单目视频的人脸语义神经辐射场重构

谢锦东

摘要

神经辐射场（NeRF）算法在近三年来受到了极大的关注并取得了明显的进展，该算法能够通过获取单场景的不同视角图片，从而构建整个场景的神经辐射场，通过神经网络隐式表达整个三维场景。基于双线性的三维形变模型（3DMM）表征的头部是由一组解耦的基底组成，并且可以由低维表达系数驱动。本文提出了一种结合三维可形变模型和神经辐射场表示的头部语义模型。由于神经辐射场强大的表示能力，所构建的模型可以表示复杂的面部属性，包括头发、着装等，这些属性是传统 3DMM 方法所无法表示的。为了构建个性化的语义面部模型，本文将基底转换为几个多尺度的体素网格，以单目 RGB 短视频作为输入，使用哈希编码的方式，可以在 10 到 20 分钟内构建视频主体人物语义面部的 NeRF 模型，并且可以在给定的表情系数和视图方向下在数十毫秒内渲染出逼真的人头图像。本文所提出的这种新颖的表示能够应用于人脸重演和表情编辑等多种任务。[1]

关键词：神经辐射场；三维可形变模型；表情编辑

1 引言

三维人脸表示是计算机视觉和计算机图形学中的一个重要的研究课题，在虚拟现实 (VR, Virtual Reality) 和增强现实 (AR, Augmented Reality) 中具有大量的应用。如何从单目摄像头所拍摄的视频中重建出一个逼真的三维人脸是一个重要并具有挑战性的问题。假设人脸能够表征为一个低维空间的向量数据，研究者提出了一些参数化人脸的方法例如 BlendShape，这种 BlendShape 人脸模型通过人脸表情基底的线性组合来构成一个新的人脸，于是每一个真实人脸都能够在参数空间中找到能够表达对应表情的系数，并对人脸进行重构。同时，这种具有语义含义的 BlendShape 模型能够通过手工编辑对重构的人脸进行自由的语义控制。

广泛被使用的 BlendShape 模型例如 FaceWareHouse[2]，使用三维可变形模型[3](3DMM, 3D Morphable Model) 的方式，对具有不同表情的多个对象进行建模，但是对人脸的几何形状结构以及其纹理难以进行精确重建。传统基于 mesh 的 3DMM 方法无法对人的头发、牙齿等非面部部分进行建模，并且真实人脸的面部表情受到多种因素例如年龄和肌肉的影响，而这些因素是难以别一个预定义的 BlendShape 所定义的。

近年来，随着计算机视觉和计算机图形学的广泛研究，研究者发现基于神经辐射场 (NeRF, Neural Radian Field) 的方法能够生成逼真的人脸图像，并具有三维一致性。最近，基于 NeRF 的说话人脸的重建被大量研究，如[4-6]等方法通过输入一段人脸说话视频，通过神经网络隐式对人脸进行建模，构造单人的神经辐射场，并渲染出逼真的生成人脸，同时该模型还能学习到许多人脸细节。但是这些方法需要大量的时间来训练神经辐射场，同时缺乏对人脸表情的精细控制，这是因为他们直接将表情系数通过傅里叶位置编码并采用拼接的方式输入到多层感知机 (MLP, Multilayers Perceptron) 中, 而这种方法对 MLP 的收敛性并不友好，并且拼接的方式并没有包含任何的组合法则以发掘局部特征和整体特

征之间的关系 (在 NeRF 中，指代位置信息和表情条件)，因此，这种方式对 MLP 来说需要花费大量的时间去学习如何使用表情条件来预测颜色和辐射场的密度。

近年来，研究者不断探索神经辐射场中的局部特征以提升模型的性能和效率。原始的 NeRF 算法是使用傅里叶位置编码作为局部特征输入到 NeRF 中，这种做法网络需要大量的时间才能够收敛，一些工作尝试去设计不同类型的局部特征来改进 NeRF，如基于体素场的方法如 DVGO[7]等，而在这些方法中，InstantNGP[8]的方法展现出其在训练时间和渲染质量上的优越性能，通过高度压缩的多尺度哈希表来存储三维空间中的局部特征，不同尺度的特征能够同时一起训练。通过这种高性能的光线投射算法，静态的神经辐射场场景训练时间不超过 1 分钟，并能够在毫秒级别渲染出一帧图像。

于是这篇文章主要参考了 InstantNGP 的方式存储空间的局部特征，不同于 IntantNGP，对于单目说话人脸视频，这是动态的场景而非静态，人的头部会在不同的帧中呈现不同的姿态和表情。为了精细控制表情，作者又借用了 BlendShape 模型以量化人脸表情。通过人脸表情和多尺度哈希表的线性组合来操纵人脸变形。该算法将自动从视频人脸图像中自动学习人脸的表情基底，并能够通过基底对人脸表情进行自由的编辑，人脸操纵示意图如图 1所示。本文的方法在基于 NeRF 的人脸编辑算法中具有启发性的意义，但是由于本文的训练代码并未开源，使得其他大量的工作无法与本篇文章进行性能比较。本次复现选取这篇文章，旨在通过自己的复现为开源社区贡献自己的一份力量。

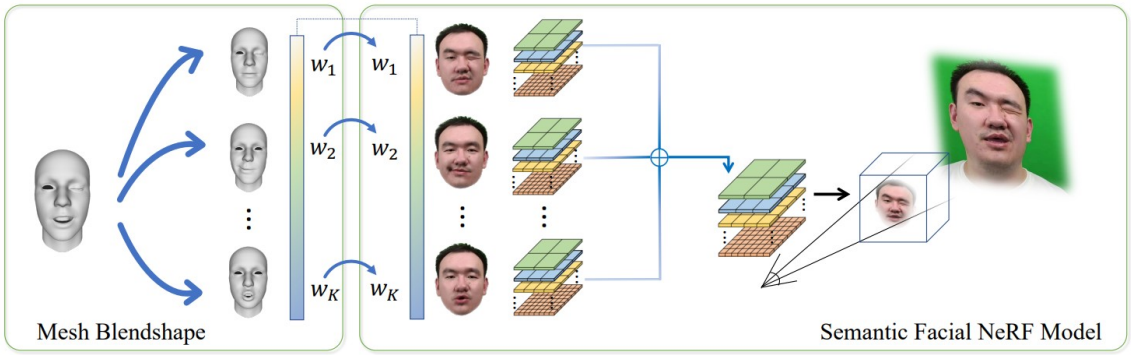

图 1: 方法示意图

2 相关工作

2.1 人脸参数化模型

人脸参数化模型是基于人脸能够表征为低维空间中的线性嵌入这一假设，通过人脸身份、人脸表情以及人脸纹理等几个参数来将三维人脸数据化。在 1999 年，Blanz[3]等人提出了 3DMM 模型，通过收集 200 个人的三维顶点数据，通过主成分分析算法获取三维人脸的主成分向量，然后通过顶点对齐的方式来生成各种不同形状的人脸，于是就可以通过一组系数，通过线性组合来生成各种各样的人脸形状。后来 BFM(Basel Face Model)[9]模型的开源使得基于主成分分析的 3DMM 模型被大量使用于各种任务中。但原始的 3DMM 模型仅考虑人脸的顶点变化即形状变化而没有考虑其不同的表情。为了提升人脸的表达能力，一些研究者尝试将此拓展为多线性模型[10]，FaceWareHouse[2]同样采取了多线性模型表示，通过采集不同人的多种表情，构造一个表情张量来作为线性组合的基底。本文参考 FaceWareHouse 对带语义的表情系数的处理，通过在图像序列自动学习 NeRF 基底，通过线性组合的方式来重构不同表情的三维人脸。

2.2 神经辐射场算法

神经辐射场是近几年来被广泛研究的一种视角生成算法，其使用神经网络隐式表征一个三维场景，能够渲染出不同视角的三维一致的物体，受到广泛的关注。NeRF 使用一个 MLP 以及使用体积渲染的方法以生成场景新的视角。近来 NeRF 同样在人脸建模上展现出其强大的表示能力。许多工作[11]采用 NeRF 算法来动态的表征人脸场景，并且能够生成高质量的三维一致的结果。但都没有对独立个体进行建模。而 ADNeRF[4]和 NerFace[5]等模式是针对单一个体进行建模的，并且能通过声音或者表情来高质量控制人脸动作，但是这些方法都需要大量的时间来训练，并且有时还会忽视人脸的高频细节。本文同样采用神经辐射场算法，并参考 InstantNGP 的方式，使用多尺度哈希表来存储位置特征，以快速高效对进行人脸表情的控制。

3 本文方法

3.1 本文方法概述

本文的人脸生成方法如图 2所示。通过一组人脸的 BlendShape 表情系数，将这些系数与表征不同人脸的多尺度哈希表格做线性组合得到生成人脸对应的哈希表，在后续 Nerf 体积渲染的过程中，光线采样点的对应的位置特征由各个尺度下的领域八个点的特征进行三线性插值得到，从而得到各个尺度下的特征通过拼接输入到 MLP 中，并同时将视角信息输入到网络中，预测采样点的颜色和密度，随后通过体积渲染的方式渲染出对应表情的人脸图像。

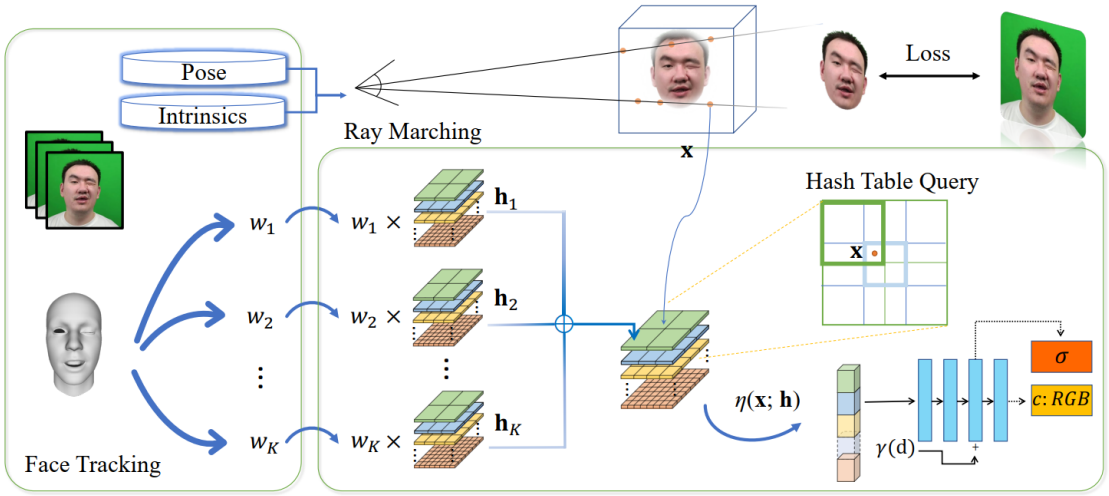

图 2: 算法框架图

值得注意的是，与 FaceWareHouse[2]所提出的 BlendShape 相同，本文采用的 BlendShape 模型中的每一个基都有其特定的语义，如左眼睛闭上和嘴角上翘等，这些都使得使用者能够使用低维的向量来生成想要的人脸图像。

3.2 基于 NeRF 的人脸线性表示

本文的算法能够在图像序列中自动学习对应人脸的特定表情的 NeRF 基底，记 C 为本文的模型在渲染过程中应用到的相机参数，R 为本文提出的模型，w 为表情系数控制人脸动作，则一帧通过本文的模型渲染出的图像可以由式子 1表示：

$$I = R_\theta(C, \boldsymbol{h_0} + \boldsymbol{H}\boldsymbol{w}), \tag{1}$$

其中 $\boldsymbol{h_0} \in \mathbb{R}^{\boldsymbol{L}\times\boldsymbol{T}\times\boldsymbol{F}}$ 是使用 BlendShape 表示下的多尺度平均哈希表格，L 是哈希表的层数，T 是哈希表的大小，F 是哈希表中存储的每一个特征的维度，$\boldsymbol{H}=\{\boldsymbol{h_1},\boldsymbol{h_2},\cdots,\boldsymbol{h_K}\},\boldsymbol{h_i}\in\mathbb{R}^{\boldsymbol{L}\times\boldsymbol{T}\times\boldsymbol{F}}$ 为 $\boldsymbol{K}$ 个哈希表,$\boldsymbol{w}=\{w_1,w_2,\cdots,w_K\}\in\mathbb{R}^K$ 为对应的 K 个表情系数。通过多个哈希表的线性组合可以得到最终用于 NGP 查询特征用的哈希表 h，如式子 2所示：

$$\boldsymbol{h}=\boldsymbol{h_0}+\boldsymbol{H}\boldsymbol{w}=\boldsymbol{h_0}+\sum_{i=1}^{K}w_i\boldsymbol{h_i},\boldsymbol{h}\in\mathbb{R}^{L\times T\times F}, \tag{2}$$

模型中的 MLP 网络是一个从输入的局部位置特征到采样点的颜色和密度的函数，这里记作 g_θ，这个隐式表示的过程可以由式子 3表示：

$$g_\theta:(\eta(\boldsymbol{x},\boldsymbol{h}),\gamma(\boldsymbol{d}))\mapsto(\sigma,c), \tag{3}$$

其中 $\boldsymbol{x}\in\mathbb{R}^3$ 为光线采样过程中的采样点，$\boldsymbol{d}\in\mathbb{R}^3$ 为表示光线的单位向量，$\eta(\boldsymbol{x},\boldsymbol{h})\in\mathbb{R}^{L\dot{F}}$ 是采样点在对应的哈希表中查询得到的特征向量，$\gamma(\boldsymbol{d})$ 为角度 $\boldsymbol{d}$ 的球谐系数编码，将光线角度投影至球谐函数基底的前 16 位的系数，σ 和 c 分别表示采样点对应的密度和颜色。

最后通过体积渲染获取对应的人脸图像：

$$I(r)=\int_0^\infty p(t)c(r(t))dt, \tag{4}$$

其中

$$p(t)=\exp(-\int_0^t\sigma(r(s))ds)\sigma(r(t)), \tag{5}$$

其中 $r(t)$ 表示从相机位置发射出的光线。同样的，人脸脸部的掩膜可以通过式子 6生成：

$$M(r)=\int_0^\infty p(t)dt, \tag{6}$$

3.3 损失函数设计

为了快速渲染出高质量的人脸图像，本文采用了三种损失函数，分别为像素损失函数，掩膜损失函数以及感知损失函数。

像素损失函数是渲染出的图像和训练图像的逐像素的损失。该损失函数可以由公式 7表示：

$$L_{color}=\sum_{r\in S}\|I(t)-I_{GT}(r)\|_1. \tag{7}$$

其中 S 为光线的集合，$I(r)$ 和 $I_{GT}(r)$ 分别是光线在预测的渲染图像以及数据集中原本的真实图像中采样到的颜色。

由于本文实验处理的是人脸，需要将人脸前景和背景分离，神经辐射场能够预测出图像掩膜，通过掩膜损失函数尽可能地将图像中非人脸部分的区域置为 0。于是损失函数如公式 8所示：

$$L_{mask}=\sum_{r\in S}\|M(r)-M_{GT}(r)\|_1. \tag{8}$$

感知损失函数 (LPIPS)[12]常用于在重构任务中捕捉人脸的高频细节信息，使用卷积神经网络能够更好的对图像上下文中的细节进行捕捉，但是在神经辐射场中是对光线进行采样，在图像中随机选取坐标值而没有一个图像块进行卷积，本文参考了[13]中的处理方法，在图像中随机采样 B 个 Patch，每个 Patch 的大小为 $W\times W$，并每一个 Patch 中总共采样 $B\times W\times W$ 条光线。对于每一个 Patch 同样可以在原始图像中采样到同样位置和大小的 Patch，于是将两个 Patch 输入到 VGG 卷积神经网络中计

算两者的感知损失。于是，总的损失函数可以写为公式:

$$L_{total} = \lambda_1 L_{color} + \lambda_2 L_{mask} + \lambda_3 L_{LPIPS}, \tag{9}$$

其中 λ_i 为不同损失函数的平衡变量，为了最小化总体损失函数，原始文章中采用了一个良设计的训练策略，包含以下三个步骤：在前两个 epochs 中，将 λ_1 和 λ_2 设置为 1，λ_3 设置为 0，这是为了使用掩膜损失函数更快地让模型学习到密度场的分布，但是由于语义分割算法存在的不准确性，掩膜损失函数并不能很精确地对头发部分进行约束，所以在第二个到第七个 epoch 中，文章设置 λ_1 为 1 并将另外两个置为 0，只留下像素损失函数，而到第七个 epoch 之后，将 λ_1 和 λ_3 置为 0.1，而 λ_2 置为 0，使用感知损失函数优化模型以捕捉高频细节。与原论文不同的是，本人复现过程中在最后阶段将 λ_3 置为 0.01。

4 复现细节

4.1 预处理

本文需要对输入的视频图像进行大量的预处理，由于神经辐射场的输入需要相机的位姿，而单目视频中的相机是固定不动的，于是本文将人脸的朝向作为相机的相对位子作为神经辐射场的相机姿态参数输入。并且本文使用预训练的人脸 BiseNet[14]对人脸进行语义分割，依照 ADNerf[4,15]的处理方式使用 kdtree 算法进行图像背景提取。

而预处理的重点在于人脸的 BlendShape 的估计，而作者由于版权原因不能够提供预训练好的 BlendShape 估计网络，相关预处理代码也不能开源，于是本次复现使用了开源的 CPEM[16]算法预测输入人脸图像序列的 blendshape 系数，参考 FaceWareHouse[2]设计的 46 个 blendshape 系数，每一个系数都有其特定的语义含义。

4.2 密度网格场更新策略

在 InstantNGP 模型中，使用了密度网格场 (Density Grid field) 以加速 NeRF 模型的收敛速度，这个 Density Grid 能够避免光线在空间中无密度的位置上进行采样，从而进行了一种跳跃采样的策略，大大提升了神经辐射场收敛的速度。但是原本的 Instant-NGP 算法是静态的神经辐射场模型，对于动态的人脸表情无法使用静态的密度场进行加速，否则人脸部分将会越界超出密度场而不被渲染得到。如果采用静态的密度网格场则会出现如图 3所示的情况。

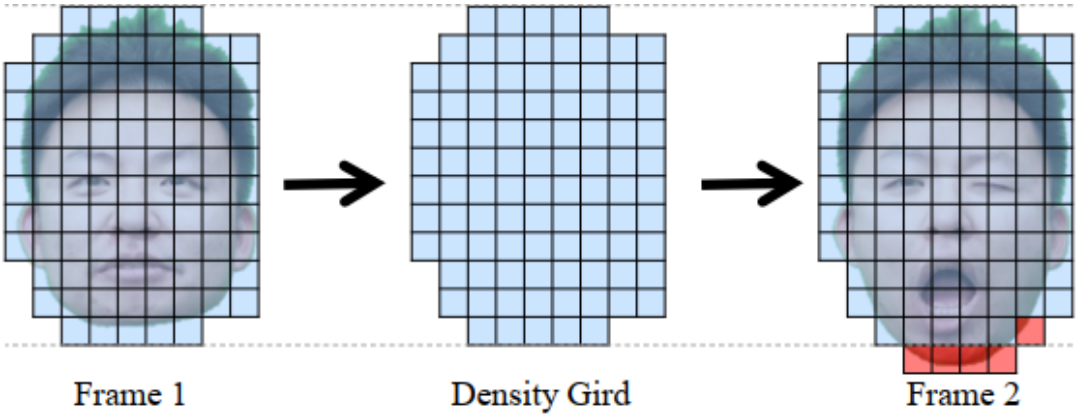

图 3: 密度网格场示意

为了防止出现这种现象，文章提出了一种动态的人脸密度场更新策略, 该策略也是本次实验复现

中的重点，文章使用了一个 128^3 大小的密度网格来存储局部密度信息，并设计一个阈值，当密度低于阈值时将其记为 0，其他为 1，从而得到一个密度网格的二值场，于是在光线采样的时候可以避开密度低的位置，从而提高模型的收敛速度。在实现过程中，每一个基对应的密度场可以按照公式 10进行计算:

$$\hat{h}_i = h_0 + \hat{w}_i h_i, \tag{10}$$

其中 $\hat{w}_i$ 为所有帧的 i 表情的最大值，获得了每一个基对应的密度场之后计算所有的 $\hat{h}_i$ 的最大值以获取最终的密度网格来逼近自然人脸的活动范围。

4.3 参数设置

与原始论文的不同之处，在原来的文章中，加入了 batchsize 这个参数，并设置为 4，并且设置训练时单张图像采样的光线的数目设置为 1024，这样设置是为了提高训练速度，但是本人在实验过程中认为这个设置并没有太大必要并且会提高代码的复杂程度，而且在 NeRF 中采样的是光线，其采样的个数已经具备 batchsize 的含义了，所以复现实验中设置 batchsize 为 1。其他的参数设置如表 1所示。

表 1: 网络参数设置

参数	值
网格层数	16
哈希表长度	2^14
特征维数	4
初始网格分辨率	16
最终网格分辨率	1024
初始化分布	$U(-10^4, 10^4)$

4.4 与已有开源代码对比

本文作者并无开源训练代码，于是本文在 torch-ngp[17]代码的基础上对本文提出的算法进行复现，torch-ngp 是对 Nvidia 提出的 Intant-NGP 的 pytorch+CUDA 的复现代码，能够在以秒为时间单位对输入的图像序列进行静态场景的三维隐式重建。不同于静态场景，本文针对的是具有不同表情动作的人脸说话视频序列，属于动态的场景，通过输入的表情系数对人脸进行驱动。由于人脸说话视频的特殊性，相机的位置是没有变动的，于是本文参考 ADNeRF[4]的处理方式，通过估计人脸朝向以作为相机的不同姿态作为神经辐射场的输入，使用预训练的 Bisenet[18]对人脸进行语义分割。

针对动态的人脸说话视频，本人按照原文的处理方式，将单个哈希表换成多个哈希表作为待优化的参数，并且通过哈希表来指定对应的密度网格。仿照原文提出的损失函数，本人在预处理中获取图像掩膜并和辐射场积分所获取的图像透明图图像作损失，并且采用 VGGLOSS[12]作为 LPIPS 损失。由于人脸的嘴部动作较多，复现过程中在获取光线的过程中，更多地在嘴唇部分进行采样，在使用 LPIPS 损失的时候，1/2 的概率选取嘴巴部分，1/2 的概率选取其他部分。

4.5 实验环境搭建

本文在 torch-ngp[17]代码的基础上对本文提出的算法进行复现, 并使用[15]的预处理方式，使用 CPEM[16]估计人脸的表情系数。通过克隆这几份仓库以及本文开源的代码即能够搭建本文实验环境。

4.6 创新点

本文在复现的基础上仍提出以下创新点，注意到原论文中提出了掩膜损失函数，该损失函数用于将每一个像素的透明度尽可能地接近 0 和 1，于是参考[15]方式使用了以下熵正则损失函数如公式 11所示：

$$L_{entropy} = -\sum_{\alpha \in I} \alpha \log \alpha + (1-\alpha) \log (1-\alpha) \tag{11}$$

其中 α 为每一个像素对应的透明度，实验过程中发现这处改进对实验结果没有太大变化，仅对 PSNR 提升 0.4 个点左右。

5 实验结果分析

本次实验仿照原论文，使用了 5 个计算图像相似度的评价指标，本文的复现结果和论文展示的结果对比如表 2所示。由于没有一个公开公用的数据集用于公平评价，本次复现使用本文作者提供的 8 个视频分别进行实验和计算这几个视频在 5 个指标下的结果取平均并取标准差，得到了如表 2所示的结果，可以看到，本次的复现结果与原论文相差不大，部分指标低于原论文但仍有部分指标高于原论文的结果。

表 2: 实验结果

指标	原论文结果	复现指标
PSNR↑	**34.15(2.58)**	32.58(1.78)
MSE(10^{-3})↓	**0.48(0.32)**	0.63(0.23)
L1(10^{-2})↓	0.70(0.23)	**0.60(0.27)**
SSIM(10^{-1})↑	**9.73(0.13)**	9.68(0.11)
LPIPS(10^{-2})↓	2.67(1.32)	**2.50(1.08)**

另外，本次实验还尝试输入其他不同人的人脸姿态和表情系数，驱动训练好的 NeRFBlendShape 模型中，以生成相同动作的人脸图像，图 4中展示了部分结果，从中可以看到本次复现实验能够较好的完成人脸表情的迁移。在附件中有更多迁移结果。本次复现过程中，一个人脸的 NeRF 模型训练大概需要 10 20 分钟，训练时间与原论文提到的一样快。

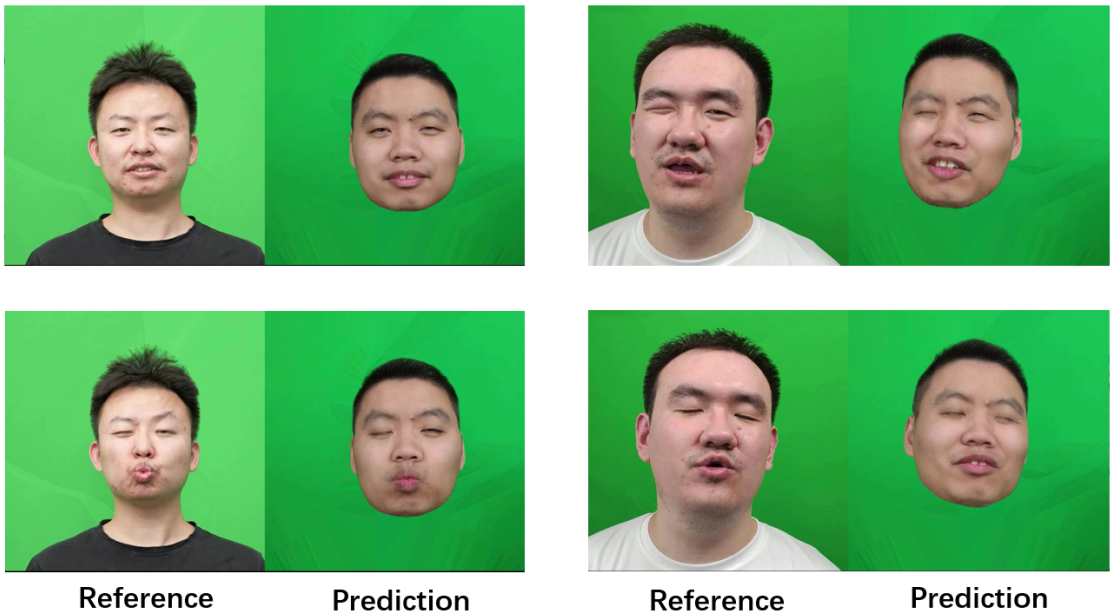

图 4: 表情迁移实验

6 总结与展望

本篇报告对一篇近期工作 NeRFBlendShape 模型进行完整复现，并介绍整个工作的动机、算法设计以及给出了与原文相近的实验结果。由于该工作的作者并没有开源代码，本次复现同样希望该工作能给予研究者们予启示。本文主要通过预测输入图像序列中人脸的表情系数，通过表情系数的线性组合来得到最终用于特征查询的多尺度哈希表中，然后通过一个 MLP 预测采样点的颜色和密度，通过体积渲染的方式获取到预测图像，通过与原图的多个损失函数使得网络能够在较短的时间内收敛。与原论文中提到的一样，本文的方法仍然在部分图像中存在有噪声，并且在输入训练时没做过的夸张表情时难以实现表情的迁移，如图 5所示。这也是该工作的不足之处，未来的研究工作可以从这方面出发，将表情迁移做得更加精细同时还能保持较高的一致性。

图 5: 表情迁移不足之处

参考文献

[1] GAO X, ZHONG C, XIANG J, et al. Reconstructing Personalized Semantic Facial NeRF Models From Monocular Video[J]. ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia), 2022, 41(6). DOI: 10.1145/3550454.3555501.

[2] CAO C, WENG Y, ZHOU S, et al. FaceWarehouse: A 3D Facial Expression Database for Visual Computing[J/OL]. IEEE Trans. Vis. Comput. Graph., 2014, 20(3): 413-425. https://doi.org/10.1109/TVCG.2013.249. DOI: 10.1109/TVCG.2013.249.

[3] BLANZ V, VETTER T. A morphable model for the synthesis of 3D faces[C]//Proceedings of the 26th annual conference on Computer graphics and interactive techniques. 1999: 187-194.

[4] GUO Y, CHEN K, LIANG S, et al. AD-NeRF: Audio Driven Neural Radiance Fields for Talking Head Synthesis[C]//IEEE/CVF International Conference on Computer Vision (ICCV). 2021.

[5] GAFNI G, THIES J, ZOLLHÖFER M, et al. Dynamic Neural Radiance Fields for Monocular 4D Facial Avatar Reconstruction[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2021: 8649-8658.

[6] ZHENG Y, ABREVAYA V F, BÜHLER M C, et al. Im avatar: Implicit morphable head avatars from videos[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022: 13545-13555.

[7] SUN C, SUN M, CHEN H T. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022: 5459-5469.

[8] MÜLLER T, EVANS A, SCHIED C, et al. Instant neural graphics primitives with a multiresolution hash encoding[J]. ACM Transactions on Graphics (ToG), 2022, 41(4): 1-15.

[9] PAYSAN P, KNOTHE R, AMBERG B, et al. A 3D face model for pose and illumination invariant face recognition[C]//2009 sixth IEEE international conference on advanced video and signal based surveillance. 2009: 296-301.

[10] VLASIC D, BRAND M, PFISTER H, et al. Face transfer with multilinear models[G]//ACM SIGGRAPH 2006 Courses. 2006: 24-es.

[11] CHAN E R, LIN C Z, CHAN M A, et al. Efficient geometry-aware 3D generative adversarial networks [C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022: 16123-16133.

[12] ZHANG R, ISOLA P, EFROS A A, et al. The unreasonable effectiveness of deep features as a perceptual metric[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 586-595.

[13] SCHWARZ K, LIAO Y, NIEMEYER M, et al. Graf: Generative radiance fields for 3d-aware image synthesis[J]. Advances in Neural Information Processing Systems, 2020, 33: 20154-20166.

[14] YU C, WANG J, PENG C, et al. Bisenet: Bilateral segmentation network for real-time semantic segmentation[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 325-341.

[15] TANG J, WANG K, ZHOU H, et al. Real-time Neural Radiance Talking Portrait Synthesis via Audio-spatial Decomposition[J]. arXiv preprint arXiv:2211.12368, 2022.

[16] MO L, LI H, ZOU C, et al. Towards Accurate Facial Motion Retargeting with Identity-Consistent and Expression-Exclusive Constraints[C]//Proceedings of the AAAI Conference on Artificial Intelligence (AAAI). 2022.

[17] TANG J. Torch-ngp: a PyTorch implementation of instant-ngp[Z]. https://github.com/ashawkey/torch-ngp. 2022.

[18] YU C, GAO C, WANG J, et al. BiSeNet V2: Bilateral Network with Guided Aggregation for Real-Time Semantic Segmentation[J/OL]. Int. J. Comput. Vis., 2021, 129(11): 3051-3068. https://doi.org/10.1007/s11263-021-01515-2. DOI: 10.1007/s11263-021-01515-2.