Deep Generative Modeling for Scene Synthesis via Hybrid Representations （混合表示）

我们的目标是使用前馈神经网络训练一个生成模型，该网络将先验分布（例如，正态分布）映射到室内场景中主要物体的分布。我们在基准数据集上展示了我们的场景表示和网络训练方法的有效性。我们还展示了这个生成模型在场景插值和场景完成中的应用。

> We demonstrate the effectiveness of our scene representation and the network training method on benchmark datasets. We also show the applications of this generative model in scene interpolation and scene completion

1. Introduction

在计算机图形学中，自动生成场景是一个长期存在的问题。传统的实现方法是聚焦于递归，从一个根物体开始，不断往场景中插入新物体（考虑空间约束），而最近流行使用神经网络model the recurrent synthesis procedure 。但这不是最好的解决方法，因为它们无法在低功耗的情况下对于新插入的物体进行适应，此外，这些方法并没有明确地建立一个映射，例如从潜参数空间（ latent parameter space ）到三维场景空间的映射，这使得它们不适合场景插值和场景外推等应用。

这个神经网络的输入：根据先验分布对潜参数空间进行随机采样(对场景的低维编码)；输出：物体组合形成的3D场景

挑战：包括对3D场景进行参数化，以编码3D场景的连续和离散变化；开发合适的神经网络和训练程序以捕捉多个对象之间的几何相关性；以及从无组织的3D场景中进行训练，这些场景并不明显具有一致的方向和标度。应对挑战的三个关键：representations of 3D scenes, network design and training（2.本文的主要贡献）, and joint learning of scene generators and poses of the scenes used for training the generator（3.如何从无组织的场景集合中训练生成器）

将3D场景参数化成一个矩阵，其中一列代表这个物体出现与否。如果出现，则指定其几何属性（其他元素指定其位置、方向、大小和几何属性）。基于这种矩阵编码，引入稀疏密集生成网络（a sparse dense generative network）来生成三维场景。这种网络设计有效地解决了全连接网络中存在的过拟合问题，同时保持了网络的表达能力。

为了进一步提高生成的3D生成器的质量，我们通过结合两个损失项来训练网络：第一个是标准的基于矩阵的VAE-GAN loss（基于对象排列）；第二个损失项将生成的场景投影到一个图像域中，并使用具有卷积层的鉴别器来捕获相邻物体之间的几何关系（基于图像）。

目前：使用SUNCG数据集将方法应用到客厅和卧室的生成中。客厅和卧室类别分别包含6401和8054个场景。对于每个类别，我们分别使用6,000和7,000个场景进行训练，剩下的场景作为测试。生成一个场景大约30ms。

> Our approach trains 3D scene generators in 1,001 minutes and1,198 minutes, respectively, using a desktop with 3.2GHZ CPU,64GB main memory, and a GTX 1080 GPU.

RELATED WORKS

建模具有显著几何、拓扑可变性的复杂对象是非常困难的。由于这个原因，许多物体场景的参数化模型(例如，家具形状和场景)不存在。从数据中学习参数模型是最近计算机视觉的一个研究热点：之前的研究对象都是人体和人脸，参数化模型知识简单的对数据库里的数据进行线性插值，明显不适用于室内场景这种场景。

现有的建立参数化三维模型的方法大多集中在三维形状上，而不是由多个物体组成的三维场景。而拓展它们到三维场景也是不容易的，比如：三维物体的生成要求方向的一致性，而三维场景需要方向的一定随机性。此外，场景的优化和模型的优化存在一定的互斥。

> Assembly-based geometric synthesis （具体详见论文第三页，感觉没什么用就快速扫过去了）

从训练数据中学到的先验知识也可以用于三维场景的矫正。[Yu et al. 2011]提出了一个优化框架，将粗糙的对象排列转化为显著改进的对象排列。我们基于图像的鉴别器loss在概念上与这种方法相似，但是我们自动地从数据中学习这个loss。

OVERVIEW

Problem Statement

在这篇文章中，将一个场景表现为一组以有语义意义的方式排列在空间中的刚性物体，并且没有相互渗透。我们假设每个对象属于预定义的一组对象类中的一个，并且每个类中的对象可以由形状描述符参数化（然后使用该描述符从形状数据库检索对象的3D几何图形）。

我们进一步假设物体停留在地面上并且正确垂直于地面，这样每个物体在场景中的位置可以通过xy平面(俯视图)调整方向、位置以及缩放来指定。（当然，数据库的数据不需要都这么严格，这个网络具有足够的健壮性）

Approach Overview

矩阵表示下，典型的策略是利用全连通层来设计生成网络。然而，这容易导致过拟合。我们提出了两个关键的创新。第一种方法是使用稀疏密集网络对三维场景进行编码，极大地提高了泛化性能。第二是将生成的三维场景投影到一个图像域，即垂直方向的地平面上，使用具有卷积层的鉴别器（discriminator with convolution layers ）来捕获输入对象之间的细粒度空间相关性

最后，为了处理无组织的场景集合，我们为训练集中的每个场景引入了一个位姿变量，并执行交替最小化来优化场景位姿和3D场景生成器及相关变量(如鉴别器)。每个组件的设计和动机：

- Object arrangement scene representation：虽然每个矩阵完全指定了一个3D场景，但这种表示不是唯一的。为了处理这种编码的非唯一性(即，打乱相同类别的列导致相同场景)，我们引入了潜在的排列变量，可以有效地排除这种排列变异性。
- Scene generator ： 我们将场景产生器设计为一个前馈网络，在场景的矩阵表示上有稀疏交叉层和全连通层。
- Image-based module ： We leverage a CNN-based discriminator loss。我们将基于cnn的image-based discriminator loss加在场景俯视图上，然后将其反向传播到场景生成器，迫使它更准确地学习局部相关模式。
- Joint scene alignment.（联合场景校准） ： 从无组织的场景集合中训练网络是困难的。每个场景都不可能完全按照规则来，此外，尽管对象可以按类分组，但同一个类中的对象没有规范的顺序。作者发现，首先aligning the input scenes 可以显著提高3D场景生成器的效果
- Network training ： 我们通过优化一个目标函数来学习生成器，该函数结合了一个自动编码器loss和之前的两个loss。变量包括生成网络、两个鉴别器、每个场景的姿态(pose)以及每个三维场景中

物体的顺序。为了便于优化，为场景引入了一个潜在变量，它描述了场景的底层配置（loss都在这个变量中）。

APPROACH

Scene Representation

首先，假设有$n_c = 30$个物体类别，每个类别取出$m_k = 4$个物体，那么明显矩阵的维数等于物体的总数。我们假设每个类中的对象都可以用d维形状描述符唯一地标识，所有类都使用d常数，矩阵的列向量v^o描述了一个物体，除了存在，位置，方向，大小之外，从v_9^0到v_{d+8}^o是物体形状的描述符（使用的别人预训练好的网络的倒数第二层的输出）

这种直观的3D场景编码的一个技术挑战是，它不受M_k列的排列的影响（同一类物体的顺序）。此外，每个物体的位置和方向依赖于每个场景的全局姿态。根据这个观察，我们在矩阵编码M上引入了两个算子。第一个算子（operator）将排列表k应用于每个类：(S独立地对每个类的对象应用排列)：个人感觉作用是给每个类的物体加上有序性。

$$S(M;\sigma_1,\ldots,\sigma_{n_k}):$$

$$\mathbb{R}^{(d+9)\times n_o}\times\prod_{k=1}^{n_c}S_{m_k}\rightarrow\mathbb{R}^{(d+9)\times n_o}$$

$$[\,M_1\quad\cdots\quad M_{n_c}\,]\mapsto[\,M_1\sigma_1\quad\cdots\quad M_{n_c}\sigma_{n_c}\,]$$

第二个算子给予每个物体一定的平移和旋转。我们通过为每个场景i引入一个潜伏矩阵编码$\overline{M_i}$∈R(d+9)×no来因子化（factor out）对象的排列组合和每个输入场景的全局姿态，下面描述的反编码器和discriminator loss将施加在$\overline{M_i}$上，我们通过最小化下面的损失项来执行Mi和$\overline{M_i}$的一致性。

$$f_d\left(\overline{M}_i,M_i\right)=\min_{\mathcal{T}_i,\mathcal{S}_i}\left\|\overline{M}_i-\left(\mathcal{T}_i\circ\mathcal{S}_i\right)\left(M_i\right)\right\|_F^2$$

3D Object Arrangement Module

作者网络的心脏是如下两个网络：

由于我们对3D场景的矩阵编码本质上是一种矢量化的表示(与基于图像的表示相反，行向量与列向量)，对于生成器（generator）网络和编码器（encode）网络，使用FC是很自然的。但是我们观察到，FC直连FC几乎不起作用，且容易对训练数据进行过拟合，从而导致生成的场景质量较差。如下图（FC，卷积层，SC，SC+Image-Based）

此外，我们使用卷积层代替fc类型的层。但实验表明，这种方法不能学习成对的对象关系(参见上图第二列）。为了解决这个过拟合的问题，使用稀疏连接的层。每一层的每个节点连接到上一层的h个节点，在作者的实现中，设置h = 4，并将连接随机化，即每个节点独立连接上一层的一个节点，其连接概率为h/L，其中L为上一层节点的数量。如下图（编码器Net），网络在稀疏连接的层和完全连接的层之间交错。仍然保留一些完全连接的层，以使网络具有足够的表达能力来进行网络拟合。

由于我们对3D场景的矩阵编码本质上是一种矢量化的表示(与基于图像的表示相反，行向量与列向量)，对于生成器（generator）网络和编码器（encode）网络，使用FC是很自然的。但是我们观察到，FC直连FC几乎不起作用，且容易对训练数据进行过拟合，从而导致生成的场景质量较差。如下图（FC，卷积层，SC，SC+Image-Based）

此外，我们使用卷积层代替fc类型的层。但实验表明，这种方法不能学习成对的对象关系(参见上图第二列）。为了解决这个过拟合的问题，使用稀疏连接的层。每一层的每个节点连接到上一层的h个节点，在作者的实现中，设置h = 4，并将连接随机化，即每个节点独立连接上一层的一个节点，其连接概率为h/L，其中L为上一层节点的数量。如下图（编码器Net），网络在稀疏连接的层和完全连接的层之间交错。仍然保留一些完全连接的层，以使网络具有足够的表达能力来进行网络拟合。

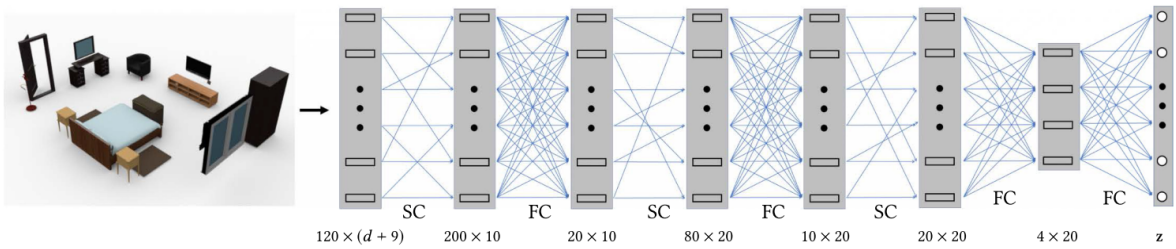

Fig. 4. This figures illustrates the network module that is used for the encoder $\mathcal{G}_{\theta_E}$. The decoder $\mathcal{G}_\theta$ is reversed from the encoder $\mathcal{G}_{\theta_E}$. The arrangement discriminator $\mathcal{D}_\phi$ shares the same network architecture but replaces the latent vector by a value. This network module interweaves between sparsely connected (or SC) layers and fully connected (or FC) layers.

为什么使用稀疏层作为Decode Net，有两个原因：第一：3D场景中的模式通常涉及到小组对象，例如椅子和桌子，或者床头柜和床，因此期望对象类之间的稀疏关系；第二，避免过拟合。

遵循深度卷积生成对抗网络(DCGAN)，作者将Decode循环的架构设置为与Encode循环的架构相反。使用VAE-GAN对编码器和解码器网络进行训练：

the latent distribution p是标准正态分布， and the discriminator Dϕ 和编码器网络有着相同的结构。

Image-based Module

正如在概述中所讨论的，我们介绍了基于图像的鉴别器，以更好地捕捉基于几何细节的物体的局部排列，例如在生成的场景中桌椅之间的空间关系和床头柜与床之间的空间关系。如之前图第三列所示，在没有此模块的情况下，场景生成器出现了各种本地兼容性问题(例如，对象相互交叉）。

鉴于CNN能够很好的捕捉邻近物体之间的局部交互模式，我们俯视场景来进行场景降维，然后在其上使用CNN（卷积神经网络）（这里使用的是ResNet-18 ）。D_{ϕ_I}作为图像表示的鉴别器，ϕ_I代表了网络参数。We then use the following discriminator loss for learning the object arrangement generator：

$$f_I = \frac{1}{N}\sum_{i=1}^{N} \mathcal{D}_{\phi_I}\left(\mathcal{P}\left[\overline{M}_i\right]\right) - E_{\mathbf{z}\sim p}\mathcal{D}_{\phi_I}\left(\mathcal{P}\left[\mathcal{G}_\theta(\mathbf{z})\right]\right).$$

$$\mathcal{P} : \mathbb{R}^{(d+9)\times n_o} \rightarrow \mathbb{R}^{r\times r}$$

r = 128, and details of the projection operator are described in detail below。我们根据投影到顶视图中的无符号距离的总和定义了一个模糊（fuzzy）的投影operator（上图类似P的符号）。此外，对于每个物体o，让$E_o(M)$作为该物体在平面内占据的点的集合。

> Although it is possible to use a rendering operator for the projection $\mathcal{P}$, as as described by Wang et al. [2018a], we want the image-based discriminator $\mathcal{D}_{\phi_I}$ to provide smooth gradients for the generator $\mathcal{G}_\theta$, and such gradients are hard to compute, even when using very simple rendering operations. We therefore instead define a fuzzy projection operator $\mathcal{P}$ in terms of summed truncated signed distance fields of objects projected into the top view. Specifically, for each object o, let $E_o(M)$ denote the set of points in the plane computed by (1) embedding object o in 3D as encoded by the parameters in M, and (2) orthogonally projecting that object onto the xy plane. Denote the truncated signed distance function of object o by

其中，c_o是一个与物体关联的常量，这里，仅仅使用物体分类的编号，

Joint Scene Alignment

作为预处理步骤，我们通过为每个场景分配一个刚性变换和一组排列来对齐所有输入训练场景，我们首先执行成对匹配，然后将这些成对匹配聚合为所有场景的一致全局对齐。这种两步方法的常见特征是，第二步可以有效地删除第一步产生的嘈杂的成对匹配，导致高质量的对齐。在我们的例子中，同时优化场景中的每个物体是棘手的（庞大的数据量），We therefore propose to align the input scenes in a sequential manner by first optimizing rotations, then translations, and finally permutations

Pairwise matching:

Pairwise matching. Given a pair of scenes M^i and M^j, we solve the following optimization problem to determine the optimal transformation $\mathcal{T}_{ij}^{in} = (R_{ij}^{in}, \mathbf{t}_{ij}^{in})$ aligning M_i to M_j, as well as permutations $\mathcal{S}_{ij}^{in}$ mapping objects of each class in M^i to their closest match in M^j:

$$\mathcal{T}_{ij}^{in}, \mathcal{S}_{ij}^{in} = \underset{\mathcal{T},\mathcal{S}}{\operatorname{argmin}} \left\| (\mathcal{T} \circ \mathcal{S}) \left(M^i\right) - M^j \right\|_{2,1}, \tag{5}$$

where $\|A\|_{2,1} = \sum_{j=1}^{m} \|\mathbf{a}_j\|_\star, A := (\mathbf{a}_1, \ldots, \mathbf{a}_m)$ is a robust norm used to handle continuous and discrete variations between M_i and M_j.

Consistent scene alignment.

解决第一步产生的对齐噪声问题。

Network Training

上述公式很难求解，因为目标函数是非凸的(即使当鉴别器是固定的)。我们再次应用交替最小化优化，以便每一步解决一个更容易的优化子问题。

EXPERIMENTAL EVALUATION

Experimental Setup

正如之前所言，使用了来自SUNCG数据库中的bedroom和living rooming模型

Experimental Results

总体上，生成场景的质量接近作者的预期，在场景中的对象数量、空间布局和相关的对象组上显示出很大的变化。图1比较了生成的场景和训练数据中最接近的场景。在这里，我们简单地使用隐藏场景空间的欧氏距离来计算最近的场景。我们可以看到，生成的场景在空间对象布局和对象存在方面表现出明显的变化。这意味着我们的方法在训练数据中学习有意义的模式并使用它们进行场景合成，而不是仅仅记忆数据

Perceptual Study

进行了一项用户研究，以评估方法视觉质量。具体来说，对于每种方法和每种场景类型，我们生成20个场景。对于每个场景，我们提取训练数据中最近的场景。然后我们将这20对图片展示给用户，让他们选择他们认为是生成的场景。每项研究都使用一对百分比(a, 100−a)进行总结，其中a表示合成数据中的场景被标记为生成的百分比。

我们的方法的表现好过其他方法（有选择的去掉稀疏层的使用和Imaged-based的使用）。我们的方法在卧室和客厅分别达到了61.4%/38.6%和55.6%/44.4%。考虑到训练数据大部分是由高质量的用户设计的场景组成，这些数字是相当令人鼓舞的，超过30%的时候用户更喜欢我们的合成结果而不是用户设计的场景。

What Was Learned

在这一节中，我们通过研究神经网络学到的东西来分析我们的方法的性能。我们的协议是评估网络是否学习了训练数据中关于对象和对象对的重要分布，即生成的场景是否有类似于训练数据的分布。

对象的成对关联。我们首先评估对象之间的重要成对分布是否被我们的生成器正确地学习。我们画出第二个物体和第一个物体之间的相对位置的分布。为了简单起见，我们只在x-y平面(或顶视图)上绘制边缘分布，它捕获了大多数信号。在这个实验中，我们考虑桌子/椅子，床/床头柜，床/电视，椅子/电脑作为卧室，沙发/桌子，桌子/电视，植物/沙发，沙发/电视为客厅。如果一个场景中有多个对，我们只提取空间距离最近的对

下图显示了成对物体前方向之间的相对角度的分布。

The Importance of Joint Scene Alignment

在本节中，我们将进行额外的研究，以证明联合优化3D场景作为预处理步骤的重要性。

Global scene alignment.：取消全局场景对齐步骤(即直接在原始输入数据上应用我们的交替最小化程序)会导致网络无法学习显著模式之间的关联。生成的场景上的绝对位置分布与训练数据上的绝对位置分布有显著差异。这证明了全球场景对齐对我们系统的成功至关重要。换句话说，用局部公式来对齐输入场景是不够的。

Applications in Scene Interpolation

详见论文

Applications in Scene Completion

详见论文

CONCLUSIONS

为了最大限度地权衡利弊，提出了一种混合方法，通过结合3D对象排列表示和投影2D图像表示来训练3D场景生成器，并结合两种表示的优点。三维对象排列表示保证了合成场景的局部和全局邻域结构，而基于图像的表示保留了局部视图依赖的模式。此外，基于图像的表示所得到的结果有利于训练三维生成器。

我们的3D场景生成器是一个前馈神经网络。该网络设计借鉴了常用的三维场景合成和建模的递归方法。前馈架构的好处是，它可以联合优化3D合成的所有因素，而递归方法很难从顺序处理过程中出现的错误中恢复。初步的定性评估显示了前馈架构优于两种循环方法的优势。虽然说前馈方法将在循环方法中占主导地位还为时过早，但我们相信前馈网络已经在几个场景中显示了巨大的前景，值得进一步的研究和开发。

我们的方法的一个限制是，所有的训练数据应该由语义分割的3D场景组成，这并不总是可能的，例如，从点云中重建的3D场景通常不会被分割成单个物体。解决这个问题的潜在方法是扩展本文描述的一致混合表示，例如，通过强制三个网络之间的一致性--(1)基于图像表示下的场景合成，(2)3D对象排列表示下的场景合成，以及(3)将3D场景转换为其对应的3D对象排列表示的网络。

此外，我们的方法限制了生成场景中合成实例的数量。这种策略可能会给大型场景中的对象安排带来问题。一个潜在的解决方案是扩展我们的模型，以便它可以为现有的场景布局合成新的对象安排。对于大型场景，我们可以反馈生成的场景，以获得更多的合成对象安排。我们将此作为未来的研究。

我们的方法使用形状描述符在数据库中查询合适的形状，以确定每个合成场景中物体的形状。这种方法的一个限制是，在对象级别上，我们不创建新的形状