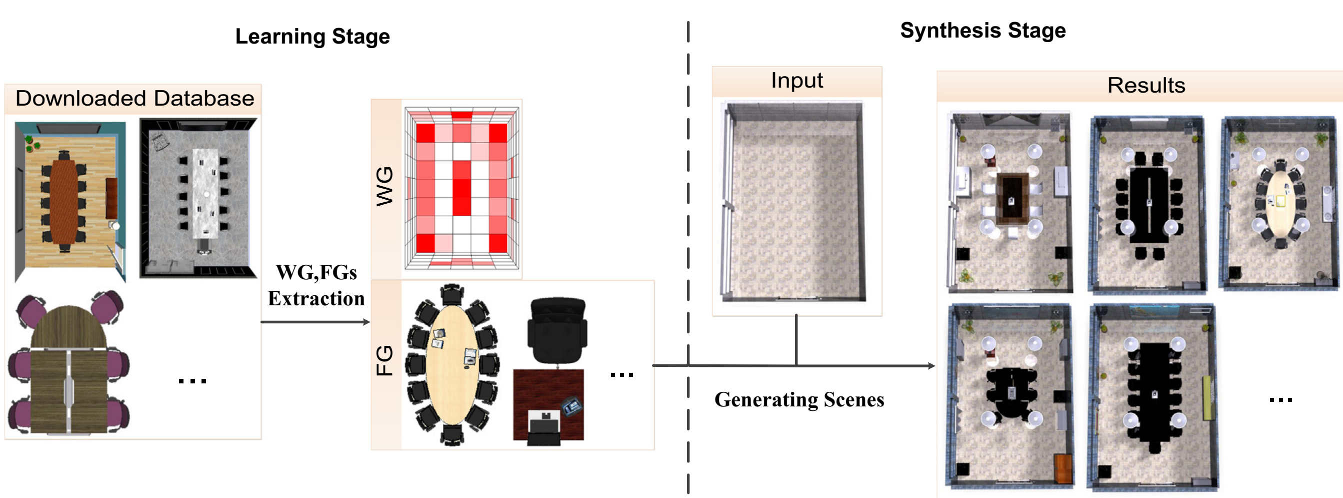

Fig. 1. Algorithm pipeline. In the learning stage, our technique pre-processes a small set of 3D interior scenes to extract functional groups and corresponding wall grid structure with arrangement information. In the synthesis stage, our system takes a room shape as an input and synthesize a diverse set of well-arranged interior scenes based on the learned information.

typically a cuboid-shaped 3D room model, together with its type, numbers and locations of windows and doors. Doors relate to pathways in the room and windows have relationship with illumination. It should be guaranteed that doors and windows are not overlapped by furniture objects. Our algorithm then automatically analyzes available space and generates a diverse set of plausible and well-arranged interior scenes. The pipeline of our framework is shown in Fig. 1.

The learning stage requires a repository of 3D interior scenes as the training data. We separate our input data into three distinct databases: conference rooms, bedrooms, and living rooms containing 52 scenes, 45 scenes, and 48 scenes respectively. Each database is composed of complete interior scenes with room structures and scenes with a set of arranged models. All scenes are pre-segmented into semantic objects with labels. Several relationships have been specified beforehand. Thus, it is possible to algorithmically infer the front direction of models using the contextual information in scenes (Section 4.1) and extract functionally related model groups, named as *functional groups* (Section 4.2). We use functional groups instead of models to synthesize interior scenes. We demonstrate later that our concept of functional groups is different from structure groups [13] and is more suitable for our purpose. To capture the arrangement information of functional groups, we introduce the WGS and show that this structure can help us find relationships between functional groups and their locations quickly and effectively (Section 4.3).

In the synthesis stage, given a room shape as input, we first establish the corresponding WGS and analyze its occupancy information (Section 6.1). Next, we use the learnt arrangement information and functional groups to generate a diverse set of well-arranged scenes (Section 6.2). In order to make the results plausible and aesthetic, we propose a few criteria to filter out the bad results. Note that our input room does not contain any guidance of the models in the room, which is different from previous approaches [13,14,11].

4. Wall grid structure

Placing models according to the relationships between room space and arranged models is the right way to synthesize an interior scene. However, directly placing models according to the relationships may result in two problems. First, the models placed in the right locations may not be placed in correct orientations. Second, the functions of the placed models could be incomplete.

WGS is a new analytical tool for learning the relationships between room space and arranged models. It breaks the surfaces of a room into grid cells, and encodes in each grid cell the probability of FGs appearing at that location. To build the WGS we use three steps. First, we propose an algorithm to extract from interior scenes the arrangement information of individual models, which contains a front direction and back-against-wall information. Second, to ensure the integrity of functions, a new concept of functional groups (FG for short), which characterizes the functional relationships among models, is introduced. The orientations of FGs are inferred from context information and orientations of models. Third, our WGS analyzes the arrangement and symmetry information of FGs and uses probability values to measure the compactness between room space and FGs. By synthesizing interior scenes using FGs instead of models, our technique not only greatly simplifies evaluations for function integrity but also reduces the computation for arranging models in certain patterns.

4.1. Context-based front extraction

The orientations of 3D models are essential information in scene synthesis and layout optimization. The way to extract up-direction has been explored by Fu et al. [25] using machine learning methods. However, the extraction of front direction is even harder than up direction. There is no clue as effective as supporting faces. Most of the scene synthesis methods have the front direction pre-specified, which is a time consuming task. In our case, models have a default up direction, as interior scenes downloaded from google warehouse have the up direction in line with y-axis. It can also be quickly calculated by analyzing the supporting frame of the room shape. Having the context information, our algorithm automatically extracts the front directions of models.

Our method is based on several observations. First, models with their backs against walls are most likely to have the front direction perpendicular to the wall and face to the room center. Second, model groups which are functionally related are likely to face to the group center for functional purpose, such as conversation and entertainment. Third, the front direction is more likely to be perpendicular to faces with larger area which show more details. We calculate the oriented bounding box of models and limit the candidate front directions to the normals of four faces (excluding the up and down faces). According to the positions and relationships of the models, we extract the front directions as follows.

Models against wall: We say a model is against wall if the distance between its center and a wall is less than the radius of its bounding sphere. As shown in Fig. 2(a), we consider the candidate front directions $g_{i,i=1,2,3,4}$ of a sofa, the vector f_c from the center of the sofa to the center of the room and the normals of walls f_w. Models against walls are likely to face to the center of the room and have their backs against walls. Thus, if g_i is the front direction, the angle $\alpha_{g_i w}$ between g_i and f_w and the angle θ_i between g_i and f_c should be in the range of $[0, 90)$. The smaller the angle $\alpha_{g_i w}$ is, the better the model is placed against the wall. Thus, as shown in Fig. 2(a), we can get the front direction g_1. For models in the corner, it can be placed either against the wall or diagonally as shown in Fig. 2(b). In this case, to decide the front direction, we need to take the angles $\alpha_{g_i w_1}, \alpha_{g_i w_2}$ between the model and both walls, and the angle θ_i between g_i and f_c into consideration. If all

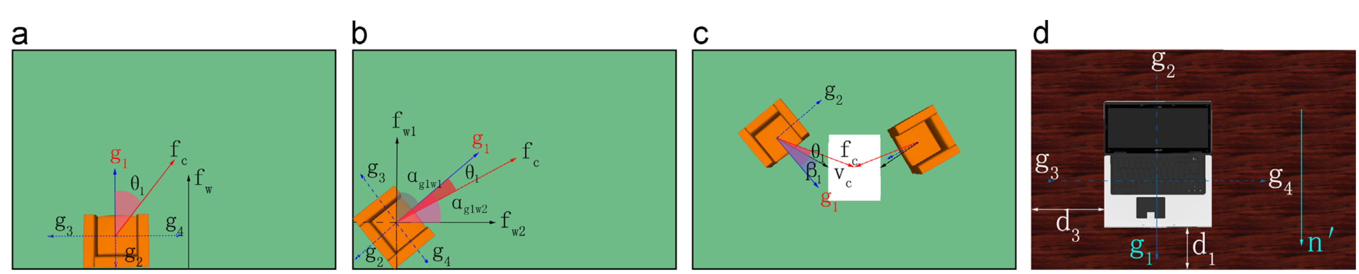

Fig. 2. Front direction extraction. g_1 is the front direction in these examples: (a) sofa against wall and (b) sofa in the corner. We need to consider both two walls in this case: (c) sofas in groups and (d) laptop supported by the desk.

the angles are in the range of $[0, 90)$, then g_i is the front direction, such as g_1 in Fig. 2(b).

Models in groups: Elements in groups are likely to face to the center for functional purpose, such as conversation, projection, and entertainment. We consider the candidate front directions $g_{i,i=1,2,3,4}$ of elements, the vector f_c from the center of the element to the center of the central model, and the vector v_c from the center of the element to the center of a corresponding face of the central model's bounding box. The corresponding face is found by checking the angles between g_i and normals of the bounding box faces and the distance from the center of the model to the bounding box faces. If the angle θ_i between g_i and f_c and the angle β_i between g_i and v_c are both in the range of $[0, 90)$, then g_i has the highest probability of being the front direction. As shown in Fig. 2 (c), g_1 is the most probable front direction.

Supported models: To calculate the front direction of supported models, we need to consider three factors: (1) the distance between the supported model and the edges of supporting frame. Models are likely to be placed near the edge with a good view of the front face. (2) Front faces are likely to have larger area, showing more details. (3) Supported models are likely to have orientations similar to the supporting models. Thus, for each candidate front face, we calculate its confidence coefficient $T(f)$:

$$T(f) = \kappa \exp\left(-\frac{d_f}{\sum_{f \in F} d}\right) + \lambda \frac{s_f}{\sum_{f \in F} s} + \mu(g \cdot n'), \tag{1}$$

where F is the set of candidate faces, f is a candidate face in F, d is the shortest distance between the face center and edges of supporting frame, s is the area of face f and n' is the front direction of supporting model. We set κ, λ, μ to 0.4, 0.5, 0.1, empirically. The face with the largest term value is set as the front face and its normal is the front direction. As shown in Fig. 2(d), g_1 is set as the front direction of the laptop.

Other models: Some models, like bottle, bonsai, vase, do not have a clear front direction. If they are in accordance with conditions mentioned above, we use them to calculate the front

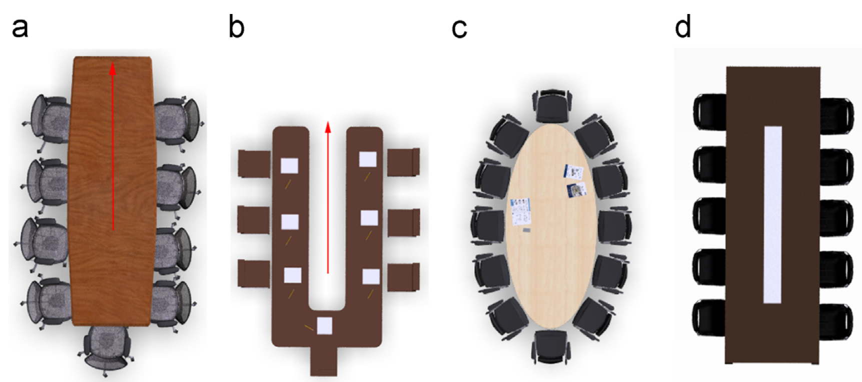

Fig. 3. (a) and (b) Desks with clear front direction, shown by the red arrow. (c) and (d) Desks that do not have a clear front direction. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

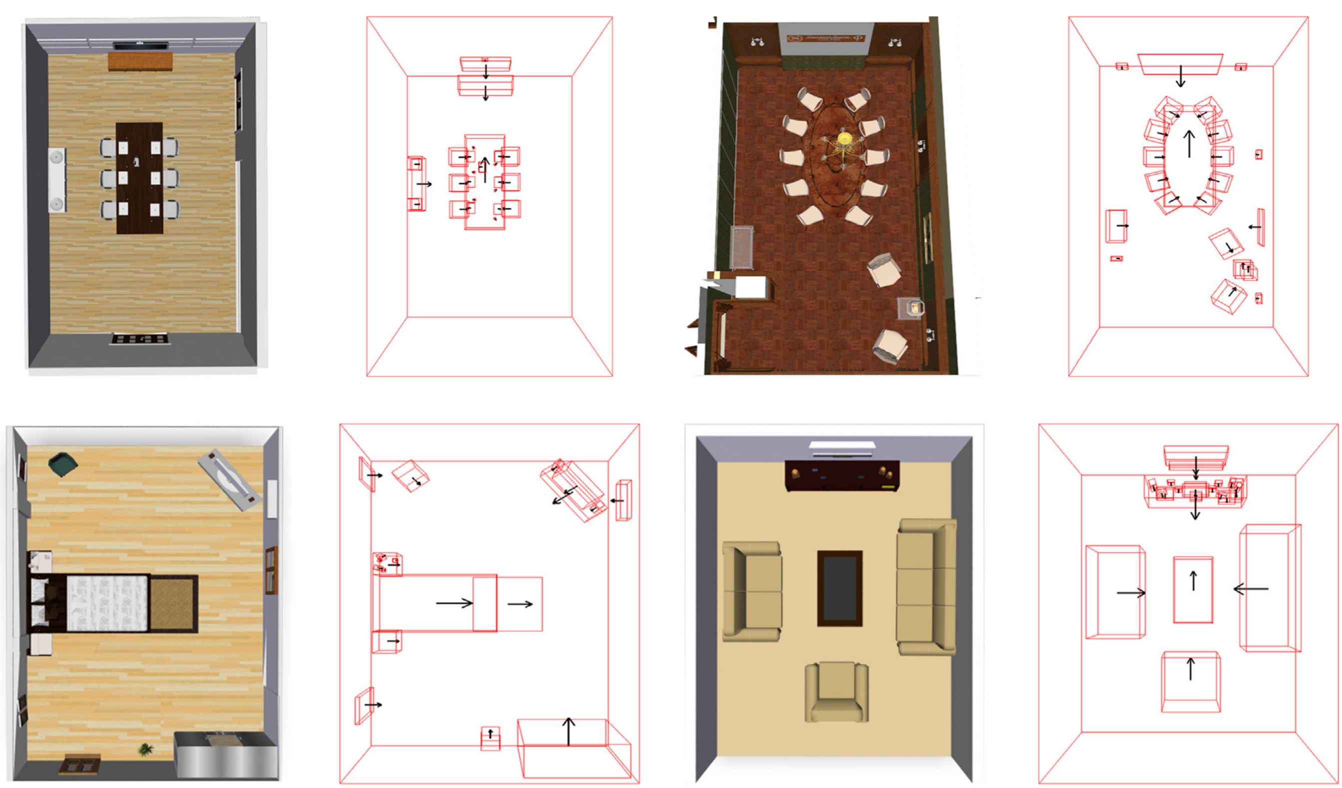

Fig. 4. Results of front extraction in three types of interior scenes: conference room, bedroom and living room.

directions. For models like table and desk, we check models surrounding them. If only one side or three sides have chairs facing it, the front direction corresponds to the special face. If the number of sides is even, the front direction is decided by the front direction of the room, as shown in Fig. 3. We define the front direction of a room based on the orientations of models of core importance, for example, screens of conference rooms, beds of bedrooms and the sofas and TV of living rooms.

We start calculating front directions with models in group relationships. Next, we calculate those with backs against walls, which correct some miscalculated directions in group-based step. The front directions of supported models are calculated after the calculation of their supporting models. Finally, we decide the front direction of the room and recalculate those without clear front directions. Fig. 4 shows some results of our front extraction method in three types of interior scenes.

To evaluate the performance of our front extraction method, we manually specify the front directions of models in the three kinds of interior scenes as a benchmark. The evaluation result is shown in Fig. 5. The accuracy of conference rooms is lower than those of living rooms and bedrooms because conference rooms are more sophisticated. Among the wrongly calculated models, many are supported decorations whose placement has a certain randomness. As our synthesis method is based on FGs, the wrongly calculated decorations do not affect our results. Models in the corner such as a cube style cabinet may also be wrongly calculated, because there are two possible faces reasonable to be the front face. We need to manually correct those ones. The placement of unconventional cases is rare and not considered in our cases.

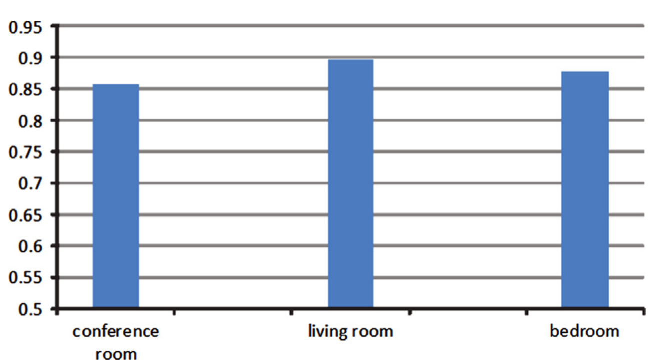

Fig. 5. The accuracy of our front extraction method in conference room, living room and bedroom.

4.2. Functional group

As we do not have examples or sketches as guides, simply recombining models may result in cases like a table with eight different chairs. This is not aesthetically plausible. We use FGs to synthesize scenes, thus guaranteeing that functionally related models have a unified style.

Functional Groups contain one or more models that are functional related. Xu et al. [13] propose the concept of structural groups (SG for short), which consist of multiple objects that have a high co-occurrence frequency. Our FGs capture more of the functional relationships, such as a projector and a screen for presentation purpose, a table and several chairs for conversation purpose. Some of the FGs are the same as SGs, but they are different in some aspects. For example, Sofas, a TV and a tea table is a SG, as they often co-occur in interior scenes. But they are two FGs in our case. Sofas and a tea table form an FG for conversation purpose and a TV alone is an FG for display purpose. In our case, a TV is changeable with a computer, a screen etc., and sofas and a tea table can be replaced by chairs and a table, chairs and sofas and a table, providing us with more flexibility. Moreover, the relationships described in SGs and FGs are also different. We will describe this in the next paragraphs.

The FG is represented as a graph $G=(V,E)$. Each node $v_i \in V$ represents an object and each edge $e_{i,j} \in E$ describes a pairwise relationships between v_i and v_j. The pairwise relationships in FG are listed below:

- *Center-element*: a model with core importance is defined as the center. Objects surrounding it are the elements, e.g., the table and chair in Fig. 6. A center may have no element around it, such as cabinet and bookshelf.
- *Supporting relationship*: models supported by others have the supporting relationship, e.g., the laptop and the desk in Fig. 6.

FGs are extracted automatically based on the relationships which are specified by human. For models that do not have co-appeared models belonging to above relationships, they are treated individually as an FG. Note that we do not consider the spatial relationship, symmetry relationship and coplanar relationship as SGs do, because we treat FGs as the basic units when synthesizing the indoor scenes.

However, some models are not appropriate to do the transformations together although they are functionally related. For example, a screen is always put on the front wall and a projector is hanging on the ceiling. But if the room structure changes dramatically, placing them as one FG may result in an unappropriate projection distance. So we need to split them into two FGs to

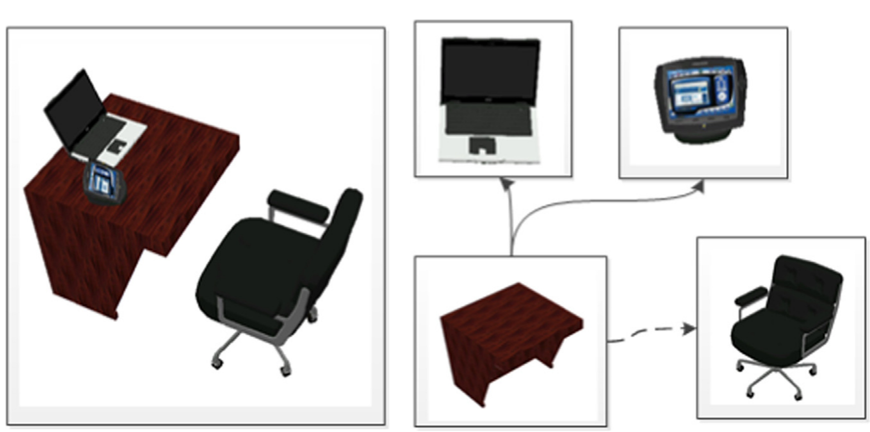

Fig. 6. The solid lines represent the supporting relationship and the arrows point to supported objects. The dashed line represents center-element relationship with arrow pointing to elements.

make sure that each one is correctly placed. To ensure the functional integrity in such cases, the split FGs should be either arranged or not arranged at all. We find this kind of FGs by their corresponding supporting walls. In an FG, if the center and elements are supported by different walls, we need to split them for proper arrangements.

Through this step, scenes in our database are translated into sets of FGs. We then find out the main FGs (mFG for short), which appear in more than 80 percent of scenes in our database. The mFGs are linked to the main functions of scenes, such as FGs of table and chairs, projector and screen in conference room and FGs of bed in bedroom.

4.3. Wall grid structure analysis

The WGS is used to learn the arrangement information from the database, which incorporates design guidelines as well as designers' experience. While the Gaussian Mixture Model can find possible positions of models, it typically knows their types and decides possible positions by fixing one model. Our WGS learns the relationships between model categories and positions, making it possible to both decide model categories by positions and find positions by model categories. To establish the WGS, we select the bottom left corner of the room as origin after rotating the entire scene according to the front and up directions of the room model. As shown in Fig. 7(b), each wall is divided into a set of grid cells. The WGS is the set of grid cells divided on the wall, denoted as $WG = \{wg_1, wg_2, \dots\}$. We use M, N, K to represent the resolutions of the length, width and height of the room, respectively. The lager the resolution is, the more precise the WGS is. In our implementation, we set $M = 7$, $N = 5$ and $K = 5$ empirically by default. These values can be adapted to the aspect ratio of the given room shapes. 3D cells are not used because many 3D cells in the center contain models that are either supported or indirectly supported by walls and models suspending in the air are rare in real world.

Arrangement analysis: We first find the supporting wall surfaces w of an FG and project the center of the FG to w. We say an FG lies in a wall grid cell wg if its projected center lies in wg. For each $wg \in WG$, we record the categories c of FGs that lie in it and their corresponding occurrence times t. Thus, for each wg, we get a set $S = \{(c_1, t_1), (c_2, t_2)\cdots\}$. We calculate the probabilities $P(wg)$ and $P(c|wg)$ using the following equations:

$$P(wg) = \frac{\sum_{s \in S} t(s)}{\sum_{wg \in WG} \sum_{s \in S} t(s)}, \tag{2}$$

$$P(c|wg) = \frac{t(s|c)}{\sum_{s \in S} t(s)}, \tag{3}$$

$$P(c, wg) = P(wg) * P(c|wg), \tag{4}$$

where $t(s)$ is the occurrence number corresponding to an item $s \in S$, $P(wg)$ is the probability of a wg occupied by FGs. Given a wg, $P(c|wg)$ is the probability an FG of category c lies in it. $P(c, wg)$ denotes the probability an FG of category c lies in a wg, according to which we decide our needed FGs.

Symmetry detection: The symmetry in our scene is detected in a wall and between walls. As show in Fig. 7(c), the red lines are symmetry axis for detecting the up and down, left and right symmetry in vertical walls and on the floor. If two FGs are geometrically the same and the grid cells in which they lie, wg_1, wg_2, are symmetrical according to the symmetry axis, we say that a symmetry pair in category c, located in wg_1, wg_2, is detected. For example, the bonsai in Fig. 7(a) are detected to be

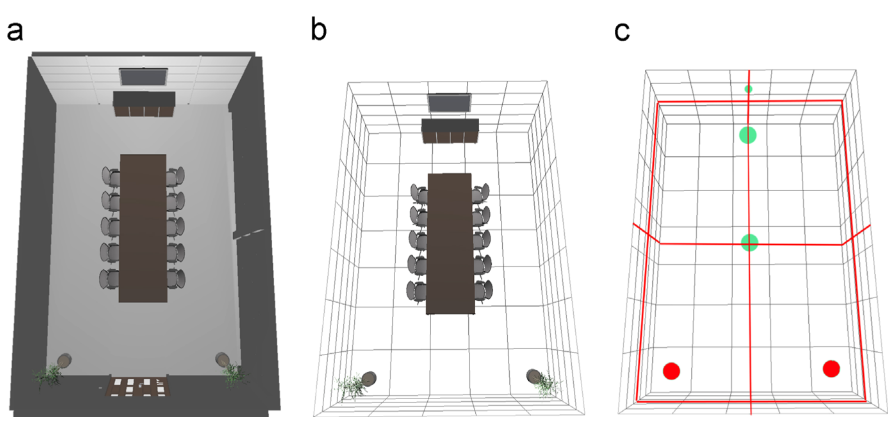

Fig. 7. (a) A conference room. (b) The WGS of (a). (c) The symmetry axis in WGS. Two bonsai in (a) are symmetric as labeled in red dots. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

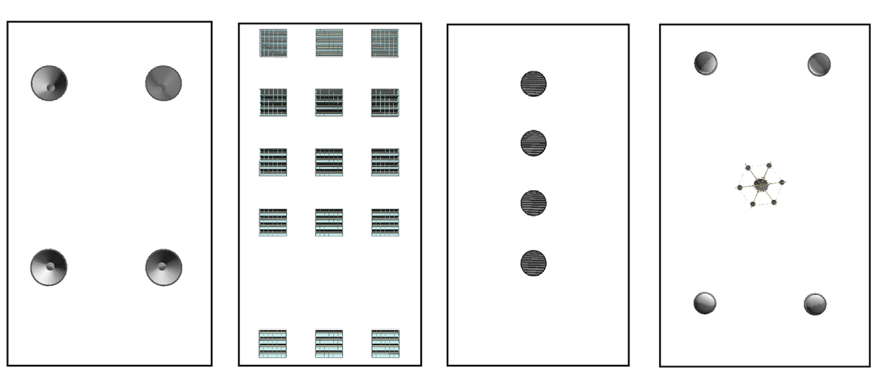

Fig. 8. Different ceiling light patterns.

symmetric with respect to vertical symmetry axis on the floor. The symmetry information between walls is detected between the left and right, front and back walls. If two FGs are geometrically the same and lie in symmetric grid cells in different walls, we say that they are symmetric between walls. For example, the wall lamps on the left and right wall are always symmetric.

Ceiling light pattern: The illumination of a room is highly related to the number and arrangement of ceiling lights. We cannot directly learn the arrangement information of ceiling lights in the same way as what we do in arrangement analysis step. Because ceiling lights are always placed in aesthetic ways, as shown in Fig. 8. Directly placing ceiling lights by probabilities may produce disordered and inaesthetic results. We do not choose to treat all ceiling lights as an FG and place them as a whole part, because transformations of the FG may damage the structure of individual ceiling lights. Thus, learning the arrangement patterns of ceiling lights is needed. We treat the arrangement of ceiling lights in a scene as a pattern $P=\{(FG_1, wg_1), (FG_2, wg_2)\ldots\}$. Each ceiling light in the pattern is represented by its corresponding FG and location wg. When placing a ceiling light pattern, lights should be all placed in their corresponding locations. If not, we will change to another pattern rather than deforming the used one.

5. Synthesizing new scenes

In this section, we present an approach to generate a diverse set of plausible and well-arranged indoor scenes using WGS. This method can automatically analyze the available space in the room, find FGs with the maximum probabilities and place them with appropriate orientations and locations. We use the height of a room as a reference to calculate the scaling factor. The scene synthesis method consists of three sub-steps: room occupancy detection, FGs arrangement and scene assessment.

5.1. Occupancy and overlapping detection

When users input an empty room including its structure, scene type and front direction, our method automatically establishes its corresponding WGS and calculates its occupancy information, as shown in Fig. 9. Our WGS consists of $2*(M*N+M*K+N*K)$ grid cells. Each grid cell has a state flag indicating whether this cell is occupied or not. Fig. 9(a) is an empty input room and Fig. 9(b) shows its occupancy information. Fig. 9(d) shows the updated occupancy information of scenes with FGs placed. Note that the occupancy information on both the floor and vertical wall is updated if models are placed against wall such as the cabinets in Fig. 9(c). The occupancy information is calculated by analyzing the bounding boxes of FGs and its covering area.

Overlapping may happen between FGs. Some guidelines [11,12] reveal that pathways and clearance around models are necessary for function and accessibility purposes, e.g., cabinets need space in front of them and dinning tables need space around them. Merrell et al. [11] use an anthropometric constraints table to specify the clearance of different models. In our method, we ensure that enough clearance is set aside in the front faces of models against walls and around the entire perimeter of models in the center. To check the clearance constraints, we simply enlarge the bounding box of FGs to cover the clearance and check overlapping based on them. For FGs in the central part of walls, such as pictures, clocks and desks, overlapping is checked based on grid cells it covers. For FGs against walls, such as bookshelves, cabinets and beds, grid cells on the floor and grid cells on the vertical wall which they put against are considered. Only when no conflict exists in all covered grids, we say that no overlapping is detected.

5.2. FGs arrangement

To furnish an interior scene, we consider the design process that furniture objects with great importance have the priorities. mFGs which appear in more than 80% of scenes in our databases are placed first. Other supplemental FGs are added next. Finally, the arrangement of illumination objects is considered according to the layout of the scene. We achieve our goal through three steps: seed scene generation, supplemental FG arrangement and ceiling pattern arrangement.

Seed scene generation: Seed scenes correspond to scenes with mFGs arranged as shown in Fig. 10(b). Some kinds of interior scenes only have one mFG, such as bedrooms. Others may have more than one, such as conference rooms. If more than one mFG exist, we then generate seed scenes based on the number of mFGs. For example, if there are M FGs of conference table and N FGs of projector and screen, there will be $M*N$ seed scenes. Some of them might be removed according to the conflict test. The locations of mFGs are determined by $P(wg|c)$ which is derived from Eqs. (2) to (4) using Bayes' rule and the orientations of mFGs are decided by the rules used in the front direction extraction method. The scale factor is calculated by the ratio of height between mFG and room shape. If the mFG has back-against-wall property and is placed near the wall, we automatically transform it to have its back against wall.

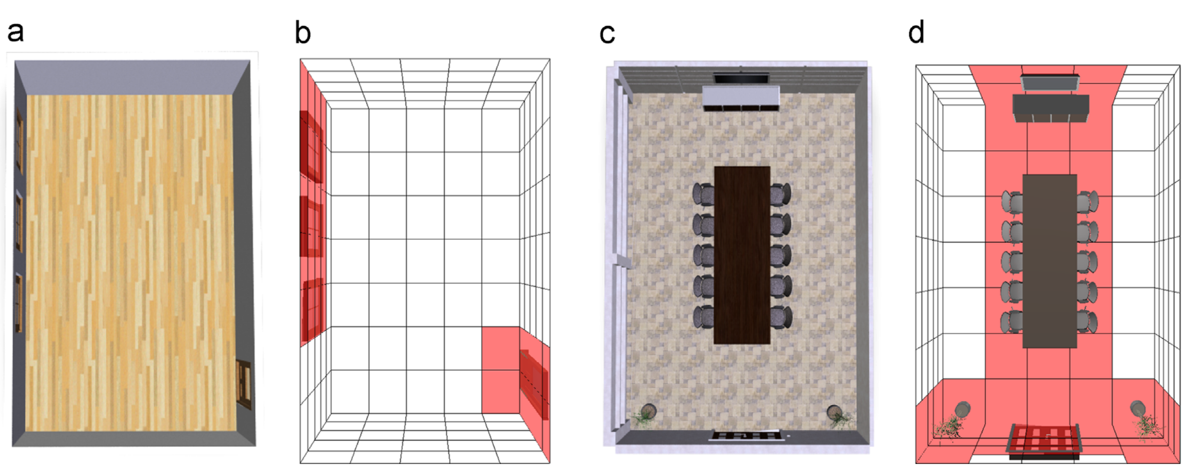

Fig. 9. Red means the grid cell is occupied. (a) Input empty room structure. (b) The occupancy information of (a) is shown. (c) Room with some FGs placed is shown. (d) The occupancy information of (c). (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

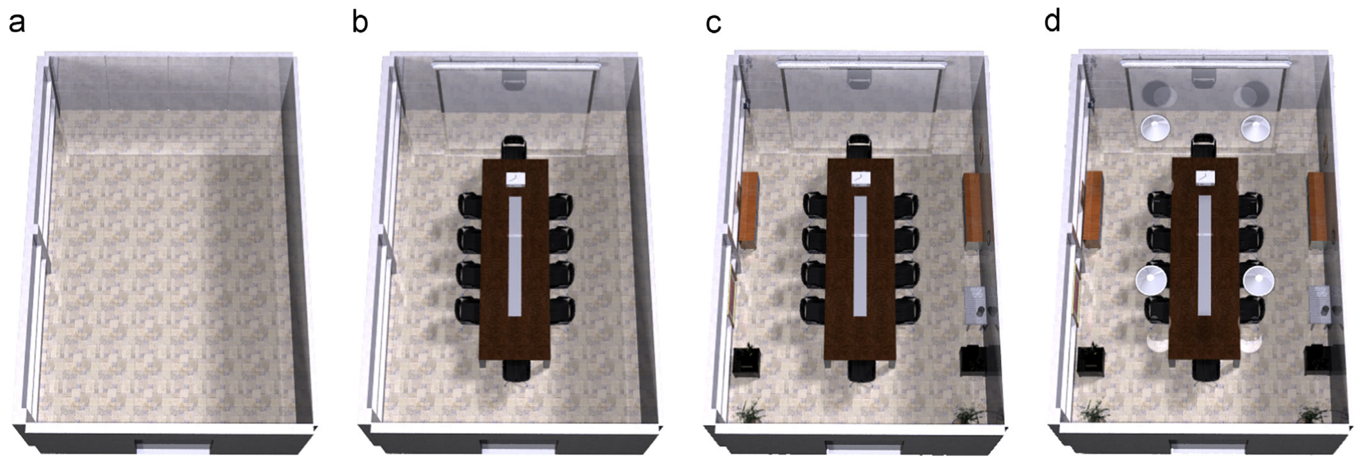

Fig. 10. (a) The input empty room structure. (b) The result with mFGs placed is shown. (c) The result with supplemental FGs placed is shown. (d) The final result with ceiling light pattern placed.

Supplemental FG arrangement: Supplemental FGs are added after the placement of mFGs. The decision of supplemental FGs is evaluated by $P(c, wg)$. When placing FGs, we need to consider the symmetry information detected. If wg_1 and wg_2 are symmetry grid cells in category c and, a FG of category c is going to be placed in wg_1, then the same FG has to be also placed in wg_2. In order to lower the probabilities of chosen FGs, we decrease $P(c, wg)$ by a decreasing factor α after the arrangement of an FG of category c in wg. When no FG is added, an intermediate scene with supplemental FGs arranged is generated as shown in Fig. 10(c). For one seed scene, it can generate lots of intermediate scenes with different supplemental FGs.

Ceiling pattern arrangement: Ceiling lights which have a tight relationship with the illumination of a room are decided based on the layout of the room [1]. Inspired by Lam et al. [26], we use *point-wise mutual information* (PMI) to characterize the relationships between the number of ceiling lights and the number of furniture:

$$P(C|n_f) = \sum_{n \in N} \frac{PMI(n_c, n)}{0.01 + \omega * abs(n_f - n)} \quad (5)$$

where C is the ceiling pattern, n_c is the number of lights in C, n_f is the furniture number and N is the set of furniture numbers related to C. Given an intermediate scene, we find its most suitable ceiling patterns by $P(C|n_f)$ and place it into the room (Fig. 10(d)).

Ceiling pattern is added immediately after the generation of an intermediate scene. The generation step of a seed scene ends after a repeat times T or a failure result generated. We reset the probability $P(c, wg)$ in the beginning of the generation step of another seed scene.

5.3. Scene assessment

Since our results are achieved by recombining FGs based on $P(c, wg)$, some results may not be pleasant. To assess the quality of synthesized scenes, some basic requirements that a good scene should meet are introduced:

- *Functional requirements*: A good synthesized scene should meet the main functional requirements of its type. For example, bedrooms should have beds and conference rooms should have conference tables, chairs and displaying devices.
- *Illumination requirements*: Illumination equipments in an interior scene are related to windows and ceiling lights. Good synthesized scenes should consider the overlapping problem of windows, doors and the patterns of ceiling lights.
- *Aesthetic requirements*: Well-designed interior scenes share a common point that they are in accordance with some aesthetic standards. We consider three kinds of aesthetic requirements: symmetry requirements, the richness of arranged furniture and the degree of crowdedness. A good scene can neither be too crowd nor too empty.
- *Greening requirements*: The greening condition which corresponds to the arrangement of plants is another consideration in interior design nowadays.

As our algorithm already considers mFGs and ceiling light pattern arrangements in scene synthesis stage, our assessing algorithm concentrates on aesthetic requirements and greening requirements. Symmetry information is considered but not guaranteed in arrangement step. The richness of generated scenes, which is considered by the number of categories of FGs in a scene and the degree of crowdedness can be approximately calculated by checking the states of grid cells. Greening condition is judged by checking if there is an FG corresponding to plants in the results. Ten penalty points are added if one condition is not matched.We discard scenes with penalty scores above 20. This step guarantees that our results meet the function, illumination, aesthetics and greening requirements of interior design.

6. Results

6.1. Synthesis results

We evaluate the proposed framework with three kinds of interior scenes: conference room, bedroom and living room. Our database are directly downloaded from Google warehouse. Conference room database consists of 52 scenes and 1111 models. Bedroom database contains 45 scenes and 741 models. Living room database has 48 scenes and 875 models. It takes about 7 min per scene to do the segmentation, annotating and relationship specifying using sketchup. Functional groups are extracted based on the relationships automatically. The generated number of interior scenes is decided by the input room shape, seed number, repeat times T and scene assessment. For the input room structure in Fig. 11(a) and the repeated times T set to 10, we get 198 new scenes. For the input room structure in Fig. 11(d) and the repeated times T set to 5, we get 154 new scenes. As the selection of models has some kinds of randomness, the number of generated scenes diverse each time. However, the larger the database and repeated times T are, the more results will be generated.

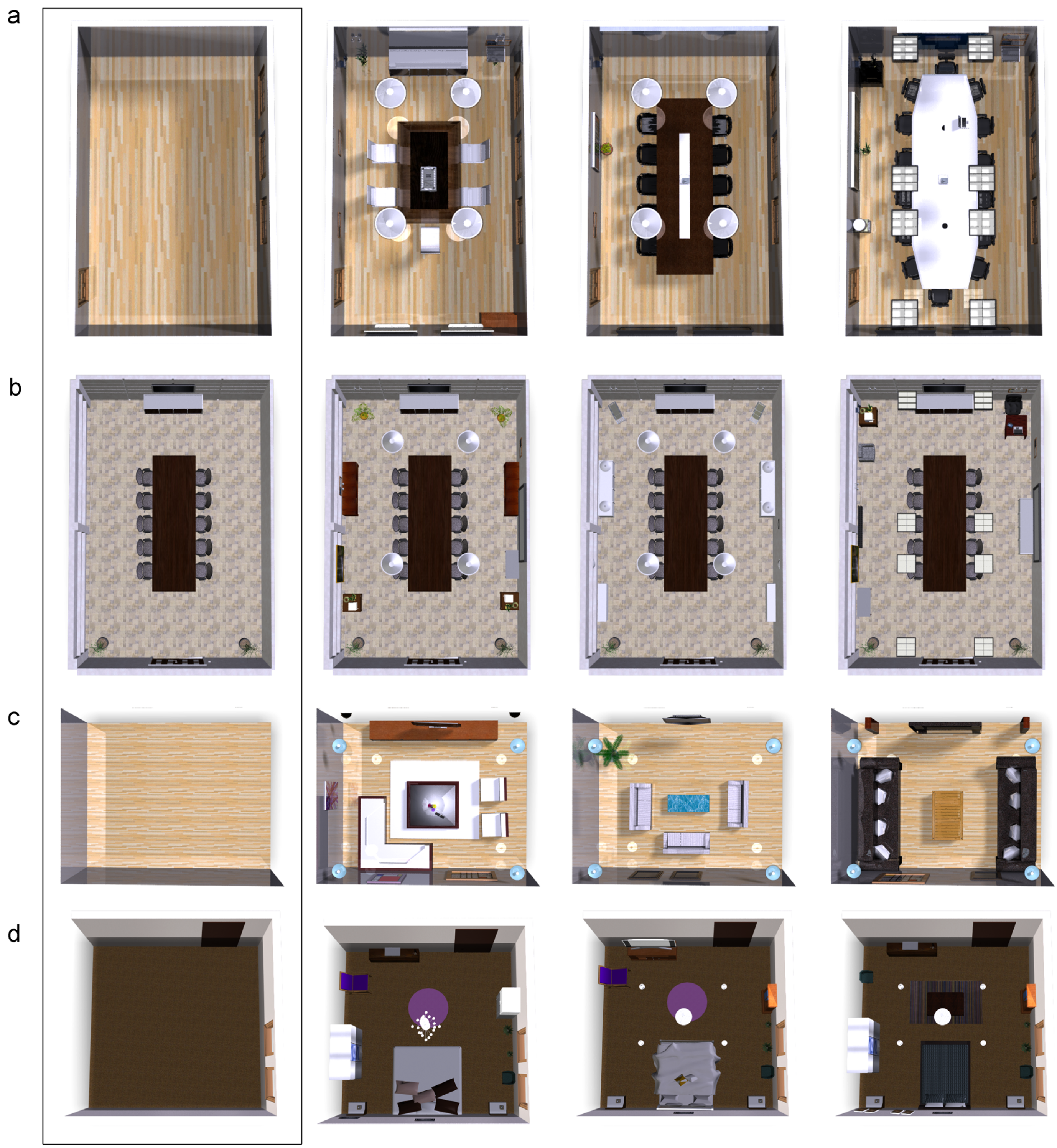

Fig. 11. (a) Results of giving an empty conference room. (b) Results of giving a conference room with some pre-placed models. (c) Results of giving an empty living room. (d) Results of giving an empty bedroom.

Fig. 11 shows some selected results generated by our framework. Fig. 11(a) shows the results of generated new conference rooms from an empty room. The room has three windows and a door. From the results, we can see that our method can automatically generate interior scenes without violating the constraints of the input room and design guidelines. Besides, the selection of ceiling light pattern corresponds to the numbers of furniture used. Fig. 11(b) shows that our method can be used to replenish a half-made room with models, which is very useful for designers, who only have a general idea about their design. Fig. 11(c) and (d) shows the results of bedroom and living room.

Users can also orientate the results by selecting their desired FGs from generated results. As shown in Fig. 12, given the empty room shape in the top left corner, our system automatically generates a diverse set of results. Furniture objects that users interested in are combined as an intermediate scene, shown in

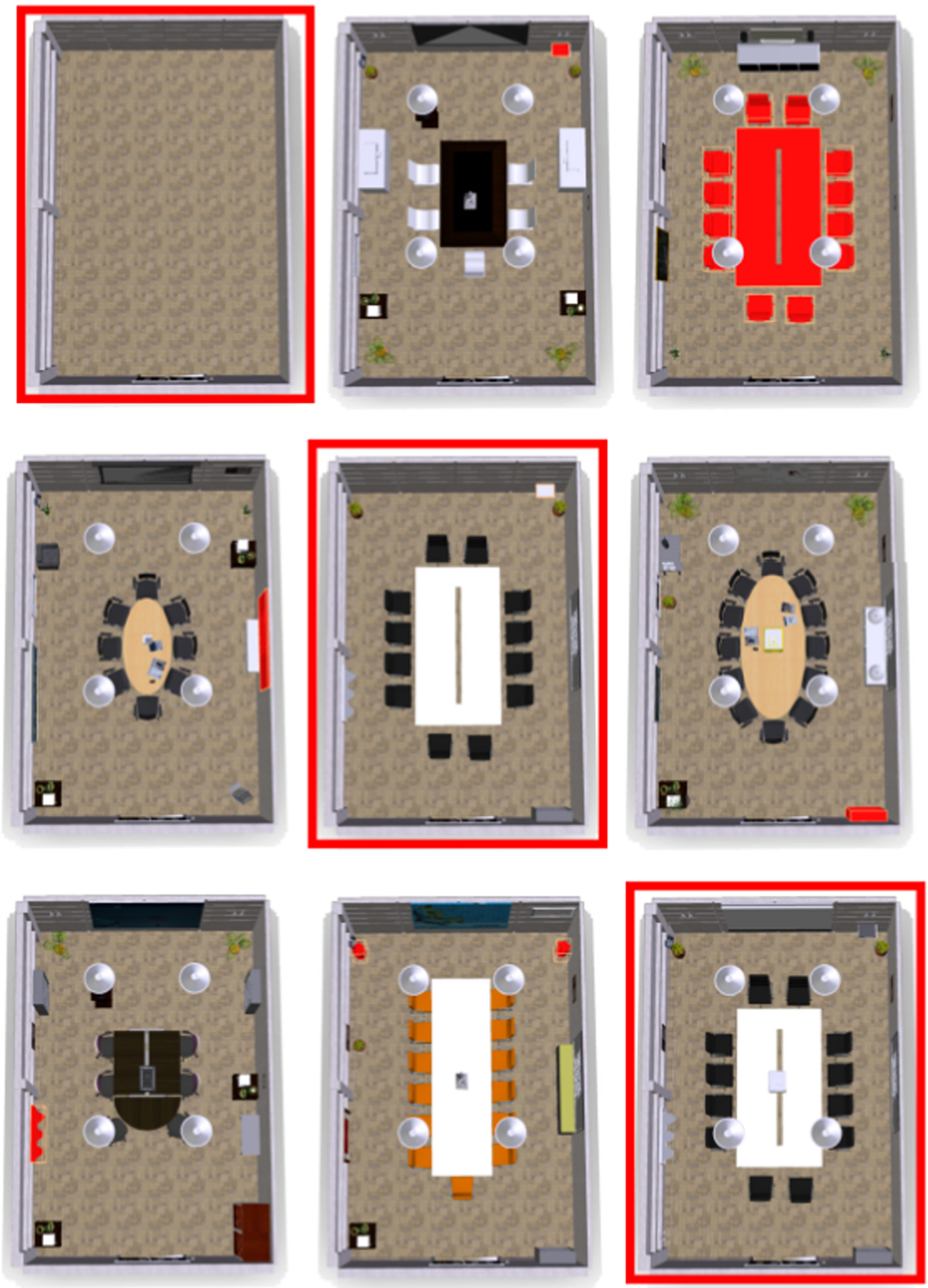

Fig. 12. Top left corner is the input room shape. The center is the intermediate scene generated by users' desired furniture. The bottom right corner is the synthesised scenes.

the center. Our method can turn the intermediate result to a functionally and aesthetically plausible interior scene, shown in the bottom right corner. All furniture objects user preferred are preserved and some more furniture which can replenish and beautify the results are added, such as the screen, projector and cabinet.

Fig. 14 shows the influence of windows and doors on room layouts. Compared to Fig. 14(a) and (c), boards and cabinets are not

Fig. 13. Unsatisfied results.

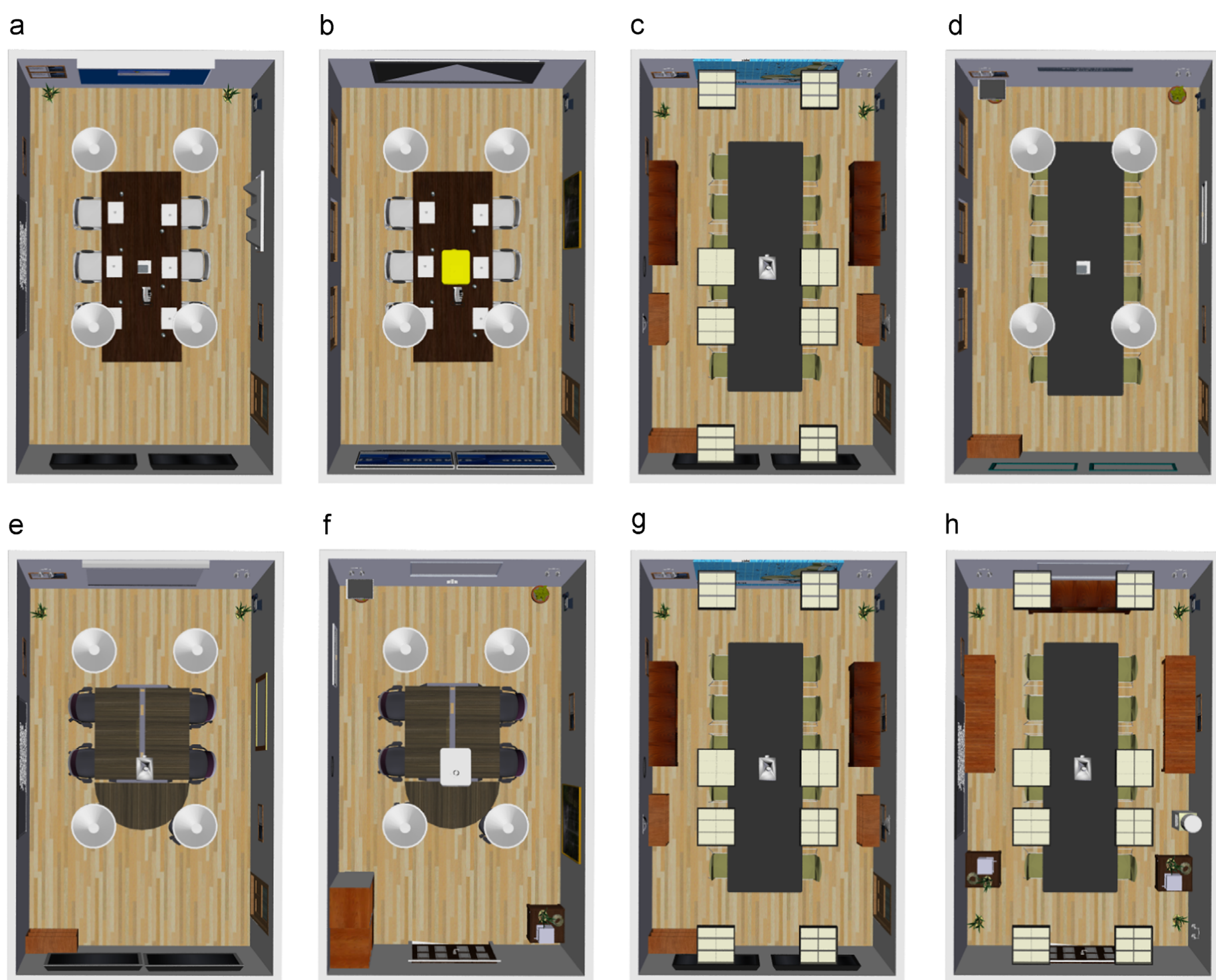

Fig. 14. Boards, pictures and cabinets are placed on and against the left wall in (a) and (c). They are not placed in (b) and (d) for the visibility of windows. TVs are placed on the back wall in (e) and (g), cabinets and bonsai are placed in the bottom right corner. In (f) and (h), since the door is located to the middle of back wall, the TVs cannot be preserved anymore.

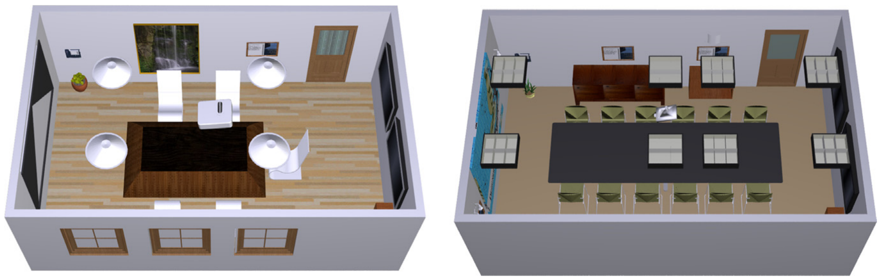

Fig. 15. Another viewpoint shows the doors and windows of our results.

Fig. 16. Results of applying our method to a house plan. Our method generates the layouts of two bedrooms, one living room and one conference room in the house plan step by step.

placed in Fig. 14(b) and (d), because they damage the visibility of windows. In Fig. 14(e) and (g), TVs are placed on the back wall. However, since the door is enlarged and located to the middle of back wall, the TVs cannot be preserved in Fig. 14(f) and (h). Fig. 15 shows the windows and doors of our results from another viewpoint.

We also apply our method to house plans to generate a complete design as shown in Fig. 16. By gradually applying our method to different room space in the house plan, some designed plans with living rooms, conference rooms and bedrooms well furnished are synthesized.

Our framework might also generate unsatisfied results in some cases. Fig. 13(left) shows the case that the selected mFGs are too small, such that the result contains a lot vacant space. Fig. 13(right) shows the case that same FGs (cabinets) appear too many times in one room. Our method cannot filter out such cases. However, it only happens in a very low possibility.

The style and quality of our generated results partially rely on the database used for learning. If all scenes in the database are modern with a very few furniture but a high percentage of useful ones and a few purely aesthetic ones, many grid cells in the learned WGS may be vacant. The generated results shall share the same style with scenes in the database. Penalty scores need to be modified to filter out crowded scenes.

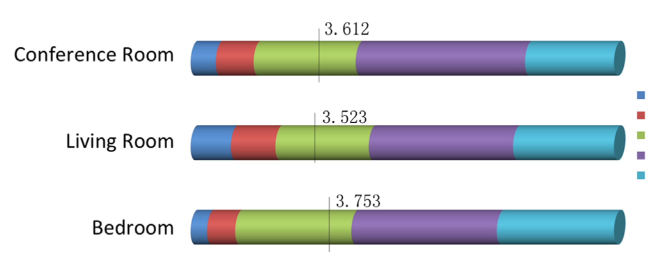

Fig. 17. User study on three types of generated scenes.

6.2. User evaluation

To evaluate whether our system generates plausible scenes, we ran a user study on the synthesized scenes of three types : conference room, bedroom and living room. For each scene type, we set $T=3$ and randomly select 30 scenes from the synthesis results. We then render images of all these scenes and 13 participants are asked to specify the plausibility of the scene on a 5-point Likert scale [14] (1=implausible scene, 3=somewhat plausible scene, and 5=very plausible scene).

Fig. 17 shows a summary of the ratings obtained. About 60% scenes get the results of 4 and 5, and over 80% scenes are somewhat plausible. The result suggests that on average, at least 80% of synthesized scenes can give you some design ideas in the real world.

7. Conclusion

In this paper, we propose a novel framework for interior scene synthesis from an empty room. The context-based front extraction method introduced can automatically get a knowledge of the orientations of models. Our newly introduced concept of WGS can effectively analyze the arrangement and symmetry information of models. We also present a few criteria to assess the quality of generated scenes. Experimental results demonstrate that our approach can synthesize well-arranged, functionally and aesthetically plausible interior scenes given an empty room shape, with few user intervention. The results are diverse and can provide clients and designers with suggestions and inspirations.

There are several limitations to our current system. First, the WGS is cuboid, which is intrinsic and not suitable for space in complex shapes. We would like to try polyhedra cells which can considerably change based on the room shape instead of rectangle cells and explore a more general WGS shape for complex shape rooms. Second, we only consider the type, fitness and symmetry information when recombining new FGs. Adding "style semantic information" can make the results more consistent. New scenes, such as "antique style scene" or "modern style scene" can be generated either. In our current implementation, we did not take the relationships between neighboring FGs into account. We would like to address this issue in the future. Applying WGS-based synthesis method to large scale indoor scenes, such as coffee shops or exhibition halls, would be a new challenge.

Acknowledgments

This work was supported by the National Basic Research Program of China (2010CB328001), the National Science Foundation of China (61373071, 61372168), and the KAUST Visual Computing Center.

Appendix A. Supplementary data

Supplementary data associated with this paper can be found in the online version at http://dx.doi.org/10.1016/j.cag.2014.09.032.

References

[1] Dodsworth S. The fundamentals of interior design. AVA Publishing; 2009 p. 12–88.
[2] Kalogerakis E, Chaudhuri S, Koller D, Koltun V. A probabilistic model for component-based shape synthesis. ACM Trans Graph 2012;31(4) p. 55:1–11.
[3] Chaudhuri S, Kalogerakis E, Guibas L, Koltun V. Probabilistic reasoning for assembly-based 3D modeling. ACM Trans Graph 2011;30(4) p. 35:1–10.
[4] Jain A, Thormählen T, Ritschel T, Seidel H-P. Exploring shape variations by 3D-model decomposition and part-based recombination. In: Computer graphics forum, vol. 31. Wiley Online Library; 2012. p. 631–40.
[5] Zheng Y, Cohen-Or D, Mitra NJ. Smart variations: functional substructures for part compatibility. Comput Graph Forum (EUROGRAPHICS) 2013;32 (2pt.2):195–204.
[6] Mitra NJ, Wand M, Zhang H, Cohen-Or D, Bokeloh M. Structure-aware shape processing. In: EUROGRAPHICS state-of-the-art report; 2013.
[7] Nan L, Xie K, Sharf A. A search-classify approach for cluttered indoor scene understanding. ACM Trans Graph 2012;31(6) p. 137:1–10.
[8] Shao T, Xu W, Zhou K, Wang J, Li D, Guo B. An interactive approach to semantic modeling of indoor scenes with an rgbd camera. ACM Trans Graph 2012;31(6) p. 136:1–11.
[9] Kim YM, Mitra NJ, Yan D-M, Guibas L. Acquiring 3D indoor environments with variability and repetition. ACM TOG 2012;31(6) p. 138:1–11.
[10] Sharf A, Huang H, Liang C, Zhang J, Chen B, Gong M. Mobility-trees for indoor scenes manipulation. Comput Graph Forums 2014;33:2–14.
[11] Merrell P, Schkufza E, Li Z, Agrawala M, Koltun V. Interactive furniture layout using interior design guidelines. ACM Trans Graph 2011;30(4) p. 87:1–10.
[12] Yu L-F, Yeung S-K, Tang C-K, Terzopoulos D, Chan TF, Osher SJ. Make it home: automatic optimization of furniture arrangement. ACM Trans Graph 2011;30 (4) p. 86:1–12.
[13] Xu K, Chen K, Fu H, Sun W-L, Hu S-M. Sketch2scene: sketch-based co-retrieval and co-placement of 3D models. ACM Trans Graph 2013;32(4) p. 123:1–15.
[14] Fisher M, Ritchie D, Savva M, Funkhouser T, Hanrahan P. Example-based synthesis of 3D object arrangements. ACM Trans Graph 2012;31(6) p. 135:1–11.
[15] Fisher M, Savva M, Hanrahan P. Characterizing structural relationships in scenes using graph kernels. ACM Trans Graph 2011;30(4) p. 34:1–12.
[16] Fisher M, Hanrahan P. Context-based search for 3D models. ACM Trans Graph (TOG) 2010;29(1–10):182.
[17] Yeh Y-T, Yang L, Watson M, Goodman ND, Hanrahan P. Synthesizing open worlds with constraints using locally annealed reversible jump mcmc. ACM Trans Graph 2012;31(4) p. 56:1–11.
[18] Lin S, Ritchie D, Fisher M, Hanrahan P. Probabilistic color-by-numbers: suggesting pattern colorizations using factor graphs. ACM Trans Graph 2013;32(4):37.
[19] Yeh Y-T, Breeden K, Yang L, Fisher M, Hanrahan P. Synthesis of tiled patterns using factor graphs. ACM Trans Graph 2013;32(1) p. 3.
[20] Xie H, Xu W, Wang B. Reshuffle-based interior scene synthesis. In: Proceedings of the VRCAI; 2013. p. 191–8.
[21] Eitz M, Richter R, Boubekeur T, Hildebrand K, Alexa M. Sketch-based shape retrieval. ACM Trans Graph 2012;31(4) p. 31:1–10.
[22] Le Roux O, Gaildrat V, Caubet R. Using constraint propagation and domain reduction for the generation phase in declarative modeling. In: Conference on information visualisation; 2001. p. 117–23.
[23] Sanchez S, Roux O, Luga H, Gaildrat V. Constraint-based 3D-object layout using a genetic algorithm. In: Conference on computer graphics and artificial intelligence; 2003.
[24] Larive M, Roux OL, Gaildrat V. Using meta-heuristics for constraint-based 3D objects layout. In: Conference on computer graphics and artificial intelligence; 2004. p. 11–23.
[25] Fu H, Cohen-Or D, Dror G, Sheffer A. Upright orientation of man-made objects. ACM Trans Graph (TOG) 2008;27(1–8):42.
[26] Lam L, Lin S, Hanrahan P. Using text *n*-grams for model suggestions in 3D scenes. In: SIGGRAPH Asia 2012 technical briefs; 2012. pp. 30:1–4.