DEEP REINFORCEMENT LEARNING FOR FURNITURE LAYOUT SIMULATION IN INDOOR GRAPHICS SCENES

Xinhan Di∗
Digital Technology Department
Longfor Group Holdings Limited
dixinhan@longfor.com

Pengqian Yu∗
IBM Research
Singapore
peng.qian.yu@ibm.com

ABSTRACT

In the industrial interior design process, professional designers plan the size and position of furniture in a room to achieve a satisfactory design for selling. In this paper, we explore the interior graphics scenes design task as a Markov decision process (MDP), which is solved by deep reinforcement learning. The goal is to produce an accurate layout for the furniture in the indoor graphics scenes simulation. In particular, we first formulate the furniture layout task as a MDP problem by defining the state, action, and reward function. We then design the simulated environment and deploy a reinforcement learning agent that interacts with the environment to learn the optimal layout for the MDP. We conduct our experiments on a large-scale real-world interior layout dataset that contains industrial designs from professional designers. Our numerical results demonstrate that the proposed model yields higher-quality layouts as compared with the state-of-art model.

1 INTRODUCTION

People spend plenty of time indoors such as the bedroom, living room, office, and gym. Function, beauty, cost, and comfort are the keys to the redecoration of indoor scenes. Many online virtual interior tools are developed to help people design indoor spaces in the graphics simulation. Machine learning researchers make use of the virtual tools to train data-hungry models for the auto layout (Dai et al., 2018; Gordon et al., 2018), including a variety of generative models (Fisher et al., 2012; Li et al., 2019b).

Reinforcement learning (RL; Sutton & Barto (2018)) consists of an agent interacting with the environment, in order to learn an optimal policy by trial and error for the Markov decision process (MDP; Puterman (2014)). The past decade has witnessed the tremendous success of deep reinforcement learning in the fields of gaming, robotics and recommendation systems (Gibney (2016); Schrittwieser et al., 2020; Silver et al., 2017). Researchers have proposed many useful and practical algorithms such as DQN (Mnih et al., 2013) that learns an optimal policy for discrete action space, DDPG (Lillicrap et al., 2015) and PPO (Schulman et al., 2017) that train an agent for continuous action space, and A3C (Mnih et al., 2016) designed for a large-scale computer cluster. These proposed algorithms solve stumbling blocks in the application of deep RL in the real world.

The models proposed in previous work (e.g., Fisher et al., 2012; Li et al., 2019b; Qi et al., 2018; Wang et al., 2018) for furniture layout only produce approximate size of the furniture, which is not practical for industry use as illustrated in Figure 1. Moreover, these prior work neglect the fact that the industrial interior design process in the simulated graphics scenes is indeed a sequential decision-making process, where professional designers need to make multiple decisions for furniture layout by trial and error.

Motivated by the above challenges, we formulate the furniture layout task as a MDP problem by defining the state, action, and reward function. We then design a simulated environment and deploy a RL agent to learn the optimal layout for the MDP. Our work is related to data-hungry methods for synthesizing indoor graphics scenes simulations through the layout of furniture. Early work in the scene modeling implement kernels and graph walks to retrieve objects from a database (Choi

∗Denotes equal contribution.
Figure 1: Examples of layouts produced by the state-of-the-art models (Wang et al., 2019). These layouts are for bedroom and tatami room. The ground truth layout in the simulator and the real-time renders can be found in the first row. The layouts produced by the state-of-art models are shown in the second and third rows. It can be observed that the state-of-art model produces inaccurate position and size of the furniture.

Figure 2: Simulation environment for the layout of furniture in the indoor scenes.

We highlight our two main contributions. First, we formulate the interior graphics scenes design task as a Markov decision process problem. To the best of our knowledge, this is the first time that the task is studied from a sequential decision-making perspective. Secondly, we develop an indoor graphics scenes simulator and use deep reinforcement learning technique to solve the MDP in the learning of the simulated graphic scenes. The developed simulator and codes are available at https://github.com/CODE-SUBMIT/simulator1

2 PROBLEM FORMULATION

We formulate the planning of furniture layout in the simulation of graphics indoor scenes as a Markov decision process (MDP) augmented with a goal state $G$ that we would like an agent to learn. We define this MDP as a tuple $(S, G, A, T, \gamma)$, in which $S$ is the set of states, $G$ is the goal,
Figure 3: Given the sizes and positions of the walls, windows, doors and furniture in a real room, the developed simulator transfers the real indoor scenes to simulated graphics indoor scenes. Different components are in different colors in the simulation.

\( A \) is the set of actions, \( T \) is the transition probability function in which \( T(s, a, s') \) is the probability of transitioning to state \( s' \in S \) when action \( a \in A \) is taken in state \( s \in S \), \( R_t \) is the reward function at timestamp \( t \), and \( \gamma \) is the discount rate \( \gamma \in [0,1) \). At the beginning of each episode in a MDP, the solution to a MDP is a control policy \( \pi : S, G \rightarrow A \) that maximizes the value function

\[
V_\pi(s, g) := \mathbb{E}_\pi \left[ \sum_{t=0}^{\infty} \gamma^t R_t | s_0 = s, g = G \right]
\]

for given initial state \( s_0 \) and goal \( g \).

In this paper, we define the states \( S = (S_w, S_{\text{win}}, S_d, S_f) \) where \( S_w \) consists of the geometrical position \( p \) and the size \( l \) of the wall. Here \( p \) is the location for the center of the wall, and the size \( l \) consists of the width and height of the wall. \( S_{\text{win}}, S_d \) and \( S_f \) are defined similarly for the window, door and furniture. The goal state \( G \) describes the correct position \( p \) and direction \( d \) of the furniture. The action set \( A \) is discrete (left, right, up or down) and it specifies the direction of the furniture should move towards in the indoor scenes. The reward function is designed to encourage the furniture to move towards the correct position. It is defined as the following:

\[
R := \theta \text{IoU}(f_{\text{target}}, f_{\text{state}})
\]

where \( \theta \) is a positive constant, \( f_{\text{target}} \) and \( f_{\text{state}} \) represent the ground truth and the current layouts for the furniture in the indoor scenes. \( \text{IoU} \) computes the intersection over union between \( f_{\text{state}} \) and \( f_{\text{target}} \).

The state, action, and reward are illustrated in Figure 2.

3 SIMULATION ENVIRONMENT

Given the sizes and positions of the walls, windows, doors and furniture in a real room, we develop a simulator to transfer the real indoor scenes to simulated graphics indoor scenes as illustrated in Figure 3. The RL environment is a simulator \( \mathcal{E} \) where \( (s', R) = \mathcal{E}(s, a) \), \( a \in A \) is the action from the agent in the current state \( s \in S \), \( s' \in S \) is the next state and \( R \) is the reward associated with the action \( a \). In the next state \( s' \), the geometrical position and size of walls, doors, and windows are not changed, and only the geometrical position of the furniture is updated according to the action. Recall that the action space is discrete and the center of the furniture can move left, right, up and down in each timestamp. The simulator \( \mathcal{E} \) drops the action if furniture moves out of the room, and the furniture will stay in the same place. As the action space is discrete, we adapt the DQN algorithm (Mnih et al., 2013) for the RL agent in the simulation. The agent network contains three convolution layers with 8, 16, and 32 hidden features and 2 fully connected layers with 512 and 4 hidden features. We train the agent using 9 episodes where each episode the agent is initialized at random states in the indoor simulation environment.

4 EXPERIMENTS

We discuss the qualitative and quantitative results in this section. In our numerical experiments, four main types of indoor rooms including the bedroom, bathroom, study room and kitchen are evaluated.

We compare our proposed model with the state-of-art models including the PlanIT (Wang et al., 2019) and the LayouGAN (Li et al., 2019a). Note that we do not compare with layoutVAE (Jyothi et al., 2019a) and NDN (Lee et al., 2020) since they generates outputs in a conditional manner.
Figure 4: Given a bathroom with random furniture positions, the trained RL agent is able to produce a good layout for the bathroom graphics scenes. The first row represents the the ground truth layout for a bathroom in the simulation and its corresponding render. The second row represents the bathroom with random furniture positions. The third row represents the final layouts produced by the proposed method. The fourth row represents the corresponding layout renders.

Table 1: IoU scores for various models.

<table>
<thead>
<tr>
<th>Room</th>
<th>PlanIT</th>
<th>LayoutGAN</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bathroom</td>
<td>0.623 ± 0.008</td>
<td>0.651 ± 0.010</td>
<td>0.961 ± 0.014</td>
</tr>
<tr>
<td>Bedroom</td>
<td>0.647 ± 0.009</td>
<td>0.648 ± 0.017</td>
<td>0.952 ± 0.026</td>
</tr>
<tr>
<td>Study</td>
<td>0.619 ± 0.006</td>
<td>0.662 ± 0.014</td>
<td>0.957 ± 0.018</td>
</tr>
<tr>
<td>Kitchen</td>
<td>0.637 ± 0.006</td>
<td>0.639 ± 0.012</td>
<td>0.948 ± 0.049</td>
</tr>
</tbody>
</table>

We also conduct a two-alternative forced-choice (2AFC) perceptual study to compare the images from generated scenes with the corresponding scenes from the sold industrial solutions. The generated scenes are generated from our models, PlanIT (Wang et al., 2019) and LayoutGAN (Li et al., 2019a), respectively. Ten professional interior designers were recruited as the participants. The scores are shown in Table 2 and the examples generated by our proposed method are shown in Figure 4. It can be concluded that our proposed approach outperforms the state-of-art models.

Table 2: Percentage (± standard error) of 2AFC perceptual study for various models where the real sold solutions are judged more plausible than the generated scenes.

<table>
<thead>
<tr>
<th>Room</th>
<th>PlanIT</th>
<th>LayoutGAN</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bathroom</td>
<td>79.61 ± 4.12</td>
<td>78.23 ± 6.17</td>
<td>61.08 ± 2.71</td>
</tr>
<tr>
<td>Bedroom</td>
<td>75.29 ± 3.89</td>
<td>79.15 ± 4.93</td>
<td>69.35 ± 2.83</td>
</tr>
<tr>
<td>Study</td>
<td>77.42 ± 5.92</td>
<td>82.03 ± 4.32</td>
<td>62.74 ± 5.72</td>
</tr>
<tr>
<td>Kitchen</td>
<td>78.13 ± 6.92</td>
<td>83.17 ± 5.98</td>
<td>64.51 ± 2.79</td>
</tr>
</tbody>
</table>
5 Discussion

In this paper, we tackle the interior graphics scenes design problem by formulating it as a sequential decision-making problem. We further solve the problem using a deep reinforcement learning agent that learns to produce high quality layout. It is worthwhile to extend our work and consider planning for multiple furniture layouts or learning in 3D space.

References


Akash Abdu Jyothi, Thibaut Durand, Jiawei He, Leonid Sigal, and Greg Mori. Layoutvae: Stochastic scene layout generation from a label set. In International Conference on Computer Vision (ICCV), 2019a.

Akash Abdu Jyothi, Thibaut Durand, Jiawei He, Leonid Sigal, and Greg Mori. Layoutvae: Stochastic scene layout generation from a label set. In The IEEE International Conference on Computer Vision (ICCV), October 2019b.


