

# NeRF 渲染原理

## 1 NeRF 中 $\sigma$ 和 $c$ 的估计步骤

在 NeRF (Neural Radiance Fields) 模型中, 通过多层感知机 (MLP) 来实现对场景的表示和渲染。输入是一个 3D 空间位置  $x$  和观察方向  $d$ , 输出是该位置的颜色  $c$  和体积密度  $\sigma$ 。下面总结了这一过程的主要步骤。

### 1.1 总体流程

1. **输入 5D 坐标:** 输入包含位置  $x = (x, y, z)$  和观察方向  $(\theta, \phi)$ , 后者在实现过程中用一个 3D 笛卡尔单位向量表示。
2. **通过 MLP 网络估计:** MLP 网络  $F_{\Theta}$  以  $(x, d)$  为输入, 输出颜色  $c$  和体积密度  $\sigma$ 。在网络结构设计上:
  - 首先, 输入的 3D 坐标  $x$  被送入一个有 8 个全连接层 (ReLU 激活函数和 256 个 channels) 的 MLP, 输出体积密度  $\sigma$  和一个 256 维的特征向量。
  - 然后, 将这个特征向量与观察方向  $d$  连接起来, 经过一个额外的全连接层 (ReLU 激活函数和 128 个 channels) 后产生与视角方向相关的颜色  $c$ 。

### 1.2 体积渲染原理

1. **体积渲染的积分计算:** 渲染时, 通过路径  $r(t) = o + td$  的近似数值积分来估计积分式  $C(r)$ , 即射线经过每个采样点颜色的积累:

$$C(r) = \int_{t_n}^{t_f} T(t) \sigma(r(t)) c(r(t), d) dt,$$

其中  $T(t) = \exp\left(-\int_{t_n}^t \sigma(r(s)) ds\right)$  表示从起点  $t_n$  到  $t$  的累积透过率。

2. **采用分层采样估计积分:** 为了进行数值估计, 我们将积分区间  $[t_n, t_f]$  等分成  $N$  个间隔, 并在每个间隔里均匀随机地采样一个点  $t_i$ , 完成积分估计。用此方法, 可以覆盖连续场景的采样点:

$$t_i \sim U\left(t_n + \frac{i-1}{N}(t_f - t_n), t_n + \frac{i}{N}(t_f - t_n)\right)$$

3. **积分的数值估计：**最终估计公式如下：

$$\hat{C}(r) = \sum_{i=1}^N T_i (1 - \exp(-\sigma_i \delta_i)) c_i,$$

其中  $\delta_i = t_{i+1} - t_i$  是相邻采样点的距离， $T_i = \exp\left(-\sum_{j=1}^{i-1} \sigma_j \delta_j\right)$  表示从起点到第  $i$  个采样点的累积透过率。

### 1.3 训练过程

1. **损失函数：**通过比较合成图像与实际观察到的图像间的残差，计算损失函数并反向传播来优化网络参数  $\Theta$ 。
2. **优化：**通过随机梯度下降 (SGD) 或 Adam 优化等方法，迭代更新 MLP 的权重参数，使得合成图像逐渐逼近真实图像。

### 1.4 总结

- **输入：** 5D 坐标  $(x, d)$
- **输出：** 颜色  $c$  和体积密度  $\sigma$
- **网络结构：** 3D 坐标通过 8 层全连接层产生体积密度  $\sigma$  和特征向量；特征向量结合观察方向通过 1 层全连接层产生颜色  $c$
- **渲染：** 根据体积渲染原理积分，输出颜色  $C(r)$

通过这些步骤，可以在实际场景中生成高质量的视角相关图像，实现逼真渲染。