Human-centric Indoor Scene Synthesis Using Stochastic Grammar

Siyuan Qi[1] Yixin Zhu[1] Siyuan Huang[1] Chenfanfu Jiang[2] Song-Chun Zhu[1]

[1] UCLA Center for Vision, Cognition, Learning and Autonomy
[2] UPenn Computer Graphics Group

Abstract

We present a human-centric method to sample and synthesize 3D room layouts and 2D images thereof, to obtain large-scale 2D/3D image data with perfect per-pixel ground truth. An attributed spatial And-Or graph (S-AOG) is proposed to represent indoor scenes. The S-AOG is a probabilistic grammar model, in which the terminal nodes are object entities. Human contexts as contextual relations are encoded by Markov Random Fields (MRF) on the terminal nodes. We learn the distributions from an indoor scene dataset and sample new layouts using Monte Carlo Markov Chain. Experiments demonstrate that our method can robustly sample a large variety of realistic room layouts based on three criteria: (i) visual realism comparing to a state-of-the-art room arrangement method, (ii) accuracy of the affordance maps with respect to ground-truth, and (ii) the functionality and naturalness of synthesized rooms evaluated by human subjects. The code is available at `https://github.com/SiyuanQi/human-centric-scene-synthesis`.

1. Introduction

Traditional methods of 2D/3D image data collection and ground-truth labeling have evident limitations. i) High-quality ground truths are hard to obtain, as depth and surface normal obtained from sensors are always noisy. ii) It is impossible to label certain ground truth information, *e.g.*, 3D objects sizes in 2D images. iii) Manual labeling of massive ground-truth is tedious and error-prone even if possible. To provide training data for modern machine learning algorithms, an approach to generate large-scale, high-quality data with the perfect per-pixel ground truth is in need.

In this paper, we propose an algorithm to automatically generate a large-scale 3D indoor scene dataset, from which we can render 2D images with pixel-wise ground-truth of the surface normal, depth, and segmentation, *etc*. The proposed algorithm is useful for tasks including but not limited to: i) learning and inference for various computer vision tasks; ii) 3D content generation for 3D modeling and games; iii) 3D reconstruction and robot mappings problems; iv) benchmarking of both low-level and high-level task-planning problems in robotics.

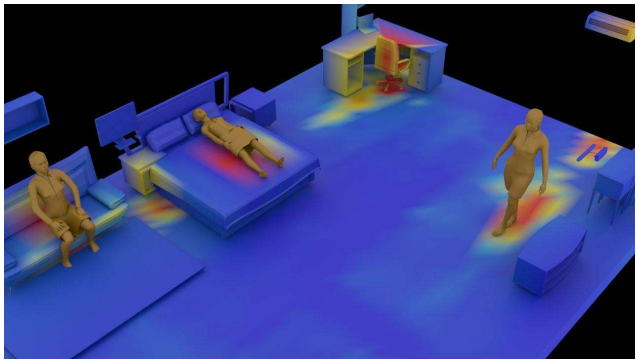


Figure 1: An example of synthesized indoor scene (bedroom) with affordance heatmap. The joint sampling of a scene is achieved by alternative sampling of humans and objects according to the joint probability distribution.

Synthesizing indoor scenes is a non-trivial task. It is often difficult to properly model either the relations between furniture of a functional group, or the relations between the supported objects and the supporting furniture. Specifically, we argue there are four major difficulties. (i) In a functional group such as a dining set, the number of pieces may vary. (ii) Even if we only consider pair-wise relations, there is already a quadratic number of object-object relations. (iii) What makes it worse is that most object-object relations are not obviously meaningful. For example, it is unnecessary to model the relation between a pen and a monitor, even though they are both placed on a desk. (iv) Due to the previous difficulties, an excessive number of constraints are generated. Many of the constraints contain loops, making the final layout hard to sample and optimize.

To address these challenges, we propose a human-centric approach to model indoor scene layout. It integrates human activities and functional grouping/supporting relations as illustrated in Figure 1. This method not only captures the human context but also simplifies the scene structure. Specifically, we use a probabilistic grammar model for images and scenes [49] – an attributed spatial And-Or graph (S-AOG),

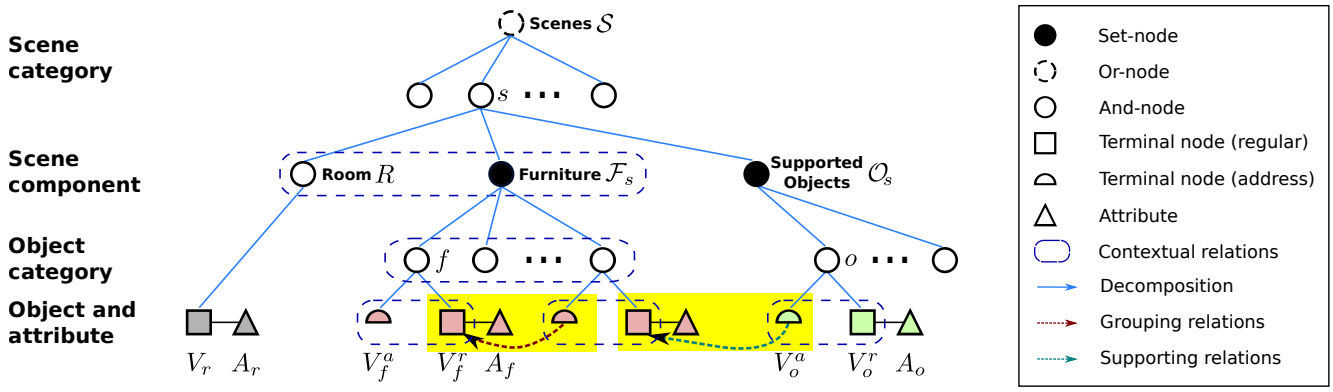


Figure 2: Scene grammar as an attributed S-AOG. A scene of different types is decomposed into a room, furniture, and supported objects. Attributes of terminal nodes are internal attributes (sizes), external attributes (positions and orientations), and a human position that interacts with this entity. Furniture and object nodes are combined by an address terminal node and a regular terminal node. A furniture node (*e.g.*, a chair) is grouped with another furniture node (*e.g.*, a desk) pointed by its address terminal node. An object (*e.g.*, a monitor) is supported by the furniture (*e.g.*, a desk) it is pointing to. If the value of the address node is null, the furniture is not grouped with any furniture, or the object is put on the floor. Contextual relations are defined between the room and furniture, between a supported object and supporting furniture, among different pieces of furniture, and among functional groups.

including vertical hierarchy and horizontal contextual relations. The contextual relations encode functional grouping relations and supporting relations modeled by object affordances [8]. For each object, we learn the affordance distribution, *i.e.*, an object-human relation, so that a human can be sampled based on that object. Besides static object affordance, we also consider dynamic human activities in a scene, constraining the layout by planning trajectories from one piece of furniture to another.

In Section 2, we define the grammar and its parse graph which represents an indoor scene. We formulate the probability of a parse graph in Section 3. The learning algorithm is described in Section 4. Finally, sampling an indoor scene is achieved by sampling a parse tree (Section 5) from the S-AOG according to the prior probability distribution.

This paper makes three major **contributions**. (i) We jointly model objects, affordances, and activity planning for indoor scene configurations. (ii) We provide a general learning and sampling framework for indoor scene modeling. (iii) We demonstrate the effectiveness of this structured joint sampling by extensive comparative experiments.

1.1. Related Work

3D content generation is one of the largest communities in the game industry and we refer readers to a recent survey [13] and book [31]. In this paper, we focus on approaches related to our work using probabilistic inference. Yu [44] and Handa [10] optimize the layout of rooms given a set of furniture using MCMC, while Talton [39] and Yeh [43] consider an open world layout using RJMCMC. These 3D room re-arrangement algorithms optimize room layouts based on constraints to generate new room layouts using a given set of objects. In contrast, the proposed method is capable of adding or deleting objects without fixing the number of objects. Some literature focused on fine-grained room arrangement for specific problems, *e.g.*, small objects arrangement using user-input examples [6] and procedural modeling of objects to encourage volumetric similarity to a target shape [29]. To achieve better realism, Merrell [22] introduced an interactive system providing suggestions following interior design guidelines. Jiang [17] uses a mixture of conditional random field (CRF) to model the hidden human context and arrange new small objects based on existing furniture in a room. However, it cannot direct sampling/synthesizing an indoor scene, since the CRF is intrinsically a discriminative model for structured classification instead of generation.

Synthetic data has been attracting an increasing interest to augment or even serve as training data for object detection and correspondence [5, 21, 24, 34, 38, 46, 48], single-view reconstruction [16], pose estimation [3, 32, 37, 41], depth prediction [36], semantic segmentation [28], scene understanding [9, 10, 45], autonomous pedestrians and crowd [23, 26, 33], VQA [18], training autonomous vehicles [2, 4, 30], human utility learning [42, 50] and benchmarks [11, 27].

Stochastic grammar model has been used for parsing the hierarchical structures from images of indoor [20, 47] and outdoor scenes [20], and images/videos involving humans [25, 40]. In this paper, instead of using stochastic grammar for parsing, we forward sample from a grammar model to generate large variations of indoor scenes.

2. Representation of Indoor Scenes

We use an attributed S-AOG [49] to represent an indoor scene. An attributed S-AOG is a probabilistic grammar model with attributes on the terminal nodes. It combines i) a probabilistic context free grammar (PCFG), and ii) contextual relations defined on an Markov Random Field (MRF), *i.e.*, the horizontal links among the nodes. The PCFG represents the hierarchical decomposition from scenes (top level) to objects (bottom level) by a set of terminal and non-terminal nodes, whereas contextual relations encode the spatial and functional relations through horizontal links. The structure of S-AOG is shown in Figure 2.

Formally, an S-AOG is defined as a 5-tuple: $\mathcal{G} = \langle S, V, R, P, E \rangle$, where we use notations S the root node of the scene grammar, V the vertex set, R the production rules, P the probability model defined on the attributed S-AOG, and E the contextual relations represented as horizontal links between nodes in the same layer. [1]

Vertex Set V can be decomposed into a finite set of non-terminal and terminal nodes: $V = V_{NT} \cup V_T$.

- $V_{NT} = V^{And} \cup V^{Or} \cup V^{Set}$. The non-terminal nodes consists of three subsets. i) A set of **And-nodes** V^{And}, in which each node represents a decomposition of a larger entity (*e.g.*, a bedroom) into smaller components (*e.g.*, walls, furniture and supported objects). ii) A set of **Or-nodes** V^{Or}, in which each node branches to alternative decompositions (*e.g.*, an indoor scene can be a bedroom or a living room), enabling the algorithm to reconfigure a scene. iii) A set of **Set nodes** V^{Set}, in which each node represents a nested And-Or relation: a set of Or-nodes serving as child branches are grouped by an And-node, and each child branch may include different numbers of objects.
- $V_T = V_T^r \cup V_T^a$. The terminal nodes consists of two subsets of nodes: regular nodes and address nodes. i) A **regular terminal node** $v \in V_T^r$ represents a spatial entity in a scene (*e.g.*, an office chair in a bedroom) with attributes. In this paper, the attributes include internal attributes A_{int} of object sizes (w, l, h), external attributes A_{ext} of object position (x, y, z) and orientation ($x-y$ plane) θ, and sampled human positions A_h. ii) To avoid excessively dense graphs, an **address terminal node** $v \in V_T^a$ is introduced to encode interactions that only occur in a certain context but are absent in all others [7]. It is a pointer to regular terminal nodes, taking values in the set $V_T^r \cup \{\text{nil}\}$, representing supporting or grouping relations as shown in Figure 2.

Contextual Relations E among nodes are represented by the horizontal links in S-AOG forming MRFs on the terminal nodes. To encode the contextual relations, we define different types of potential functions for different cliques. The contextual relations $E = E_f \cup E_o \cup E_g \cup E_r$ are divided into four subsets: i) relations among furniture E_f; ii) relations between supported objects and their supporting objects E_o (*e.g.*, a monitor on a desk); iii) relations between objects of a functional pair E_g (*e.g.*, a chair and a desk); and iv) relations between furniture and the room E_r. Accordingly, the cliques formed in the terminal layer could also be divided into four subsets: $C = C_f \cup C_o \cup C_g \cup C_r$. Instead of directly capturing the object-object relations, we compute the potentials using affordances as a bridge to characterize the object-human-object relations.

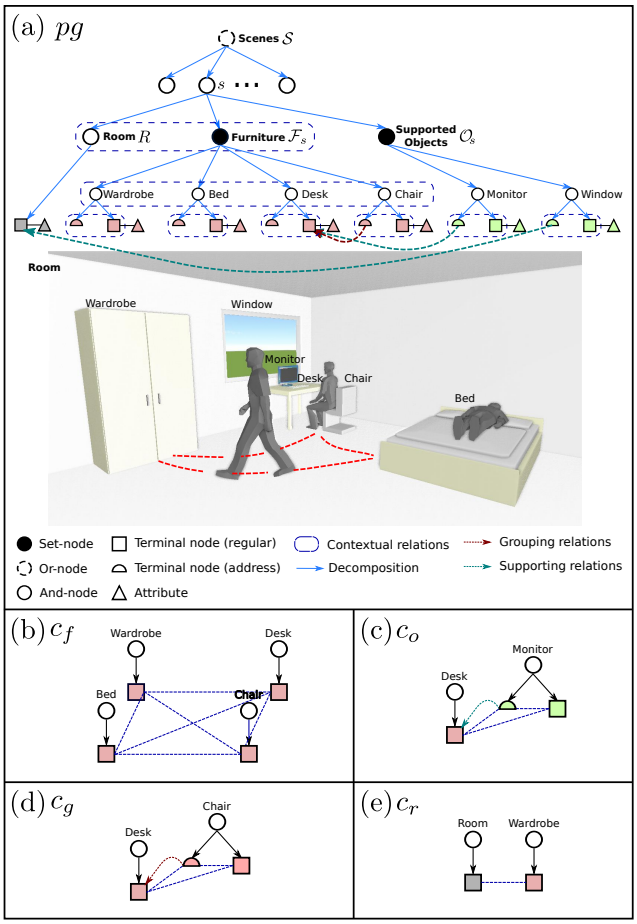


Figure 3: (a) A simplified example of a parse graph of a bedroom. The terminal nodes of the parse graph form an MRF in the terminal layer. Cliques are formed by the contextual relations projected to the terminal layer. Examples of the four types of cliques are shown in (b)-(e), representing four different types of contextual relations.

A hierarchical parse tree pt is an instantiation of the S-AOG by selecting a child node for the Or-nodes as well as determining the state of each child node for the Set-nodes. A parse graph pg consists of a parse tree pt and a number of contextual relations E on the parse tree: $pg = (pt, E_{pt})$. Figure 3 illustrates a simple example of a parse graph and four types of cliques formed in the terminal layer.

[1] We use the term "vertices" instead of "symbols" (in the traditional definition of PCFG) to be consistent with the notations in graphical models.

3. Probabilistic Formulation of S-AOG

A scene configuration is represented by a parse graph pg, including objects in the scene and associated attributes. The prior probability of pg generated by an S-AOG parameterized by Θ is formulated as a Gibbs distribution:

$$p(pg|\Theta) = \frac{1}{Z}\exp\{-\mathcal{E}(pg|\Theta)\} \quad (1)$$

$$= \frac{1}{Z}\exp\{-\mathcal{E}(pt|\Theta) - \mathcal{E}(E_{pt}|\Theta)\}, \quad (2)$$

where $\mathcal{E}(pg|\Theta)$ is the energy function of a parse graph, $\mathcal{E}(pt|\Theta)$ is the energy function of a parse tree, and $\mathcal{E}(E_{pt}|\Theta)$ is the energy term of the contextual relations.

$\mathcal{E}(pt|\Theta)$ can be further decomposed into the energy functions of different types of non-terminal nodes, and the energy functions of internal attributes of both regular and address terminal nodes:

$$\mathcal{E}(pt|\Theta) = \underbrace{\sum_{v\in V^{Or}}\mathcal{E}_{\Theta}^{Or}(v) + \sum_{v\in V^{Set}}\mathcal{E}_{\Theta}^{Set}(v)}_{\text{non-terminal nodes}} + \underbrace{\sum_{v\in V_T^r}\mathcal{E}_{\Theta}^{A_{in}}(v)}_{\text{terminal nodes}}, \quad (3)$$

where the choice of the child node of an Or-node $v \in V^{Or}$ and the child branch of a Set-node $v \in V^{Set}$ follow different multinomial distributions. Since the And-nodes are deterministically expanded, we do not have an energy term for the And-nodes here. The internal attributes A_{in} (size) of terminal nodes follows a non-parametric probability distribution learned by kernel density estimation.

$\mathcal{E}(E_{pt}|\Theta)$ combines the potentials of the four types of cliques formed in the terminal layer, integrating human attributes and external attributes of regular terminal nodes:

$$p(E_{pt}|\Theta) = \frac{1}{Z}\exp\{-\mathcal{E}(E_{pt}|\Theta)\} \quad (4)$$

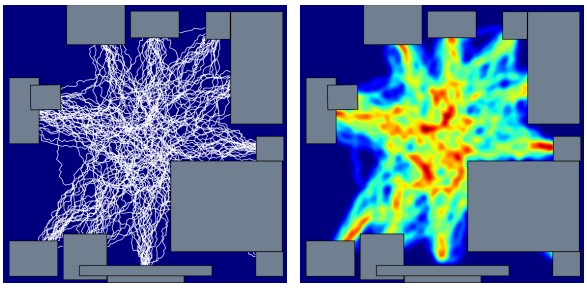
$$= \prod_{c\in C_f}\phi_f(c)\prod_{c\in C_o}\phi_o(c)\prod_{c\in C_g}\phi_g(c)\prod_{c\in C_r}\phi_r(c). \quad (5)$$

Human Centric Potential Functions:

• Potential function $\phi_f(c)$ is defined on relations between furniture (Figure 3(b)). The clique $c = \{f_i\} \in C_f$ includes all the terminal nodes representing furniture:

$$\phi_f(c) = \frac{1}{Z}\exp\{-\lambda_f \cdot \langle \sum_{f_i\neq f_j} l_{\text{col}}(f_i, f_j), l_{\text{ent}}(c)\rangle\}, \quad (6)$$

where λ_f is a weight vector, $<\cdot,\cdot>$ denotes a vector, and the cost function $l_{\text{col}}(f_i, f_j)$ is the overlapping volume of the two pieces of furniture, serving as the penalty of collision. The cost function $l_{\text{ent}}(c) = -H(\Gamma) = \Sigma_i p(\gamma_i)\log p(\gamma_i)$ yields better utility of the room space by sampling human trajectories, where Γ is the set of planned trajectories in the room, and $H(\Gamma)$ is the entropy. The trajectory probability map is first obtained by planning a trajectory γ_i from the center of every piece of furniture to another one using bi-directional rapidly-exploring random tree (RRT) [19], which forms a heatmap. The entropy is computed from the heatmap as shown in Figure 4.



(a) Planned trajectories (b) Probability map

Figure 4: Given a scene configuration, we use bi-directional RRT to plan from every piece of furniture to another, generating a human activity probability map.

• Potential function $\phi_o(c)$ is defined on relations between a supported object and the supporting furniture (Figure 3(c)). A clique $c = \{f, a, o\} \in C_o$ includes a supported object terminal node o, the address node a connected to the object, and the furniture terminal node f pointed by a:

$$\phi_o(c) = \frac{1}{Z}\exp\{-\lambda_o \cdot \langle l_{\text{hum}}(f, o), l_{\text{add}}(a)\rangle\}, \quad (7)$$

where the cost function $l_{\text{hum}}(f, o)$ defines the human usability cost—a favorable human position should enable an agent to access or use both the furniture and the object. To compute the usability cost, human positions h_i^o are first sampled based on position, orientation, and the affordance map of the supported object. Given a piece of furniture, the probability of the human positions is then computed by:

$$l_{\text{hum}}(f, o) = \max_i p(h_i^o|f). \quad (8)$$

The cost function $l_{\text{add}}(a)$ is the negative log probability of an address node $v \in V_T^a$, treated as a certain regular terminal node, following a multinomial distribution.

• Potential function $\phi_g(c)$ is defined on functional grouping relations between furniture (Figure 3(d)). A clique $c = \{f_i, a, f_j\} \in C_g$ consists of terminal nodes of a core functional furniture f_i, pointed by the address node a of an associated furniture f_j. The grouping relation potential is defined similarly to the supporting relation potential

$$\phi_g(c) = \frac{1}{Z}\exp\{-\lambda_c \cdot \langle l_{\text{hum}}(f_i, f_j), l_{\text{add}}(a)\rangle\}. \quad (9)$$

Other Potential Functions:

• Potential function $\phi_r(c)$ is defined on relations between the room and furniture (Figure 3(e)). A clique $c = \{f, r\} \in C_r$ includes a terminal node f and r representing a piece of furniture and a room, respectively. The potential is defined as

$$\phi_r(c) = \frac{1}{Z}\exp\{-\lambda_r \cdot \langle l_{\text{dis}}(f, r), l_{\text{ori}}(f, r)\rangle\}, \quad (10)$$

where the distance cost function is defined as $l_{\text{dis}}(f, r) = -\log p(d|\Theta)$, in which $d \sim \ln\mathcal{N}(\mu, \sigma^2)$ is the distance between the furniture and the nearest wall modeled by a log

normal distribution. The orientation cost function is defined as $l_{\text{ori}}(f,r) = -\log p(\theta|\Theta)$, where $\theta \sim p(\mu,\kappa) = \frac{e^{\kappa\cos(x-\mu)}}{2\pi I_0(\kappa)}$ is the relative orientation between the model and the nearest wall modeled by a von Mises distribution.

4. Learning S-AOG

We use the SUNCG dataset [35] as training data. It contains over 45K different scenes with manually created realistic room and furniture layouts. We collect the statistics of room types, room sizes, furniture occurrences, furniture sizes, relative distances, orientations between furniture and walls, furniture affordance, grouping occurrences, and supporting relations. The parameters Θ of the probability model P can be learned in a supervised way by maximum likelihood estimation (MLE).

Weights of Loss Functions: Recall that the probability distribution of cliques formed in the terminal layer is

$$p(E_{pt}|\Theta) = \frac{1}{Z}\exp\{-\mathcal{E}(E_{pt}|\Theta)\} \tag{11}$$

$$= \frac{1}{Z}\exp\{-\langle\lambda, l(E_{pt})\rangle\}, \tag{12}$$

where λ is the weight vector and $l(E_{pt})$ is the loss vector given by four different types of potential functions.

To learn the weight vector, the standard MLE maximizes the average log-likelihood:

$$\mathcal{L}(E_{pt}|\Theta) = -\frac{1}{N}\sum_{n=1}^{N}\langle\lambda, l(E_{pt_n})\rangle - \log Z. \tag{13}$$

This is usually maximized by following the gradient:

$$\frac{\partial\mathcal{L}(E_{pt}|\Theta)}{\partial\lambda} = -\frac{1}{N}\sum_{n=1}^{N} l(E_{pt_n}) - \frac{\partial\log Z}{\partial\lambda} \tag{14}$$

$$= -\frac{1}{N}\sum_{n=1}^{N} l(E_{pt_n}) - \frac{\partial\log\sum_{pt}\exp\{-\langle\lambda, l(E_{pt})\rangle\}}{\partial\lambda} \tag{15}$$

$$= -\frac{1}{N}\sum_{n=1}^{N} l(E_{pt_n}) + \sum_{pt}\frac{1}{Z}\exp\{-\langle\lambda, l(E_{pt})\rangle\} l(E_{pt}) \tag{16}$$

$$= -\frac{1}{N}\sum_{n=1}^{N} l(E_{pt_n}) + \frac{1}{\widetilde{N}}\sum_{\widetilde{n}=1}^{\widetilde{N}} l(E_{pt_{\widetilde{n}}}), \tag{17}$$

where $\{E_{pt_{\widetilde{n}}}\}_{\widetilde{n}=1,\cdots,\widetilde{N}}$ is the set of synthesized examples from the current model.

It is usually computationally infeasible to sample a Markov chain that burns into an *equilibrium distribution* at every iteration of gradient ascent. Hence, instead of waiting for the Markov chain to converge, we adopt the contrastive divergence (CD) learning that follows the gradient of difference of two divergences [14]

$$\text{CD}_{\widetilde{N}} = \text{KL}(p_0||p_\infty) - \text{KL}(p_{\widetilde{n}}||p_\infty), \tag{18}$$

where $\text{KL}(p_0||p_\infty)$ is the Kullback-Leibler divergence between the data distribution p_0 and the model distribution p_∞, and $p_{\widetilde{n}}$ is the distribution obtained by a Markov chain started at the data distribution and run for a small number $\widetilde{n}$ of steps. In this paper, we set $\widetilde{n} = 1$.

Contrastive divergence learning has been applied effectively to addressing various problems; one of the most notable work is in the context of Restricted Boltzmann Machines [15]. Both theoretical and empirical evidences shows its efficiency while keeping bias typically very small [1]. The gradient of the contrastive divergence is given by

$$\frac{\partial\text{CD}_{\widetilde{N}}}{\partial\lambda} = \frac{1}{N}\sum_{n=1}^{N} l(E_{pt_n}) - \frac{1}{\widetilde{N}}\sum_{\widetilde{n}=1}^{\widetilde{N}} l(E_{pt_{\widetilde{n}}}) - \frac{\partial p_{\widetilde{n}}}{\partial\lambda}\frac{\partial\text{KL}(p_{\widetilde{n}}||p_\infty)}{\partial p_{\widetilde{n}}}. \tag{19}$$

Extensive simulations [14] showed that the third term can be safely ignored since it is small and seldom opposes the resultant of the other two terms.

Finally, the weight vector is learned by gradient descent computed by generating a small number $\widetilde{N}$ of examples from the Markov chain

$$\lambda_{t+1} = \lambda_t - \eta_t\frac{\partial\text{CD}_{\widetilde{N}}}{\partial\lambda} \tag{20}$$

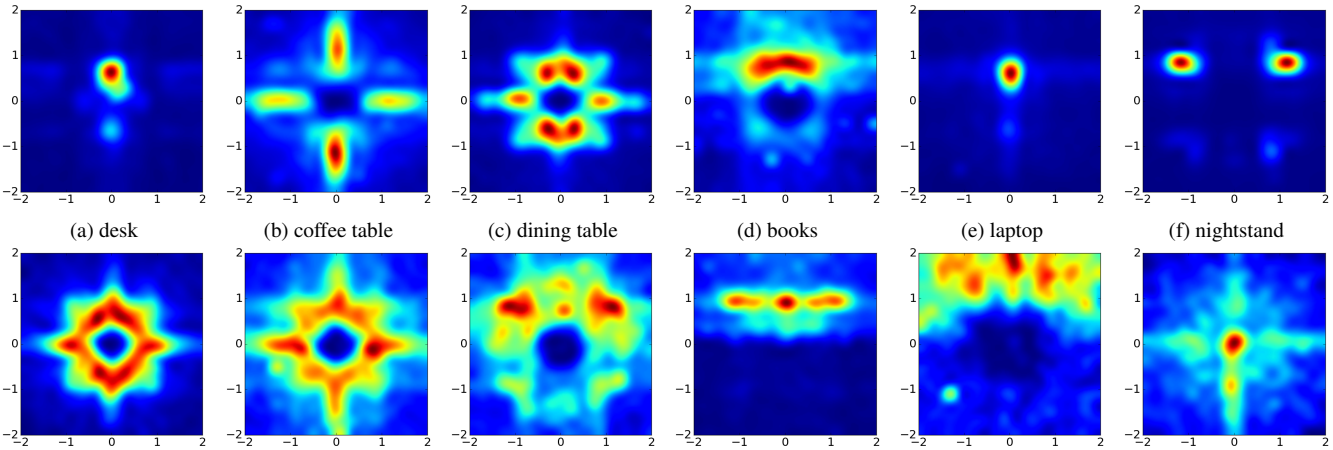
$$= \lambda_t + \eta_t\left(\frac{1}{\widetilde{N}}\sum_{\widetilde{n}=1}^{\widetilde{N}} l(E_{pt_{\widetilde{n}}}) - \frac{1}{N}\sum_{n=1}^{N} l(E_{pt_n})\right). \tag{21}$$

Branching Probabilities: The MLE of the branch probabilities ρ_i of Or-nodes, Set-nodes and address terminal nodes is simply the frequency of each alternative choice [49]: $\rho_i = \#(v \to u_i) / \sum_{j=1}^{n(v)} \#(v \to u_j)$.

Grouping Relations: The grouping relations are hand-defined (*i.e.*, nightstands are associated with beds, chairs are associated with desks and tables). The probability of occurrence is learned as a multinomial distribution, and the supporting relations are automatically extracted from SUNCG.

Room Size and Object Sizes: The distribution of the room size and object size among all the furniture and supported objects is learned as a non-parametric distribution. We first extract the size information from the 3D models inside SUNCG dataset, and then fit a non-parametric distribution using kernel density estimation. The distances and relative orientations of the furniture and objects to the nearest wall are computed and fitted into a log normal and a mixture of von Mises distributions, respectively.

Affordances: We learn the affordance maps of all the furniture and supported objects by computing the heatmap of possible human positions. These position include annotated humans, and we assume that the center of chairs, sofas, and beds are positions that humans often visit. By accumulating the relative positions, we get reasonable affordance maps as non-parametric distributions as shown in Figure 5.



(a) desk (b) coffee table (c) dining table (d) books (e) laptop (f) nightstand

(g) fruit bowl (h) vase (i) floor lamp (j) wall lamp (k) fireplace (l) ceiling fan

Figure 5: Examples of the learned affordance maps. Given the object positioned in the center facing upwards, *i.e.*, coordinate of $(0, 0)$ facing direction $(0, 1)$, the maps show the distributions of human positions. The affordance maps accurately capture the subtle differences among desks, coffee tables, and dining tables. Some objects are orientation sensitive, *e.g.*, books, laptops, and night stands, while some are orientation invariant, *e.g.*, fruit bowls and vases.

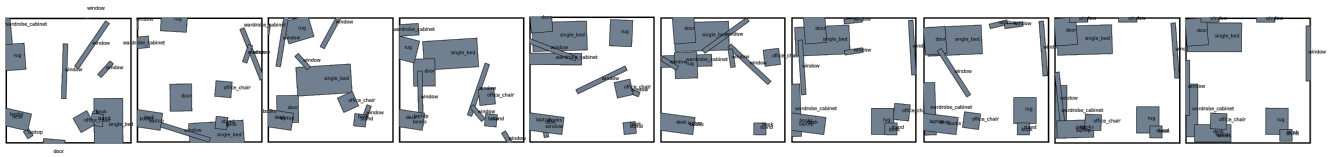


Figure 6: MCMC sampling process (from left to right) of scene configurations with simulated annealing.

5. Synthesizing Scene Configurations

Synthesizing scene configurations is accomplished by sampling a parse graph pg from the prior probability $p(pg|\Theta)$ defined by the S-AOG. The structure of a parse tree pt (*i.e.*, the selection of Or-nodes and child branches of Set-nodes) and the internal attributes (sizes) of objects can be easily sampled from the closed-form distributions or non-parametric distributions. However, the external attributes (positions and orientations) of objects are constrained by multiple potential functions, hence they are too complicated to be directly sampled from. Here, we utilize a Markov chain Monte Carlo (MCMC) sampler to draw a typical state in the distribution. The process of each sampling can be divided into two major steps:

1. Directly sample the structure of pt and internal attributes A_{in}: (i) sample the child node for the Or-nodes; (ii) determine the state of each child branch of the Set-nodes; and (iii) for each regular terminal node, sample the sizes and human positions from learned distributions.

2. Use an MCMC scheme to sample the values of address nodes V^a and external attributes A_{ex} by making proposal moves. A sample will be chosen after the Markov chain converges.

We design two simple types of Markov chain dynamics which are used at random with probabilities $q_i, i = 1, 2$ to make proposal moves:

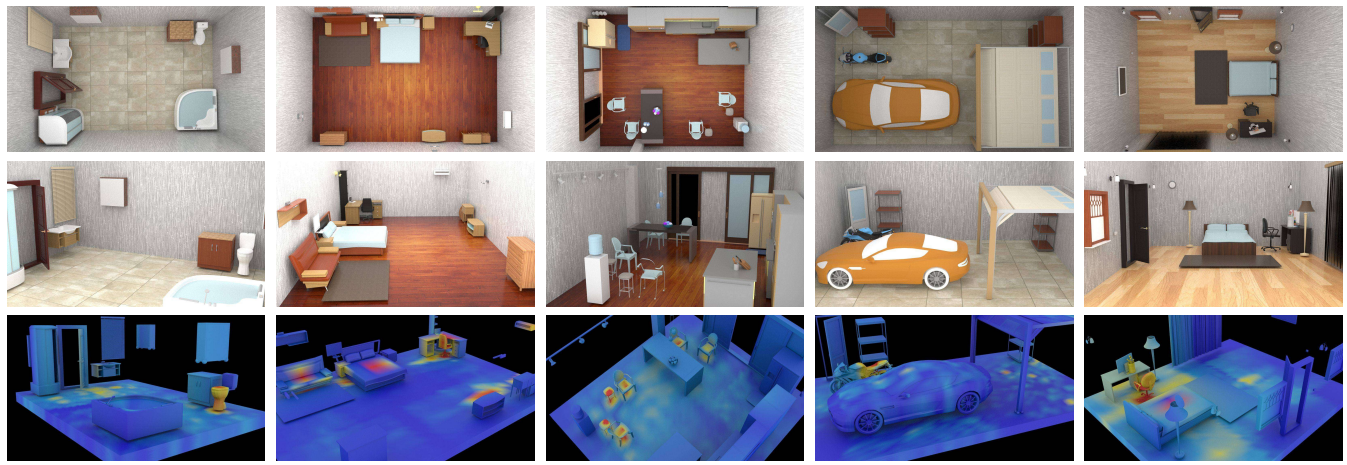
- Dynamics q_1: translation of objects. This dynamic chooses a regular terminal node, and samples a new position based on the current position x: $x \rightarrow x + \delta x$, where δx follows a bivariate normal distribution.
- Dynamics q_2: rotation of objects. This dynamic chooses a regular terminal node, and samples a new orientation based on the current orientation of the object: $\theta \rightarrow \theta + \delta\theta$, where $\delta\theta$ follows a normal distribution.

Adopting the Metropolis-Hastings algorithm, the proposed new parse graph pg' is accepted according to the following acceptance probability:

$$\alpha(pg'|pg, \Theta) = \min(1, \frac{p(pg'|\Theta)p(pg|pg')}{p(pg|\Theta)p(pg'|pg)}) \quad (22)$$

$$= \min(1, \exp(\mathcal{E}(pg|\Theta) - \mathcal{E}(pg'|\Theta))), \quad (23)$$

where the proposal probability rate is canceled since the proposal moves are symmetric in probability. A simulated annealing scheme is adopted to obtain samples with high probability as shown in Figure 6.



(a) bathroom (b) bedroom (c) dining room (d) garage (e) guest room

(f) gym (g) kitchen (h) living room (i) office (j) storage

Figure 7: Examples of scenes in ten different categories. Top: top-view. Middle: a side-view. Bottom: affordance heatmap.

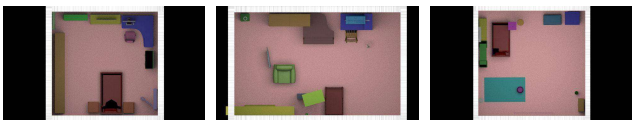
6. Experiments

We design three experiments based on different criteria: i) visual similarity to manually constructed scenes, ii) the accuracy of affordance maps for the synthesized scenes, and iii) functionalities and naturalness of the synthesized scenes. The first experiment compares our method with a state-of-the-art room arrangement method; the second experiment measures the synthesized affordances; the third one is an ablation study. Overall, the experiments show that our algorithm can robustly sample a large variety of realistic scenes that exhibits naturalness and functionality.

Layout Classification. To quantitatively evaluate the visual realism, we trained a classifier on the top-view segmentation maps of synthesized scenes and SUNCG scenes.

Table 1: Classification results on segmentation maps of synthesized scenes using different methods vs. SUNCG.

Method	Yu *et al*. [44]	SUNCG Perturbed	Ours
Accuracy(%) $\downarrow$	87.49	63.69	76.18



(a) SUNCG Perturbed (b) Yu *et al*. [44] (c) Ours

Figure 8: Top-view segmentation maps for classification.

Specifically, we train a ResNet-152 [12] to classify top view layout segmentation maps (synthesized vs. SUNCG). Examples of top-view segmentation maps are shown in Figure 8. The reason to use segmentation maps is that we want to evaluate the room layout excluding rendering factors such as object materials. We use two methods for comparison: i) a state-of-the-art furniture arrangement optimization method proposed by Yu *et al*. [44], and ii) slight perturbation of SUNCG scenes by adding small Gaussian noise (*e.g*. $\mu = 0, \sigma = 0.1$) to the layout. The room arrangement algorithm proposed by [44] takes one pre-fixed input room and re-organizes the room. 1500 scenes are randomly

Table 2: Comparison between affordance maps computed from our samples and real data

Metric	Bathroom	Bedroom	Dining Room	Garage	Guest Room	Gym	Kitchen	Living Room	Office	Storage
Total variation	0.431	0.202	0.387	0.237	0.175	0.278	0.227	0.117	0.303	0.708
Hellinger distance	0.453	0.252	0.442	0.284	0.212	0.294	0.251	0.158	0.318	0.703

Table 3: Human subjects' ratings (1-5) of the sampled layouts based on functionality (top) and naturalness (bottom)

Method	Bathroom	Bedroom	Dining Room	Garage	Guest Room	Gym	Kitchen	Living Room	Office	Storage
no-context	1.12 ± 0.33	1.25 ± 0.43	1.38 ± 0.48	1.75 ± 0.66	1.50 ± 0.50	3.75 ± 0.97	2.38 ± 0.48	1.50 ± 0.87	1.62 ± 0.48	1.75 ± 0.43
object	3.12 ± 0.60	3.62 ± 1.22	2.50 ± 0.71	3.50 ± 0.71	2.25 ± 0.97	3.62 ± 0.70	3.62 ± 0.70	3.12 ± 0.78	1.62 ± 0.48	4.00 ± 0.71
Yu *et al*. [44]	3.61 ± 0.52	4.15 ± 0.25	3.15 ± 0.40	3.59 ± 0.51	2.58 ± 0.31	2.03 ± 0.56	3.91 ± 0.98	4.62 ± 0.21	3.32 ± 0.81	2.58 ± 0.64
ours	4.58 ± 0.86	4.67 ± 0.90	3.33 ± 0.90	3.96 ± 0.79	3.25 ± 1.36	4.04 ± 0.79	4.21 ± 0.87	4.58 ± 0.86	3.67 ± 0.75	4.79 ± 0.58
no-context	1.00 ± 0.00	1.00 ± 0.00	1.12 ± 0.33	1.38 ± 0.70	1.12 ± 0.33	1.62 ± 0.86	1.00 ± 0.00	1.25 ± 0.43	1.12 ± 0.33	1.00 ± 0.00
object	2.88 ± 0.78	3.12 ± 1.17	2.38 ± 0.86	3.00 ± 0.71	2.50 ± 0.50	3.38 ± 0.86	3.25 ± 0.66	2.50 ± 0.50	1.25 ± 0.43	3.75 ± 0.66
Yu *et al*. [44]	4.00 ± 0.52	3.85 ± 0.92	3.27 ± 1.01	2.99 ± 0.25	3.52 ± 0.93	2.14 ± 0.63	3.89 ± 0.90	3.31 ± 0.29	2.77 ± 0.67	2.96 ± 0.41
ours	4.21 ± 0.71	4.25 ± 0.66	3.08 ± 0.70	3.71 ± 0.68	3.83 ± 0.80	4.17 ± 0.75	4.38 ± 0.56	3.42 ± 0.70	3.25 ± 0.72	4.54 ± 0.71

Figure 9: **Top**: previous methods [44] only re-arranges a given input scene with a fixed room size and a predefined set of objects. **Bottom**: our method samples a large variety of scenes.

selected for each method and SUNCG: 800 for training, 200 for validation, and 500 for testing. As shown in Table 1, the classifier successfully distinguishes Yu *et al*. vs. SUNCG with an accuracy of 87.49%. Our method achieves a better performance of 76.18%, exhibiting a higher realism and larger variety. This result indicates our method is much more visually similar to real scenes than the comparative scene optimization method. Qualitative comparisons of Yu *et al*. and our method are shown in Figure 9.

Affordance Maps Comparison. We sample 500 rooms of 10 different scene categories summarized in Table 2. For each type of room, we compute the affordance maps of the objects in the synthesized samples, and calculate both the total variation distances and Hellinger distances between the affordance maps computed from the synthesized samples and the SUNCG dataset. The two distributions are similar if the distance is close to 0. Most sampled scenes using the proposed method show similar affordance distributions to manually created ones from SUNCG. Some scene types (*e.g*. Storage) show a larger distance since they do not exhibit clear affordances. Overall, the results indicate that affordance maps computed from the synthesized scenes are reasonably close to the ones computed from manually constructed scenes by artists.

Functionality and naturalness. Three methods are used for comparison: (i) direct sampling of rooms according to the statistics of furniture occurrence without adding contextual relation, (ii) an approach that only models object-wise relations by removing the human constraints in our model, and (iii) the algorithm proposed by Yu *et al*. [44]. We showed the sampled layouts using three methods to 4 human subjects. Subjects were told the room category in advance, and instructed to rate given scene layouts without knowing the method used to generate the layouts. For each of the 10 room categories, 24 samples were randomly selected using our method and [44], and 8 samples were selected using both the object-wise modeling method and the random generation. The subjects evaluated the layouts based on two criteria: (i) functionality of the rooms, *e.g*., can the "bedroom" satisfies a human's needs for daily life; and (ii) the naturalness and realism of the layout. Scales of responses range from 1 to 5, with 5 indicating perfect functionalilty or perfect naturalness and realism. The mean ratings and the standard deviations are summarized in Table 3. Our approach outperforms the three methods in both criteria, demonstrating the ability to sample a functionally reasonable and realistic scene layout. More qualitative results are shown in Figure 7.

Complexity of synthesis. The time complexity is hard to measure since MCMC sampling is adopted. Empirically, it takes about 20-40 minutes to sample an interior layout (20000 iterations of MCMC), and roughly 12-20 minutes to render a 640×480 image on a normal PC. The rendering speed depends on settings related to illumination, environments, and the size of the scene, *etc*.

7. Conclusion

We propose a novel general framework for human-centric indoor scene synthesis by sampling from a spatial And-Or graph. The experimental results demonstrate the effectiveness of our approach over a large variety of scenes based on different criteria. In the future, to synthesize physically plausible scenes, a physics engine should be integrated. We hope the synthesized data can contribute to the broad AI community.

Acknowledgment

The authors thank Professor Ying Nian Wu from UCLA Statistics Department and Professor Demetri Terzopoulos from UCLA Computer Science Department for insightful discussions. The work reported herein is supported by DARPA XAI N66001-17-2-4029 and ONR MURI N00014-16-1-2007.

References

[1] M. A. Carreira-Perpinan and G. E. Hinton. On contrastive divergence learning. In *AI Stats*, 2005.

[2] C. Chen, A. Seff, A. Kornhauser, and J. Xiao. Deepdriving: Learning affordance for direct perception in autonomous driving. In *ICCV*, 2015.

[3] W. Chen, H. Wang, Y. Li, H. Su, D. Lischinsk, D. Cohen-Or, B. Chen, et al. Synthesizing training images for boosting human 3d pose estimation. In *3DV*, 2016.

[4] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun. CARLA: An open urban driving simulator. In *CORL*, 2017.

[5] D. Dwibedi, I. Misra, and M. Hebert. Cut, paste and learn: Surprisingly easy synthesis for instance detection. *ICCV*, 2017.

[6] M. Fisher, D. Ritchie, M. Savva, T. Funkhouser, and P. Hanrahan. Example-based synthesis of 3d object arrangements. *TOG*, 2012.

[7] A. Fridman. Mixed markov models. *PNAS*, 2003.

[8] J. J. Gibson. *The ecological approach to visual perception.* Houghton, Mifflin and Company, 1979.

[9] A. Handa, V. Pătrăucean, V. Badrinarayanan, S. Stent, and R. Cipolla. Understanding real world indoor scenes with synthetic data. In *CVPR*, 2016.

[10] A. Handa, V. Patraucean, S. Stent, and R. Cipolla. Scenenet: an annotated model generator for indoor scene understanding. In *ICRA*, 2016.

[11] A. Handa, T. Whelan, J. McDonald, and A. J. Davison. A benchmark for rgb-d visual odometry, 3d reconstruction and slam. In *ICRA*, 2014.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[13] M. Hendrikx, S. Meijer, J. Van Der Velden, and A. Iosup. Procedural content generation for games: A survey. *TOMM*, 2013.

[14] G. E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 2002.

[15] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 2006.

[16] Q. Huang, H. Wang, and V. Koltun. Single-view reconstruction via joint analysis of image and shape collections. *TOG*, 2015.

[17] Y. Jiang, H. S. Koppula, and A. Saxena. Modeling 3d environments through hidden human context. *PAMI*, 2016.

[18] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. L. Zitnick, and R. Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. *CVPR*, 2017.

[19] S. M. LaValle. Rapidly-exploring random trees: A new tool for path planning. Technical report, 1998.

[20] X. Liu, Y. Zhao, and S.-C. Zhu. Single-view 3d scene parsing by attributed grammar. In *CVPR*, 2014.

[21] J. McCormac, A. Handa, S. Leutenegger, and A. J. Davison. Scenenet rgb-d: Can 5m synthetic images beat generic imagenet pre-training on indoor segmentation? In *ICCV*, 2017.

[22] P. Merrell, E. Schkufza, Z. Li, M. Agrawala, and V. Koltun. Interactive furniture layout using interior design guidelines. *TOG*, 2011.

[23] J. Ondřej, J. Pettré, A.-H. Olivier, and S. Donikian. A synthetic-vision based steering approach for crowd simulation. *TOG*, 2010.

[24] C. R. Qi, H. Su, M. Niessner, A. Dai, M. Yan, and L. J. Guibas. Volumetric and multi-view cnns for object classification on 3d data. In *CVPR*, 2016.

[25] S. Qi, S. Huang, P. Wei, and S.-C. Zhu. Predicting human activities using stochastic grammar. In *ICCV*, 2017.

[26] S. Qi and S.-C. Zhu. Intent-aware multi-agent reinforcement learning. In *ICRA*, 2018.

[27] W. Qiu and A. Yuille. Unrealcv: Connecting computer vision to unreal engine. *ACM Multimedia Open Source Software Competition*, 2016.

[28] S. R. Richter, V. Vineet, S. Roth, and V. Koltun. Playing for data: Ground truth from computer games. In *ECCV*, 2016.

[29] D. Ritchie, B. Mildenhall, N. D. Goodman, and P. Hanrahan. Controlling procedural modeling programs with stochastically-ordered sequential monte carlo. *TOG*, 2015.

[30] S. Shah, D. Dey, C. Lovett, and A. Kapoor. Aerial Informatics and Robotics platform. Technical report, Microsoft Research, 2017.

[31] N. Shaker, J. Togelius, and M. J. Nelson. *Procedural Content Generation in Games*. Springer, 2016.

[32] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter-sensitive hashing. In *ICCV*, 2003.

[33] W. Shao and D. Terzopoulos. Autonomous pedestrians. In *SCA*, 2005.

[34] S. Song and J. Xiao. Sliding shapes for 3d object detection in depth images. In *ECCV*, 2014.

[35] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser. Semantic scene completion from a single depth image. In *CVPR*, 2017.

[36] H. Su, Q. Huang, N. J. Mitra, Y. Li, and L. Guibas. Estimating image depth using shape collections. *TOG*, 2014.

[37] H. Su, C. R. Qi, Y. Li, and L. J. Guibas. Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views. In *ICCV*, 2015.

[38] B. Sun and K. Saenko. From virtual to reality: Fast adaptation of virtual object detectors to real domains. In *BMVC*, 2014.

[39] J. O. Talton, Y. Lou, S. Lesser, J. Duke, R. Měch, and V. Koltun. Metropolis procedural modeling. *TOG*, 2011.

[40] W. Wang, Y. Xu, J. Shen, and S.-C. Zhu. Attentive fashion grammar network for fashion landmark detection and clothing category classification. In *CVPR*, 2018.

[41] H. Yasin, U. Iqbal, B. Krüger, A. Weber, and J. Gall. A dual-source approach for 3d pose estimation from a single image. In *CVPR*, 2016.

[42] T. Ye, S. Qi, J. Kubricht, Y. Zhu, H. Lu, and S.-C. Zhu. The martian: Examining human physical judgments across virtual gravity fields. *TVCG*, 2017.

[43] Y.-T. Yeh, L. Yang, M. Watson, N. D. Goodman, and P. Hanrahan. Synthesizing open worlds with constraints using locally annealed reversible jump mcmc. *TOG*, 2012.

[44] L. F. Yu, S. K. Yeung, C. K. Tang, D. Terzopoulos, T. F. Chan, and S. J. Osher. Make it home: automatic optimization of furniture arrangement. *TOG*, 2011.

[45] Y. Zhang, M. Bai, P. Kohli, S. Izadi, and J. Xiao. Deepcontext: Context-encoding neural pathways for 3d holistic scene understanding. In *ICCV*, 2017.

[46] Y. Zhang, S. Song, E. Yumer, M. Savva, J.-Y. Lee, H. Jin, and T. Funkhouser. Physically-based rendering for indoor scene understanding using convolutional neural networks. *CVPR*, 2017.

[47] Y. Zhao and S.-C. Zhu. Scene parsing by integrating function, geometry and appearance models. In *CVPR*, 2013.

[48] T. Zhou, P. Krähenbühl, M. Aubry, Q. Huang, and A. A. Efros. Learning dense correspondence via 3d-guided cycle consistency. In *CVPR*, 2016.

[49] S.-C. Zhu and D. Mumford. A stochastic grammar of images. *Foundations and Trends® in Computer Graphics and Vision*, 2007.

[50] Y. Zhu, C. Jiang, Y. Zhao, D. Terzopoulos, and S.-C. Zhu. Inferring forces and learning human utilities from videos. In *CVPR*, 2016.

Supplementary Material for Human-centric Indoor Scene Synthesis Using Stochastic Grammar

Siyuan Qi[1] Yixin Zhu[1] Siyuan Huang[1] Chenfanfu Jiang[2] Song-Chun Zhu[1]

1 [1] UCLA Center for Vision, Cognition, Learning and Autonomy
[2] UPenn Computer Graphics Group

8. Simulated Annealing

The simulated annealing schedule is important for synthesizing realistic scenes. In our experiments, we set the total sampling iterations to 20000, and it takes around 20 minutes to sample an interior layout. We use the following simulated schedule for sampling:

$$T(t) = \frac{T_0}{\ln(1+t)} \tag{24}$$

where $T(t)$ is the temperature at iteration t. Geman *et al*. [**?**] proved that $T(t) \geq \frac{T_0}{\ln(1+t)}$ is a necessary and sufficient condition to ensure convergence to the global minimum with probability one.

9. Data Effectiveness

We further demonstrate that our data can be utilized to improve performance on two scene understanding tasks: depth estimation and surface normal estimation from single RGB images. We show that the performance of state-of-art methods can be improved when trained with our synthesized data along with natural images.

Depth estimation Single-image depth estimation is a fundamental problem in computer vision, which has found broad applications in scene understanding, 3D modeling, and robotics. The problem is challenging since no reliable depth cues are available. In this task, the algorithms output a depth image based on a single RGB input image.

To demonstrate the efficacy of our synthetic data, we compare the depth estimation results provided by models trained following protocols similar to those we used in normal prediction with the network in [**?**]. To perform a quantitative evaluation, we used the metrics applied in previous work [**?**]:

- Abs relative error: $\frac{1}{N}\sum_p \frac{\left|d_p - d_p^{gt}\right|}{d_p^{gt}}$,
- Square relative difference: $\frac{1}{N}\sum_p \frac{\left|d_p - d_p^{gt}\right|^2}{d_p^{gt}}$,
- Average $\log_{10}$ error: $\frac{1}{N}\sum_x \left|\log_{10}(d_p) - \log_{10}(d_p^{gt})\right|$,
- RMSE : $\sqrt{\frac{1}{N}\sum_x \left|d_p - d_p^{gt}\right|^2}$,
- Log RMSE: $\sqrt{\frac{1}{N}\sum_x \left|\log(d_p) - \log(d_p^{gt})\right|^2}$,
- Threshold: % of d_p s.t. $\max\left(\frac{d_p}{d_p^{gt}}, \frac{d_p^{gt}}{d_p}\right) <$ threshold,

where d_p and d_p^{gt} are the predicted depths and the ground truth depths at the pixel indexed by p, respectively, and N is the number of pixels in all the evaluated images. The first five metrics capture the error calculated over all the pixels; lower values are better. The threshold criteria capture the estimation accuracy; higher values are better.

Table 4 summarizes the results. We can see that the model pretrained on our dataset and fine-tuned on the NYU-Depth V2 dataset achieves the best performance, both in error and accuracy. This demonstrates the usefulness of our dataset in improving algorithm performance in scene understanding tasks.

Surface normal estimation Predicting surface normals from a single RGB image is an essential task in scene understanding since it provides important information in recovering the 3D structure of the scenes. We train a neural network with our synthetic data to demonstrate that the perfect per-pixel ground truth generated using our pipeline could be utilized to improve upon the state-of-the-art performance on a specific scene understanding task. Using the fully convolutional network model described by Zhang *et al*. [46], we compare the normal estimation results given by models trained under two different protocols: (i) the network is directly trained and tested on the NYU-Depth V2 dataset, and (ii) the network is first pre-trained using our synthetic data, then fine-tuned and tested on NYU-Depth V2.

Following the standard evaluation protocol [**?**, **?**], we evaluate a per-pixel error over the entire dataset. To evaluate the prediction error, we computed the mean, median, and RMSE of angular error between the predicted normals and ground truth normals. Prediction accuracy is given by calculating the fraction of pixels that are correct within a

Table 4: Depth estimation with different training protocols.

pre-Train	fine-Tune	Error					Accuracy		
		Abs Rel	Sqr Rel	Log10	RMSE(linear)	RMSE(log)	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
NYUv2	-	0.233	0.158	0.098	0.831	0.117	0.605	0.879	0.965
Ours	-	0.241	0.173	0.108	0.842	0.125	0.612	0.882	0.966
Ours	NYUv2	**0.226**	**0.152**	**0.090**	**0.820**	**0.108**	**0.616**	**0.887**	**0.972**

Table 5: Normal estimation with different training protocols.

pre-train	fine-tune	mean↓	median↓	$11.25^\circ \uparrow$	$22.5^\circ \uparrow$	$30^\circ \uparrow$
NYUv2		27.30	21.12	27.21	52.61	64.72
Eigen [**?**]		22.2	15.3	38.6	64.0	73.9
[46]	NYUv2	21.74	14.75	39.37	66.25	76.06
ours+[46]	NYUv2	**21.47**	**14.45**	**39.84**	**67.05**	**76.72**

threshold t, where $t = 11.25^\circ, 22.5^\circ, 30^\circ$. Our experimental results are summarized in Table 5. By utilizing our synthetic data, the model achieves better performance. The error mainly accrues in the area where the ground truth normal map is noisy. We argue that part of the reason is due to the sensor's noise or sensing distance limit. Such results in turn imply the importance to have perfect per-pixel ground truth for training and evaluation.

10. More Qualitative Results

See page 13-17.