Notes of "Configurable 3D scene Synthesis and 2D Image Rendering with Per-Pixel Ground Truth using Stochastic Grammars"

Jihang Li

October 17, 2019

Contents

Abstract

Devising a learning-based pipeline of algorithms capable of automatically generating and rendering a potentially infinite variety of indoor scenes by using a stochastic grammar, represented as an attributed Spatial And-Or Graph, in conjunction with SOTA PBR.

Synthesizing detailed, per-pixel ground truth data:

- surface depth
- surface normal
- object identity
- material information (detailed to object parts)
- environments (*e.g.*, illuminations and camera viewpoints)

1 Introduction

Current RGB-D dataset limitations:

1. Insufficient labeled RGB-D pairs (*e.g.*, NYU-Depth V2).
2. Manual labeling of per-pixel ground truth information is tedious and error prone.

This work is unique in devising a complete **learning-based** pipeline for synthesizing large scale *learning-based configurable* scene layouts via stochastic sampling with PBR of these scenes with associated per-pixel ground truth. This pipeline has the characteristics:

- By utilizing an attributed S-AOG, the sampling algorithm combines **hierarchical compositions** and **contextual constraints** to enable the generation of 3D scenes with high variability.
- As shown in Figure 1, SOTA PBR is employed.

Since the synthesizing is in a forward manner — by rendering 2D images from 3D scenes containing detailed geometric object models — ground truth information is naturally available.

Contributions:

1. The first work that, for the purposes of indoor scene understanding, introduces a *learning-based configurable* pipeline for generating massive photorealistic images and indoor scenes with per-pixel ground truth.
2. Propose S-AOG for scene generation, which supports **arbitrary addition and deletion** of objects and **modification** of their categories.
3. The first work to provide comprehensive diagnostics w.r.t. algorithm stability and sensitivity to certain scene attributes.
4. Demonstrate the effectiveness of the synthesized scene dataset by advancing the SOTA in the prediction of surface normal and depth from RGB images.

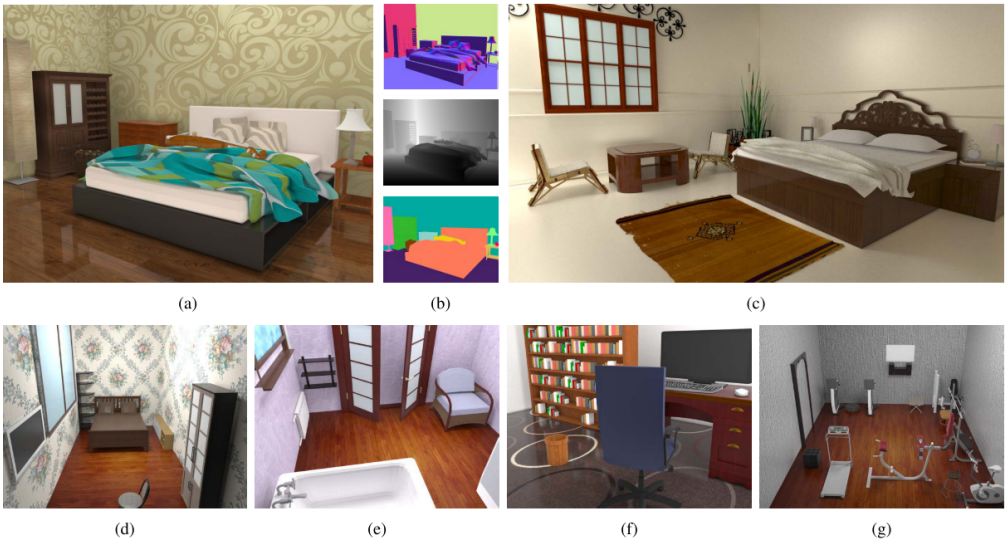

Figure 1: (a) An example automatically-generated 3D bedroom scene, rendered as a photorealistic RGB image, along with its (b) per-pixel ground truth (from top) surface normal, depth, and object identity images. (c) Another synthesized bedroom scene. Synthesized scenes include fine details — objects (*e.g.*, duvet and pillows on beds) and their textures are changeable, by sampling the physical parameters of materials (reflectance, roughness, glossiness, *etc.*.), and illumination parameters are sampled from continuous spaces of possible positions, intensities, and colors. (d) – (g) Rendered images of four other example synthetic indoor scenes — (d) bedroom, (e) bathroom, (f) study, (g) gym.

2 Representation and Formulation

2.1 Representation: Attributed Spatial And-Or Graph

A scene model should be capable of (i) representing the compositional/hierarchical structure of indoor scenes, and (ii) capturing the rich contextual relationships between different components of the scene:

- Compositional hierarchy models the decomposition into sub-components and the switch among multiple alternative sub-configurations.
- Contextual relations:
  - between furniture and walls
  - among furniture
  - between supporting and supported objects
  - between objects of a functional pair

Representation:

An attributed S-AOG, which is a **Stochastic Context-Sensitive Grammar (SCSG)**, combines:

- a **stochastic context-free grammar (SCFG)**
- contextual relations defined on a **Markov random field (MRF)**

Definitions:

An S-AOG is defined as a 5-tuple: $\mathcal{G} = \langle S, V, R, P, E \rangle$, where:

- S: the root node of the scene grammar
- V: the vertex set
  - $V = V_{NT} \cup V_T$
  - $V_{NT} = V^{And} \cup V^{Or} \cup V^{Set}$[1]
  - $V_T = V_T^r \cup V_T^a$
    1. Regular terminal node: $v \in V_T^r$, represents a spatial entity in a scene with **internal attributes** of object sizes (w, l, h), and **external attributes** A_{ext} of object position (x, y, z) and orientation $(x - y$ plane$)$ θ and sampled human positions.
    2. Address terminal node: $v \in V_T^a$, encodes interactions that only occur in a certain context but a absent in all others; point to $v \in V_T^r$ and take values in $V_T^r \cup \{\text{nil}\}$.
- R: the production rules
  - And rules for $v \in V^{And}$:

$$v \to u_1 \cdot u_2 \cdot \ldots \cdot u_{n(v)}. \tag{1}$$

[1] V^{Set}: 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.

– Or rules for $v \in V^{Or}$:

$$v \to u_1|u_2|\ldots|u_{n(v)}, \tag{2}$$

with $\rho_1|\rho_2|\ldots|\rho_{n(v)}$.

– Set rules for $v \in V^{Set}$:

$$v \to (\text{nil}|u_1^1|u_1^2|\ldots)\ldots(\text{nil}|u_{n(v)}^1|u_{n(v)}^2|\ldots), \tag{3}$$

with $(\rho_{1,0}|\rho_{1,1}|\rho_{1,2}|\ldots)\ldots(\rho_{n(v),0}|\rho_{n(v),1}|\rho_{n(v),2}|\ldots)$, where u_i^k denotes that object u_i appear k times, and the probability is $\rho_{i,k}$.

- P: the probability model
- E: the contextual relations, $E = E_f \cup E_o \cup E_g \cup E_w$
  - E_f: relations among furniture
  - E_o: relations between supported objects and their supporting objects
  - E_g: relations in a functional pair
  - E_w: relations between furniture and walls[2]

  E inherited from their parents; hence the relations at a higher level will eventually collapse into cliques $C = C_w \cup C_f \cup C_o \cup C_g$ among the $v \in V_T$. E also form an MRF on $v \in V_T$.

A pt instantiates the S-AOG by selecting a child node for V^{Or} as well as determining the state of each child node for the V^{Set}. And $pg = (pt, E_{pt})$.

2.2 Probabilistic Formulation

The **prior probability** of pg generated by an S-AOG parameterized by Θ is formulated as a Gibbs distribution[3]:

$$p(pg|\Theta) = \frac{1}{Z}\exp(-\mathcal{E}(pg|\Theta)) \tag{4}$$

$$= \frac{1}{Z}\exp(-\mathcal{E}(pt|\Theta) - \mathcal{E}(E_{pt}|\Theta)), \tag{5}$$

where:

- $\mathcal{E}(pg|\Theta)$: the energy function of a pg
- $\mathcal{E}(pt|\Theta)$: the energy function of a pt
- $\mathcal{E}(E_{pt}|\Theta)$: the energy term of the contextual relations

[2]This is different from E_r in "Human-centric Indoor Scene Synthesis Using Stochastic Grammar" (hereinafter referred to as "Human-centric Synthesis").

[3]The style of "E" and "Θ" are different from those in the original paper, but consistent with those in "Human-centric Synthesis" for the convenience of comparison.

$\mathcal{E}(pt|\Theta)$ can be decomposed into:

$$\mathcal{E}(pt|\Theta) = \underbrace{\sum_{v\in V^{Or}} \mathcal{E}_\Theta^{Or}(v) + \sum_{v\in V^{Set}} \mathcal{E}_\Theta^{Set}(v)}_{\text{non-terminal nodes}} + \underbrace{\sum_{v\in V_T^r} \mathcal{E}_\Theta^{A_{in}}(v)}_{\text{terminal nodes}}, \tag{6}$$

where the choice of the child node of an $v \in V^{Or}$ and the child branch of a $v \in V^{Set}$ follows a **Bernoulli distribution**. A_{in} of V_T follows a non-parametric probability distribution learned by **kernel density estimation**. Note that the $v \in V^{And}$ are deterministically expanded; hence Equation 6 lacks an energy term for V^{And}.

$\mathcal{E}(E_{pt}|\Theta)$ combines the potentials of the 4 types of cliques which are computed based on A_{ex} of V_T^r:

$$p(E_{pt}|\Theta) = \frac{1}{Z}\exp(-\mathcal{E}(E_{pt}|\Theta)) \tag{7}$$

$$= \prod_{c\in C_w} \phi_w(c) \prod_{c\in C_f} \phi_f(c) \prod_{c\in C_o} \phi_o(c) \prod_{c\in C_g} \phi_g(c). \tag{8}$$

where:

- $\phi_w(c)$:

$$\phi_w(c) = \frac{1}{Z}\exp\Bigg(-\lambda_w \cdot \Bigg\langle \underbrace{\sum_{w_i\neq w_j} l_{con}(w_i, w_j)}_{\text{constraint between walls}} , \underbrace{\sum_{w_i}[l_{dis}(f, w_i) + l_{ori}(f, w_i)]}_{\text{constraint between walls and furniture}} \Bigg\rangle\Bigg), \tag{9}$$

  where:

  - $c = \{f, \{w_i\}\}$: a clique includes a terminal node representing a furniture f and terminal nodes representing $\{w_i\}$.
  - λ_w: a weight vector.
  - $l_{con}(w_i, w_j)$: defines the consistency between the walls; *i.e.*, adjacent walls should be connected, whereas opposite walls should have the same size. **It is usually zero as the walls are enforced to be consistent in practice.**
  - $l_{dis}(x_i, x_j)$: defines the geometric distance compatibility between two objects

$$l_{dis}(x_i, x_j) = |d(x_i, x_j) - \bar{d}(x_i, x_j)|, \tag{10}$$

    where $d(x_i, x_j)$ is the distance between object x_i and x_j, and $\bar{d}(x_i, x_j)$ is the **mean distance** learned from all the examples.
  - $l_{ori}(x_i, x_j)$: defines the relative orientation compatibility

$$l_{ori}(x_i, x_j) = |\theta(x_i, x_j) - \bar{\theta}(x_i, x_j)|, \tag{11}$$

    where $\theta(x_i, x_j)$ is the distance between object x_i and x_j, and $\bar{\theta}(x_i, x_j)$ is the **mean distance** learned from all the examples.

- $\phi_f(c)$:

$$\phi_f(c) = \frac{1}{Z}\exp\Bigg(-\lambda_f \sum_{f_i\neq f_j} l_{occ}(f_i, f_j)\Bigg), \tag{12}$$

  where:

- $c = \{f_i\} \in C_f$: a clique includes all the terminal nodes representing a piece of furniture.
- $l_{occ}(f_i, f_j)$ defines the compatibility of two pieces of furniture in terms of **occluding accessible space**

$$l_{occ}(f_i, f_j) = \max(0, 1 - d(f_i, f_j)/d_{acc}). \tag{13}$$

- $\phi_o(c)$:

$$\phi_o(c) = \frac{1}{Z} \exp(-\lambda_o \cdot \langle l_{pos}(f, o), l_{ori}(f, o), l_{add}(a) \rangle), \tag{14}$$

where:

- $c = \{f, a, o\} \in C_o$: a clique includes a supported object $o \in V_T$, $a \in V_T^a$ connected to o, and a furniture $f \in V_T$ pointed by a.
- $l_{pos}(f, o)$ defines the relative position of o to the four boundaries of the bounding box of f[4]:

$$l_{pos}(f, o) = \sum_i l_{dis}(f_{face}, o). \tag{15}$$

- l_{ori}: the same as Equation 11.
- $l_{add}(a)$ is the **negative log probability** of a $v \in V_T^a$, treated as a certain $v \in V_T^r$, following a multinomial distribution.

- $\phi_g(c)$:

$$\phi_g(c) = \frac{1}{Z} \exp\left(- \sum_{f_i^g \neq f_j^g} \lambda_g \cdot \langle l_{dis}(f_i^g, f_j^g), l_{ori}(f_i^g, f_j^g) \rangle \right), \tag{16}$$

where $c = \{f_i^g\} \in C_g$ consists of terminal nodes representing furniture in a g.

3 Learning, Sampling, and Synthesis

The configurable synthesis pipeline includes:

- A sampling algorithm based on the learned S-AOG for synthesizing, which controls the size of the individual objects as well as their pair-wise relations.
- An attributes assignment process, which sets different materials attributes to each object part, as well as various camera parameters and illuminations of the environment.

[4] l_{dis} is the same as Equation 10

3.1 Learning the S-AOG

The parameters Θ of the probability model P can be learned in a **supervised** way from a set of N observed parse trees $\{pt_n\}_{n=1,...,N}$ by **maximum likelihood estimation (MLE)**:

$$\Theta^* = \arg\max_{\Theta} \prod_{n=1}^{N} p(pt_n|\Theta). \tag{17}$$

Weights of the Loss Functions: Recall Equation 7

$$p(E_{pt}|\Theta) = \frac{1}{Z}\exp(-\mathcal{E}(E_{pt}|\Theta)) \tag{18}$$

$$= \frac{1}{Z}\exp(-\lambda \cdot l(E_{pt})), \tag{19}$$

where:

- λ: weight vector.
- $l(E_{pt})$: the loss vector given by the 4 types of potential functions.

To learn the weight vector, the standard MLE maximizes the average log-likelihood[5]:

$$\mathcal{L}(E_{pt}|\Theta) = \frac{1}{N}\sum_{n=1}^{N} \log p(E_{pt}|\Theta) \tag{20}$$

$$= -\frac{1}{N}\sum_{n=1}^{N} \langle \lambda, l(E_{pt_n})\rangle - \log Z, \tag{21}$$

usually by gradient descent[6]:

$$\frac{\partial \mathcal{L}(E_{pt}|\Theta)}{\partial \lambda} = -\frac{1}{N}\sum_{N=1}^{n} l(E_{pt_n}) - \frac{\partial \log Z}{\partial \lambda} \tag{22}$$

$$= -\frac{1}{N}\sum_{N=1}^{n} l(E_{pt_n}) - \frac{\partial \log \sum_{pt} \exp(-\lambda \cdot l(E_{pt}))}{\partial \lambda} \tag{23}$$

$$= -\frac{1}{N}\sum_{N=1}^{n} l(E_{pt_n}) + \sum_{pt} \frac{1}{Z}\exp(-\lambda \cdot l(E_{pt}))l(E_{pt}) \tag{24}$$

$$= -\frac{1}{N}\sum_{N=1}^{n} l(E_{pt_n}) + \frac{1}{\widetilde{N}}\sum_{\widetilde{n}=1}^{\widetilde{N}} l(E_{pt_{\widetilde{n}}}), \tag{25}$$

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

Original Texts

Unfortunately, it is 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

[5]Similar to footnote 3.
[6]Different with Equation 15 and 16 in "Human-centric Synthesis".

follows the gradient of difference of two divergences:

$$\mathrm{CD}_{\widetilde{N}} = \mathrm{KL}(p_0 \| p_\infty) - \mathrm{KL}(p_{\widetilde{n}} \| p_\infty), \tag{26}$$

where $\mathrm{KL}(p_0 \| p_\infty)$ is the **Kullback-Leiber** 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, $\widetilde{n} = 1$.

The gradient of CD is given by:

$$\begin{aligned} \frac{\partial \mathrm{CD}_{\widetilde{N}}}{\partial \lambda} = & \frac{1}{N} \sum_{n=1}^{N} l(E_{pt_n}) - \frac{1}{\widetilde{N}} \sum_{\widetilde{n}=1}^{\widetilde{N}} l(E_{pt_{\widetilde{n}}}) \\ & - \frac{\partial p_{\widetilde{n}}}{\partial \lambda} \frac{\partial \mathrm{KL}(p_{\widetilde{n}} \| p_\infty)}{\partial p_{\widetilde{n}}}, \end{aligned} \tag{27}$$

where the third term can be ignored.

Finally, the weight vector is learned by gradient descent computed by generating a small number $\widetilde{n}$[7] of examples from the Markov chain:

$$\lambda_{t+1} = \lambda_t - \eta_t \frac{\partial \mathrm{CD}_{\widetilde{N}}}{\partial \lambda} \tag{28}$$

$$= \lambda_t + \eta_t \left(\frac{1}{\widetilde{N}} \sum_{\widetilde{n}=1}^{\widetilde{N}} l(E_{pt_{\widetilde{n}}}) - \frac{1}{N} \sum_{n=1}^{N} l(E_{pt_n}) \right) \tag{29}$$

Branching Probabilities: The MLE of the branch probabilities ρ_i of V^{Or}, V_T^a, and V^{Set} is the frequency of each alternative choice:

$$\rho_i = \frac{\#(v \to u_i)}{\sum_{n=1}^{n(v)} \#(v \to u_j)}; \tag{30}$$

Original Texts

however, the samples we draw from the distributions will rarely cover all possible terminal nodes to which an address node is pointing, since there are many unseen but plausible configurations. For instance, an apple can be put on a chair, which is semantically and physically plausible, but the training examples are highly unlikely to include such a case. Inspired by the Dirichlet process, we address this issue by altering the MLE to include a small probability α for all branches:

$$\rho_i = \frac{\#(v \to u_i) + \alpha}{\sum_{n=1}^{n(v)} \#(v \to u_j) + \alpha}. \tag{31}$$

For V^{Set}, set α to have probability 1[8].

Parameters: Use SUNCG dataset as training data and collect the statistics of:

[7] It is "$\widetilde{N}$" in "Human-centric Synthesis".

[8] Equation 31 is different with the branching probability equation in "Human-centric Synthesis".

- room types
- room sizes
- furniture occurrences
- furniture sizes
- relative distances
- orientations between furniture and walls
- furniture affordance
- grouping occurrences
- supporting relations

Loss Function: The parameters are learned from the constructed scenes by computing the statistics of **relative distances** and **relative orientations** between different objects.

Grouping Relations: Manually defined; a pair is regarded as a group if the distance of the pieces is smaller than a threshold (*e.g.*, 1m). The supporting relations are automatically discovered by computing the vertical distance between pairs of objects and checking **if one bounding polygon contains another**.

Distribution of Object Size: Learned from the 3D models in ShapeNet and SUNCG. The size information is firstly extracted from the 3D models, then a non-parametric distribution is fitted using **kernel density estimation**[9].

3.2 Sampling Scene Geometry Configurations

Sampling scene configurations is based on the prior probability $p(pg|\Theta)$, using an **Markov Chain Monte Carlo (MCMC)** sampler:

1. Top-down sampling of the pt and A_{in} (sizes). For pt, this step selects a branch for each $v \in V^{Or}$ and a child branch for each $v \in V^{Set}$. This can be done by sampling from **closed-form distributions**.
2. MCMC sampling of A_{ex} (positions and orientations) and the values of $v \in V_T^a$. Samples are proposed by Markov chain dynamics[10]:
   (a) q_1: **translation** of objects. It chooses a regular terminal node, and samples a new position based on the current position,
   $$x \to x + \delta x, \tag{32}$$
   where δx follows a **bivariate normal distribution**.

[9] Another step is mentioned in "Human-centric Synthesis".
[10] Two more moves (*i.e.*, q_3 and q_4) than those in "Human-centric Synthesis".

(b) q_2: **rotation** of objects. It chooses a regular terminal node, and samples a new orientation based on the current orientation,

$$\theta \to \theta + \delta\theta, \tag{33}$$

where $\delta\theta$ follows a **normal distribution**.

(c) q_3: **swapping** of objects. It chooses two regular terminal nodes, and swap the positions and orientations of the objects.

(d) q_4: **swapping** of supporting objects. It chooses an address node and a new regular furniture terminal node pointed to, then sample a new 3D location (x, y, z) for the supported object:

- Randomly sample $x = u_x w_p$, where $u_x \sim \text{unif}(0, 1)$, and w_p is the **width** of the supporting object.
- Randomly sample $y = u_x l_p$, where $u_y \sim \text{unif}(0, 1)$, and w_p is the **length** of the supporting object.
- The **height** z is the height of the supporting object.

where q_1 and q_2 are diffusion, while q_3 and q_4 are reversible jump.

Adopting the **Metropolis-Hastings** algorithm, the proposed new $pg\prime$ is accepted according to the acceptance probability:

$$\alpha(pg\prime|pg, \Theta) = \min(1, \frac{p(pg\prime|\Theta)p(pg|pg\prime)}{p(pg|\Theta)p(pg\prime|pg)}) \tag{34}$$

$$= \min(1, \frac{p(pg\prime|\Theta)}{p(pg|\Theta)}) \tag{35}$$

$$= \min(1, \exp(\mathcal{E}(pg|\Theta) - \mathcal{E}(pg\prime|\Theta))), \tag{36}$$

The proposal probabilities cancel since the proposed moves are **symmetric** in probability[11].

Algorithm 1: Sampling Scene Configurations

Input: Attributed S-AOG $\mathcal{G}$, Landscape parameter β, sample number n
Output: Synthesized room layouts $\{pg_i\}_{i=1,\dots,n}$

for $i = 1$ **to** n **do**
  Sample the child nodes of the Set-nodes and Or-nodes from $\mathcal{G}$ directly to obtain the structure of pg_i.
  Sample the sizes of room, furniture f, and objects o in pg_i directly.
  Sample the Address-nodes V^a.
  Randomly initialize positions and orientations of furniture f and objects o in pg_i.
  $iter = 0$
  while $iter < iter_{max}$ **do**
    Propose a new move and obtain proposal pg_i'.
    Sample $u \sim \text{unif}(0, 1)$.
    if $u < \min(1, \exp(\beta(\mathcal{E}(pg_i|\Theta) - \mathcal{E}(pg_i'|\Theta))))$ **then**
      $pg_i = pg_i'$
    end
    $iter+ = 1$
  end
end

[11] Explained the texts after Equation 23 in "Human-centric Synthesis".

Convergence: A histogram of the energy of the last w samples is used to tell if the Markov chain has converged to the prior probability. When the difference between two histograms separated by s sampling steps is smaller than a threshold ε, then Markov chain is considered to have converged.

Tidiness of Scenes: Control the level of tidiness of the sampled scenes by adding an extra parameter β to control the landscape of the prior distribution[12]:

$$p(pg|\Theta) = \frac{1}{Z}\exp(-\beta\mathcal{E}(pg|\Theta)). \tag{37}$$

Original Texts

Note that the parameter β is analogous to but differs from the temperature in **simulated annealing optimization** — the temperature in simulated annealing is time-variant; *i.e.*, it changes during the simulated annealing process. In our model, we simulate a Markov chain under one specific β to get typical samples at a certain level of tidiness. When β is small, the distribution is "smooth"; *i.e.*, the differences between local minima and local maxima are small.

3.3 Scene Instantiation using 3D Object Datasets

Five steps:

1. For each object in the scene layout, find the model that has the closest **length/width ratio** to the dimension specified in the scene layout.
2. Align the **orientations** of the selected models according to the orientation specified in the scene layout.
3. Transform the models to the specified **positions**, and scale the models according to the generated scene layout.
4. Adjust the object position along the **gravity direction** to eliminate floating models and models that penetrate into on another.
5. Add the floor, walls, and ceiling to complete the instantiated scene.

3.4 Scene Attribute Configurations

The rendered images are determined by combinations of the following 4 factors:

- Illuminations, including the number of light sources, and their positions, intensities, and colors.
- Material and textures of the environment.
- Various cameras types (*e.g.*, fisheye, panorama), F-stop, focal distance, depth of field.
- Different object materials and textures with various roughness, metallicness, and reflectivity.

[12] Recall Equation 4.

4 Photorealistic Scene Rendering

Using Houdini Mantra for PBR.

(The details of this section are omitted.)

5 Experiments

(Omitted.)

6 Discussion

Configurable Scene Synthesis: Benefits:

- Has the potential to generate unlimited training data.
- Can be used to diagnose AT systems. In the future, hopefully such methods can assist in building explainable AI.

  Original Texts

  For instance, in the field of causal reasoning[a], causal induction usually requires turning on and off specific conditions in order to draw a conclusion regarding whether or not a causal relation exists. Generating a scene in a controllable manner can provide a useful tool for studying these problems.

  [a]Pearl, J.: Causality. Cambridge university press (2009).

- May be used to generate various virtual environment in a controllable manner in order to train virtual agents to learn task planning[13] and control policies[14].

The Importance of the Different Energy Terms: Ranking from the largest weight to the smallest:

1. Distances between furniture pieces and the nearest wall.
2. Relative orientations of furniture pieces and the nearest wall.
3. Supporting relations.
4. Functional group relations.
5. Occlusions of the accessible space of furniture by other furniture.

[13][69]A Virtual Reality Platform for Dynamic Human-scene Interaction, [155]Target-driven Visual Navigation in Indoor Scenes using Deep Reinforcement Learning

[14][53]Emergence of Locomotion Behaviours in Rich Environments, [130]Robust Imitation of Diverse Behaviors

Balancing Rendering Time and Quality: (Omitted.)

The Speed of the Sampling Process: To speed up the sampling the improve the synthesis quality, the sampling process is split into 5 stages:

1. Sample the objects on the wall (*e.g.*, windows, switches, paints, lights).
2. Sample the core functional objects in functional groups (*e.g.*, desks and beds).
3. Sample the objects that are associated with the core functional objects (*e.g.*, chairs and nightstands).
4. Sample the objects that are not paired with other objects (*e.g.*, wardrobes and bookshelves).
5. Sample small objects that are supported by furniture (*e.g.*, laptops and books).

Original Texts

By splitting the sampling process in accordance with functional groups, we effectively reduce the computational complexity, and different types of objects quickly converge to their final positions.

7 Conclusion and Future Work

Future work:

- The scene generation process can be improved using a multi-stage sampling process; *i.e.*, sampling large furniture objects first and smaller objects later, which can potentially improve the scene layout.
- Considering modeling human activity inside the generated scenes, especially with regard to functionality and affordance.
- Considering the introduction of moving virtual humans into the scenes, which can provide additional ground truth for human pose recognition, human tracking, and other human-related tasks. Then an ST-AOG is needed, which unlocks the potential to synthesize outdoor environments.
- Domain adaptation techniques is planned to apply, which has been shown to be important in learning from synthetic data[15].

[15] [76]From Virtual to Real World Visual Perception using Domain Adaptation — the DPM as Example, [106]A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes, [123]Unbiased Look at Dataset Bias