NeRF in the wild 复现

李梓维

摘要

摘要。

该论文提出一种基于学习的方法，仅使用非结构化标准化的互联网图片进行户外建筑的三维重建。基于 NeRF（Neural Radiance Fields）[1]，利用多重感知机（MLP）的权重将场景的密度与颜色建模为三维坐标函数。虽然 NeRF 能实现照片级的重建，但是数据集的光照一致性跟遮挡物都会严重影响重建质量，使得重建模型产生伪影等情况。该论文以 NeRF 为基础提出 NeRF-W 模型，解决了 NeRF 在户外条件下受限的问题。经实验证明该模型能使用互联网上的游客照片实现高质量的三维重建。本复现基于 NeRF in the wild[2]的基础上添加了 Instant-NGP 的多分辨率哈希编码方法替换位姿编码对该模型的训练速度进行改进，经实验证明训练速度得到较大提升。

关键词：三维重建；计算机视觉；计算机图形学

1 引言

从稀疏的图像数据集合成新视图是三维重建领域长期问题，是 VR、AR 应用的基础条件。尽管传统的技术已使用基于 SFM 和 IBR 方法解决该问题，但是神经渲染技术在最近得到重大进展。其中神经辐射场（NeRF）[1]利用 8 层 MLP 权值实现三维空间坐标到密度与颜色的映射进行建模。体渲染技术用于新视图的合成，实现照片级的渲染图片在一系列工作中脱引而出。然而，NeRF 存在几大问题。

其中 NeRF 实现一个 lego 模型建模需要花费 3080 显卡 30 个小时。NSVF[3]、Plenosels[4]、Instant-NGP[5]等工作利用空间体素网格化及载入特征向量等方法实现训练时间大大缩短。NeRF 需要数据集保证位姿信息的准确性、数据集图片光照一致性、数据集对象单一无遮挡，其中一项出现问题都会导致重建效果不理想。其中 iNeRF[6]、BARF[7]等工作实现数据集位姿信息的纠正提高重建质量。而本文所复现的论文则通过基于 NeRF 网络模型提出 NeRF-W 网络模型解决 NeRF 要求数据集需要保证图像在一致光照和无遮挡物这一苛刻条件导致无法在户外实现高质量三维重建问题。为后续 NeRF 城市级三维重建的实现打下基础。

2 相关工作

本工作主要为分为两个部分，一是实现 NeRF 的基础模型，二是以 NeRF 为基础加入 appearance embedding 跟 transient embedding 实现统一光照一致性跟去除遮挡物影响。

2.1 NeRF

NeRF 发表于 2020ECCV，是当年的最佳论文提名，但影响力胜于最佳论文。该工作提出了神经辐射场，与传统图形学体渲染技术相结合产生了三维重建领域的一个新的思路。该方法输入为多视图的 rgb 图像，输出为神经辐射场如图 1。通过体渲染技术渲染出连续的图片实现三维重建效果。相比于传统的其他三维重建算法，nerf 最大的优点是实现了照片级的渲染，在追求真实渲染效果的三位重建领域有很大的应用前景。

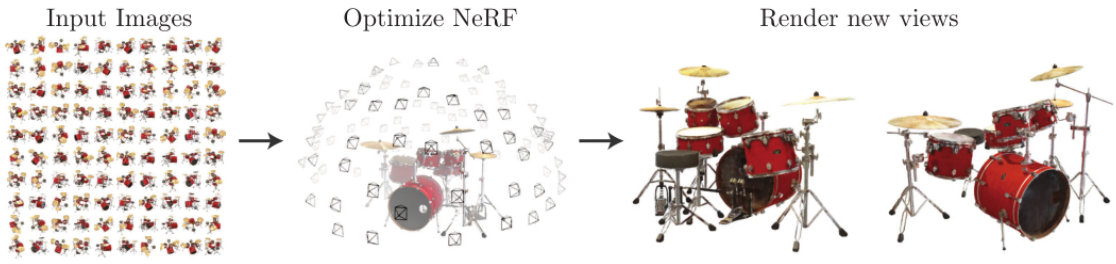

图 1: NeRF 输入输出

NeRF 的数据集在训练时必须要有 5 维的位姿信息（坐标：x，y，z；方向：θ，φ），若位姿信息误差较大，则会严重影响重建质量。这一步往往通过另一篇工作 llff 的 image2pose 函数或者 colmap 完成。

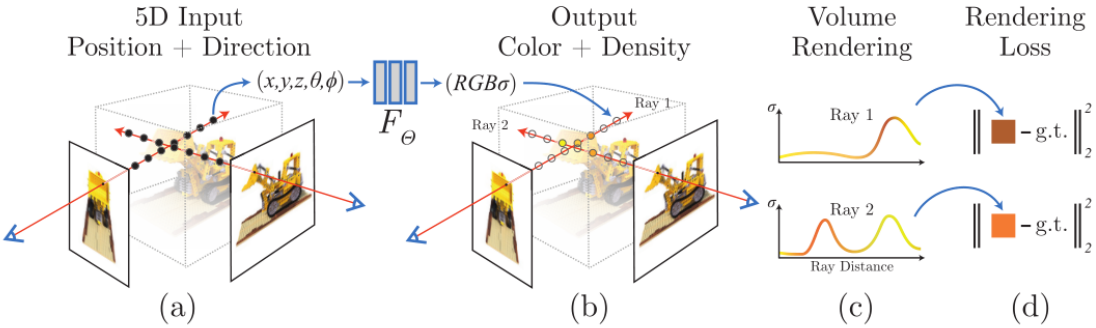

图 2: NeRF 光线追踪和体渲染

NeRF 在训练时会先建立一个三维空间即世界坐标，在三维空间中将拥有位姿信息的输入图片恢复位置，后面就是最为关键的体渲染训练环节如图 2。体渲染：NeRF 首先会在图片的后方生成一个光线发生器，光线发生器射出光线穿过图像的一个像素进入三维空间。沿着光线进行采样，获取 n 个点，每个点都有各自的位姿信息（x，y，z，θ，φ），通过神经网络 MLP 实现（x，y，z，θ，φ）->（r，g，b，σ）映射，其中 r，g，b 代表颜色，σ 代表密度即权重。最后通过体渲染方程（如图 4将该光线所有采样点进行积分，得到的结果为渲染出来的像素与真实图片的像素求 loss 如图 5实现一个神经网络的训练。

$$\gamma(p) = \left(\sin\left(2^0\pi p\right), \cos\left(2^0\pi p\right), \cdots, \sin\left(2^{L-1}\pi p\right), \cos\left(2^{L-1}\pi p\right)\right)$$

图 3: 位姿编码

$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t), \mathbf{d})dt\,, \text{ where } T(t) = \exp\left(-\int_{t_n}^{t} \sigma(\mathbf{r}(s))ds\right)$$

图 4: NeRF 光线追踪和体渲染

$$\mathcal{L} = \sum_{\mathbf{r}\in\mathcal{R}}\left[\left\|\hat{C}_c(\mathbf{r}) - C(\mathbf{r})\right\|_2^2 + \left\|\hat{C}_f(\mathbf{r}) - C(\mathbf{r})\right\|_2^2\right]$$

图 5: l2loss

2.2 Instant-NGP

在 NeRF 中 positional encoding 的提出解决了 MLP 网络无法重建出高频细节问题，但是光线中的每个采样点都需要经历 8+1 层 256 节点的 MLP 网络训练，positional encoding3将 5 维位姿信息编码为 90 维向量一定程度都导致 NeRF 建模代价非常的高，建一个 lego 模型需要 rtx3080 花费 30 个小时。同时在基于深度学习的图形学任务中，每个工作都针对自己特定的 task，设计不同的网络结构，这样的缺点是这些方法只限制在特定的任务上，同时这些工作在优化整个网络的过程中，需要对整个网络进行优化，这加大了网络的花费。

本文提出一种基于多分辨率的哈希编码方案能使的 NeRF[1]的 MLP 网络从 8 层缩减为 2 层，这个方案使得 NeRF 建图的时间降低为秒级别（CUDA 编程），若用 pytorch 复现则为 5 分钟，并且该方案是通用型的方案，可以在不同的工作中得到应用。该方法与采用 LOD 方法的 SDF[8]有一定相似。

Instant-NGP 会在三维空间中建立多分辨率体素网格，将 NeRF 中的空间信息利用起来存放信息。如图 9所示为二维空间中的多分辨网格举例（3x3 和 2x2 的多分辨网格），在这个过程中，每个层是独立的 (如上图中的红蓝两层，分别代表不同层级。红色代表分辨率较高的体素网格、蓝色代表分辨率较低的体素网格)，同时存储网格顶点所代表的特征向量，当从空间中采样得到 x 同时会命中两种分辨率下的两个像素。此时可以通过二线性插值从网格顶点获取 x 点的特征信息。词图可以获取两个特征向量。高分辨率可以使得该模型更能表现出高频信息而与之的代价就是空间存储的特征向量更多。

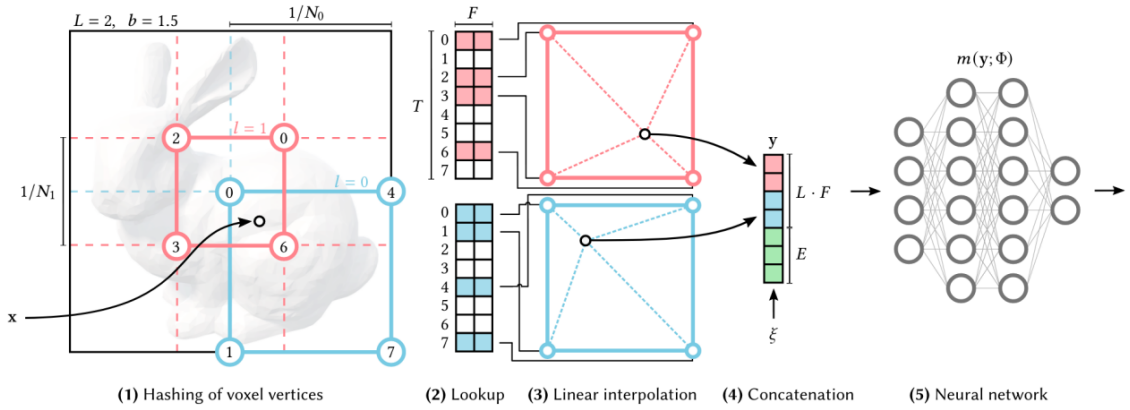

图 6: Instant-NGP 流程图

哈希查找的时间复杂度为 O（1）。Instant-NGP 会为 L 层分辨率创建 N 个哈希表，区别于其他工作[3][4]，Instant-NGP 将网格顶点位置存放哈希索引而非特征向量，而是通过哈希查表得到特征向量再进线性插值。为了提升反向传播性能，Instant-NGP 存储哈希表条目在一个半精度浮点数中，即每个条目两个字节。另外，算法维护了一个全精度的参数主副本，以便稳定的混合精度参数更新。为了优化使用的 GPU 缓存，作者逐层的评估哈希表：当处理一个 batch 的输入位置时，作者安排计算来查看第一层所有输入的第一层哈希编码，跟随对应的第二层所有输入，逐层查找。因此只有少量连续的哈希表不得不驻留在缓存，依赖于 GPU 的并行计算能力。

Parameter	Symbol	Value
Number of levels	L	16
Max. entries per level (hash table size)	T	2^{14} to 2^{24}
Number of feature dimensions per entry	F	2
Coarsest resolution	N_{min}	16
Finest resolution	N_{max}	512 to 524288

图 7: 多分辨率-哈希表参数

3 本文方法

3.1 本文方法概述

本工作对 Nerf 进行了另一种补充改进。考虑到 Nerf 建模对数据集有较高的要求，每张图片描述的场景具备同样的光照条件以及数据集不能有遮挡物，因此在非实验理想条件下例如室外场景，NeRF 的性能得到很大的衰减。

简单来讲，就是 NeRF-W 使得 NeRF 能够使用在不同时间光照条件以及被人车动物遮挡的图片作为数据集进行训练且得到较好的结果。

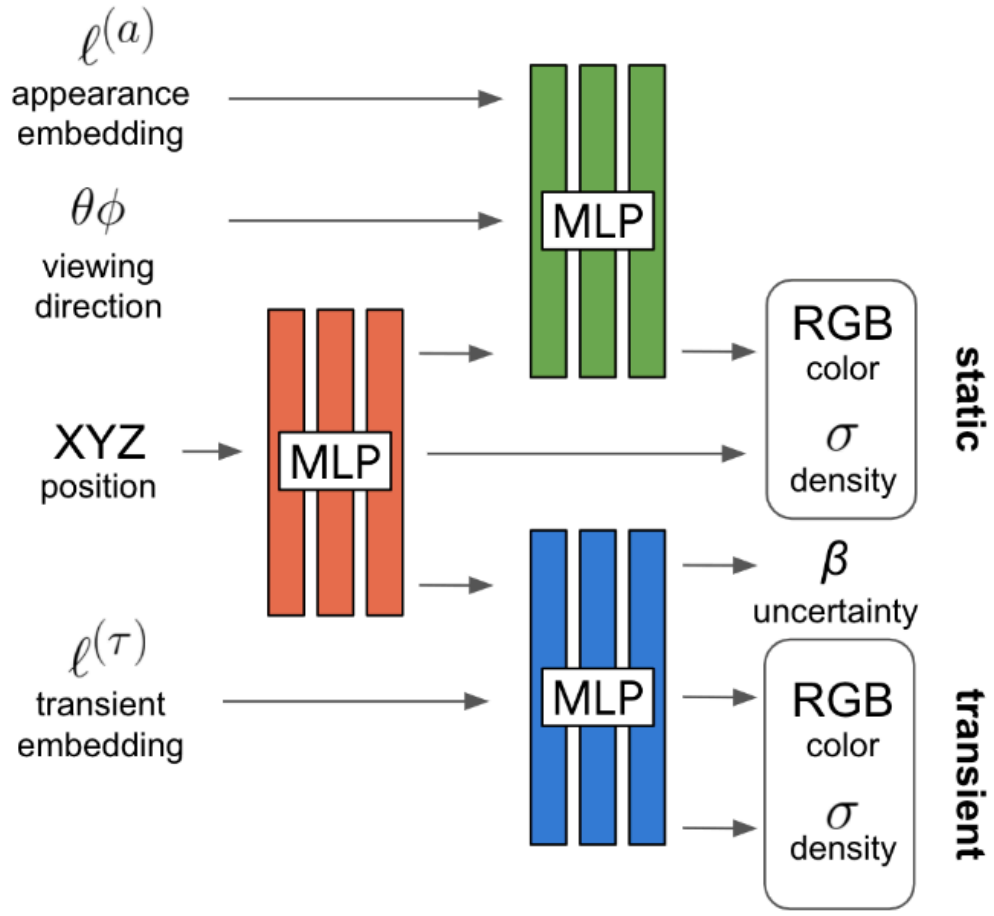

图 8: NeRF in the wild 模型结构图

3.2 特征提取模块

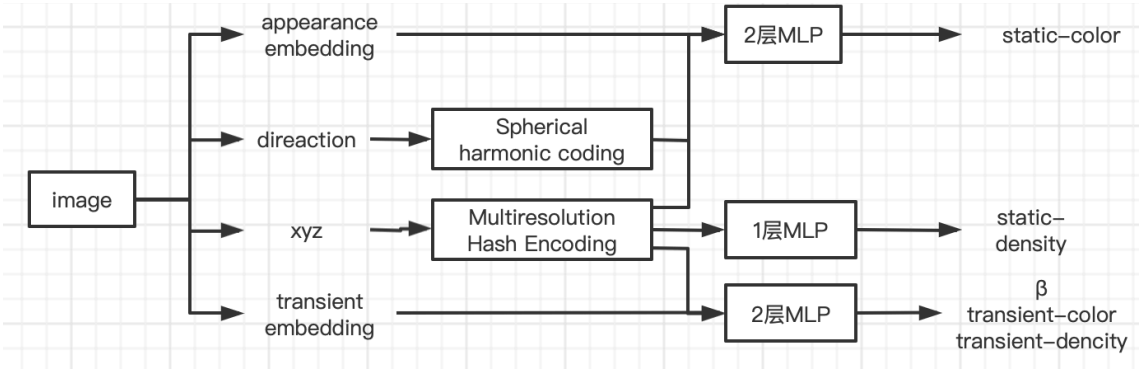

图 9: NeRF in the wild X Instant-NGP

4 复现细节

4.1 与已有开源代码对比

本复习工作参考了：https://github.com/kwea123/nerf—pl.git

https://github.com/yenchenlin/nerf-pytorch.git

本复现在 NeRF 的基础上添加 3 个模块。一是加入对每一张图片进行编码 apearance embedding 向量表示该图片的光照信息，在 NeRF 的模型中加入 apearance embedding 向量使得模型能够区分出各个照片的光照信息。二是对每一张图片进行编码 trainsient embedding 向量，能够通过神经网络对每张图片训练出 β，transient-color，transient-dencity。其中 β 为渲染时采样的权值，利用训练将拥有遮挡物的照片权重降低。三是添加多分辨率哈希编码取代原工作中的位姿编码，该方法能够将 MLP 的层数从 8 层减少为 2 层。大大降低了训练时间。

4.2 实验环境搭建

本人实验环境：系统：ubuntu20.04.5LTS

GPU：A3000

cuda 版本：11.3

安装环境：

1，创建 conda 虚拟环境:conda create -n nerf-w python=3.8

2，进入环境：conda activate nerf-w

3，安装 torch：pip install torch==1.11.0 –extra-index-url https://download.pytorch.org/whl/cu113

4，安装 torch-scatter：pip install torch-scatter -f https://data.pyg.org/whl/torch-1.11.0+11.3.html

5，安装 tinycudann:git clone –recursive https://github.com/nvlabs/tiny-cuda-nn

6，安装 apex：pip install apex

4，安装包：pip install pip install -r requirements.txt

运行：python train.py

4.3 界面分析与使用说明

```
(base) ziwei@ziwei-Precision-5770:~$ conda env list
# conda environments:
#
base                  *  /home/ziwei/anaconda3
nerf-w-ngp               /home/ziwei/anaconda3/envs/nerf-w-ngp
nerf_pl                  /home/ziwei/anaconda3/envs/nerf_pl
ngp_pl                   /home/ziwei/anaconda3/envs/ngp_pl

(base) ziwei@ziwei-Precision-5770:~$ conda activate nerf_pl
(nerf_pl) ziwei@ziwei-Precision-5770:~$ cd /home/ziwei/Documents/GitHub/nerf-w-n
gp_pl
(nerf_pl) ziwei@ziwei-Precision-5770:~/Documents/GitHub/nerf-w-ngp_pl$ python tr
ain.py --dataset_name blender    --root_dir /media/ziwei/软件/Documents/data/ner
f/nerf_synthetic/lego  --N_importance 64 --img_wh 400 400 --noise_std 0   --num_
epochs 20 --batch_size 1024   --optimizer adam --lr 5e-4 --lr_scheduler cosine
 --exp_name exp  --data_perturb occ   --encode_t --beta_min 0.1
```

图 10: 操作界面示意

4.4 创新点

添加多分辨率哈希编码取代原工作中的位姿编码，该方法能够将 MLP 的层数从 8 层减少为 2 层。大大降低了训练时间。原文训练一个 epoch 时间为 40 分钟，引入多分辨率哈希优化网络结构后，训练一个 epoch 时间为 12 分钟。

5 实验结果分析

如图 11左侧的图片（PSNR：18）为使用原始 NeRF 的模型在数据集受污染（随即光照污染，随即遮挡物污染）的情况下建出来的模型，右侧的图片（PSNR：24）为使用 NeRF-W 的模型在数据集受污染的情况下建出来的模型。

图 11: 原文方法效果对比

如图 12未加入多分辨率哈希的模型训练一个 epoch 需要 45 分钟左右。引入多分辨率哈希后（PSNR：23）训练一个 epoch 时间下降为 12 分钟。

```
Epoch 0:   0%|          | 78/15633 [00:13<44:53,  5.78it/s, loss=2.14, v_num=5, train/c_l=0.0329, train/f_l=0.550, train/b_l=1.490, train/s_l=0.00261, train/psnr=11.30]
```

图 12: 未加入多分辨率哈希

```
h 19: 100%|██████████| 15626/15633 [12:18<00:00, 21.16it/s, loss=0.904, v_num=6, train/c_l=0.00526, train/f_l=0.102, train/b_l=0.784, train/s_l=0.00274, train/psnr=22.10, val/psnr=23.70]
```

图 13: 加入多分辨率哈希

6 总结与展望

本文采用 NeRF in the wild 的方法在 NeRF 的模型上进行复现，在此基础上采用 Intant-NGP 的方法对训练速度进行优化。实验证明方法有效，训练时间降为原文的 4 分之一。本文训练时间为 3 个小时（20 个 epoch），但是 Intant-NGP 的方法有着更大的潜力：原文中训练时间为 5～10 分钟（20 个 epoch），这有在 NeRF-W 中引入 appearance embedding 和 transient embedding 导致模型参数加大的原因。在后续工作中通过调整优化模型结构应该能使得速度进一步加强。

参考文献

[1] MILDENHALL B, SRINIVASAN P P, TANCIK M, et al. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis[C]//European Conference on Computer Vision. 2020.

[2] MARTIN-BRUALLA R, RADWAN N, SAJJADI M S M, et al. NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections[J]. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020: 7206-7215.

[3] LIU L, GU J, LIN K Z, et al. Neural Sparse Voxel Fields[J]. ArXiv, 2020, abs/2007.11571.

[4] YU A, FRIDOVICH-KEIL S, TANCIK M, et al. Plenoxels: Radiance Fields without Neural Networks[J]. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021: 5491-5500.

[5] MÜLLER T, EVANS A, SCHIED C, et al. Instant neural graphics primitives with a multiresolution hash encoding[J]. ACM Transactions on Graphics (TOG), 2022, 41: 1-15.

[6] LIN Y C, FLORENCE P R, BARRON J T, et al. iNeRF: Inverting Neural Radiance Fields for Pose Estimation[J]. 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2020: 1323-1330.

[7] LIN C H, MA W C, TORRALBA A, et al. BARF: Bundle-Adjusting Neural Radiance Fields[J]. 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 2021: 5721-5731.

[8] TAKIKAWA T, LITALIEN J, YIN K, et al. Neural Geometric Level of Detail: Real-time Rendering with Implicit 3D Shapes[J]. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021: 11353-11362.