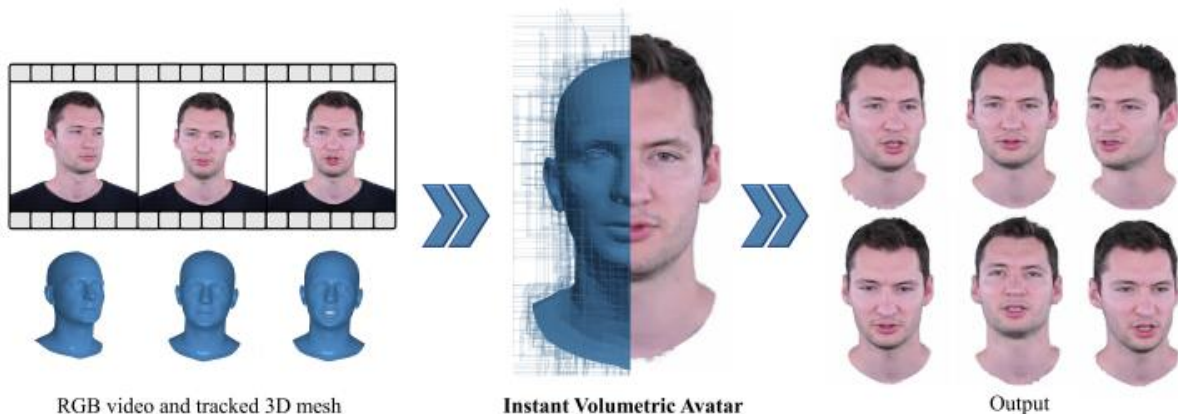


Instant Volumetric Head Avatars

Wojciech Zielonka, Timo Bolkart, Justus Thies
Max Planck Institute for Intelligent Systems, Tübingen, Germany
CVPR 2023

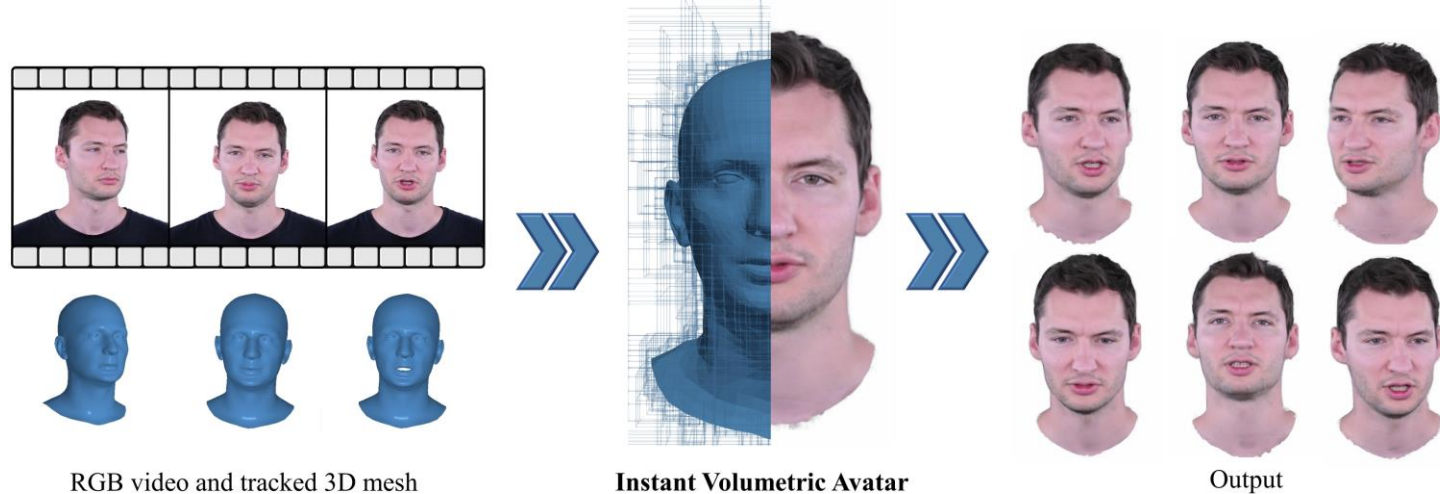
汇报人：余斐然

2023/11/07



<https://zielon.github.io/insta/>

概要



本工作可以从视频流构建关于人头的NeRF动画，并且支持新的人体动画生成。

主要贡献：

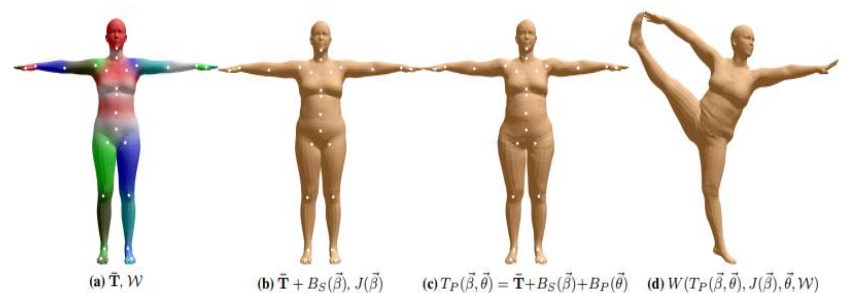
- 极大提速：由原本的一两天多张3090卡的训练时间-> 单卡3090在10分钟；可以实时播放NeRF动画；
- 使用新的映射方法：提高了诸如牙齿、头发等部位的表现细节

目录

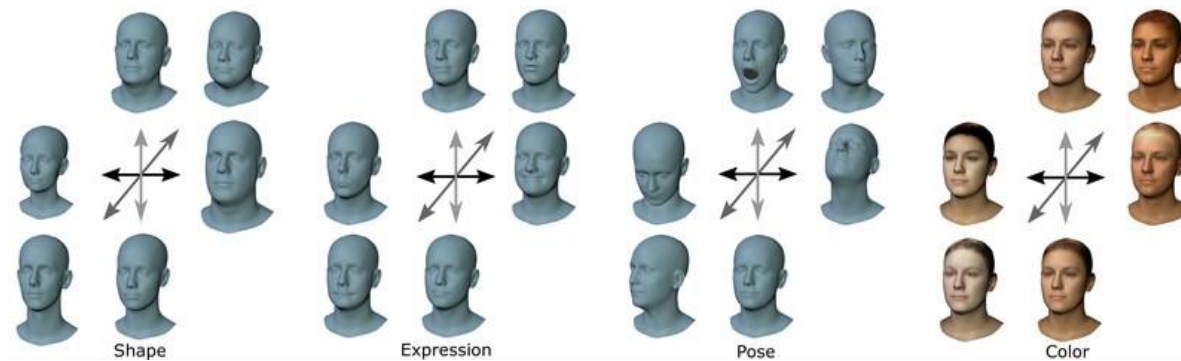
- 背景
- Nearest triangle search
- Deformed Space 到 Canonical Space 的映射方法
- 构建NeRF

背景：什么是人体/服装参数化模型？

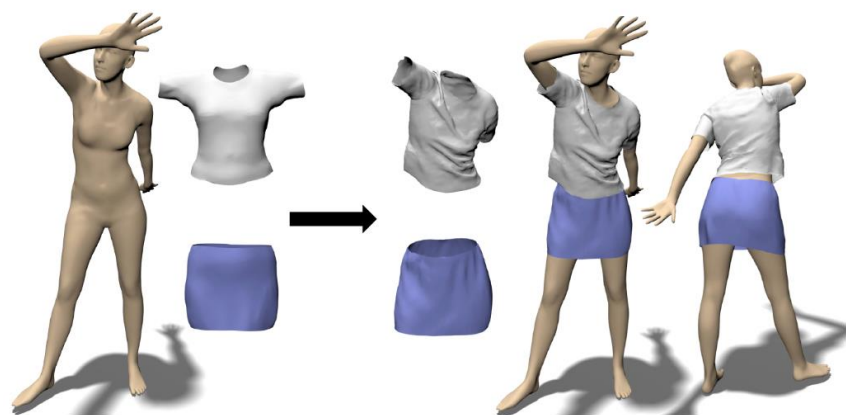
用一组参数（体型，姿态）表达人体/服装的三维模型，支持灵活的**姿态动画**



SMPL 人体参数化模型(CVPR 2015)



FLAME人头参数化模型(SIGGRAPH Asia 2017)

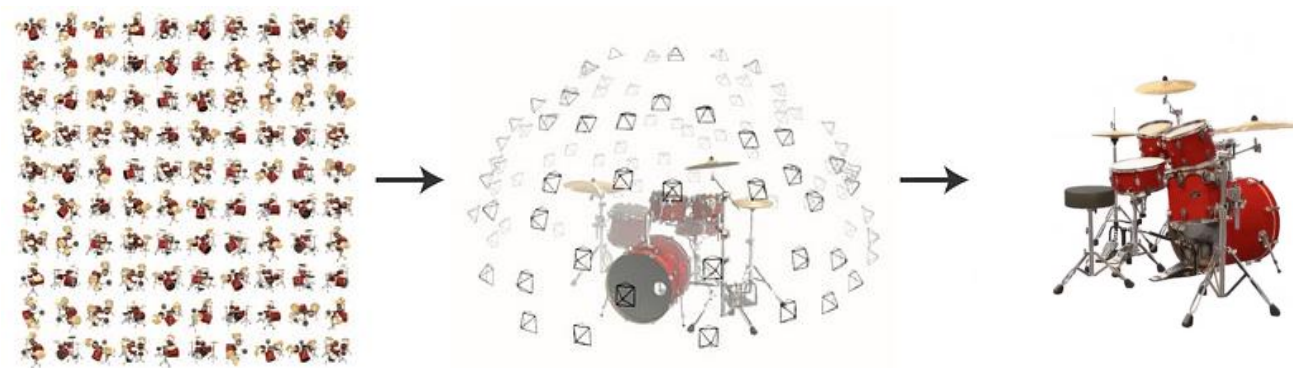


TailorNet 服装参数化模型(CVPR 2020)

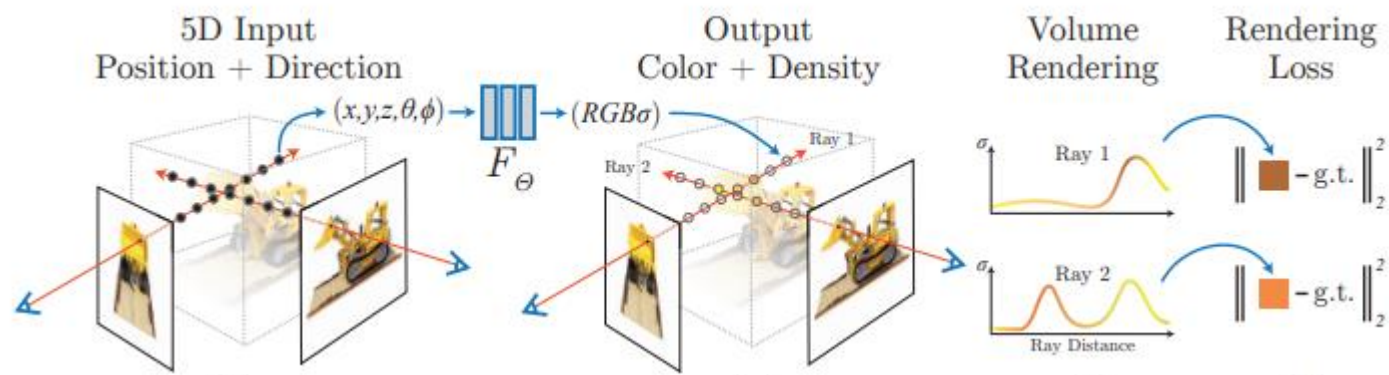


SMPL-X(CVPR 2019)
增加了手部 脚部 脸部的建模

背景：什么是NeRF?



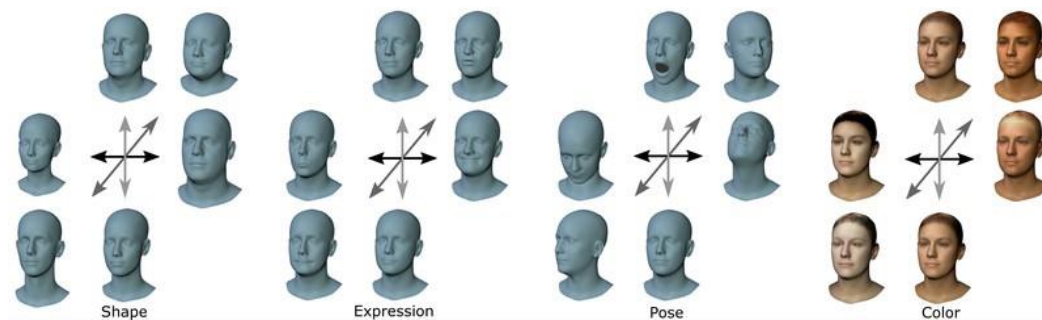
学习对同个**静态**场景的不同角度拍摄的图片，构建神经辐射向量场(Neural Radiance Fields),从而能够生成**新角度**下的场景渲染结果。



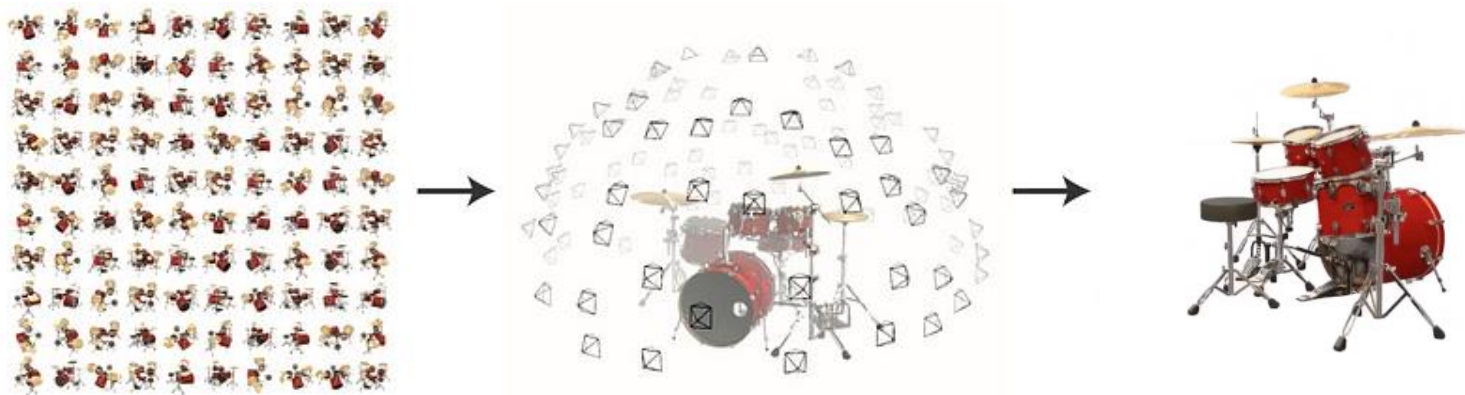
NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis (CVPR 2020)

背景

- 人体参数化模型支持灵活的运动，但是无法表达几何细节



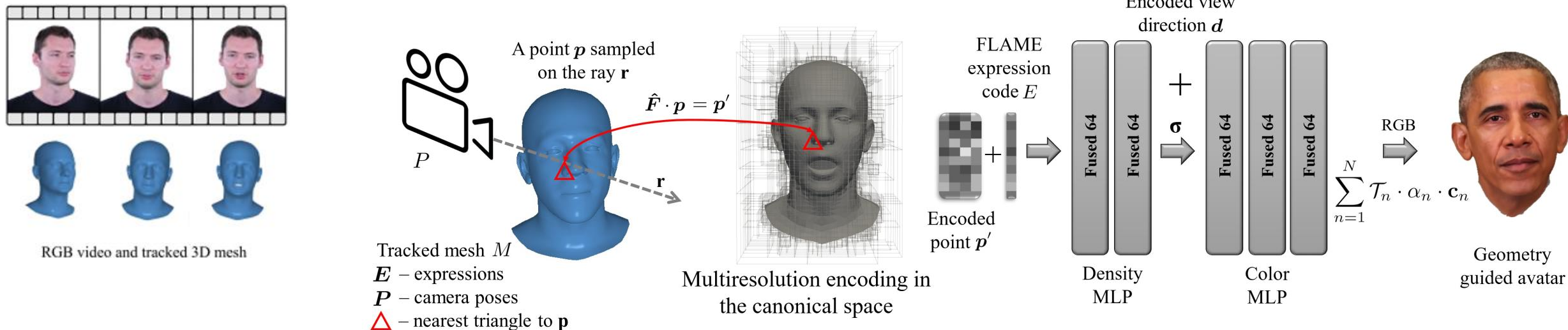
- NeRF重建出高质量的静态场景，但是无法重建运动的物体



那我们能不能把它们的优势结合起来？

方法 OverView

Instant Volumetric Head Avatars

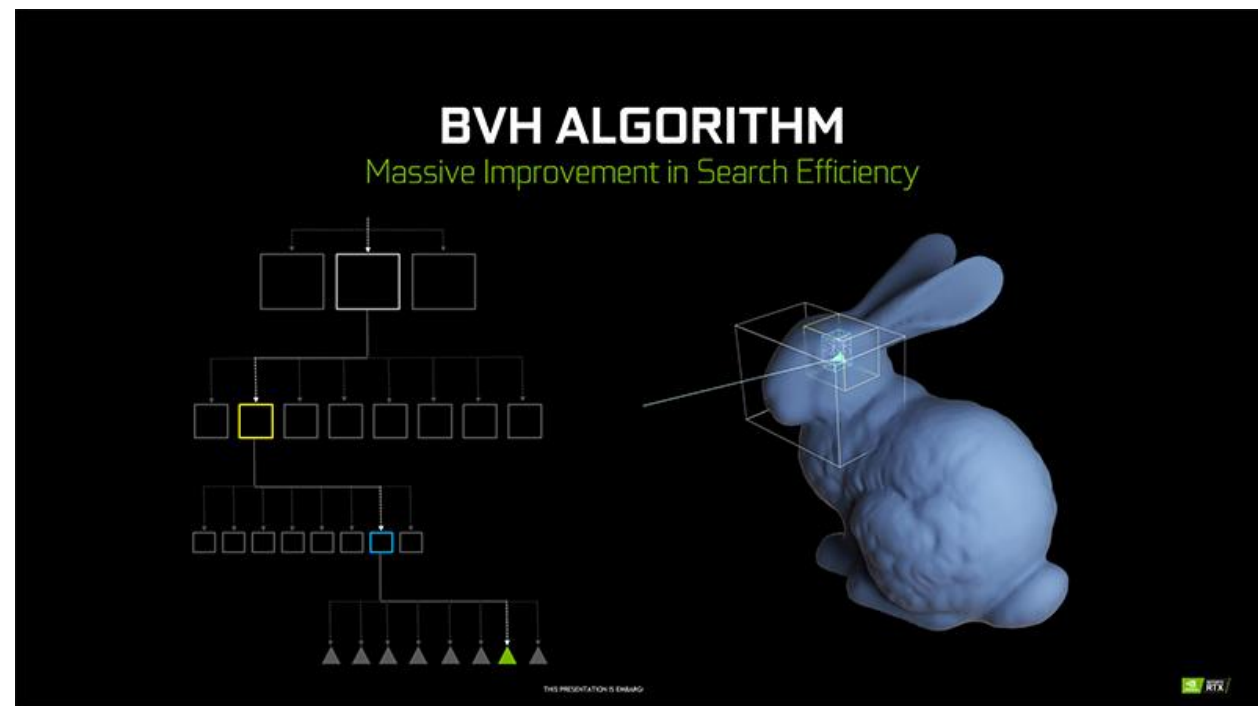
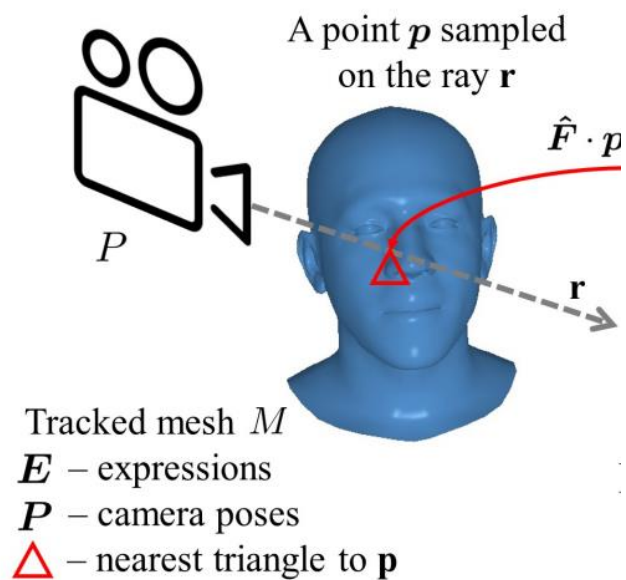


流程:

- 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 r 上进行采样,对于采样点 p :
 1. 找到离该点最近的人脸网格上的三角形
 2. 计算到标准空间中对应的三角形的变形矩阵 \hat{F} ,将点 p 变换到标准空间点 p'
 3. 使用NeRF方法计算

方法： 1. 查找点最接近的三角形

是查找点最接近的三角形，而不是求射线相交的三角形！

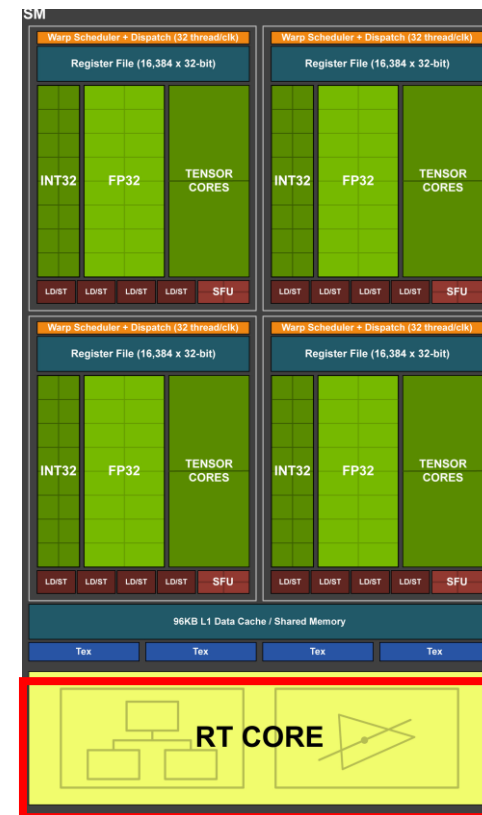


该工作使用GPU加速（CUDA）的基于BVH的寻找最接近的三角形，极大加速了这个十分耗时的步骤

方法： 1.查找点最接近的三角形

该工作只使用了CUDA进行加速，未使用光线追踪加速RT核心进行加速，可能有优化空间？

RT核心寻找射线最接近的三角形是一个可以探索的问题？



GTC 2020: Working with Tensor Cores and RTCore for compute

<https://developer.nvidia.com/gtc/2020/video/cwe21106>

Hardware Accelerated Ray Tracing
硬件加速 光线追踪的
原理、设计及应用

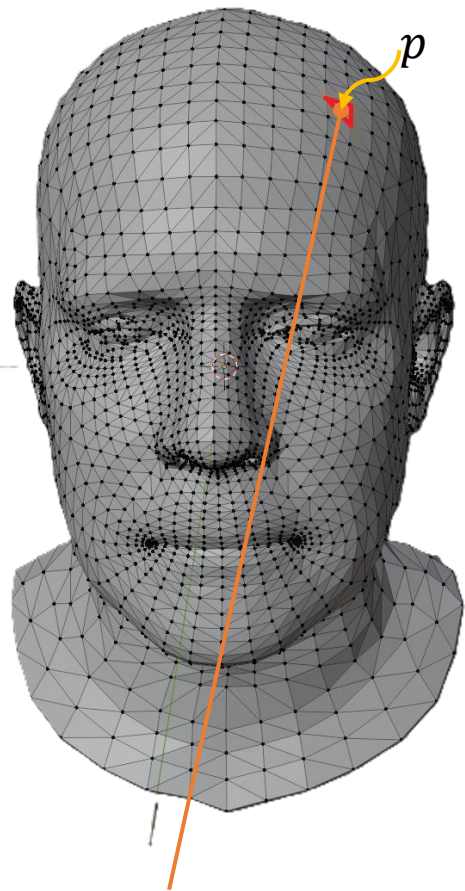
汇报人：余斐然

2023/4/17



方法： 2. 简单映射到标准空间

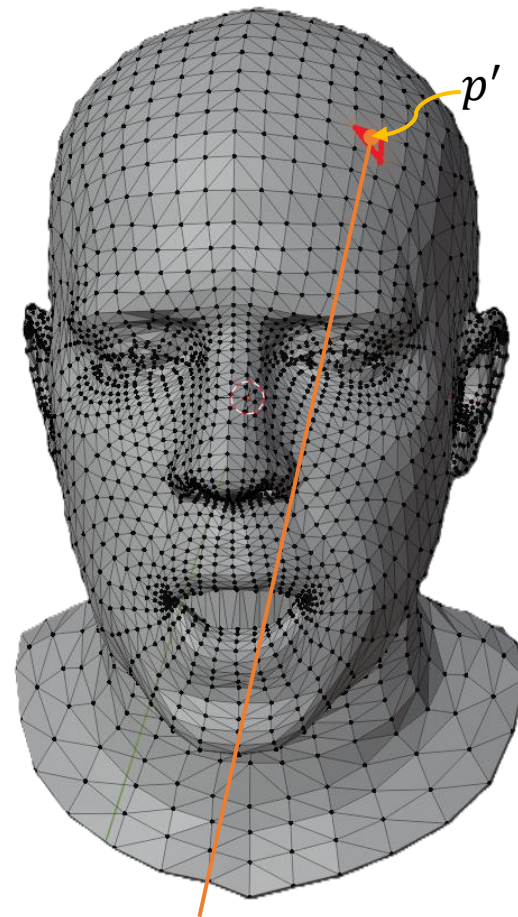
首先只考虑在FLAME网格上的点 p 从Deformed Space 到 Canonical Space 的映射



Deformed Space

利用FLAME参数化模型对应关系
得到一个转换矩阵 \tilde{F}

$$p' = \tilde{F} \cdot p$$



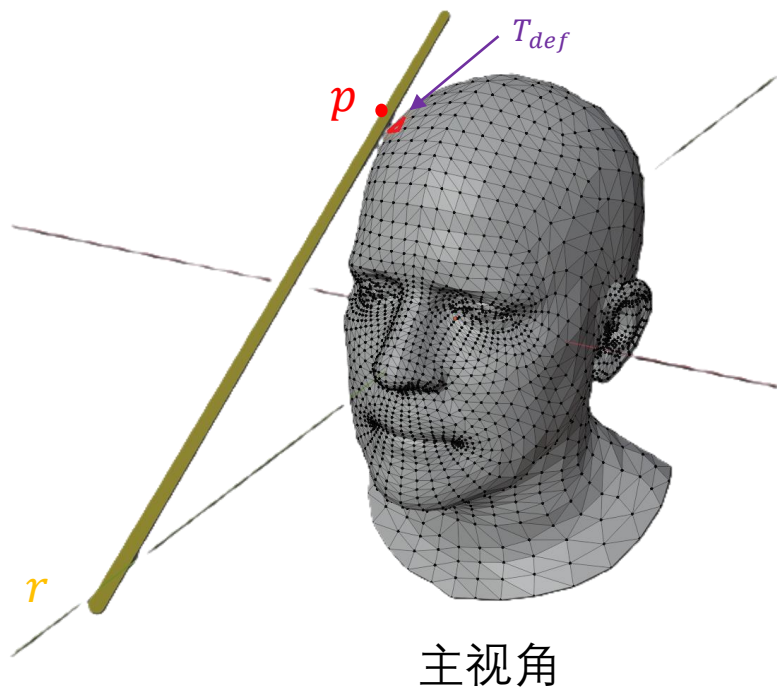
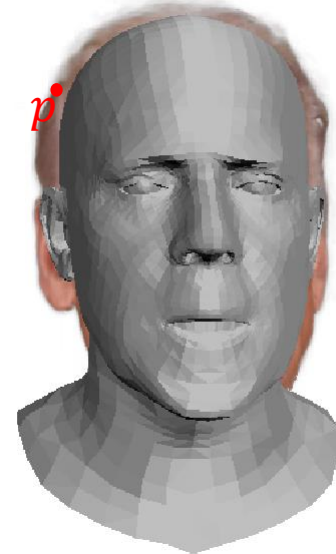
Canonical Space

只能处理点在FLAME网格上的情况!

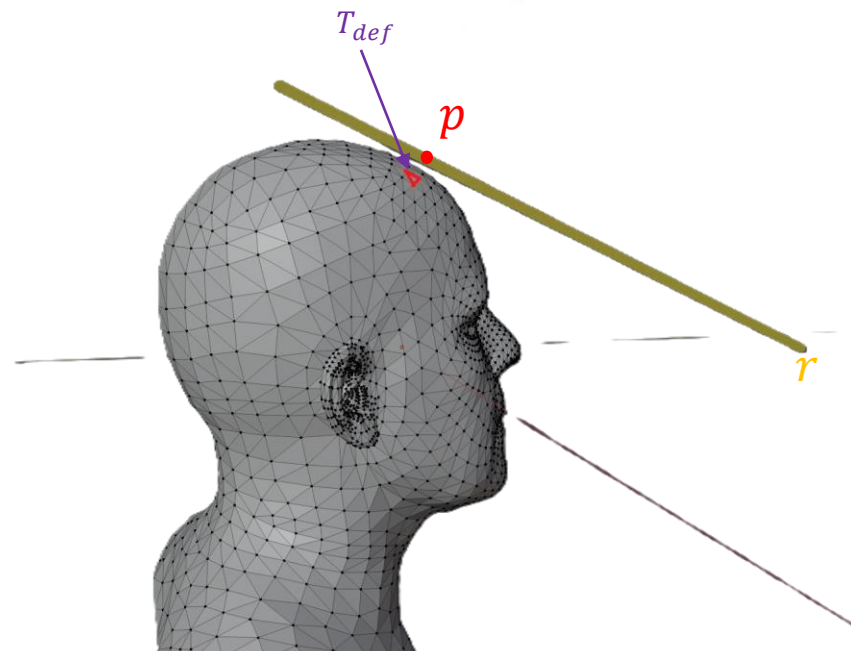
方法： 2.进行映射到标准空间

但是有不在FLAME网格上的我们想要采样的点，该如何映射？

(比如FLAME模型中没有的牙齿、头发等)



主视角

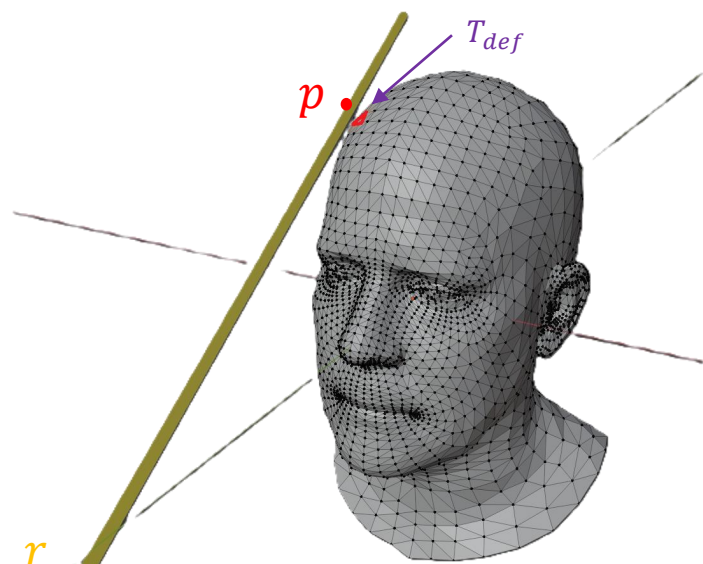


侧视角

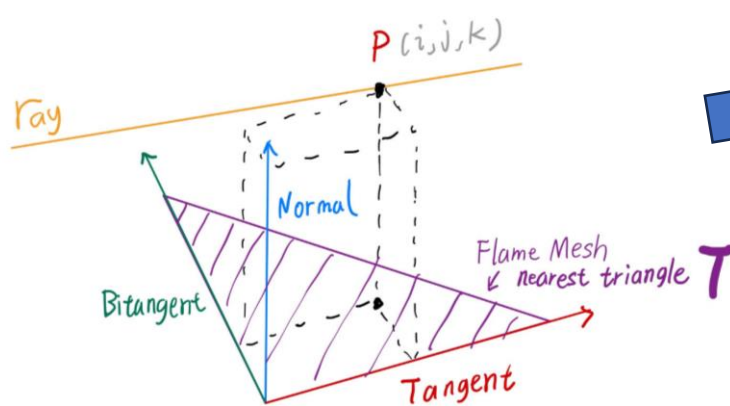
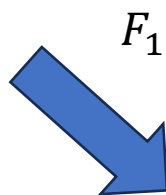
光线 r 产生的采样点 p 不在FLAME网格上,无法用前面的方法转换到标准空间!

方法：2.如何映射到标准空间？

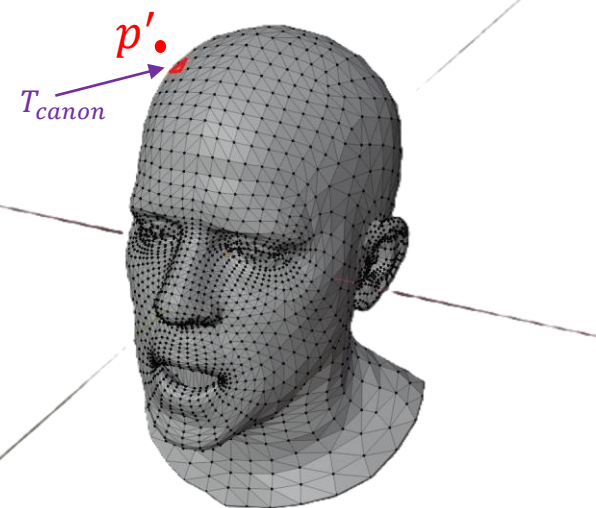
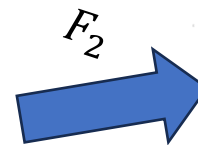
该工作利用切线空间来保持点 p 到三角形 T 的偏移



Deformed Space



Tangent Space



Canonical Space

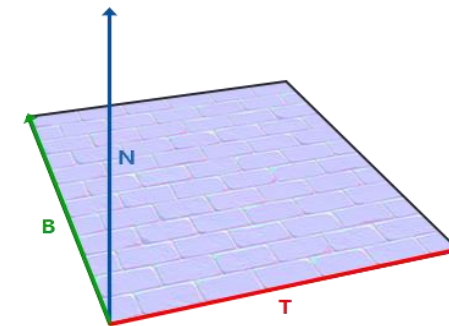
切向空间

切向空间是一个相对于三角形的本地参考坐标系。

法线永远指着正z方向。

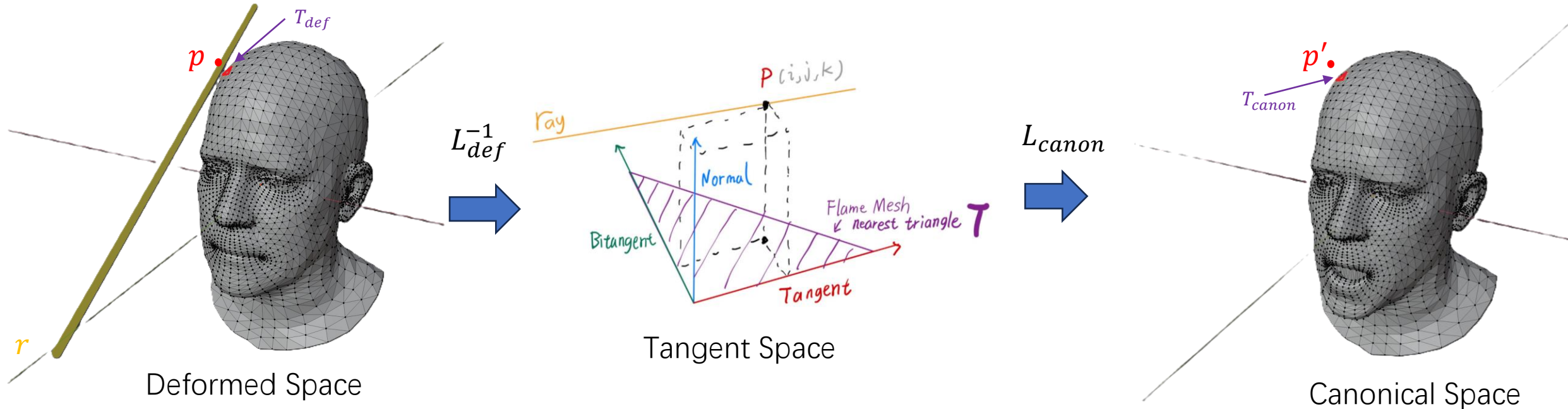
法线相对于单个三角形的本地参考框架。

经常用于Normal Mapping



-Learn OpenGL: Normal Mapping

方法：2.如何映射到标准空间？



$$L_{def} = \begin{bmatrix} R_{def} & t_{def} \\ 0^T & 1 \end{bmatrix}$$

$$L_{canon} = \begin{bmatrix} R_{canon} & t_{canon} \\ 0^T & 1 \end{bmatrix}$$

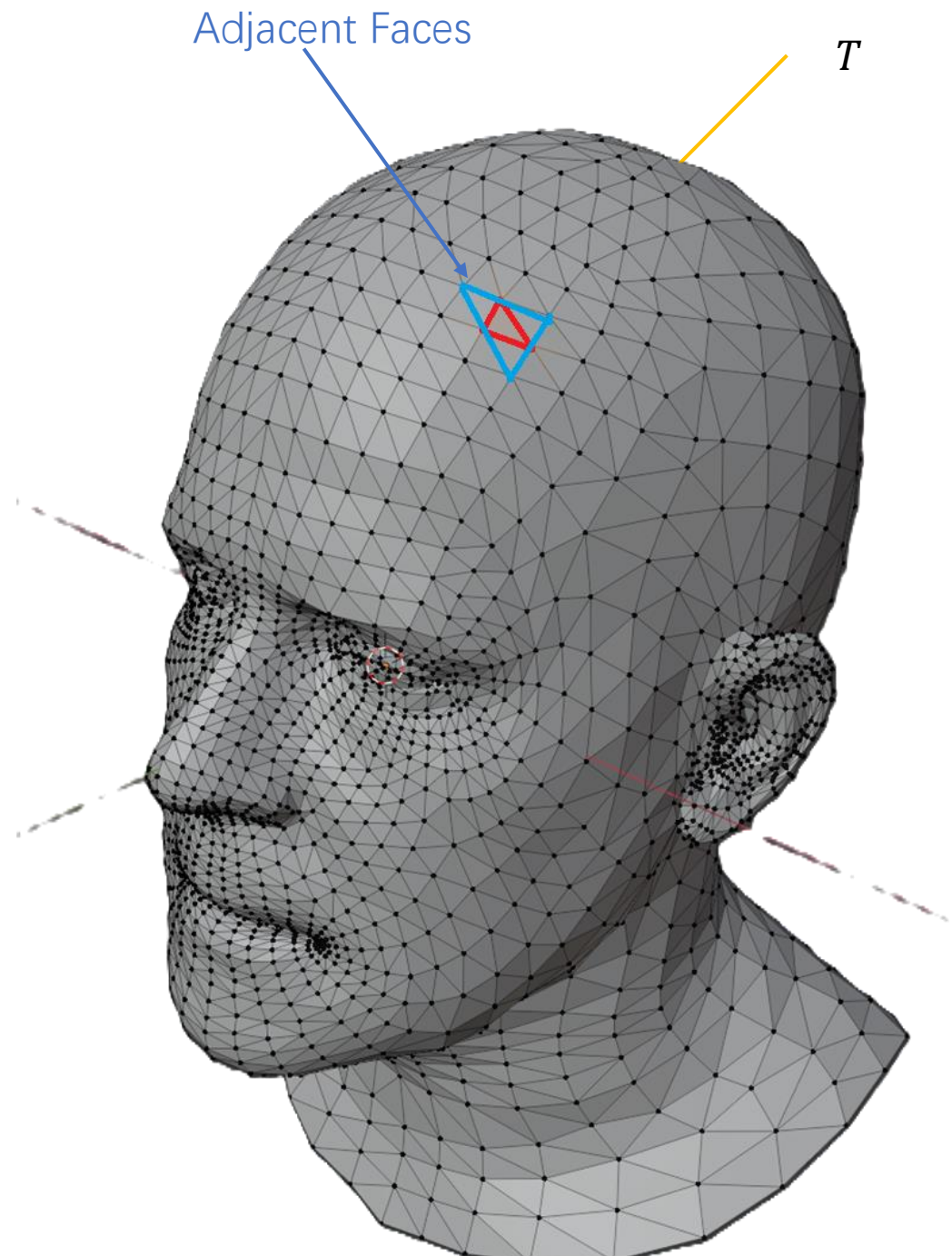
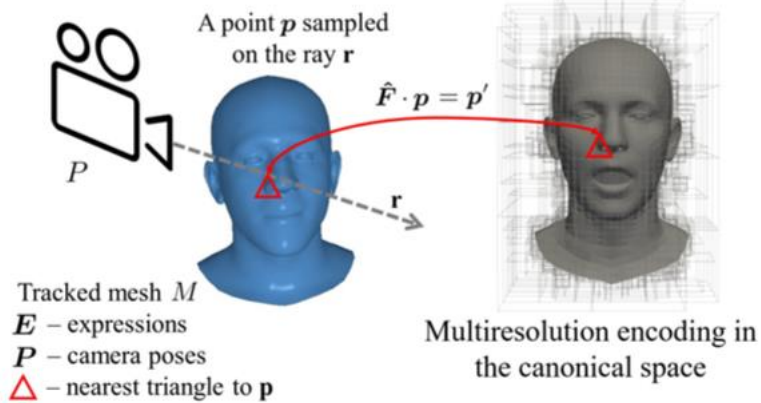
(2)

得到最终的变换矩阵F: $F = L_{canon} \cdot \Lambda \cdot L_{def}^{-1}$,

$$\Lambda = \begin{bmatrix} \lambda I & 0 \\ 0^T & 1 \end{bmatrix}$$

(3)

(变换过程中三角形可能会有缩放，所以再乘以一个缩放系数 $\lambda = \frac{a_{def}}{a_{canon}}$, a是三角形面积)



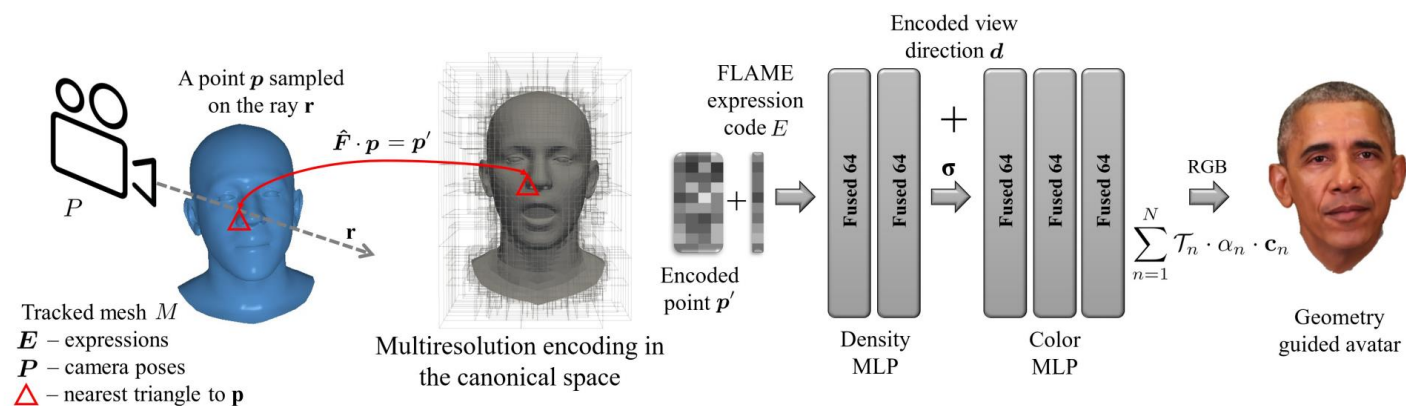
为了保证变换的连续性，对利用周围的三角形变换矩阵进行指数平均求得最终的变换矩阵 \hat{F} ：

$$\hat{F} = \frac{1}{\sum_{f \in A} \omega_f} \cdot \sum_{f \in A} \omega_f F_f \quad (4)$$

其中 $\omega_f = \exp(-\beta \|c_f - p\|_2)$, $\beta = 4$, A 是关于当前三角形 T 的adjacent faces集合（包含 T 本身，此时 $\beta = 1$ ）

$$\hat{F} \cdot p = p'$$

方法： 3.使用NeRF方法进行重建



$$\hat{C} = \int_0^D \mathcal{T}(t) \cdot \sigma(t) \cdot \mathbf{c}(t) dt + \mathcal{T}(D) \cdot \mathbf{c}_{bg} \quad (1)$$

其中 $\mathcal{T}(t_n) = \exp(-\int_0^{t_n} \sigma(t) dt)$

$\mathbf{c}(t)$ 是在点 p_t 的辐射场，点 p_t 是在标准空间中的向量场

使用Instant NGP对构建NeRF进行加速

Loss Function

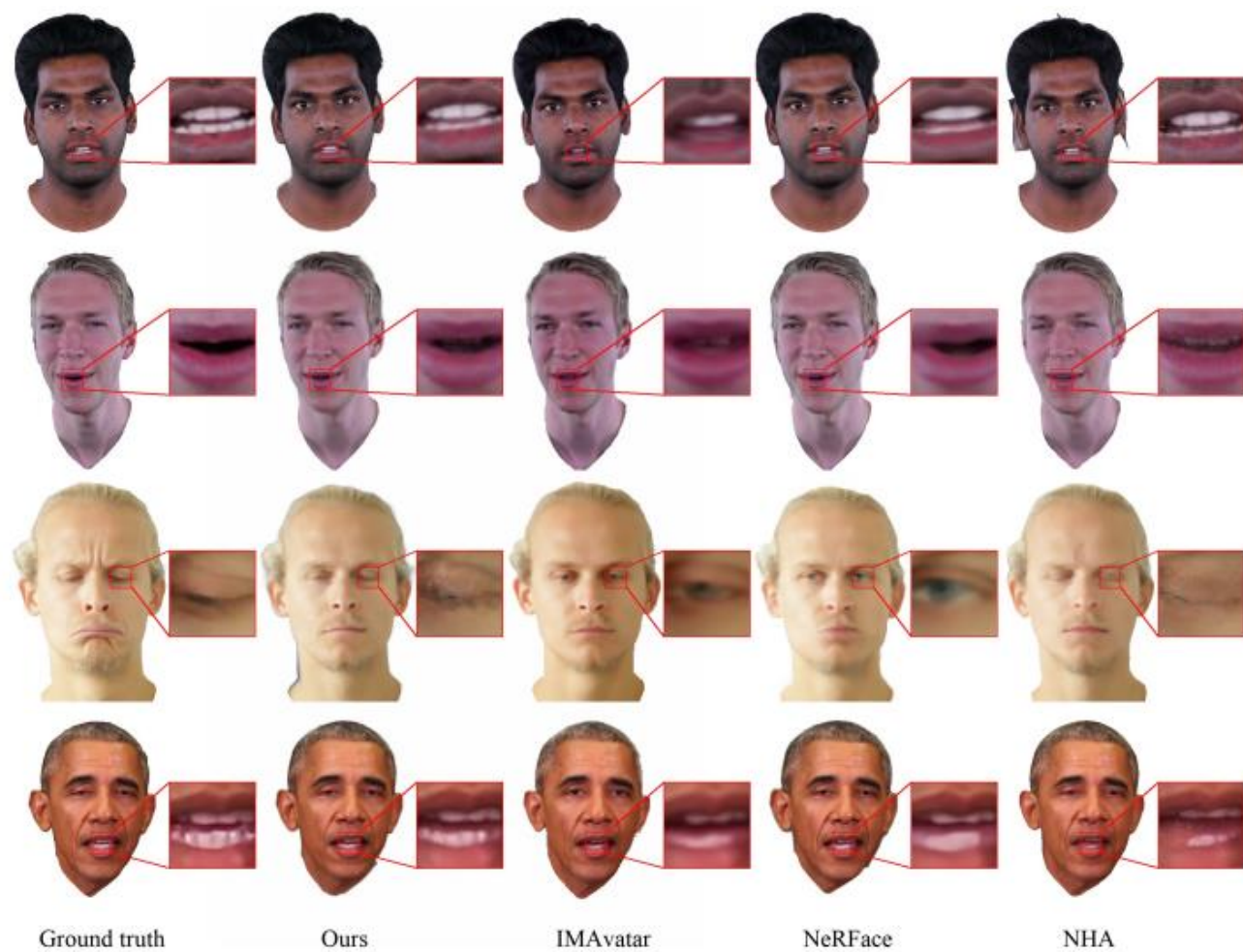
$$\mathcal{L}_{color} = \begin{cases} \frac{1}{2}(\mathbf{C} - \hat{\mathbf{C}})^2 & \text{if } |(\mathbf{C} - \hat{\mathbf{C}})| < \rho \\ \rho((\mathbf{C} - \hat{\mathbf{C}}) - \frac{1}{2}\rho) & \text{otherwise} \end{cases} \quad (6)$$

$$\mathcal{L}_{geom} = \sum_{\mathbf{r}} |\mathbb{1}_{face}\{(z(\mathbf{r}) - \hat{z}(\mathbf{r}))\}|, \quad (7)$$

$$\mathcal{L} = \sum_{\mathbf{r}} \lambda_{color}(\mathbf{r})\mathcal{L}_{color}(\mathbf{r}) + \lambda_{geom}\mathcal{L}_{geom}(\mathbf{r}), \quad (8)$$

*细节请参考论文

结果



利用使用切向空间的映射方法，该工作对于牙齿等不在FLAME网格中的部位重建效果比以往好

演示 INSTA

谢谢