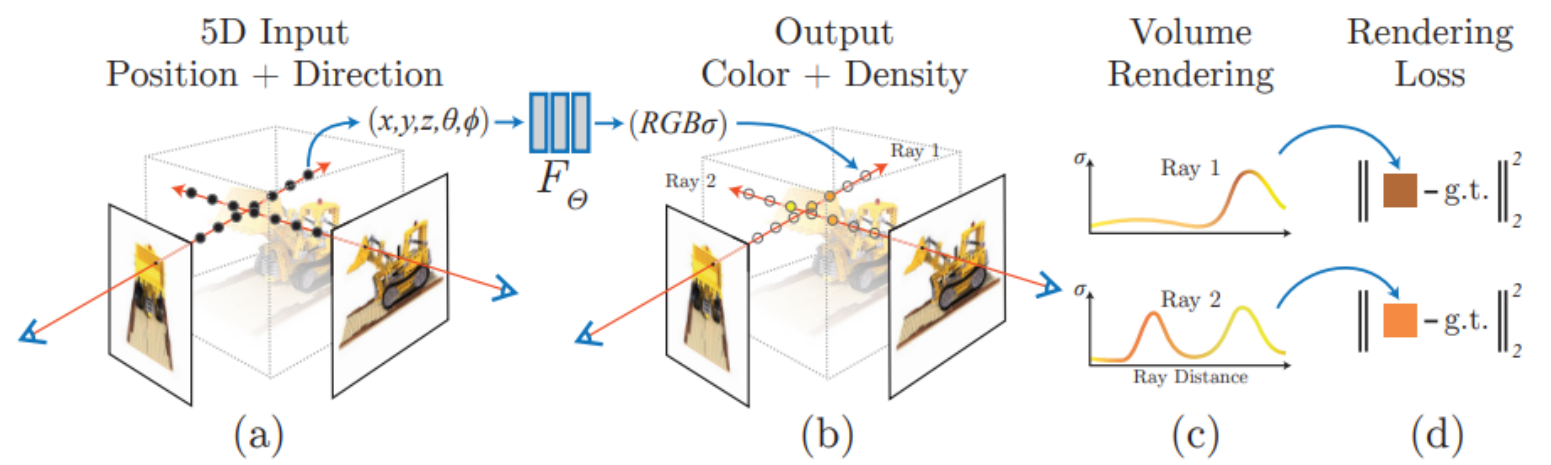


NERF++: ANALYZING AND IMPROVING NEURAL RADIANCE FIELDS

论文贡献

- 1.首先对辐射场中的形状与辐射量之间的歧义性进行了分析，并且分析得到说那为什么NeRF可以避免这种歧义性。
- 2.将原始的NeRF扩展到一个大尺度无边界的场景中。在这一点中，主要是通过将场景划分为前景与背景来实现的。

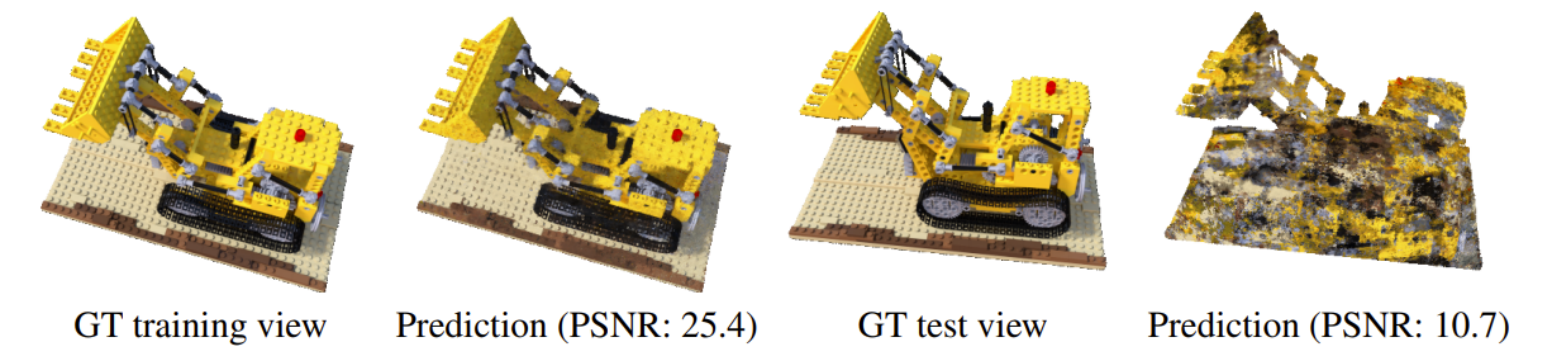
形状与辐射量之间的歧义性



$$C^{predict} = \int_0^{+\infty} T(s)\sigma(s)C(s)ds$$

一个像素点的颜色是由 $\sigma(s)$ 与 $C(s)$ 共同决定的，存在这样的情况，当 $\sigma(s)$ 预测错误时， $C(s)$ 也预测错误，但是两个错误的东西可以很好的拟合原来的颜色分布。

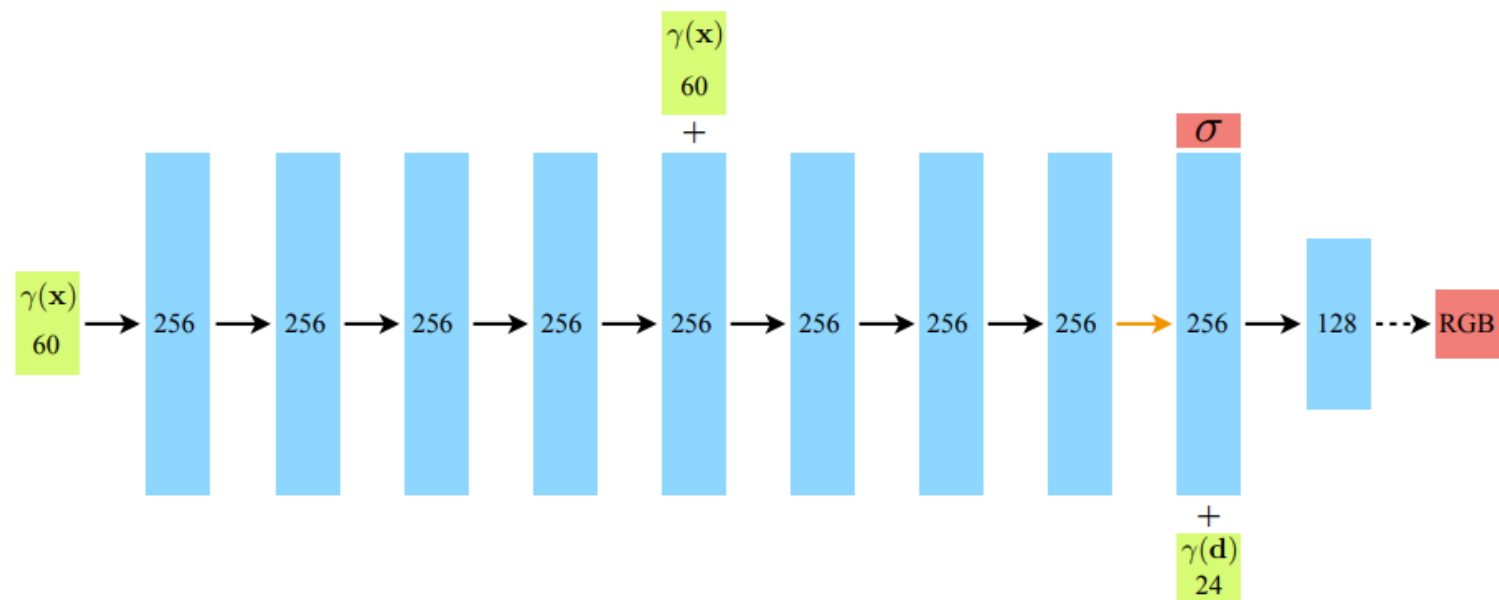
论文里使用一个实验来对这一现象进行说明，将密度固定为一个球壳，对颜色进行训练，发现其也可以很好的训练数据进行拟合，但是无法生成新视角的图片。



既然有这种情况存在，那为什么原始Nerf那篇论文实验结果还蛮好的？测试视角生成的图片都还不错？作者分析主要有两个原因：

1. 不正确的几何形状迫使辐射场具有更高的内在复杂性;
2. NeRF特殊的MLP结构可以隐式地编码一个光滑的BRDF的先验;

先来看一下原始Nerf的结构



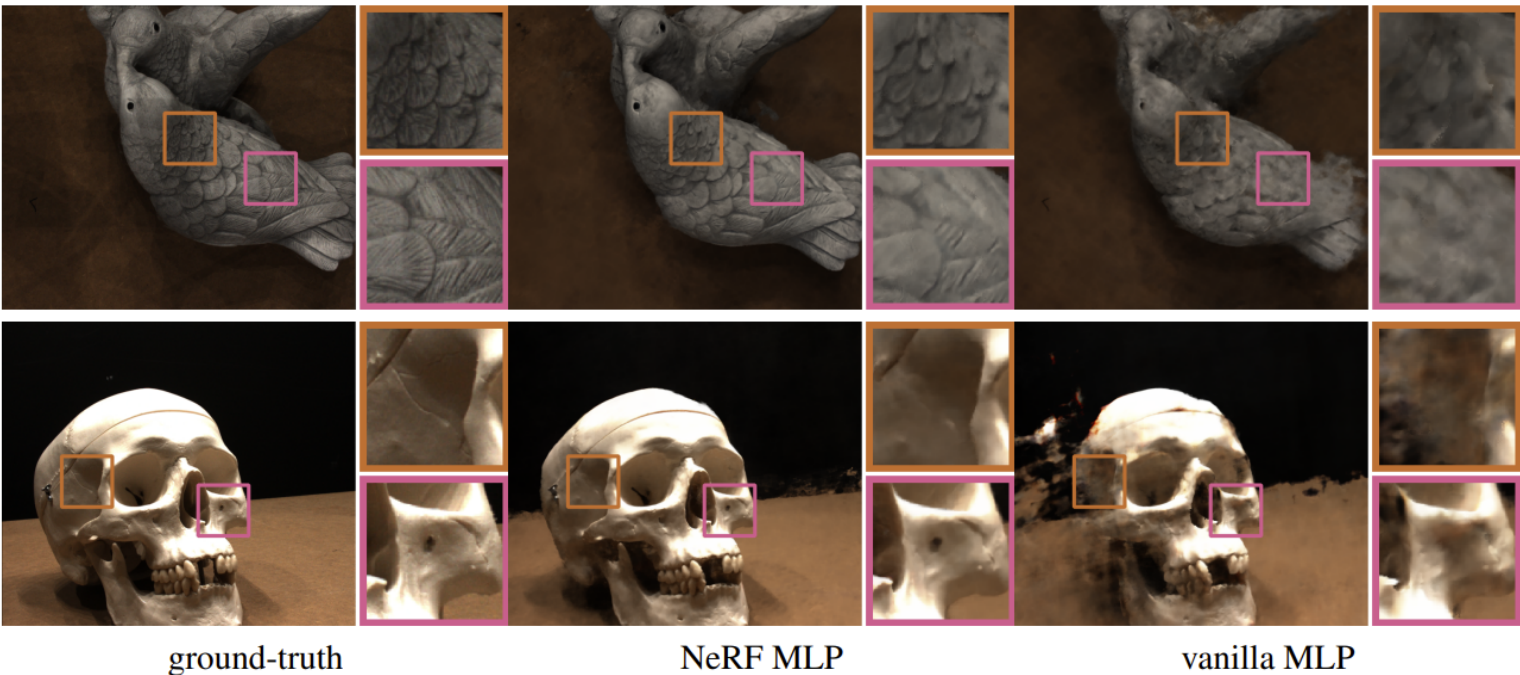
再回忆一下位置编码：

先来看第一点：

但是密度是正确的话，其实对观测角度的依赖性不强。如果密度的分布是错误的话，那你要让图片跟训练数据长得一样，那就是说从不同角度看起来，你颜色区别就会很大，但观测角度变化一点的时候，你颜色的信息变化会很剧烈，但是目前的Nerf的MLP层的能力无法做到这一点。

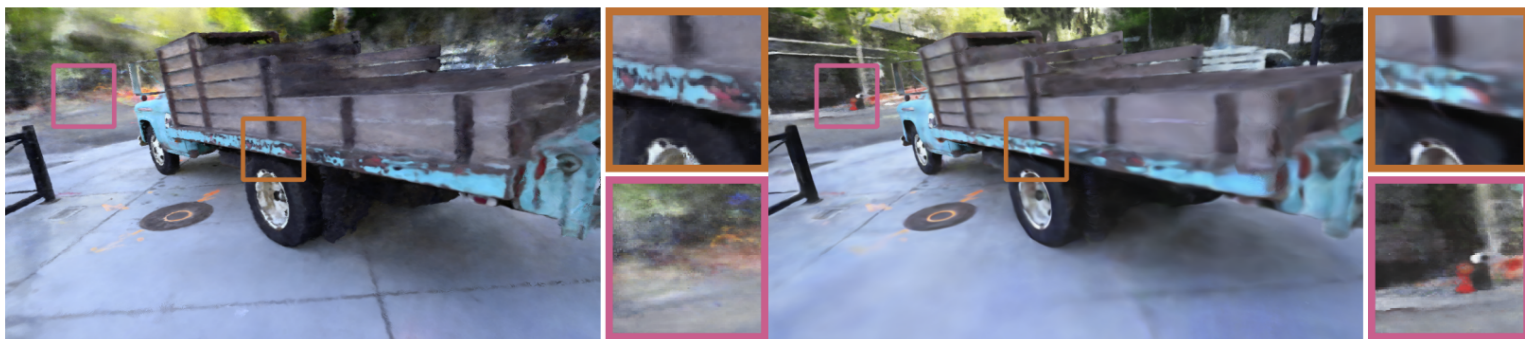
再来看第二点：

我们的观测角度是在最后几层的时候才放进去，用于预测颜色的，也就是说颜色主要取决于xyz的坐标，另外在编码的时候，xyz坐标用了10个频率进行编码，但是只用了4个频率对观测角度进行编码，使得无法提取观测角度的高频信息，也就是很难出现观测角度变化一点，但是颜色的信息变化剧烈的情况，为了验证这个想法，作者做了一个实验，将xyz坐标和观测角度同时喂入神经网络，同时，同时都使用10个频率进行编码，实验结果如下：



无边界场景Nerf

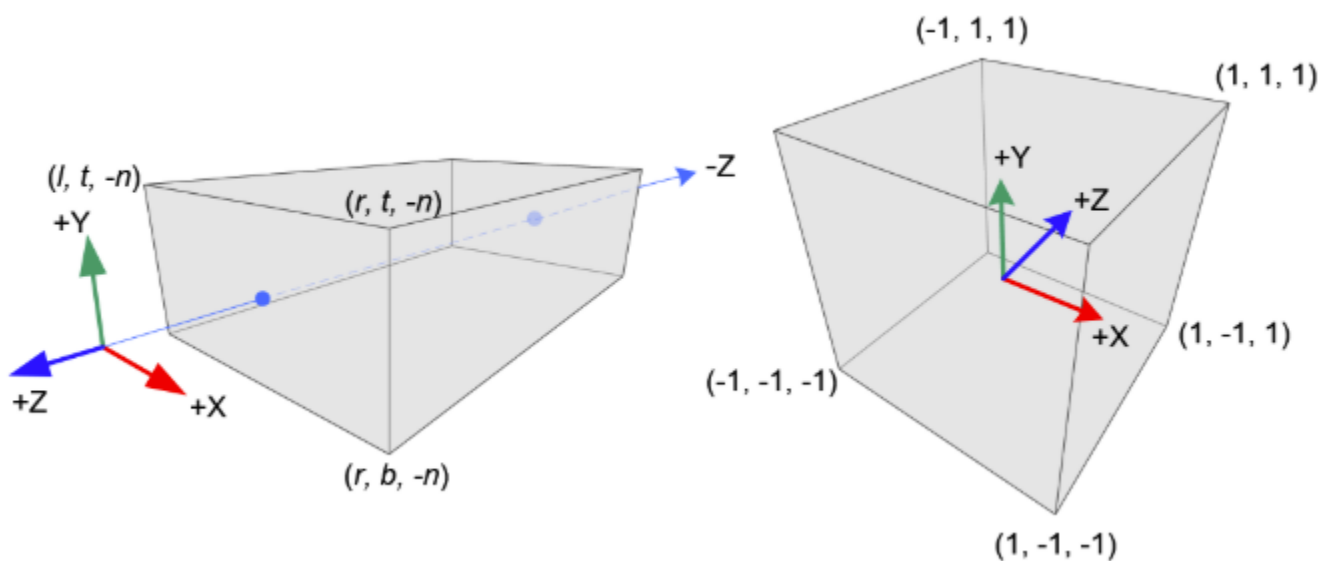
目前Nerf比较适合做这种有边界的场景的重建，但是完成无边界场景，特别是360度无边界场景的重建是比较困难的。



(a) bounding volume for the truck only

(b) bounding volume for the entire scene

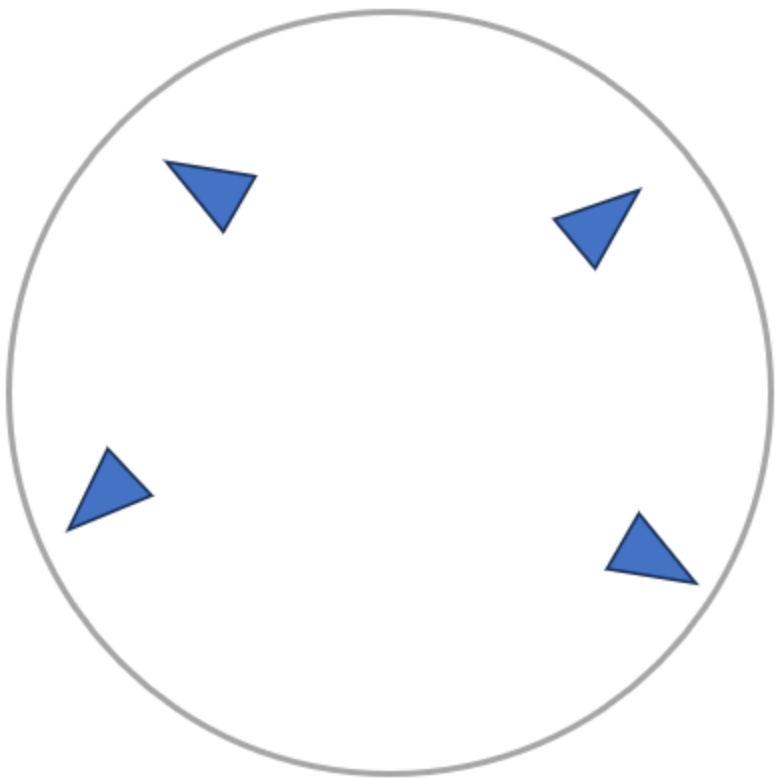
左边情况主要是因为采样时，没采到后面的点，右边情况主要是采样时采得足够远了，但是近处样本不够，造成模糊。当然一种解决的方案是使用NDC坐标，把场景里的物体放到NDC坐标系下，但是NDC坐标系无法覆盖参考视点锥体之外的空间。



Nerf++的解决方案

相机处理

相机pose 归一化，所有相机translation均值作为重参数化后前景单位圆的圆心，距离圆心最远的相机pose到圆心的距离为前景半径，同时scale为1。



如何知道某一点的RGB值

利用球体，把整个物体空间为了前景和背景部分，分别进行处理。

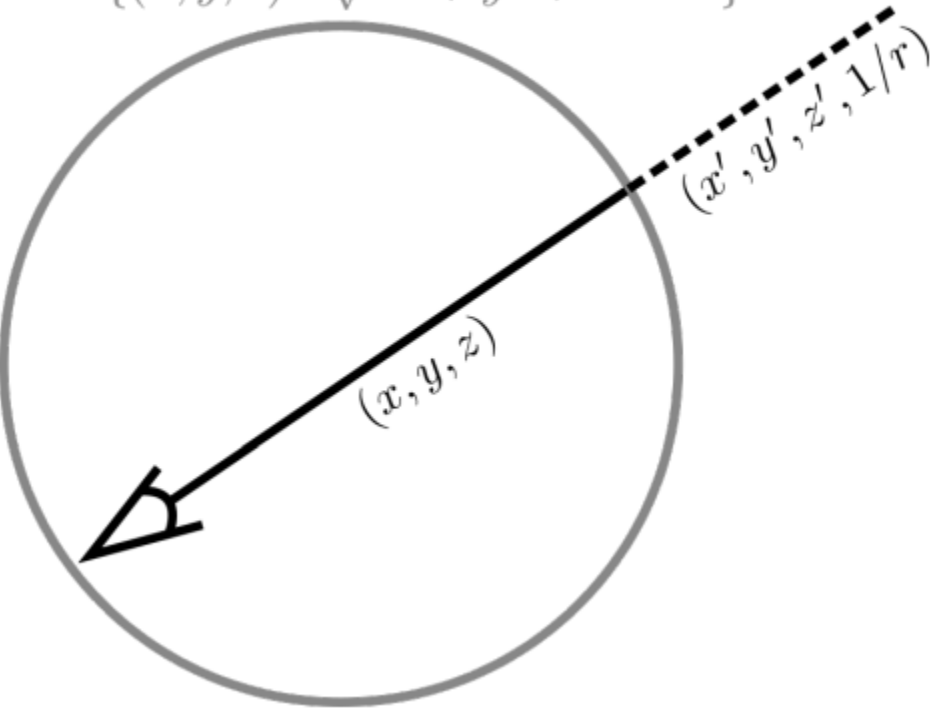
前景

前景与原始nerf处理一样，圆内的点有采样得到xyz坐标与观察角度，经过位置编码之后，扔到MLP网络，得到密度 σ 与颜色值 C 。

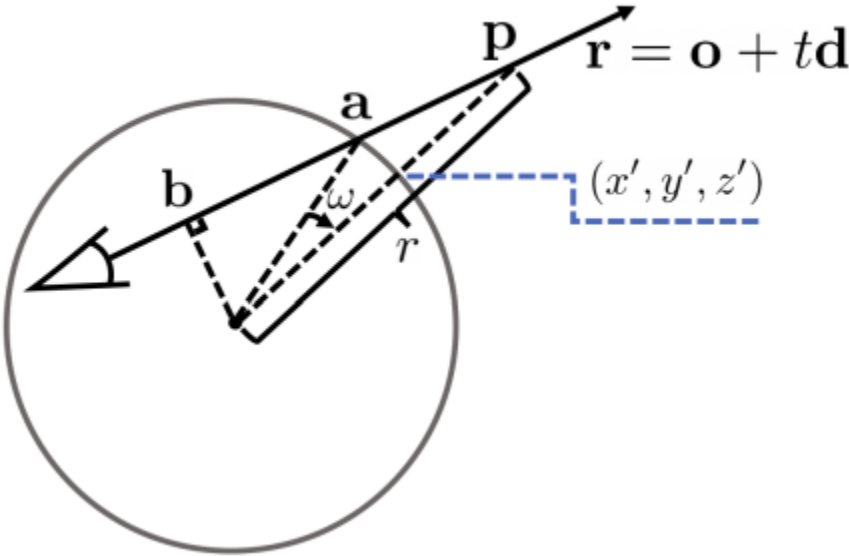
背景

跟前景不太一样的，因为不知道背景这个深度到底有多深，这里使用了一个叫逆球面重参数化的技术，把原来的xyz坐标变成了x'y'z'及 $\frac{1}{r}$ 。

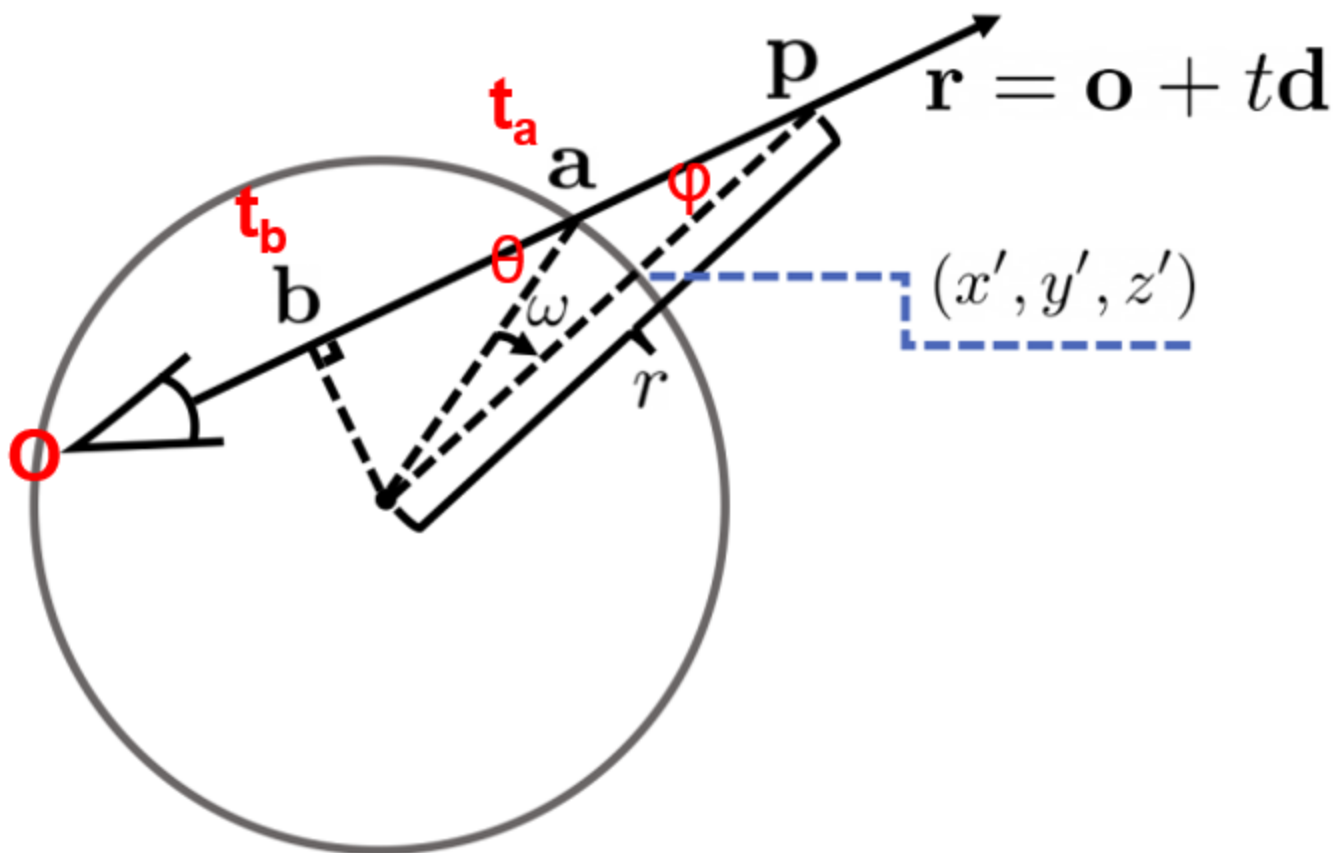
$$B = \{(x, y, z) : \sqrt{x^2 + y^2 + z^2} = 1\}$$



这样的做法，主要是方便采样，当背景的点离前景很近的时候， $\frac{1}{r}$ 的值接近于 1，当背景在无穷远处时， $\frac{1}{r}$ 的值接近于 0，所以可以对 $\frac{1}{r}$ 值进行 0 - 1 的采样。



下面讲如何通过 $\frac{1}{r}$ 的采样值，得到相应的 $x'y'z'$ 。



已知条件：

相机坐标： \mathbf{o} ；

相机观测角度： \mathbf{d} ；

采样深度： $\frac{1}{r}$ ；

令：

$$\mathbf{b} = \mathbf{o} + t_b \mathbf{d}$$

$$\mathbf{a} = \mathbf{o} + t_a \mathbf{d}$$

1. 求 t_b ：

因为 $\mathbf{b} \perp \mathbf{d}$ ，有：

$$\mathbf{d}^T (\mathbf{o} + t_b \mathbf{d}) = 0$$

$$t_b = -\frac{\mathbf{d}^T \mathbf{o}}{\mathbf{d}^T \mathbf{d}}$$

2. 求 t_a ：

t_b 已经求出，所以 \mathbf{b} 已知，由勾股定理：

$$(t_a - t_b)^2 = 1^2 - \mathbf{b}^T \mathbf{b}$$

可以将 t_a 求出，从而 \mathbf{a} 求出。

3. 求 θ ：

由：

$$\sin \theta = \frac{|\mathbf{b}|}{1}$$

$$\theta = \arcsin|\mathbf{b}|$$

3.求 ϕ :

由:

$$\sin\phi = \frac{|\mathbf{b}|}{r}$$

$$\phi = \arcsin\frac{|\mathbf{b}|}{r}$$

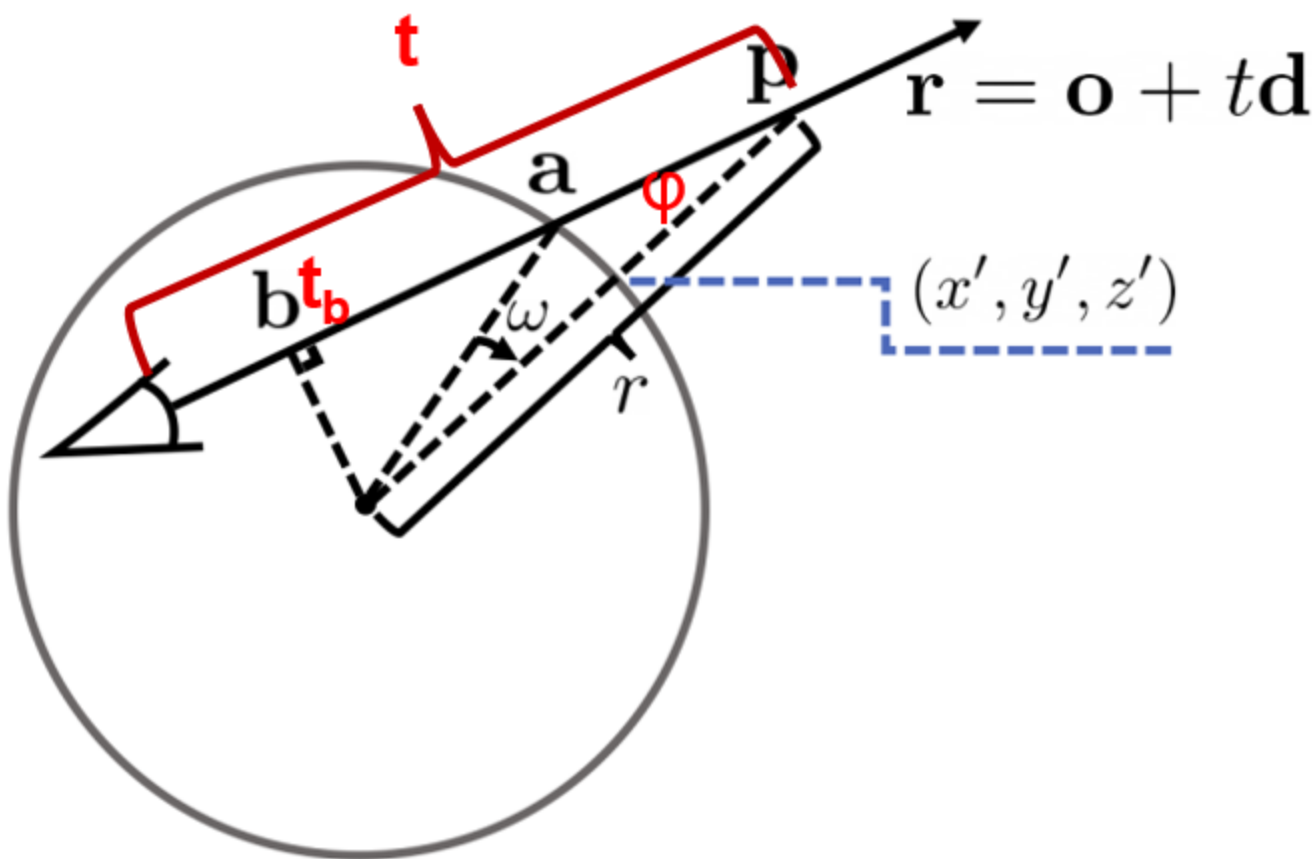
因此,

$$w = \theta - \phi = \arcsin|\mathbf{b}| - \arcsin\frac{|\mathbf{b}|}{r}$$

x', y', z' 的坐标其实就是 \mathbf{a} 向量旋转 w 角度, 根据向量旋转公式可求出。

(以上操作作为代码中的depth2pts_outside函数)

这里就完成了根据 $\frac{1}{r}$ 的采样值, 得到相应的 $x' y' z'$ 。
同时将采样的深度做一点处理, 用真实的深度 t 替代:



$$t = t_b + \cos\phi r * \frac{1}{|d|}$$

最后把采样点的 $x' y' z'$ 与 t 进行位置编码再扔进神经网络, 就可以预测背景的密度与颜色了, 注意位置的输入这里变成了4维的, 所以位置编码后的维度为(4+1024=84)维, 其它与原始Nerf一样。

现在有了前景采样点和背景采样点的密度 σ 与颜色值 C ，怎么求某一点的颜色呢？

$$\begin{aligned} \mathbf{C}(\mathbf{r}) = & \underbrace{\int_{t=0}^{t'} \sigma(\mathbf{o} + t\mathbf{d}) \cdot \mathbf{c}(\mathbf{o} + t\mathbf{d}, \mathbf{d}) \cdot e^{-\int_{s=0}^t \sigma(\mathbf{o} + s\mathbf{d}) ds} dt}_{(i)} \\ & + \underbrace{e^{-\int_{s=0}^{t'} \sigma(\mathbf{o} + s\mathbf{d}) ds}}_{(ii)} \cdot \underbrace{\int_{t=t'}^{\infty} \sigma(\mathbf{o} + t\mathbf{d}) \cdot \mathbf{c}(\mathbf{o} + t\mathbf{d}, \mathbf{d}) \cdot e^{-\int_{s=t'}^t \sigma(\mathbf{o} + s\mathbf{d}) ds} dt}_{(iii)}. \end{aligned} \quad (4)$$

如何训练

大致的训练策略，与原始Nerf一样，具体细节大家可以参考论文与代码。

