

第一章 引言

1.1 研究背景及意义

近年来，三维虚拟数字人正逐渐走入大众视野，如 2021 年登上春晚舞台的虚拟偶像“洛天依”、2022 年亮相跨年晚会的数字人邓丽君、央视的虚拟主播小 C 等^[1]。随着硬件与虚拟现实设备的快速发展，人们说话时的面部表情、唇部动作，甚至是头部与肢体的动作，都可以帮助听众理解对话内容。视觉和听觉的双模态信息融合的交互方式，不仅能提高用户对内容的理解度，还能提供一种更为准确的交互体验，提高交互的艺术性和观赏度。基于虚拟数字人的应用，如虚拟主持人、虚拟客服和虚拟教师等，不仅提供了更好的人机交互方式，还在感知研究、声音辅助学习等方面具有重要的应用价值，同时也能降低游戏和电影特效的制作成本。

虽然目前大多三维虚拟数字人模型精美、动作逼真，但面部动画的合成严重依赖人为设定的、使用面部捕捉设备、单目或双目摄像头等硬件设备的视觉表演驱动，因设备价格昂贵、获取和处理数据过程复杂，还有受面部遮挡、光照、姿态的影响较大等原因限制了应用场景。由于语音获取方便，受外界影响较小，因此业内开始使用语音驱动的方法合成 3D 人脸动画^[1]。本课题使用深度学习方法实现语音到视觉的映射来达到合成数字人脸动画的目的。

本课题核心是利用深度神经网络拟合语音与 3D 模型 BlendShape 值之间的关系，从而实现语音驱动 3D 数字人的表情和口型。通过本课题学习应用虚拟数字人制作技术，不仅能跟进此领域的最新发展，也能学习深度学习相关知识，最后使用虚幻引擎 UE5 跟进数字人的渲染模块，达到逼真的视觉效果。

1.2 研究内容

基于深度学习方法实现数字人脸动画合成的实现主要包括以下核心功能：数字人大脑、身体、语音、人脸动画、肢体动作。各个部分组织、衔接，最终形成一个较为完整的数字人项目系统。如图 1.1：

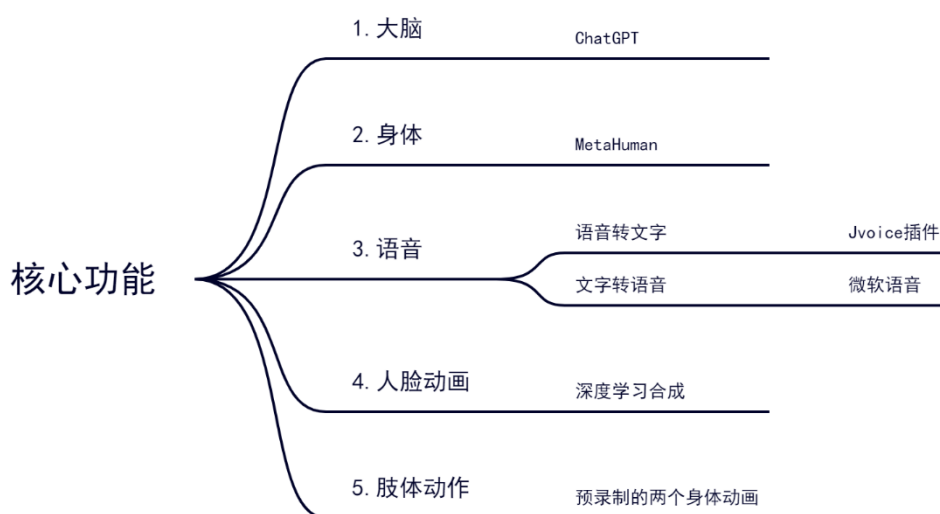


图 1.1 核心功能

下面是工作步骤：

1. 前期准备：使用 UE5 的 MetaHuman Creator 制作高保真的数字人，角色制作完成后会包含完整的绑定，并可以直接在虚幻引擎中制作动画；申请 OpenAI Api Key 用于设置数字人的 ChatGPT 大脑。
2. 语音及相应面部参数收集：录制按首字母从 A-Z 排序的中文字、词、句的视频，在 Maya 中导入录制的声音数据后根据自己的绑定做出符合模型面部特征的对应发音的动画。将动画中每一帧的人脸图像的面部表情编码系统 (Facial Action Coding System ,FACS) 的 AU(Action Unit, 运动单元) 的强度值提取出来（该值又称作表情参数值，记为 AU 值），生成 BlendShape 权重数据。
3. 语音信号处理：将上一步制作的音频中的语音信号进行分帧，将分帧后语音信号的 LPC 特征提取出来。
4. 生成训练集：将以上序号 3 和序号 4 的结果数据组成训练集，用于语音-视觉映射模型的训练。
5. 参数预测：使用卷积神经网络 CNN 进行 BlendShape 参数预测，输入是单帧的语音信号特征，输出是单帧的 BlendShape 值。此模块的工作主要分为两部份：①训练模型：利用参数提取模块和语音信号处理模块分别得到的 BlendShape 参数值和语音信号特征进行训练，得到一个参数预测模型。②参数预测：将新的多帧语音信号特征输入到参数预测模型，得到多帧

BlendShape 参数值组。

6. 语音识别及合成：识别提问者的语音转换成文字，通过 ChatGPT 获取相应的文本答案。在会话过程中将文字答案转换成语音并与动画同步播放。
7. 渲染：该模块基于虚幻引擎 UE5，将序号 5 得到的多帧 BlendShape 参数值通过 UE5 的接口按帧来渲染，从而得到一个连续的人脸动画。渲染的同时也播放语音，wav 语音文件来自于序号 6。

1. 3 内容结构及章节安排

综上所述，本文将划分为五个章节对毕业设计进行详细论述，具体安排如下：

第一章， 介绍本文研究背景、意义、研究内容，并给出文章结构安排。

第二章， 阐述数字人的系统架构，包括各个部分的设置、衔接以及其相应的功能和应用方法。

第三章， 介绍 AU 及 BlendShape 面部表情控制参数，以及从语音到人脸动画的原理。

第四章， 详细叙述从语音到人脸动画的深度学习训练的数据准备、训练过程。

第五章， 分析结果，阐述改进的方向。

第二章 系统架构

2. 1 系统衔接

数字人制作包括大脑、身体、语音（语音转文字&文字转语音）、人脸动画和肢体动作五个部分。在使用虚幻引擎官方的 MetaHuman Creator 制作好具有面部骨骼绑定、身体骨骼绑定的 3D 人物模型（即数字人身体）后，使用以下插件完善数字人制作的其它功能：

- ① 语音转文字：使用 Jvoice 插件返回语音识别后的文本，记为“传入的话”。
- ② 调用 ChatGPT：使用 OpenAI Api Key 针对①中“传入的话”获取 ChatGPT 文本回答并存储为“收到的话”。
- ③ 文字转语音：使用②中的文本“收到的话”作为微软语音的输入，从而输出相应的语音文件（wav 格式）。

- ④ 生成人脸动画：使用 ynnk 插件的 speak 函数作为深度学习预测人物模型 BlendShape 值的 UE5 接口，将③中生成的 wav 语音文件实时生成口型动画；手动添加眨眼动画。
- ⑤ 添加肢体动作：使用 MobReceiver 插件提取绿幕视频中的人物动作导入 UE5，并将两个不同的动作分配给数字人不说话和说话两个不同的状态，根据回调函数进行动作切换。

下面介绍以上五个部份的具体实现：

2.2 语音转文字

数字人的语音交互部分之一是实时识别提问者的语音，这里我们使用 Jvoice 插件完成此功能。在 UE5 中添加插件 JvoiceLib_Course，重启激活插件。使用角色蓝图对接语音服务器，输入的是语音，输出的是文字。

在数字人蓝图中，新建事件图表，自定义事件“来自关卡蓝图的调用”。自定义事件“初始化语音插件”，调用 Jvoice_WindowsInit 初始化函数，Get Jvoice Pointer 作为该函数的目标输入。连接新节点 Jvoice_Openexe 调用外部 exe URL（即本地 Jvoice_main.exe 的位置），之后用 Jvoice_Execute 函数（参数设为 1）执行该 exe。最后打印识别出来的语音文本，加入 Delay 函数将识别出来的文字设置为延迟 0.2 秒以^①便于用户查看。将自定义事件“来自关卡蓝图的调用”和自定义事件“初始化语音插件”关联起来，最后在关卡蓝图中拖入人物资产 BP_Asset 作为“来自关卡蓝图的调用”的输入。将输出的结果保存为字符串“传入的话”。

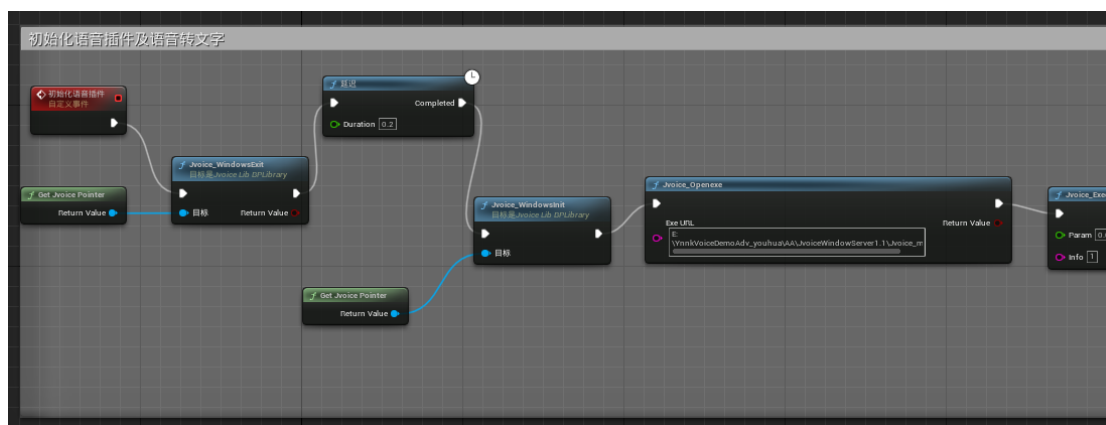


图 2.1 初始化语音插件及语音转文字

2.3 调用 ChatGPT

语音输入并对接 OpenAI，输入的是文字，输出的是 ChatGPT 返回的结果。

在数字人蓝图中添加自定义事件“获取 GPT 返回文本”，调用函数 Set Open AI Api Key，填入 OpenAI Api Key^[10]。调用函数 Open AICall GPT3（Engine 参数设置为效果最好的 TEXT DAVINCI 003，Prompt 参数设置为“传入的话”，将 Settings 分割结构体引脚后设置其中的 Settings Max Tokens 为 150）。获取 Completion 结构数组后使用循环 For Each Loop 获取并存储 ChatGPT 返回的内容为字符串变量“收到的话”，最后使用 Print String 打印“收到的话”。

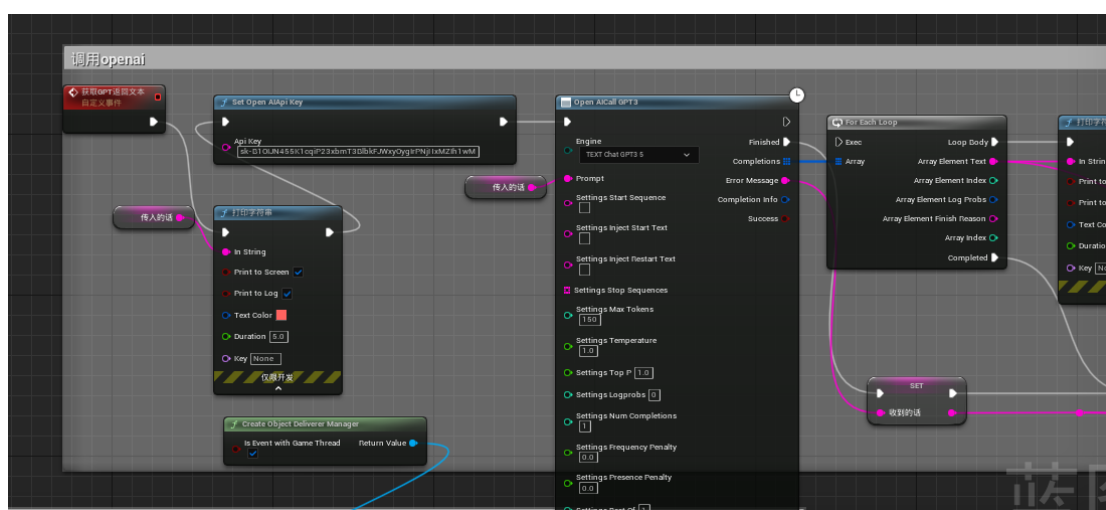


图 2.2 调用 ChatGPT 获取返回文本

2.4 文字转语音

数字人语音交互的另一部分是将 ChatGPT 返回的文本答案合成为语音，这里我们使用微软语音完成此功能。使用 AzSpeech 插件，配置好微软语音的 Api Key^[15]，最后将生成好的语音保存为 wav 文件，以便于生成口型时调用。

最终生成的 wav 语音保存在本地，命名为 wqeqweqweq.wav。

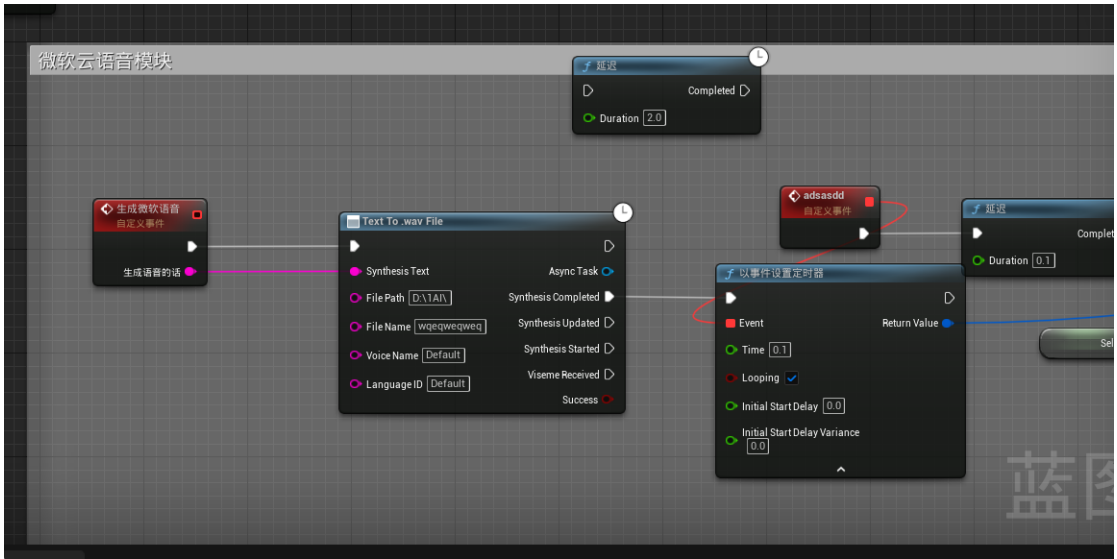


图 2.3 使用微软语音服务合成语音

2. 5 生成人脸动画

新建自定义事件“生成语音”，调用 Load Wave File to Sound Wave 函数，填入 wqeqwqewq.wav 文件的路径，设置分支判断音频是否获取成功，若成功则调用 Recognize Runtime Sound Wave 函数载入音频，最后调用 Ynnk Lipsync Controller 接口的 speak 函数^[13]利用音频生成口型 BlendShape 数据，并手动加上眨眼动画

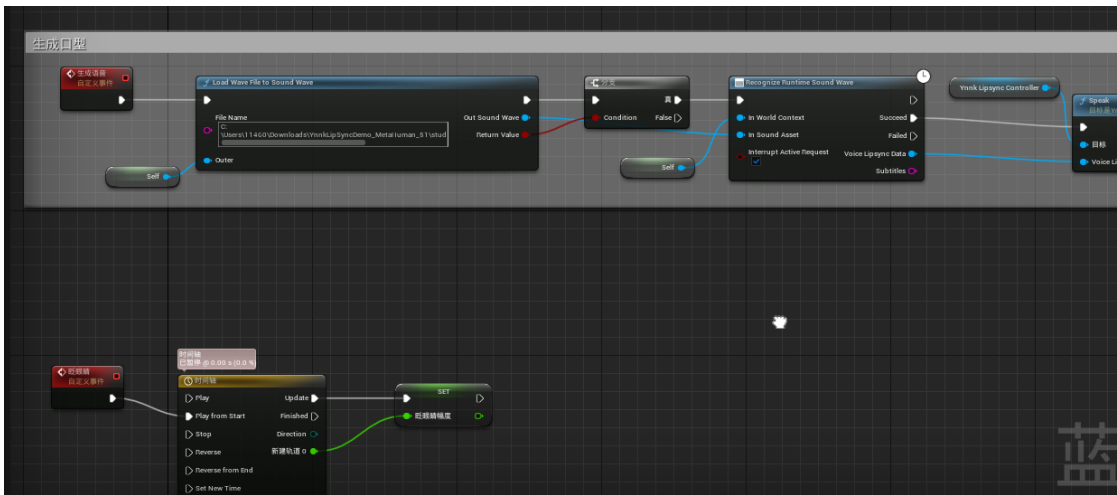


图 2.4 生成人脸动画

2. 6 添加肢体动作

利用 MobReceiver 插件及它的两个外部程序 VMctoMOP.exe、ThreeDPoseTracker.exe^[14]识别绿幕视频中人物的动作，并录制相应的动画序列保存在 UE5 中等待数字人使用。在建立好 Receiver Actor 之后给数字人绑定 Mop Receiver，数字人的身体使用 MetaHuman_Rotation_only 蓝图即可调用肢体动作动画。



图 2.5 添加肢体动作

第三章 从音频到面部表情参数

3. 1 FACS 面部表情编码系统和 AU 运动单元

面部动作编码系统（FACS）是根据面部表情对人类面部动作进行分类的系统，该系统最初由瑞典解剖学家 Carl Herman Hjortsjö 开发。后来被 Paul Ekman 和 Wallace V.Friesen 引用，并于 1978 年出版了一本关于这套系统的书。Ekman, Friesen, Joseph C.Hager 在 2002 年发表了一篇关于 FACS 的文章，文章对 FACS 系统进行了内容更新与补充。FACS 根据面部外观的细微变化的瞬间来对面部肌肉的运动进行编码。这个系统是对情绪的具体表现进行分类的通用标准，并且已经证明对心理学家和动画师是有用的。由于主观性和时间消耗的问题，FACS 已经被建立为一个计算机自动系统，可以检测视频中的人脸，提取人脸的几何特征，然后生成每个人脸运动的时间轮廓。

FACS 的核心是将人脸划分了 70 个左右的独立运动单元，也叫 AU（Action

Units)。AU 可以用来描述面部表情的不同方面，例如眼睛张开程度，眉毛挑起程度以及嘴巴张开程度等。每个 AU 都有一个编号，表示其在面部表情分析中的重要性，并且还有一个对应的数字强度，表示其在当前表情中的活动程度。下面是部份 AU 编号及其对应的面部表情描述：

表 3.1 AU 编号及其对应的面部表情描述^[6]

1	AU1	Inner Brow Raiser (内眉毛提起)	描述眉毛向上移动的程度
2	AU2	Outer Brow Raiser (外眉毛提起)	描述眉毛向上移动的程度
3	AU4	Brow Lowerer (眉毛下垂)	描述眉毛向下移动的程度
4	AU5	Upper Lid Raiser (上眼皮提起)	描述上眼皮向上移动的程度
5	AU6	Cheek Raiser (颧骨提起)	描述颧骨向上移动的程度
6	AU7	Lid Tightener (眼皮紧缩)	描述眼皮紧缩的程度
7	AU9	Nose Wrinkler (鼻子皱纹)	描述鼻子皱缩的程度
8	AU10	Upper Lip Raiser (上唇提起)	描述上唇向上移动的程度
9	AU12	Lip Corner Puller (嘴角拉动)	描述嘴角向上移动的程度
10	AU14	Dimpler (颊骨凹陷)	描述颊骨凹陷的程度
11	AU15	Lip Corner Depressor (嘴角下垂)	描述嘴角向下移动的程度
12	AU17	Chin Raiser (下巴提起)	描述下巴向上移动的程度
13	AU20	Lip Stretcher (嘴唇伸展)	描述嘴唇横向拉动并变得平扁
14	AU23	Lip Tightener (嘴唇紧缩)	描述嘴唇紧缩的程度
15	AU25	Lips Part (嘴唇分开)	描述嘴唇分开的程度
16	AU26	Jaw Drop (下巴下垂)	描述下巴向下移动的程度
17	AU45	Blink (眨眼)	描述眨眼的次数

AU 单独看起来都非常奇怪，但是当这些 AU 单元组合起来，就能呈现不同情绪的表情。例如 AU1+AU2、AU1+AU4、AU1+AU2、AU1+AU2+AU4：

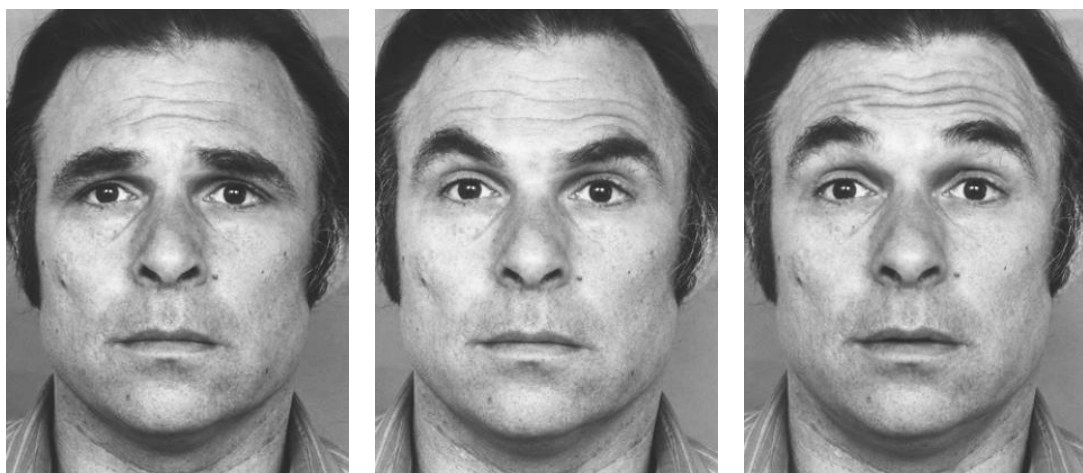


图 3.1 从左到右依次是 AU1、AU2、AU1+AU2

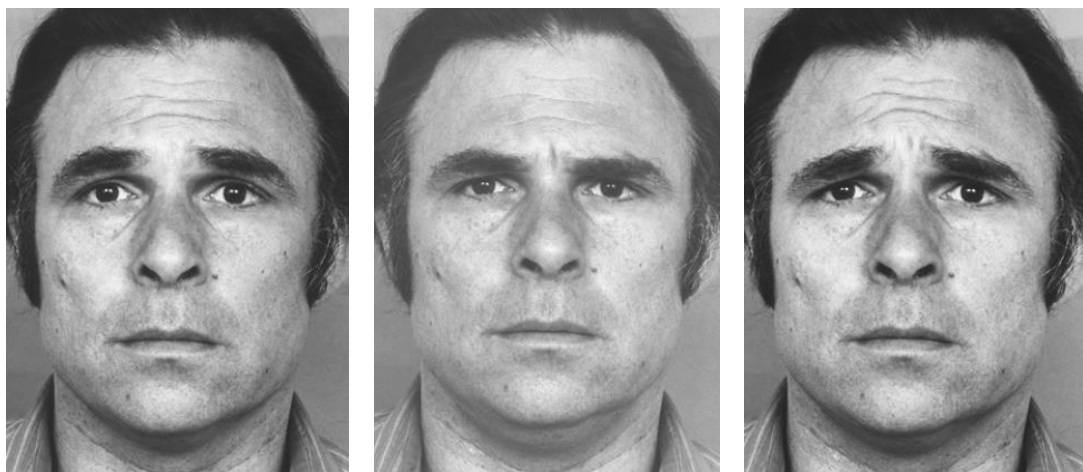


图 3.2 从左到右依次是 AU1、AU4、AU1+AU4

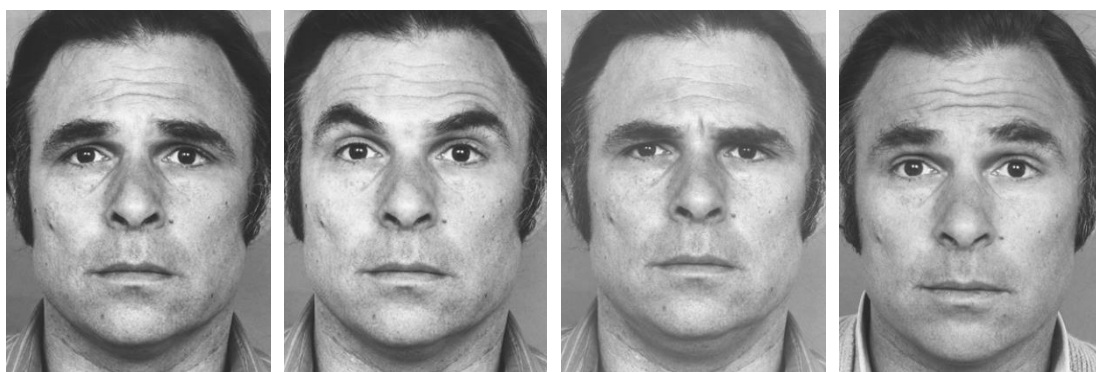


图 3.3 从左到右依次是 AU1、AU2、AU4、AU1+AU2+AU4

3. 2 BlendShape 面部表情参数

FACS 是面部表情的权威参照标准，被各行各业广泛使用。BlendShape 融合形变变形器就关联了 AU，在建模软件或游戏引擎中调整各个 AU 的强度即可让人物模型脸部做出形变。

BlendShape 的主要工作原理是读取“目标形状”中的顶点索引与位置信息，依据权重让“基础网格”的各个顶点向“目标形状”做插值运算，使“基础网格”变形。假设网格体有三个目标形状，且对应权重分别为 0.2、0.5、0.8，则落实到模型网格中的每一个顶点，顶点根据权重通过 BlendShape 做插值运算获得新坐标，对网格上的所有顶点都如此操作即可让网格体呈现出不同形状^[7]。如图 3.4：

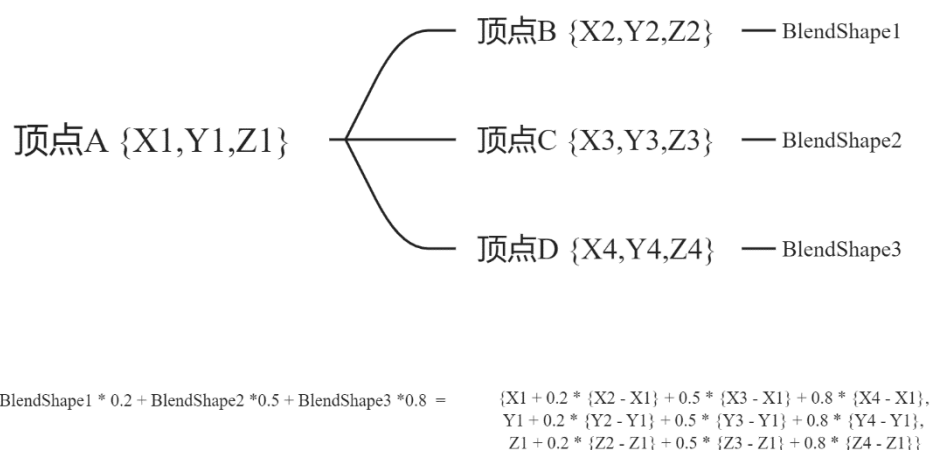


图 3.4 BlendShape 对顶点 A 做插值运算

BlendShape 具有灵活控制每一个顶点的优点，且可以控制顶点位移的权重、叠加位移效果，也不会影响网格的原始数据。使用 BlendShape 制作人脸表情动画可以弥补传统骨骼和控制器绑定在表情制作上的不足，通过将人脸部分拆为 AU 来制作和控制，可以做出多种多样的表情组合，达到较好的效果。但是通过上面的权重公式可以看出顶点的位移是线性的，每个顶点都在往目标位置做平移运动，因此 BlendShape 很难做到有弧度的动画效果。

在 UE5 中每个 MetaHuman 脸部使用 61 个 BlendShape 配合蒙皮骨骼制作人脸表情动画，动画效果更上一层楼，如：在闭眼时修改眼皮形状，使上下眼皮更贴合；在张嘴时修改嘴唇形状，使嘴唇更饱满立体等。且我们在此课题中使用标准表情参数序列与变形后的目标参数序列进行混合，最终利用 BlendShape 中的 AU 值强度变换做出效果较好的人脸动画。

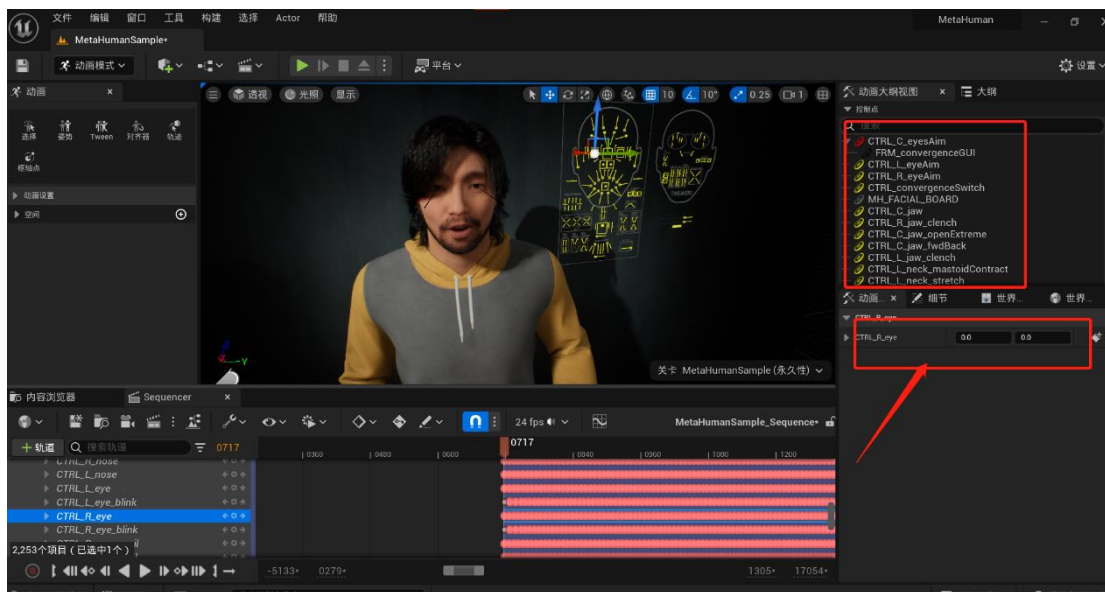


图 3.5 MetaHuman 中的 BlendShape

3.3 语音驱动面部算法 Audio2Face

Audio2Face 算法可以将语音实时转换为 BlendShape 动画，即实时将音频数据转换为驱动数字人面部 BlendShape 的权重数据。

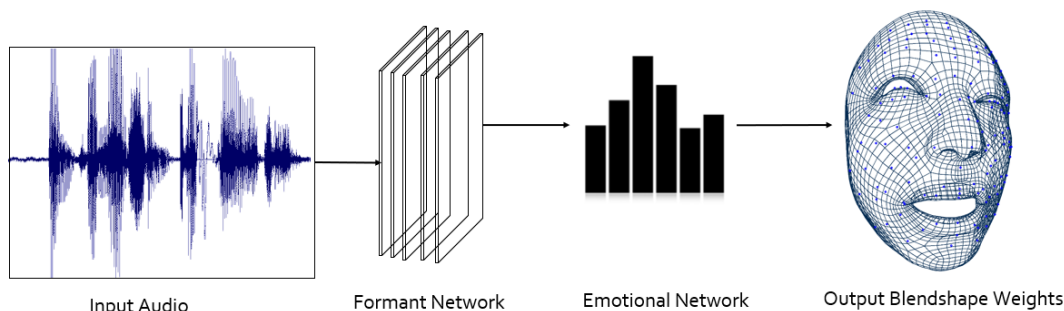


图 3.6 Audio2Face 算法

Audio2Face 的主要原理是利用深度学习训练语音-视觉映射模型，从而预测语音对应的视觉数据。在这里我们使用了两种数据，分别是声音数据和声音对应的动画数据。声音数据主要是录制中文字母表的发音，以及一些特殊的爆破音。而动画数据就是，在 maya 中导入录制的声音数据后，根据自己的绑定做出符合模型面部特征的对应发音的动画。

我们将声音数据和动画数据按帧分割，并标注好二者一一对应的标签，在提取每一帧声音的 LPC 特征和动画的 BlendShape 数据后建立数据集，利用深度学习卷积神经网络 CNN 对二者做非线性映射关系的训练。最后利用训练好的模型对从新输入的语音提取出来的语音信号特征做出其对应的视觉 BlendShape 数据

预测，将离散的 BlendShape 值滤波平滑后输入到建模软件或游戏引擎的人物模型中即可获得预测之后的脸部动画。

第四章 深度学习训练

4.1 章节要点

深度学习神经网络模型训练大致可以分为三个阶段：数据采集制作，数据预处理，数据模型训练。

第一阶段，数据采集制作。这里主要包含两种数据，分别是声音数据和声音对应的动画数据。

声音数据主要是录制中文字母表的发音，以及一些特殊的爆破音。而动画数据就是，在 maya 中导入录制的声音数据后，根据自己的绑定做出符合模型面部特征的对应发音的动画。

第二阶段，主要是通过 LPC 对声音数据做处理，将声音数据分割成与动画对应的帧数据，及 maya 动画帧数据的导出。

第三阶段就是将处理之后的数据作为神经网络的输入，然后进行训练直到 loss 函数收敛即可。

4.2 数据准备

录制语音和视频，并在 Maya 中从视频创建动画。声音必须包含元音，且尽可能包括更多的发音。使用 LPC 处理语音，将语音分割成与 Maya 中的动画帧对应的片段帧。

使用 Exportbsweights.py 从 Maya 导出 BlendShape 权重文件，得到 BS_name.npy 和 BS_value.npy。使用 step1_LPC.py 处理 wav 文件以获得 lpc_*.npy。将 wav 文件预处理为二维数据。其中 BS_name.npy 存储了 61 个 BlendShape 节点名字，包括 EyeBlinkLeft、EyeLookDownLeft、EyeLookInLeft、EyeLookOutLeft、EyeLookUpLeft、EyeSquintLeft 等等；BS_value.npy 是这 61 个节点对应的取值范围为-1~1 的小数组成的向量，向量的每一行代表一帧的所有节点数据。

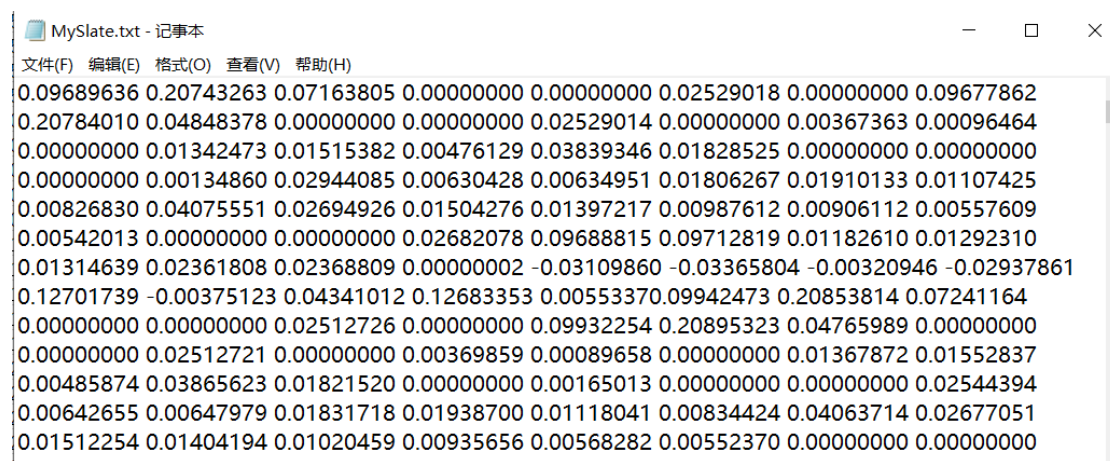


图 4.1 将 BS _ value. npy 转换为 txt 后的数据

4.3 训练过程

使用的框架包含三个部分：①Formant networ：对输入音频剪辑进行固定功能分析；②Articulation network：在 ReLU 激活后将 BlendShape 向量连接到每个卷积层的输出；③Output network：将音频特征扩展为 BlendShape 权重^[8]。

各个网络层具体信息如图 4.2：

Formant network

Layer Type	Kernel	Stride	Outputs	Activation
Autocorrelation	-	-	1x64x32	-
Convolution	1x3	1x2	72x64x16	ReLU
Convolution	1x3	1x2	108x64x8	ReLU
Convolution	1x3	1x2	162x64x4	ReLU
Convolution	1x3	1x2	243x64x2	ReLU
Convolution	1x2	1x2	256x64x1	ReLU

Articulation network

Layer Type	Kernel	Stride	Outputs	Activation
Conv + concat	3x1	2x1	(256+E)x32x1	-
Conv + concat	3x1	2x1	(256+E)x16x1	ReLU
Conv + concat	3x1	2x1	(256+E)x 8x1	ReLU
Conv + concat	3x1	2x1	(256+E)x 4x1	ReLU
Conv + concat	3x1	2x1	(256+E)x 1x1	ReLU

Output network

Layer Type	Kernel	Stride	Outputs	Activation
Fully connected	-	-	150	Linear
Fully connected	-	-	116	Linear

图 4.2 网络层

loss 函数包括三个损失项：①位置项（Position Term）：描述了预测输出与期望输出之间逐顶点位置坐标的均方差值，使得每个输出的顶点的位置大致正确，位置项只约束了单帧预测误差，并未考虑帧间关系，容易导致帧间表情波动；②运动项（Motion Term）：确保输出的帧间顶点运动趋势与数据集保持一致，可以有效避免表情抖动，使得表情动作更加平滑；③正则项（Regularization Term）：用于确保网络正确地将短时效应归因于音频信号，而将长时效应归因于 BlendShape，避免学习到的 BlendShape 包含与音频信号相似的特征^[3]。

将数据集按 7: 3 比例分为训练集和验证集进行模型的训练和评估。对数据进行迭代训练直到 loss 函数收敛。最后使用验证集对准确率进行评估。

第五章 结果分析与改进

5.1 结果分析

- 1) 只看口型部分：口型部分能根据语音驱动起来，虽然较为粗糙，但是可以实现基本的“数字人开口说话”的口型效果。
- 2) 口型和语音合成部分：二者是在提问者的问题输入 ChatGPT 获得返回文本后开始合成。但口型部分需要等返回文本合成为语音并分析语音每一帧的特征后预测相应的口型 BlendShape 数据之后才能合成，虽然我们进行了一些设置让口型和语音同时播放，但是合成好的语音还是比口型领先一步，产生了“音画不同步”的效果。
- 3) 时间延迟：①最主要的时间延迟在获取 ChatGPT 返回文本部份。由于需要开全局代理并输入文本来获得返回文本，这种在线应答的方式需要有一段反应时间。不论输入文本字数多少，获取返回文本的等待时间是大体一致的。②其次是生成微软语音部份。这部分的延迟时间与返回文本长度成正比，也影响着口型部分的生成。③其余部分因为均为本地部署，延迟率较低。
- 4) 动作部分：只录制了两个动作给数字人使用，一个是提问前不说话的動作，一个是回答问题时的动作，动作较少、持续时间较短，所以如果回答问题时间较长，回答问题时的动作动画序列就会反复播放，产生重复，影响观感。

5.2 改进

- 1) 口型和语音合成部分：调参让语音比口型提前播放，改善“音画不同步”的效果。
- 2) 时间延迟：可以尝试本地部署 ChatGPT，降低延迟率。
- 3) 动作部分：录制多个动作给数字人使用，保证动作数量多且持续时间适中或较长。设置动作的随机播放，改善“很多重复动作”的效果。可以的话还可以设置返回文本的情感分析来匹配相应的动作动画序列，使得效果更好。
- 4) 表情部分：可以设置返回文本的情感分析来匹配相应的面部参数。如：当数字人说到“开心”、“很好”等词汇时给她加上微笑的表情，说到“糟

糕”、“我不明白您在说什么”这些语句时给她加上失落的表情等。

参考文献

- [1] 刘贤梅,刘露,贾迪,赵娅,田枫. 基于语音驱动的三维人脸动画技术综述[J].计算机系统应用,2022,31(10):44-50.DOI:10.15888/j.cnki.csa.008776.
- [2] 刘露. 基于 Blendshapes 的语音驱动三维人脸动画技术研究[D].东北石油大学,2022.DOI:10.26995/d.cnki.gdqsc.2022.000383.
- [3] 蔡国鑫. Audio2Face 基于音频文件智能生成虚拟角色面部动画[J].现代电影技术,2021(09):60-61.
- [4] Tero Karras, Timo Aila, Samuli Laine, Antti Herva, and Jaakko Lehtinen. Audio-Driven Facial Animation by Joint End-to-End Learning of Pose and Emotion[J]. ACM Transactions on Graphics, Vol. 36, No. 4, Article 94. Publication date: July 2017.
- [5] 岳旸. 基于深度学习网络模型实现语音驱动的人脸动画合成[D].北京交通大学,2020.DOI:10.26944/d.cnki.gbfju.2020.002004.
- [6] CMU. FACS-Facial Action Coding System [EB/OL]. <http://www.cs.cmu.edu/face/index2.htm>. 访问日期: 2023-05-03.
- [7] CSDN 博客.[技术]blendShape 基础知识与创建方法 [EB/OL].https://blog.csdn.net/weixin_54482403/article/details/128865028.访问日期: 2023-05-03
- [8] Github.FACEGOOD-Audio2Face[EB/OL].<https://github.com/FACEGOOD/FACEGOOD-Audio2Face/tree/main>.访问日期: 2023-05-03
- [9] CSDN 博客.[技术]Unity & FACEGOOD Audio2Face 通过音频驱动面部 BlendShape [EB/OL].https://blog.csdn.net/qq_42139931/article/details/125891735.访问日期: 2023-05-03.
- [10] CSDN 博客.[技术]OpenAI 的 API key 获取方法 [EB/OL].<https://blog.csdn.net/pondbay/article/details/129782572>.访问日期: 2023-05-03.
- [11] 哔哩哔哩官网.【教程】AI 语音生成原神崩三插件 VITS 手把手教你安装[EB/OL].https://www.bilibili.com/video/BV1ot4y1P7a7/?share_source=copy_web&vd_source=b8c16c5eb7c4912850895608cccef681.访问日期: 2023-05-03.
- [12] CSDN 博客.[技术]UE4ObjectDeliever[EB/OL].https://blog.csdn.net/weixin_44840850/article/details/129161434.访问日期: 2023-05-03.
- [13] UnrealEngine 官网. Ynnk Voice Lipsync[EB/OL].<https://www.unrealengine.com/marketplace/en-US/product/ynnk-voice-lipsync>.访问日期: 2023-05-03.
- [14] CSDN 博客.[技术]UE4.27.0 动捕 + 面捕 [EB/OL].<https://blog.csdn.net/xiaoyuchixiaomao/article/details/120514880>.访问日期: 2023-05-03.
- [15] 哔哩哔哩官网.【微软云 Azure】快速上手 Azure Speech Services AI 语音合成技术 [EB/OL].https://www.bilibili.com/video/BV1J44y1n7ay/?spm_id_from=333.337.search-card.all.click&vd_source=57852e4af3f23ca3c0d2c2247951b06a.访问日期: 2023-05-03.

致谢

是结束也是开始。感谢所有我的老师以及帮助和鼓励过我的同学。