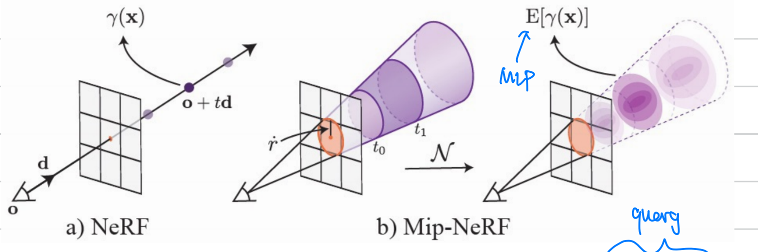
Abstract and Intro

介绍 nerf family.

不同分辨率导致的问题. 如果 training 和 testing 分辨率不同. 解决方案: { supersampling : too costly ; mipnerf } 参考 rendering 中的两种解决方法. mipmap 是一种 pre-filtering 方法

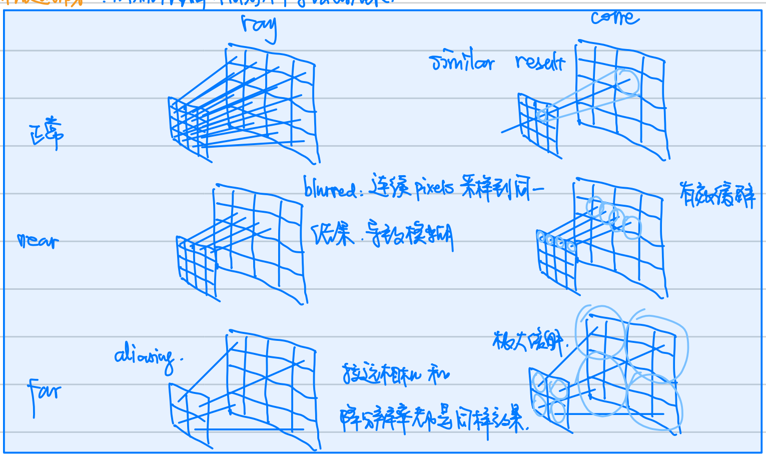
导致: { blurred : 升 reso testing close-up views ; aliased : 降 reso testing distant views } 介绍这两种 results.

mipnerf input 是 Gaussian 参数



nerf : query 点. (o+td) mipnerf : conical frustum $\xrightarrow{\text{IPE}}$ multi Gaussian

介绍这部分. 从示例说明不同分辨率导致的结果.



Related Work

anti-aliasing in rendering

两种基本方法 { supersampling (multi rays a pixel → offline) ; pre-filtering (low pass images → realtime) } 直观上. prefiltering tracing a cone instead of ray

mipnerf 也有和 multiscale 不同的地方: 介绍异同 介绍 mipmap

1. 不会预先计算 map. 因为不是 rendering.

2. 连续而不是 discrete. (mipmap like) mipnerf 是自由的 scale.

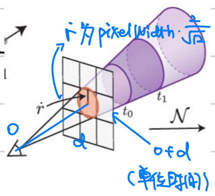
Scene representations for view synthesis

简单介绍

说在 nerf 只能靠 supersampling 来做 anti-aliased. (bad!)

Method

与 nerf 对比: { point → conical frustum also single model ; positional encoding → IPE }



cone tracing & positional encoding

讨论一个 pixel 的圆锥

落在一个圆锥内的点可表示为:

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

点 x 是否在 $(o,d,\dot r,t_0,t_1)$ 内? 时间范围内 在锥内 比较 cos → 大N.

得到这些点 x. 如何对它们 PE? 即: 如何计算一个圆锥的 PE?

$$\gamma^*(o,d,\dot r,t_0,t_1)=\frac{\int\gamma(x)F(x,o,d,\dot r,t_0,t_1)\,dx}{\int F(x,o,d,\dot r,t_0,t_1)\,dx}$$ 全部相加 (PE 相加) $\xrightarrow{\text{开销太大}}$ 用 IPE 近似

Integrated positional encoding (IPE)

核心内容.* 如何把锥体近似成多维高斯? 要计算所有点的均值和方差 (mean and covariance of F(x,·))

所有点又是已知量 $o.d.\dot r.t_0.t_1$ 的示性函数. 因此是在对 $o.d.\dot r.t_0.t_1$ 编码! 即 mipnerf 的 input! 那么如何编码?

基于以上 9 dim 信息. 继续推导:

mid point : $t_\mu=(t_0+t_1)/2$; half-width : $t_\delta=(t_1-t_0)/2$ } math 公式得到: { μ_t : mean distance along ray ; σ_t^2 : variance along ray ; σ_r^2 : variance perpendicular to ray } conical frustum 的坐标空间转世界空间 (world coord) { $\mu=o+\mu_t d \in\mathbb{R}^3$; $\Sigma=\sigma_t^2(dd^T)+\sigma_r^2\left(I-\frac{dd^T}{\|d\|_2^2}\right)\in\mathbb{R}^{3\times3}$ } 这就是圆锥内所有点的均值和方差 用于近似多维高斯, 即求 IPE.

IPE 的内涵就是一个范围内的 expectation! 不是一个实际值 (如 nerf 那样)

对 PE 进行重写: $x\in\mathbb{R}^{3\cdot1}\to\gamma(x)\in\mathbb{R}^{6L\cdot1}$ γ(x) 的内涵是用 P (一种 coding) 和 sin.cos 对 x 进行 lifting ⟶ 现在我们对 μ 和 Σ 进行 lifting, 而不是 x. 结果则为:

PE: $\gamma(x)=[\sin(x),\cos(x),\cdots\sin(2^{L-1}x),\cos(2^{L-1}x)]^T\in\mathbb{R}^{6L\cdot1}$

PE 的 Fourier 形式: $\gamma(x)=\begin{bmatrix}\sin(Px)\\\cos(Px)\end{bmatrix}$. $P=\begin{bmatrix}1&&&2&&&&2^{L-1}&&\\&1&&&2&&\cdots&&2^{L-1}&\\&&1&&&2&&&&2^{L-1}\end{bmatrix}^T$ { $P\in\mathbb{R}^{3L\cdot3}$; $Px\in\mathbb{R}^{3L\cdot1}$; $\gamma(x)\in\mathbb{R}^{6L\cdot1}$ }

{ $\mu_\gamma=P\mu$; $\Sigma_\gamma=P\Sigma P^T$ } 高斯近似锥体编码后的均值和方差

那期望是什么? 即 IPE 是什么?

数学上: $E_{x\sim\mathcal N(\mu,\sigma^2)}[\sin(x)]=\sin(\mu)\exp(-\tfrac12\cdot\sigma^2)$

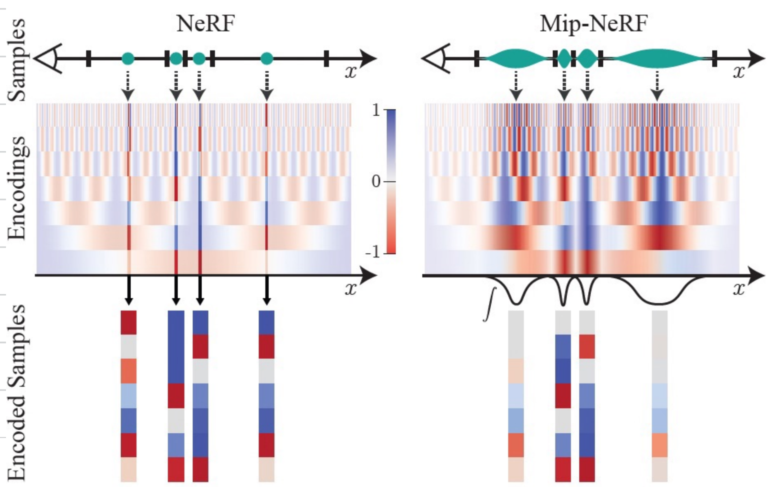
$E_{x\sim\mathcal N(\mu,\sigma^2)}[\cos(x)]=\cos(\mu)\exp(-\tfrac12\cdot\sigma^2)$

对应我们的多维情况: $\gamma(\mu,\Sigma)=E_{x\sim\mathcal N(\mu_\gamma,\Sigma_\gamma)}[\gamma(x)]=\begin{bmatrix}\sin(\mu_\gamma)\circ\exp(-(1/2)\mathrm{diag}(\Sigma_\gamma))\\\cos(\mu_\gamma)\circ\exp(-(1/2)\mathrm{diag}(\Sigma_\gamma))\end{bmatrix}$

期望值 ; x 服从 Gaussian 分布 ; 位置编码 ; ↳ element-wise

$\mu_\gamma\in\mathbb{R}^{3L}$; $\Sigma_\gamma\in\mathbb{R}^{3L\cdot3L}$; $\gamma(\mu,\Sigma)\in\mathbb{R}^{6L}$

实际: $\Sigma_\gamma=P\Sigma P^T$ 不用计算. 直接算 $\mathrm{diag}(\Sigma_\gamma)$ $\mathrm{diag}(\Sigma_\gamma)=[\mathrm{diag}(\Sigma),4\,\mathrm{diag}(\Sigma),\cdots4^{L-1}\mathrm{diag}(\Sigma)]^T\in\mathbb{R}^{3L}$ 其中 $\mathrm{diag}(\Sigma)=\sigma_t^2(d\circ d)+\sigma_r^2\left(1-\frac{d\circ d}{\|d\|_2^2}\right)\in\mathbb{R}^3$



介绍:

高频可以很大 (large L). 反正平均下来值很小.

横向比较: 高频相比. 低频相似.

纵向比较: 频率越大 encoding 值越小. (越无代表性. 因为过大的 averaging)

architecture

Sampling:

nerf : n samples for t_k.

mip : n+1 samples for t_k. get intervals

仍然分为 coarse 和 fine. 用 λ 控制 loss balance.

t_c (128) → inverse sample 128 t_f

Sample t_f 之前. 修改 weights from t_c:

$w'_k=\frac12(\max(w_{k-1},w_k)+\max(w_k,w_{k+1}))+\alpha$.

{ padding α : 0.01 ; a balance between N_c and N_f : ; α 越大. fine sample 会更 uniform. }