This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2021.3111729, IEEE Transactions on Visualization and Computer Graphics

Context-Consistent Generation of Indoor Virtual Environments based on Geometry Constraints

Yu He, Ying-Tian Liu, Yi-Han Jin, Song-Hai Zhang*, Yu-Kun Lai, and Shi-Min Hu

Abstract—In this paper, we propose a system that can automatically generate immersive and interactive virtual reality (VR) scenes by taking real-world geometric constraints into account. Our system can not only help users avoid real-world obstacles in virtual reality experiences, but also provide context-consistent contents to preserve their sense of presence. To do so, our system first identifies the positions and bounding boxes of scene objects as well as a set of interactive planes from 3D scans. Then context-consistent virtual objects that have similar geometric properties to the real ones can be automatically selected and placed into the virtual scene, based on learned object association relations and layout patterns from large amounts of indoor scene configurations. We regard virtual object replacement as a combinatorial optimization problem, considering both geometric and contextual consistency constraints. Quantitative and qualitative results show that our system can generate plausible interactive virtual scenes that highly resemble real environments, and have the ability to keep the sense of presence for users in their VR experiences.

Index Terms—Virtual Reality, Obstacle Awareness, User Interaction, Scene Generation, Geometric Constraints, Contextual Relation, Layout Patterns

1 Introduction

In virtual reality (VR) systems for head-mounted displays (HMDs), the sense of presence is mainly affected by visual and aural cues [1], [2], [3]. However, most of the time, VR users are in environments containing real-world objects. When users are wearing HMDs, they can easily collide with real-world objects since the surrounding physical environments cannot be seen, which may reduce their sense of presence in the virtual environment, and make it impossible to interact with real-world objects.

To avoid unintended collisions between users with HMDs and real-world objects, the easiest and most direct way is to alert the user of the approaching obstacles with non-visual feedback [4]. Alternatively, visualizing 3D images of real-world objects [5], [6], and incorporating visual indicators such as wireframes into virtual environments through mixed reality (MR) or augmented reality (AR) techniques [7], [8] are also commonly adopted. However, these methods introduce settings that are different from the real space into the virtual environment, which may reduce the sense of presence. To enable interactions with real objects, a basic idea is to detect the 3D positions and orientations of real-world objects, and replace them with virtual objects in the virtual environment [9], [10], [11]. But existing methods mainly consider simple virtual scenes with specific virtual objects such as rocks and trees, and perform replacements based on simple rules [12], which cannot handle complex indoor scenes with various categories of objects.

- Yu He, Ying-Tian Liu, Song-Hai Zhang and Shi-Min Hu are with the Department of Computer Science and Technology, Tsinghua University, Beijing, China.
- Yi-Han Jin is with the Department of Computer Science and Engineering, Tianjin University of Technology, Tianjin, China.
- Yu-Kun Lai is with School of Computer Science and Informatics, Cardiff University, Wales, UK.

The methods above are designed for different needs, such as obstacle awareness and physical interaction, and a fundamental requirement for these methods is to ensure the sense of presence for VR systems. In this paper, we propose a method to simultaneously satisfy the above two needs with a high-level sense of presence. Our system creates a virtual scene containing virtual objects of the same size and position as objects in the real scene. Unlike simple virtual obstacle replacement [12], the generated scenes of our system have rich content with context consistency. Meanwhile, by following the object layout patterns, which include the occurrences and spatial characteristics of objects extracted from existing scene configurations, our system ensures plausibility of the generated virtual scenes, making them highly resemble real environments.

To further support interaction, unlike the work [10] that detects the position and orientation of interactive objects when the user is walking, we design an algorithm that detects interactive planes in the real scene and aligns these planes with interactive surfaces of virtual objects. In this work, we define interactive planes as support planes, where the users can place the virtual representation of a real object. The key contributions of our paper are as follows:

- We present a system that can automatically generate immersive interactive virtual reality environments with rich contents by using physical environments as geometric constraints, which ensures that real-world obstacles can be avoided.
- We propose an object layout pattern extraction pipeline for object selection and placement.
- We regard virtual object replacement as a combinatorial optimization problem considering both physical environment geometric constraints and object layout patterns.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2021.3111729, IEEE Transactions on Visualization and Computer Graphics

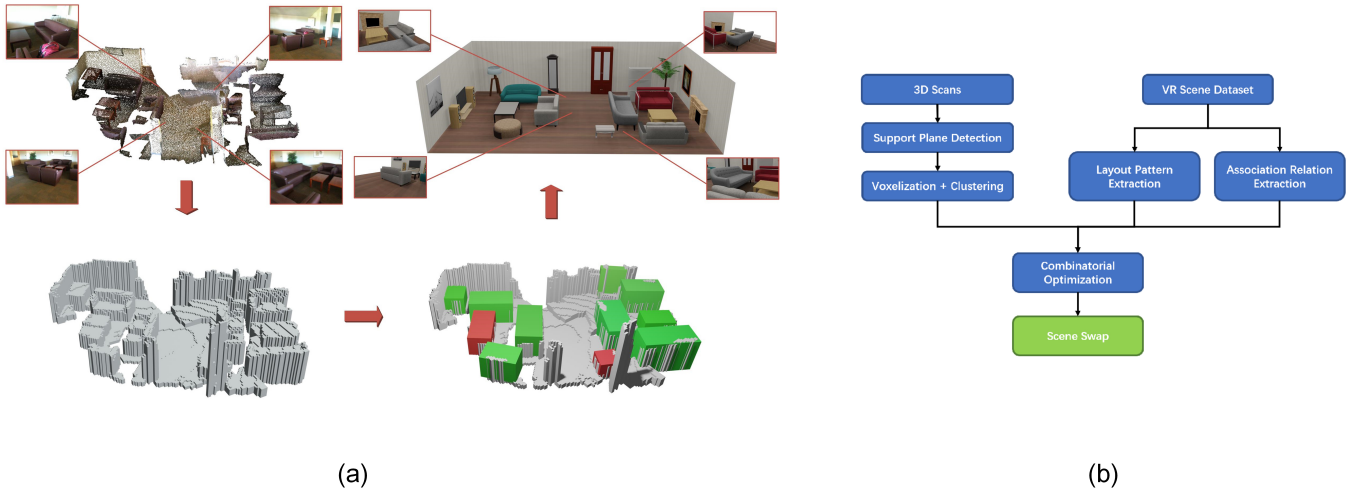


Fig. 1. Pipeline of the proposed system. (a) An example showing the major steps in our pipeline. We start with voxelizing the 3D scan of the real environment (left column). Then we detect interactive planes (red boxes at the bottom right) and cluster the voxels (bottom row). Finally, we replace the clusters with virtual objects based on geometric and context constraints to generate a virtual scene (right column). (b) Workflow of the proposed system summarizing the major steps involved.

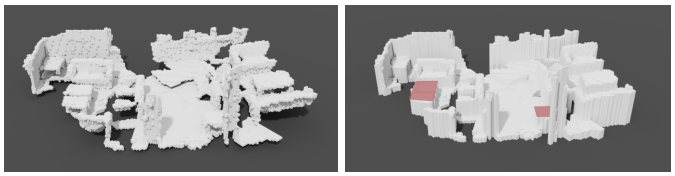
The rest of this paper is organized as follows. We first introduce related work and briefly discuss the characteristics and limitations of existing methods in Section 2. We give an overview of our virtual scene generation system in Section 3, and the pipeline of our approach is summarized in Figure 1. Followed by the technical details of the proposed approach in Section 3.1 to Section 3.3. Experimental setups and generation results are presented in Section 4. We evaluate the proposed system quantitatively and by user studies in Section 5. In Section 6, we summarize the proposed system, and discuss limitations and potential future work.

2 Related Work

2.1 3D Reconstruction and Plane Detection

3D reconstruction approaches aided with low-cost sensors have been very popular. Since the release of the Kinect, many research works [13], [14], [15], [16] have been conducted with this type of sensors for fast and stable reconstruction of indoor scenes. Shortly after that, Google's Project Tango also provides an opportunity to facilitate 3D reconstruction and applications [17] in virtual and augmented reality. In addition to this, stable reconstruction based solely on RGB images has been extensively studied [18]. The captured 3D scenes can be represented in different forms. To facilitate analysis of spatial occupancy, volumetric representations are widely used. Voxelization is a post-processing method for volume data that simplifies the representation of objects and scenes while maintaining a reasonable degree of accuracy. Many well-established works [19], [20], [21] have proposed different approaches to accelerate this process, both in terms of software and hardware. Our method also utilizes a volume-based representation and we adapt the voxelization algorithm to take into account our needs for collision avoidance.

Plane detection is widely mentioned and used in the study of 3D scenes. As one of the most commonly used methods, the RANSAC (Random Sample Consensus) method and its improvements [22], [23] excel in plane detection problems. The Hough transform [24], which is used heavily in shape detection, can also be applied to this task. More recently, a number of data-driven approaches [25], [26] have significantly increased the upper bound on the effectiveness of planar detection.



(a) (b)

Fig. 2. An example of voxelization of a scene. (a) General voxelization representation for the scene, (b) Tiny-Voxelization representation for the scene, detected interactive planes are highlighted in red.

2.2 Supervised Topic Models

In this paper, we expect to generate context-consistent scenes. To achieve this, the latent associations between scene content and scene type are essential in the generation process. Supervised topic models provide a powerful tool for latent data discovery. Therefore, we treat the virtual scene dataset as a corpus consisting of multi-label documents. And we utilize a multi-label topic model to extract the relationships between object categories and room types, which facilitate the selection of objects that fit the context of the room. Supervised Latent Dirichlet Allocation (SLDA) [27] and its variations are the most popular methods of supervised topic models. SLDA assumes that the labels are generated based on a mixture of empirical topics for each document, and limits each document to have only one label. A similar work named Discriminatively Trained LDA (DiscLDA) [28] cannot handle the case of multi-label documents as well. Alternatively, Multi-Multinomial LDA (MM-LDA) models each document as a bag of words with a bag of labels [29], and derives the topic for each observation from a shared topic distribution. MM-LDA is not constrained to single-label documents, but the learned

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2021.3111729, IEEE Transactions on Visualization and Computer Graphics

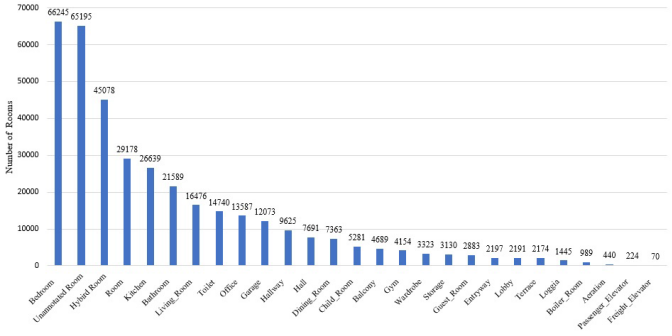


Fig. 3. Room type statistics. We show the types of rooms contained in the virtual scene dataset and their corresponding numbers of rooms.

topics cannot directly correspond to the labels in the label set. Another multi-label document topic extraction model is Labeled LDA (L-LDA) [30], which presents a solution to the credit attribution problem [27]. We utilize L-LDA in this paper to obtain latent relationships, which are the association relationships between the specific scene type and objects appearing in the scene of this type. We define the strength of a latent relationship by the probability of the object appearing in the scene of such a type.

2.3 Obstacle Awareness in Virtual Reality

Collision with real-world objects is problematic for users in experiencing virtual reality. Many methods have been developed to solve the problem. The most direct obstacle awareness method is based on non-visual feedback [4]. The authors embed vibration tactile actuators into the HMD to provide alerts when users approach obstacles. However, such external signal interference has a great impact on the sense of presence.

Alternatively, several methods try to display three-dimensional representations of the real world in the virtual environment. In [5], 3D point cloud data of the real-world objects is displayed in the virtual world. Users can recognize the information of virtual world and real world simultaneously through the point cloud display. In this method, to reduce the loss of presence, the authors filter the point cloud to only keep the part that is sufficient to guarantee the understanding of the real scene. Nevertheless, the presence of a point cloud out of the context of the virtual environment can still reduce the sense of presence for users. Other visualizations such as occupancy maps and glass walls can also negatively affect the feeling of presence [31], [32]. In addition, there are similar methods that incorporate visual indicators such as wireframes into the virtual environment to show real-world obstacles [7], [8]. Another recent approach uses the HTC Vive Pro to obtain the color and stereo information of real-world objects which are embedded into a virtual environment [33]. However the performance of this system is limited by latency and low resolution.

Some approaches rebuild the real scene into the virtual environment to keep the high level presence of users. In [34], authors proposed an idea that generates a virtual aligned copy of the real scene inside the virtual environment. The virtual copy is generated by detecting the 3D orientation and position of the mannequin. Another approach is described in [10]. An entire virtual environment is rebuilt based on the real scene where the VR system is set up. By detecting the walkable area and specific interactive objects, this approach is guaranteed to avoid collisions and supports interactions with certain virtual objects, but it is limited by the rigid virtual scene generation rules.

Valve[1] designs the Chaperone system based on their VR platform including SteamVR[2] and HTC Vive[3] for real-world obstacle awareness. When the system is set up, it keeps tracking the location of the user in the tracked area, and warns the user when he/she approaches an obstacle that cannot be seen because of the headset. While there are settings to help Chaperone to be less intrusive in a virtual environment, it still severely undermines the sense of presence [4]. In the virtual reality experience, users can choose different modalities to see the boundaries of the tracked area or the real objects they are approaching. However, both solutions produce signals that are completely out of context. [12] proposed an obstacle awareness method which replaces the real-world obstacles by virtual obstacles (such as stones, rocks and hills). Although this method can help users avoid the obstacles in the real world, its monotonous virtual substitutes usually do not conform to the context of the virtual scene, thus affecting the sense of presence. Inspired by this work, our generation pipeline adopts a similar voxelization method for input RGB-D scans. However a new combinatorial optimization method is developed in the virtual object substitution process, which not only allows users to avoid obstacles, but also maintains a sense of presence in the virtual scene. [35] and [36] align CAD models to the corresponding parts of the scene scan, and the aligned scene can avoid collisions without losing immersion of users. However, the geometric constraints of both methods are too strong: the retrieved CAD models are very limited, which makes the scene generated by the replacement have little diversity.

In this paper, we design an automatic virtual scene generation system to overcome the limitations of existing methods. The system can not only preserve the sense of presence of users, while avoiding collisions with real-world objects, but also allow users to interact with specific virtual objects. Varied scenes are generated according to the context of the virtual environment and thus can maintain a high degree of immersion.

Fig. 4. (a)The occurrence probabilities calculated by our topic model of object categories in three example types of rooms (larger font size indicates higher probability), (b) An example library of object categories for generating a bedroom virtual scene.

1. https://www.valvesoftware.com/
2. https://store.steampowered.com/steamvr
3. https://www.vive.com/us/

3 System Overview

In this section, we describe the workflow of our virtual content generation pipeline for creating an interactive virtual scene containing content with consistent context, which means the objects appearing in the room are plausible and the placement of objects is in line with everyday patterns. The main idea is to use the geometric information of the physical world as constraints, select reasonable models from the virtual object model library to fill in the restricted areas, and generate the virtual environment. The top right corner of Figure 1a shows a virtual scene generated by our system, with tables and cabinets in the red boxes as interactive virtual objects.

Our system substitutes obstacles in the physical world with context-consistent virtual objects. Different from existing methods, we use a topic model and a template extraction method [37] to extract the association relationships and layout patterns of virtual objects from virtual scene datasets. Specifically, we utilize two virtual indoor datasets, namely Strucutred3D [41] and 3D-FRONT[4] [42]. They both provide photo-realistic indoor scene data based on room designs by professional designers. With these indoor design datasets, the selection of virtual objects for replacement in our work is not limited to some specific obstacle types [12], and the generation of virtual scenes does not need to rely on complex rules [10]. As shown in Figure 1b, the workflow of our system includes: (i) reconstructing the 3D scene of the HMD user's surrounding environment and detecting the objects in the active area, (ii) detecting the interactive surfaces in the scene and identifying the areas occupied by different objects, (iii) extracting the association relations and layout patterns between different objects from the virtual scene dataset, and (iv) substituting the real scene with virtual objects under the constraints of geometry and contextual consistency.

3.1 3D Scene Rebuilding and Plane Detection

Our system begins with a 3D representation of the real-world environment. In practice, a wide range of methods can be used to obtain 3D point cloud data from RGBD or RGB sensors [13], [43]. In our implementation, we select some 3D scenes as point clouds from the ScanNet [44] dataset as experimental environments.

3.1.1 Voxelization

In the following two subsections, we will describe how to convert point cloud data to individual objects with a simple representation. At this stage of the workflow, the 3D representation of the room is in the form of a point cloud. In order to reduce the complexity of calculations and representations, we perform voxelization on the points.

Before voxelization, we need to make corrections to the horizontal orientations of the scene, which is equivalent to floor detection, so that all objects are placed approximately horizontally. In our implementation, we use Hough transform [24] for floor detection. Let $S = \{s_i(x_i, y_i, z_i)\}_{i=1}^{N}$ denote the positions of all the points in the point cloud data and N is the number of points. In order to make the parameter space of planes bounded, we use the polar coordinate parameter equations to represent each plane. In polar coordinates, a plane can be uniquely represented by a triplet (θ, φ, r), with the corresponding plane equation as follows:

$$x \sin\theta \cos\varphi + y \sin\theta \sin\varphi + z \cos\theta = r \tag{1}$$

where $\theta \in [0, \frac{\pi}{2}], \varphi \in [-\pi, \pi], r \in [-\max \|s_i\|_2, \max \|s_i\|_2]$, z-axis points up, and x and y are in the horizontal plane. Since the parameter space is bounded, which is a cuboid in $\mathbb{R}^3$, we can discretize the space into blocks. After performing 3D Hough transform on the point set S, we select the parameter triplet $(\theta_v, \varphi_v, r_v)$ corresponding to the most voted block, i.e. the block with the highest Hough value, as the floor. In more detail, since θ here indicates how much the plane slopes, and the floor is usually close to horizontal, we can limit it to a much smaller range, e.g. $[0, \frac{\pi}{8}]$. This will reduce the calculation time by a large amount. Upon detecting the floor, the overall point cloud will be adjusted to make the floor parallel to the x-y plane, so that all the obstacles are placed horizontally.

In the general process of voxelization, a cube is slid along the axis orientations through the point cloud and instantiates a copy every time it collides with one or more points. The quality of voxelization depends on the side length of the sliding cube.

In our workflow, we use the same method as Tiny-Voxelization in [12] (see Figure 2 for an example). Each time the cube collides with the point cloud, we instantiate a pillar from the position of collision to the ground, which is different from the standard voxelization method, in which a single cube is instantiated. In our implementation, the bottom of each pillar is a square of 0.05m side length. This method reduces the accuracy of voxelization, as it fills some of the holes in the scene. However, it also reduces scene complexity and eliminates the impact of holes on interactive surface detection and scene filling. It is thus more suitable for our purpose of object replacement and collision avoidance.

3.1.2 Clustering

Since we consider the relative position relationships between objects and the virtual object layout patterns, we need to cluster the pillars from the previous step into individual objects. We consider each pillar as a whole, which is represented by its topmost point, rather than independent cubes. This greatly simplifies the processing and is sufficient for our purpose.

The clustering method performed is based on the distance. We define the distance between two pillars as the 3D Euclidean distance between their topmost points. In our implementation, we use the DBSCAN [45] clustering method, which is designed to fit the non-convex data structure. All the clusters must contain a minimum number of pillars (set as 50 pillars). In addition, in order to exclude the floor and walls in subsequent stages, only clusters with heights between 0.2m and 2m are retained; these parameters are determined by the height and volume of some common objects. We have tried several parameter settings and selected the setting with the most reasonable results. Since when the parameters change in a reasonable range, the results are not very different, the setting does not need to be adjusted for different datasets and we do not show comparative results with different settings. Let $C = \{L_i\}_{i=1}^{N_C}$ denote the clustered objects, and N_C denote the number of objects. The object L_i consists of a group of pillars $\{h_{ij}(x_{ij}, y_{ij}, z_{ij})\}_{j=1}^{K_i}$, where K_i is the number of pillars and each pillar is denoted by its topmost cube.

4. https://tianchi.aliyun.com/specials/promotion/alibaba-3d-scene-dataset

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2021.3111729, IEEE Transactions on Visualization and Computer Graphics

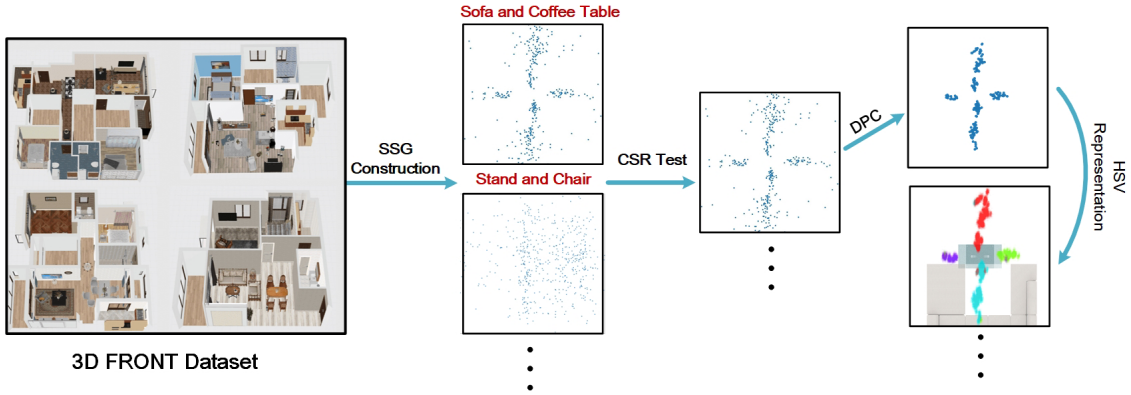


Fig. 5. Procedures of the layout pattern extraction pipeline proposed in [37]. We first learn a specific spatial strength graph model [37] to obtain the relative position between two given furniture objects (blue points in the second column), then we utilize tests for complete spatial randomness (CSR) [38] to get filtered patterns (third column). At last, we utilize density peak clustering (DPC) [39] to refine the layout patterns and use the system of hue, saturation and value (HSV) to visualize the refined patterns [40].

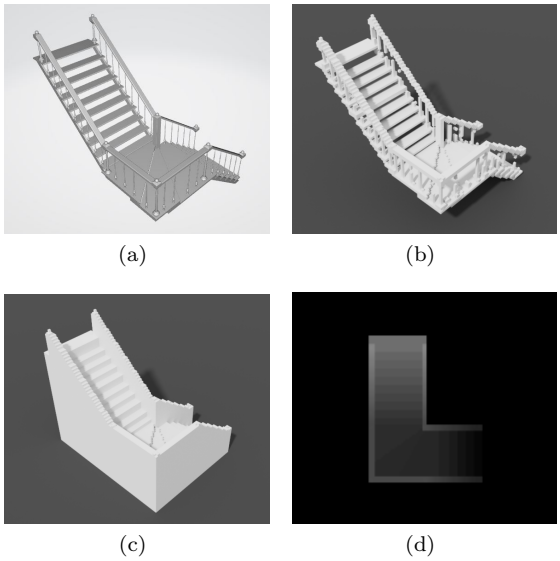


Fig. 6. An example of pillar-based representation and height map. (a) A model of a staircase, (b) General voxelization representation, (c) Tiny-Voxelization representation, (d) Height map of the model.

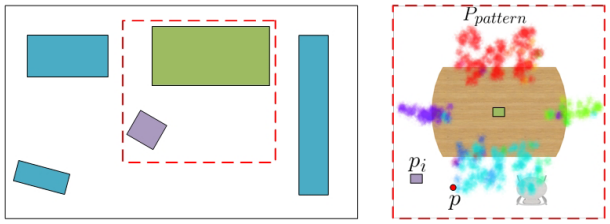


Fig. 7. Similarity of patterns. The layout of two objects that make up the pattern in the scene (in the red dashed box on the left), and the layout in the pattern library (in the red dashed box on the right).

3.1.3 Interactive Plane Detection

After obtaining the cubical representations of 3D objects in the scene, we detect the interactive planes on the objects. Typical interactive planes include the tops of tables and chairs. So we limit the direction of the interactive planes to near horizontal.

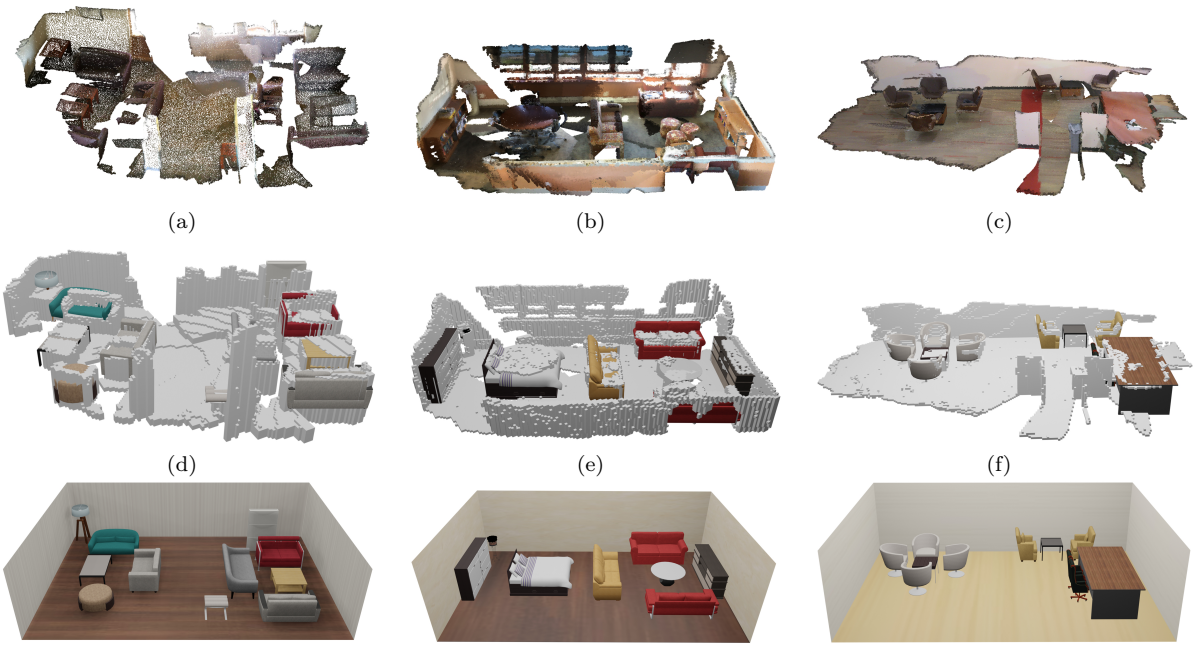
For efficiency, we reuse the Hough transform results from Section 3.1.1 and select the parameter triplets with high Hough values (threshold set as $\frac{N}{100}$) as the detected planes in the room, denoted by $W = \{z_k\}_{k=1}^{N_W}$ where N_W is the number of planes. Some of the planes in W correspond to the interactive surfaces of some objects. In order to identify whether plane z_k is indeed voted by some object's interactive surface, we make intersection tests between the objects in C and the planes in W. For an object L_i with K_i pillars and a plane z_k, if z_k intersects most of the topmost cubes of pillars of L_i, we consider L_i's top surface to be interactive. Mathematically, we compare the plane's height and the pillar's height at each (x_i, y_i). After rewriting z_k as a function of z w.r.t. (x,y) in the voxelized coordinates, the number of intersected topmost cubes will be $T_i = \sum_{j=1}^{K_i} t_j$ where

$$t_j = \begin{cases} 1, & 0 < z_{ij} - z_k(x_{ij}, y_{ij}) < 1 \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

The top surface of L_i is then marked as interactive if T_i exceeds a high percentage (set as 50% in our experiments) of K_i. As illustrated in Figure 8a, two detected interactive planes are marked in red in the original point cloud of the scene. Figure 2 shows the two interactive planes marked in the tiny-voxelization representation for the scene. According to the following workflow, we will select interactive virtual models for such objects in order to maintain the semantic consistency of the interactivity during virtual object substitution. Figure 8b shows two interactive planes marked in the generated virtual scene.

Fig. 8. Two interactive planes marked in red in the original point cloud of the scene (a) and the generated virtual scene (b).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2021.3111729, IEEE Transactions on Visualization and Computer Graphics



(a) (b) (c) (d) (e) (f) (g) Living room (h) Bedroom (i) Office

Fig. 9. The proposed method applied to real-world scenes. The point clouds of the real environments (first row), the matching of tiny-voxelization and substituted virtual objects (second row). Generated virtual scenes with added walls and floor (bottom row): (g) living room, (h) bedroom and (i) office.

3.2 Contextual Relation and Layout Pattern Extraction

To preserve a high level of presence in the virtual scene, the objects used to construct the virtual scene should match the theme of the scene, and the objects should have coherent placements in line with human intuition. For example, when the user is in a kitchen, if there is a bathtub or a little hill next to the stove, these objects are out of the context of the scene and will spoil the user's sense of presence [31], [32]. Therefore, we extract associations between objects and layout patterns from the virtual scene dataset to guide our system to generate a reasonable virtual scene.

3.2.1 Extraction of Association Relationships

We utilize two virtual indoor scene datasets, namely Structured3D [41] and 3D-FRONT [42] to extract the correlations and layout patterns between virtual objects. We select 45,622 designed house layouts with a total of 303,474 rooms from those datasets. Figure 3 shows the statistics of room types.

We propose an association relationship extraction method based on the topic model. Treating the house layouts as a corpus. Each room r is represented by a tuple consisting of a list of object category indices $O^{(r)} = (o_1, \cdots, o_{N_r})$ and a list of binary room type indicators $\Lambda^{(r)} = (l_1, \cdots, l_U)$ where each $o_i \in \{1, \cdots, V\}$ and $l_i \in \{0, 1\}$. Here N_r is the number of object categories in the room, V is the number of all object categories and U is the total number of unique room types in the datasets. Following the process in [30], we draw a multinomial mixture distribution θ^r over all U room types for room r from a Dirichlet prior α, and a multinomial room type distribution over object category library O for each room type l_u, from a Dirichlet prior β. The constraint is that the room type prior $\alpha^{(r)}$ is restricted to the set of annotated room types R_T^r. The joint probability of object and room type $p(O, R_T|\alpha, \beta)$ can be learned from the training set. Figure 4a illustrates two examples of $p(O|R_T)$ in word clouds, in which the occurrence probability of an instance category is represented by the size of the corresponding name tag.

In order to implement the interactivity of the generated scene, we reorder the categories in the model library according to interactivity. Figure 4b shows an example library of object categories we used to generate a bedroom virtual scene using the combinatorial optimizations described in Section 3.3. In the simulated annealing process, each new object combination is sampled from the library according to the occurrence probability, and the objects are sampled from the corresponding library according to their characteristic (Normal or Interactive), such that the interactive objects are sampled from the interaction library.

3.2.2 Extraction of Layout Patterns

Layout patterns are a priori knowledge of how we arrange objects in real life. By incorporating relative transformations, patterns can naturally avoid irrational situations, such as collisions. In this paper, we utilize the 3D-FRONT [42] dataset which contains 175 object categories and 9317 objects to extract the layout patterns between objects. We follow the prior learning pipeline in [37] as shown in Figure 5.

We first learn a specific spatial strength graph model indicating how objects are spatially related with each other. In this graph, vertices represent objects, and edges are associated with weights to encode the spatial strengths between objects [37]. The blue points in the second column of Figure 5 represent the relative positions between two given furniture objects. Axes are aligned to walls, and coffee table/chair is centered. In the top subgraph, the sofa and the coffee table are obviously spatially related, while there is no obvious spatial relation between the nightstand and the chair. To measure

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2021.3111729, IEEE Transactions on Visualization and Computer Graphics

how obvious certain patterns exist in a set of points, we utilize tests for complete spatial randomness (CSR) [38] to measure the strength of spatial relations between objects. The third column in Figure 5 shows the discrete priors which passed the CSR test. Then we utilize density peak clustering (DPC) [39] to refine the layout patterns. The fourth column bottom subgraph in Figure 5 shows the fine layout patterns. Similar to the visualization of dense optical flows [40], we apply the system of hue, saturation and value (HSV) to represent orientations as proposed in [37], where angles are normalized within $(0, 2\pi)$ as hue, probability densities are represented as saturation, and values are all set to 1. In this paper, we extract a total of 355 layout patterns. Some of them are displayed in the supplementary materials.

Fig. 10. A virtual living room enriched with associations and layout patterns.

3.3 Virtual Scene Substitution Generation

The final step of our system is to select appropriate virtual objects to replace the clusters rebuilt in Section 3.1.2 according to the context of the prebuilt virtual environment. In order to satisfy the geometric constraints of the real scene and the context consistency of virtual scene simultaneously, we cast the substituting process as a combinatorial optimization. Our optimization objective function is as follows:

$$E = E_{geometry} + \lambda E_{pattern} \tag{3}$$

which consists of two energy terms: geometric size constraints from the real scene $E_{geometry}$ and constraints of layout patterns between objects $E_{pattern}$. λ is the corresponding weight to balance the energy terms. We set λ to 0.5 in implementation.

3.3.1 Geometric Constraint

In order to fit the virtual objects to the physical objects in the scene, we have to measure the geometric similarity between them. We have already voxelized the scene and the objects within it to simplify the representations and computation. So similar voxelization processes are applied to virtual objects with the same resolution (0.05m per cube) as described in Section 3.1.1.

After voxelizing both physical and virtual objects, since both of them are represented as pillars at regular grids, we can compactly represent them by converting the pillar representation of each object into a grayscale image representation, called a height map (see Figure 6 for an example). Looking down on each object from the positive direction of the z-axis, each pillar corresponds to a pixel at the same location in the height map, and the height of it corresponds to the grayscale value of the pixel. After making such a conversion, all the geometric similarity measurements are cast as the image similarity measurements as follows:

$$\chi(\hat{U}, \hat{V}) = \sum_{i=1}^{m} \sum_{j=1}^{n} \sqrt{(\hat{U}_{i,j} - \hat{V}_{i,j})^2} \tag{4}$$

where $\chi(\cdot)$ defines the degree of similarity between the two height maps. (m, n) is the size of the picture, $\hat{U}$, $\hat{V}$ are two height maps of the same size, $\hat{U}_{i,j}$ and $\hat{V}_{i,j}$ are the grayscale values of corresponding pixel (i, j). In addition, we can easily achieve the geometric matching between physical and virtual objects when the axes are not aligned, e.g. matching after rotating the virtual object by a specific angle since rotating an image is easy and efficient.

For each rebuilt cluster, the replaced virtual object should have a similar geometry to it to ensure there is no collision between the user and real-world objects during the virtual reality experience. $E_{geometry}$ is formulated as follows

$$E_{geometry} = \sum_{1}^{N_C} D_i \tag{5}$$

where N_C is the number of clusters, D_i is the degree of geometric difference between the ith cluster and the corresponding substituted object.

$$D_i = \min_{\forall k \in K_{Model}} \chi(c_i, k) \tag{6}$$

where c_i is the height map of the ith cluster.

In terms of the placement of virtual objects, their centers are set to the bounding box centers of clusters. And to cope with rotation, the height map of each object is rotated around the center at intervals of 10° to generate 36 copies of different rotation angles, available in K_{Model}. k is one of the height maps of substituted model. The size of the models in the library we used for substitution are consistent with the actual size. There is no size scaling during the optimization process. However, during the substitution process of the models with interaction planes, small scaling adjustments are performed to align with the interaction planes.

3.3.2 Layout and Context Consistency Constraint

$E_{pattern}$ is used to assess the reasonableness of the virtual scene layout generated through object substitution. As shown in Figure 7, in the left black wireframe is the top view of the clusters replaced by virtual objects. In the red dashed box on the left are two objects between which the pattern exists. We compute their relative position in the horizontal orientation and compare it with the possible relative positions of similar objects in the virtual datasets we utilize. The pattern matching rate is formulated as follows,

$$M_i = \min_{\forall p \in P_{pattern}} \|p_i - p\|_2 \tag{7}$$

where p is a layout point in the pattern space $P_{pattern}$, and p_i is the relative spatial position of the ith pair of clusters. It is important to note that here we only consider the relative spatial relationship between every pair of object categories. The energy term $E_{pattern}$ is calculated as

$$E_{pattern} = \begin{cases} \sum_{i}^{N_p} M_i & N_p \neq 0 \\ 0 & N_p = 0 \end{cases} \tag{8}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2021.3111729, IEEE Transactions on Visualization and Computer Graphics

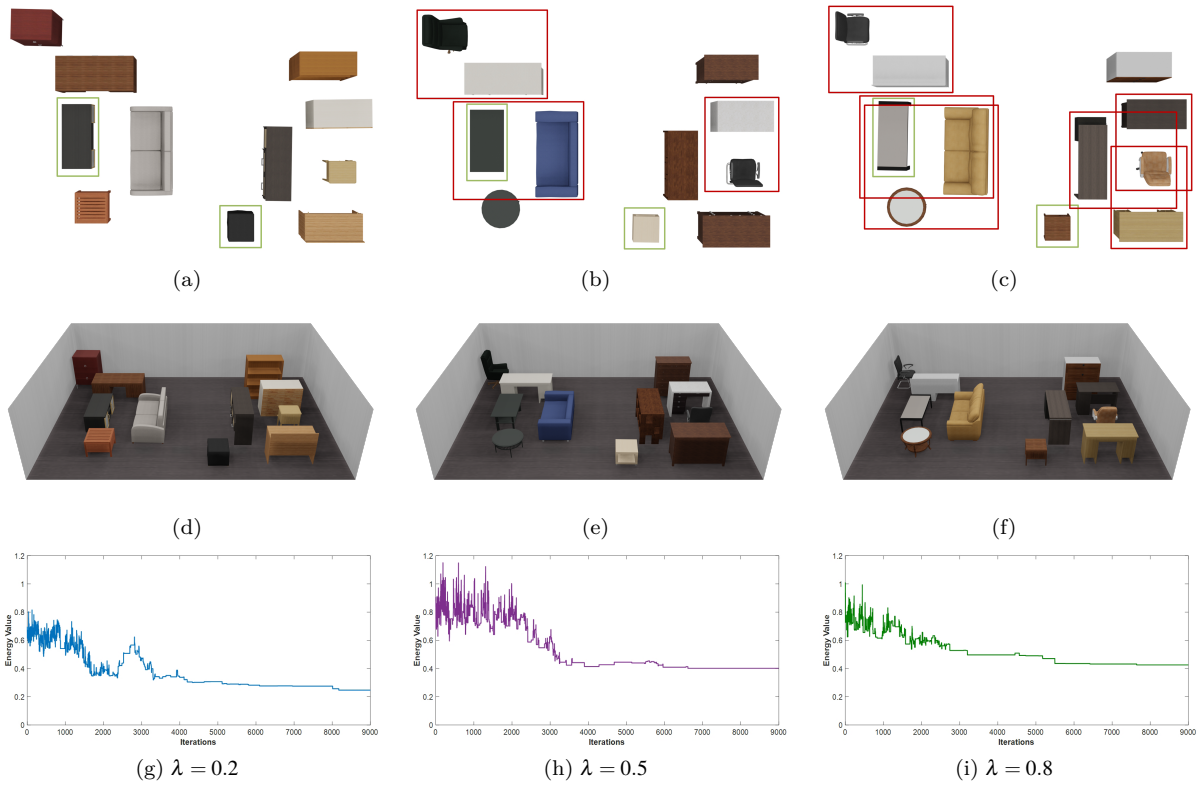


Fig. 11. The effect of the energy term of layout pattern constraint $E_{pattern}$.

where N_p is the number of patterns that exist in the scene.

To minimize the energy function in Equation 3, which is a combinatorial optimization problem, we use simulated annealing [46] to find the best virtual object substituting combination C that minimizes the cost function in Equation 9. In a typical simulated annealing process, the initial C_0 is generated by sampling from the category library of corresponding theme as described in Section 3.2.1. Then in the ith iteration, we sample a new combination $C_i^{'}$, and $C_i^{'}$ is accepted as C_{i+1} with probability

$$P_{C_i^{'} \to C_{i+1}} = \min\left[1, \exp\left(-\frac{E(C_i^{'}) - E(C_i)}{T_0 - \sigma_T \cdot i}\right)\right] \tag{9}$$

where T_0 is the initial temperature, i is the number of current iteration and σ_T is the temperature drop at each iteration. As the temperature drops, the probability of accepting a worse solution becomes lower. In this paper, we set $T_0 = -\log(1e-5)$ and $\sigma_T = T_0/20N_C^2$, the maximum number of iterations is 9000.

4 Experiments

4.1 Implementation Details

We perform the 3D real-scene rebuilding, association relationships and layout pattern extraction and virtual environment generation process on a Windows 10 computer with an Intel Core i7-6700HK @2.7GHz 2.71GHz processor and 16GB RAM. We used the Unity game engine to generate the virtual scene, the textures, materials, and other 3D elements of models are obtained from the virtual scene dataset. In the voxelization stage, we set the cube's edge length to 0.05m. The hyper parameters for DBSCAN in clustering is set as eps = 3.5 and minPts = 10. The time to reconstruct the real scene varies from 1 to 7 minutes, depending on the fineness of the input point cloud of real scene. The preprocessing includes the extraction of association relationships and layout patterns, which takes 6-10 minutes. Our current code is not parallelized or running on the GPU, so we believe that with some engineering optimizations, the running time will be further reduced.

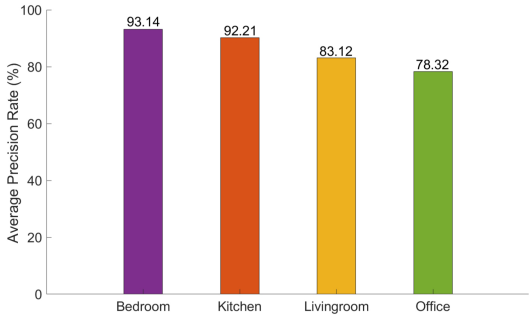


Fig. 12. Evaluation of generation accuracy in terms of themes using subject judgment: for each theme, we show the percentage that the theme selected by the subjects the same as the generated.

4.2 Virtual Scene Generation

We tested all steps described in the previous section including 3D rebuilding, voxelization, clustering and virtual object substituting in multiple real environments. Figure 9 shows three virtual scenes in different contexts generated by different real scene from the ScanNet [44] dataset. In all scenes under

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2021.3111729, IEEE Transactions on Visualization and Computer Graphics

consideration, our method is capable of building a virtual reality scene, selecting appropriate virtual objects to replace the obstacles in the real scene. More generated scene results are shown in the supplementary materials. Considering the indoor scene dataset we used, we manually add walls and floor structure to each generated scene for the sake of aesthetics. The method presented in this paper cannot generate wall structures by itself.

The scene we generated takes into account the context consistency and the plausibility of the object placement in the scene. The optimized virtual object combination is placed according to the layout patterns. The first row of Figure 9 is input point cloud from ScanNet. The second row shows the matching of tiny-voxelization and substituted virtual objects. Due to the geometric constraints of the real scene, the geometry of the selected virtual object matches the replaced voxels as much as possible. Although some layouts will unavoidably deviate from the pattern to some extent, all are within the acceptable range. The bottom row in Figure 9 shows the virtual scenes generated by our system. Due to the geometric constraints of the real scene, we only add the ground and wall to the optimized object combination. When needed, more real virtual scenes can be generated on the premise of ensuring obstacle awareness and interactivity by relaxing geometric constraints. Figure 10 shows a living room enriched with associations and layout patterns.

(a) (b)

Fig. 13. Distribution of subjects' judgment of room type for the sample scene. (a) A sample living room, (b) the pie chart showing the distribution of user judgment for the room type.

5 Evaluation

5.1 Qualitative Evaluation

Figure 11 evaluates the effect of the energy term of layout pattern constraint (Equation 8). We test 11 values of λ with an interval of 0.1 from 0 to 1, each test generates five virtual scenes. Figure 11 shows three representative test results. The first row shows the existing layout patterns in the generated scenes, each red wireframe represents a layout pattern, the second row gives the generated scenes, and the third row shows the convergence curve of the energy function (Equation 3). As shown in Figure 11 first column, there are no layout patterns in the scene. Although the geometry of the replacement virtual objects is consistent with the geometric constraints (the convergent energy value is low), the layout of the furniture in the scene is not reasonable, such as an office desk in front of a sofa and a stand surrounded by three cabinets. As shown in Figure 11 third column, there are six layout patterns in the scene. Excessive layout patterns also lead to unreasonable furniture layouts, such as two coffee tables in front of a sofa and three office desks around an office chair. Thus it can be seen that too few or too many layout patterns can lead to unreasonable furniture layouts. According to our test, when λ is equal to about 0.5, the layouts of generated scenes are generally reasonable as shown in Figure 11 second column. The final experimental results presented in this paper are all generated when $\lambda = 0.5$, and the energy functions always converge in our experiments.

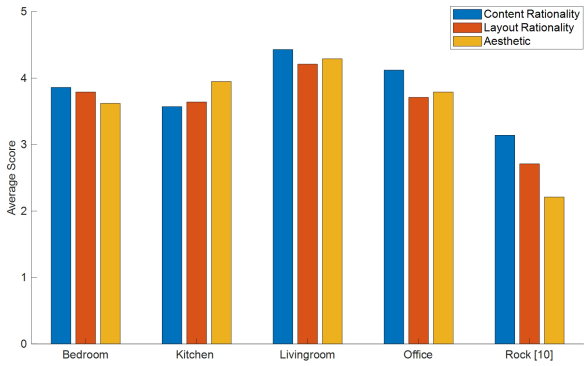


Fig. 14. The results of the average scores of virtual scenes generated by our method and [12] on content rationality, layout rationality and aesthetics.

To evaluate the context consistency of the content in the scene generated by our method, we invite 20 subjects to participate in the following user study. We generate four different themed virtual scenes for each input 3D scan. A total of 7 3D scan inputs are selected to generate 28 virtual scenes. Each generated scene is in the form of a rendering for the subjects to view. Subjects are asked to choose an appropriate theme for each virtual scene based on their general knowledge. Figure 12 shows the percentage that the theme selected by the subjects is the same as the generated for each theme. From the accuracy distribution, it can be seen that the accuracies of bedroom and kitchen are relatively high, because there are some exclusive types of furniture which provide strong clues for subjects to judge, such as beds and stoves. However, due to the high repetition rate of furniture types for living rooms and offices, subjects sometimes get confused when judging room types from the renderings. Figure 13b shows the percentage distribution of subjects' judgment of room type for the sample scene (Figure 13a). The scene is a living room, but 35% of the subjects thought it was an office.

To evaluate the performance of our method in the scene content rationality, layout rationality and aesthetics, we perform another user study. 20 subjects are invited to score 35 virtual scenes generated by [12] and our method. These scenes are also generated from the same input scans as in the previous user study. Figure 14 shows a comparison of the average scores for the four themes of scenes generated by our method (Bedroom, Kitchen, Living room, Office) and the scenes generated by [12] (Rock). Due to the rich content collocation and reasonable layout in the scene, the virtual scenes generated by our method are rated higher than the monotonous scenes generated by [12] in each score.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2021.3111729, IEEE Transactions on Visualization and Computer Graphics

TABLE 1
Comparison of geometric errors between the real scenes and the corresponding generated scenes by different methods.

Method	Method in [12]	Our Method			
Scene Theme	Rock	Bedroom	Living room	Kitchen	Office
Average Error (cm)	13.2	10.3	12.4	11.7	10.8

5.2 Geometric Accuracy

To evaluate the geometric similarity between the original scene represented by the point cloud and the generated virtual scene, we compute the geometric error as follows. For each cubic pillar instantiated from the point cloud, we find the highest point it contains. We compute the geometric error as the average distance between these points and their vertical projections at the generated scene.

Formally, let $\{q_i(x_{q_i}, y_{q_i}, z_{q_i})\}_{i=1}^{N_Q}$ be the highest points of all the pillars where N_Q denotes the number of pillars, and for an arbitrary 2D position (x_0, y_0), $S(x_0, y_0)$ is the height of the highest intersection of the generated scene and the line that goes through $(x_0, y_0, 0)$ and is parallel to the z-axis. Then the geometric error is

$$\varepsilon_{geometric} = \frac{1}{N_Q} \sum_{i=1}^{N_Q} |S(x_{q_i}, y_{q_i}) - z_{q_i}| \tag{10}$$

Table 1 shows the comparison results of geometric errors between the real scenes and corresponding generated scenes by different methods. We compare the average errors of stone theme scenes generated by [12] and the four indoor theme scenes (Bedroom, Living room, Kitchen and Office) generated by our method. Each theme contains 15 scenes with different geometric structures. Figure 15a shows the geometry of one of the scenes. The input 3D scans we used are basically from ScanNet, in which a large number of objects have the geometric structures of indoor furniture, which have a high matching degree with our model library, so the geometric errors are low in our method. However, the method of [12] could scale each virtual rock, so the geometric error is not much larger than our method.

Figure 15a shows the geometric structure of a sample input scene, and Figure 15b shows the geometric error between each virtual object in generated scenes in different themes and the original scene. Scenes in the first four themes are generated by our method, and the last one is generated by [12]. Both methods have large geometric matching errors in places with severe point-cloud absence (Model No.9 and Model No.11). However, for the interactive objects in the red square section shown in Figure 15a (Model No.1 and Model No.6), our method detects the interaction surfaces and makes individual matching, so our method has a smaller geometric errors on these objects compared with [12].

5.3 User Experience

We conducted a user experience experiment to assess the difference between the method we proposed and the state-of-the-art Chaperone method in terms of presence. We compared the Chaperone with the scene obtained combining the Tiny-Voxelization with the clustering by height approach [12] and our approach in terms of sense of presence.

5.3.1 Experience Environment

Prior to the user experimentation, it is necessary to align virtual scenes with the real world. Alignment is accomplished by rotation and translation in 3D. In Section 3.1.1, the floor has been detected so we align the detected floor with the ground in the HTC Vive. After aligning the floor, two extra pairs of points in the virtual and real world are needed to finish the alignment. Since the positions of the two base stations of the HTC Vive are easily obtained both in the VR system and the scanned scene, we use them as key points to align the whole scenes. The rotation is determined by aligning the normal direction of the ground and the direction of the line connecting the stations at the same time, while the translation is determined by aligning the virtual and real midpoints of the line connecting the two base stations.

We assume that users tend to use VR devices in a familiar environment. We conduct the experience experiment in a space that the subjects often use. Figure 16 shows the experimental scene. The subjects are assumed to know the rough three-dimensional structure and the target object (Vive Controller) in the real scene. In the experimental task, subjects roam freely in the green area in Figure 16. Note that the green area will not be displayed in the HMD. We use [12] and our method respectively to generate virtual scenes for this experiment. Figure 17 shows an example pair of virtual scenes.

5.3.2 Experience Task

We conduct the experimental tasks using each method by the following procedure:

1) Subjects start the VR roaming.
2) Subjects roam in the walkable area of the virtual scene, and go to pick up the Vive Controller on the interactive object.
3) Subjects continue to roam with the Vive Controller, then choose a location on the interactive plane of the virtual object and lay down the Controller.
4) Subjects roam back to the starting point.

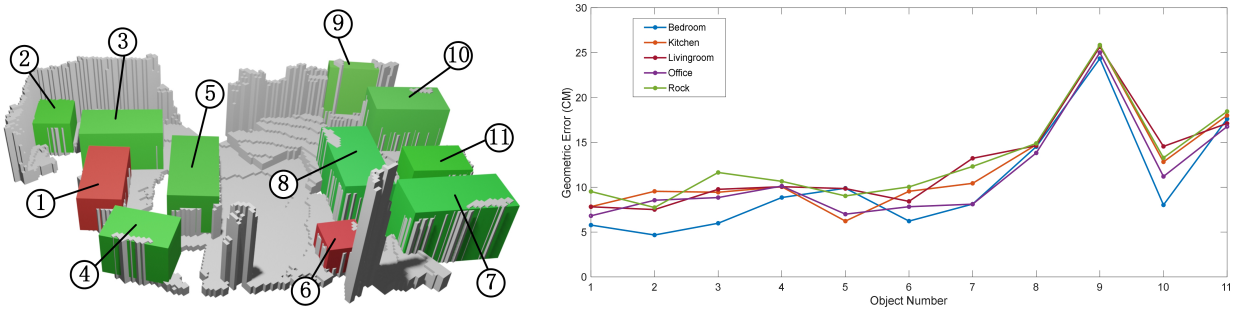
Our goal is that when the users are experiencing virtual reality, they can perceive the obstacles in real world without reducing the immersion of the virtual scene, and some interactive properties (support interaction currently considered in this paper) of real-world objects can be extended into the virtual scene.

5.3.3 Experience Evaluation

To measure the sense of presence, we use a standard questionnaire named Igroup Presence Questionnaire (IPQ[5]) [47], which is available for measuring presence in a virtual reality scene. There are 14 10-points Likert scale questions in the questionnaire to evaluate the three main aspects which are related to the sense of presence: the sense of geometric presence, the sense of physical existence in the virtual scene; involvement is regarded as both attention in the process of interaction and perceptual participation; sense of reality, the sense of real existence in VR experience. To evaluate perceiving of the real-world geometry information, a supplementary

5. http://www.igroup.org/pq/ipq/index.php

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2021.3111729, IEEE Transactions on Visualization and Computer Graphics



(a) (b)

Fig. 15. Geometric errors of an example scene. (a) The voxelization of the sample scene, (b) geometric error of each substituted virtual object.

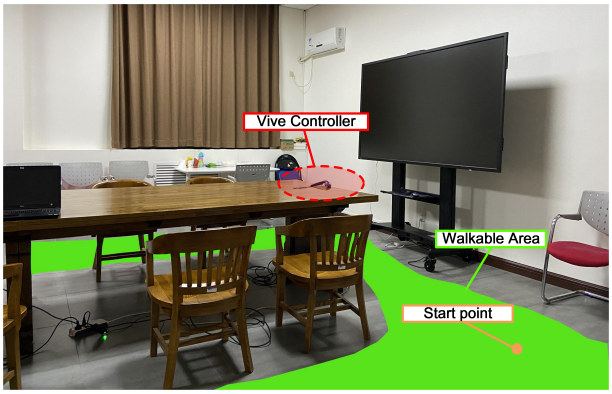


Fig. 16. Experience scene

questionnaire asking about the sense of distance to obstacles, the fear while walking in the virtual scene and the experience of interacting with virtual objects.

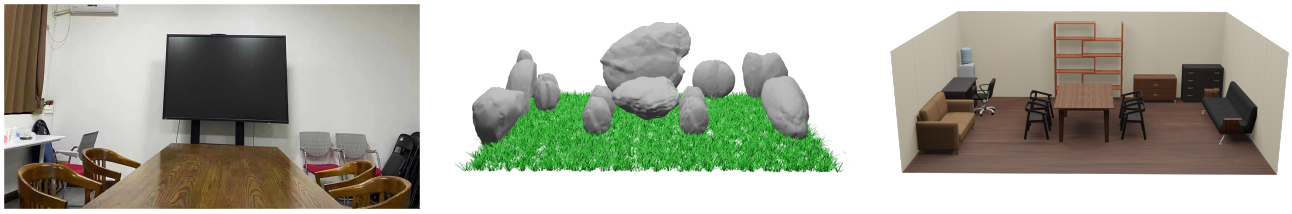
After virtual scene generation, we invited 13 users aged between twenty and twenty-nine with an average of twenty-four to participate in our experiment session. They all have normal vision and can perform the VR experience operations independently. Each subject is randomly selected to complete a three-part experiment. Participants are asked to explore in the generated virtual environment containing real-world obstacles and finish the experience task. Participants need to wander in the virtual environment for five minutes, and fill out the IPQ questionnaire and supplementary questionnaire after every experiment.

Figure 18 shows the average values of the three methods we compared in IPQ. After investigating the normality and homoscedasticity of each distribution of the obtained data, we use Friedman-test to examine whether each evaluation item is influenced by the scene generation methods, then we find significant differences for all the items ($p < 0.01$). After that, we conduct Wilcoxon signed rank test to examine how distributions of scores of the evaluation items are different. In Figure 18, pairwise connections with black dots indicate different levels of statistical significance ($p < 0.001$, $p < 0.01$ and $p < 0.05$). It can be seen that all pairwise differences are statistically significant at least at $p < 0.05$. In terms of the sense of presence into the virtual space, the Chaperone method reduces the sense of presence across all items in the IPQ compared to [12] and our method. In Chaperone, we believe that display of extra real-world structure information that is not needed for roaming and superimposed display that does not match the virtual scene result in significantly lowered user presence. [12] seems to meet the user's demand for presence to some extent, but according to the feedback of the questionnaire, this monotonous virtual obstacle display method greatly reduces the sense of reality of the scene. Using our method, the real-world obstacles are substituted with diversified scene contents and kept consistent with the theme of the entire virtual scene. Users could obtain richer visual information and comfortable sensation when roaming in the scene we generated than the simplistic obstacle replacement method. Users feel aware and confident of the position of real objects in our generated scenes.

Table 2 shows the results of the supplementary questionnaire. The values for the items are the average that subjects evaluate for each item at equal intervals of 7 scales (1-7) after each method. The first four items in the table are to evaluate the understanding of real-world information and the last three items are to evaluate the interaction of virtual objects. Values in brackets are the standard deviations. Focusing on the top four items in Table 1, it shows that subjects can understand real-world geometry information better in [12] and our method, compared with Chaperone. The average number of collision times of the subjects in the experiment is 4 times for Chaperone, and 0 times in [12] and our method. We believe that the reason for the high number for the Chaperone is that the subjects find it difficult to understand the distance to the obstacle because the Chaperone have no distance information. It is to be expected that by showing enough virtual objects, users will be able to understand the real space in more detail. According to these results, we can see that [12] and our method enable interaction with the real-world objects while walking without reducing immersion in a better manner than Chaperone. Moreover, our method is much better than [12] in terms of the virtual-real interaction surface matching and the realistic sense of interaction, as it detects the interactive surface and selects virtual objects with more similar geometric structures for scene replacement. As shown in Figure 19, representing a specific snapshot where a subject is told to "put the controller on the table". More details about the interaction can be found in the attached video.

Our preliminary results are promising and suggest that it is possible to help avoid obstacles while experiencing VR in a typical sized room. Normally, it is difficult to keep an unobstructed space exclusively for VR. If users try to play

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2021.3111729, IEEE Transactions on Visualization and Computer Graphics



(a) Real Scene (b) Generated by [12] (c) Generated by our method

Fig. 17. The input real scene and the virtual scenes generated by different methods. (a) Real Scene, (b) virtual scene generated by [12], (c) virtual scene generated by our method.

TABLE 2
Result of the supplementary questionnaire, we show the mean scores as well as standard deviations (in brackets).

Items	Vive Chaperone	Method in [12]	Our Method
Accurate perception of distance from obstacles	4.32 (1.32)	5.41 (1.62)	6.14 (1.03)
Understand the direction of view in real space	5.71 (1.75)	4.24 (1.12)	5.12 (1.35)
Wander in real space without fear	4.12 (1.33)	3.92 (1.47)	5.21 (1.13)
Can guess own position in real space	5.62 (1.08)	4.98 (1.25)	5.84 (1.41)
Can estimate the size of the interaction surface	-	3.24 (1.51)	5.88 (1.23)
Match accuracy between the virtual and real interaction surfaces	-	3.76 (1.17)	5.46 (1.09)
Interactive realism	-	5.41 (1.39)	6.52 (1.28)

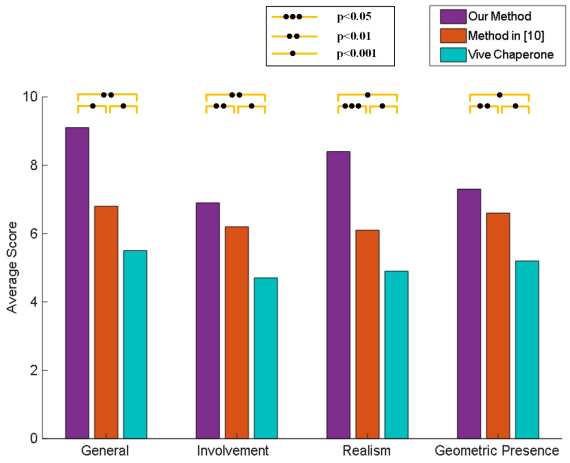


Fig. 18. Results of the IPQ questionnaire.

a room-scale VR with obstacles, our method can guarantee them to play safely without reducing immersion.

6 Conclusion and Future Work

In this paper, we described a novel VR generation system which uses the real scene as geometric constraints and allows interaction with specific virtual objects. The system generates a highly immersive environment by combining context-consistent scenes and interaction feedback. Our system allows ordinary users to quickly and easily create immersive and interactive virtual scenes that can be experienced in HMD. The generated virtual environment considers the existence of real obstacles, and the virtual objects added to the virtual environment have the same position and occupation as the real obstacles. In this way, users can avoid colliding with real objects and interact with them.

(a) Virtual View

(b) Real Scene

Fig. 19. A user is asked to interact with the table. (a) The interactive table in the virtual perspective, (b) Interact with a table in the real world.

We implemented a method that obtains the 3D structure data of the real scene through RGB-D device, and rebuilds it by voxelization and clustering, so as to obtain the 3D model of objects in the scene. Meanwhile, the association relations and layout patterns are extracted from the virtual scene dataset. Then, the 3D models are substituted with corresponding contextualized virtual objects through combinato-

rial optimization. This approach has been successfully applied to a number of real-world scenes with different structural complexity, showing the possibility of using it in any real indoor environment. An experimental user experience study shows that the system allows a higher level of presence than the state-of-the-art Chaperone technique. Thus, these results demonstrate that our system generates a secure space for interactions in a virtual environment by augmented virtuality.

The main limitation of the devised method is that it relies on datasets to extract the association relationships and layout patterns between different categories of virtual objects. In virtual reality, the virtual scenes we want to experience can be bizarre and even imaginary, such as outer space. Our method is not yet suitable for generating this rare fictional scene without corresponding datasets. Moreover, our system cannot acquire and process the real scene in real-time yet, but once the scene is reconstructed, the virtual object selection and replacement can be completed in real time. In terms of interaction, our system can only deal with scenes containing static obstacles at present, only supporting interaction is available, and it is also unable to perceive and substitute dynamic obstacles in the scenes. The current approach only considers a one-to-one relationship between clusters and virtual objects. It is perfectly possible to substitute a set of adjacent real objects with a large virtual one, but it is not yet possible with our method.

In addition to overcoming the limitations mentioned earlier, there are several aspects of the system that we can improve in future work:

- Adopting a more refined voxelization method to complement the geometry loss of the rebuilt real-world object in the vertical direction. Making the geometric features of the substituted virtual objects more consistent with the real objects.
- Extracting association relationships and layout patterns from the existing datasets surveyed in [48], and building more 3D virtual scene datasets with different themes (such as space, battleground, fantasy), allowing users to generate more diverse virtual scenes.
- Handling dynamic obstacles in dynamic scenes to achieve more complex interactions (such as moving, grabbing and twisting virtual objects).

Acknowledgments

This work was supported by the National Key Technology R&D Program (Project Number 2017YFB1002604), the National Natural Science Foundation of China (Project Numbers 61521002, 61772298), Research Grant of Beijing Higher Institution Engineering Research Center, and Tsinghua–Tencent Joint Laboratory for Internet Innovation Technology. We thank Prof. Sheng-Yong Chen from Tianjin University of Technology for his valuable comments in the conception of this work and the process of experiments which have greatly improved the manuscript.

References

[1] I. E. Sutherland, "A head-mounted three dimensional display," in Proceedings of the December 9-11, 1968, fall joint computer conference, part I, 1968, pp. 757–764.

[2] C. Cruz-Neira, D. J. Sandin, T. A. DeFanti, R. V. Kenyon, and J. C. Hart, "The cave: audio visual experience automatic virtual environment," Communications of the ACM, vol. 35, no. 6, pp. 64–73, 1992.

[3] M. Wang, X.-Q. Lyu, Y.-J. Li, and F.-L. Zhang, "Vr content creation and exploration with deep learning: A survey," Computational Visual Media, vol. 6, no. 1, pp. 3–28, 2020.

[4] D. Valkov and L. Linsen, "Vibro-tactile feedback for real-world awareness in immersive virtual environments," in 2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), March 2019, pp. 340–349.

[5] K. Kanamori, N. Sakata, T. Tominaga, Y. Hijikata, K. Harada, and K. Kiyokawa, "Obstacle avoidance method in real space for virtual reality immersion," in 2018 IEEE International Symposium on Mixed and Augmented Reality (ISMAR). IEEE, 2018, pp. 80–89.

[6] M. McGill, D. Boland, R. Murray-Smith, and S. Brewster, "A dose of reality: Overcoming usability challenges in vr head-mounted displays," in Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, 2015, pp. 2143–2152.

[7] M. Sousa, D. Mendes, and J. Jorge, "Safe walking in vr," in The 17th International Conference on Virtual-Reality Continuum and its Applications in Industry, 2019, pp. 1–2.

[8] F. Wu and E. S. Rosenberg, "Combining dynamic field of view modification with physical obstacle avoidance," in 2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR). IEEE, 2019, pp. 1882–1883.

[9] E. Girau, F. Mura, S. Bazurro, M. Casadio, M. Chirico, F. Solari, and M. Chessa, "A mixed reality system for the simulation of emergency and first-aid scenarios," in 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). IEEE, 2019, pp. 5690–5695.

[10] M. Sra, S. Garrido-Jurado, C. Schmandt, and P. Maes, "Procedurally generated virtual reality from 3d reconstructed physical space," in Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology, 2016, pp. 191–200.

[11] Y. Liang, F. Xu, S.-H. Zhang, Y.-K. Lai, and T. Mu, "Knowledge graph construction with structure and parameter learning for indoor scene design," Computational Visual Media, vol. 4, no. 2, p. 8, 2018.

[12] I. Valentini, G. Ballestin, C. Bassano, F. Solari, and M. Chessa, "Improving obstacle awareness to enhance interaction in virtual reality," in 2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR). IEEE, 2020, pp. 44–52.

[13] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison et al., "Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera," in Proceedings of the 24th annual ACM symposium on User interface software and technology, 2011, pp. 559–568.

[14] B. Peasley and S. Birchfield, "Real-time obstacle detection and avoidance in the presence of specular surfaces using an active 3d sensor," in 2013 IEEE Workshop on Robot Vision (WORV). IEEE, 2013, pp. 197–202.

[15] U. Qayyum, J. Kim et al., "Inertial-kinect fusion for outdoor 3d navigation," 2013.

[16] M.-H. Guo, J.-X. Cai, Z.-N. Liu, T.-J. Mu, R. R. Martin, and S.-M. Hu, "Pct: Point cloud transformer," Computational Visual Media, vol. 7, no. 2, pp. 187–199, 2021.

[17] A. Nassani, H. Bai, G. A. Lee, and M. Billinghurst, "Tag it!: AR annotation using wearable sensors," in SIGGRAPH Asia Mobile Graphics and Interactive Applications. ACM, 2015, pp. 12:1–12:4.

[18] J. Stühmer, S. Gumhold, and D. Cremers, "Real-time dense geometry from a handheld camera," in DAGM-Symposium, ser. Lecture Notes in Computer Science, vol. 6376. Springer, 2010, pp. 11–20.

[19] Z. Dong, W. Chen, H. Bao, H. Zhang, and Q. Peng, "Real-time voxelization for complex polygonal models," in PG. IEEE Computer Society, 2004, pp. 43–50.

[20] M. Schwarz and H. Seidel, "Fast parallel surface and solid voxelization on gpus," ACM Trans. Graph., vol. 29, no. 6, p. 179, 2010.

[21] S. Fang and H. Chen, "Hardware accelerated voxelization," Comput. Graph., vol. 24, no. 3, pp. 433–442, 2000.

[22] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," Commun. ACM, vol. 24, no. 6, pp. 381–395, 1981.
[23] L. Li, F. Yang, H. Zhu, D. Li, Y. Li, and L. Tang, "An improved RANSAC for 3d point cloud plane segmentation based on normal distribution transformation cells," Remote. Sens., vol. 9, no. 5, p. 433, 2017.
[24] R. O. Duda and P. E. Hart, "Use of the hough transformation to detect lines and curves in pictures," Commun. ACM, vol. 15, no. 1, pp. 11–15, 1972.
[25] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in CVPR. IEEE Computer Society, 2017, pp. 77–85.
[26] C. Liu, K. Kim, J. Gu, Y. Furukawa, and J. Kautz, "Planercnn: 3d plane detection and reconstruction from a single image," in CVPR. Computer Vision Foundation / IEEE, 2019, pp. 4450–4459.
[27] D. M. Blei and J. D. McAuliffe, "Supervised topic models," in NIPS. Curran Associates, Inc., 2007, pp. 121–128.
[28] S. Lacoste-Julien, F. Sha, and M. I. Jordan, "Disclda: Discriminative learning for dimensionality reduction and classification," in NIPS. Curran Associates, Inc., 2008, pp. 897–904.
[29] D. Ramage, P. Heymann, C. D. Manning, and H. Garcia-Molina, "Clustering the tagged web," in WSDM. ACM, 2009, pp. 54–63.
[30] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning, "Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora," in EMNLP. ACL, 2009, pp. 248–256.
[31] M. Keller and F. Exposito, "Game room map integration in virtual environments for free walking," in VR. IEEE Computer Society, 2018, pp. 763–764.
[32] M. Keller and T. Tchilinguirian, "Obstacles awareness methods from occupancy map for free walking in VR," in VR. IEEE, 2019, pp. 1012–1013.
[33] M. Rauter, C. Abseher, and M. Safar, "Augmenting virtual reality with near real world objects," in VR. IEEE, 2019, pp. 1134–1135.
[34] E. Girau, F. Mura, S. Bazurro, M. Casadio, M. Chirico, F. Solari, and M. Chessa, "A mixed reality system for the simulation of emergency and first-aid scenarios," in EMBC. IEEE, 2019, pp. 5690–5695.
[35] A. Avetisyan, T. Khanova, C. Choy, D. Dash, A. Dai, and M. Nießner, "Scenecad: Predicting object alignments and layouts in rgb-d scans," ArXiv, vol. abs/2003.12622, 2020.
[36] A. Avetisyan, M. Dahnert, A. Dai, M. Savva, A. Chang, and M. NieBner, "Scan2cad: Learning cad model alignment in rgb-d scans," 06 2019, pp. 2609–2618.
[37] S. Zhang, S. Zhang, W. Xie, C. Luo, and H. Fu, "Fast 3d indoor scene synthesis with discrete and exact layout pattern extraction," CoRR, vol. abs/2002.00328, 2020.
[38] M. A. García-Pérez, "Parameter estimation and goodness-of-fit testing in multinomial models," British Journal of Mathematical and Statistical Psychology, vol. 47, no. 2, pp. 247–282, 1994. [Online]. Available: https://bpspsychub.onlinelibrary.wiley.com/doi/abs/10.1111/j.2044-8317.1994.tb01037.x
[39] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," Science, vol. 344, no. 6191, pp. 1492–1496, 2014. [Online]. Available: https://science.sciencemag.org/content/344/6191/1492
[40] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," in SCIA, ser. Lecture Notes in Computer Science, vol. 2749. Springer, 2003, pp. 363–370.
[41] J. Zheng, J. Zhang, J. Li, R. Tang, S. Gao, and Z. Zhou, "Structured3d: A large photo-realistic dataset for structured 3d modeling," in ECCV (9), ser. Lecture Notes in Computer Science, vol. 12354. Springer, 2020, pp. 519–535.
[42] H. Fu, B. Cai, L. Gao, L. Zhang, C. Li, Z. Xun, C. Sun, Y. Fei, Y. Zheng, Y. Li et al., "3d-front: 3d furnished rooms with layouts and semantics," arXiv preprint arXiv:2011.09127, 2020.
[43] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, "RGB-D mapping: Using kinect-style depth cameras for dense 3d modeling of indoor environments," Int. J. Robotics Res., vol. 31, no. 5, pp. 647–663, 2012.
[44] A. Dai, A. X. Chang, M. Savva, M. Halber, T. A. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in CVPR. IEEE Computer Society, 2017, pp. 2432–2443.
[45] M. Ester, H. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in KDD. AAAI Press, 1996, pp. 226–231.
[46] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing." Science, vol. 220, no. 4598, pp. 671–680, 1983.
[47] H. Regenbrecht and T. Schubert, "Real and illusory interactions enhance presence in virtual environments," Presence Teleoperators Virtual Environ., vol. 11, no. 4, pp. 425–434, 2002.
[48] Y.-P. Xiao, Y.-K. Lai, F.-L. Zhang, C. Li, and L. Gao, "A survey on deep geometry learning: From a representation perspective," Computational Visual Media, vol. 6, no. 2, pp. 113–133, 2020.

Yu He received the master's and doctoral degrees from the Zhejiang University of Technology, in 2015 and 2019, respectively. He is currently a postdoctoral researcher in the Department of Computer Science and Technology at Tsinghua University. His research interests include 3D vision, virtual reality.

Ying-Tian Liu received his B.S. degree of Computer Science and Technology from Tsinghua University, Beijing, in 2020, where he is currently pursuing his master's degree in the Department of Computer Science and Technology of the same university. His research interests include computer graphics and computer vision.

Yi-Han Jin received her bachelor degree of Computer Science from Xuchang University in 2020. she is currently a master student in the Department of Computer Science and Engineering, Tianjin University of Technology. Her research interests include computer graphics and virtual reality.

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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2021.3111729, IEEE Transactions on Visualization and Computer Graphics

Yu-Kun Lai received the bachelor's and Ph.D. degrees in computer science from Tsinghua University in 2003 and 2008, respectively. He is currently a Professor with the School of Computer Science and Informatics, Cardiff University. His research interests include computer graphics, geometry processing, image processing, and computer vision. He is on the editorial boards of Computer Graphics Forum and The Visual Computer.

Shi-Min Hu is currently a professor in the Department of Computer Science and Technology, Tsinghua University. He received his Ph.D. degree from Zhejiang University in 1996. His research interests include digital geometry processing, video processing, rendering, computer animation, and computer aided geometric design. He is the Editor-in-Chief of Computational Visual Media (Springer), and on the editorial boards of several journals, including Computer Aided Design and Computer &Graphics.