ISSN 1000-0054 清华大学学报（自然科学版）2011 年 第 51 卷 第 9 期 5/33
CN 11-2223/N J Tsinghua Univ (Sci & Tech), 2011, Vol. 51, No. 9 1180-1186

实时语音驱动的虚拟说话人

李冰锋， 谢 磊， 周祥增， 付中华， 张艳宁

（西北工业大学 计算机学院，陕西省语音与图像信息处理重点实验室，西安 710129）

摘 要：该文实现了一个实时语音驱动的虚拟说话人面部动画方案。随着语音信号的输入，同步生成对应的面部动画。这种实时语音驱动的虚拟说话人在可视电话、虚拟会议、音视频聊天等即时通讯与娱乐媒体领域具有巨大的应用潜力。由于音素是最小的可分发音单元，因此构建音素识别器，对输入语音信号进行实时音素识别。为提高语音与口型的同步效果，改进了音素识别与输出算法。考虑协同发音影响，利用动态视素生成算法，将识别得到的音素转化为对应的面部动画参数序列。最后用参数序列驱动按照 MPEG-4 面部动画标准参数化的 3-D 头部模型，实现面部动画的同步生成。主观 MOS 评测结果表明：本文所实现的实时语音驱动虚拟说话人在的同步性和逼真度上的 MOS 评分分别达到了 3.42 和 3.50。

关键词：可视语音合成；虚拟说话人；面部动画

中图分类号：TP 391 **文献标志码**：A

文章编号：1000-0054(2011)09-1180-07

Real-time speech driven talking avatar

LI Bingfeng, XIE Lei, ZHOU Xiangzeng, FU Zhonghua, ZHANG Yanning

(Shaanxi Provincial Key Laboratory of Speech and Image Information Processing, School of Computer Science, Northwestern Polytechnical University, Xi'an 710129, China)

Abstract: This paper presents a real-time speech driven talking avatar. Unlike most talking avatars in which the speech-synchronized facial animation is generated offline, this talking avatar is able to speak with live speech input. This life-like talking avatar has many potential applications in videophones, virtual conferences, audio/video chats and entertainment. Since phonemes are the smallest units of pronunciation, a real-time phoneme recognizer was built. The synchronization between the input live speech and the facial motion used a phoneme recognition and output algorithm. The coarticulation effects are included in a dynamic viseme generation algorithm to coordinate the facial animation parameters (FAPs) from the input phonemes. The MPEG-4 compliant avatar model is driven by the generated FAPs. Tests show that the avatar motion is synchronized and natural with MOS values of 3.42 and 3.5.

Key words: visual speech synthesis; talking avatar; facial animation

虚拟说话人技术又称面部动画、可视语音合成，是指利用计算机生成开口说话的虚拟人物的技术[1-2]。听觉和视觉是人类最主要、最便捷的 2 种沟通交流方式，虚拟说话人结合了听视觉双模态的沟通方式，提供了自然的人机交互手段和真实感体验。因此，虚拟说话人技术在电影特效、虚拟现实、游戏娱乐、可视电话、语言学习等领域具有广泛的应用[1]。

按照驱动方式来分，虚拟说话人可以分为动作驱动(motion driven)、文本驱动(text driven)和语音驱动(speech driven)。动作驱动方法通过相机捕捉表情，采用面部特征跟踪技术，提取面部特征点以驱动头部模型。文本驱动方法利用文语转换(text to speech, TTS)技术将文本转换为音素序列，再映射为口型等视素信息，产生嘴部或面部动作。吴志勇、蔡莲红等[3]提出基于 MPEG-4 面部动画参数(facial animation parameter, FAP[4])文本驱动的中文可视语音与面部表情合成方法；文[5]利用三音素单元选择算法，从录制的面部视频库中选取相关视频单元进行拼接合成，进而生成文本对应的面部动画序列。与上述 2 种方法不同，语音驱动方法则是利用真实语音作为输入，通过算法直接将语音转换为面部或口型参数，驱动面部或头部模型说话。例如，Matthew[6] 提出的基于隐 Markov 模型(hidden markov model, HMM)的相关控制方法，通过学习算法获得人脸动画的控制模型。谢磊等[7] 则从发音的角度出发，采用动态 Bayes 网络对发音器官的听视觉表现进行混合建模，实现了语音到面部动画的直接映射。

收稿日期：2011-07-15

基金项目：国家自然科学基金青年基金资助项目（60802085）；
国家自然科学基金面上项目（61175018）；
陕西省科技计划青年科技新星项目（2011KJXX29）；
陕西省自然科学基础研究计划（2011JM8009）

作者简介：李冰锋(1988—)，男(汉)，河南，硕士研究生。

通信作者：谢磊，教授，E-mail：lxie@nwpu.edu.cn

与当前大多数有关虚拟说话人的研究不同，本文研究实时语音驱动(real-time speech driven)的虚拟说话人技术，即随着语音信号的输入，同步生成对应的面部动画。这种技术在可视电话、虚拟会议、音视频聊天等即时通讯与娱乐媒体领域具有巨大的应用潜力。该技术可以使通信双方只通过传递语音信号(或辅以少量的视频信号)就可以看到对方讲话的实时面部视频，从而满足网络低带宽、通信质量不佳情况下的视频通讯需求。在娱乐方面，即时聊天的虚拟化身、角色扮演也是实时语音驱动的虚拟说话人的典型应用。

实现实时语音驱动的虚拟说话人，关键在于语音信号的实时处理和逼真面部动画的同步生成。不同于非实时技术，在实时语音驱动面部动画中语音内容事先无法获知。例如，在可视电话应用中，通信双方的通话内容事先未知，仅能随着说话人的讲话即时合成对应的面部动画。

本文实现了一个实时语音驱动的虚拟说话人方案。由于音素是最小的可分发音单元，因此本文构建了音素识别器，对输入语音信号进行实时音素识别；然后利用动态视素生成算法[8]，将识别得到的音素转化为对应的面部动画参数序列；最后用参数序列驱动 3-D 头部模型，实现面部动画的同步生成。

1 系统整体简介

本文的系统可以随着语音信号的输入，同步输出与之对应的面部动画。系统结构如图 1 所示。从图 1 可以看出，系统共由 4 个主要模块组成：控制模块、实时识别模块、合成模块和驱动模块。其中，控制模块负责系统初始化、模块间消息的传送和同步等功能；识别模块是一个音素识别器，其功能是检测语音输入、语音识别和实时将识别得到的音素发送到合成模块；合成模块针对识别模块输入的音素，计算其对应的 FAP 并及时发送到驱动模块；驱动模块则利用接收到的 FAP 来驱动面部模型，得到面部动画。系统的详细工作流程如图 2 所示。

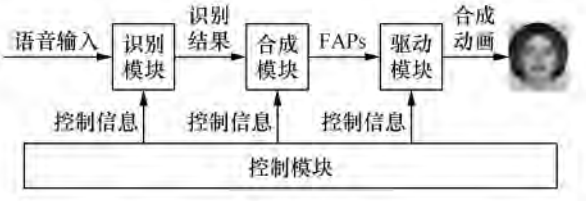

图 1 实时语音驱动的虚拟说话人系统模块构成

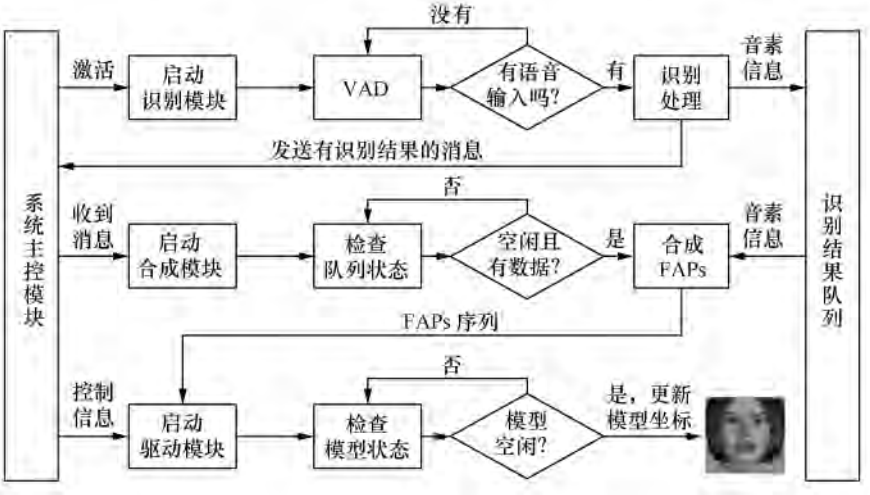

图 2 实时语音驱动的虚拟说话人系统流程图

在系统工作时，主控模块首先启动识别器，识别器收到指令检测是否有语音输入，即语音激活检测(voice activation detection，VAD)。一旦检测到语音输入即开始识别处理，将识别结果发送到结果队列，同时告诉主控模块已得到识别结果。主控模块收到有识别结果的消息，启动合成模块，合成模块检测队列状态，从中取出识别结果即音素，将其映射为对应视素后，利用动态视素生成算法合成 FAP 序列，发送到驱动模块。驱动模块利用收到的 FAP 序列，驱动头部模型得到对应的面部动画。

2 基于实时音素识别的面部动画生成

2.1 3-D 头部模型建立

本文首先使用 FaceGen 工具生成一个原始的 3-D 头部模型，然后使用 XfaceEd 将它按照 MPEG-4 面部动画标准进行参数化。该标准定义了 1 个标准的人脸部模型，又在此模型上定义了 66 个描述最小可理解面部动作(如嘴角上扬等)的 FAP 和 84 个特征

点,每个特征点通过变形函数控制着一个影响区域。

本文使用 XfaceEd 在 FaceGen 生成的原始模型上标出了脸部 18 个特征点,为每个特征点指定各自影响区域,使用上升余弦函数来控制影响区域内非特征点的运动。

2.2 实时音素识别

音素是语言的最小可分单元,为了实现实时语音驱动的效果,本文采用实时音素识别器将输入的语音实时转换为音素。在连续语音识别系统中,一段语音信号经过特征提取后得到一个特征帧序列 $O=(o_1,\cdots,o_I)(I\geqslant 1)$,假定该特征帧序列对应的一个音素串为 $W=(w_1,\cdots,w_n)(n\geqslant 1)$,那么连续语音识别的任务就是在语言 L 中找到与 O 对应的最可能音素串 $\hat{W}$ 满足:

$$\hat{W}=\underset{W\in L}{\operatorname{argmax}}P(W\mid O). \tag{1}$$

在基于 HMM 的语音识别中,使用 Veterbi 搜索算法,每处理一个特征帧序列就可以得到当前最佳的音素串。假设 t 时刻输入的特征帧为 o_t,由式(1)得到此时的最佳音素串为

$$\hat{W}_t=(\hat{w}_t^1,\hat{w}_t^2,\cdots,\hat{w}_t^{\text{last}(t)})=\underset{W\in L}{\operatorname{argmax}}P(W\mid(o_1,\cdots,o_t)). \tag{2}$$

其中 $\hat{w}_t^{\text{last}(t)}$ 表示该音素串的最后一个音素。设 $t+1$ 到 $t+N-1$ 时刻输入的特征帧分别为 o_{t+1},$\cdots$,o_{t+N-1},由式(1)同样可以得到 $N-1$ 个音素序列:

$$\begin{cases}\hat{W}_{t+1}=(\hat{w}_{t+1}^1,\hat{w}_{t+1}^2,\cdots,\hat{w}_{t+1}^{\text{last}(t+1)}),\\ \quad\vdots\\ \hat{W}_{t+N-1}=(\hat{w}_{t+N-1}^1,\hat{w}_{t+N-1}^2,\cdots,\hat{w}_{t+N-1}^{\text{last}(t+N-1)}).\end{cases} \tag{3}$$

为了实现实时面部动画效果,识别器需要随着语音的输入快速输出当前发音的音素。考虑到不可避免的识别错误,本文采用的识别输出方法是:如果连续 N 个特征帧搜索到的最佳音素序列的最后一个音素都相同,设为 $\hat{w}$,并且与上一个已经识别输出的发音音素 $\hat{w}_{\text{pre}}$ 不同,那么就将 $\hat{w}$ 作为当前的发音音素输出。即如果 t 到 $t+N-1$ 时刻得到的最佳音素序列满足:

$$\begin{cases}\hat{w}_t^{\text{last}(t)}=\hat{w}_{t+1}^{\text{last}(t+1)}\cdots=\hat{w}_{t+N-1}^{\text{last}(t+N-1)}=\hat{w},\\ \hat{w}\neq\hat{w}_{\text{pre}}.\end{cases} \tag{4}$$

那么就把 $\hat{w}$ 当作当前正在发音的音素输出,其中 $\hat{w}_{\text{pre}}$ 为上一个已经识别输出的发音音素。该算法的伪代码如图 3 所示。

```
Algorithm: phoneme recognition and output
initDecoder(); //初始化解码器
New_Phone_Judge_Queue.size = N; //新音素判别器队列式缓冲区大小设为 N
New_Phone_Judge_Queue.Reset(); //新音素判别器队列式缓冲区置空
While( !Speech_In_Buffer.isEmpty() ) //输入语音缓冲区不空
{
    Oi = Extract_Feature(); //实时提取语音特征,得到当前特征帧
    W = Decoder( Oi ); //进入解码器,得出当前最佳音素序列 W
    New_Phone_Judge_Queue.outQueue(); // NewPhoneJudgerQueue 队尾出队
    New_Phone_Judge_Queue.inQueue( W[last ] ); //W 最后一个音素进入 NewPhoneJudgerQueue 队头
    If ( New_Phone_Judge_Queue.isAllSame() && New_Phone_Judge_Queue.notEqual( LastOutput ))
        // NewPhoneJudgerQueue 中所有音素相同并且不同于上一个输出的音素
        Output_to_Generate_Model(); //将 NewPhoneJudgerQueue 中音素输出到合成模块可视化
        Update (LastOutput); //将 LastOutput 更新为当前输出的音素
}
```

图 3 音素识别与输出算法

关于 N 的选择需要考虑到 2 方面的因素:如果 N 取值过大,会延长音素的输出时间,降低语音与面部动画的同步效果,而且可能漏掉持续时间较短的音素,从而造成动作的明显不连贯现象;如果 N 取值太小,虽然可以加快音素的输出时间,但会增加音素识别的错误率,影响逼真度。本文通过反复实验,最终选取 N 为 3 作为经验值。

在输出某个音素的同时,为了后续计算 FAP,需要为该音素估计持续时间。本文通过语料统计出每个音素持续时间经验值,作为音素的持续时间,连同音素一同发送到合成模块。

2.3 动态视素生成

2.3.1 视素

音素识别器将输入的语音实时的转化为对应的音素后,需要进一步将音素转换为动态视素,从而在 3-D 模型上同步地生成该音素对应的面部动画。

由于某些不同的音素对应着相同或类似的口型动作(比如"p"、"b"等),可以将它们归为一类,称为一个视素(viseme)。MPEG-4 中定义视素为发

某一音素时典型的口型状态，称为静态视素[4]。汉语普通话中有 21 个声母和 38 个韵母(统称 59 个音素)，将具有相同或类似发音口型的音素合并，得到如表 1 所示的声、韵母到视素的映射表。

表 1 汉语普通话声、韵母到视素的映射表[3]

声母		韵母(单一口型)		
b,p,m	g,k,h	a,ang	ou	i
f	j,q,x	ai,an	e,eng	u
d,t,n	zh,ch,sh,r	ao	ei,en	v(/yu/)
l	z,c,s	o	er	SIL

注：复合口型韵母共有 13 个，映射时被切分为单一口型韵母的组合，如 iao=i+ao。

根据 MPEG-4 中 FAP 的定义，每个静态视素都对应着一组 FAP。为了得到每个中文视素的所有 FAP，录制了一组说话人发音的面部视频，该视频包含了所有中文声母和韵母的发音，然后使用面部特征跟踪算法，获得每个静态视素对应的所有 FAP(以下称为静态参数)。

2.3.2 动态视素

由于发音过程是连续的、动态的，因此静态视素不能准确描述人发音的连续过程。本文使用动态视素[8]来描述某一视素发音时口型从产生到消失的完整变化过程。用来实现整个过程的参数称为该视素的动态参数。为了描述视素的动态过程，每个视素的每一个 FAP 都定义了 1 个控制函数，同时引入了 2 个无声模型以及它们各自的控制函数[8]。

假设 p 表示视素 s 的某一个 FAP，那么 p 在 s 中的控制函数 D_{sp} 可以表示为：

$$D_{sp}=\begin{cases}\alpha_{sp}\,\mathrm{e}^{-\theta_{sp(-)}|\tau|^{c}}, & \tau\geqslant 0;\\ \alpha_{sp}\,\mathrm{e}^{-\theta_{sp(+)}|\tau|^{c}}, & \tau<0.\end{cases}\tag{5}$$

其中：α_{sp} 表示控制函数的峰值；$\theta_{sp(-)}$、$\theta_{sp(+)}$ 是衰减系数；c 是常数；τ 表示当前时刻到控制函数中心时刻的距离，$\tau=t_{so}-t$，t_{so} 表示控制函数中心时刻，t 表示当前时刻。

2 个无声模型分别表示无声到有声和有声到无声的口型转变[8]。从无声到有声(左无声模型)的控制函数可表示如下：

$$D_{lp}=\alpha_{lp}\,\mathrm{e}^{\mathrm{sgn}(\sigma)\theta_{lp}|\sigma|^{c}}.\tag{6}$$

其中 $\sigma=t_{lo}-t$，t_{lo} 表示左无声模型中心的时刻。从有声到无声(右无声模型)的控制函数为

$$D_{rp}=\alpha_{rp}\,\mathrm{e}^{-\mathrm{sgn}(\upsilon)\theta_{rp}|\upsilon|^{c}}.\tag{7}$$

其中 $\upsilon=t_{ro}-t$，t_{ro} 表示右无声模型中心的时刻。

任意时刻视素的动态参数由它们的静态值按照式(5)—(7)表示的控制函数加权构成[8]：

$$F_{sp}(t)=\frac{D_{sp}(t)\times T_{sp}+(D_{lp}(t)+D_{rp}(t))\times T_{0p}}{D_{sp}(t)+D_{lp}(t)+D_{rp}(t)}.\tag{8}$$

其中：T_{0p} 为自然状态下中参数 p 的静态值；T_{sp} 为视素 s 中参数 p 的静态值。控制函数及参数变化过程如图 4 所示。

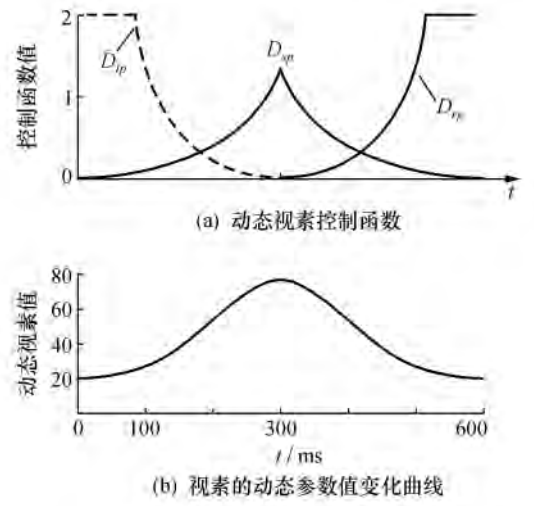

图 4 动态视素模型的控制函数和视素参数[8]

2.3.3 基于实时音素识别的动态视素生成

动态视素算法中，每个参数的计算都同时受到前后视素的影响。但是在实时音素识别中，某一时刻只知道当前和之前的音素。计算当前视素的动态参数，只能考虑之前视素对它的影响。因此本文对动态视素算法进行了修改，以满足实时性的需要。考虑到在当前视素前越早出现的视素对其动态参数的计算影响越小，本文在计算当前视素的动态参数时，只与其前一个视素的控制函数进行加权，这也降低了计算复杂度。

为了实现相邻视素间动态参数的平滑过渡，本文将每个视素的动态参数分 2 个时段进行计算。处理当前视素时，只计算它的控制函数中心时刻到上一个视素控制函数中心时刻之间的参数；而当前视素控制函数中心时刻之后的参数，在下一个视素到来时再计算。因此将式(5)表示为 2 个函数：

$$D_{sp(-)}=\alpha_{sp}\,\mathrm{e}^{-\theta_{sp(-)}|\tau|^{c}},\quad \tau\geqslant 0;\tag{9}$$

$$D_{sp(+)}=\alpha_{sp}\,\mathrm{e}^{-\theta_{sp(+)}|\tau|^{c}},\quad \tau<0.\tag{10}$$

此时的控制函数如图 5 所示。

假设某一时刻识别出视素 j，其上一个识别出的视素为 i，则参数 p 在视素 i 和 j 控制函数中心时刻之间的动态参数为

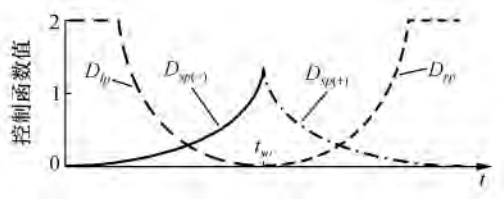

图 5 动态视素模型分时段定义的控制函数

$$F_p(t)=\frac{D_{jp(-)}(t)\times T_{jp}+D_{ip(+)}(t)\times T_{ip}}{D_{jp(-)}(t)+D_{ip(+)}(t)}. \tag{11}$$

其中：$t_{io}\leqslant t\leqslant t_{jo}$，$t_{io}$表示视素 i 控制函数的中心时刻，t_{jo}表示视素 j 控制函数的中心时刻；T_{ip}和 T_{jp}分别为视素 i 和 j 的参数 p 的静态值。由式(11)可以求出 t_{io}与 t_{jo}之间任一时刻的参数。假设连续识别得到 i、j、k 这 3 个视素，按照上述方法生成动态视素参数的过程如图 6 所示。

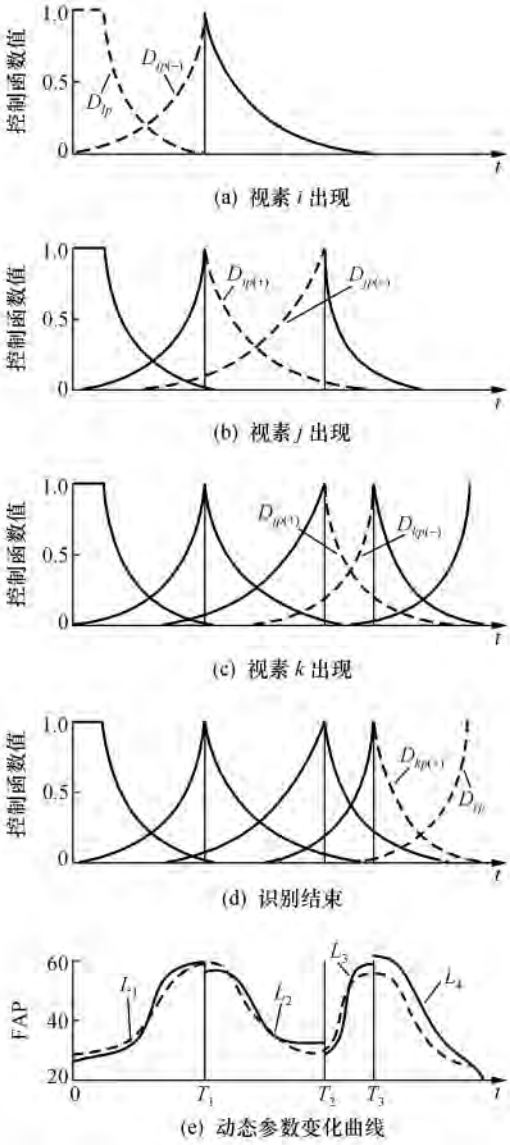

图 6 控制函数的变化以及参数的生成过程

图 6a 表示某时刻视素 i 被识别出，这时动态参数由 i 的静态参数值由在 0 到 T_1 时刻之间的 $D_{ip(-)}$和 D_{lp}加权得到，结果为图 6e 中的 L_1；

图 6b 表示某时刻视素 j 被识别出来，这时动态参数由 i、j 的静态参数值由在 T_1 到 T_2 时刻之间的 $D_{ip(+)}$和 $D_{jp(-)}$加权得到，结果为图 6e 中的 L_2；

图 6c 表示某时刻视素 k 被识别出来，这时动态参数由 j、k 的静态参数值由在 T_2 到 T_3 时刻之间的 $D_{jp(+)}$和 $D_{kp(-)}$加权得到，结果为图 6e 中 L_3；

图 6d 表示识别结束，这时动态参数由 k 的静态参数值由在 T_3 到结束时刻之间的 $D_{kp(-)}$和 D_{rp}加权得到，结果为图 6e 中的 L_4；

通过上述算法，每当识别器识别出一个音素，合成模块就会将它转换为对应的视素，然后计算该视素对应的动态参数，进而驱动头部模型，实时得到此时的面部动画。

3 实验结果及分析

3.1 音素识别器

本文建立的中文音素识别器所用声学模型为 5 状态、从左至右、无跳转的 HMM。用混合 Gauss 模型刻画每个状态的观测分布，训练了包括短时停顿(sil)在内的 60 个音素的 HMM。模型训练所用语料库是本实验室采集的近 90 h 的汉语语料库，包括朗读语音和广播新闻语音。

HMM 的建立、训练采用 HTK[9] 工具包完成，语音输入为 16 kHz、16 bit 的单声道信号。语音特征采用的是 12 维的 Mel 频率倒谱系数(MFCC)、能量特征以及它们的一阶、二阶差分共 39 维参数。特征提取的窗长为 25 ms，窗移为 10 ms。识别器采用实验室开发的基于 Veterbi 搜索的识别工具，音素之间无语法约束。

3.2 实验方案

采用主观评测方法对本文实现的实时语音驱动的面部动画方法进行实验测试。邀请了 21 位评测者，以平均意见评分(mean opinion score，MOS)为标准，对生成面部动画的语音-口型同步性和逼真度进行评判。为便于性能对比，本文设计了 4 种不同的实验方案。

1) 方案 1：采用 HTK 工具，将测试语音与其真实抄本进行强制对齐(forced alignment)，得到每个音素的持续时间；然后将这些音素合成面部动画。该方案为真实抄本驱动的非实时面部动画，作为面部动画效果的评判上限(upper bound)。

2）方案 2：将测试语音通过音素识别器进行识别，识别出有时间信息的音素序列；然后将这些音素序列转化为视素序列，离线合成面部动画。

3）方案 3：即本文提出的面部动画方案。

4）方案 4：每个音素等新的音素到来时才将其作为识别结果，其他与方案 3 相同。

测试语句为 20 句，来自于 10 个人（5 男 5 女）的朗读语音，每人 2 句。这 20 个测试语句平均长度为 4 s，囊括了所有 60 个音素，并尽可能多地涵盖不同音素上下文。本文建立的音素识别器对测试语句的音素级识别错误率为 26.34%。使用测试语句，按照上述 4 种方案分别录制了 20 个测试视频（共 80 个，观看视频样例请访问 http://www.nwpu-aslp.org/talkingavatar.html），将这 80 个视频的顺序随机打乱播放给受测者，受测者根据主观感觉，给出每个视频的同步性和逼真度得分。

3.3 实验结果

对受试者的打分采用四分位数差（inter quartile range，IQR）进行统计，结果如图 7 和 8 所示。去除野点（outlier）后得到的各方案的 MOS 如表 2 所示。方案 3 中每个音素的识别需要至少 3 个特征帧，因此忽略解码搜索和面部模型驱动等与计算平台相关的延时，最小理论延时为窗长加 2 倍的帧移即 45 ms；方案 4 中每个音素是在被完全识别出来以后才输出到合成模块，因此忽略其他因素，最小理论延迟为 1 个音素的持续时长，即平均 110 ms[10]。

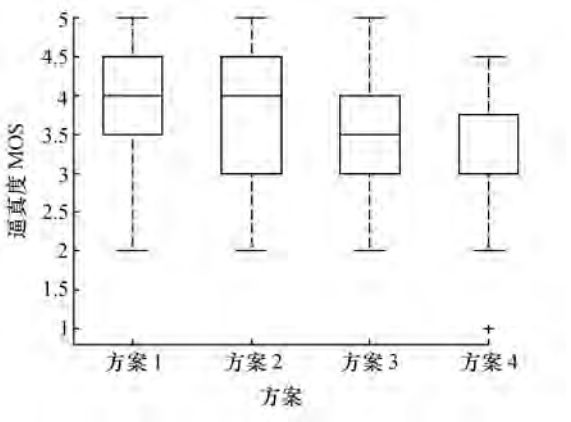

图 7 4 种方案的逼真度 MOS 箱线图

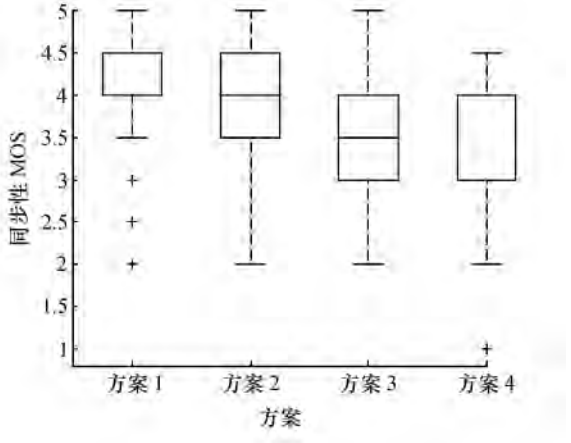

图 8 4 种方案的同步性 MOS 箱线图

表 2 各方案的 MOS 值对比

方案	同步性 MOS	逼真度 MOS	理论最小延时/ms	音素识别错误
1	4.34	4.01	0	无
2	3.98	3.81	0	有
3	3.42	3.50	45	有
4	3.21	3.16	110	有

从表 2 中可以看出：方案 1 事先已知合成语音的抄本以及每个音素对应的持续时间，以此为输入得到非实时生成的面部动画，具有最佳的同步性和逼真度。方案 2 使用的是识别器输出的有识别错误的抄本，但合成时已知音素序列，因此在同步性和逼真度上略低于方案 1。这说明音素识别错误会影响虚拟说话人的同步性和逼真度。

方案 3 与 4 属于实时语音驱动的面部动画，与非实时的方案（方案 1 和 2）相比，由于不能事先获取音素序列的完整信息，合成动画的同步性和逼真度有明显下降。方案 4 需要得到一个音素的完整时长信息才输出到合成模块，与方案 3 相比识别速度慢，因此同步性明显低于方案 3。

方案 4 每个音素的持续时间为真实值，比方案 3 的估计值更为准确，且音素识别的正确率比方案 3 高，因此理论上方案 4 合成的动画逼真度比方案 3 高，但表 2 中方案 4 的逼真度 MOS 反而比方案 3 低了 0.34。由此可见：在实时语音驱动的面部动画中，同步性的降低会在一定程度上降低整体的逼真度。

方案 3 即本文提出的方案，虽然失去了音素正确的持续时长，但缩短了音素的识别时间，使同步性和逼真度的 MOS 值分别达到 3.42 和 3.5。

4 结 论

本文通过改进音素识别器，首先完成了实时音素识别，然后结合动态视素生成算法，实现了实时语音驱动的面部动画方案。

由于音素是最小的可分发音单元，因此本文构

建音素识别器,对输入语音信号进行实时音素识别。同时为提高语音与口型的同步效果,改进了音素识别与输出算法。利用动态视素生成算法,将识别得到的音素转化为对应的FAP序列。最后用FAP驱动3-D头部模型,实现面部动画的同步生成。主观MOS评测结果表明:本文所实现的实时语音驱动的虚拟说话人系统在的同步性和逼真度上的MOS分别达到了3.42和3.50。

本文采用音素作为语音到面部动画的过渡单元,使得系统可以接收非特定人的语音输入;采用模块化设计,且模块间只需传递较少数据,因此各模块可以采用分布式计算,在基于网络的面部动画中具有一定的应用和推广价值。

参考文献 (References)

[1] Cosatto E, Ostermann J, Graf H P, et al. Lifelike talking faces for interactive services [J]. *Proceedings of the IEEE*, 2003, **91**(9): 1406-1429.

[2] TANG Hao, FU Yun, TU Jilin, et al. Humanoid audio-visual avatar with emotive text-to-speech synthesis [J]. *IEEE Transactions on Multimedia*, 2008, **10**(6): 969-981.

[3] WU Zhiyong, ZHANG Shen, CAI Lianhong, et al. Real-time Synthesis of Chinese Visual Speech and Facial Expressions using MPEG-4 FAP Features in a Three-dimensional Avatar [C]// The International Conference on Spoken Language Processing, Pittsburgh, 2006: 1802-1805.

[4] Pandzic I S, Forchheimer R. MPEG-4 Facial Animation [M]. New York: Wiley, 2002.

[5] HUANG Fujie, Cosatto E, Graf H. Triphone based units election for concatenative visual speech synthesis [C]// IEEE international conference on acoustics, speech, and signal processing. NJ: IEEE Press, 2002: 2037-2040.

[6] Brand M. Voice puppetry [C]// Proceedings of the SIGGRAPH'99. NY: ACM Press, 1999: 21-28.

[7] XIE Lei, LIU Zhiqiang. Realistic mouth-synching for speech-driven talking face using articulatory modeling [J]. *IEEE Transactions on Multimedia*, 2007, **9**(3): 500-510.

[8] 王志明, 蔡莲红. 动态视位模型及其参数估计 [J]. 软件学报, 2003, **14**(3): 461-466.
WANG Zhiming, CAI Lianhong. A dynamic viseme model and parameter estimation [J]. *Journal of Software*, 2003, **14**(3): 461-466. (in Chinese)

[9] Young S, Evermann G, Kershaw D, et al. The HTK Book [M]. Cambridge University Engineering Department, 2009.

[10] 王理嘉, 林焘. 语音学教程 [M]. 北京大学出版社, 1992.
WANG Lijia, LIN Tao. Phonetics Course [M]. Peking University Press, 1992. (in Chinese)

(上接第1179页)

语音损伤。其他3种方法均有较多的残留噪声,在SNR较低时会有语音损伤,对冲击噪声无降噪效果。

3 结 论

本文针对于近讲场景,提出了一种使用双麦克的近讲语音增强算法,运用ITD、IID等特征信息及掩蔽原理,将目标语音从带噪语音中选择出来。与传统的单通道增强方法如谱减法等相比,该方法对于各种类型的环境噪声都有更好的降噪效果。

参考文献 (References)

[1] Cherry E C. Some experiments on the recognition of speech with one and with two ears [J]. *J of ASA*, 1953, **25**: 975-979.

[2] Bregman A S. Auditory Scene Analysis [M]. Cambridge: MIT Press, 1990.

[3] Hansler E, Schmidt G. Topics in Acoustic Echo and Noise Control [M]. Berlin: Springer, 2006.

[4] Roman N, Wang D L, Brown G J. Speech segregation based on sound localization [J]. *J of ASA*, 2003, **114**(4): 2236-2252.

[5] Meddis R. Simulation of mechanical to neural transduction in the auditory receptor [J]. *J of ASA*, 1988, **83**(3): 1056-1063.

[6] Berouti M, Schwartz M, Makhoul J. Enhancement of speech corrupted by acoustic noise [J]. *Proc IEEE Int Conf Acoust, Speech, Signal Process*, 1979: 208-211.

[7] Scalart P, Filho J. Speech enhancement based on a priori signal to noise estimation [J]. *Proc IEEE Int Conf Acoust, Speech, Signal Process*, 1996: 629-632.

[8] Ephraim Y, Malah D. Speech enhancement using a minimum mean-square error log-spectral amplitude estimator [J]. *IEEE Trans Acoust, Speech, Signal Process*, 1985: **23**(2): 443-445.