三维人体建模与多模态信号驱动手势动作生成

张颖颖

指导教师 曾鸣

厦门大学

厦門大學

本 科 毕 业 论 文 （设 计）

（主修专业）

三维人体建模与多模态信号驱动手势动作生成

3D Human Body Modeling and Multimodal Signal-Driven Gesture Action Generation

姓　　名：张颖颖

学　　号：21620182203547

学　　院：信息学院

专　　业：数字媒体技术

年　　级：2019 级

校内指导教师：曾鸣 副教授

二〇二三年 5 月 8 日

厦门大学本科学位论文诚信承诺书

本人呈交的学位论文是在导师指导下独立完成的研究成果。本人在论文写作中参考其他个人或集体已经发表的研究成果，均在文中以适当方式明确标明，并符合相关法律规范及《厦门大学本科毕业论文（设计）规范》。

该学位论文为（　　）课题（组）的研究成果，获得（　　）课题（组）经费或实验室的资助，在（　　）实验室完成（请在以上括号内填写课题或课题组负责人或实验室名称，未有此项声明内容的，可以不作特别声明）。

本人承诺辅修专业毕业论文（设计）（如有）的内容与主修专业不存在相同与相近情况。

学生声明（签名）：

2023 年 5 月 8 日

致 谢

在这篇论文完成之际，我想衷心感谢那些曾经帮助过或支持过我毕业设计的人们！

首先，我要感谢我的导师曾鸣老师，从选题的确定，到实验的指导，大纲的拟定等等，曾鸣老师都在百忙之中给予非常重要的指导。我非常感谢曾鸣老师对我的毕设提出的专业指导和建设性建议，这些指导和建议对我完成毕设非常有帮助。

其次，我要感谢 Vcg 实验室同学们的帮助。无论是问题的答疑解惑还是工程的部分技术支持，都离不开同学们的帮助与引导。

最后，我要感谢我的朋友们对我精神上的鼓励以及我的父母对我生活上的支持。

摘　要

随着当代数字技术的不断进步，数字人技术已成为计算机图形学和计算机视觉领域的重要研究课题。本文提出了一种数字人系统，该系统是一个综合性的研究课题，涵盖了人体建模、动作生成等多个领域。

在人体建模方面，本文采用了基于融合形状 （Blendshape）和基于骨骼驱动的两种方法，分别实现了自定义人脸建模、表情建模以及体型建模，并将这些模型应用到数字人系统中。

在多模态信号驱动手势动作生成模块中，本文介绍了基于CaMN网络的5个编码器和2个解码器，以及相应的损失函数。利用CaMN网络，本文成功实现了多模态信号驱动手势动作生成模块。

同时，由于需要将驱动生成的动作在Unity引擎中对数字人进行应用和测试，本文也介绍了数字人的动作模块。最后，本文通过将以上不同领域的技术相结合，构建了一个完备的数字人系统，并应用于Unity引擎中。

综上所述，本文设计并实现了一个数字人系统，包含了三维人体建模以及多模态信号驱动手势动作生成模块，并成功应用于Unity引擎中。

关键词：数字人；三维人体建模；多模态信号驱动；手势动作生成；Unity 引擎

Abstract

With the continuous advancement of contemporary digital technology, digital human technology has become an important research topic in the fields of computer graphics and computer vision. This paper proposes a digital human system, which is a comprehensive research topic that covers multiple fields such as human body modeling and motion generation.

In terms of 3D human body modeling, this paper adopts two methods based on Blendshape and skeleton-driven, respectively, to achieve custom face modeling, expression modeling, and figure modeling, and applies these models to the digital human system.

In the multimodal signal-driven gesture action generation module, this paper introduces 5 encoders and 2 decoders based on the CaMN network, as well as corresponding loss functions. Using the CaMN network, this paper successfully realized multimodal signal-driven gesture action generation module.

Moreover, because the generated actions need to be applied and tested on digital humans in the Unity engine, this paper also introduce the motion module of the digital human. Finally, by combining the technologies from different fields mentioned above, we construct a complete digital human system and apply it to the Unity engine.

In conclusion, this paper designs and implements a digital human system that includes 3D human body modeling and multimodal signal-driven gesture action generation modules, which is successfully applied to the Unity engine.

Key words: Digital Human; 3D Human Body Modeling; Multimodal Signal-Driven; Gesture Action Generation; Unity Engine

目 录

Contents

第一章 绪论

1.1 研究背景

在当今数字化时代的背景下，数字人的研究已经成为了人工智能、虚拟现实、游戏等领域的热门话题。数字人的研究旨在提高用户体验和创造更真实的虚拟世界。然而，在科学研究领域，构建真实的数字虚拟人仍需要进行漫长的探索和研究。在《2023年数字人产业发展白皮书》中指出，数字人可以分为三种类型：“内容/IP型”、“功能服务型”和“虚拟分身型”。这三种数字人的核心竞争力分别是“形象的艺术性与IP打造”、“智能交互能力”和“沉浸化、实时化以及体验感”。因此，数字人如果能够拥有独特的形象和真实自然的交互，就能提升数字人的核心竞争力。此外，白皮书还提到，数字人拥有人的外观、人的行为和人的思维三个方面的特征。为了更好地掌握这些特征，可以开发一个数字人系统，实现自定义外观，并自动生成人的行为，以表达人的思想，从而使数字人具备更完善的能力和个性。基于这一研究背景，本研究旨在探索数字虚拟人的三维人体建模和多模态信号驱动手势动作生成方法，以提高数字人的逼真度和表现力。

1.2 研究现状

在三维人体建模方面，传统的建模方法需要耗费大量的时间和精力，而且难以满足个性化需求。近年来，许多游戏开发商都开发了个性化的捏脸系统，并不断迭代提高其自由度，旨在让玩家能够创造属于自己的人物角色，增加游戏的自由度和可玩性。基于这一研究现状，本文的数字人系统也实现了个性化的人体建模模块。

另一方面，为了使数字人更加完整，除了具备人的形象，数字人还应该具备人的动作行为。国内外各大厂商都在探索更加多样化和个性化的数字人手势和动作表达方式，以提高用户体验。针对数字人的手势和动作，目前有三种主要的解决方案：

1、运用动捕技术，即对特定情境下的行为进行动作捕捉，并生成对应手势和动作。

2、另外，一些对话系统直接复用之前已有的一些动作模板来表达意图。然

而动作复用带来的问题很明显，例如在游戏中，由于NPC手势和动作的不断重复，会导致审美疲劳的问题。

3、使用人工智能实现动作生成，使数字虚拟人呈现出多样、真实、自然的动作和手势。多模态信号驱动手势动作生成是一种基于人工智能技术的动作生成方法，它可以利用语音、姿势、面部表情等多种信息来生成自然流畅的动作和手势，从而增强数字虚拟人的表现力和互动性。这也是本文数字人系统实现的一个模块，旨在探究多模态信号驱动手势动作生成方案。

1.3 研究目的和意义

以下能力在数字人研究中具备重要性：

1、自定义形象能力：能够实现数字人的外貌自定义，包括拥有特殊相貌和体型。

2、表达能力：可以通过身体动作、手势和面部表情来表达情感，并展示人类的行为。

为了让数字人拥有自定义外观、表情和动作，需要利用人体建模技术。通过这个技术，用户可以实现高自由度地创造属于自己的数字人形象。此外，为了让创造出来的独特数字人形象进一步拥有人的行为，期望存在一种动作生成技术，该技术能够通过语音和情感等多种模态输入来驱动数字人自然地表现出相应的手势和动作，以显著提升数字人的表达能力，从而降低虚拟主播、游戏 NPC 等多种对话和直播场景的开发和管理成本。此技术还有望应用于增强现实、虚拟现实和智能机器人等多个领域。因此，进一步研究和探索该技术具有重要意义。

1.4 论文结构

本文主要介绍了一个数字人系统的设计和实现，其中包含三维人体建模以及多模态信号驱动手势动作生成模块。论文结构如下：

第一章：绪论。这章介绍本文的研究背景和现状，阐述了研究目的、意义以及论文结构的安排。

第二章：三维人体建模系统的设计与实现。这章介绍了基于 Blendshape 和基于骨骼驱动的人体建模的原理和不同之处，并基于这些原理实现了自定义人脸建模，表情建模以及体型建模并应用于本文的数字人。

第三章：多模态信号驱动手势动作生成模块的设计与实现。这章介绍了基于多模态信号驱动生成手势与动作的系统，同时介绍本系统采用的 CaMN 网络中的 5 个编码器和 2 个解码器及其损失函数，基于这个系统实现了多模态信号驱动手势动作生成并落地相关技术。

第四章：动作模块的设计与实现。这章介绍了数字人的动作系统原理及其在 Unity 引擎中的应用。通过结合多模态信号驱动手势动作生成模块，解决了数字人在 Unity 引擎中应用此模块遇到的问题。

第五章：数字人系统综合与测试。第五章介绍了在 Unity 引擎中，如何将人体建模模块、多模态信号驱动手势模块和动作模块相结合，以构建一个更完备的数字人系统，并对这个综合的系统进行了测试。此外，还会提及需要完成这个综合数字人系统所需要的其他部分的技术。

第六章：总结与展望。第六章总结了全文的研究内容，以及介绍了未来会进行的改进与优化。

第二章 三维人体建模系统的设计与实现

本章介绍了一个基于Blendshape和骨骼驱动的三维人体建模系统的设计和实现方案。该系统可以对人体的面部、表情和体型进行建模，以实现数字人形象的自定义功能。第一部分介绍基于Blendshape和骨骼驱动的两种不同的人体建模方法，并对它们进行了比较和分析。第二部分给出实验结果，即根据上述原理方法，介绍在Unity中实现的人脸建模、表情建模和体型建模功能。

2.1 基于 Blendshape 与骨骼驱动的人体建模

2.1.1 基于 Blendshape 的人体建模方法

1、Blendshape 简介

Blendshape在计算机图形学中是一种常用的技术，也被称为形状融合。具体来说，Blendshape是在两个网格（Mesh）之间做插值运算，即从一个网格到另一个网格的融合变形。例如在左嘴角从放松到上扬的变换中，左嘴角放松可以作为基形状，左嘴角上扬可以作为变形目标，它们之间就可以实现融合变形。可以通过制作大量的这些Blendshape来实现数字人体建模中更加精细和可控的表情、形体和外貌等细节。

Blendshape作为数字人物建模技术，常常用于面部表情动画中。例如在表情建模的应用中，它基于一系列预定义的面部表情，每个表情由一组顶点的位移量来表示。通过对这些表情进行线性组合，可以生成几乎无限数量的面部表情，使得面部动画更加丰富、生动。

Blendshape的计算公式可以表示为：

$$jE = \sum_{i=1}^{n} w_i \cdot F_i + F_{neutral} \qquad \text{(公式2-1)}[1]$$

其中，jE 表示任意面部表情，F_i 表示第 i 个表情基（Blendshape），$F_{neutral}$ 表示基准表情，w_i 表示第 i 个表情基的权重，n 表示表情基的数量。该公式将每个表情基与其相应的权重相乘，然后将结果相加，得到最终的面部表情。

2、基于Blendshape的人体建模方法

由上述原理可知，可以使用基于顶点的Blendshape方法实现人体建模。

需要注意的是，基本Blendshape只需制作出表情变化的最小单位即可。例如，微笑是一个调用众多肌肉的表情，涉及的部位通常包括眼睛、眉毛、鼻子、脸颊

和嘴等部位。再比如嘴的运动，又可以分为上唇、下唇、嘴角、牙齿、舌头等的运动。而其中的上唇，又包含了上嘴唇中部、左部和右部。其中每一个部位的运动方式又是多样的，只需制作其中最基本的Blendshape即可。

在具体实现时，可以通过使用PCA（Principal Component Analysis）算法对不同的Blendshape（如面部表情）进行降维处理，然后通过权重调整来控制变形[2]。此外，还可以通过使用神经网络和深度学习算法来实现更加准确和逼真的最小单位Blendshape生成。

Blendshape的权重即从初始基形状到目标变形（即另一个拓扑结构一样，但顶点的transform不同的网格）的程度。

2.1.2 基于骨骼驱动的人体建模方法

在捏人系统中，使用Blendshape需要大量的工作量。因此，本文探索了基于骨骼驱动的方法，该方法在制作量和性能消耗方面相对较低。

基于骨骼驱动的人体建模方法的核心是改变部分骨骼的变换矩阵。例如，在人脸建模系统中，需要控制面部的关键部位，比如眼、耳、口、鼻、下巴等部位。但是实际的生物学上控制人脸变形的结构更加复杂。因此需要将真实世界中人脸的结构与虚拟重建世界中的骨骼进行映射，以达到较为真实的效果。

本文使用了骨骼变换和线性蒙皮（Linear Blending Skinning, LBS）原理实现了该模块。

1、骨骼变换

每个骨骼都可以通过矩阵变换（如旋转、平移、缩放）来控制模型的姿态。右手坐标系下骨骼变换可以使用以下公式表示：

$$T_{total} = T_{translate} \times T_{rotate} \times T_{scale} \quad \text{(公式2-2)}$$

其中，$T_{translate}$是平移矩阵，T_{rotate}是旋转矩阵，T_{scale} 是缩放矩阵，T_{total}是骨骼的总变换矩阵。

2、线性蒙皮

线性蒙皮即LBS，它通过将网格顶点与一组骨骼相关联来实现。当骨骼移动时，相应的顶点也会被移动，并且每个顶点都会根据其在各个骨骼周围的位置分配一个权重值，以便更精确地跟随骨骼的动作。而顶点和骨骼的关系，如果要在三维建模软件中手动制作，则会进行刷权重操作来赋值，即对应这个原理反向操

作：选中一根骨骼，对这根骨骼运动会带动的顶点赋予不同的权重，代表骨骼运动对不同顶点的影响轻重。将生成的顶点权重图与骨骼的变换一起计算出最终的顶点位置[3]。

在LBS算法中，可以使用骨骼的变换矩阵来计算当前骨骼位置与基准位置之间的相对位置，并结合顶点相对于骨骼的权值信息来计算受骨骼影响后的顶点位置。

具体公式如下：

$$L_i = \sum_{j=1}^{|bonesN|} w_{ij} \left(R_j S_j L0_i + T_j\right) \qquad \text{（公式2-3）}[4]$$

其中，顶点i的最终位置为L_i ,w_{ij}是第i个顶点受到第j个骨骼影响的权重，bonesN是骨骼个数。T_j是骨骼j的平移矩阵，即公式2-2中的$T_{translate}$，R_j骨骼j的旋转矩阵，即公式2-2中的T_{rotate}，S_j骨骼j的缩放矩阵，即公式2-2中的T_{scale}。$L0_i$是顶点i的初始位置坐标[5]。

2.1.3 Blendshape 与骨骼驱动方案对比

Blendshape能够很精准地控制表情或者人脸的建模，且在引擎中方便控制。但是Blendshape的缺点在于计算量相比于骨骼驱动大许多。通常一个模型中最多可能会有几百个骨骼，而顶点数量可能达到数千甚至更多。因此，相较于骨骼控制，Blendshape的计算量更大。

举个例子，假设游戏中有100个不同的表情。使用基于Blendshape的方法，那么每一个表情都需要存储所有面部顶点的位置信息，合理假设有1000个顶点。如果使用基于骨骼驱动的方法，即将表情分解为骨骼驱动的蒙皮动画，那么假设有100个骨骼。则使用基于Blendshape的方法总共需要存储10^5个顶点的位置信息，而基于骨骼驱动的方法只需要存储一张顶点权重图和10^4个骨骼的位置信息，就可以实现这些表情了。可以说，基于骨骼驱动对比基于Blendshape既能够有效地减少存储空间的使用，而且还可以提高动画播放的效率。

骨骼驱动也有明显的缺点：骨骼权重的分配有过多的限制，人体建模的细节很难像基于Mesh顶点的Blendshape那样收放自如。Blendshape更像所见即所得，制作的变形目标是什么，这个部位就固定是这么融合计算的，而骨骼的蒙皮有着Rotation、Position和Scale三种变形矩阵，例如要达到某种表情效果时，需要对其

做大量测试与计算。

2.2 实验结果与分析

2.2.1 人脸建模功能实现

这里分别展示基于Blendshape和基于骨骼驱动的人脸建模功能的实现。

1、基于Blendshape：

本文一共用51个Blendshape作为基来控制人脸建模。包括以下几个部位：脸型、眼型、嘴型、脖型、眉型、鼻型、耳型、脸颊和下巴，涵盖了人脸的几乎所有部位。每一个部位提供多个Blendshape可供用户捏脸，基本上可以实现用户自定义出自己想要的人脸。

这里只展示了Blendshape一个方向，但实际上本研究也实现了逆向。即嘴可以突出，也可以凹进去，只需通过调整Blendshape权重即可完成，而Unity的UI上用户只需要通过拉滚动条来调整正向或者是逆向操作。

图2-1展示了用于人脸建模的部分Blendshape，包括头型，眼型，嘴型，脸型，耳型等。第一个子图为初始的人脸建模基准。

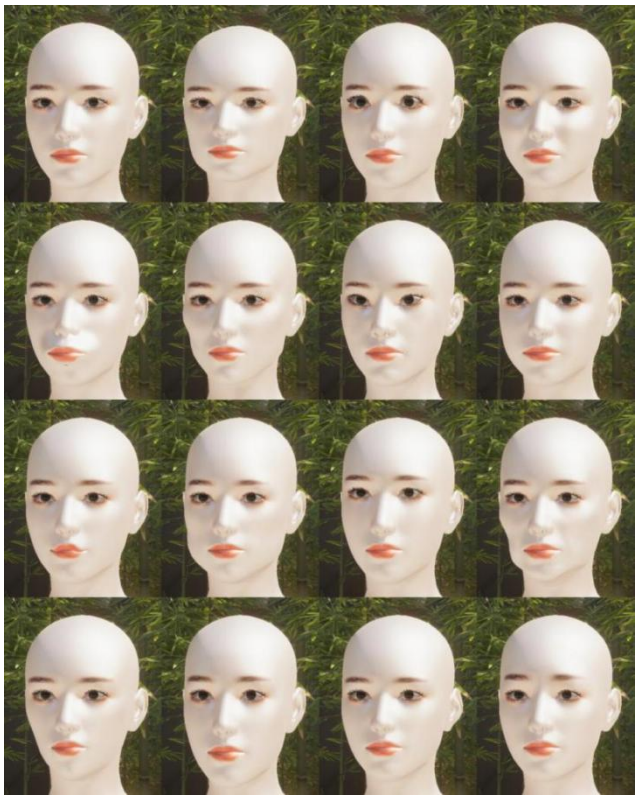

图2-1 用于人脸建模的部分Blendshape

用户可以通过本项目在Unity上设置的UI滚动条来自定义设置人脸，组合上述所说的51个Blendshape进行人脸建模。例如在图2-2中用本项目实现的数字人系统进行了捏脸。左边是用户可以操作的UI滑动条，在捏脸部分一共有51个提供给用户编辑。

图2-2 捏脸效果与对应UI

图2-3是一些捏脸后的图与原图对比。第一个子图是Blendshape基准模型。

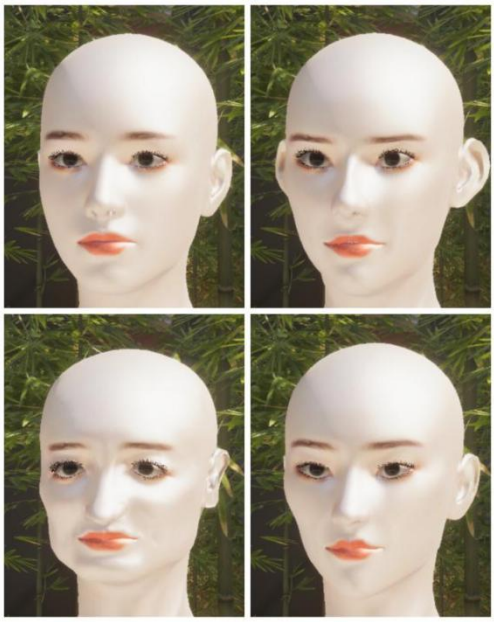

图2-3 捏脸效果

2、基于骨骼驱动：

本研究的数字人一共有251根骨骼来控制整个人体建模，其中159根骨骼用于人脸包括表情的建模，可见人脸骨骼的复杂之处。这159根骨骼同样能够覆盖到Blendshape所覆盖的几乎所有部位。

图2-4中，251根骨骼被动态获取并显示在屏幕上，点击某个骨骼右边会出现这根骨骼的缩放和位置滚动条（因为旋转值一般不用于人脸建模）。拉动滚动条，数字人的骨骼对应的蒙皮顶点会随之改变。可以看到，直接对骨骼进行调整，效果并不如Blendshape好。本项目的改进方案是，给用户提供如Blendshape所呈现的直观形变滚动条。然而，单个相应形变背后可能需要多根骨骼作用，因此需要预先进行判断和定义。

图2-4 使用骨骼建模实现的捏脸效果

2.2.2 表情建模功能实现

这里分别展示基于Blendshape和基于骨骼驱动的表情建模功能实现。由于嘴部也是人脸的一部分，所以这里提到的表情建模也包含部分口型的建模。表情建模与人脸建模的不同之处在于，人脸建模即捏脸，创造新的形象，表情建模是在固定形象上做表情与口型的变化。

1、基于Blendshape：

首先，图2-5展示了初始的Blendshape表情基准模型，即标准放松姿态模型，这个模型上没有任何表情：

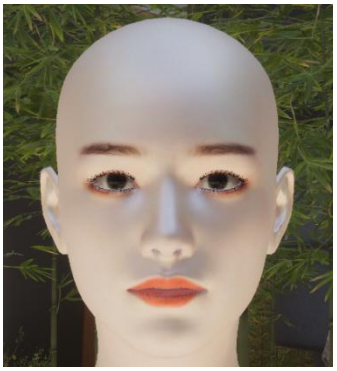

图2-5 初始的Blendshape表情基

本项目一共使用64个Blendshape作为基来控制表情建模。包括嘴角的各种牵动，嘴的开合，鼻子的抽动，脸颊的鼓起与吸气，下巴的移动，眉毛里外的移动等等，基本能覆盖所有的脸部移动，也基本能够做出所有表情。

在基于Blendshape的表情建模中，图2-6（a）展示了部分Blendshape所控制的口部形状。对于与口型驱动相关的任务来说，这些Blendshape至关重要。其中图2-6（a）中的第一个子图是表情基。有一些Blendshape的变化非常细节，但是在真实世界中，人们的各个表情也是由一些细微的肌肉拉伸变化实现的。从图2-6（a）中可以看到，很多的基本Blendshape是需要成对制作的，比如左嘴角上扬和右嘴角上扬。

图2-6（b）展示了部分眼睛部位的Blendshape。在图中使用一个Blendshape来表示成对的眼睛动作，例如闭合左眼和闭合右眼中的闭合左眼动作。其中第一个子图为初始的基准表情，即图2-5，用于对比。

图2-6（c）展示其他部位的部分Blendshape，包括眉毛，脸颊等。

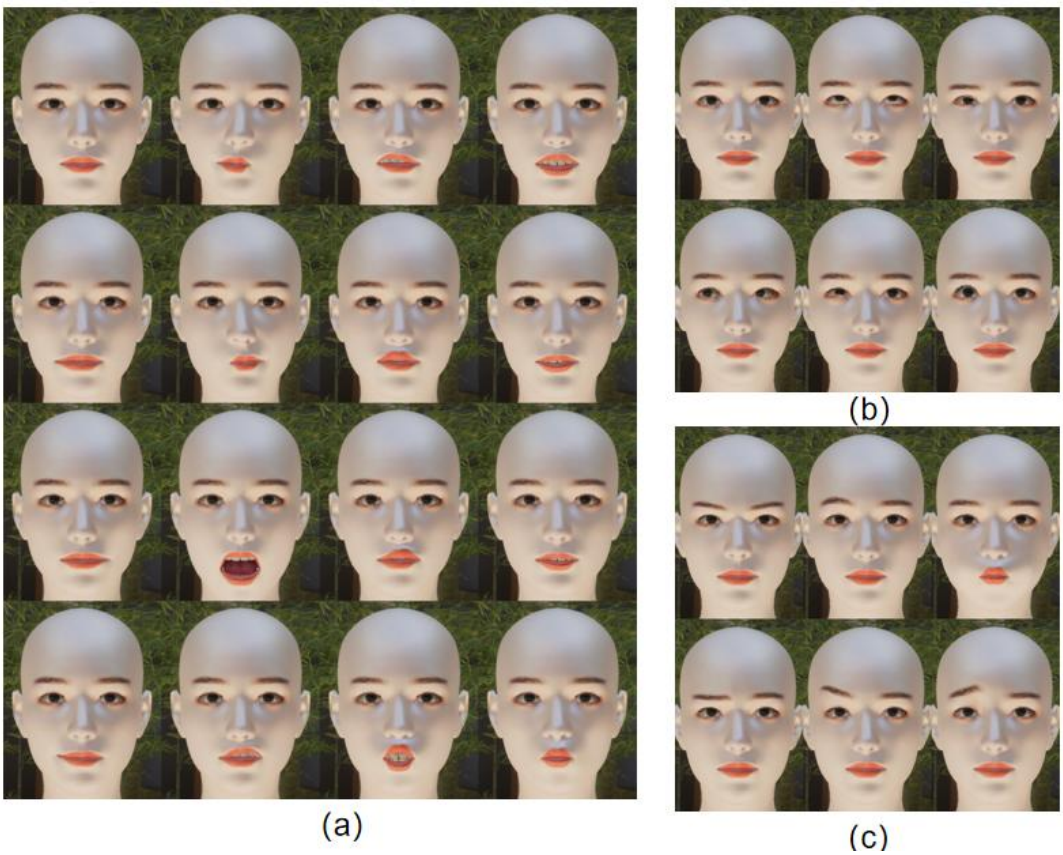

图2-6 表情建模的部分Blendshape效果

下图2-7展示了使用这些基本Blendshape进行组合，以实现一些具体的表情。其中每一个都是由众多的基本Blendshape组合而成。

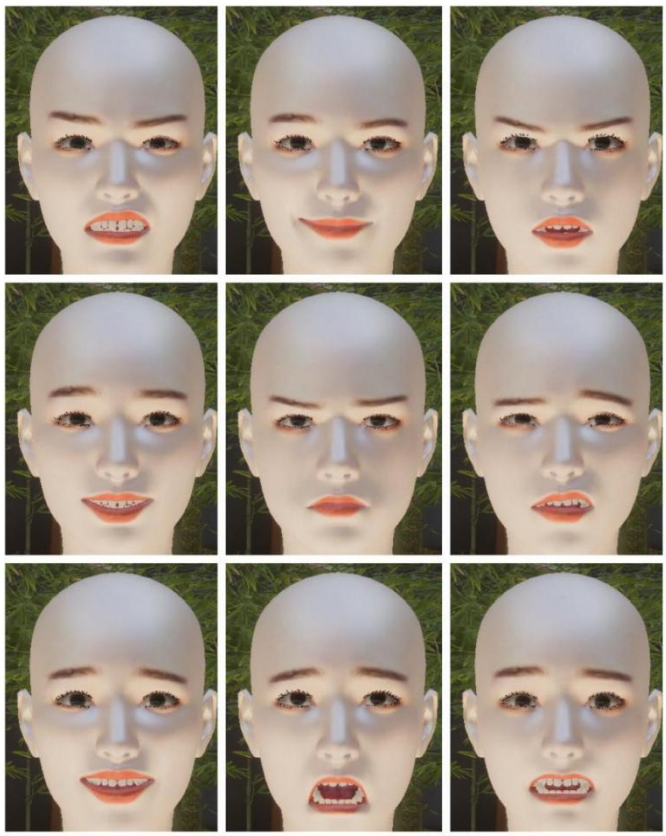

图2-7 使用基本Blendshape组合成不同表情

以上数字人表情都是由众多基本Blendshape组成，举个例子：例如图2-7中的子图1，就包含了：上唇左侧上下，上唇右侧上下，上唇进出，不张开牙齿的张嘴唇等等很多与嘴相关的Blendshape，还有眼睑上侧下侧、脸颊左侧右侧、左右侧眉毛的内外侧等等一共是29个Blendshape的融合。

2、基于骨骼驱动：

这部分用到的骨骼与上一部分人脸建模功能实现用到的骨骼一致，不过多介绍了。

图2-8使用骨骼驱动的表情建模，图2-8为愤怒表情，Unity的UI与人脸建模的骨骼驱动部分一样。

图2-8 使用骨骼驱动合成愤怒表情

2.2.3 体型建模功能实现

这里分别展示基于Blendshape和基于骨骼驱动的体型建模功能的实现。

1、基于Blendshape：

本项目通过Blendshape来分别实现自定义人体体型的肌肉含量，即健美程度、人体梨型身材程度和肥胖或瘦弱程度等等。

图2-9表示了控制梨形身材程度的Blendshape在不同权重下对人体体型的建模结果。

图2-9 控制梨形身材程度的Blendshape

图2-10表示了控制肌肉含量的Blendshape在不同权重下对人体体型的建模结果。

图2-10 控制肌肉含量的Blendshape

除了前述功能外，本研究基于Blendshape的体型建模还实现了对腹肌、皱纹等皮肤质量细节的控制。人体建模Blendshape之间本身也能叠加使用，人的体型建模外加上面的人脸建模，可以得到类似图2-11的拥有特殊外貌和体型的独一无二的数字人。

图2-11 人体建模和人脸建模的结合

2、基于骨骼驱动：

关于体型的身体单个部位建模，主要是通过骨骼来驱动。本研究中的数字人一共有251根骨骼来控制整个人体建模，除去人脸骨骼，共有92根骨骼用于人体建模。具体部位包括各个关节，如手部关节，脚部关节，腿部关节，胸部关节，肩部关节等等，基本涵盖了所有可能的人体部位。

图2-12中展示了基于骨骼进行体型建模，分别是小腿，肚子和胸部。可以看到，骨骼驱动可以很方便调整人体的各个部位。不过骨骼驱动对于细节的处理仍然很难比得上Blendshape，比如对肌肉、皮肤皱纹的模拟等。

图2-12 基于骨骼的体型建模

2.2.4 Unity 中的实现细节

1、基于Blendshape：

在Unity中，需要同时控制人体、眉毛和人物各个服装的Blendshape。例如需要保证在类似闭眼睁眼等场景下人脸和眉毛的Blendshape会一起更改，或者更改体型时人物服装也会随着体型更改。

在Unity的UI呈现上，本项目通过获取所有的skinnedMeshRenderer下的sharedMesh中之前制作了的各个Blendshape的名称，将其Blendshape权重分别对应到Unity的滑动条权重上，实现表情建模的功能。

需要注意的是，不同数字人会对应制作不同数量的Blendshape，因此需动态获取数字人的Blendshape并在UI上给用户提供自定义的接口。

具体的代码时序可以参考第五章图5-2 Unity中Blendshape人体建模功能时序图。

2、基于骨骼驱动：

在Unity中，动态获取数字人所有的骨骼，包括面部骨骼，动态生成带有所有所需骨骼名称的button，对每个button启用监听事件，然后将其transform的属性对应到UI的滑动条上，这样就可以根据自己的需要使用骨骼来进行人体重建。

在实验过程中，可能会遇到穿模问题，本文的解决方案是对每根骨骼的变换范围做一个限定，以防止穿模问题的发生。

第三章 多模态信号驱动手势动作生成模块的设计与实现

数字人常常会以虚拟主播、虚拟偶像、品牌代言人等形象出现，这些形象不仅需要有特定的外貌，也需要大量处于对话或演说的场景之中。当演说时，其身体特别是手势的动作，也常常是富有特点，带有情感的动作。而这些动作的动画资源获取方式费时费力，只能找到千篇一律的手势动作，或者花费大量时间制作适配于这个情景的动作，又或者使用开销较大的动捕。具体来说，传统的手势动作生成方法通常是基于关键帧的方式，这样不仅需要大量的人工工作，还难以实现细腻的手势运动。又或者，采用动捕的方式需要付出高昂的人力成本和设备成本。

因此需要一种方法，使得数字人能够在不同场景和对话中自适应地生成手势动作。本文使用了一种基于 CaMN 网络与 BEAT 数据集的多模态驱动手势动作生成方法[6]，通过学习多模态数据集中的语义和情感信息，自动地生成流畅、自然的手势动作，从而提高数字人的表现力和交互性。

本研究的手势动作生成模块是基于多模态数据的，其中涉及到了语音、语言和情感等多个模态。为了获取充足的多模态数据并提升生成效果，本文使用 BEAT 数据集作为网络输入来驱动生成数字人的手势动作，最终实现了一个具有较高生成质量的多模态驱动手势动作生成模块[6]。

在本章中会介绍网络总体架构，以及各个编码器与解码器结构，还有损失函数，来探究手势动作是如何从多模态数据中生成的。最后展示在 Unity 中这个模块应用在数字人身上的效果。

3.1 基于 CaMN 网络与 BEAT 数据集的多模态驱动手势动作生成

CaMN（Cascaded Motion Network）网络是一种多模态级联网络，可以将多个输入模态的信息融合在一起，提高模型的表现力。BEAT 数据集是一个大规模的语义和情感多模态数据集[6]。本文使用 BEAT 数据集中的文本，演讲者信息、情感标签、语音、面部表情信息作为网络的输入，使用 bvh 格式作为手势动作的输出，并在 Unity 中应用于本研究的数字人身上。

3.1.1 网络结构与代码结构

本文的手势动作生成模块基于编码器-解码器结构，采用了级联网络，包含

文本编码器，演讲者 ID 编码器，情感编码器，音频编码器，面部表情编码器这五个编码器，解码器包含身体解码器和手势解码器。

图 3-1 是 CaMN 网络结构示意图，但是有部分做了简化，例如解码器其实包含了两个（身体和手势的解码器是分开的），但图 3-1 为了不过于复杂，只显示一个解码器。

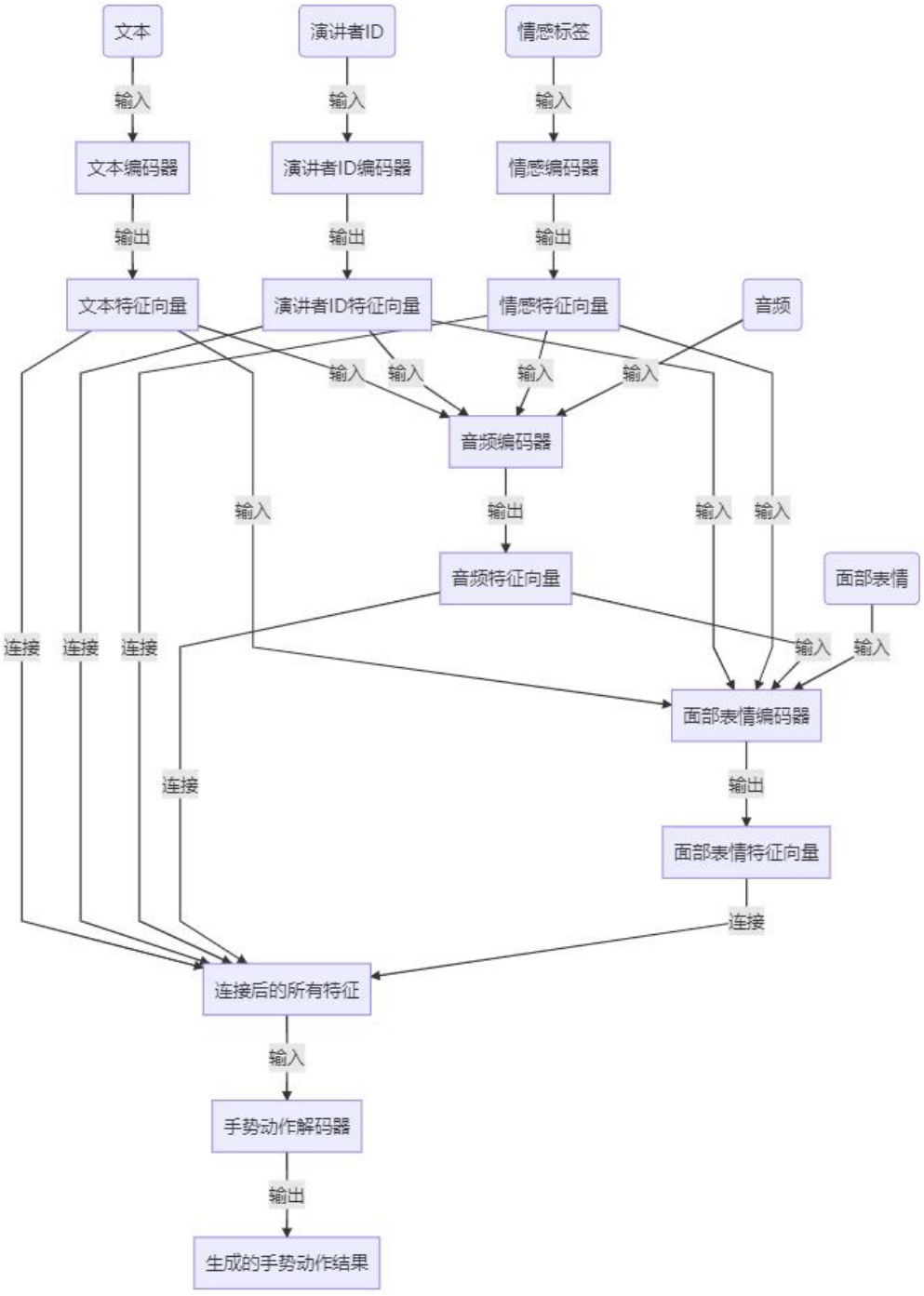

图 3-1 CaMN 网络结构示意图

多模态信号驱动手势动作生成的网络模块由 python 实现，图 3-2 是其 uml 时序图（其中，CaMN 类继承于 PoseGenerator 类）。

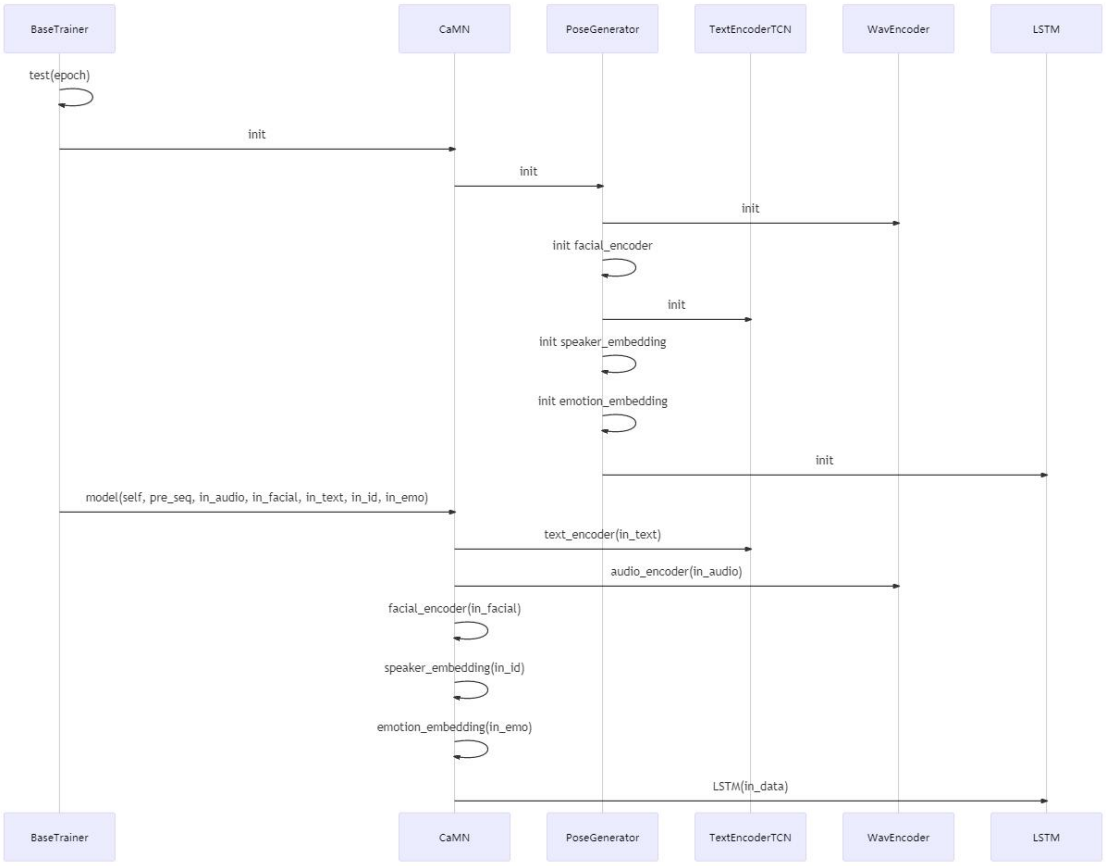

图3-2 CaMN网络实现的时序图

3.1.2 编码器

1、文本编码器

文本编码器的输入是来自于将演讲者的语音通过自动语音识别器(ASR)转为文本，并进行了人工的文本校对，最终形成的字符串列表[7]。

第一步是将文本数据中的词转换为词嵌入集，具体实现是使用预训练的 FastText 模型[8]，将每个单词映射为一个低维度的词向量表示。这一步的目的是减少特征维度，简化模型复杂度，提高训练效率。

第二步是对文本序列进行编码，采用自定义的编码器E_{text}进行实现，该编码器包含一个 8 层的 TCN 网络以及跳过连接(skip connections)。为了提取当前帧的

特征，通过 TCN 对时间维度进行卷积操作，并分析提取前 17 帧和后 17 帧的信息[9]。之后使用跳过连接将信息从较浅的层直接传递到较深的层，有助于缓解梯度消失的问题[10]。网络中一共包含 4 个 TemporalBlock 块，每块一共有 2 个卷积层，组成 8 层 TCN 网络。此外，还包含了一些 ReLU 激活函数和 dropout 操作[11]。公式如下：

$$p_i^{text} = E_{text}(e_{i-17}^{text}, e_{i-16}^{text}, e_{i-15}^{text}, e_{i-14}^{text}, ..., e_i^{text}, ..., e_{i+16}^{text}, e_{i+17}^{text})$$（公式3-1）

公式 3-1 中的E_{text}代表文本编码器，e_{i-f}^{text}（e_{i+f}^{text}）代表的是 i 的前 f 帧（后 f 帧）的包含文本信息的向量。p_i^{text}代表文本编码器最终输出的文本特征向量。

2、演讲者 ID 编码器

演讲者 ID 编码器的输入是一个 one-hot 向量，代表演讲者的 ID。

由于演讲者 ID 在同一个视频的各个帧不会改变，因此仅使用当前帧的演讲者 ID 计算其潜在特征。

这个模型采用了嵌入层（Embedding）作为演讲者 ID 编码器，线性层进行线性变换以及 LeakyReLU 作为激活函数。其中，嵌入层将每个演讲者 ID 映射为一个向量，然后经过线性层和激活函数层进行处理。最终将其与其他输入数据进行拼接，参与后续的级联网络，以增强神经网络的表达能力。

这个编码器定义了一个能够将演讲者 ID 编码转换为固定长度向量表示的神经网络模型，为后续对话手势合成任务提供了演讲者 ID 的上下文信息，从而可以更好地反映对话中不同演讲者之间的个性以及情感和语境。

$$p_i^{speakerID} = E_{speakerID}(e_i^{speakerID})$$（公式3-2）

公式 3-2 中的$E_{\text{speakerID}}$代表演讲者 ID 编码器，$e_i^{speakerID}$是代表演讲者 ID 信息的 one-hot 向量。$p_i^{speakerID}$代表演讲者 ID 编码器最终输出的演讲者 ID 特征向量。

3、情感编码器

情感编码器$E_{emotion}$的输入是一个包含情感标签的 one-hot 编码向量。

$E_{emotion}$由四层 TCN 组成，每层 TCN 的基本结构都由卷积层、批量归一化层 (Batch Normalization, BN)和 LeakyReLU 作为的激活函数组成。

每一层 TCN 的 BN 层会进行归一化操作，LeakyReLU 作为激活函数进行非

线性变换，卷积层输出情感特征向量。这个过程可以提取出输入情感的高级特征，提取情感随着时间的变化。

$$p_i^{emotion} = E_{emotion}(e_{i-f}^{emotion}, ..., e_{i+f}^{emotion}) \quad （公式3-3）$$

公式 3-3 中的$E_{emotion}$代表情感编码器，$e_{i-f}^{emotion}$（$e_{i+f}^{emotion}$）是第 i 帧的前 f 帧（后 f 帧）的包含代表情感信息的向量。$p_i^{emotion}$代表第 i 帧情感编码器最终输出的情感特征向量。

4、音频编码器

音频文件首先会被预处理程序下采样为 16KHZ，同时设置帧速率为 15fps。

音频编码器不仅接收预处理后的音频信息，还会将文本特征，演讲者 ID 特征和情感特征与音频信息拼接并处理，形成级联网络结构，最终得到更合适的音频特征。

音频编码器包含了一系列卷积层，第一个卷积层使用了较大的卷积核和较大的膨胀率（即卷积核中像素之间的距离）来捕捉长期的时间依赖关系，以提取较宽的频带信息。接下来的卷积层逐渐缩小卷积核和膨胀率，以提取更局部的特征[12]。最终，经过一系列的卷积操作，将音频数据转换为时域上的特征表示。总共是由 12 层 TCN 和 2 层 MLP 组成[13]。

$$p_i^{audio} = E_{audio}(e_{i-f}^{audio}, ..., e_{i+f}^{audio}; p_i^{text}; p_i^{emotion}; p_i^{speakerID}) \quad （公式3-4）$$

公式 3-4 中的E_{audio}代音频编码器，e_{i-f}^{audio}（e_{i+f}^{audio}）是第 i 帧的前 f 帧（后 f 帧）的包含代表音频信息的向量。p_i^{audio}代表第 i 帧音频编码器最终输出的音频特征向量。p_i^{text}、$p_i^{emotion}$、$p_i^{speakerID}$ 是前面提到的文本、情感、演讲者 ID 编码器的特征向量，在音频编码器中将他们与音频信息拼接一起来提取音频特征[14]。

5、面部表情编码器

面部表情的输入使用苹果 ARKit 标准，表示为由 52 个 Blendshape 权重构成的向量。

面部表情编码器与音频编码器类似，会串联所有之前的特征，以提取更合适的面部表情特征。

面部表情编码器是一个包含四个基本块（BasicBlock）的序列模型。这些基

本块具有不同的输入维度、输出维度、卷积核大小、步幅、扩张率等参数。在实现中，第一个基本块具有更大的扩张率和下采样，而其他三个基本块具有较小的扩张率和下采样。这些基本块包含两个卷积层和一个可选的下采样层，这些层通过残差连接来连接在一起，以使网络能够更好地学习特征。面部表情编码器总共包含 8 层 TCN，以及 2 层 MLP。

$$p_i^{facial} = E_{facial}(e_{i-f}^{facial}, ..., e_{i+f}^{facial}; p_i^{text}; p_i^{emotion}; p_i^{speakerID}; p_i^{audio}) \quad \text{（公式3-5）}$$

公式 3-5 中的E_{facial}代面部表情编码器。p_i^{facial}代表第 i 帧面部表情编码器最终输出的面部表情特征向量。p_i^{text}、$p_i^{emotion}$、$p_i^{speakerID}$、p_i^{audio}是前面提到的其他各个编码器的特征向量，在面部表情编码器中将他们与面部信息拼接一起来提取面部表情特征。

最后，以上所有特征会拼接起来与预处理的种子姿势序列进行连接，再通过 LSTM 解码器进行处理，得到最终的输出姿势序列。

3.1.3 解码器

解码器可以分为身体解码器和手势解码器。这两个解码器被分开是因为为了更好地生成手势，需要利用身体动作作为手势解码器的一部分输入[15]。

上述编码器最后融合后的包含了音频、演讲者 ID、情感、文本和面部表情特征的向量和前一时刻的姿势特征一起输入到 LSTM 中进行训练，得到手势动作的预测结果。即将五种模态的特征和之前的手势动作（即种子姿势）结合起来，来合成潜在手势动作。

身体解码器和手势解码器都是使用 LSTM 结构进行潜在特征提取，并使用 2 层 MLP 进行动作重建。

身体的特征向量p_i^{body}公式如下：

$$p^{body} = D_{body}(p_0^{audio} \otimes p_0^{text} \otimes p_0^{emotion} \otimes p_0^{speakerID} \otimes p_0^{audio} \otimes p_0^{facial} \otimes e_0^{body} \otimes e_0^{hand}, ..., p_n^{audio} \otimes p_n^{text} \otimes p_n^{emotion} \otimes p_n^{speakerID} \otimes p_n^{audio} \otimes p_n^{facial} \otimes e_n^{body} \otimes e_n^{hand}) \quad \text{（公式3-6）}$$

手势的特征向量p_i^{hand}公式如下：

$$p^{hand} = (p_0^{audio} \otimes p_0^{text} \otimes p_0^{emotion} \otimes p_0^{speakerID} \otimes p_0^{audio} \otimes p_0^{facial} \otimes e_0^{body} \otimes e_0^{hand} \quad D_{hand} \quad , \ldots, \quad p_n^{audio} \otimes p_n^{text} \otimes p_n^{emotion} \otimes p_n^{speakerID} \otimes p_n^{audio} \otimes p_n^{facial} \otimes e_n^{body} \otimes e_n^{hand}; \ p^{body})$$

（公式3-7）

公式 3-8 和公式 3-9 表示的是提取完特征之后，使用 MLP 进行手势和动作的重建。

$$\hat{e}^{Body} = MLP_{Body}(p^{body}) \quad （公式3-8）$$

$$\hat{e}^{Hand} = MLP_{Hand}(p^{hand}) \quad （公式3-9）$$

3.1.4 损失函数

网络的损失函数主要有 2 个，分别是手势动作重建损失函数以及对抗损失函数。

1、手势动作重建损失函数

$$L_{L_1}(Body) = E[\| e^{Body} - \hat{e}^{Body} \|_1] \quad (公式3-10)$$

$$L_{L_1}(Hand) = E[\| e^{Hand} - \hat{e}^{Hand} \|_1] \quad (公式3-11)$$

$$L_{L_1} = L_{L_1}(Body) + \gamma L_{L_1}(Hand) \quad (公式3-12)$$

公式 3-10 是生成的身体动作与源身体动作在空间上的距离，公式 3-11 是生成的手势动作与源手势动作在空间上的距离，公式 3-12 是总的手势动作重建损失函数，γ决定手势损失的权重。

使用 L1 损失来衡量生成的手势动作与源手势动作之间的空间差异程度，以此来鼓励网络生成与源手势动作尽可能相似的手势动作。

2、对抗损失函数

$$L_D^{GAN} = -E[\log(D(\hat{e}^{Body}; \hat{e}^{Hand}))] \quad (公式3-13)$$

公式 3-13 中的 D 表示生成对抗模型中的判别器，此公式表示手势和动作的对抗损失函数。

该对抗损失是通过训练一个判别器模型，来鼓励生成的手势动作越来越难以被判别器区分出来，从而提高前面的级联网络构成的生成器模型的生成能力和生成动作的真实感[16]。

3、总损失函数

$$L = \alpha_0 L_D^{GAN} + \beta\alpha_1 L_{L_1} \quad \text{(公式3-14)}$$

α_0与α_1分别代表二损失函数的预定义权重。

其中β也能一定程度决定手势动作重建的权重，其值越小，表示语义相关性可能越低，与情感的关联会更小，同时对抗损失更大，更不容易被察觉是生成动作而非真人动作。

这两种损失函数相互协作，共同优化网络的训练过程，提高生成手势动作的质量和逼真度。

3.2 实验结果与分析

本项目在 Unity 中应用了这个模块，点击播放对应测试用例的同时，Unity 会开始播放对应的测试音频，且数字人会自然地做出相应手势动作。这个手势动作会有对应的情感和节奏变化。

同一段语音，演讲者不同，生成的手势动作也会有不同的风格。

经过调研，大部分人已经无法区分这里的手势动作是由神经网络生成的还是由动捕这类从真人身上获取的手势动作，可见此时生成出来的手势和动作已经非常自然和真实。

此处用到的测试输入数据为 BEAT 数据集提供的部分数据，包括文本信息，情感标签信息、演讲者信息、音频信息和面部表情信息。

输出数据为CaMN 神经网络自动生成的手势和动作的动画文件，格式为bvh。放入 bvh 解析器中读取其动作。图 3-3 表示的是输入数据中演讲者 ID 不同，输出生成不同动作的截图。图中共有 a、b、c、d 分别是输入 4 个不同的演讲者 ID，生成不同风格的个性化手势动作。可以看到，不同演讲者有着各不相同的演讲风格，比如 a、b 这 2 位演讲者为男性，生成站姿就与下面的 c、d 两位女性演讲者不同；同时 b 演讲者生成的动作幅度较 a、c、d 演讲者较大；d 演讲者重心放在右腿，而 a、b、c 演讲者重心更多放在左腿。

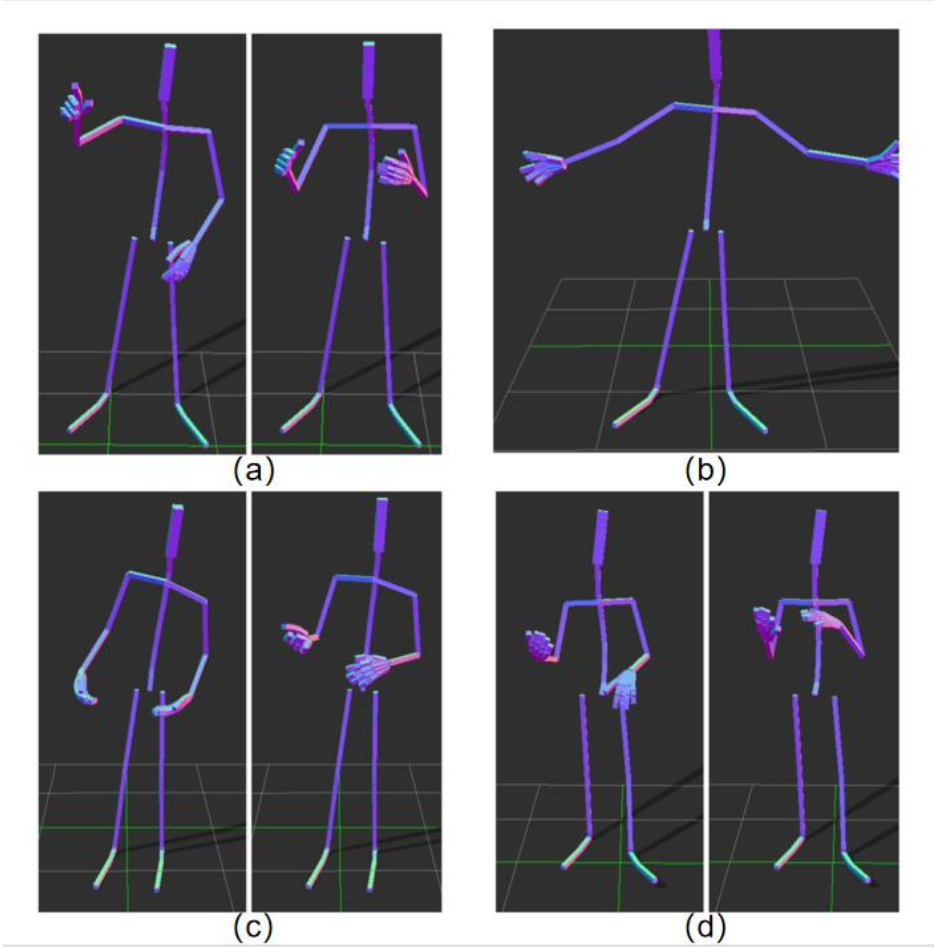

图3-3 输入不同演讲者ID生成不同风格的个性化手势动作

当然，还需要将多模态驱动手势动作应用到本项目的数字人身上，以下是在Unity中的效果，具体应用于数字人模型后出现的问题与解决方案在第四章中会提到。

图 3-4 应用于 Unity 数字人的多模态驱动手势动作生成

第四章 动作模块的设计与实现

在将多模态数据生成的手势动作应用于数字人时，需要考虑数字人执行动作时所涉及的各种细节。为了实现这些细节，需要设计一种有效的动作模块。

在动作模块中，通常会使用反向动力学（IK）和蒙皮技术来模拟数字人的动作。反向动力学是一种能够根据目标末端效应器的位置计算出关节角度的技术，它有助于模拟数字人的运动。蒙皮技术是一种将数字人的表面网格与骨架进行绑定的技术，它可以让数字人的外观与骨架相匹配，并且可以让数字人的运动更加自然。

动作模块是数字人系统中非常重要的一个部分，它有助于实现数字人的运动行为，并且可以让数字人看起来更加自然和真实。

4.1 数字人动作模块设计

4.1.1 基于 LBS 的动画系统设计

本节介绍基于线性蒙皮(LBS)的动画系统。该系统能够处理数字人的动作数据，通过权重与骨骼相结合，来实现数字人的动画效果。

1、线性蒙皮

在第一章的第二小节中已经提到过LBS的原理及其公式等，LBS不仅仅是基于骨骼驱动的人体建模的原理，同时也是动作系统的原理。LBS会根据骨骼变换矩阵，得到人体皮肤等的顶点变换，实现相应动画效果。

2、动画骨骼

在本系统中，使用LBS来进行动画蒙皮，同时尝试使用两套骨骼进行动画骨骼控制，一套是Unity统一的humanoid骨骼，一套是为数字人定制的专门骨骼。通过将每个骨骼与相应的权重结合，可以计算出每个顶点受到骨骼影响的程度。这些权重以及骨骼的位置将用于计算动画蒙皮的结果。

3、动画插值

为了实现更加流畅的动画效果，采用了动画插值技术。在运行时，动画系统会根据用户的输入动态地混合多个动画。这种方法能够实现更加自然的过渡，从而提高动画的质量和真实感。

在Unity中，动画的关键帧插值采用三次多项式插值。其中可调整的参数设

为inT(inTangent，左侧的斜率)，outT(outTangent，右侧的斜率)和时间t。同时也对应公式4-1的参数，y是t时刻动画的插值结果。

$$y = f(t) = at^3 + bt^2 + ct + d \quad \text{(公式4-1)}$$

代入t_0、t_1时刻的y及其导数，可以得到a、b、c、d的矩阵表示，见公式4-2。

$$\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} t_0^3 & t_0^2 & t_0 & 1 \\ t_1^3 & t_1^2 & t_1 & 1 \\ 3t_0^2 & 2t_0 & 1 & 0 \\ 3t_1^2 & 2t_1 & 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} y_0 \\ y_1 \\ outT_0 \\ intT_1 \end{bmatrix} \quad \text{(公式4-2)}$$

实验结果表明，Unity的动画系统在数字人动画的制作中具有良好的表现和应用前景[17]。

4.1.2 基于 IK 的角色动画校正

1、反向动力学

反向动力学（IK，Inverse Kinematics）的原理是根据角色的目标点位置来自动计算每个关节的旋转和位移，从而实现更加精细、自然的动作效果。即是子节点带动父节点的运动，相比较之下的正向动力学就是从父节点到子节点的运动。基于IK的角色动画校正有助于更加方便地对角色的动作进行控制和校正。对于关节树来说，当已知末端的坐标时，需要逆向求解每个关节的坐标，这就是反向动力学的核心思想[18]。

常见的反向动力学解决方案包括CCD（循环坐标下降）和CAA（圆形对齐算法）等[19]。

在Unity中，IK可以设置的部位包括头部、左右手、左右脚、身体、左右手肘和左右膝盖这10个关节。

例如，在一个包含人体对话手势的动画中，可以利用手部的IK来控制和校正角色的手部位置，从而避免一些穿模现象，实现更加真实、自然的效果。

4.2 数字人动作模块实现与优化

4.2.1 数字人动作模块实现

1、骨骼系统

首先遇到的问题就是使用的模型骨骼与传统的软件和Unity引擎中的骨骼系

统有所不同，因此并不能让本项目的数字人模型骨骼无缝衔接Unity的骨骼系统。

本项目使用了两种可行的解决方案：

（1）全部使用和模型骨骼一致的动画，此时Unity中动画资产和角色模型资产的Rig选项卡全部设置为Generic（通用动画系统），并且动画和人物的骨骼需要保持一致。

（2）使用其他常见动画素材库中的骨骼动画资源（如Mixamo），此时Mixamo下载的动画资产和本研究的数字人的动画资产全部都要适配于Unity的骨骼系统，也就是把骨骼映射为Humanoid（人形动画系统），即需要进行重定向的操作。

2、动画资产

（1）自己手动制作关键帧动画。

（2）使用网上动画资源。

（3）根据需求自动生成动作。即上一章实现的多模态驱动手势动作生成。

4.2.2 数字人动作模块优化

1、动画资产错误解决方案

大部分动画资产应用在数字人身上，都会有一些抖动，突变和骨骼不适配问题，此时都可以通过对动画资产进行精修的方式来适当地解决该问题，解决方案有很多，比较常见的是修改序列帧动画，最好是在Blender或Maya这样的专门建模软件当中进行，当然也可以直接在Unity中做调整。

2、穿模问题解决方案

IK校正动画: 为了校正动画，可以使用反向动力学（IK）技术。需要注意的是，当应用到Unity中时，类型必须是Humanoid，且在动画Layer的设置里勾选“IK Pass”选项，然后在C#代码中设置反向动力学目标的转换权重，让对应关节更靠近IK Goal，这样可以减少关节位置不自然和穿模问题的发生。

限制角度：在Unity中，肌肉(Muscles)可以控制骨骼的运动范围。对数字人Avater的肌肉选项卡进行调整，比如将其旋转角限制得比较小，可以防止穿模，同时，也能让角色动作更加自然。

3、部分骨骼不对应问题

在将多模态驱动生成的手势动作结果应用到数字人身上时，使用的是将本项目的数字人和生成结果动画的Rig都调整为统一的Humanoid的方案，即进行骨骼

重定向以适配Unity的骨骼系统。在骨骼重定向过程中，会出现很多骨骼不对应的现象，这是由于骨骼不匹配的原因。需要将这些不匹配的骨骼进行手动的大量的比对，以寻找最合适的相对应的骨骼。

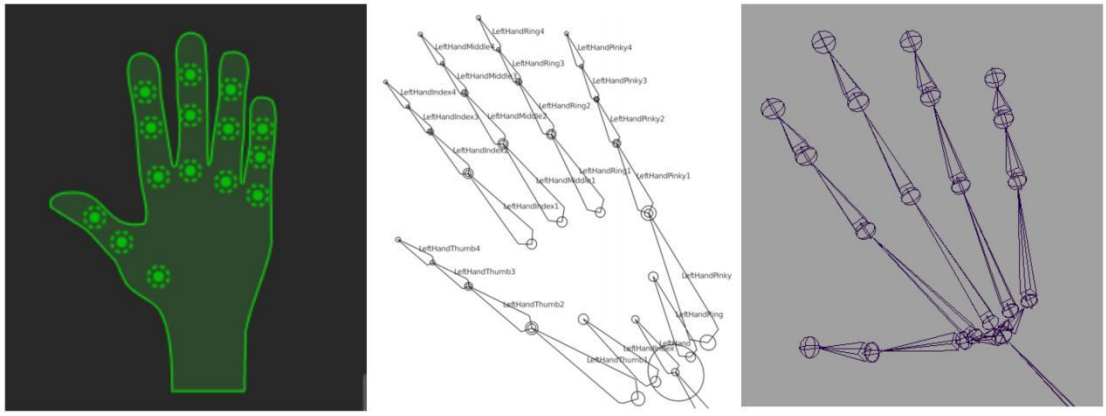

图 4-1 Unity humanoid、多模态驱动生成的动画、数字人三者的左手骨骼对比

从图 4-1 中可以看到 Unity humanoid、多模态驱动生成的动画、数字人三者的左手骨骼有很大区别，比如多模态驱动生成的动画骨骼中每根手指关节有 4 个，但是其他两个骨骼只有 3 个。多模态驱动生成的动画骨骼中手指有四个关节是为了在动作捕捉时更准确地捕捉手部姿势，而数字人骨骼和 Unity humanoid 骨骼是为了模拟真实世界中，符合人类手指只有三个关节的事实。且多模态驱动生成的动画骨骼例如左手食指和拇指会有一个额外的父骨骼，而其他两套骨骼都没有。

最终产生的骨骼不匹配效果可以在 Unity 中进行手动重定向，完成较合理的映射。

4.3 实验结果与分析

4.3.1 多模态驱动生成的手势动作在 Unity 中的应用

图 4-2 多模态驱动生成的手势动作的 Unity 应用

图 4-2 中可以看到右边的 UI 上有多个测试用例按键，其实本项目实际上有几十个测试用例，这里只做个别展示测试。用户点击不同的测试用例会驱动不同的多模态数据去生成手势动作，并最终展示在数字人身上。同时，利用 Unity 播放多模态数据中的语音，就可以直观体会到不同语音对应不同的手势动作。

4.3.2 数字人动作模块优化效果

1、IK 矫正

从图 4-3 可以看到，同一帧动画中，IK 矫正前的动画应用于数字人穿模严重，IK 矫正后，纠正了穿模问题，且动作也很自然。

图 4-3 同一帧 IK 矫正前后对比

第五章 数字人系统综合与测试

本章主要介绍了数字人综合系统的设计与实现，包括人体建模与手势驱动动作结合、服装适配问题、头发计算量问题以及口型驱动等。该系统能够根据用户自定义输入，生成具有个性化特征和相貌的数字人，并能够根据多模态驱动手势动作生成，应用于不同对话场景中。在数字人综合系统实现过程中，还遇到了很多其他问题，本章也介绍了这些模块的解决方案。本项目采用了骨骼点替换技术、面片式头发和语音驱动口型算法，解决了服装适配、头发计算量和口型驱动问题。实验结果表明，该系统能够实现数字人的个性化表现和动作生成。

5.1 数字人综合系统的设计与实现

5.1.1 人体建模与多模态驱动手势动作结合

由于数字人系统本质上是一个由多个模块组成的复合系统，而这些模块之间具有兼容性和共存性，因此本项目也成功地将动作模块和人体建模模块融合到了系统中。例如，数字人在多模态驱动生成的动作下，可以随时去调整数字人的人脸建模，表情建模和体型建模，让数字人更具有个性化。当然更合理的做法是用户先用本项目的系统自定义一个角色，即按照自己喜好捏人，让数字人拥有特殊的体型，相貌，然后这个数字人就可以通过多模态驱动手势动作生成，实现有独特个性特征和相貌的数字人，在不同场景下，赋予其个性化的动作和手势行为。不管是对话，新闻播报，还是游戏中的聊天都可以有广泛应用。

5.1.2 其他部分

由于数字人工程是一个完整的工程，因此也不仅仅只有人体建模模块，多模态驱动手势模块和动作模块，为了确保数字人的完整性和真实性，服装模块和头发模块也是不可或缺的，它们为数字人提供了视觉上的真实感和自然感，本文并不展开细说这些模块，而是说一说在人体建模模块，多模态驱动手势模块和动作模块下，为了数字人更合理的表现，这些其他模块出现的问题和解决方案。

1、服装适配问题

服装方面采用了骨骼点替换技术，将服装的模型骨骼点替换为数字人模型的骨骼点，这样能够使数字人模型运动时，服装也能贴着人体运动。当需要换装时，

只需在对应文件夹下放入服装资源，即使骨骼并不完全适配也并不需要手动的给服装制作新的骨骼，而是直接自动识别人体骨骼并贴合人体运动。且在做例如基于骨骼驱动的人体建模任务的时候，也不会穿模，而是随骨骼适配人体。

2、头发计算量问题

出于数字人的美观的考虑，需要为其配备头发。本项目尝试过面片式与发丝式的头发，发丝式即头发的每一根发丝都被建模出来，视觉效果上确实更真实，也更方便物理模拟，但是计算量过于庞大。为了追求数字人的实时性，最终使用了手机游戏端常见的面片式头发。面片式即头发由多个面片组成，面片的材质包含透明通道，部分材质贴图绘制了发丝。采用面片式，在Unity中定义适配头发渲染的材质。此时出现的问题是会出现错误的半透明效果，解决方案是将深度写入打开来做正常的透明度混合。

3、口型驱动

本项目的数字人可以在给定语音等输入数据后，驱动其作出相应手势动作，但是如果场景中数字人闭着嘴播放相应音频作出相应动作，那么会显得很奇怪。因此本项目使用了语音驱动口型算法，结合第二章实现的基于Blendshape的表情建模功能，在Unity中应用于数字人口型驱动中[20]。

5.1.3 数字人综合系统

1、总体流程图

图5-1展示了数字人综合系统的部分功能与流程图。

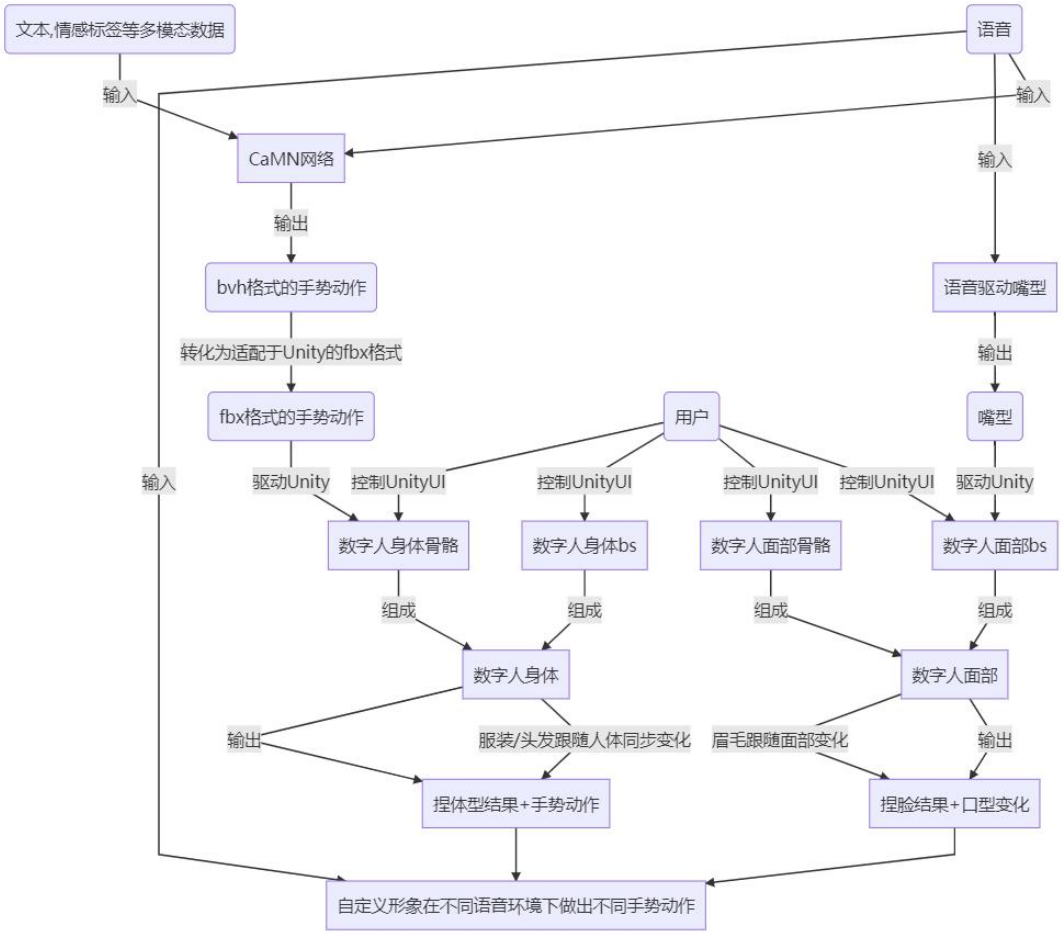

图 5-1 数字人综合系统图

2、Unity部分的代码时序

由于多模态驱动手势动作模块的CaMN网络部分的代码时序图已经放在第三章了，这里仅展示Unity部分的代码时序图。

图5-2展示了Unity中，实现基于Blendshape的人脸建模、表情建模和人体建模的代码时序图。图5-2展示了从用户输入到自定义数字人的部分过程。

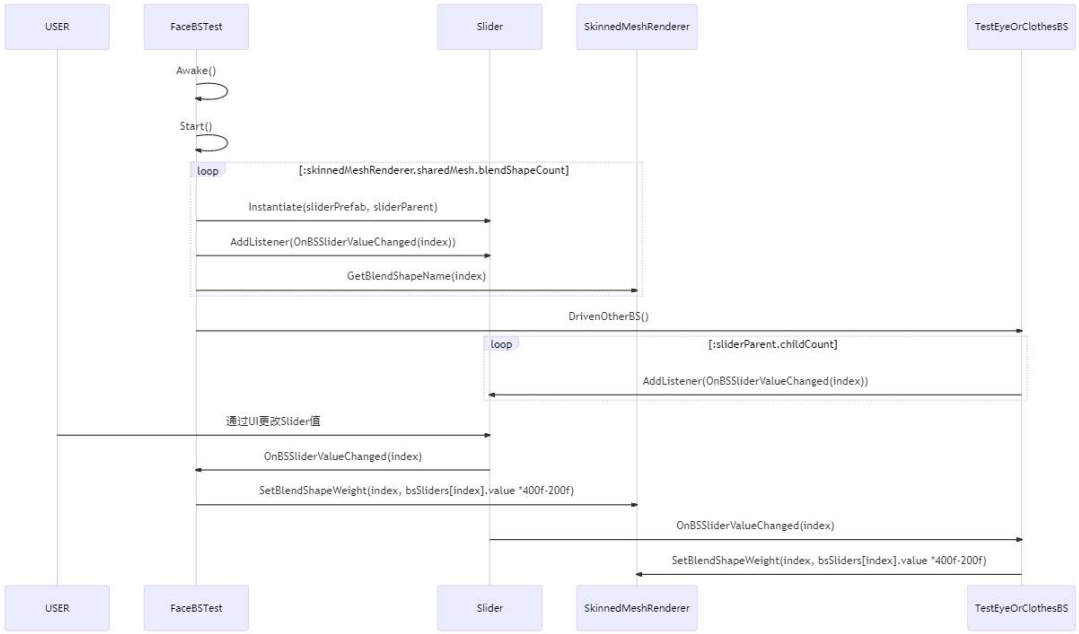

图 5-2 Unity 中 Blendshape 人体建模功能时序图

图5-3展示了在Unity中，从用户输入到手势动作驱动生成和嘴型的驱动生成，然后作用于数字人身上，展现给用户的过程。

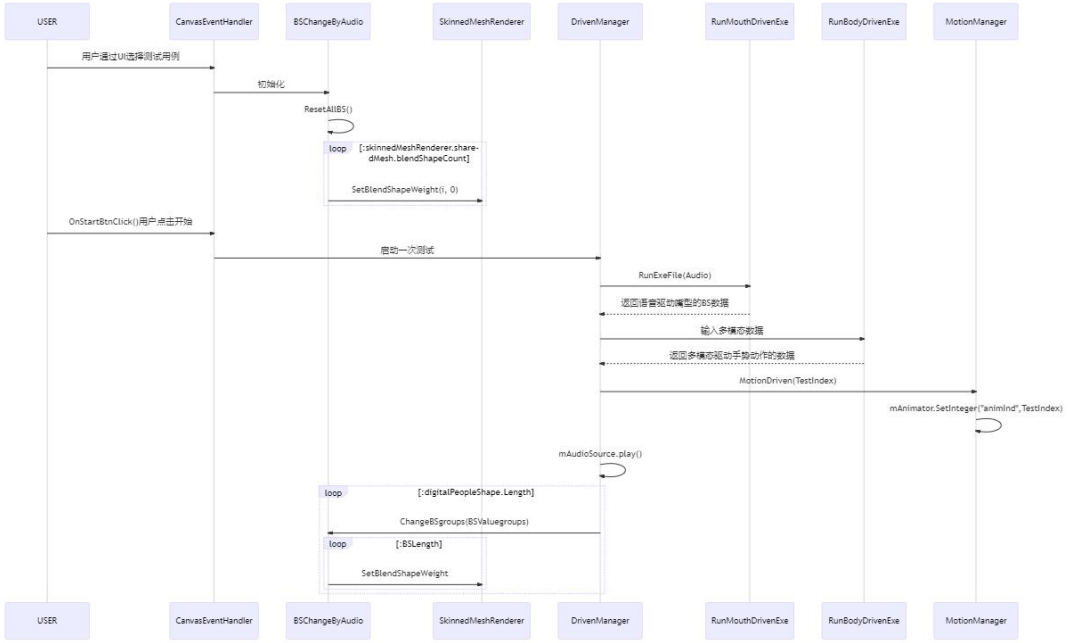

图5-3 Unity中多模态驱动模块时序图

图5-2和图5-3所呈现的功能可以同时并行进行，即可以同时编辑数字人的自

定义形象，以及执行多模态驱动任务。换句话说，就是可以创造同时具有人类形象和行为的数字人

5.2 实验结果与分析

图5-4的图（a）左侧UI中用户可以调整控制脸部和全身的滚动条，来进行自定义形象，右上角UI中用户可以点击驱动手势动作生成。点击切换相机按钮转换为图（b）可看到数字人全身，用户可以进行体型的自定义。

(a) (b)

图 5-4 数字人综合系统

图5-5中可以看到，用户可以通过本项目的数字人系统，去自定义各种各样体型不同，相貌不同的数字人，并通过语音等多模态输入去驱动他们自动生成适配这段语音的手势动作。不仅如此，他们也可以有不同的表情以及与语音实时匹配的口型。

图 5-5 利用数字人综合系统实现的效果

通过这套数字人系统，用户可以有较高自由度地去创造独一无二的数字人形象，并通过语音、身份等多模态信息驱动数字人生成相应对话场景下的手势动作。因此，该系统能够尝试应用于虚拟主播、虚拟客服、数字专家、数字代言人、虚拟观众和虚拟教师等领域，帮助用户实现个性化需求。总之，本项目的数字人系统具有一定的灵活性，可以适应不同的应用场景。

第六章 总结与展望

6.1 总结

首先，论文介绍基于 Blendshape 和基于骨骼驱动的三维人体建模系统的设计与实现，分别介绍了两者的原理，并对它们进行优劣对比。然后基于这些原理，介绍了本项目在 Unity 中实现的人脸、表情和体型建模功能。

其次，本文介绍了数字人系统中基于多模态信号驱动手势动作生成模块的设计与实现。本项目参考了相关论文的方法进行了实现，并详细介绍了 CaMN 网络的整体架构，包括各个编码器和解码器的结构，以及损失函数的设置。通过这些内容来探讨如何从多模态数据中生成手势动作。最后，展示了该网络在 Unity 中应用于数字人所取得的效果。

在将多模态驱动生成的手势动作应用到数字人身上时，还要经过许多步骤，因此也引出了动作模块，介绍了数字人要实现动作模块所涉及的原理，以及在 Unity 中的应用方案。

最后，将各个模块整合在一起，让数字人能够在拥有特殊的自定义外观的同时，可以个性化地驱动手势和动作生成，生成有独特外观，有独一无二手势和动作行为的数字人。

6.2 工作展望

尽管已经实现了上述功能，该数字人系统还存在很多不足之处，接下来将继续对数字人系统进行进一步的改进和优化：

1、多模态数据集获取和处理很麻烦，希望后续能通过优化网络以及强化预处理功能，使得能从更方便获取的更少的信息中提取出更多有效的结果。

2、多模态驱动手势动作应用在本项目的数字人模型上时，出现了因数字人和动画文件的骨骼不匹配带来动作并没有那么自然的问题，因此可以更加细致地使用重定向，确保数字人骨骼与驱动生成的手势动画骨骼足够接近。如果重定向不能解决，可以对数字人手动添加或删除骨骼与重新蒙皮，使其与生成的动画文件骨骼相匹配。

3、代码的优化。后续将减少代码的耦合度。通过将相关的功能模块解耦，

让代码更容易理解和维护。同时，也会进行算法和数据结构的优化，以及提高代码的可读性和可维护性。

参考文献

[1] Li Q， Deng Z . Facial motion capture editing by automated orthogonal blendshape construction and weight propagation. 2008.

[2] Luo Y， Wu C M， Zhang Y . Facial expression recognition based on fusion feature of PCA and LBP with SVM[J]. Optik - International Journal for Light and Electron Optics, 2013, 124(17):2767-2770.

[3] Lewis J P . Pose space deformations : A unified approach to shape interpolation and skeleton-driven deformation[C]// Proc. of ACM SIGGRAPH, Computer Graphics Annual Conference Series, 2000. 2000.

[4] Le B H, Deng Z . Smooth skinning decomposition with rigid bones[J]. Acm Transactions on Graphics, 2012, 31(6):1-10.

[5] Manteaux P L， Vimont U， Wojtan C , et al. Space-time sculpting of liquid animation[C]// the 9th International Conference. ACM, 2016.

[6] Liu H, Zhu Z, Iwamoto N, et al. BEAT: A Large-Scale Semantic and Emotional Multi-Modal Dataset for Conversational Gestures Synthesis[C]// 2022.

[7] Mcauliffe M， Socolof M， Mihuc S , et al. Montreal Forced Aligner: Trainable Text-Speech Alignment Using Kaldi[J]. 2017.

[8] Bojanowski P， Grave E， Joulin A , et al. Enriching Word Vectors with Subword Information[J]. Transactions of the Association for Computational Linguistics, 2017, 5:135-146.

[9] Bai S, Kolter J Z, Koltun V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling[J]. arXiv preprint arXiv:1803.01271, 2018.

[10] He K， Zhang X， Ren S , et al. Deep Residual Learning for Image Recognition[C]// IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2016.

[11] Yoon Y， Cha B， Lee J H , et al. Speech gesture generation from the trimodal context of text, audio, and speaker identity: ACMPUB27New York, NY, USA, 10.1145/3414685.3417838[P]. 2020.

[12] Raj S， Prakasam P， Gupta S . Multilayered convolutional neural network-based auto-CODEC for audio signal denoising using mel-frequency cepstral coefficients[J]. Neural

Computing and Applications, 2021(2):1-11.

［13］Xiang Z， Sang J， Zhang Q，et al. A new convolutional neural network-based steganalysis method for content-adaptive image steganography in the spatial domain[J]. IEEE Access, 2020, PP(99):1-1.

［14］Manohar K， Logashanmugam E . Hybrid deep learning with optimal feature selection for speech emotion recognition using improved meta-heuristic algorithm[J]. Knowledge-based systems, 2022(Jun.21):246.

［15］Ng E， Ginosar S， Darrell T，et al. Body2Hands: Learning to Infer 3D Hands from Conversational Gesture Body Dynamics[C]// Computer Vision and Pattern Recognition. IEEE, 2021.

［16］Goodfellow I， Pouget-Abadie J， Mirza M，et al. Generative Adversarial Nets[C]// Neural Information Processing Systems. MIT Press, 2014.

［17］Foley J D， Dam A V， Feiner S K . Computer Graphics- Principles and practice, Second Edition in C, Pearson Education, 2007. 2015.

［18］Aristidou A， Lasenby J， Chrysanthou Y，et al. Inverse Kinematics Techniques in Computer Graphics: A Survey[J]. Computer Graphics Forum, 2018, 37(6):35-58.

［19］Aristidou A， Lasenby J . Inverse Kinematics: a review of existing techniques and introduction of a new fast iterative solver. Cambridge University Engineering Department, 2009.

［20］Cudeiro D， Bolkart T， Laidlaw C，et al. Capture, Learning, and Synthesis of 3D Speaking Styles[J]. 2019.

附 录