Mip-NeRF

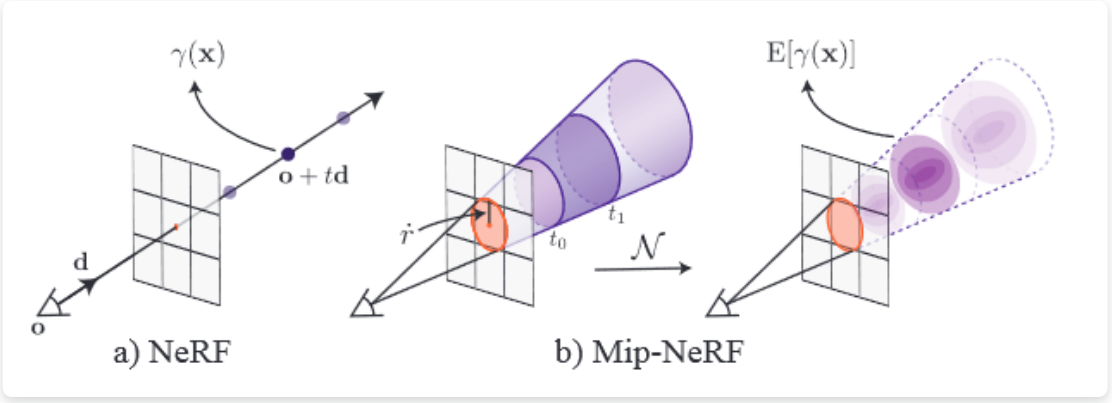
论文：《Mip-NeRF: A Multiscale Representation for Anti-Aliasing Neural Radiance Fields》
地址：https://arxiv.org/abs/2103.13415
年份：ICCV 2021 (Oral, Best Paper Honorable Mention)

Introduction

任务：新视角合成
技术贡献：
(1) 当 NeRF 在不同分辨率下进行训练时，在近处渲染的图像会出现模糊问题，在远处渲染的图像会出现锯齿问题。Mip-NeRF 借鉴 mipmap 的思想，用圆锥体替代光线来解决以上问题。
(2) 提出 integrated positional encoding 来替代原始的位置编码，能准确描述像素包含区域随物体远近变化的关系，避免了远处样本点的突然出现的高频信号的影响。

Method

渲染图像时的抗锯齿方法一般有两种，supersampling 和 prefiltering。第一种方法就是对于一个像素，多发射几条光线，以此提高采样率，接近奈奎斯特频率，但这种方法计算开销会非常大。第二种方法就是对场景进行低通滤波，降低奈奎斯特频率来抗锯齿。例如 mipmap (https://blog.csdn.net/qq_42428486/article/details/118856697)，如果我在近处观察场景，此时屏幕中一个像素可能对应纹理中的一个像素。当我在远处观察场景时，此时屏幕中一个像素对应了纹理中的多个像素，如果随意选取某个值作为像素值，就会造成锯齿问题，解决方法就是对这些像素的值取一个平均。mipmap 就是提前计算好低分辨率下的纹理，然后在渲染时根据距离远近来选择合适分辨率的纹理来得到像素值。
在 Mip-NeRF 中，由于场景是未知的，所以我们是不能提前计算好 mipmap 的，所以 Mip-NeRF 需要在训练时学习到场景的一种 prefiltered 的表示。



Mip-NeRF 将一条光线替换为了一个圆锥体，从光线中采样点这个过程就变成了计算一段段圆台区域的积分，这样就相当于对一段区域取平均，能够有效避免锯齿和模糊。

Cone Tracing

记相机位置 (也是圆锥顶点) 为 $\mathbf{o}$，成像平面位于 $\mathbf{o}+\mathbf{d}$，圆锥在此处的半径为 $\dot{r}$，将这个 $\dot{r}$ 设为世界坐标系下像素的宽度乘上一个 $2/\sqrt{12}$，这个值实际上是让圆锥被像平面截取的面积与像素面积近似所用的一个参数 (https://github.com/google/mipnerf/issues/5)，圆锥被像平面截取的面积为 πr^2，像素面积是 dx^2，有：$\pi r^2 = dx^2 \rightarrow r = dx/\sqrt{pi} \approx dx/0.56$，而 $2/\sqrt{12} \approx 0.577$
首先需要确定位于 $[t_0, t_1]$ 中的点，哪些位于圆台中：

$$\mathrm{F}(\mathbf{x}, \mathbf{o}, \mathbf{d}, \dot{r}, t_0, t_1) = \mathbb{1}\left\{\left(t_0 < \frac{\mathbf{d}^{\mathrm{T}}(\mathbf{x}-\mathbf{o})}{\|\mathbf{d}\|_2^2} < t_1\right) \wedge \left(\frac{\mathbf{d}^{\mathrm{T}}(\mathbf{x}-\mathbf{o})}{\|\mathbf{d}\|_2\|\mathbf{x}-\mathbf{o}\|_2} > \frac{1}{\sqrt{1+(\dot{r}/\|\mathbf{d}\|_2)^2}}\right)\right\}, \quad (5)$$

位于圆台中的点 F 的值为 1，其中第一项是判断点是否在 $[t_0, t_1]$ 区间内，第二项判断点与相机连线与 z 轴形成的夹角是否小于圆锥的最大夹角。

Positional Encoding

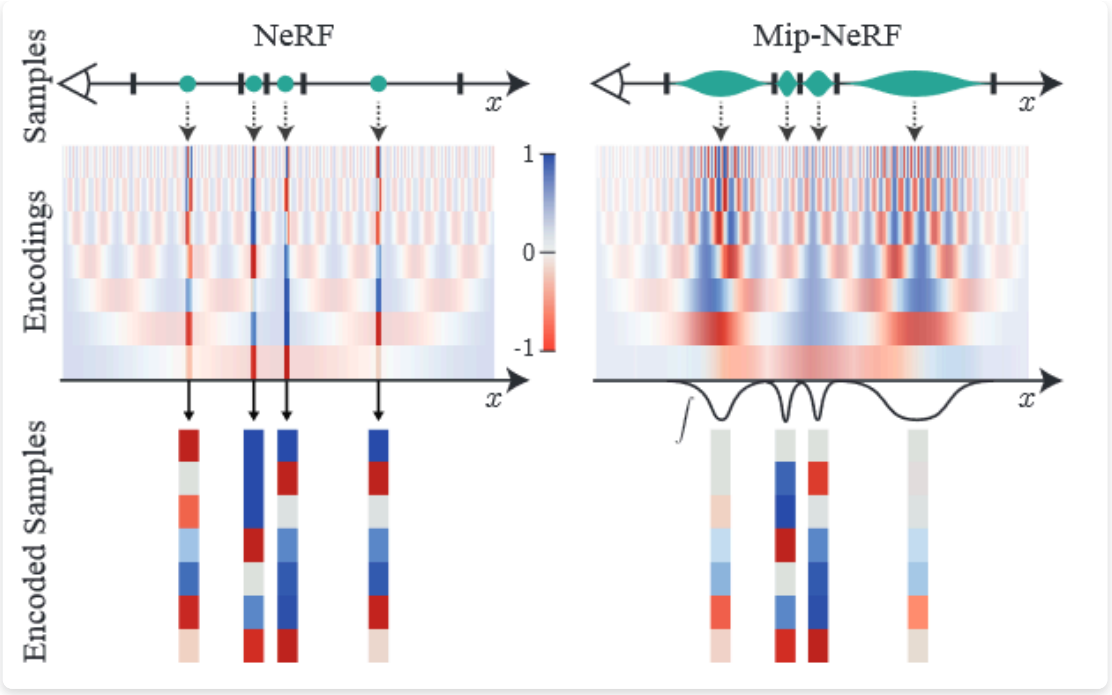
然后就需要考虑如何对这一区域中的点计算位置编码，因为位置编码对 NeRF 性能影响是非常大的。
文章给出了一种简单但有效的计算方式，就是算这一区域位置编码的期望：

$$\gamma^*(\mathbf{o}, \mathbf{d}, \dot{r}, t_0, t_1) = \frac{\int \gamma(\mathbf{x})\, \mathrm{F}(\mathbf{x}, \mathbf{o}, \mathbf{d}, \dot{r}, t_0, t_1)\, d\mathbf{x}}{\int \mathrm{F}(\mathbf{x}, \mathbf{o}, \mathbf{d}, \dot{r}, t_0, t_1)\, d\mathbf{x}}.$$

但问题是上式是没有解析解的，所以考虑用计算方便的多元高斯来近似圆台区域，从而计算这个式子，文章称这个式子为 integrated positional encoding。
首先是确定高斯的均值和方差，详细推导过程在论文附录中。
设高斯的均值为 μ_t，光线上的方差为 σ_t^2，垂直于光线上的方差为 σ_r^2，将相机坐标系下的均值方差转换为世界坐标系下，有

$$\boldsymbol{\mu} = \mathbf{o} + \mu_t \mathbf{d}, \boldsymbol{\Sigma} = \sigma_t^2(\mathbf{d}\mathbf{d}^{\mathbf{T}}) + \sigma_r^2(\mathbf{I} - \frac{\mathbf{d}\mathbf{d}^{\mathbf{T}}}{||\mathbf{d}||_2^2})$$

对于上式的推导，可见 https://github.com/google/mipnerf/issues/6 中作者的回答，$\boldsymbol{\Sigma}$ 中第一项相当于高斯在光线方向 (即 $\mathbf{d}$ 的方向) 上的方差，第二项就是垂直于光线方向上 ($\mathbf{I} - \frac{\mathbf{d}\mathbf{d}^{\mathrm{T}}}{||\mathbf{d}||_2^2}$ 为 $\mathbf{d}$ 的 null space) 的方差。
这里还有一个**问题**是为什么第一项不需要归一化，而第二项也进行了归一化。
可能的原因是，在附录中做坐标转换时有 $(x, y, z) = \phi(r, t, \theta) = (rt\sin\theta, rt\cos\theta, t)$，在第三个方向上的值是 t，也就是说此时坐标系中，光线方向上使用的不是单位向量作为方向向量，而是一个非单位向量 (从 $\boldsymbol{\mu} = \mathbf{o} + \mu_t \mathbf{d}$ 也能看出)，因此在计算世界坐标系下光线方向上的方差时，需要用 $\mathbf{d}$ 的模长做一个“补偿”。而另外两个方向一开始就是使用的单位方向向量，因此需要做归一化。(个人理解)
得到高斯的期望方差后，就可以计算位置编码的期望和方差，这一部分具体可以看论文。

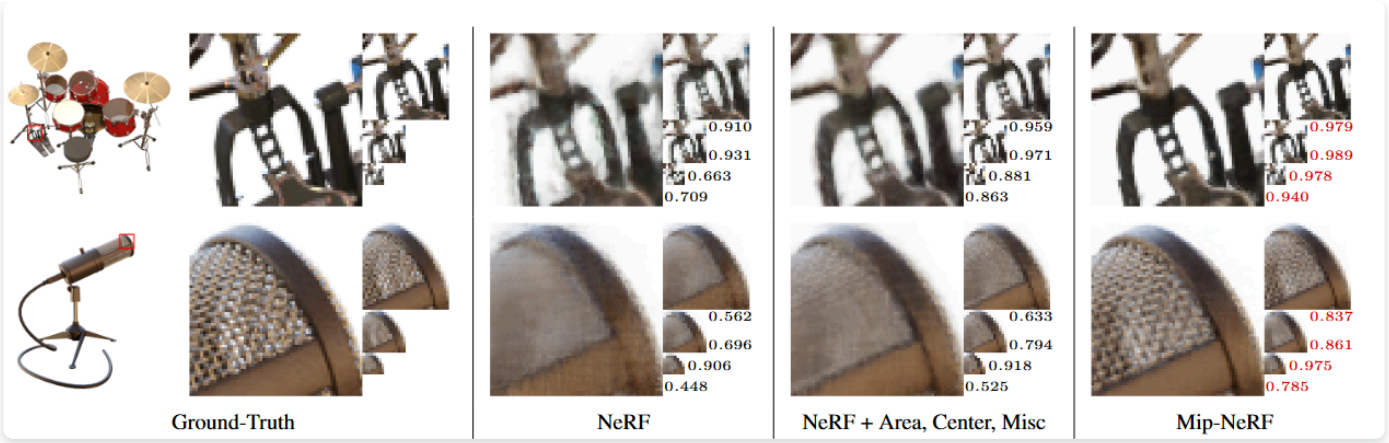


上图展示了 Mip-NeRF 以及 IPE 的作用。”Mip-NeRF 准确描述（进行了合理的建模）了像素包含区域随物体远近变化的关系（近大远小）。对像素来说，越远的区域截断视锥越大，即积分区域越大，此时 Encoding 高频部分的均值迅速衰减到 0（等价于 MipMap 的 Prefiltered 的功能），避免了远处样本点的突然出现的高频信号的影响“”NeRF 则没有这个概念（可以理解为相比 Mip-NeRF，NeRF 对深度没有那么敏感）。NeRF 的 PE 会导致同一个点在不同光线/同一光线不同远近上也会产生一样的 Encoding 结果表示这一段光线的特征。当这一样本点处于较远位置，但它又具有高频信号时，则不利于 NeRF 的学习（因为越远的点应当提供越少的信息量，但这种采样编码的结果违背了这一原则，可以理解为编码信号的走样）“”所以文章 Intro 就指出当数据集中有距离目标物体远近不一以及分辨率不一的图片时，NeRF 就容易产生失真。得益于 IPE，Mip-NeRF 能够只用一个 MLP 来实现 Coarse-to-Fine，因为此时采样不再会导致信号的失真，MLP “看到” 的是一样的频率信号。“(这段话来自 https://zhuanlan.zhihu.com/p/614008188)

IPE 相比 PE 能够有效避免远处样本点的突然出现的高频信号的影响，这样就可以不需要人为调整位置编码中 L 这个超参数。

Experiments

原始 NeRF 使用的 Blender dataset 有一个特点是所有相机都有相同的焦距和分辨率，且与物体的距离都是相同的。这样就掩盖了 NeRF 在多分辨率图像下训练表现不佳的缺点。因此作者重新构造了一个 multiscale Blender benchmark。



从实验效果来看，NeRF 在近处图像会存在模糊问题，在远处图像则有锯齿问题。
实验中作者也对原始 NeRF 进行了几处改动，提升了原始 NeRF 的性能，一个是调整不同分辨率图像的 loss 的 scale，例如对于 $1/4$ 图像，它的 loss 会乘上 16，使其影响与高分辨率图像相同。另一个是调整光线发出的位置，原始 NeRF 实现时，光线是从每个像素的左上角发出的，这里改成了从每个像素的中心处出发。

Limitations

https://zhuanlan.zhihu.com/p/614008188 中也提到了 Mip-NeRF 的缺点：
Integrated Positional Encoding (IPE) 比 Positional Encoding 运算复杂度稍高（但单个 MLP 在计算资源层面的优势弥补了这一劣势）。
Mip-NeRF 相比 NeRF 能够非常有效且准确地构建 Multi-View 与目标物体的关系，但这也意味着相机标定误差（即相机 Pose 的偏差）会更容易使 Mip-NeRF 产生混淆，出现更严重的失真。
很多研究者观察到了这一现象，侧面说明了高质量的 NeRF 重建的前提是实现准确的相机标定，于是研究者们后续提出了一系列的 Self-Calibration 的工作。
同理，当拍摄过程中存在运动模糊 (motion blur) 、曝光等噪声时，Mip-NeRF 也会很容易受到影响。只有当图片成像质量高且相机姿态准确时，Mip-NeRF 才能实现非常棒的效果。后续关于 Self-Calibration、deblur 和曝光控制等工作在一定程度上也是通过更改网络和设置使得 NeRF 能够对这些因素鲁棒，增加容错率（减少失真）。