Zhang SH, Zhang SK, Liang Y *et al.* A survey of 3D indoor scene synthesis. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 34(3): 594–608 May 2019. DOI 10.1007/s11390-019-1929-5

A Survey of 3D Indoor Scene Synthesis

Song-Hai Zhang[1,2], *Member, CCF, ACM, IEEE*, Shao-Kui Zhang[1], Yuan Liang[1], *Member, CCF, ACM*, and Peter Hall[3]

[1]*Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China*

[2]*Beijing National Research Center for Information Science and Technology (BNRist), Beijing 100084, China*

[3]*Department of Computer Science, University of Bath, Claverton Down, Bath, BA2 7AY, U.K.*

E-mail: shz@tsinghua.edu.cn; {zhangsk18, liangyua14}@mails.tsinghua.edu.cn; maspmh@bath.ac.uk

Received March 15, 2019; revised April 17, 2019.

Abstract Indoor scene synthesis has become a popular topic in recent years. Synthesizing functional and plausible indoor scenes is an inherently difficult task since it requires considerable knowledge to both choose reasonable object categories and arrange objects appropriately. In this survey, we propose four criteria which group a wide range of 3D (three-dimensional) indoor scene synthesis techniques according to various aspects (specifically, four groups of categories). It also provides hints, through comprehensively comparing all the techniques to demonstrate their effectiveness and drawbacks, and discussions of potential remaining problems.

Keywords content generation, indoor scene synthesis, layout arrangement, probabilistic model

1 Introduction

Indoor scene synthesis is a process of generating a room, with appropriate furniture and layout, which has received a great deal of attention in recent years, e.g., [1, 2]. With the development of virtual reality and increasing demands of open-world video games, many scenes are built, where automation mitigates the tedious repetition of hand-crafted scenes. Interior designers benefit too, typically from faster, easier-to-use tools, employed as they consult with customers, to generate suggested layouts[3,4]. Fig.1 shows several examples of indoor scenes.

Synthesizing 3D (three-dimensional) indoor scenes is inherently difficult. Firstly, different rooms usually provide different functionalities, such as sleeping, watching TV, or eating, but sometimes one room may serve several purposes[6], e.g., a living room where we can cook. Secondly, designs (specifically, selecting, positioning and orienting objects) of indoor scenes should take into account many aspects. For instance, there should be clear routes to walk among objects[7,8], and all objects should be accessible. Furthermore, room design may also be required to satisfy non-functional requirements: be visually pleasing, for example. Next, the state of an indoor scene, i.e., what objects are in the room and where they are, exists in an extremely complex and high-dimensional space[9], including discrete categories for the number of objects, continuous values for objects' positions and orientations, etc. Measures and optimisation algorithms in such a space are hard to define.

The problem of automatic synthesis of indoor scenes is the problem of "where" to put "what", subject to constraints, for example, the *use* constraint that determines "what" objects are relevant to a room, and a bed is likely to be placed in a bedroom and unlikely to be placed in a kitchen. Some rooms have multiple uses: so-called "bed-sit" is a single room that combines a kitchen, a bedroom, and a sitting room all into one space. The "functional" constraints determine "where" objects should be placed. For example, people should

Survey

Special Section of CVM 2019

This work was supported by the National Key Technology Research and Development Program under Grant No. 2017YFB1002604, the National Natural Science Foundation of China under Grant Nos. 61772298 and 61832016, the Research Grant of Beijing Higher Institution Engineering Research Center and Tsinghua-Tencent Joint Laboratory for Internet Innovation Technology.

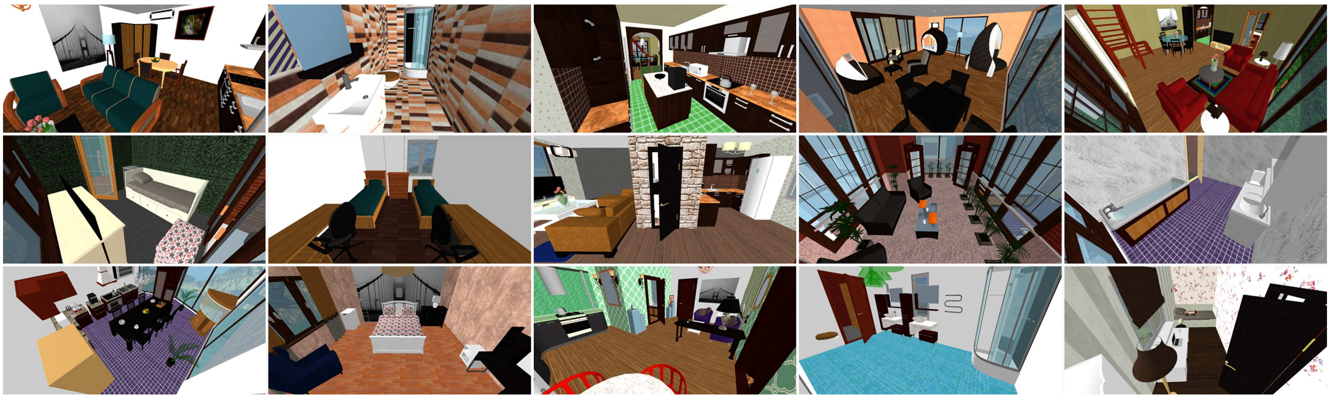


Fig.1. Examples of 3D indoor scenes, rendered using SUNCG datasets[5].

be able to walk through the room, the furniture should be accessible, windows should not be blocked, and so on. Additionally, there are "aesthetic" constraints, which are often interpreted as that the room should be visually pleasing, but adherence to reality could also be considered as such a constraint. There are even "application" constraints: a game designer and an interior designer will typically have different requirements.

Fig.2 shows the typical and general workflow for automatically generating 3D indoor scenes. Ideally, any format of inputs is acceptable, and most work would incorporate learned priors with human knowledge of layouts and functionalities. Next, we would like to reformat the input, because sometimes input is not structural. For example, considering the binary relations and objects' attributes, we may translate text input to a graph[10]. Finally, the "what" and "where" questions are commonly answered by optimizing a scene given all the factors above.

A general way to model the problem is regarded as a search through a space of labelled graphs, e.g., [6, 8]. In this case nodes of the graph correspond to objects, and the arcs represent the relationships between them. The space is very large and complex, making the problem apparently difficult. The graph can vary in the number of nodes, which means the underlying dimension of the space can change. Some dimensions of the space will be categorical, e.g., "chair", "TV" or "bed". Other dimensions will be continuous, such as the distance between objects. Ideally, functions defined over this space will measure both aesthetic and application values, but these are subjective and hard to specify.

The literature forms four criteria, specifically four groups of categories, for dividing work. Each dimension is categorical and contains several classes. The criteria are:

• input, including the explicitly initial scene, sketch, texts, and so on;

• internal representation, including graph, activity-based representation and projection;

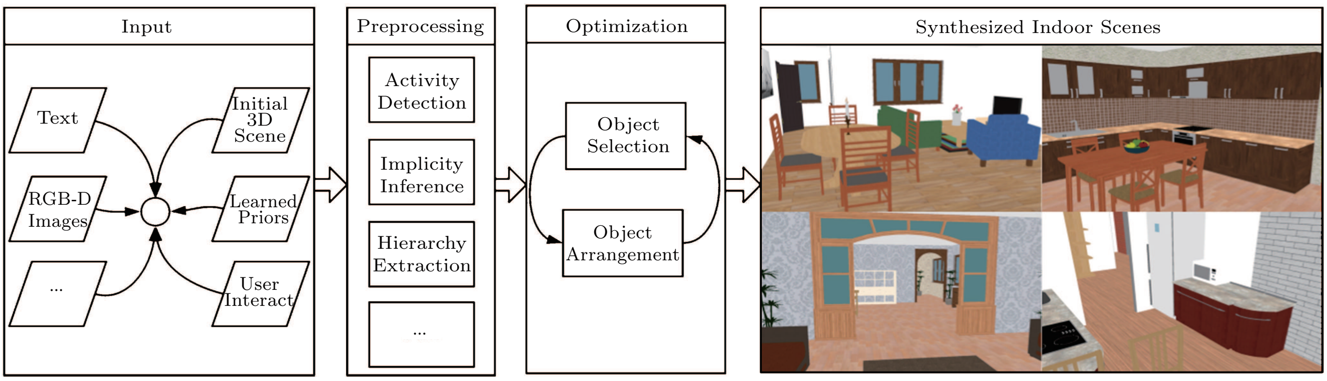


Fig.2. Typical workflow of synthesizing a 3D indoor scene. Given inputs to the algorithm, including constraints of the desired scene (Section 5) and learned priors (Section 7), we would like to generate a plausible scene through several processing stages. Before achieving the final result, some preprocessing techniques are assembled (Section 6). Finally, resulted scenes are optimized by answering two questions: what objects should occur in the scene and where to place the objects (Section 8).

- prior, including hard-coded priors, context, activity-driven priors, examples and neural networks;
- optimization, including interactions, explicit iterations, gradient-based approach and Markov Chain Monte Carlo (MCMC).

The reason why we follow these four criteria is that they form an entire process of synthesis. Intuitively, the four criteria are completely sufficient for a synthesis technique. A large amount of work counts on even fewer aspects, e.g., [11] which has no prior. Thus, following the comprehensive process of an algorithm, we decompose literature at each step or subprocess.

Other decompositions of the literature exist. For example, "interactivity" may be considered by some to be a valid dimension. We include human interaction as a class in the optimization dimension (see Subsection 8.1).

In this survey, we make the following contributions.

1) We formulate comprehensive criteria, which could be instantiated by all indoor scene synthesis literature.

2) We compare state-of-the-art techniques for generating indoor scenes, during which strengths and weaknesses are discussed. As a result, stakeholders, such as researchers, engineers or designers, can easily choose their most appropriate algorithms, considering their own demands.

3) We present the potential remaining problems that most work ignores.

The rest of the paper is organized as follows. Section 2 discusses several topics related to scene synthesis. Section 3 presents the overall criteria of scene synthesis techniques, and relations among three subprocesses are discussed. Section 4 introduces public datasets and their comparisons. Section 5 discusses various input formats. Section 6, Section 7, and Section 8 discuss three subprocesses and we group different concrete techniques for each subprocess, respectively. Finally, we summarize current scene synthesis techniques and propose future directions in Section 9.

2 Related Topics

Scene understanding is the problem of parsing existing scenes into semantic parts. Scenes could be monocular images, RGB-D images, 3D models, and so on. Some authors[12,13] clustered point clouds to segment objects in input RGB-D scans. Satkin *et al.*[14,15] predicted labels for each object in a scene, incorporating 3D auxiliary information, which is similar to [16, 17] that use 3D models to localize objects. Savva *et al.*[18] perceived human activities of given 3D reconstruction indoor scenes. Geometric information is also an alternative solution of recognizing objects in scenes[19]. Scene understanding has many uses, such as [20], but it is the inverse task of scene synthesis. Compared with scene understanding, scene synthesis is trying to generate new scenes given variegated inputs (Section 5). As mentioned in [21], scene understanding and scene synthesis benefit from each other.

Clustering groups scenes that share semantic content. Xu *et al.*[22] clustered a 3D indoor scene repository with merely objects labelled and each cluster contains several focal points that are substructures of scenes, i.e., objects in scenes. All focal points can characterize a cluster because they are discriminative for different clusters and frequent for their own clusters. Quantitative comparison between scenes is also needed[23], which is a way to measure the similarities between different and heterogeneous scenes, based on graph comparisons.

Several studies[24,25] focus on the entire layout of a house or flat. For example, Merrell *et al.*[26] constructed a house at room-to-room level using graphs. They considered the relations between different rooms, while scene synthesis focuses on the content in a particular room. The same is true for [27].

Illustrating above work suggests similarities and differences compared with scene synthesis, while the following contributions are subproblems for scene synthesis. 3D model retrieval retrieves 3D objects given descriptions such as context[28], deformation[29], orthogonal views[30], or even sketches[31]. Chen *et al.*[32] utilized a contour-matching technique[31] to retrieve 3D models and a shape-recovering technique[33] to complete part models, in order to synthesize indoor scenes based on RGB-D images. Even for image retrieval, Schuster *et al.*[34] used a structural representation of an image to generate plausible 3D indoor scenes. Model retrieval is an essential technique to construct indoor scene dataset for priors (Section 7).

Graph models[35] and probabilistic inference are two of the most common approaches to scene synthesis. Recognizing an entire scene pure geometrically is not effective since operations such as convolution are less efficient and ignore structures. As a result, most work chooses structural and hierarchical graphs (Subsection 6.1), and operations are conducted on graphs instead of geometry. To sample a graph or a model generally, many statistical strategies are assembled, especially Markov Chain Monte Carlo (Section 8).

Datasets are needed for data-driven approaches. To generate sufficient training data, tedious and manual annotations or designs are common[5]. Some strategies to automatically generate data exist[7,36], but still require considerable manual work. Training tools are publicly available, e.g., Sketchup① and Planner 5D②. Some authors may choose to create their own training data to suit their particular usage, e.g., [37, 38]. Section 4 will introduce more popular datasets for indoor scene synthesis.

3 Overview

3D indoor scene synthesis is a process of optimizing a plausible, functional and even aesthetic interior scene, given inputs implicitly or explicitly specifying scenes with appropriate priors, under a set of constraints. In this section, we present the overall criteria of 3D indoor scene synthesis and demonstrate each criterion.

Fig.3 shows the criteria of scene synthesis, which can be divided into several coupled subprocesses. An input of algorithms could be any format, and we may choose to encode the input into a more adaptive form. During input preprocessing (internal representation), the input is parsed, transformed or abstracted into other forms that are adaptive to the following operations (Section 6). Offline learning is a process of training a model capable of common sense knowledge such as aesthetics, ergonomics or praxeology (Section 7). After aggregating input, preprocessed input (internal representation) and priors into a set of constraints, each synthesis algorithm would try to optimize the 3D scene in order to make it plausible (Section 8). The concrete structures and contents of inputs can vary extensively from pure texts[10,39] to an initial scene[3,6,40–43], or even a set of examples[11,37], which is discussed in Section 5. In this paper, "input" specifically refers to data fed into input preprocessing or into optimization algorithm directly, while data used to train priors offline refers to a dataset or repository.

Note that a specific algorithm has its own choice for each group of categories, and it need not incorporate each criterion. For example, Fisher *et al.*[44] preprocessed the input to a scene template containing activity distributions and geometric properties. They trained a prior based on a corpus of annotated indoor scenes. During optimization, a specific candidate object with its scale, location and orientation is added into the scene in each iteration. In contrast, Xie *et al.*[11] introduced a non-learning-based method for scene synthesis, i.e., their technique does not contain offline learning stage (prior).

4 Datasets

Training priors requires enormous quantities of data, especially for data-hungry models such as neural networks. Thus, before introducing the criteria, we first discuss several popular datasets that researchers use to train their priors.

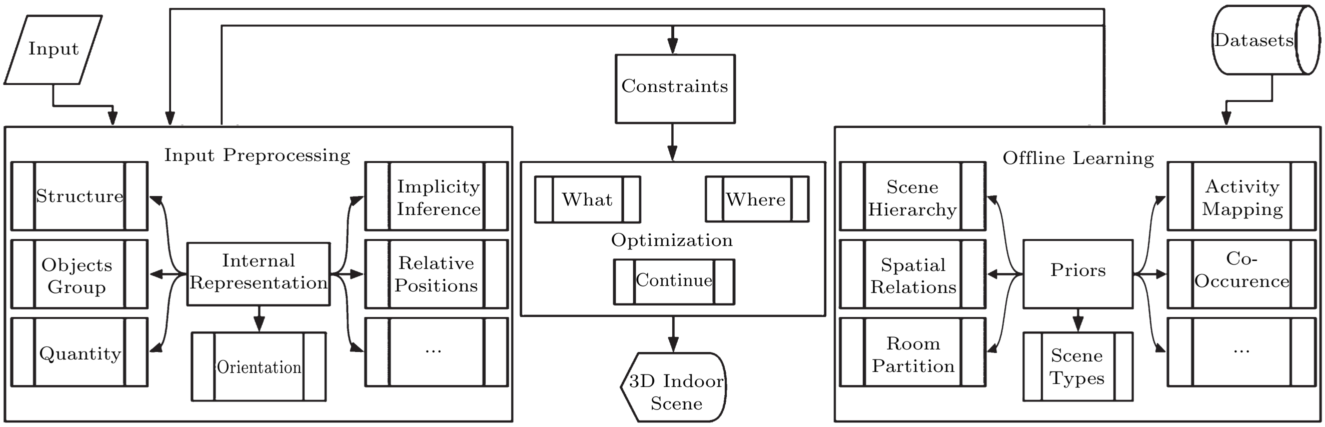


Fig.3. Criteria of scene synthesis. During input preprocessing, we convert the input data into internal representations of models, possibly inferring objects that are not inputted, the structure of objects, relative positions of furniture, etc. To learn a prior, there is alternative knowledge we may choose to encode, e.g., object co-occurrence probability. The prior may not only guide optimization but also direct input preprocessing. Finally, priors and internal data would form constraints to the optimization to determine what is placed and where it should be placed.

①Sketchup. https://www.sketchup.com, April 2019.

②Planner 5D. https://planner5d.com, April 2019.

4.1 Model Collection and Scene Construction

Constructing a huge 3D indoor scene dataset is a sophisticated and tedious task[36]. Crowd-Sourcing[45,46] is one of the most common ways to collect scene configurations, such as Google 3D warehouse③ or Planner 5D④. Placing models into a scene is actually not a trivial task. For example, 3D models could have different measurements, e.g., meter or foot. Labels on models are also of concern because without proper labels users are not able to search for the desired model. To tackle these difficulties, many techniques are proposed, e.g., [47-52].

4.2 Public Datasets

Many datasets exist based on RGB-D data, including NYUv2[53], SUN RGB-D[54], Cornel RGB-D[55], Washington RGB-D[56] and UZH[57]. Illustrations of RGB-D datasets are covered in detail in [58]. Although they are useful for various tasks, noise and mislabeling may still occur due to the low quality of raw depth images or human errors. An alternative way to cope with this problem is to reconstruct triangle meshes of scenes[59], or to use a fully annotated 3D indoor scene to generate images with arbitrary perspectives[36].

SUN3D[60] and ScanNet[61] focus on 3D-based indoor scene datasets instead of limited viewpoints. Given an RGB-D video of indoor scenes, both of them reconstruct a 3D representation of indoor scenes. The idea is that semantic labels of a frame can be propagated into the next frame based on camera poses, and a labelled frame could also be assembled to correct errors. Compared with SUN3D, camera poses are available in ScanNet and it contains larger quantities of indoor scenes.

However, SUN3D is still limited in size where only 254 spaces were captured. In contrast, several datasets go another direction. SUNCG[5] consists of over 45 000 indoor scenes with manually designed rooms and object arrangements. Indoor scenes of SUNCG are completely based on 3D meshes, though a toolbox is available for simulating RGB-D images.

Based on SceneNet[62], SceneNet RGB-D[63] proposes another 3D-based way to generate photorealistic and accurate images. Compared with SUN3D, SceneNet RGB-D is created by human designers instead of 3D reconstructions. Compared with SUNCG, it has fewer layouts, but photorealistic rendering and video trajectories are available. Perturbations of object arrangements are added to increase real-world authenticity.

One recent dataset is InteriorNet[7], which comprises nearly 22 million indoor scenes with realistic rendering. Compared with SceneNet RGB-D, it provides three types of trajectories with camera jitters.

Other relatively small datasets are also available. These datasets are constructed and used for particular purposes. Fisher *et al.*[37] distributed a software tool, which is a platform allowing users to construct scenes by populating objects drawn from Google 3D Warehouse with transformations of objects into indoor scenes, containing kitchens, living rooms, etc. [64] is a special dataset. Its scenes are annotated in natural language.

In addition to scene datasets, 3D models repositories are also needed, e.g., ShapeNet[65,66]. They have annotations for each 3D model, including the category, transformation, front orientation, etc. Based on ShapeNet and ScanNet, [67] introduces Scan2CAD dataset mapping alignment from CAD models to scenes.

5 Input Data

Commonly, synthesizing an indoor scene requires some inputs as constraints or "hints" to guide algorithms, i.e., given a constraint c we want to find the scene s that satisfies:

$$s = \underset{s}{\operatorname{argmax}}\, p(s|c),$$

where s is the scene that we want to generate, and $p(\cdot)$ is the learned or coded model. Often, but not always, an input is needed to fix positions of particular furniture items, or to insist some objects exist in the scene, etc. Several purely automatic algorithms require no input or meaningless input, e.g., [7, 8]. It is also possible to consider a "prior" as "input". Since this survey extracts prior learning as a stand-alone stage, we consider priors as a sub-process of algorithms instead of inputs.

5.1 Explicitly Initial Scene

One of the most intuitive and common ways to constrain scene synthesis is feeding the initial 3D scene. The shape and the area of the room are required by

③Google 3D Warehouse. https://3dwarehouse.sketchup.com/, April 2019.

④Planner 5D. https://planner5d.com, April 2019.

some studies[6,69], and others fix the positions and categories of objects[3,70]. Typical work also includes [11, 37, 40–43]. Among those studies, few papers consider rooms with arbitrary shapes.

5.2 Implicit Input

Implicit inputs are not directly presenting an initial 3D scene, which adds numerous combinations and flexible ideas to scene synthesis. Implicit inputs should be processed into a more structural or meaningful format in the following stages. Thus, internal representation is usually conducted (Section 6). One interesting study[38] inputs human sketches of 3D scenes and converts them into structural groups. We can also use a human language as input to guide the synthesis. [10, 39] are two successive studies. They accept a sentence which can be parsed into a furniture hierarchy. [12, 32, 44] take in one or several RGB-D scans and reconstruct the 3D layout of indoor scenes. [58] is a review, specifically for "RGB-D data to 3D indoor scenes".

6 Internal Representation

The internal representation is a stage where we preprocess the input data into a format that the following algorithms are able to handle. Representation for indoor scenes is complex due to various objects and layouts[9]. There are two significant objectives of internal representation. Firstly, it should be structural, i.e., attributes of objects and relationships between objects are expressed explicitly and consistently. Secondly, it should infer attributes that input data does not provide, object categories for example. In this section, we present several techniques for internal representation. Representation depends slightly on input data; thus we will combine different types of input data as introducing patterns of representation.

6.1 Graph-Based Representation

It is obvious that modelling an indoor scene with a purely geometric mechanism is difficult. An inherent property of indoor scenes is the relationship between different objects, including semantic and spatial meanings. Indoor scenes themselves are commonly comprised of objects supporting other objects[71]. For example, a room can support a table, while the table could support several plates on it. A graph is one of the most intuitive ways to represent such relations, e.g., [72]. Thus, we can represent an indoor scene as a holistic hierarchical relationship. This structural representation is not restricted to 3D indoor scene synthesis. For example, Schuster *et al.*[34] formulated a scene graph for image retrieval.

Chang *et al.*[10,39] preprocessed input text into a static support hierarchy named "scene template", within which nodes are objects including rooms and edges are relationships such as "supporting" or "to the left". For a scene template $T = (O, C, C_s)$, a set of objects O, constraints C, and a scene type C_s are included. Objects also have their own attributes containing colors, materials, etc. Consequently, a single scene template contains considerable detail about a room. A scene template initially contains only explicit information extracted from the input text but is expanded using priors during preprocessing. Eventually, the scene templates are used to iteratively add objects into the pending indoor scene during optimization. Li *et al.*[68] preprocessed the input, which is a vector sampled from Gaussian distribution, into a hierarchical encoding leveraging a neural network based prior (Section 7). Their hierarchical graph is slightly different from that of Chang *et al.*[10,39] Li *et al.*[68] formulated a node-based graph representing a room. All non-leaf nodes contain several pointers to their children and corresponding vectors denoting relative positions of child nodes. A child node of a non-leaf node could be either a leaf node (object category) or another non-leaf node.

Some work uses particular forms of graph. Qi *et al.*[8] learned an attributed spatial And-Or graph (S-AOG)[73], which encodes distributions about indoor scenes. Each scene hierarchy (specifically, a parse tree) is derived from the overall S-AOG through sampling. Compared with the aforementioned work, this work enables that the representation and the prior share the same semantic and format, i.e., the representation of graph is interpretatively and explicitly extracted from the entire graph model which is the prior.

Graphs are also feasible for non-contextual situations. Chen *et al.*[32] utilized graphs to infer objects from noisy point clouds and formed a graph for each scene, in which vertices represent point clouds and edges represent how likely two clouds belong to the same object.

6.2 Activity-Based Representation

Graph-based representations commonly focus on scenes themselves without human-centric inference, i.e., each indoor scene serves several semantical functionalities, such as sleeping or using computers in a bedroom, cooking or having dinner in a kitchen. Namely, a number of particular activities may happen in an indoor scene. Activities can provide important knowledge for inferring categories and arrangement of objects.

Fisher *et al.*[44] assembled the prior of Savva *et al.*[74] so that a distribution of activities is mapped to input RGB-D images denoting continuously how likely activities could happen at each position.

Fu *et al.*[6] required each indoor scene has one or more scene types, which indicates that activities should be executed in the room; therefore they converted input objects categories into a graph. Thus more potential objects categories can be added into pending indoor scenes. Given a room area and few categories, they used a prior called activity-associated object relation graphs to explore what object categories are also needed to satisfy particular room types.

Ma *et al.*[43] associated each initial scene with an action sequence where each action introduces some object categories and human interactions with them. These actions are used to iteratively add and arrange objects in the 3D scene during optimization. Action sequences are sampled from an action graph trained as a prior (Section 7).

6.3 Projection-Based Representation

Previous representations encode input semantically, while projection-based methods project the input into 2D planes geometrically. Wang *et al.*[41] converted the 3D scene into a plan view, which is similar to the work of Ritchie *et al.*[75] Wang *et al.*[41] used convolutional neural networks to generate a probability distribution suggesting objects occurrence and placement on top-down views. Fu *et al.*[6] calculated orthogonal layout masks, which are the weighted sum of similar exemplary layouts from a 2D floor plan database. Xu *et al.*[76] used wall grids to facilitate arrangement. Projecting indoor scenes into 2D views can simplify problems, e.g., convolution becomes feasible. However, synthesis discards 3D information.

7 Priors

Most scene synthesis techniques use priors, which are commonly learned offline and based on probabilistic models. The purpose of introducing priors is to incorporate aesthetic, functional knowledge that is not trivial to manually encode into pending indoor scenes, which significantly increases the plausibility of indoor scenes. Two commonly employed outcomes of prior are object selection and object placement. However, achieving the same outcome could result in completely independent patterns of prior. For example, to automatically select object categories, we can leverage human-object relationship to provide activity-associated suggestions[6,43,44,74], while we can also infer probabilities of object category occurrence based on binary object-object relations[28].

Note that it is not necessary for a prior to yield entire suggestions including object category occurrence and arrangement, i.e., a single prior may merely generate probability about object occurrence. In this section, we discuss priors with respect to their patterns (groups), as well as subtle differences among work.

7.1 Hard-Coded Priors

Merrell *et al.*[3] developed an interactive system that generates design suggestions for each iteration when users fix an instance. They splited the prior into a set of design guidelines, and each guideline corresponds to a formula measuring how much the current layout satisfying the guideline. There are two groups of criteria: functional constraints and aesthetic constraints. For each group, a set of formulas are presented mathematically. Although each formula is rigid, they considered many aspects of indoor scenes, such as clearance (accessibility), pairwise relations and conversation support. However, numerous criteria remain unused. Furthermore, considering merely arrangement, their formulas can support only a few object categories.

Fixed constraints can be partially assembled. Yeh *et al.*[69] coded several factors into a factor graph[77]. For example, a factor could be that the total area of plates must not exceed 70% of the table area. As a result, though fixed priors are less flexible, they still play important roles.

7.2 Contextual Priors

Contextual priors are one kind of the most frequently used priors. A typical form of a context is learning a set of mathematical models, such as (1), which is simply a frequency statistic. Contextual priors focus on relations among entities in indoor scenes. Commonly assembled priors are binary object-object relations, objects with scene type relations, etc. A context-based relationship is not firstly investigated in scene synthesis. Fisher and Hanrahan[28] encoded objects' co-occurrence and spatial relations as contextual priors in order to retrieve desired 3D models.

Chang *et al.*[10,39] learned a set of straight-forward contextual priors. Given a contextual hint, they measured how frequently (probably) an input pattern is likely to occur in the scene, as shown in (1) and (2), where c_p and C denote pattern and context respectively.

$$P(c_p|C) = \frac{count(c_p, C)}{|C|}, \quad (1)$$

$$count(c_p, C) = |\{X|c_p \in X, X \in C\}|. \quad (2)$$

They used this model for inferring how probable an object is likely to occur in a particular scene type, how likely an object is supported by other objects and how likely an object should be placed on a particular surface of another object. They also calculated probabilities of the position and orientation (x, y, θ) of an object, given the object category $C_{\rm obj}$, a reference object category $C_{\rm ref}$, the scene type, and the hierarchical relations of objects.

A co-occurrence prior is a specific form of contextual priors. It explores the binary relationship between two objects o_a and o_b, which is $p(o_a, o_b, e)$, where e represents a particular relation. Chen *et al.*[32] used a Bayesian network[78] to encode probabilities of all relations for each objects pair. Thus, an obvious drawback of contextual priors is that information is utilized merely from the scenes themselves, while a room may have several semantic functionalities. Though Chen *et al.*[32] encompassed floors, walls and ceilings as examples, knowledge is still learned based on pure object-object relations.

In addition to co-occurrence prior, Xu *et al.*[38] proposed priors, called structural groups (SG), which encode both object co-occurrence and co-placement. An SG $G = (V, E)$ is a complete graph, where vertices (V) denote object categories and edges (E) denote relationships such as A-supports-B and relative positions. Each SG has a corresponding probability distribution suggesting co-placement hints. Thus, during optimization, objects selection could be refined using SGs, i.e., a set of objects chosen from the input sketch should satisfy SGs. A placement distribution of an SG can also refine sub-groups of objects of pending indoor scenes.

7.3 Activity-Driven Priors

Objects are related to human poses and activities naturally[79]. Compared with contextual priors, utilizing activities for scene synthesis can adaptively augment scenes, e.g., an activity can judge whether or not an indoor scene satisfies it. Liang *et al.*[70] assembled a straight-forward way to combine room functions together with objects. Fu *et al.*[6] trained activity-associated object relation graphs, which are able to indicate implicit object categories, given the only size and initially specified object categories. Ma *et al.*[43] learned an action model to generate instances of Markov chains for each input scene, i.e., they combined each input scene with an action sequence which can adjust the 3D scene with more object categories or finer object arrangements.

Fisher *et al.*[44] trained a model based on a database of 125 3D scenes[37] and a set of 3D models. They annotated several agent proxies representing interaction with other models. Thus, each 3D model is correspondingly annotated with interaction maps, which are components or regions that agents have to see or touch. Their activity model consists of an occurrence model and an interaction model, judging quantitatively how reasonably a particular set of 3D objects presents in each activity and how good an arrangement of the objects is for the interaction, with respect to two objectives of optimization. Both occurrence and interaction models will be used to iteratively optimize the entire 3D indoor scene during optimization shown in Fig.3. Note that Fisher *et al.*[44] also leveraged another prior trained by Savva *et al.* to generate activity map[74], which is a continuous distribution suggesting how likely an activity can occur and is similar to affordance maps[80] of [8, 79, 81]. This prior is used to convert input RGB-D scan into scene template during preprocessing.

7.4 Example-Based Priors

An intuitive way to synthesize indoor scenes is to learn knowledge directly from provided examples of the scenes. Xie *et al.*[11] reshuffled the objects across a small set of input indoor scenes, which is not based on the learning process. Fu *et al.*[6] selected top K similar room layouts of the same scene type for arrangement hint.

Fisher *et al.*[37] extracted knowledge from a small amount of input example scenes and a database (see Subsection 4.2) so that a similar indoor scene is generated compared with input scenes. They learned two priors: occurrence model and arrangement model, both of which are trained using the input and the database. Weights are added as hyper-parameters to leverage how much we want to bias the influence from the database. For example, if we set $\lambda_{\rm arrange} = 0$, the arrangements of synthesized scenes conform to merely inputs. If we set a higher λ_{arrange}, i.e., adding knowledge from the database into synthesis, synthesized scenes would have more variance. Since their occurrence model is trained with both input scenes and the database, time consumption could be a concern.

7.5 Neural Network Priors

Neural network has become one of the most popular research directions in recent years. Convolutional neural network (CNN) is a neural network incorporating specific layers including convolutional layers, pooling layers, etc. Wang *et al.*[41] trained three convolutional neural networks as separate priors using the ResNet architecture. At each iteration, the first CNN will be assembled to suggest whether or not to continue adding objects, given the current scene. If continuing, the second CNN is used to calculate categories c and locations (x, y) of the next object to add. Finally, the object instance is retrieved through the third CNN into the current scene which would be the input of the next iteration. Avetisyan *et al.*[67] trained a different CNN that accepts voxels and outputs matching probability between a CAD model and an RGB-D scan.

However, Li *et al.*[68] argued that CNN is not suitable for indoor scene synthesis, because traditional convolution does not fit well with the latent structure of indoor scenes. Thus, they combined a recursive neural network (RvNN)[82] with a variational recursive autoencoder (VAE)[83] to construct RvNN-VAE, which is a generative model containing an encoder and a decoder. After training, the decoder is assembled to parse random vectors sampled from Gaussian distribution to a hierarchical representation of indoor scenes, followed by reconstructing 3D scenes.

8 Optimization

Optimization is the stage that combines representations of input and trained priors to synthesize the final 3D indoor scene. Typically, existing work leverages iterative methods that adjust the pending 3D scene to satisfy a set of constraints. As mentioned in Section 3, optimization is optional. For example, Fu *et al.*[6] used a weighted sum of associated scenes to generate the final arrangements. Li *et al.*[68] trained a prior generating the entire hierarchy about a room, which means that details of categories and arrangements are all contained in the generated graph; therefore Li *et al.* merely needed to construct the indoor scene following the graph.

8.1 Human Interaction

Manually optimizing the putative 3D scene requires human interactions with systems, resulting in the semi-automatic process, but priors about plausible, aesthetic and functional arrangements are no longer needed. Some work even focuses on interactions, e.g., [84]. Merrell *et al.*[3] built up a system to aid indoor scene synthesis. By initially inputting a shape and a set of objects, the system can offer a sequence of suggestions following several interior design constraints. A user can iteratively choose suggestions to automatically arrange the indoor scene. For each iteration, users can fix items in specific positions, which would further add constraints for generating suggestions.

Chang *et al.*[39] built up a system that allows users to refine indoor scenes using natural languages. They firstly entered a sentence to generate an initial scene. After that, they modified object categories and positions in that scene using natural language for each iteration.

Savva *et al.*[42] developed another system. Given an initial scene, users can add an object instance by clicking the desired position on the screen, through each iteration. When clicking, the system can automatically list a set of objects that are plausible according to its context as well as the estimated positions and orientations. This is also very similar to [4], which focuses on recommending small-scaled objects, such as plates, books or shoes.

8.2 Explicitly Iterative Methods

Some techniques optimize the putative indoor scenes under several explicit rules. Wang *et al.*[41] trained three priors for continuation, object placement and object instance selection. During optimization, they used the first prior to judge whether or not to continue modifying the indoor scene. If the first prior suggests continuing, the rest two priors are assembled to add new object instances. Otherwise, synthesis is over.

Xie *et al.*[11] proposed a non-learning based approach. After extracting relationships between objects, they reshuffled objects in different scenes, trying every possible position. Next, according to a size factor and an environmental factor, they scored each candidate and discarded failing scenes. Fisher *et al.*[44] also generated a set of candidate scenes and also defined a score function, and they sampled candidates using activity-centric agents.

8.3 Gradient-Based Optimizer

Given several learned priors and other constraints, we can formulate cost functions (or score functions) to measure how reasonable the pending indoor scene is. After that, gradient-based approaches are utilized to maximize (or minimize) cost functions. Chang *et al.*[10,39] defined their cost function with respect to 25% of learned priors and 75% of ergonomic design strategies. Fisher *et al.*[37] optimized a total arrangement score function based on a learned arrangement model and other penalty measurement based on hill climbing optimizer. Xu *et al.*[38] optimized a score function for object arrangements based on gradient descent. Using gradient-based approaches is straight-forward and effective, but cost functions do not guarantee convexity.

8.4 Markov Chain Monte Carlo

Markov chain Monte Carlo (MCMC) is a widely used stochastic optimization method for 3D content generation[85]. It is prohibitively complex to express a state of indoor scenes mathematically, because a relatively simple scene could comprise a set of object categories as discrete values and their positions as continuous values, resulting in an extraordinary high dimensional hypothetical space[9]. Thus, MCMC is used to approximate those multi-dimensional spaces.

Yu *et al.*[40] and Qi *et al.*[8] leveraged simulated annealing[86] with Metropolis-Hastings (MH)[87,88] to iteratively arrange positions and orientations of objects inside indoor scenes, given initially a set of object instances placed randomly. For example, Yu *et al.*[40] treated objects as atoms in annealing metals, and states of indoor scenes are represented as $S = \big\{(p_i, \theta_i)|i = 1, ..., n\big\}$, where (p_i, θ_i) is the position and orientation of the i-th object. For each iteration, they proposed a new subsequent state $S^{'}$. If current state S and new state $S^{'}$ are satisfied with a function $\alpha(S^{'}|S)$, the move would be conducted according to probabilities. They formulated their cost function based on a set of design rules such as "functional objects should be accessible to people", "objects like television should be visible to people", in addition to their priors. Compared with Yu *et al.*[40], Qi *et al.*[8] considered positions and orientations together as potentials according to their Markov random field prior. Thus, for each iteration, they tried to accept the sampled scene (specifically, proposed walk-in MCMC), if the next sample has relatively low energy compared with the current sample or an energy-based probability is used to judge whether to move or not.

Yeh *et al.*[69] formulated an improved version of Reversible Jump MCMC (RJ-MCMC) named Locally Annealed Reversible Jump MCMC (LARJ-MCMC). The pure RJ-MCMC is less compatible with dimensional changes, i.e., inserting or removing objects of indoor scenes often results in low probabilities. To address this, they added a sequence of intermediates during jumping between dimensions, which is named locally annealed jump. Consequently, their work is notably successful in generating scenes without a fixed number of objects. Their work is also flexible to many open-world generating tasks, such as synthesizing a golf field. However, with regard to scene synthesis, LARJ-MCMC requires users to provide latent objects. Furthermore, it can only synthesize a fixed type of scenes. If we want a new type, another factor graph[77] is needed as constraints.

Liang *et al.*[70] presented another technique, which is similar to the work of Yeh *et al.*[69] To address the problem that users need to manually provide object categories, they trained an object selection model based on latent Dirichlet allocation (LDA)[89], in addition to the object placement prior based on Bayesian framework. During optimization, they employed Metropolis-Hastings as an MCMC solver to iteratively adjust the pending indoor scene. Their work is more specific in indoor scene generation but slightly lacks general usage.

Markov chain can also be assembled separately. Ma *et al.*[43] learned an action graph suggesting action transitions, and they used the prior to generating an instance of Markov chain, which is also an action sequence. During optimization, they iteratively placed and re-placed objects according to the sequence.

However, MCMC has an issue with computational cost, e.g., convergence[90]. Fisher *et al.*[44] tried two sampling approaches. During optimization, they utilized activity-associated agents to propose categories and arrangements of candidate objects. They found that, compared with using MCMC, their chain is notably faster.

9 Conclusions

In this survey, we presented the general criteria for decomposing 3D indoor scene synthesis literature. For input data, we discussed the explicit format of data which is basically the initial 3D scene configuration and the implicit format of data which requires further internal representation.

The internal representation is commonly conducted in structural and hierarchical graphs or trees, while projection-based representation is an alternative solution. Priors statistically encode human knowledge about aesthetics and functionalities of rooms. We organized priors into five categories. Two of the most popular priors, according to this survey, are contextual priors and activity priors, which could be basic guidelines. Little work assembles the rest three priors. Although hierarchical information is unlikely to be preserved by convolving and 3D convolution is not efficient, there is still potential usage of the neural network prior which is discussed subsequently. Example-based priors require quite an amount of input and rerunning the entire algorithm including training; thus time is a concern. Unless knowing exactly how synthetic rules behave, we should avoid using hard-coded priors. Finally, with input data or internal representation, and learned priors, optimizations are conducted to generate the final plausible scene. Table 1 lists all related work of 3D indoor scene synthesis of this survey.

Various unsolved problems remain currently.

Table 1. Comparisons Among Different Indoor Scene Synthesis Techniques

Index	Input	Internal Representation	Prior	Optimization
Avetisyan *et al.*[67] (2018)	RGB-D image	-	CNN (3D)	Levenberg-Marquardt
Chang *et al.*[10] (2014)	Text	Graph (static support hierarchy)	Context	Sampling implicit object & gradient
Chang *et al.*[91] (2014)	Text	Graph	Context	User interaction
Chang *et al.*[39] (2017)	Follow Chang *et al.*[10]	Follow [10]	Follow [10]	User interaction
Chen *et al.*[32] (2014)	RGB-D image	Graph based on point cloud segment	Context	Gradient descent
Fisher *et al.*[37] (2012)	Example scenes	Static support hierarchy	Context (Bayesian network & Gaussian mixture models)	Sampling & gradient-based approach
Fisher *et al.*[44] (2015)	RGB-D image	Activity map[74] & geometry	Context with activity	Iterative sampling
Fu *et al.*[6] (2017)	RoomShape, objects' category & quantity	Graph	Activity-associated object relation graphs	Iterations
Jiang *et al.*[81] (2016)	Initial scene (point cloud)	Graph	Conditional random field	MCMC & iteration
Kermani *et al.*[92] (2016)	Initial scene	-	Context (factor graph & K-means)	MCMC & iteration
Li *et al.*[68] (2018)	-	Graph (hierarchy)	RvNN-VAE	-
Liang *et al.*[70] (2017)	RoomShape, doors, windows, room functions	-	Topic model & Bayesian theory (MAP)	MCMC
Liang *et al.*[93] (2018)	Initial scene or scene types	Graph	Context	User interaction & MAP
Ma *et al.*[43] (2016)	Initial scene	-	Action graphs	Action sequence (Markov chain)
Ma *et al.*[94] (2018)	Text	Graph	Context (GMM & frequency)	User interaction (using text)
Merrell *et al.*[3] (2011)	Room shape & furniture set	-	Hard-coded layout guidelines	User interaction
Qi *et al.*[8] (2018)	-	Sampled parse tree	Spatial and-or graph with contextual MRF	MCMC
Ritchie *et al.*[75] (2018)	Initial scene	Top-down scene	CNN	Iteration
Savva *et al.*[42] (2017)	User interaction	Contextual tree & [66]	Context[10]	-
Shao *et al.*[95] (2012)	RGB-D image	-	[96], context (K-means) & random forest	Gradient descent
Wang *et al.*[41] (2018)	Initial scene	Top-down scene	CNN	Iteration
Xie *et al.*[11] (2013)	Example scenes	-	Hard-coded factors	Iteration
Xu *et al.*[38] (2013)	Sketch	-	Structural groups (graphs)	Iteration & gradient descent
Xu *et al.*[76] (2015)	RoomShape, doors, windows, room type	Wall grids	Context	Iterative filtering
Yeh *et al.*[69] (2012)	Room shape & potential furniture objects	-	Factor graph	MCMC (locally annealed reversible jump)
Yu *et al.*[40] (2011)	Initial scene with furnitures placed randomly	-	Context	Simulated annealing

Firstly, one significant limitation is that existing work usually treats each room as a single scene type, e.g., a kitchen, a bedroom, or a living room. A room may be partitioned to some subareas, and each of them serves more coherent functionalities.

Secondly, most work based on context considers one-to-one relations, but relations involving more than two objects are useful in some particular cases. For instance, when placing a fork, a plate and a knife, very specific order should follow from left to right, while with one-to-one relations, a fork could be at any position relative to a plate. This implies using (a, b, c) rather than (a, b), which is an example of a hypergraph[97]. Alternatively, the local field is also an option.

Thirdly, many existing techniques are not sufficiently flexible. They only provide an "integrated" algorithm, i.e., given input the algorithm just generates the final synthesized room. The learned priors can play far more tasks, such as refining an existing scene that has already been arranged adequately, which is useful for incorporating knowledge from various priors.

Though scene synthesis considers simple and regular shapes instead of complex terrains, little work considers rooms with arbitrary shapes, i.e., much existing work still considers rectangular room instead of "L"-shaped, for example. Next, the accessibility to room items such as sockets is commonly ignored. Additionally, style consistencies between furniture are rarely considered for object-object relations[98].

With the development of artificial intelligence, artificial neural network (ANN) becomes one of the most powerful and overwhelming learning techniques. Thus, assembling ANN as a prior could be potentially useful, which is rarely explored in current work. Conventionally, we could use a CNN to tackle spatial entities, but convolution ignores the overall hierarchy of the scene and is less efficient for complex indoor layout[68]. Thus, the specific architecture of ANN for scene synthesis should be considered, and this may become a breakthrough point for it.

References

[1] Lyons G H. Ten Common Home Decorating Mistakes & How to Avoid Them. Blue Sage Press, 2008.

[2] Germer T, Schwarz M. Procedural arrangement of furniture for real-time walkthroughs. *Computer Graphics Forum*, 2009, 28(8): 2068-2078.

[3] Merrell P, Schkufza E, Li Z *et al.* Interactive furniture layout using interior design guidelines. *ACM Transactions on Graphics*, 2011, 30(4): Article No. 87.

[4] Yu L F, Yeung S K, Terzopoulos D. The clutterpalette: An interactive tool for detailing indoor scenes. *IEEE Transactions on Visualization and Computer Graphics*, 2016, 22(2): 1138-1148.

[5] Song S, Yu F, Zeng A *et al.* Semantic scene completion from a single depth image. In *Proc. the 2017 IEEE Conf. Computer Vision and Pattern Recognition*, July 2017, pp.1746-1754.

[6] Fu Q, Chen X, Wang X *et al.* Adaptive synthesis of indoor scenes via activity-associated object relation graphs. *ACM Transactions on Graphics*, 2017, 36(6): Article No. 201.

[7] Li W, Saeedi S, McCormac J *et al.* InteriorNet: Mega-scale multi-sensor photo-realistic indoor scenes dataset. In *Proc. the 29th British Machine Vision Conference*, September 2018, Article No. 77.

[8] Qi S, Zhu Y, Huang S *et al.* Human-centric indoor scene synthesis using stochastic grammar. In *Proc. the 2018 IEEE Conf. Computer Vision and Pattern Recognition*, June 2018, pp.5899-5908.

[9] Li Y, Zhang J, Cheng Y *et al.* DF^{2}Net: Discriminative feature learning and fusion network for RGB-D indoor scene classification. In *Proc. the 32nd AAAI Conference on Artificial Intelligence*, February 2018, pp.7041-7048.

[10] Chang A, Savva M, Manning C D. Learning spatial knowledge for text to 3D scene generation. In *Proc. the 2014 Conference on Empirical Methods in Natural Language Processing*, October 2014, pp.2028-2038.

[11] Xie H, Xu W, Wang B. Reshuffle-based interior scene synthesis. In *Proc. the 12th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry*, November 2013, pp.191-198.

[12] Nan L, Xie K, Sharf A. A search-classify approach for cluttered indoor scene understanding. *ACM Transactions on Graphics*, 2012, 31(6): Article No. 137.

[13] Yang S, Xu J, Chen K *et al.* View suggestion for interactive segmentation of indoor scenes. *Computational Visual Media*, 2017, 3(2): 131-146.

[14] Satkin S, Lin J, Hebert M. Data-driven scene understanding from 3D models. In *Proc. the 2012 British Machine Vision Conference*, September 2012, Article No. 128.

[15] Lim J J, Pirsiavash H, Torralba A. Parsing IKEA objects: Fine pose estimation. In *Proc. the 2013 IEEE International Conference on Computer Vision*, December 2013, pp.2992-2999.

[16] Lim J J, Khosla A, Torralba A. FPM: Fine pose parts-based model with 3D CAD models. In *Proc. the 13th European Conference on Computer Vision*, September 2014, pp.478-493.

[17] Kim Y M, Mitra N J, Yan D M *et al.* Acquiring 3D indoor environments with variability and repetition. *ACM Transactions on Graphics*, 2012, 31(6): Article No. 138.

[18] Savva M, Chang A X, Hanrahan P *et al.* PiGraphs: Learning interaction snapshots from observations. *ACM Transactions on Graphics*, 2016, 35(4): Article No. 139.

[19] Bao S Y, Sun M, Savarese S. Toward coherent object detection and scene layout understanding. *Image and Vision Computing*, 2011, 29(9): 569-579.

[20] Jiang Y, Lim M, Zheng C *et al.* Learning to place new objects in a scene. *The International Journal of Robotics Research*, 2012, 31(9): 1021-1043.

[21] Cheng M M, Hou Q B, Zhang S H *et al.* Intelligent visual media processing: When graphics meets vision. *Journal of Computer Science and Technology*, 2017, 32(1): 110-121.

[22] Xu K, Ma R, Zhang H *et al.* Organizing heterogeneous scene collections through contextual focal points. *ACM Transactions on Graphics*, 2014, 33(4): Article No. 35.

[23] Fisher M, Savva M, Hanrahan P. Characterizing structural relationships in scenes using graph kernels. *ACM Transactions on Graphics*, 2011, 30(4): Article No. 34.

[24] Wu W, Fan L, Liu L *et al.* MIQP-based layout design for building interiors. *Computer Graphics Forum*, 2018, 37(2): 511-521.

[25] Sanchez V, Zakhor A. Planar 3D modeling of building interiors from point cloud data. In *Proc. the 19th IEEE International Conference on Image Processing*, September 2012, pp.1777-1780

[26] Merrell P, Schkufza E, Koltun V. Computer-generated residential building layouts. *ACM Transactions on Graphics*, 2010, 29(6): Article No. 181.

[27] Wang W, Gao W, Hu Z. Effectively modeling piecewise planar urban scenes based on structure priors and CNN. *Science China Information Sciences*, 2019, 62(2): Article No. 29102.

[28] Fisher M, Hanrahan P. Context-based search for 3D models. *ACM Transactions on Graphics*, 2010, 29(6): Article No. 182.

[29] Ovsjanikov M, Li W, Guibas L *et al.* Exploration of continuous variability in collections of 3D shapes. *ACM Transactions on Graphics*, 2011, 30(4): Article No. 33.

[30] Chen D Y, Tian X P, Shen Y T *et al.* On visual similarity based 3D model retrieval. *Computer Graphics Forum*, 2003, 22(3): 223-232.

[31] Eitz M, Richter R, Boubekeur T *et al.* Sketch-based shape retrieval. *ACM Transactions on Graphics*, 2012, 31(4): Article No. 31.

[32] Chen K, Lai Y, Wu Y X *et al.* Automatic semantic modeling of indoor scenes from low-quality RGB-D data using contextual information. *ACM Transactions on Graphics*, 2014, 33(6): Article No. 208.

[33] Shen C H, Fu H, Chen K *et al.* Structure recovery by part assembly. *ACM Transactions on Graphics*, 2012, 31(6): Article No. 180.

[34] Schuster S, Krishna R, Chang A *et al.* Generating semantically precise scene graphs from textual descriptions for improved image retrieval. In *Proc. the 4th Workshop on Vision and Language*, September 2015, pp.70-80.

[35] Koller D, Friedman N. Probabilistic Graphical Models: Principles and Techniques. MIT Press, 2009.

[36] Handa A, Patraucean V, Badrinarayanan V *et al.* Understanding real world indoor scenes with synthetic data. In *Proc. the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, June 2016, pp.4077-4085.

[37] Fisher M, Ritchie D, Savva M *et al.* Example-based synthesis of 3D object arrangements. *ACM Transactions on Graphics*, 2012, 31(6): Article No. 135.

[38] Xu K, Chen K, Fu H *et al.* Sketch2Scene: Sketch-based co-retrieval and co-placement of 3D models. *ACM Transactions on Graphics*, 2013, 32(4): Article No. 123.

[39] Chang A X, Eric M, Savva M *et al.* SceneSeer: 3D scene design with natural language. arXiv:1703.00050, 2017. https://arxiv.org/abs/1703.00050, March 2019.

[40] Yu L F, Yeung S K, Tang C K *et al.* Make it home: Automatic optimization of furniture arrangement. *ACM Transactions on Graphics*, 2011, 30(4): Article No. 86.

[41] Wang K, Savva M, Chang A X *et al.* Deep convolutional priors for indoor scene synthesis. *ACM Transactions on Graphics*, 2018, 37(4): Article No. 70.

[42] Savva M, Chang A X, Agrawala M. SceneSuggest: Context-driven 3D scene design. arXiv:1703.00061, 2017. https://arxiv.org/abs/1703.00061, March 2019.

[43] Ma R, Li H, Zou C *et al.* Action-driven 3D indoor scene evolution. *ACM Transactions on Graphics*, 2016, 35(6): Article No. 173.

[44] Fisher M, Savva M, Li Y *et al.* Activity-centric scene synthesis for functional 3D scene modeling. *ACM Transactions on Graphics*, 2015, 34(6): Article No. 179.

[45] Li G, Zheng Y, Fan J *et al.* Crowdsourced data management: Overview and challenges. In *Proc. the 2017 ACM International Conference on Management of Data*, May 2017, pp.1711-1716.

[46] Chen P P, Sun H L, Fang Y L *et al.* Collusion-proof result inference in crowdsourcing. *Journal of Computer Science and Technology*, 2018, 33(2): 351-365.

[47] Shao L, Chang A X, Su H *et al.* Cross-modal attribute transfer for rescaling 3D models. In *Proc. the 2017 International Conference on 3D Vision*, October 2017, pp.640-648.

[48] Savva M, Chang A X, Bernstein G *et al.* On being the right scale: Sizing large collections of 3D models. In *Proc. the 2014 SIGGRAPH Asia Indoor Scene Understanding Where Graphics Meets Vision*, December 2014, Article No. 4.

[49] Zhu Y, Tian Y, Metaxas D *et al.* Semantic amodal segmentation. In *Proc. the 2017 IEEE Conference on Computer Vision and Pattern Recognition*, July 2017, pp.3001-3009.

[50] Du G G, Yin C L, Zhou M Q *et al.* Isometric 3D shape partial matching using GD-DNA. *Journal of Computer Science and Technology*, 2018, 33(6): 1178-1191.

[51] Jo S, Jeong Y, Lee S. GPU-driven scalable parser for OBJ models. *Journal of Computer Science and Technology*, 2018, 33(2): 417-428.

[52] Yin L, Guo K, Zhou B *et al.* 3D shape co-segmentation via sparse and low rank representations. *Science China Information Sciences*, 2018, 61(5): Article No. 054101.

[53] Silberman N, Hoiem D, Kohli P *et al.* Indoor segmentation and support inference from RGBD images. In *Proc. the 12th European Conference on Computer Vision*, October 2012, pp.746-760.

[54] Song S, Lichtenberg S P, Xiao J. SUN RGB-D: A RGB-D scene understanding benchmark suite. In *Proc. the 2015 IEEE Conference on Computer Vision and Pattern Recognition*, June 2015, pp.567-576.

[55] Anand A, Koppula H S, Joachims T *et al.* Contextually guided semantic labeling and search for three-dimensional point clouds. *The International Journal of Robotics Research*, 2013, 32(1): 19-34.

[56] Lai K, Bo L, Fox D. Unsupervised feature learning for 3D scene labeling. In *Proc. the 2014 IEEE International Conference on Robotics and Automation*, May 2014, pp.3050-3057.

[57] Mattausch O, Panozzo D, Mura C *et al.* Object detection and classification from large-scale cluttered indoor scans. *Computer Graphics Forum*, 2014, 33(2): 11-21.

[58] Chen K, Lai Y K, Hu S M. 3D indoor scene modeling from RGB-D data: A survey. *Computational Visual Media*, 2015, 1(4): 267-278.

[59] Hua B S, Pham Q H, Nguyen D T *et al.* SceneNN: A scene meshes dataset with annotations. In *Proc. the 4th International Conference on 3D Vision*, October 2016, pp.92-101.

[60] Xiao J, Owens A, Torralba A. SUN3D: A database of big spaces reconstructed using SfM and object labels. In *Proc. the 2013 IEEE International Conference on Computer Vision*, December 2013, pp.1625-1632.

[61] Dai A, Chang A X, Savva M *et al.* ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *Proc. the 2017 IEEE Conference on Computer Vision and Pattern Recognition*, July 2017, pp.2432-2443.

[62] Handa A, Pătrăucean V, Stent S *et al.* SceneNet: An annotated model generator for indoor scene understanding. In *Proc. the 2016 IEEE International Conference on Robotics and Automation*, May 2016, pp.5737-5743.

[63] McCormac J, Handa A, Leutenegger S *et al.* SceneNet RGB-D: Can 5M synthetic images beat generic imageNet pre-training on indoor segmentation? In *Proc. the 2017 IEEE International Conference on Computer Vision*, Oct. 2017, pp.2697-2706.

[64] Chang A, Monroe W, Savva M *et al.* Text to 3D scene generation with rich lexical grounding. arXiv:1505.06289, 2015. https://arxiv.org/abs/1505.06289, March 2019.

[65] Chang A X, Funkhouser T, Guibas L *et al.* ShapeNet: An information-rich 3D model repository. arXiv:1512.03012, 2015. https://arxiv.org/abs/1512.03012, March 2019.

[66] Savva M, Chang A X, Hanrahan P. Semantically-enriched 3D models for common-sense knowledge. In *Proc. the 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, June 2015, pp.24-31.

[67] Avetisyan A, Dahnert M, Dai A *et al.* Scan2CAD: Learning CAD model alignment in RGB-D scans. arXiv:1811.11187, 2018. https://arxiv.org/abs/1811.11187, March 2019.

[68] Li M, Patil A G, Xu K *et al.* GRAINS: Generative recursive autoencoders for indoor scenes. *ACM Transactions on Graphics*, 2019, 38(2): Article No. 12.

[69] Yeh Y T, Yang L, Watson M *et al.* Synthesizing open worlds with constraints using locally annealed reversible jump MCMC. *ACM Transactions on Graphics*, 2012, 31(4): Article No. 56.

[70] Liang Y, Zhang S H, Martin R R. Automatic data-driven room design generation. In *Proc. the 3rd International Workshop on Next Generation Computer Animation Techniques*, June 2017, pp.133-148.

[71] Ikehata S, Yang H, Furukawa Y. Structured indoor modeling. In *Proc. the 2015 IEEE International Conference on Computer Vision*, December 2015, pp.1323-1331.

[72] Zhu J Z, Jia Y T, Xu J *et al.* Modeling the correlations of relations for knowledge graph embedding. *Journal of Computer Science and Technology*, 2018, 33(2): 323-334.

[73] Zhu S C, Mumford D. A stochastic grammar of images. *Foundations and Trends® in Computer Graphics and Vision*, 2006, 2(4): 259-362.

[74] Savva M, Chang A X, Hanrahan P *et al.* SceneGrok: Inferring action maps in 3D environments. *ACM Transactions on Graphics*, 2014, 33(6): Article No. 212.

[75] Ritchie D, Wang K, Lin Y. Fast and flexible indoor scene synthesis via deep convolutional generative models. arXiv:1811.12463, 2018. https://arxiv.org/abs/1811.12463, March 2019.

[76] Xu W, Wang B, Yan D M. Wall grid structure for interior scene synthesis. *Computers & Graphics*, 2015, 46: 231-243.

[77] Kschischang F R, Frey B J, Loeliger H A. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 2001, 47(2): 498-519.

[78] Friedman N, Geiger D, Goldszmidt M. Bayesian network classifiers. *Machine Learning*, 1997, 29(2/3): 131-163.

[79] Jiang Y, Lim M, Saxena A. Learning object arrangements in 3D scenes using human context. arXiv:1206.6462, 2012. https://arxiv.org/abs/1206.6462, March 2019.

[80] Gibson J J. The Ecological Approach to Visual Perception (1st edition). Routledge, 2014.

[81] Jiang Y, Koppula H S, Saxena A. Modeling 3D environments through hidden human context. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016, 38(10): 2040-2053.

[82] Socher R, Lin C C, Manning C *et al.* Parsing natural scenes and natural language with recursive neural networks. In *Proc. the 28th International Conference on Machine Learning*, June 2011, pp.129-136.

[83] Kingma D P, Welling M. Auto-encoding variational Bayes. arXiv:1312.6114, 2013. https://arxiv.org/abs/1312.6114, March 2019.

[84] Lyu F, Xi R, Han Y *et al.* MagicMark: A marking menu using 2D direction and 3D depth information. *Science China Information Sciences*, 2018, 61(6): Article No. 064101.

[85] Talton J O, Lou Y, Lesser S *et al.* Metropolis procedural modeling. *ACM Transactions on Graphics*, 2011, 30(2): Article No. 11.

[86] Kirkpatrick S. Optimization by simulated annealing: Quantitative studies. *Journal of Statistical Physics*, 1984, 34(5/6): 975-986.

[87] Hastings W K. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 1970, 57(1): 97-109.

[88] Metropolis N, Rosenbluth A W, Rosenbluth M N *et al.* Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 1953, 21(6): 1087-1092.

[89] Ramage D, Hall D, Nallapati R *et al.* Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In *Proc. the 2009 Conference on Empirical Methods in Natural Language Processing*, August 2009, pp.248-256.

[90] Chen C, Wang W, Zhang Y *et al.* A convergence analysis for a class of practical variance-reduction stochastic gradient MCMC. *Science China Information Sciences*, 2018, 62(1): Article No. 12101.

[91] Chang A, Savva M, Manning C. Interactive learning of spatial knowledge for text to 3D scene generation. In *Proc. the 2014 Association for Computational Linguistics Workshop on Interactive Language Learning, Visualization, and Interfaces*, June 2014, pp.14-21.

[92] Kermani Z S, Liao Z, Tan P *et al.* Learning 3D scene synthesis from annotated RGB-D images. *Computer Graphics Forum*, 2016, 35(5): 197-206.

[93] Liang Y, Xu F, Zhang S H *et al.* Knowledge graph construction with structure and parameter learning for indoor scene design. *Computational Visual Media*, 2018, 4(2): 123-137.

[94] Ma R, Patil A G, Fisher M *et al.* Language-driven synthesis of 3D scenes from scene databases. In *Proc. SIGGRAPH Asia 2018*, September 2018, Article No. 212.

[95] Shao T, Xu W, Zhou K *et al.* An interactive approach to semantic modeling of indoor scenes with an RGBD camera. *ACM Transactions on Graphics*, 2012, 31(6): Article No. 136.

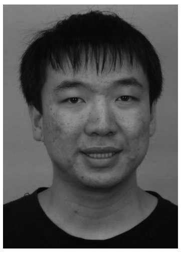
[96] Silberman N, Fergus R. Indoor scene segmentation using a structured light sensor. In *Proc. the 2011 IEEE International Conference on Computer Vision Workshops*, November 2011, pp.601-608.

[97] Berge C. Hypergraphs: Combinatorics of Finite Sets (1st edition). North Holland, 1989.

[98] Liu T, Hertzmann A, Li W *et al.* Style compatibility for 3D furniture models. *ACM Transactions on Graphics*, 2015, 34(4): Article No. 85.

Song-Hai Zhang received his Ph.D. degree in 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, Beijing. His research interests include image/video analysis and processing as well as geometric computing.

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



Yuan Liang is a Ph.D. candidate in the Department of Computer Science and Technology at Tsinghua University, Beijing. His research interests include interactive multimedia analysis and computer graphics. He received his B.S. degree in computer science and technology from Tsinghua University, Beijing, in 2014.

Peter Hall is a professor in the Department of Computer Science at the University of Bath, Bath. He is also the director of the Media Technology Research Centre, Bath. He founded a vision, video, and graphics network of excellence in the United Kingdom, and has served on the executive committee of the British Machine Vision Conference since 2003. He has published extensively in computer vision, especially where it interfaces with computer graphics. More recently he has been developing an interest in robotics.