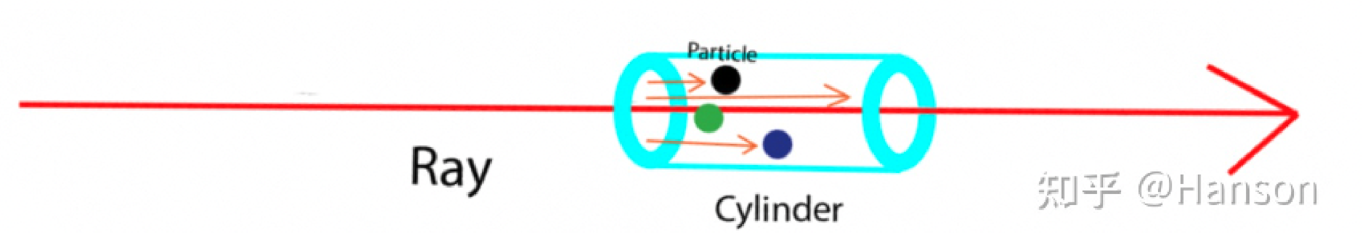
nerf论文公式推导

1. 体渲染公式推导

考虑一条光线在s处的光强为I(s)，光线上极小的一段路径（圆柱体表示）。设圆柱体底面积为E，光线前进的长度为 $\Delta S->0$ ，内部粒子密度为 $\rho(s)$ ，设光线前进途中碰见的都是半径为r的粒子。



光线碰见粒子会被吸收或者反射，通过计算光碰撞粒子的概率就可以知道有多少光通过这个圆柱体路径。因为 $\Delta S->0$ ，粒子可以被近似当作平铺在圆柱体底面，那么这些粒子所占用的面积为：$E\Delta s\rho(s)\pi r^2$ 。那么粒子所占底部面积的比例为：

$$\frac{E\Delta s\rho(s)\pi r^2}{E}=\Delta s\rho(s)\pi r^2$$

光通过这段路径之后，光强变为：

$$I(s+\Delta s)=(1-\Delta s\rho(s)\pi r^2)I(s)$$

$$\Delta I=I(s+\Delta s)-I(s)=-\Delta s\rho\pi r^2 I(s)$$

$$\frac{dI}{ds}=-\rho\pi r^2 I(s)$$

设 $\sigma(s)=\rho\pi r^2$

$$\frac{dI}{ds} = -\sigma(s)I(s)$$

$$\frac{dI}{I} = -\sigma ds$$

$$\int_0^s \frac{dI}{I} = -\int_0^s \sigma(t)dt$$

$$In(I_s) - In(I_0) = -\int_0^s \sigma(t)dt$$

$$\frac{I_s}{I_0} = e^{-\int_0^s \sigma(t)dt}$$

$$I_s = I_0 e^{-\int_0^s \sigma(t)dt}$$

设 $T(s) = e^{-\int_0^s \sigma(t)dt}$ 表示透明度，即光线到达s时光强保留的幅度，**像素最终的颜色由反射的光决定**；设 $F(s) = 1 - T(s)$ 表示不透明度，即当光线到达s处时，光强反射的幅度，F(s)可以视为累积分布函数，s处粒子的颜色都为 $c(s)$ ，则最后的颜色输出为：

$$\begin{aligned}
C(s) &= \int_0^D (F(s)^{'})c(t)dt \\
&= \int_0^D ((1 - T(t))^{'})c(t)dt \\
&= \int_0^D ((1 - e^{-\int_0^D \sigma(t)dt})^{'})c(t)dt \\
&= \int_0^D (-e^{-\int_0^D \sigma(t)dt})^{'}(-\sigma(t))c(t)dt \\
&= \int_0^D T(t)\sigma(t)c(t)dt
\end{aligned}$$

在计算机实际计算时采用离散值，因此假设在射线段[a, b]内密度 $\sigma(a)$与$c(a)$ 恒定，则有：

$$\begin{aligned}
C(a->b) &= \int_a^b T(t)\sigma(a)c(a)dt \\
&= \int_a^b e^{-\int_a^t \sigma(t)dt}\sigma(a)c(a)dt \\
&= \sigma(a)c(a)\int_a^b e^{-\int_a^t \sigma(a)dt}dt \\
&= \sigma(a)c(a)\int_a^b e^{-\sigma(a)\int_a^t dt}dt
\end{aligned}$$

$$
\begin{aligned}
&= \sigma(a)c(a)\int_a^b e^{-\sigma(a)(t-a)}dt \\
&= \sigma(a)c(a)\frac{e^{-\sigma(a)(t-a)}}{-\sigma(a)}|_a^b \\
&= c(a)(1-e^{-\sigma(a)(b-a)})
\end{aligned}
$$

透射率T(n)可以分解：

$$
\begin{aligned}
T(a\to c) &= e^{-\int_a^b \sigma(t)dt + -\int_b^c \sigma(t)dt} \\
&= e^{-\int_a^b \sigma(t)dt}e^{-\int_b^c \sigma(t)dt} \\
&= T(a\to b)T(b\to c)
\end{aligned}
$$

T(n)可以分段离散化，给定一组区间 $[t_n, t_{n+1}]_{n=1}^N$ ，在第n段内有恒定的密度 $\sigma(n)$ ，在 $t_1=0$ 和 $\delta(n)=t_{n+1}-t_n$ ，则透射率离散化为：

$$
\begin{aligned}
T(n) = T(t_n) = T(0\to t_n) &= e^{-\int_0^{t_n} \sigma(t)dt} \\
&= e^{\sum_{k=1}^{n-1} -\sigma_k\delta_k}
\end{aligned}
$$

离散化分段的体渲染公式：

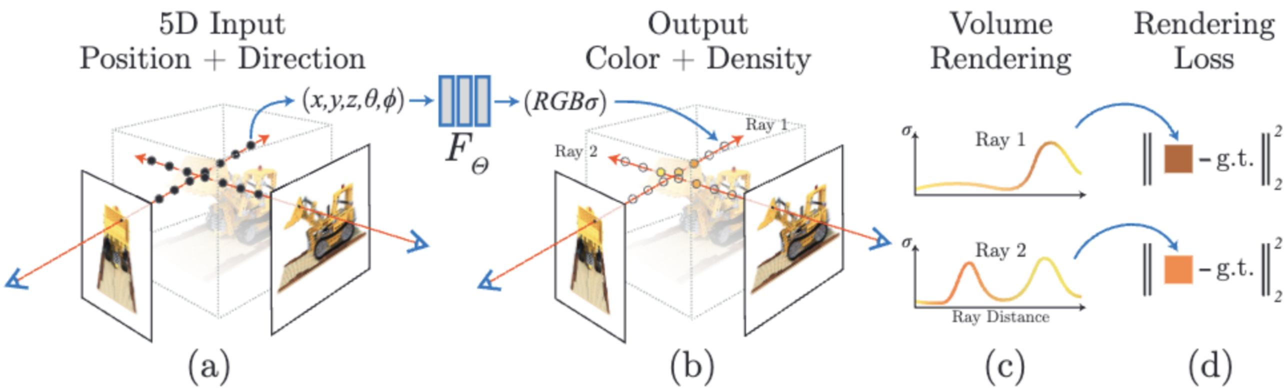
$$
\begin{aligned}
C(t_{N+1}) &= \sum_{k=1}^{n-1}\int_{t_n}^{t_{n+1}} T(t)\sigma_n c_n dt \\
&= \sum_{k=1}^{n-1}\int_{t_n}^{t_{n+1}} T(0\to t_n)T(t_n\to t)\sigma_n c_n dt \\
&= \sum_{k=1}^{n-1} T(0\to t_n)\int_{t_n}^{t_{n+1}} T(t_n\to t)\sigma_n c_n dt \\
&= \sum_{k=1}^{n-1} T_n c(n)(1-e^{-\sigma(n)(t_{n+1}-t_n)}) \\
&= \sum_{k=1}^{n-1} T_n c(n)(1-e^{-\sigma(n)\delta(n)})
\end{aligned}
$$

其中： $T(n) = e^{\sum_{k=1}^{n-1} -\sigma_k\delta_k}$

这就是真正计划时用到的公式

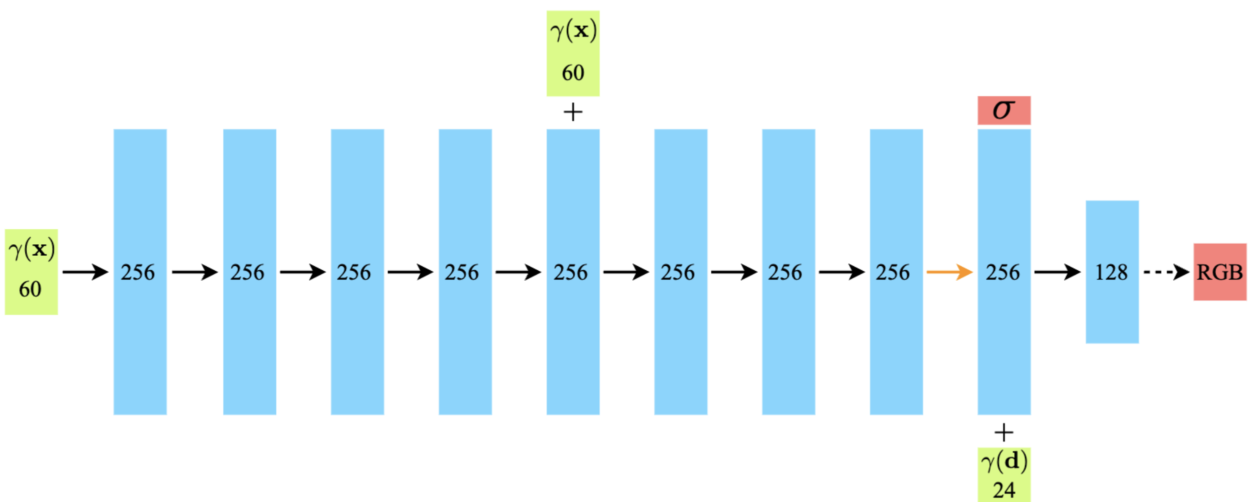
2 Nerf 介绍

2.1 方法框架



Nerf的思想：用MLP将场景表示为一个5D函数。输入查询点的坐标（三维）和方向（两维），MLP预测出该点的体密度和颜色值。对于一条相机射出的查询光线，在光线上采用多个点，查询得到这些点的体密度和颜色值，然后利用体渲染得到该点像素预测值。体密度只和位置坐标有关，这是为了保持视图一致性。颜色rgb输出值还与方向相关，这是为了使多视图颜色不一致，比如可以保持一些镜面反射。

2.2 模型架构图



位置信息先通过八层全连接层，在第五层时候位置信息又与特征向量cat在一起。然后输出体密度值和一个256维度的特征向量，这个256维度的向量与方向向量cat在一起经过两个层最后得到颜色预测值。

2.3 位置编码

MLP相当于一个插值器，MLP倾向于输出平滑的信息，这会导致模型不能学到图像中的高频信息。为了解决这个问题nerf中使用了位置编码，将输入映射到高维。

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

在实现中，对于位置向量，L取10。对于方向向量，L=4。因此三维的位置向量会被映射到60维，在实现中会吧两维的方向向量归一化到三位单位向量，最后映射得到24维的方向向量。

2.4 分层采样

为了减少对高频场景充分采用所要采样的点，nerf采用分层抽样的策略。具体来说：就是训练一个粗网络和一个精细网络。先在查询光线上均匀采样64个点，用粗网络获得这64个点的体密度。然后根据这些体密度的累积分布函数再采样128个点，用精细网络查询64+128=192个点的体密度值和颜色值，通过体渲染得到最后预测的像素值。

2.5 损失函数

用平方误差损失函数同时优化粗网络和细网络。

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