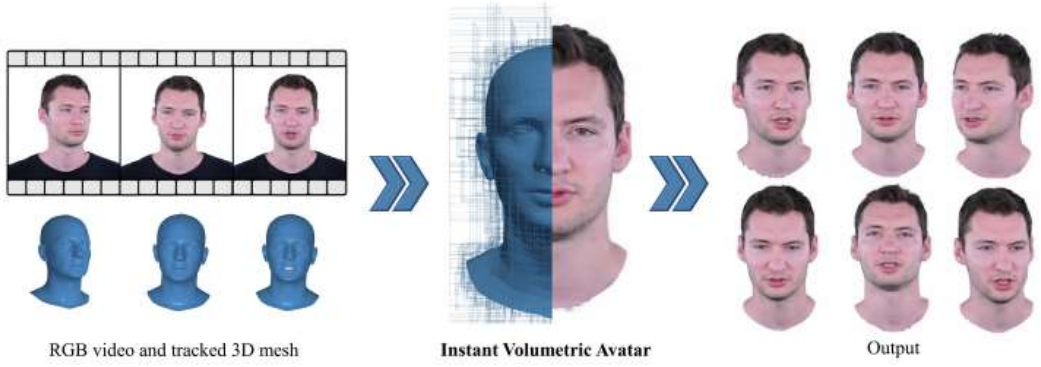
# NeRF与人体重建

汇报人：余斐然

2023/8/16




# 目录

- 体渲染(Volume Rendering)
- NeRF
- Instant-NGP
- NeRF在人体重建的应用

# 体渲染(Volume Rendering)

原本用于渲染半透明材质，如烟、云、玉等

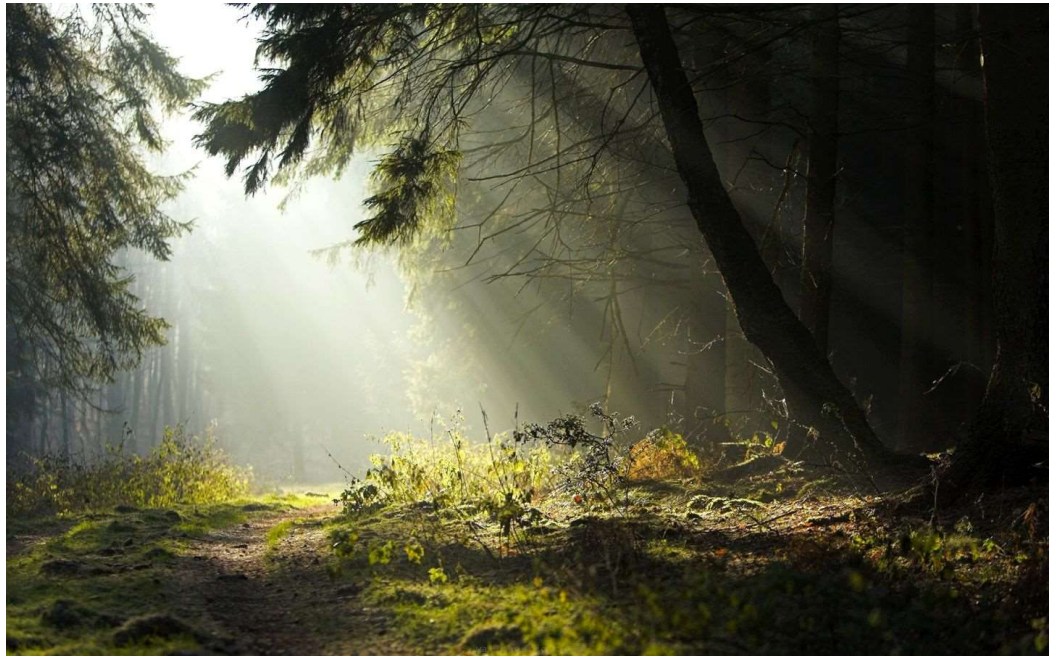
由Nelson Max于1995年提出的近似模型

**Optical Models for Direct Volume Rendering**

Nelson Max

University of California, Davis, and

Lawrence Livermore National Laboratory



# 体渲染(Volume Rendering)原理

体渲染本质上是对光子穿过介质的过程的近似模拟.

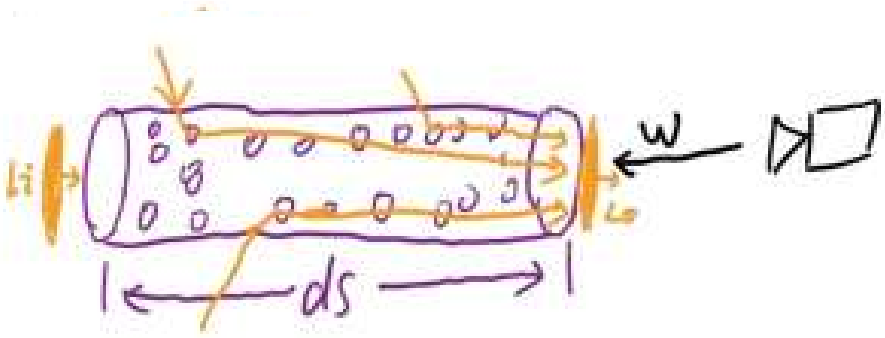
由四部分组成：$L_i$为入射光，$L_0$为出射光

1.吸收　　(absorption)

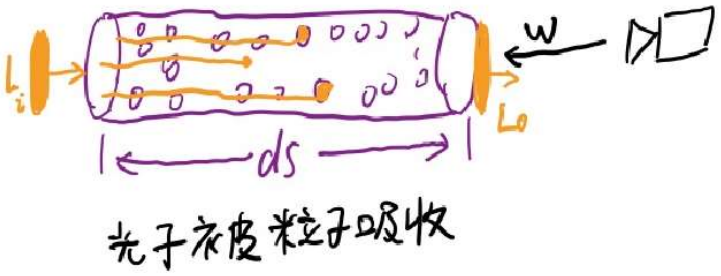
2.放射　　(emission)

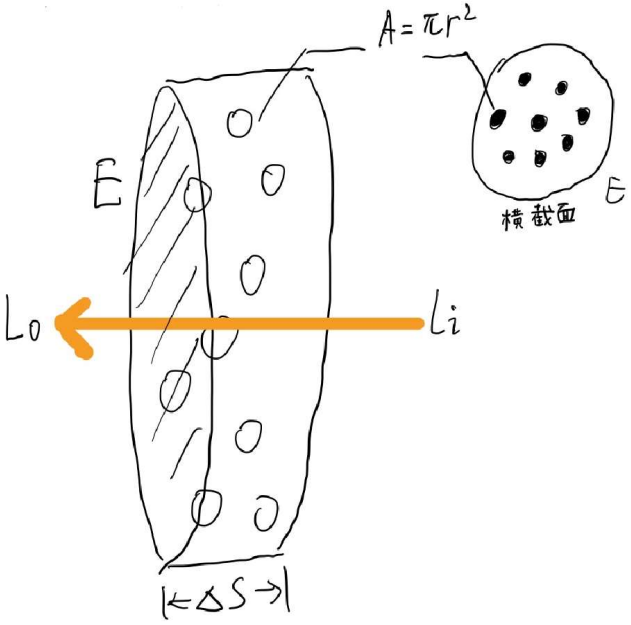
3.外散射　(out-scattering)

4.内散射　(in-scattering)



# 体渲染-吸收(absorption)




设粒子是半径为r的球体，则每个粒子在材质中的横截面的投影面积为$A = \pi r^2$.

设一个足够薄的材质圆柱宽度为$\Delta s$（薄到粒子之间不会互相重叠），
底面积为E，其粒子密度为$\rho$,则:

圆柱体积为$E\Delta s$ ->在圆柱内的粒子数为$\rho E\Delta s$->粒子总遮挡面积为$\rho E\Delta sA$->
遮挡面积占整个底面积的比例为$\frac{\rho E\Delta sA}{E} = \rho\Delta sA$

即，从一段发射大量光子，另一端接收时，会有$\rho A\Delta s$比例的衰减，
有:

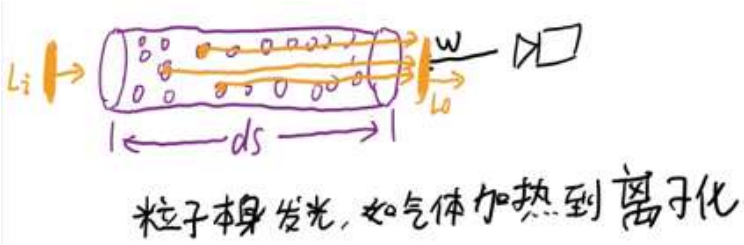
$$I_0 - I_i = \Delta I = -\rho(s)AI(s)\Delta s$$

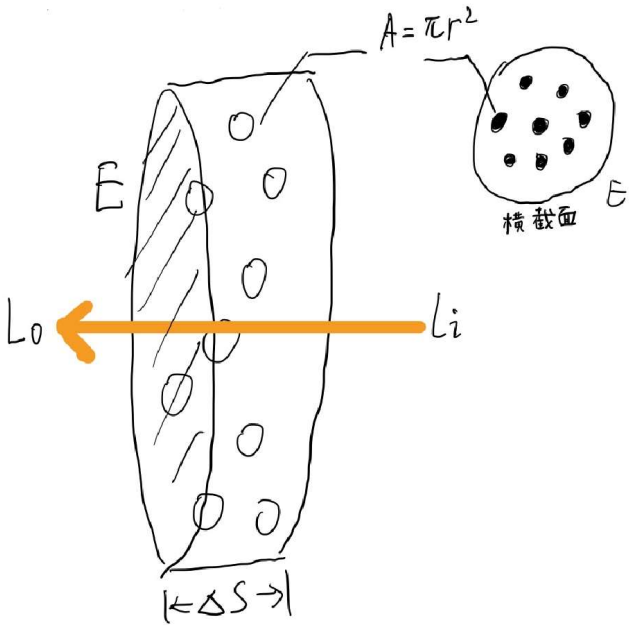
其中s为距离

设$\tau_a(s) = \rho(s)A$,吸收的常微分方程为:

$$\frac{dI}{ds} = -\rho(s)AI(s) = -\tau_a(s)I(s)$$

# 体渲染-放射(emission)



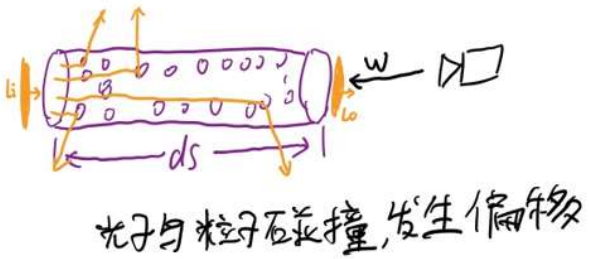




设粒子单位横截面积发光强度为$I_e$，粒子总数为$\rho E\Delta s$，总光强为$I_e\rho E\Delta s$，则接收到的平均光强为$\frac{I_e\rho AE\Delta}{E} = I_e\rho A\Delta s$。

放射部分的常微分方程为：

$$\frac{dI}{ds} = \mathrm{I_e}(s)\rho(s)A = I_e(s)\tau_a(s)$$

# 体渲染-外散射(out-scattering)



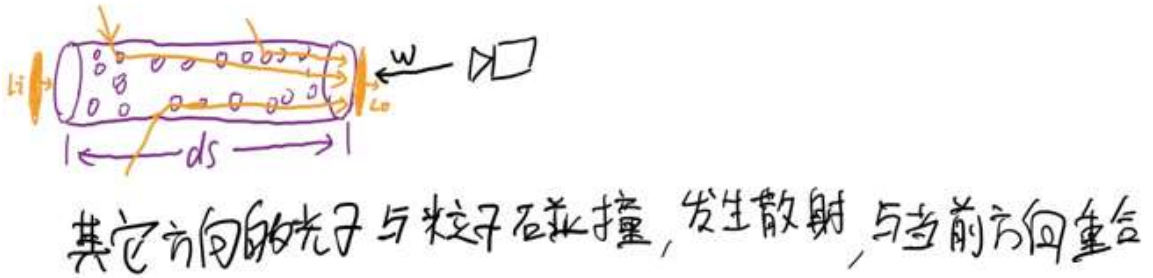

与吸收相似，外散射程度和光学厚度相关，用$\tau_s$表示外散射对光线削弱的比例

所以外散射部分的常微分方程为：

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

# 体渲染-内散射(in-scattering)

近似地，认为其他光路辐射强度为$I_s$




设散射到当前光路时的能量损失为$\tau_s$（与外散射系数一致，因为都是散射）

所以，内散射部分的常微分方程为：

$$\frac{dI}{ds} = \tau_s(s) I_s(s)$$

# 体渲染方程

得到标准的体渲染方程

$$\frac{dI}{ds} = -\tau_a(s)I(s) - \tau_s(s)I(s) + \tau_a(s)I_e(s) + \tau_s(s)I_s(s)$$

吸收 absorbing　　外散射 out-scattering　　放射 emission　　内散射 in-scattering

NeRF使用简化版本的体渲染方程（定义$\tau_t = \tau_a + \tau_s$）：

$$\frac{dI}{ds} = -\tau_t(s)I(s) + \tau_a(s)I_e(s) + \tau_s(s)I_s(s)$$

解出该微分方程，有：

$$I(s) = \int_0^s e^{-\int_0^t \tau_t(u)du}[\tau_a(t)I_e(t) + \tau_s(t)I_s(t)]dt + I_0 e^{-\int_0^s \tau_t(t)dt}$$

其中，$I_0$为入射光强度，NeRF将其看做背景光.

NeRF进一步简化模型：设$\tau_t$、$\tau_a$、$\tau_s$都相等，用$\sigma$表示，且令$C = I_e + I_s$，则有：

$$I(s) = \int_0^s e^{-\int_0^t \sigma(u)du}[\sigma(t)C(t)]dt + I_0 e^{-\int_0^s \sigma(t)dt}$$

NeRF论文中的公式1-> $= \int_0^s T(t)\sigma(t)C(t)dt + T(s)I_0$，其中$T(s) = e^{-\int_0^s \sigma(t)dt}$

# 计算机中的体渲染

计算机计算时要进一步离散化，思路为：

1.将光路[0,s]划分成N个相等的区间$[t_n, t_{n+1}]$

2.计算每个区间$[t_n, t_{n+1}]$的辐射强度$I(t_n \to t_{n+1})$

3.把N个区间辐射值相加，得到最终的光强

得到 $I(t_n \to t_{n+1})$ 后，进行累加：

$$I(s) = \int_0^s T(t)\sigma(t)c(t)dt + T(s)I_0$$

$$\approx \sum_{n=1}^{N} T(0 \to t_n)(1 - \exp(-\sigma_n(t_{n+1} - t_n)))c_n + T(s)I_0$$

假设 $\delta_n = t_{n+1} - t_n$，$\delta_n$ 表示最小区间长度，我们令 $T_n = T(0 \to t_n)$，有

$$T_n = \exp\left(-\int_0^{t_n} \sigma(u)du\right)$$

$$\approx \exp\left(-\sum_{k=1}^{n-1} \sigma_k \delta_k\right)$$

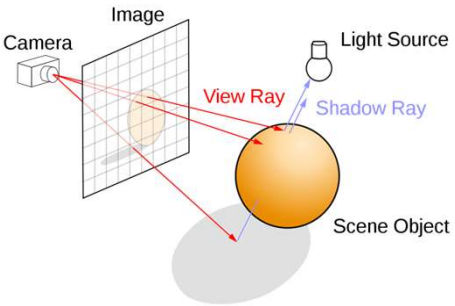
代入 $I(s)$，有

$$I(s) = \sum_{n=1}^{N} T_n(1 - \exp(-\sigma_n\delta_n))c_n + T(s)I_0$$

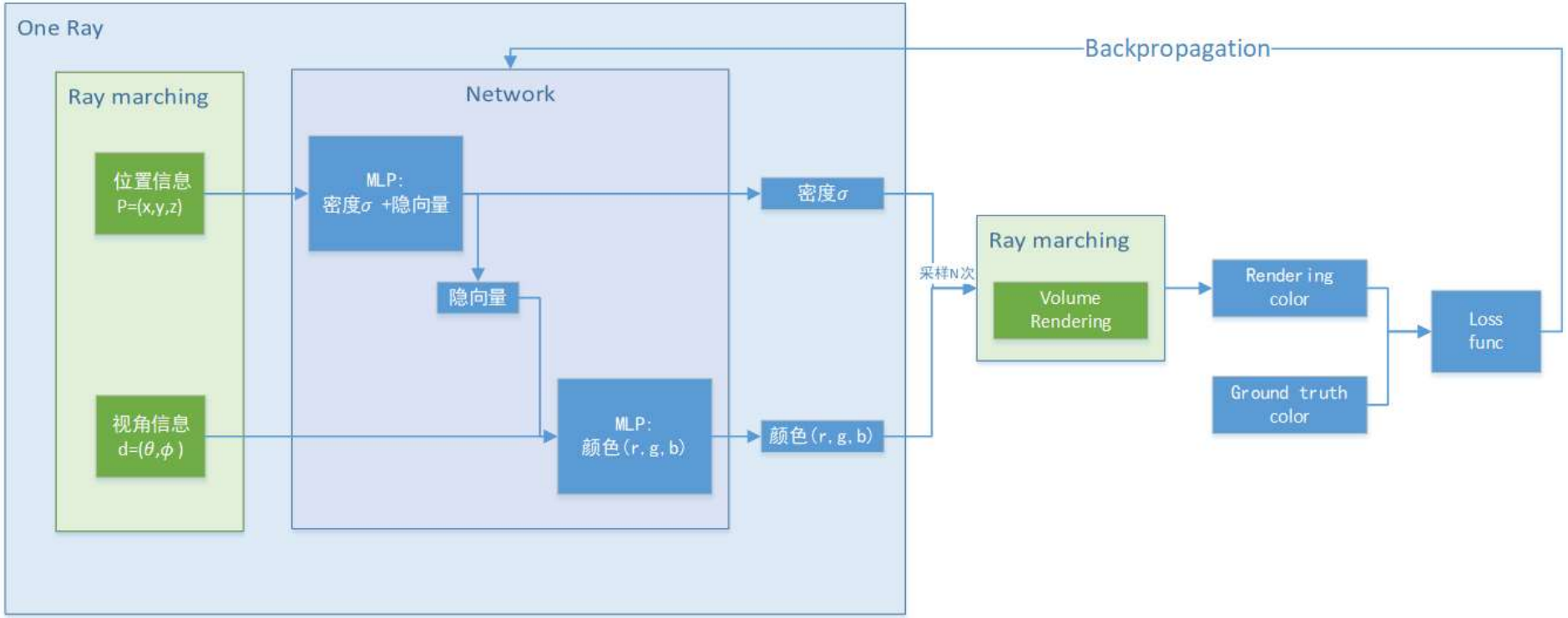
最终得到 NeRF 论文中的公式(3)

$$\hat{C}(r) = \sum_{i=1}^{N} T_i(1 - \exp(-\sigma_i\delta_i))c_i \text{ , where } T_i = \exp\left(-\sum_{k=1}^{n-1} \sigma_k\delta_k\right)$$

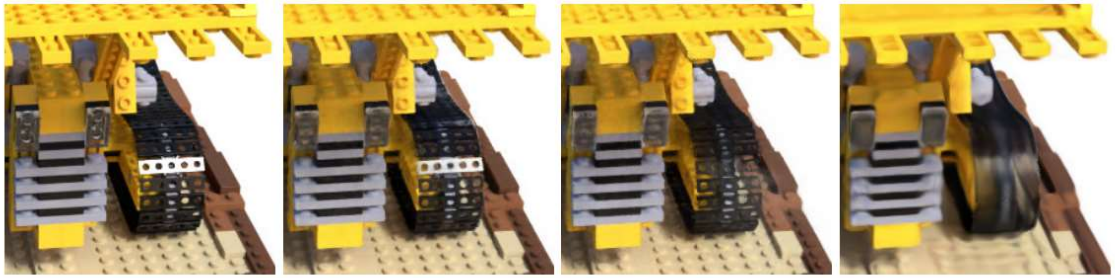
# NeRF（neural radiance field ）




NeRF本质上是使用神经网络（neural）表示辐射场(radiance field），
然后使用体渲染方法进行渲染得到结果与ground truth进行比较

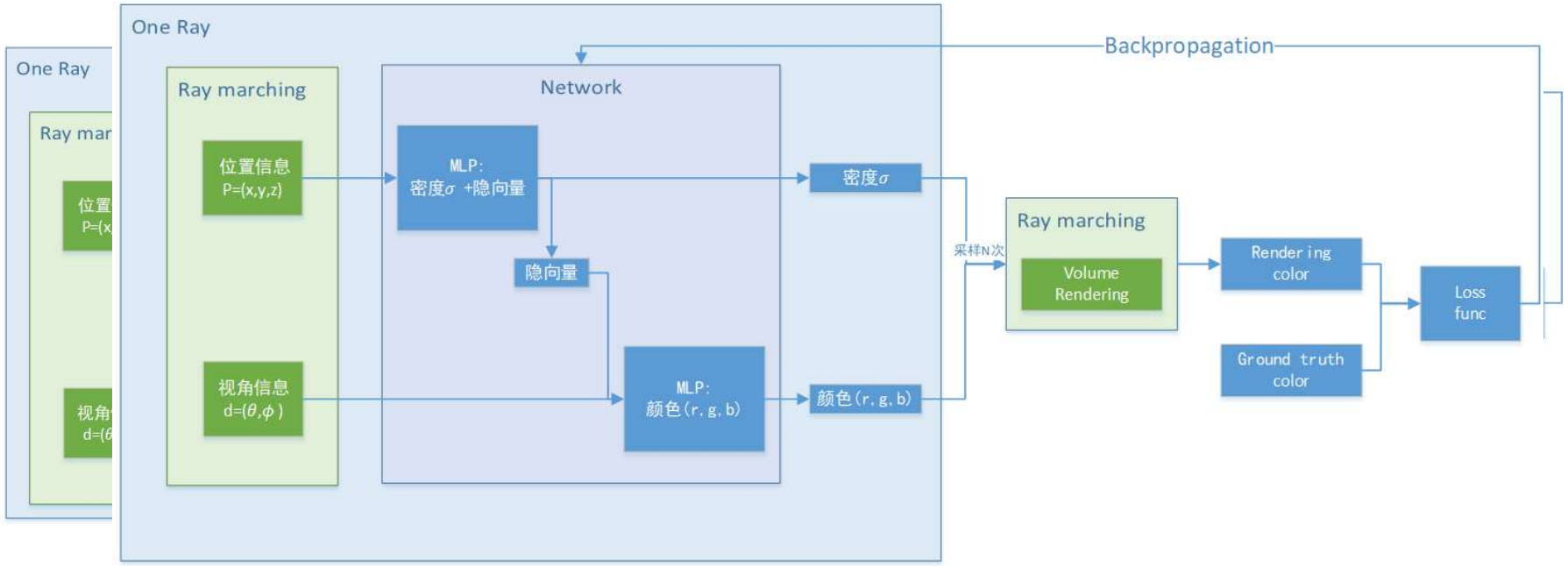


# NeRF改进（1）Encoding




原始版本的NeRF在高频的细节重建的效果不太好，NeRF通过编码升维提高其表现。

编码公式为$\gamma(p) = （\sin(2^0 p), \cos(2^0 p), \ldots \sin(2^{2L-1} p), \cos(2^{2L-1} p))$



# NeRF改进（2）分层采样

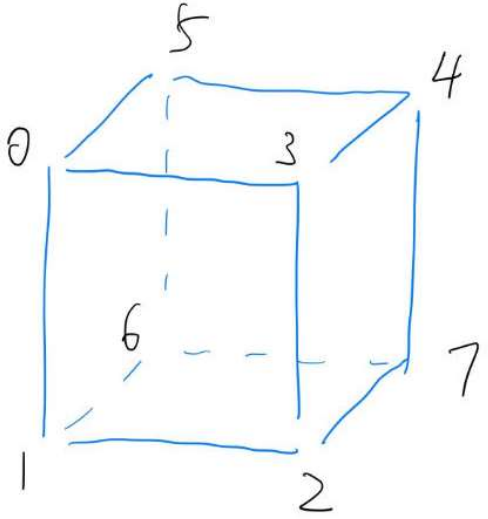
把pipeline按网格密度生成两个**粗糙**和**细致**的两个网络。

- 使用粗糙网络使用较少的采样点进行采样
- 根据粗糙采样的结果，使用细致网络采样：

在密度$\sigma$越大的地方使用越多的采样点进行采样。

# 对NeRF加速： voxel grid

先前的工作使用grid来存储feature vector ， 通过输入向量$x \in \mathbb{R}^3$进行查询和插值，再丢给神经网络。




voxel：体素，一个个同样大小的立方体，每个顶点有一个index：

$[\sigma_0, \sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_6, \sigma_7]$，同时使用一维数组存储对应的feature vector

反向传播时再更新每个顶点的feature vector

大大缩小神经网络的大小，但是引入了grid同样占用大量内存

# Instant-NGP

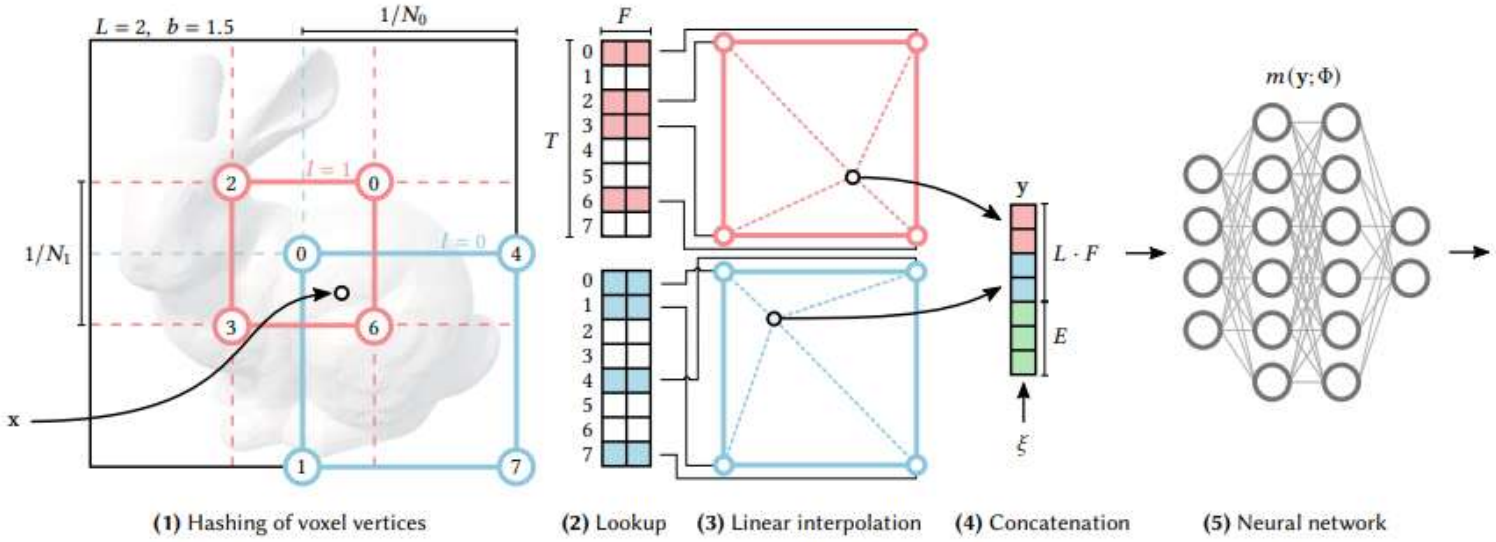
本质上是使用一种可编码的多分辨率grid哈希编码结构替换NeRF中使用的三角函数频率编码。

可编码：feature vector可以被反向传播修改

多分辨率：使用不同密度的网格分别表示feature vector

哈希:不再是存储每个grid顶点的feature vector，而是设置一个大小为T的散列表，不管理哈希冲突，由神经网络自行管理（通过控制路径上每个点的权值）

由于许多grid都是不重要的，通过散列表可以控制grid占用的大小（用T控制），同时使用多分辨率的grid。



# Instant-NGP加速效果

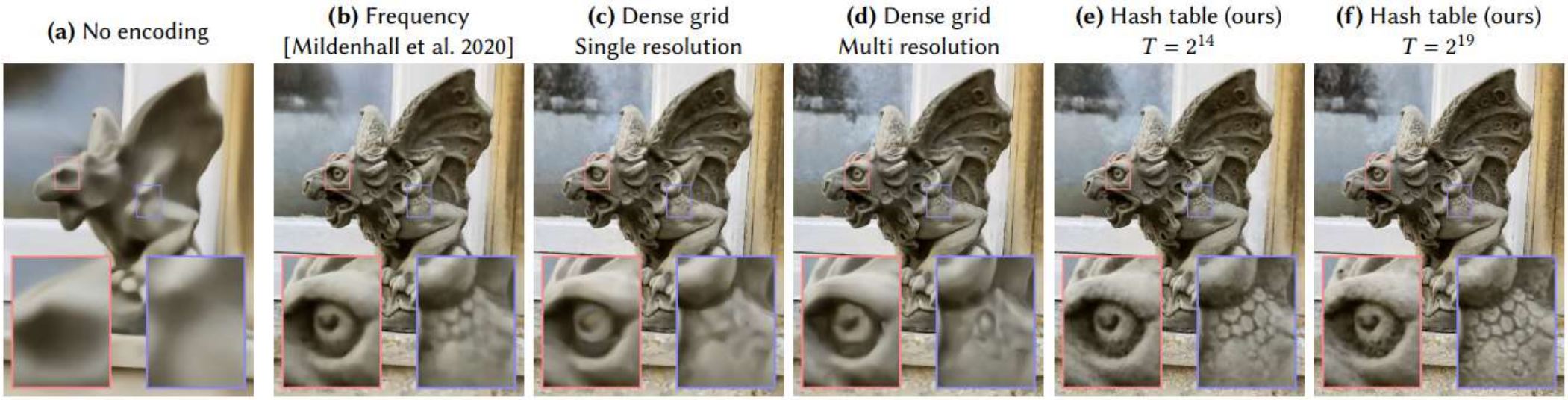



Fig. 2. A demonstration of the reconstruction quality of different encodings and parametric data structures for storing trainable feature embeddings. Each configuration was trained for 11 000 steps using our fast NeRF implementation (Section 5.4), varying only the input encoding and MLP size. The number of trainable parameters (MLP weights + encoding parameters), training time and reconstruction accuracy (PSNR) are shown below each image. Our encoding **(e)** with a similar total number of trainable parameters as the frequency encoding configuration **(b)** trains over 8× faster, due to the sparsity of updates to the parameters and smaller MLP. Increasing the number of parameters **(f)** further improves reconstruction accuracy without significantly increasing training time.

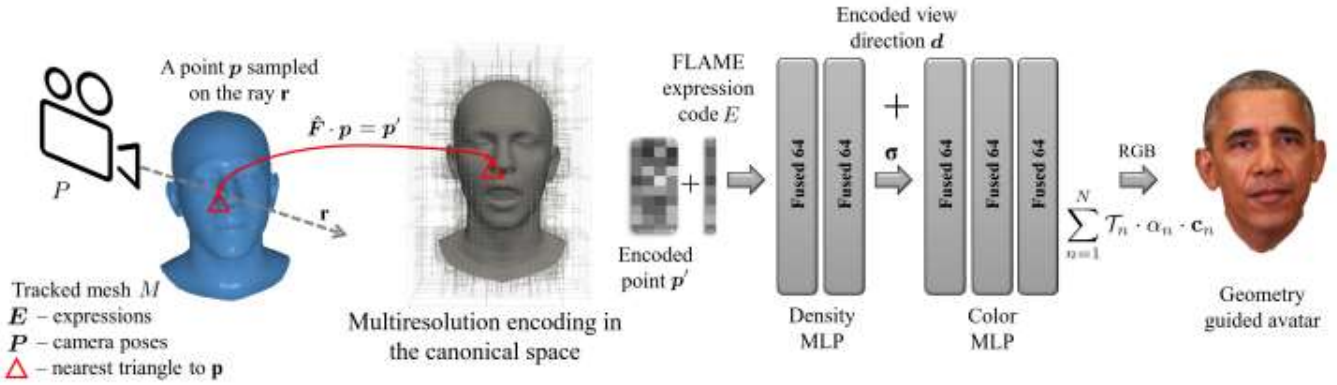
# Instant-NGP 演示

./instant-ngp data/nerf/fox

# NeRF在人体重建的应用

**Instant Volumetric Head Avatars**

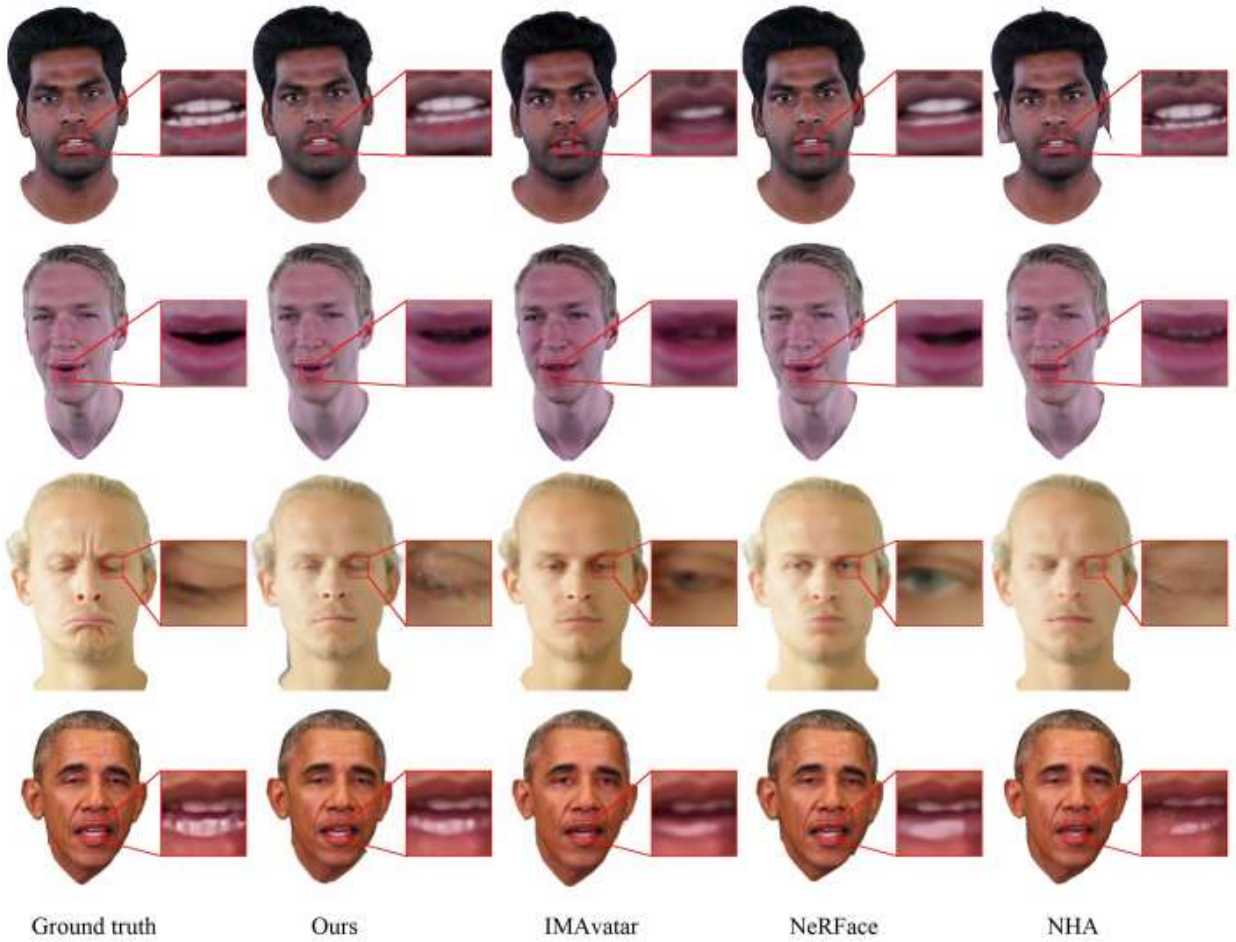
Wojciech Zielonka Timo Bolkart Justus Thies
Max Planck Institute for Intelligent Systems, Tübingen, Germany
{wojciech.zielonka, timo.bolkart, justus.thies}@tuebingen.mpg.de




流程:

- 给定一段视频,先追踪视频中每一帧的FLAME人脸表情和姿态。
- 对于某一帧，在当前空间中的相机射线上进行采样，对于采样点$p$:
  - 找到离该点最近的人脸网格上的三角形，并根据标准空间中对应的三角形计算两个三角形间的变形矩阵$\hat{F}$
  - 根据变形矩阵，将点$p$变形到标准空间点$p'$
  - 根据multi-resolution hashing表得到点$p'$处的特征向量，并和人脸表情系数$E_i$、编码的视线方向d一起传入MLP得到点$p$对应的密度$\sigma$和颜色$c$

# 重建效果



谢谢