

# 野外神经三维重建

田欢

## 摘要

近年来，神经隐式表达在计算机视觉和图形学中热度很高。它们的适用性最近已经超越了形状生成和基于图像的渲染等任务，扩展到了基于图像的 3D 重建的基本问题。然而，目前基于神经隐式表达的重建方法仅适用于带约束的 3D 环境，例如相机大致均匀分布、光线变化较小等。文中介绍了一种新的方法，该方法能够在不同照明条件下从互联网照片集中高效、准确地重建表面。为了实现这一点，文中提出了一种混合体素和表面引导采样技术，该技术允许在表面周围进行更有效的射线采样，并显著提高重建质量。同时，为了能在 UrbanScene3D 数据集中的复杂场景上取得一个较好的重建结果，我在其基础上加入了重建区域的切割定位，以获得一个较高质量的重建结果。

**关键词：**3D 重建；视图合成；神经渲染

## 1 引言

Neural 3D Reconstruction in the Wild<sup>[1]</sup>提出了一种能够对不同光照条件下的图片进行高质量 3D 表面重建的 neural method。提出的方法基于神经辐射场重建 3D 场景，使用了 NeRF-W 中的 latent appearance modeling 来提升场景建模在无约束网络图片输入下的适应性。此外，文中提出了新数据集 Heritage-Recon 并介绍了数据采集、坐标系对齐、可视性检查等数据集处理方法与步骤。

基于我学习熟悉 UrbanScene3D 数据集的目标，选择这篇论文主要有以下三个理由。第一，此工作中对比了多种传统的 3D 重建方法（如 MVS, Vis-MVSNet, NeRF-W），我可以更好的学习传统重建方法，并学习基于神经网络的重建方法，同时能够尝试用此方法重建 UrbanScene3D 数据集中的大型场景。第二，此工作中用网络上的图片构建了一个数据集，并提供了多种的数据集处理方法和步骤，我可以参照其工作完成 UrbanScene3D 数据集的接口工具。第三，此工作提供了一份规范度较高的源码，在进行复现工作时可以用以分析其工作的细节及难点的实现，利于学习；并且能帮助我提高工程代码的规范度。

## 2 相关工作

基于图像的 3D 重建任务旨在估计给定的一组照片的对象或场景的最可能的 3D 形状（以及可能的外观）。在本节中，将重点介绍与本文最密切相关的工作。

### 2.1 多视图重建

多视图 3D 重建方法使用各种表示来获取图像和估计几何图形，包括点云、深度图、网格或体积隐式函数。许多经典的多视图立体（MVS）方法通过估计每个图像的深度图，然后进行深度融合，以获得密集的点云，从而重建几何结构。然后，可以将表面重建算法（如泊松重建和 Delaunay 三角法）应用于这些点云以生成网格。最近，基于学习的多视图深度估计方法通过利用数据驱动的先验和物理约束，在众多基准上取得了最先进的性能。由于这些方法分别执行深度估计、点云融合和网格提取阶段，它们对深度图中的异常值或不一致性很敏感，并且可能产生噪声或不完整的重建。相反，文中的

方法使用全局表示对整个场景进行建模，并通过神经渲染对外观和几何体进行端到端优化。其他方法也直接估算 3D 表面，但由于其数据依赖性，需要地面真实的 3D 重建用于训练，不能推广到训练数据之外（例如，从室内场景到室外场景）。

## 2.2 NeRF: 将场景表示为用于视图合成的神经辐射场<sup>[2]</sup>

NeRF 通过使用一组稀疏的输入视图来优化基本的连续体积场景函数，从而获得复杂场景的新视图。整个过程可以概述为将每个采样点的 5D 坐标（三维坐标  $(x, y, z)$ 、和观察方向  $(\theta, \phi)$ ）输入 MLP 中得到这个采样点的体密度 和颜色  $c(r, g, b)$ （图 1 中 (a)(b) 过程）。再通过体素渲染技术将输出的颜色和密度投影到新视图图像中（图 1 中 (c)(d) 过程）。

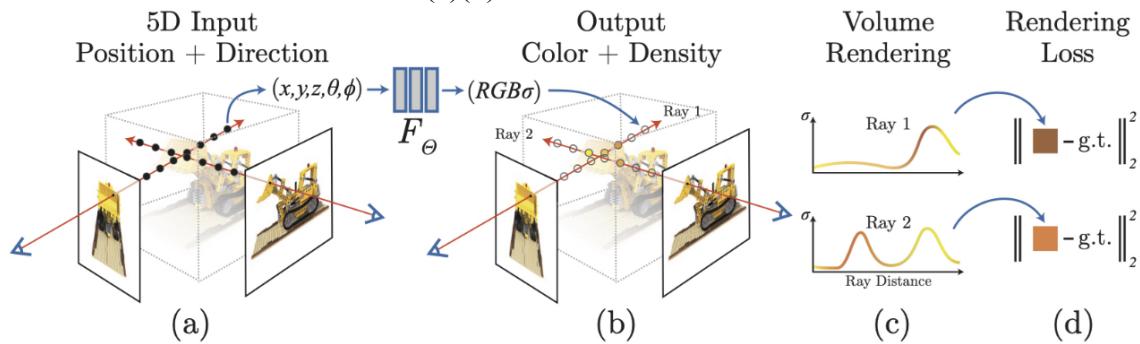


图 1: NeRF

网络如图 2 所示， $X$  经过位置编码输入到 MLP 中，并输出  $\delta$  和 256 维的特征向量，该特征向量与经过位置编码的  $d$  串联（架构图中“+”表示向量串联），再将中间特征和  $d$  一起输入到额外的全连接层中（128 维）预测颜色。所以经过此网络，体素密度只与采样点的空间位置有关；颜色与空间位置和观察方向都有关。

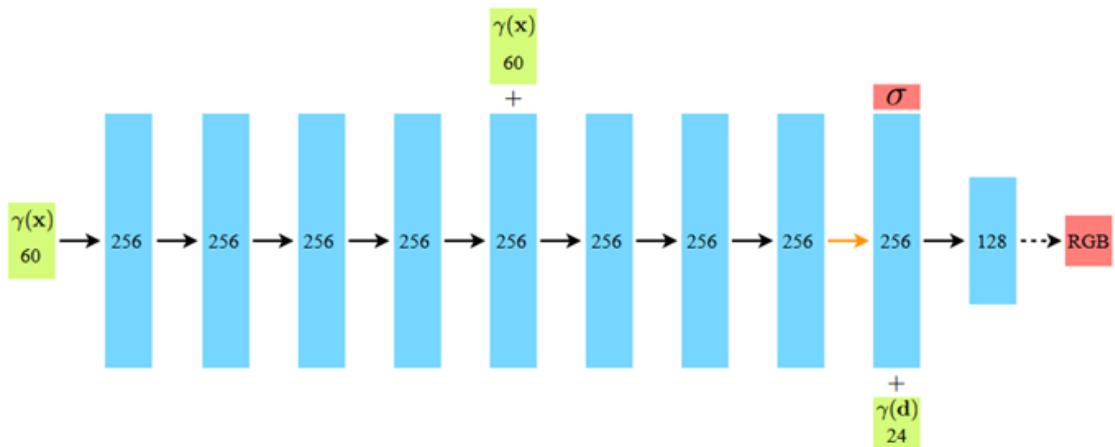


图 2: NeRF 网络架构图

NeRF 中提出了新的位置信息编码方式，直接将位置和视角信息输入得到的结果，在表达高分辨率的场景时较为模糊，而先用位置信息编码的方式将输入映射到高维，可以有效解决这个问题。为了提高渲染效率作者提出按照对最终渲染的贡献比例进行采样，设计了一种“coarse to fine”的多层次体素采样方法，同时优化 coarse 和 fine 两个网络。首先，使用分层采样的方法先采集较为稀疏的  $N_c$  个点，在这些采样点上计算 coarse 网络的渲染结果 ( $C_c(r)$ )， $\omega_i$  对进行归一化， $\omega_i$  可以看作是沿着射线方向的概率密度函数（如图 3 左所示），通过这个概率密度函数可以粗略地得到射线方向上物体的分布情况。基于这个概率密度函数，使用逆变换采样方法，再采样出  $N_f$  个点（如图 3 右所示）。这个方法可以从包含更多可见内容的区域中得到更多的采样点，然后在  $N_c + N_f$  的采样点集合上，计算 fine

网络的渲染结果。

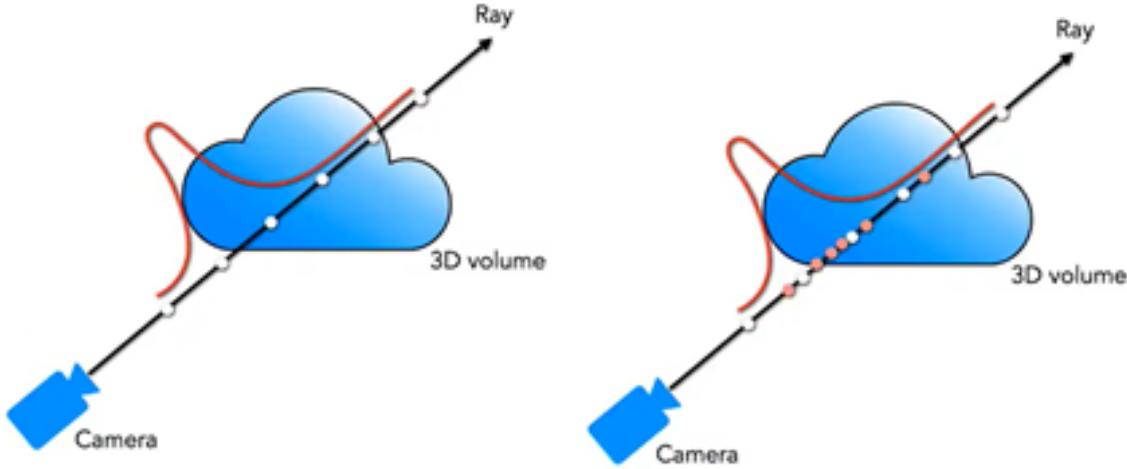


图 3: NeRF 采样方式

### 2.3 NeRF in the Wild: 无约束照片收集的神经辐射场<sup>[3]</sup>

NeRF 的主要局限性是它假设世界在几何上、物质上和光度上是静态的 geometrically, materially, and photometrically static——世界的密度 density 和亮度 radiance 是恒定的。因此，NeRF 要求在同一位置和方向拍摄的任何两张照片必须相同。这一假设在许多真实世界的数据集中被严重违反，例如旅游地标的大型互联网照片集合。NeRF-W 提出了两个模型组件来解决上述问题：

#### (1) 隐外观的建模：

为了使 NeRF 能够适应不同光照和摄影后处理，NeRF-W 采用了 GLO (Generative Latent Optimization) 技术，即每张图像  $i$  都会赋给一个对应的实值外观嵌入 (appearance embedding) 向量  $l_i^{(a)}$ ，其长度为  $n^{(a)}$ 。用图像相关的辐射亮度  $[c]_i(t)$ ，来替换原方程中的图像无关的辐射亮度  $c(t)$ ，同时引入近似的像素颜色  $C_i(r)$  对图像索引  $i$  的依赖：

$$C_i(r) = R(r, c_i, \sigma)$$

$$c_i(t) = MLP_{\theta 2}(z(t), \lambda_d(d), l_i^{(a)})$$

其中，嵌入  $\{l_i^{(a)}\}_{i=1}^N$  和  $\theta$  一同优化。让这种外观嵌入，作为网络其中分支的输入，可以赋予模型改变特定图像中场景辐射场的能力，同时依然保证 3D 几何学是静态的，由所有图像共享。把  $n^{(a)}$  设置为一个较小值，优化时能够找到一个连续空间，使得光照条件可以嵌入，进而可以保证不同条件下的光滑插值。

#### (2) 动态物体处理：

NeRF-W 使用两种设计来解决动态现象。首先，把 NeRF 中的颜色发射 (color-emitting) MLP，作为模型的静态部分 (static head)，然后添加瞬态部分 (transient head)，发出自己的颜色和密度，其中密度允许在训练图像中有变化。这使得 NeRF-W 可以重构包含遮挡物的图像，而没有把 artifact 引入到静态场景表示。其次，并不假设所有观测像素颜色都是同样可靠的，允许瞬态部分发射出不确定性 uncertainty 场 (类似于颜色和密度)，可以让模型调整重构损失，忽略不可靠的像素和 3D 位置，这些位置很可能包含遮挡物。将每个像素的颜色建模为各向同性正态分布，并且进行极大似然，然后使用立体渲染，渲染出该分布的方差。

## 2.4 NeuS：基于多视图重建的体绘制学习神经隐式曲面<sup>[4]</sup>

NeuS 的总目标是实现从 2D 图片到 3D 模型的高保真重建（使用神经渲染方法），使用一个符号距离函数（SDF， signed distance function）的零级集合（zero-level set）来表示一个表面，并部署一种新的体积渲染方法来训练神经 SDF 描述。

NeuS 面向多视点表面重建任务，使用 SDF 描述表面，并用一种新的体积渲染方法训练 SDF 描述。给出 3D 图像的多视角图片，想要构造其表面，这个表面由一个神经隐式 SDF 的零级集合表示。

第一步是构造从 3D 模型到图像的渲染方法。定义两个函数： $f : \mathbb{R}^3 \rightarrow \mathbb{R}$  空间中一点到表面的符号距离（signed distance）； $c : \mathbb{R}^3 \times \mathbb{S}^2 \rightarrow \mathbb{R}^3$  空间中一点到某个视角方向的颜色值，两个函数都通过 MLP 编码。待重构的表面则可以被表示为 SDF 的零级集合：

$$S = \{x \in \mathbb{R}^3 \mid f(x) = 0\}$$

第二步是构造体积渲染训练 SDF 网络，先定义概率密度函数  $\phi_s(f(x))$ ，称为 S 密度。其中  $f(x)$  是 SDF，外层是逻辑斯蒂密度分布：

$$\phi_s(x) = se^{-sx}/(1 + e^{-sx})^2$$

从 3D 表面渲染为 2D 图像，与 gt (ground truth) 图像比对才能获得 loss。对任意一个像素，定义从它发出的一条光线  $\{p(t) = o + tv \mid t \geq 0\}$  其中  $o$  是相机点， $v$  为光线方向，则这一像素的颜色定义为：

$$C(o, v) = \int_0^{+\infty} \omega(t)c(p(t), v)dt$$

其中值得注意的是权重函数  $w(t)$ ，文中为权重函数设置了两个要求：(1) 无偏，即此函数推导的光线与表面碰撞点要满足 SDF 为 0；(2) 可感知遮挡，即如果出现两个点 SDF 值相同的情况，此函数能够满足距相机更近的点具有更大权重。

传统工作中权重函数的形式为： $w(t) = T(t)(t)$ ，其中  $(t)$  称为体积密度， $T(t)$  表示沿着光线的光线积累。NeuS 证明了如果直接在权重函数中直接应用 SDF 会困于局部最优点不满足要求 (1) 产生偏差，于是构造了新的权重函数：

$$w(t) = T(t)\rho(t), \text{ where } T(t) = \exp(-\int_0^t \rho(u)du)$$

沿用传统权重函数的框架，但用不透明（opaque）密度函数  $(t)$  替代体积密度，计算并一般化推广最终得到：

$$\rho(t) = \max\left(\frac{-\frac{d\Phi_s}{dt}(f(p(t)))}{\Phi_s(f(p(t)))}, 0\right)$$

新构造的权重函数首先是无偏的，优化收敛点处 SDF 为 0 满足要求 (1)；其次可以通过配合 SDF 值条件判断多个碰撞点的遮挡关系，满足要求 (2)。最后类似 NeRF，采样光线上的数个离散点计算像素颜色。NeuS 构造的 loss：

$$\mathcal{L} = \mathcal{L}_{color} + \lambda \mathcal{L}_{reg} + \beta \mathcal{L}_{mask}$$

color Loss( $\mathcal{L}_{color}$ ) 为渲染图与 gt 图差异，reg Loss( $\mathcal{L}_{reg}$ ) 是对所有采样点秩序化 (regularize)，形式有些类似规范化，不过是对 SDF 的梯度。最后如果有遮罩监督，则加入遮罩 Loss( $\mathcal{L}_{mask}$ )。

### 3 本文方法

#### 3.1 本文方法概述

Neural 3D Reconstruction in the Wild(NeuralRecon-W) 提出了一种新的 hybrid neural implicit representation（混合神经隐式表示）重建三维曲面的方法。与 MVS、PMVS、MVSNet 相比，此工作使用全局表示对整个场景进行建模，并通过神经渲染端到端的优化几何。与 NeRF-W 相比，此工作主要目的是提取高精度 3D mesh。此外，此工作拓展了 NeuS 的表达，可良好适应无约束网络图片集输入。同时提出了提出体素 + 表面混合的采样策略，以提升训练效率。

#### 3.2 隐式神经场定义

受到 NeRF 启发，NeuralRecon-W 方法基于神经辐射场建模 3D 场景。使用了 NeRF-W 中的 latent appearance modeling 来提升场景建模在无约束网络图片输入下的适应性。为了高精度几何表面建模，拓展了 NeuS 的场景表达方法，使用两个神经隐式函数表示场景，函数使用 MLP 拟合。

$$d = \text{MLP}_{SDF}(x)$$

$$c_i = \text{MLP}_{COLOR}(x, v, e_i)$$

使用函数  $d$  近似到真实表面的 SDF 场，表面  $S$  通过  $d$  函数的 zero level set 提取。

$$S = \{x \mid d(x) = 0\}$$

通过计算输入图片和渲染图片之间的颜色一致性，来优化 MLP 和 appearance embeddings 的参数。给定一个射线  $r(t)$ ，可以根据以下公式渲染此射线在图片  $i$  中的颜色  $C(r)$ 。

$$C_i(r) = \int_0^{+\infty} W(t)c_i(r(t), v, e_i)dt$$

#### 3.3 混合采样

体素引导采样 (Voxel-guided sampling)，SfM 不仅提供相机内外参，还通过输出稀疏点云来提供粗略的初始表面估计。因此，在训练开始前借助 SfM 输出的稀疏点云来生成粗略体素  $V_{sfm}$ 。通过 3D 扩张操作，来确保绝大部分的可视区域被包含在粗略体素中。将射线与  $V_{sfm}$  两个交点形成的线段作为采样区间，并采样  $n_v$  个点。此外，构造的粗略体素可将场景大致划分为前景和后景区域，通过移除与  $V_{sfm}$  无交点的射线（例如天空背景射线），需要训练的射线往往可以减少超过 30%。

表面引导采样 (Surface-guided sampling)，为了训练几何  $\text{MLP}_d$ ，使其精确地拟合 3D 表面，最好在表面周围尽可能多地生成样本。通过将当前模型的 SDF 测值缓存至稀疏体素  $V_{cache}$ ，并从  $V_{cache}$  查询表面位置，进而利用估计的表面位置来生成新样本。在  $V_{sfm}$  上建立深度为 1 的八叉树得到  $V_{cache}$ 。给定查询表面位置  $x$ ，在表面周围窄区间  $(x - t_s, x + t_s)$  内生成  $n_s$  个样本。 $V_{cache}$  在训练中周期性更新，来保证 SDF 值最新。表面引导采样指导网络来解释在表面周围的样本的渲染颜色，从而使网络更准确地拟合几何表面。

混合采样 (Hybrid sampling)，由于对空间监督不足，仅使用表面引导采样将导致体素边界周围出现伪影。相比表面引导采样，体素引导采样的样本要稀疏得多，这是因为体素引导的采样范围更大。在曲面引导采样之后进行另一次重要采样迭代，以确保良好的采样密度。从而使每个射线的采样总数达到  $n_v + 2 \times n_s$ 。

### 3.4 干扰信息处理

处理动态物体，如果直接使用 NeRF-W 的 transient NeRF head，那么 transient NeRF 将会主导颜色，导致所有场景结构将会被建模为视角相关的瞬态效果而不是几何 MLPd，因为函数 d 收敛比 NeRF 更慢。本工作使用分割 masks 来在训练中移除属于动态物体的射线。

监督信号和处理无纹理天空跟随 NeuS，文章使用  $L_1$  loss 来监督渲染颜色图片 ( $L_{COLOR}$ ) 和 eikonal term  $L_{REG}$  来正则化 SDF。由于弱纹理天空缺少运动时差，若直接类似 NeuS 使用一个背景 NeRF 来分离前后景，将会导致天空重建在球体内。剩余在  $V_{sfm}$  中的背景射线（绝大部分是天空）被标记在语义 mask 中，并被  $L_{MASK}$  当作自由空间来惩罚。由于背景语义 mask 往往不是完美的，通常包含一些前景结构，因此仅给  $L_{MASK}$  设定较小的权重。作者经验性发现这一操作可以在移除天空的同时保留前景几何结构。

## 4 复现细节

### 4.1 与已有开源代码对比

参照原有工作，使用 pytorch-lightning 库构建代码，主要实现训练 training step、校验 validation step 两个函数的重写，其中 validation step 中每一轮训练后的重建结果。利用原工作中的数据集进行训练，复现原文中场景。但是最初的复现结果并不理想，与原文中展示的差距很大。

由于训练一个场景非常耗时，以文中 Brandenburg Gate 场景为例，训练花费了十天时间。在其训练过程中，我尝试将 UrbanScene3D 数据集中的场景构造为此工作可用的工作空间。

考虑到本工作在 2D 图像上进行语义分割，然后再在重建过程中移除属于动态物体的射线的方法来去除动态物体。在添加了动态物体种类之后，重建出的场景有优化，但是整体差距依然很大。故在原有工作的基础上添加了重新确定重建区域的代码，实现在通过 COLMAP 建立工作空间后，针对 UrbanScene3D 数据集中的场景重新确定重建区域。

### 4.2 实验环境搭建

一、 创建 conda 环境。

从命令行窗口进入工程目录，执行以下命令（需要先安装 Anaconda）。

---

```
conda env create -f environment.yaml  
conda activate neuconw
```

---

二、 pip 安装 python 包。

---

```
pip install -r requirements.txt
```

---

### 三、 安装 kaolin。

```
git clone -recursive https://github.com/NVIDIAAGameWorks/kaolin  
cd kaolin  
git checkout v0.12.0  
export IGNORE_TORCH_VER=1  
export KAOLIN_INSTALL_EXPERIMENTAL=1  
export TORCH_CUDA_ARCH_LIST="8.6 8.6"  
# If using heterogeneous GPU setup, set the architectures for which to compile the CUDA code.  
export CUB_HOME=/usr/local/cuda-*/include/  
python setup.py develop
```

### 四、 安装 nvdiffrast。

```
git clone https://github.com/NVlabs/nvdiffrast  
cd nvdiffrast  
python setup.py install
```

### 五、 下载数据集。

#### Heritage-Rcon Dataset:

```
mkdir data && cd data  
gdown -id 1q0IMij036L_8FRpFeJf-A0ZPul3EySiV  
gdown -id 1yR0rnuySmTUq6VU8_hwrWXyT9FCqkfXt  
gdown -id 1L7hNv1n5g19txwENWqlFGFCpP4Ys1aDt  
gdown -id 1moVu5jtczxA279qoFMPPrJfErQL4QmI4c
```

#### Data 目录如下：

```
└── NeuralRecon-W  
    └── data  
        └── hritage-rcon  
            ├── brandenburg_gate  
            ├── lincoln_memorial  
            ├── palacio_de_bellas_artes  
            └── pantheon_exterior
```

#### UrbanScene3D Dataset:

下载链接： <https://drive.google.com/drive/folders/1e91lEw56DUBbQgRTo48T3lVjo53SzEOd?usp=sharing>

### 六、 数据准备

对于 Heritage-Rcon Dataset 场景，进入 NeuralRecon-W 目录命令行执行：

---

```
for SCENE_NAME in brandenburg_gate \\
lincoln_memorial palacio_de_bellas_artes pantheon_exterior;
do
    scripts/data_generation.sh
    data/heritage-recon/${SCENE_NAME}
done
```

---

对于 UrbanScene3D Dataset 场景，首先通过 colmap 构建工作空间：

---

```
colmap feature_extractor --database_path \\
./database.db --image_path ./images
colmap exhaustive_matcher --database_path ./database.db
mkdir sparse && colmap mapper --database_path ./database.db \\
--image_path ./images --output_path ./sparse
mkdir dense && colmap image_undistorter --image_path \\
./images --input_path ./sparse/0 --output_path
./dense --output_type COLMAP
```

---

输出：dense 文件夹，目录结构如下：

```
└── dense
    ├── images
    |   ├── 0.JPG
    |   ├── ...
    |   └── 48.JPG
    ├── run-colmap-geometric.sh
    ├── run-colmap-photometric.sh
    ├── sparse
    |   ├── cameras.bin
    |   ├── images.bin
    |   └── points3D.bin
    └── stereo
        ├── consistency_graphs
        ├── depth_maps
        ├── fusion.cfg
        ├── normal_maps
        └── patch-match.cfg
```

将 dense 文件夹，转移到 data 目录下，重名为场景名（以 PolyTech 为例），进入

NeuralRecon-W 目录命令行执行:

---

```
python tools/prepare_data/prepare_semantic_maps.py -root_dir $WORKSPACE_PATH -gpu 0
```

---

将 UrbanScene3D\_config/PolyTech 中的 config 文件复制到 data/PolyTech 目录下（其他场景的需要确定重建区域，并通过 ReconArea/main.py 计算并修正 config 文件中的参数），进入 NeuralRecon-W 目录命令行执行:

---

```
scripts/data_generation.sh ./data/PolyTech
```

---

注：使用 banScene3D Dataset 构建工作空间时，如果内存空间不足，可以通过 PictureSampling.py 脚本进行压缩（要修改脚本中的图片路径）。

#### 4.3 界面分析与使用说明

由于本工作的重建时间较长，故重建过程主要由命令行命令完成。可视化界面主要进行对重建场景的展示。

##### 一、重建

对于 Heritage-Rcon Dataset 场景，进入 NeuralRecon-W 目录命令行执行:

---

```
# “SCENE_NAME” 改为要重建的场景名  
scripts/train.sh $EXP_NAME config/train_${SCENE_NAME}.yaml \\  
$NUM_GPU $NUM_NODE
```

---

对于 UrbanScene3D Dataset 场景（以 PolyTech 为例），进入 NeuralRecon-W 目录命令行执行:

---

```
# “SCENE_NAME” 改为要重建的场景名  
scripts/train.sh PolyTech config/train_PolyTech.yaml 4 1
```

---

##### 二、可视化

提交代码中包含四个重建场景的模型，可通过以下命令进行可视化:

---

场景展示 GUI	python main.py
Brandenburg 场景	python view/brandenburg_view.py
Lincoln Memorial 场景	python view/lincoln_memorial_view.py
Castle 场景	python view/castle_view.py
PolyTech 场景	python view/L7_view.py

---

场景可视化界面如图 4 所示，通过左边工具按钮选择场景，通过鼠标或方向键改变观察视角，鼠标滚轴进行放大或缩小。



图 4: 场景可视化界面

#### 4.4 创新点

本工作创新点主要表现在以下方面:

提出一种体积引导 + 表面引导的混合采样技术，采样效率、模型重建质量得到大幅提升；

提出一个新的数据基准，以及针对于野外场景的评价指标；

针对 UrbanScene3D 数据集中的场景，提出了一种新的确定重建范围的方法。

### 5 实验结果分析

本工作的实验部分主要是为获得 Heritage-Rcon 数据集和 UrbanScene3D 数据集中的场景的更好的重建结果进行，以 Heritage-Rcon 数据中的 Brandenburg 场景为例，初次重建结果如图 5:

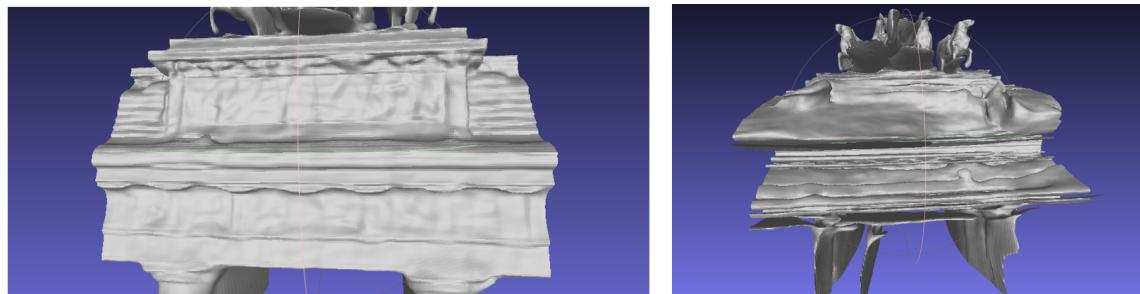


图 5: 第一次重建: 正确结果 (左), 重建结果 (右)

观察中间结果，发现在训练了 6 个 Epoch 后重建出现偏差，考虑两个原因：1、训练过程过拟合。2、原工作使用分割 masks 来在训练中移除属于动态物体的射线，所以突出部分可能是没有移除的动态物体。在 20 个 Epoch 的训练完成后，部分错误区域在逐渐消失，但是依旧存在大块的错误结果（如图 6 所示）。

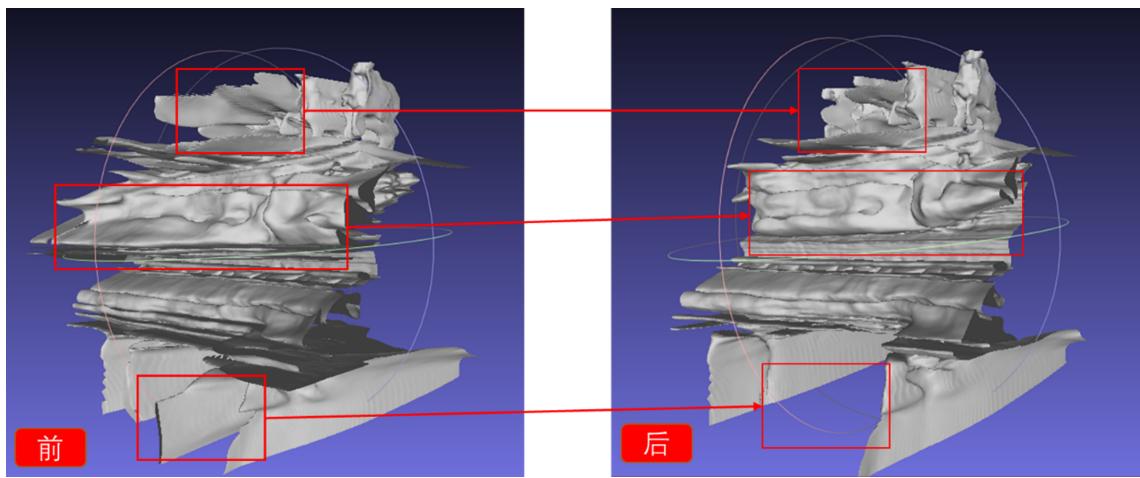


图 6: 训练过程中 (左), 训练完成 (右)

第一次的重建结果并不理想, 参考上述的两个原因进行优化。首先尝试添加动态物体类别, 但其结果如图 7 所示, 并没有明显改善。

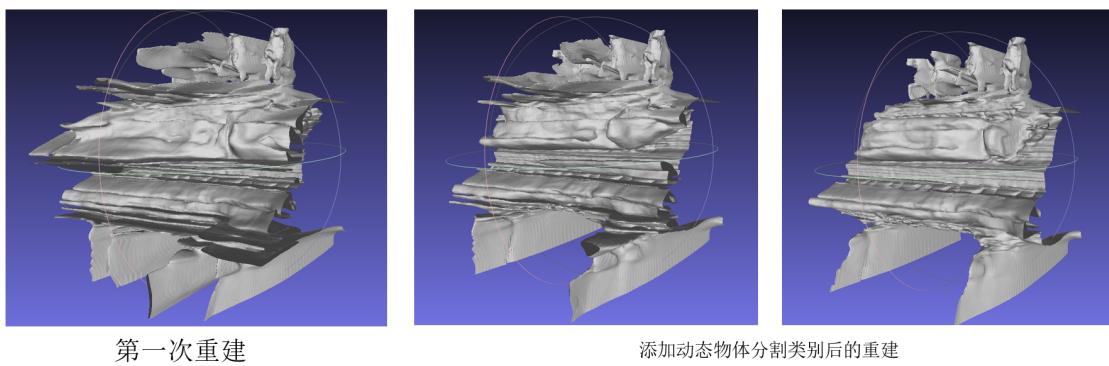


图 7: 添加动态物体类别后重建结果

其次, 在现有结果上调整不同 loss 的权重, 其重建结果有了明显改善 (如图 8 所示)。

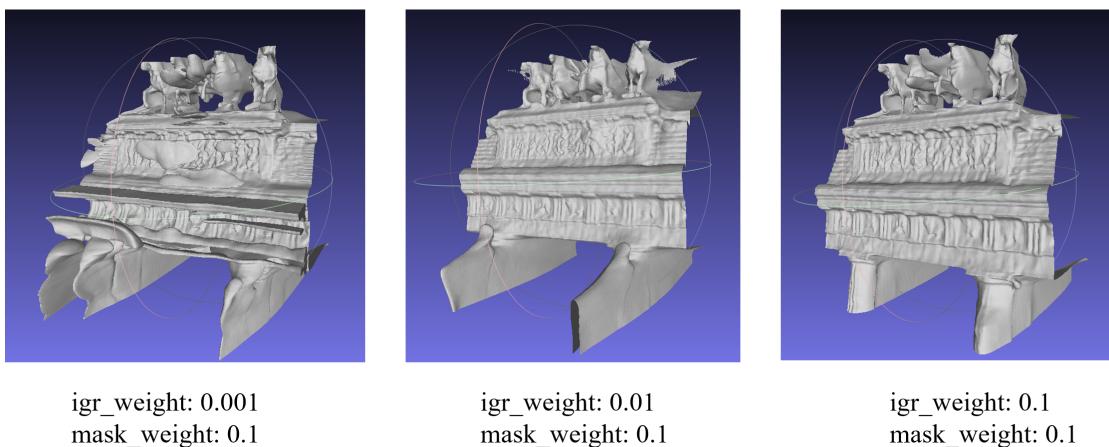


图 8: 不同权重 loss 的重建结果

在 UrbanScene3D 数据集中的 PolyTech 场景上进行重建, 首先通过 COLMAP 计算相机位姿, 构建工作空间。重建结果如图 9 右所示, 其结果无法分辨场景内容。

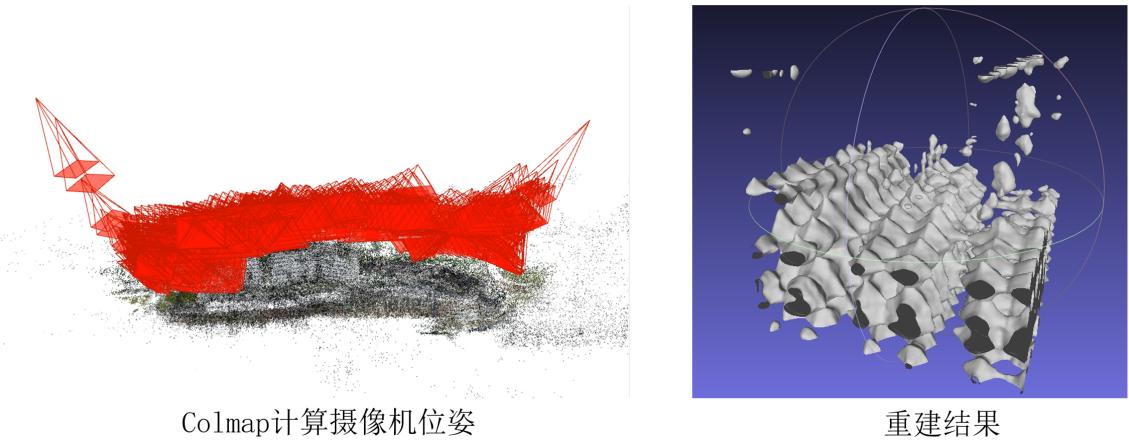


图 9: PolyTech 场景重建

在可视化重建区域后发现 COLMAP 进行稀疏重建时，由于较远的背景点导致计算出的重建区域过大，如图 10 左图所示，绿色框为重建区域，红色点为需要重建的场景。故对重建区域在原有稀疏重建的结果上进行重新划分，如图 10 中、右所示。

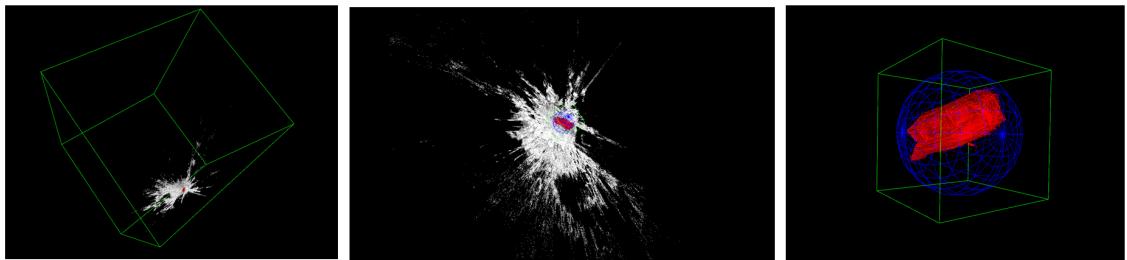


图 10: 场景重建区域划分

在重新划分重建区域之后的重建结果如图 11 所示。

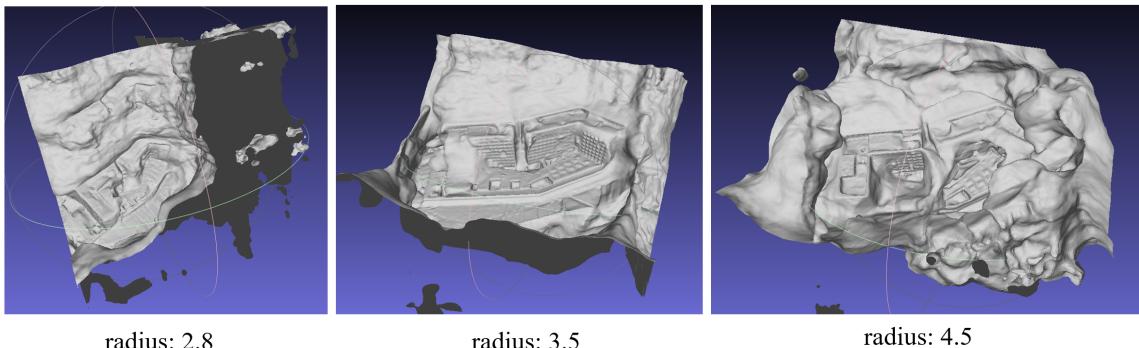


图 11: 不同划分半径下的重建结果

在重新划分重建区域后，其重建结果已经可以看到场景的大致轮廓，相较于初次重建结果已经有了明显改善。在此结果上在缩小体素大小，以获得更多的细节，最终重建结果如图 12 所示。

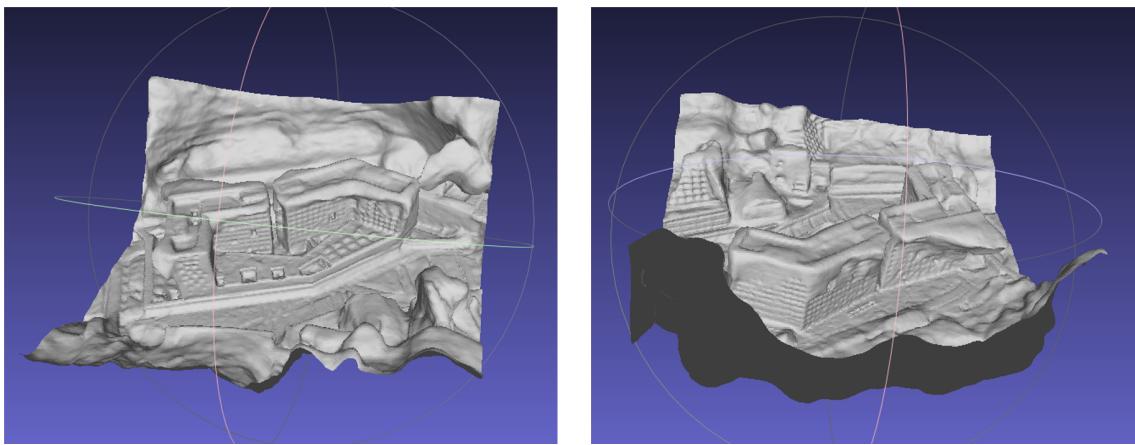


图 12: 缩小体素大小的重建结果: 正面 (左), 背面 (右)

实验最终重建结果如图 13 所示, 相较于最初的重建结果已经有了明显的改善, 但并未达到最理想的结果。



图 13: Heritage-Rcon Brandenburg 场景 (左上), UrbanScene3D Castle 场景 (右上), Heritage-Rcon Lincoln Memorial 场景 (左下), UrbanScene3D PolyTech 场景 (右下)

## 6 总结与展望

Neural 3D Reconstruction in the Wild 提出了一种能够对不同光照条件下的图片进行高质量 3D 表面重建的 neural method。提出的方法基于神经辐射场重建 3D 场景, 使用了 NeRF-W 中的 latent appearance modeling 来提升场景建模在无约束网络图片输入下的适应性。此外, 文中提出了新数据集 Heritage-Recon 并介绍了数据采集、坐标系对齐、可视性检查等数据集处理方法与步骤。在此基础上, 本文尝试使用此方法对 UrbanScene3D 数据集中的场景进行重建, 并取得了较好的重建结果。

在本次论文复现过程中我收获颇多, 第一, 在此工作中我学习了解了多种传统的 3D 重建方法 (如 MVS, Vis-MVSNet, NeRF-W), 学习了解了基于神经网络的重建方法, 以及例如 NeRF、NeuS 等通过图片进行渲染和重建的方法。第二, 此工作中用网络上的图片构建了一个数据集, 并提供了多种的数据

集处理方法和步骤，为我学习使用和组织数据集提供了新的思路。第三，此工作提供了一份规范度较高的源码，学习使用 pytorch-lightning 库构建代码，实现分布式训练和重建。

同时，对于 UrbanScene3D 数据集中场景的重建还有可以改进的方法。Neural 3D Reconstruction in the Wild 提出的方法继承了 NeRF 方法的局限性。例如，对相机内外参精度要求高，由于此方法仅从已知图像中学习表面位置，因此无法生成未观测区域的精确几何。所以，对于数据集的采集和组织方式有了新的需求。此外，与 Heritage-Recon 数据集中场景的背景多为天空不同的是 UrbanScene3D 数据集中场景的背景多为地面。因此，可以通过提高重建过程中去除背景和分割建筑物和其他景物的精度和准确度来提高重建质量。

## 参考文献

- [1] SUN J, CHEN X, WANG Q, et al. Neural 3D reconstruction in the wild[C]//ACM SIGGRAPH 2022 Conference Proceedings. 2022: 1-9.
- [2] MILDENHALL B, SRINIVASAN P P, TANCIK M, et al. Nerf: Representing scenes as neural radiance fields for view synthesis[J]. Communications of the ACM, 2021, 65(1): 99-106.
- [3] MARTIN-BRUALLA R, RADWAN N, SAJJADI M S, et al. Nerf in the wild: Neural radiance fields for unconstrained photo collections[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 7210-7219.
- [4] WANG P, LIU L, LIU Y, et al. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction[J]. arXiv preprint arXiv:2106.10689, 2021.