论文：Mip-NeRF: A Multiscale Representation for Anti-Aliasing Neural Radiance Fields

地址：https://arxiv.org/pdf/2103.13415v3.pdf

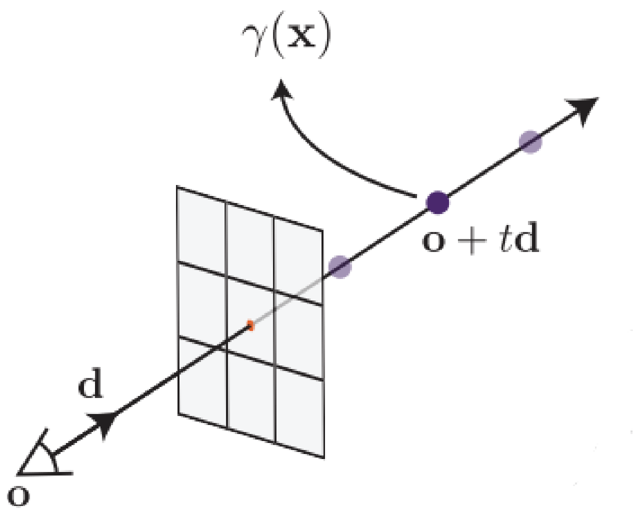
年份：ICCV2021

传统nerf的问题

项目地址：
https://jonbarron.info/mipnerf/

传统的nerf,如果在低分辨下训练，在高分辨率下测试时，会出现模糊的情况。
如果在高分辨率下训练，在低分辨下进行测试的话，会出现锯齿的情况。

问题形成的原因



a) NeRF

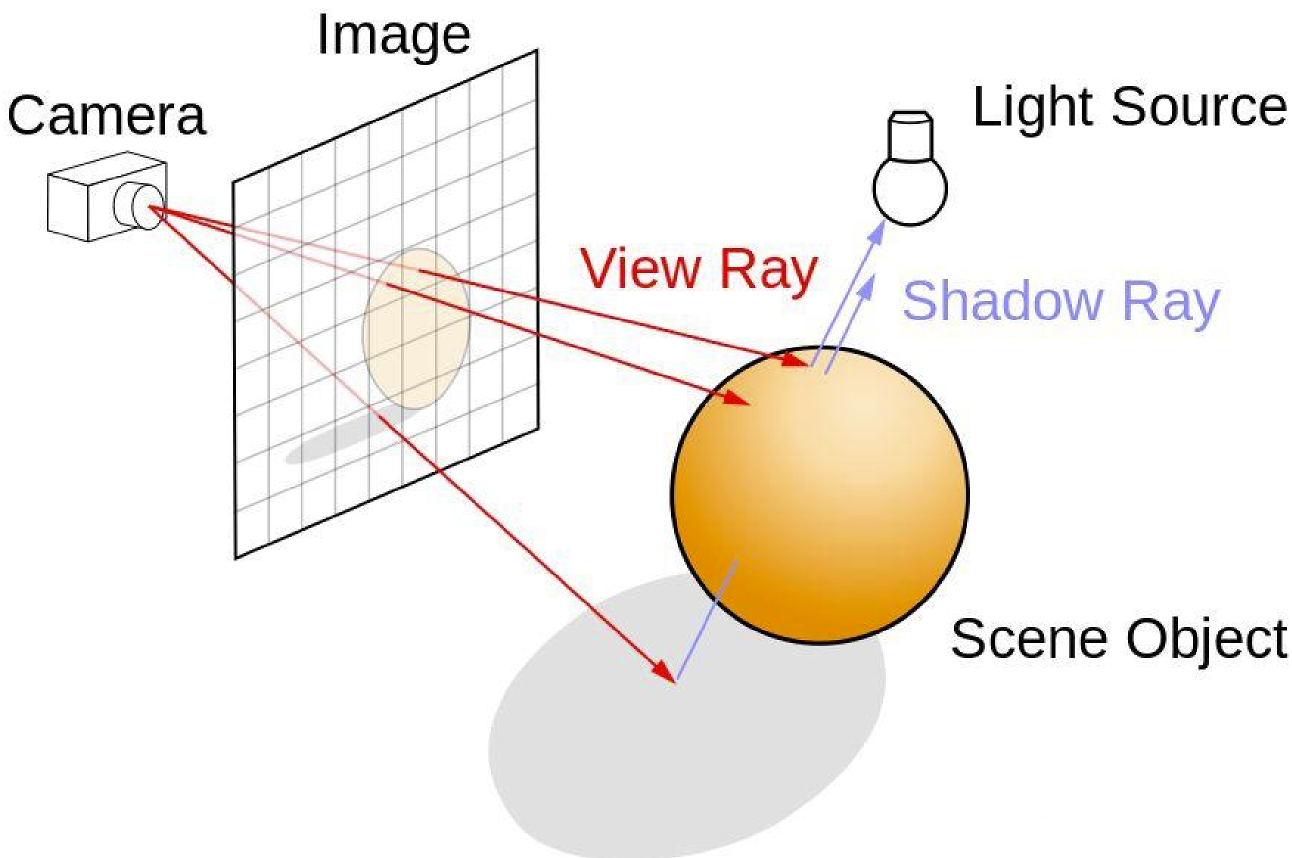
传统的nerf对一个像素点的渲染只采用像素点中心出来的这一根射线上的粒子，若在低分辨率下进行训练，高分辨下可用的粒子数据不够，会产生模糊 ，若在高分辨率下进行训练，低分辨下使用的粒子数据不够，会产生欠采样的问题，产生锯齿。

图形学中的解决方案

Mip-map

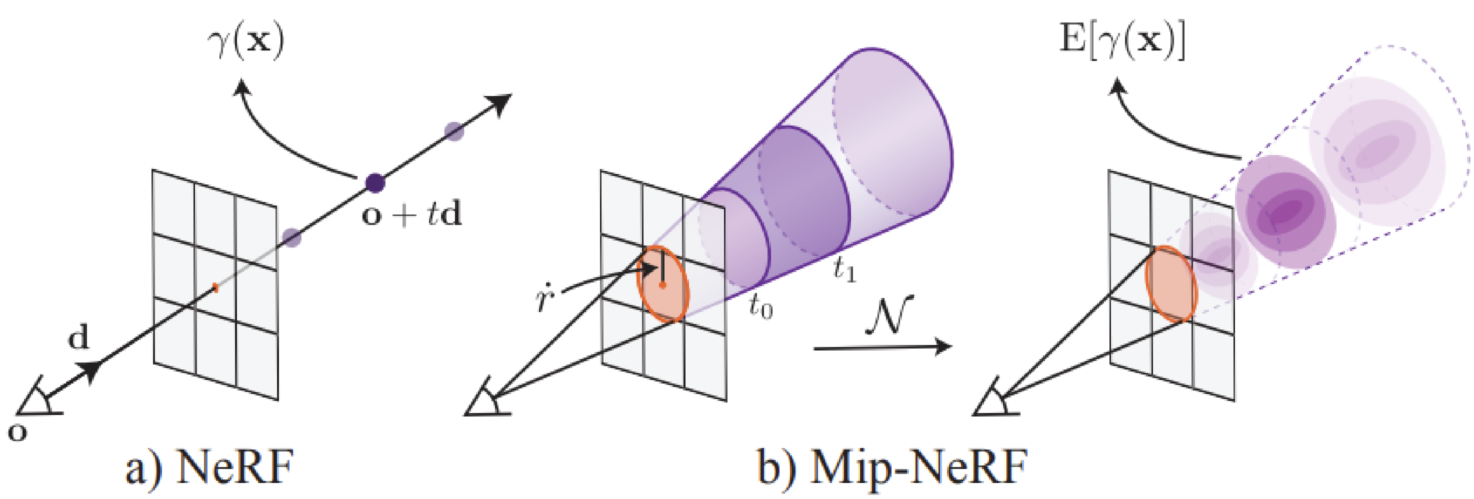
https://www.youtube.com/watch?v=v0OVto8xv_k&t=286s

超采样



作者的方法

如果使用图形学中的Mip-map的方法，需要先预训练不同分辨率下的体素分布，这种可行性不高（可能训练数据只有一个分辨率下的），而且用时长。
如果使用图形学的超采样方法，虽然工作，但是对于每个像素点需要采样多根光线，训练时间直接翻倍。
作者使用了的方法如下：



每个像素点的颜色，由圆锥内的所有粒子所决定，这样像素点的颜色不再依赖于一根射线上的粒子，可以较好的解决锯齿和模糊的问题。

在nerf原始的论文里面，神经网络中输入(x,y,z)以及观察方法，输出相应的位置的粒子的颜色与密度，我们在mip nerf如果也使用这种方法，意味着我们要对一个锥体内的粒子在三维空间进行采样，使用采样的粒子来计算像素点的颜色与密度，这样会使计算量爆炸！

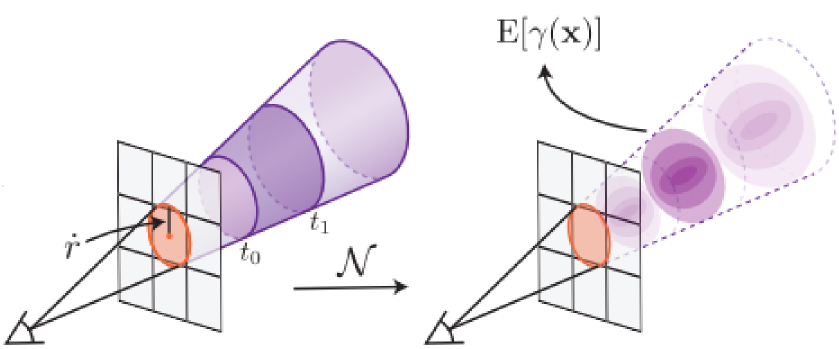
所以论文里将锥体分为一小段一小段，使用三维高斯分布来进行这些锥体段的空间分布，最后在神经网络中输入位置（均值与协方差）与观察角度，可以得到这一坨粒子的颜色期望与密度期望。

论文里涉及的两个最主要问题如下：

1. 如何使用高斯分布对这些截锥体分布进行模拟？
2. 在nerf论文里我们使用了位置编码，把(x,y,z)以及view dir分布进行了高频编码，在mip nerf里输入变成了分布的均值与方差以及观察方向，如果对这些数据进行相应的位置编码？

空间分布模拟

需要将一个均匀分布的截椎体模拟为一个高维的高斯分布



b) Mip-NeRF

为方便计算，首先进行坐标转换，将原来的xyz坐标转换为t,r, θ 坐标系，t表示沿射线方向上两个截面的距离，r表示粒子离中心射线的距离， θ 表示粒子沿中心射线旋转的角度。

$$
\begin{gathered}
x = rtcos\theta \\
y = rtsin\theta \\
z = t
\end{gathered}
$$

由坐标变换的公式可知：

$$
dxdydz = \begin{vmatrix} \frac{dx}{dr} & \frac{dy}{dr} & \frac{dy}{dr} \\ \frac{dx}{dt} & \frac{dy}{dt} & \frac{dy}{dt} \\ \frac{dx}{d\theta} & \frac{dy}{d\theta} & \frac{dy}{d\theta} \end{vmatrix} drd\theta dt = \begin{vmatrix} tcos\theta & tsin\theta & 0 \\ rcos\theta & rsin\theta & 1 \\ -rtsin\theta & rtcos\theta & 0 \end{vmatrix} drd\theta dt = rt^2 drd\theta dt
$$

其截锥体的体积为：

$$
\begin{aligned}
V = \iiint dxdydz = \int_{t_0}^{t_1} \int_0^{2\pi} \int_0^{r^{\cdot}} rt^2 drd\theta dt \\
= \frac{1}{3}\pi r^2 (t_1^3 - t_0^3)
\end{aligned}
$$

均匀分布的截锥体的密度为 $\frac{1}{V}$.
这一个均匀分布在t方向上的一阶矩为：

$$
\begin{gathered}
E(t) = \frac{1}{V} \iiint tdxdydz \\
= \frac{1}{V} \int_{t_0}^{t_1} \int_0^{2\pi} \int_0^{r^{\cdot}} trt^2 drd\theta dt \\
= \frac{3(t_1^4 - t_0^4)}{4(t_1^3 - t_0^3)}
\end{gathered}
$$

在t方向上的二阶矩是：

$$
\begin{gathered}
E(t^2) = \frac{1}{V} \iiint t^2 dxdydz \\
= \frac{1}{V} \int_{t_0}^{t_1} \int_0^{2\pi} \int_0^{r^{\cdot}} t^2 rt^2 drd\theta dt
\end{gathered}
$$

$$= \frac{3(t_1^5 - t_0^5)}{5(t_1^3 - t_0^3)}$$

在x方向上的一阶矩（均值）一定为0，在x方向上的二阶矩为：

$$E(x^2) = \frac{1}{V}\iiint (rtcos\theta)^2 dxdydz$$
$$= \frac{1}{V}\iiint r^2t^2cos^2\theta rt^2 drd\theta dt$$
$$= \frac{3r^{\cdot 2}(t_1^5 - t_0^5)}{20(t_1^3 - t_0^3)}$$

因此，在t(z)方向上的均值为：

$$u_t = \frac{3(t_1^4 - t_0^4)}{4(t_1^3 - t_0^3)}$$

在t(z)方向上的方差为：

$$\sigma_t = \frac{3(t_1^5 - t_0^5)}{5(t_1^3 - t_0^3)} - u_t^2$$

在x方向上和y方向上的均值为：

$$u_x = u_y = 0$$

在x方向上和y方向上的均值为：

$$\sigma_x = \sigma_y = \frac{3r^{\cdot 2}(t_1^5 - t_0^5)}{20(t_1^3 - t_0^3)}$$

当 `t_1和t_0` 离得比较近的情况下，以上式子中分子和分母都会很小，进行除法时，稳定性不好，因此引入
$t_\mu = (t_0 + t_1)/2, t_\delta = (t_1 - t_0)/2.$
将其替换到原式子中，增加运算时的稳定性，替换结果如下：

$$\mu_t = t_\mu + \frac{2t_\mu t_\delta^2}{3t_\mu^2 + t_\delta^2}, \qquad \sigma_t^2 = \frac{t_\delta^2}{3} - \frac{4t_\delta^4(12t_\mu^2 - t_\delta^2)}{15(3t_\mu^2 + t_\delta^2)^2},$$
$$\sigma_r^2 = \dot{r}^2\left(\frac{t_\mu^2}{4} + \frac{5t_\delta^2}{12} - \frac{4t_\delta^4}{15(3t_\mu^2 + t_\delta^2)}\right). \qquad (41)$$

其均值与协方差为：

$$u = \begin{bmatrix} 0 \\ 0 \\ u_t \end{bmatrix}$$

$$\Sigma = \begin{bmatrix} \sigma_x & 0 & 0 \\ 0 & \sigma_y & 0 \\ 0 & 0 & \sigma_t \end{bmatrix}$$

但是这个是在相机坐标下的，我们需要把它转换到世界坐标系下，设摄像机现在摆放的坐标为 $\mathbf{O}$,则在世界坐标系下原点的位置为 $\mathbf{O} + u_t\mathbf{d}$, d 为观察方向（单位向量）。

对xyz坐标先进行归一化处理，归一化后协方差矩阵不发生变化。
当观察方向为 $\mathbf{d}$ 时，则原坐标(归一化后)与新坐标(归一化后)的对应关系为：

$$x- >?$$
$$y- >?$$

$$z- > d$$

假定新坐标的另外两个基为 `u` 与 `v` ,则旧坐标与新坐标之间的变换矩阵为：

$$P = \begin{bmatrix} \mathbf{u} & \mathbf{v} & \mathbf{d} \end{bmatrix}$$

由公式：

$$cov(Ax, Ay) = Acov(x, y)A^T$$

新的协方差矩阵为：

$$\begin{aligned}
\Sigma^{'} &= \begin{bmatrix} cov(u,u) & cov(u,v) & cov(u,w) \\ cov(v,u) & cov(v,v) & cov(v,w) \\ cov(w,u) & cov(w,v) & cov(w,w) \end{bmatrix} \\
&= \begin{bmatrix} cov(Px,Px) & cov(Px,Py) & cov(Px,Pz) \\ cov(Py,Px) & cov(Py,Py) & cov(Py,Pz) \\ cov(Pz,Px) & cov(Pz,Py) & cov(Pz,Pz) \end{bmatrix} \\
&= P \begin{bmatrix} cov(x,x) & cov(x,y) & cov(x,z) \\ cov(y,x) & cov(y,y) & cov(y,z) \\ cov(z,x) & cov(z,y) & cov(z,z) \end{bmatrix} P^T \\
&= P\Sigma P^T \\
&= \begin{bmatrix} \mathbf{u} & \mathbf{v} & \mathbf{d} \end{bmatrix} \begin{bmatrix} \sigma_x & 0 & 0 \\ 0 & \sigma_y & 0 \\ 0 & 0 & \sigma_t \end{bmatrix} \begin{bmatrix} \mathbf{u}^T \\ \mathbf{v}^T \\ \mathbf{d}^T \end{bmatrix} \\
&= \sigma_x \mathbf{u}\mathbf{u}^T + \sigma_y \mathbf{v}\mathbf{v}^T + \sigma_z \mathbf{d}\mathbf{d}^T
\end{aligned}$$

令：

$$\sigma_x = \sigma_y = \sigma_r$$

$$\Sigma^{'} = \sigma_r(\mathbf{u}\mathbf{u}^T + \mathbf{v}\mathbf{v}^T) + \sigma_z \mathbf{d}\mathbf{d}^T$$

由于

$$\begin{aligned}
&= \begin{bmatrix} \mathbf{u} & \mathbf{v} & \mathbf{d} \end{bmatrix} \begin{bmatrix} \mathbf{u}^T \\ \mathbf{v}^T \\ \mathbf{d}^T \end{bmatrix} = \mathbf{I} \\
&\mathbf{u}\mathbf{u}^T + \mathbf{v}\mathbf{v}^T = \mathbf{I} - \mathbf{d}\mathbf{d}^T
\end{aligned}$$

原式为：

$$\Sigma^{'} = \sigma_r(\mathbf{I} - \mathbf{d}\mathbf{d}^T) + \sigma_z \mathbf{d}\mathbf{d}^T$$

位置编码

传统nerf中的位置编码

以位置(x,y,z)为例
令

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$P=\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\\ 2 & 0 & 0\\ 0 & 2 & 0\\ 0 & 0 & 2\\ 2^2 & 0 & 0\\ 0 & 2^2 & 0\\ 0 & 0 & 2^2\\ ... & & \\ 2^{L-1} & 0 & 0\\ 0 & 2^{L-1} & 0\\ 0 & 0 & 2^{L-1}\end{bmatrix}$$

$$r(x)=\begin{bmatrix}sin(P\mathbf{x})\\ cos(P\mathbf{x})\end{bmatrix}$$
$$=\begin{bmatrix}sin(x)\\ sin(y)\\ sin(z)\\ sin(2x)\\ sin(2y)\\ sin(2z)\\ ...\\ sin(2^{L-1}x)\\ sin(2^{L-1}y)\\ sin(2^{L-1}z)\\ cos(x)\\ cos(y)\\ cos(z)\\ cos(2x)\\ cos(2y)\\ cos(2z)\\ ...\\ cos(2^{L-1}x)\\ cos(2^{L-1}y)\\ cos(2^{L-1}z)\end{bmatrix}$$

mip-nerf中的位置编码

传统的nerf的编码是离散的形式，输入为(x,y,z)的离散点，输出为对这个点的编码，但是mip nerf里我们的输入是一坨高斯分布（均值与协方差），如何对这个协方差进行编码?

我们从一维高斯分布来考虑这个问题，假设

$$x服从N(u,\sigma)$$

则:

$$E(sinx)=\int_{-\infty}^{+\infty} sinx\frac{1}{\sqrt{2\pi\sigma}}e^{-\frac{(x-u)^2}{2\sigma^2}}dx$$
$$=\frac{1}{\sqrt{2\pi\sigma}}\int_{-\infty}^{+\infty} sinxe^{-\frac{(x-u)^2}{2\sigma^2}}dx$$
$$=\frac{1}{\sqrt{2\pi\sigma}}\int_{-\infty}^{+\infty} sin(t+u)e^{-\frac{t^2}{2\sigma^2}}dt$$
$$=\frac{1}{\sqrt{2\pi\sigma}}\int_{-\infty}^{+\infty} sintcosue^{-\frac{t^2}{2\sigma^2}}dt+\frac{1}{\sqrt{2\pi\sigma}}\int_{-\infty}^{+\infty} sinucoste^{-\frac{t^2}{2\sigma^2}}dt$$

$$
\begin{aligned}
&= \frac{1}{\sqrt{2\pi\sigma}} cosu \int_{-\infty}^{+\infty} sinte^{-\frac{t^2}{2\sigma^2}}\, dt + \frac{1}{\sqrt{2\pi\sigma}} sinu \int_{-\infty}^{+\infty} coste^{-\frac{t^2}{2\sigma^2}}\, dt \\
&= \frac{1}{\sqrt{2\pi\sigma}} sinu \int_{-\infty}^{+\infty} coste^{-\frac{t^2}{2\sigma^2}}\, dt \\
&= \frac{1}{\sqrt{2\pi\sigma}} sinuR[\int_{-\infty}^{+\infty} e^{it} e^{-\frac{t^2}{2\sigma^2}}\, dt] \\
&= \frac{1}{\sqrt{2\pi\sigma}} sinuR[\int_{-\infty}^{+\infty} e^{i\sqrt{2}\sigma u} e^{-u^2} \sqrt{2}\sigma du] = \\
&= \frac{1}{\sqrt{\pi}} sinuR[\int_{-\infty}^{+\infty} e^{-(u^2 - i\sqrt{2}\sigma u + \frac{1}{2}\sigma^2 i^2)} e^{\frac{1}{2}\sigma^2 i^2} du] \\
&= \frac{1}{\sqrt{\pi}} sinuR[\int_{-\infty}^{+\infty} e^{-(u^2 - i\sqrt{2}\sigma u - \frac{1}{2}\sigma^2)} e^{-\frac{1}{2}\sigma^2} du] \\
&= \frac{1}{\sqrt{\pi}} sinue^{-\frac{1}{2}\sigma^2} R[\int_{-\infty}^{+\infty} e^{(u-\sqrt{2}\sigma i)^2} du] \\
&= \frac{1}{\sqrt{\pi}} sinue^{-\frac{1}{2}\sigma^2} \sqrt{\pi} \\
&= sinue^{-\frac{1}{2}\sigma^2}
\end{aligned}
$$

同样计算：

$$
E(cosx) = cosue^{-\frac{1}{2}\sigma^2}
$$

把这个公式推广到高维的高斯分布上:

$$
\begin{aligned}
E(sinx) = sinue^{-\frac{1}{2}|\Sigma|} \\
E(cosx) = cosue^{-\frac{1}{2}|\Sigma|}
\end{aligned}
$$

如何得到 $E(sin2^i\mathbf{x})$,只需要设

$$
\mathbf{y} = P\mathbf{x}
$$

其中

$$
P = \begin{bmatrix} 2^i & 0 & 0 \\ 0 & 2^i & 0 \\ 0 & 0 & 2^i \end{bmatrix}
$$

$$
\begin{aligned}
E(sin2^i\mathbf{x}) = E(sin\mathbf{y}) \\
= sinu_y e^{-\frac{1}{2}|\Sigma_y|} \\
= sin(Pu)e^{-\frac{1}{2}|P\Sigma_y P^T|}
\end{aligned}
$$