

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL
FACULDADE DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**GERAÇÃO DE ANIMAÇÕES
FACIAIS PERSONALIZADAS
EM AVATARES**

ROSSANA BAPTISTA QUEIROZ

Dissertação de Mestrado apresentada como requisito para obtenção do título de Mestre em Ciência da Computação pelo Programa de Pós-graduação da Faculdade de Informática. Área de concentração: Ciência da Computação.

Orientadora: Profa. Dra. Soraia Raupp Musse

Porto Alegre, Brasil
2010

Dados Internacionais de Catalogação na Publicação (CIP)

Q3g	Queiroz, Rossana Baptista Geração de animações faciais personalizadas em avatares / Rossana Baptista Queiroz. – Porto Alegre, 2010. 104 f. Diss. (Mestrado) – Fac. de Informática, PUCRS. Orientador: Prof ^a . Dr ^a . Soraia Raupp Musse. 1. Informática. 2. Computação Gráfica. 3. Animação por Computador. 4. Avatares. I. Musse, Soraia Raupp. II. Título. CDD 006.6
-----	---

**Ficha Catalográfica elaborada pelo
Setor de Tratamento da Informação da BC-PUCRS**

TERMO DE APRESENTAÇÃO DE TESE DE DOUTORADO
OU DISSERTAÇÃO DE MESTRADO
EMITIDA E ASSINADA PELA FACULDADE



Pontifícia Universidade Católica do Rio Grande do Sul
FACULDADE DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

TERMO DE APRESENTAÇÃO DE DISSERTAÇÃO DE MESTRADO

Dissertação intitulada "Geração de Animações Faciais Personalizadas em Avatares", apresentada por Rossana Baptista Queiroz, como parte dos requisitos para obtenção do grau de Mestre em Ciência da Computação, Sistemas Interativos e de Visualização, aprovada em 19/03/10 pela Comissão Examinadora:

Profa. Dra. Soraia Raupp Musse –
Orientadora

PPGCC/PUCRS

Profa. Dra. Renata Vieira –

PPGCC/PUCRS

Prof. Dr. Luiz Carlos Pacheco Rodrigues Velho –

IMPA

Homologada em 02/08/2011, conforme Ata No. 14/11 pela Comissão Coordenadora.

Prof. Dr. Fernando Gehm Moraes
Coordenador.

PUCRS

Campus Central

Av. Ipiranga, 6681 – P32 – sala 507 – CEP: 90619-900

Fone: (51) 3320-3611 – Fax (51) 3320-3621

E-mail: ppgcc@pucrs.br

www.pucrs.br/facin/pos



Parecer sobre Dissertação de Mestrado

A aluna Rossana Baptista Queiroz, em sua dissertação de Mestrado, aborda de forma abrangente o problema de animação facial, de grande relevância para a área e com inúmeras aplicações em potencial.

Mais especificamente, ela apresenta um modelo para a geração de animações faciais que engloba de forma efetiva o controle de alto nível, baseado em descrições procedimentais, e o controle de baixo nível, dirigido por performance.

O texto da dissertação está bem escrito e relata claramente o trabalho de pesquisa, bem como descreve com detalhes a avaliação empírica dos resultados.

Além disso, a aluna fez uma apresentação oral impecável do trabalho na ocasião da defesa, respondendo de forma satisfatória as perguntas da banca na fase de arguição.

Por esses motivos, recomendo a aprovação da candidata.

Luiz Velho
Pesquisador Titular - IMPA

FACULDADE DE INFORMÁTICA - PUCRS	
PPGCC	
Recebido em:	23 / 02 / 2010
Ass:	

“Always in motion is the future.”

Yoda, Star Wars

Agradecimentos

Agradeço, em primeiro lugar, ao grande Artista criador de todos nós, que nos modelou, animou e nos renderizou neste mundo com tanta perfeição.

Agradeço à minha família e meus amigos, pelo apoio e compreensão.

À minha orientadora e ao meu co-orientador (ainda que não oficial), não tenho palavras para agradecer tudo o que aprendi. Também agradeço aos professores das disciplinas, foi muito bom ver e rever os conteúdos.

Aos meus colegas e companheiros de aula e/ou projeto nestes dois anos, um agradecimento especial. E aos meus “irmãos Soraísticos”, reforço esse agradecimento. Junto com a nossa chefe, de fato formamos uma “Equipe dos Sonhos”, modéstia a parte. Foi muito bom conhecer, trabalhar e me relacionar com vocês!

Este trabalho foi desenvolvido em colaboração com a HP Brazil R&D, junto com uma equipe de pesquisadores da HP Labs de Palo Alto. Foi muito construtiva essa parceria, e muito legal ter reuniões na Halo. Obrigada!

Por fim, agradeço a todas as “cobaias” que emprestaram suas faces durante o desenvolvimento do trabalho, para a criação dos modelos 3D ou fazendo caretas. Sem dúvidas, este trabalho tem a cara de vocês. E também às que responderam o questionário de avaliação. Foi muito importante a sua participação.

GERAÇÃO DE ANIMAÇÕES FACIAIS PERSONALIZADAS EM AVATARES

Resumo

Neste trabalho, é apresentado um modelo para a geração de animações faciais personalizadas em avatares. O modelo inclui duas abordagens de controle paramétrico da face do avatar: baseado em *scripts* de ações faciais em alto-nível ou usando a técnica de Animação Dirigida por Performance (PDA). A abordagem de controle por *script* permite a geração de animações com expressões faciais, fala sincronizada e comportamento dos olhos interativamente pela descrição em alto-nível dessas ações faciais. A abordagem de controle por Animação Dirigida por Performance permite refletir as expressões faciais do usuário em seu avatar, considerando como entrada um pequeno conjunto de pontos característicos providos por algoritmos de Visão Computacional. O modelo é baseado no padrão MPEG-4 de Animação Facial, estruturado usando uma hierarquia dos parâmetros de animação, proposta para a abordagem de PDA. Essa estratégia permite a geração de animações nas regiões da face onde não houver dados supridos por Visão Computacional. Para deformar a face, usa-se Funções de Base Radial e Funções de Cosseno. Essa abordagem é computacionalmente barata e permite a animação dos avatares em tempo-real. Foi desenvolvido um *framework* de animação facial extensível capaz de animar facilmente faces parametrizadas de acordo com o padrão MPEG-4 em aplicações interativas. Também foram desenvolvidas algumas aplicações interativas como “estudo de caso” das potencialidades do modelo e realizada uma avaliação preliminar com usuários, de maneira a avaliar qualitativamente as abordagens de controle. Os resultados mostram que o modelo proposto pode gerar animações variadas e com resultados visuais satisfatórios.

Palavras-chave: Animação Facial, Controle de Avatares, Controle por *Script*, Animação Dirigida por Performance

GENERATING PERSONALIZED FACIAL ANIMATIONS FOR AVATARS

Abstract

In this work, we present a model to generate personalized facial animations for avatars. The model includes two approaches of parametric control of the avatar's face: based on high-level face actions scripts or using Performance-driven Animation (PDA) technique. The script approach allows the generation of animations with facial expressions, lip synchronization and eye behaviors interactively by high-level description of these face actions. The Performance-driven approach allows to reflect the user face expressions in his/her avatar, considering as input a small set of feature points provided by Computer Vision tracking algorithms. The model is based on MPEG-4 Facial Animation standard of parameterization, structured using a hierarchy of the animation parameters for the PDA approach. It allows us to provide animation to face regions where lack Computer Vision data. To deform the face, we use Radial Basis Functions and Cosine Functions. This scheme is computationally cheap and allow to animate the avatars in real-time. We have developed an extensible facial animation framework capable of easily animating MPEG-4 parameterized faces for interactive applications. We also developed some "case studies" interactive applications and performed a preliminary evaluation with subjects in order to evaluate qualitatively our control approaches. Results show that the proposed model can generate varied and visually satisfactory animations.

Keywords: Facial Animation, Avatar Control, Script Control, Performance-Driven Animation.

Lista de Figuras

2.1	Imagens dos experimentos de Parke e sua topologia “ótima” de face (55) . . .	25
2.2	Técnicas de animação facial paramétricas (taxonomia de Pandzic (51)). . . .	26
2.3	MPEG-4 Feature Points (FPs)	28
2.4	FAPU	29
2.5	Exemplo de deformação na pálpebra da face 3D, mostrando o ponto de controle (FP, em rosa) e a sua zona de influência (pontos azuis), que sofre a deformação.	30
2.6	Painel ilustrando algumas das abordagens de geração de deformação de trabalhos de pesquisa.	31
2.7	Exemplo de Animação Dirigida por Performance	33
3.1	Exemplo de resultados obtidos pelo trabalho de animação dirigida por performance de Buenaposada (10)	38
3.2	Da esquerda para a direita: inicialização do algoritmo de rastreamento, as unidades de ação mapeadas para MPEG-4 FPs, um exemplo rastreamento e sua animação correspondente (77).	39
3.3	Uma visão geral das etapas do modelo de Quax (31), mostrando, abaixo, exemplos de resultados visuais obtidos.	39
3.4	Exemplo de resultado da tese de Sanchez (74)	40
3.5	Exemplo de resultado do trabalho de Dutreuve (26)	40
4.1	Arquitetura geral do modelo, mostrando suas principais etapas e modelos de geração das animações.	46
4.2	Etapas do modelo de <i>Geração/Visualização da Animação Facial</i>	47
4.3	Pontos de Controle no Modelo 3D	47
4.4	Zonas de Influência	48
4.5	Os 8 pontos característicos da parte interna dos lábios, seus parâmetros de animação na direção dos eixos x e y e seus FAPUs correspondentes. Os parâmetros de animação em vermelho correspondem aos que foram incluídos aos parâmetros-padrão do MPEG-4 (verdes e azuis).	48
4.6	Ilustração dos elementos de uma região para a aplicação de uma RCF. O ponto rosa representa o ponto de controle (FP) e o verde é um dos pontos mais distantes do FP (já que a zona é circular nesse exemplo).	50
4.7	Ilustração dos elementos de uma região para a aplicação de um RBF, chamadas de GDE por (89). O ponto vermelho representa o ponto de controle na sua posição inicial, enquanto o azul representa o ponto de controle após o deslocamento. Os pontos verdes são os pontos-âncora, que delimitam o GDE.	51

4.8	Etapas do modelo de <i>Geração de Comportamento dos Olhos</i>	53
4.9	Arquitetura do modelo de <i>Geração de Expressões Faciais</i>	54
4.10	Arquitetura do modelo de <i>Sincronização da Fala</i>	55
4.11	Exemplo de animações geradas através de algumas ações faciais da FDL. .	58
4.12	Diagrama do Modelo PDA	59
4.13	Conjunto de pontos característicos utilizado.	60
4.14	Exemplo de FAPU do usuário	61
4.15	Hierarquia de FAPs	62
4.16	Exemplo do uso de informações “alto-nível” para melhorar a animação: à esquerda, a pose de usuário, que foi rotulada (manualmente) para o estado de boca “fazendo bico”. No centro, a animação gerada sem levar em consideração o estado de boca (relacionamento entre FAPs padrão). À direita, a animação gerada levando em consideração o estado de boca.	64
4.17	Exemplo de resultados da reflexão por PDA	64
5.1	Esquema de um arquivo FAP.	67
5.2	Exemplos de animações geradas em tempo-real, capturadas diretamente da <i>webcam</i> , utilizando-se algoritmos de detecção que não exigem marcadores em um protótipo de ambiente colaborativo.	68
5.3	<i>Snapshots</i> do filme <i>Tracking is Believing</i>	69
5.4	<i>Snapshots</i> das aplicações desenvolvidas com o <i>framework</i>	70
5.5	Parâmetros do Espelho Virtual	71
5.6	Resultados da Reflexão	72
5.7	Atores e Avatares utilizados	73
5.8	Exemplo da performance de dois atores	74
5.9	Exemplos de questões da Parte 1 do questionário.	75
5.10	Exemplo de questão da Parte 2 do questionário.	76
5.11	Exemplos de questões da Parte 3 do questionário.	77
5.12	Médias dos acertos dos grupos de voluntários por partes do questionário. . .	79
5.13	Comparação das médias entre os grupos de usuários que são o os que não são da Computação, dos que trabalham ou não com CG ou PI e dos que conhecem ou não os atores, respectivamente.	80
5.14	Médias das porcentagens de acertos dos grupos de usuários por cada parte do questionário.	81

Lista de Tabelas

2.1	Definição da FAPU (51)	29
5.1	Médias e desvios-padrão dos acertos pelos voluntários por partes do questionário.	78
A.1	Definições dos FAPs.	95
A.2	Grupos dos FAPs.	99
A.3	Definições dos novos FAPs.	100
B.1	Relação Fonema-Visema pelo padrão MPEG-4 (51)	101
B.2	Mapeamento Fonemas-Visemas MPEG-4 proposto por Rodrigues (68)	102

Lista de Abreviaturas

CG	Computação Gráfica
RV	Realidade Virtual
PDA	<i>Performance-driven Animation</i>
CV	<i>Computer Vision</i>
AU	<i>Action Unit</i>
FP	<i>Feature Point</i>
FAP	<i>Facial Animation Parameter</i>
FAPU	<i>Facial Animation Parameter Units</i>
FDP	<i>Face Definition Parameters</i>
FAT	<i>Facial Animation Table</i>

Sumário

1. Introdução	21
1.1 Objetivos	23
1.1.1 Objetivos Específicos	23
1.1.2 Estrutura da Dissertação	23
2. Fundamentos	25
2.1 Faces 3D Animáveis	25
2.2 Técnicas de Animação Facial	26
2.3 Parametrização	26
2.3.1 O Padrão MPEG-4 de Animação Facial	28
2.4 Modelos para a Deformação da Face	31
2.5 Modelos para Controle dos Parâmetros	32
3. Trabalhos Relacionados	35
3.1 <i>Frameworks</i> de Animação Facial	35
3.2 Animação Facial Dirigida por Performance em Tempo Real	37
3.3 Contextualização do Trabalho no Estado-da-arte	40
3.3.1 Modelo de Parametrização	40
3.3.2 Modelo de Deformação de Faces	41
3.3.3 Técnicas de Controle	41
3.3.4 <i>Framework</i>	41
3.3.5 Avaliação	42
4. Modelo	45
4.1 Geração e Visualização da Animação Facial	45
4.1.1 Parâmetros de Descrição	46
4.1.2 Os Parâmetros de Animação	47
4.1.3 Cálculo da FAPU	49
4.1.4 Deslocamentos dos FPs e Deformação das Regiões	49
4.2 Controle por <i>Script</i>	52
4.2.1 Geração de Comportamento dos Olhos	53
4.2.2 Geração das Expressões Faciais	54
4.2.3 Sincronização da Fala	54

4.2.4	Sincronização dos Parâmetros de Animação	55
4.2.5	Facial Description Language (FDL)	56
4.3	Controle por Animação Dirigida por Performance	59
4.3.1	Rastreamento dos Pontos	59
4.3.2	Mapeamento pela FAPU	60
4.3.3	Relacionamento entre FAPs	62
4.4	Algumas Considerações	65
5.	Resultados	67
5.1	O Protótipo	67
5.2	Controle por Script: Estudos de Caso	68
5.3	Controle por PDA: Avaliação com Sujeitos	71
5.3.1	Metodologia	72
5.3.2	Análise dos Resultados	75
6.	Considerações Finais	83
	REFERÊNCIAS BIBLIOGRÁFICAS	87
A.	Conjunto de Parâmetros de Animação	95
B.	Mapeamento Fonema-Visema	101

1. Introdução

Hoje em dia, cresce o número de aplicações de Computação Gráfica (CG) e Realidade Virtual (RV) que utilizam humanos virtuais, cuja exigência é de modelos cada vez mais sofisticados e realistas. Entre elas, pode-se citar aplicações de mundos virtuais (como o *Second Life*¹ e o *Playstation Home*²) e jogos de simulação do cotidiano (como o *The Sims*³), que são povoados por personagens virtuais, autônomos (agentes) ou guiados (avatars). Para o sucesso dessas aplicações, esses personagens virtuais requerem uma identidade visual, o que envolve o desenvolvimento de modelos geométricos e animações diferenciadas, que lhes conferem características próprias.

Em um mundo virtual 3D, o usuário interage através de um avatar, que é um personagem virtual que o representa. Através de seus avatares, os usuários podem agir dentro do mundo virtual, interagindo com outras pessoas (avatars e/ou agentes) e compartilhando o mesmo espaço e objetos do mundo, mesmo que todos os usuários estejam em locais físicos muito distantes. Essa proposta de interação pode servir para vários tipos de aplicação, como jogos, treinamentos, educação à distância e até reuniões de negócio em uma teleconferência.

No entanto, para que uma aplicação que usa avatares seja atraente, é necessário prover a eles, além da customização da sua aparência (o que é possível ver nas aplicações de hoje em dia), a personalização de sua movimentação, de maneira que o usuário se identifique com o seu personagem e possa expressar-se da forma mais natural possível, incluindo a sua maneira de executar gestos e expressões faciais. Nesse sentido, vê-se que os avatares ainda são representações limitadas de pessoas, refletindo de forma muito abstrata e generalizada as ações dos usuários.

Filmes recentes como o AVATAR⁴ apresentam excelentes resultados visuais em termos de reflexão da ação de pessoas reais, usando modelos com grande número de polígonos e sofisticadas técnicas de *motion capture*. Entretanto, esses resultados não são obtidos em tempo real, sendo inviável sua geração em aplicações interativas.

Nesse contexto, o estudo de métodos para a geração de animações variadas e personalizadas em avatares pode ser considerado um relevante tópico de pesquisa. Focando-se na animação facial, encontra-se uma variedade de trabalhos que buscam por modelos realistas e exploram a expressividade e comunicação de forma interativa. Muitas dessas pesquisas são focadas em maneiras para prover uma boa parametrização e descrição das

¹<http://secondlife.com/>

²<http://www.us.playstation.com/PS3>

³www.thesims.ea.com

⁴<http://www.avatarmovie.com>

ações faciais (60; 14; 70; 1; 12; 13; 49; 4; 79), desde o trabalho pioneiro de Parke (55).

Além disso, existe uma variedade de trabalhos que busca a correspondência dos movimentos e o estado interno das pessoas (como o estado emocional, o humor e a personalidade). Um exemplo é o trabalho dos pesquisadores canadenses Arya, DiPaola e Zamitto (6; 90), que procura responder a seguinte pergunta: “Que pistas visuais os humanos podem usar para passar suas características de personalidade aos personagens animados?”. Em (6) são apresentadas algumas dessas “pistas”, relativas à movimentação da cabeça em conjunto com expressões faciais. Em (90), é fornecida uma visão mais abrangente da metodologia por eles usada para a associação entre ações faciais e traços de personalidade, no contexto de aplicações de jogos. Ainda nesse contexto comportamental, pode-se citar outros trabalhos que geram animações (faciais e/ou corporais) a partir de modelos de emoções e personalidade baseados em estudos de Psicologia, como (42; 34; 76; 62; 68; 3; 33). No entanto, as animações propriamente ditas são em sua maioria pré-definidas ou geradas a partir de *templates*, feitos manualmente por artistas, o que é um trabalho dispendioso.

De acordo com Tang *et al.* (77), existem três diferentes maneiras de se guiar um avatar: dirigida por texto, dirigida por fala e dirigida por performance. Respectivamente, essas abordagens estão relacionadas com técnicas de TTS (*text-to-speech*), que convertem texto em falas animadas através da geração de suas unidades de representação do som (fonemas) e suas de representação visual (visemas); que processam o som da fala e transformam em animação; e que processam vídeos de pessoas reais e geram a animação conforme o movimento delas.

Focando-se no controle do avatar usando a técnica de Animação Dirigida por Performance (*Performance Driven Animation*, PDA) (85), o grande desafio de pesquisa é mapear os movimentos de atores para modelos 3D, com o mínimo de intervenção por parte do usuário. A captura dos movimentos pode ser feita através de equipamentos especializados ou através de seqüências de imagens utilizando-se Visão Computacional (*Computer Vision*, CV) (10; 41; 21; 77). É importante ressaltar que esse tipo de trabalho não invalida ou substitui os estudos baseados na Psicologia, e pode ser visto como complementar ao desses trabalhos, uma vez que tem o potencial de fornecer dados reais para a definição de comportamentos corporais e faciais. Da mesma forma, os estudos baseados na Psicologia podem fornecer informações que auxiliem no mapeamento dos dados capturados para dados de animação (por exemplo, os componentes faciais que atuam juntos em determinada expressão), validando, corrigindo e até mesmo suprindo informações que estejam faltando para gerar uma animação mais coerente.

1.1 Objetivos

Neste contexto, o objetivo deste trabalho é prover um modelo para a geração automática de animações faciais personalizadas para avatares, em tempo real. O modelo permite duas opções de controle para a geração das animações: por *scripts* de ações faciais descritas em alto-nível ou por Animação Dirigida por Performance, utilizando uma *web cam* comum e algoritmos de Visão Computacional, buscando refletir no avatar a expressão do usuário.

1.1.1 Objetivos Específicos

Mais especificamente, são objetivos desse trabalho:

- Fazer um levantamento do Estado-da-arte sobre modelos de animação facial, parametrização e controle das animações, sendo este último com maior enfoque em modelos de PDA;
- Fazer um levantamento dos *frameworks* e/ou *engines* disponíveis (de preferência, de código aberto) para o desenvolvimento de aplicações com animação facial, e verificar a viabilidade de usar algum para o desenvolvimento do protótipo. Se não houver, implementar uma solução conforme o Estado-da-arte.
- Escolher ou definir um conjunto de parâmetros faciais adequado para o modelo de animação;
- Definir uma abordagem de controle em alto-nível de ações faciais básicas, como expressões faciais e visemas via *scripts* de animação;
- Para o controle por PDA, definir uma estratégia de mapeamento dos dados de Visão Computacional e integrar com o modelo de animação;
- Implementar um protótipo para validação do modelo;
- Realizar testes com usuários e avaliar qualitativamente os resultados obtidos.

1.1.2 Estrutura da Dissertação

A dissertação está organizada da seguinte maneira: os capítulos 2 e 3 apresentam, respectivamente, conceitos e trabalhos científicos relacionados com o tema deste trabalho, os quais em grande parte serviram para embasar o desenvolvimento do modelo apresentado no Capítulo 4. No Capítulo 5 são apresentadas algumas aplicações usando o modelo desenvolvido, mostrando o controle dos avatares pela linguagem de *script* proposta. Logo após, são apresentados alguns resultados visuais do modelo de PDA e uma avaliação qualitativa feita com usuários, que apresenta um estudo preliminar sobre o impacto da reflexão

dos movimentos pelos avatares. Por fim, as contribuições deste trabalho são apresentadas e discutidas no Capítulo 6.

2. Fundamentos

A geração de animações (faciais e corporais) personalizadas é um tema de pesquisa que está no topo do estado da arte de animação de humanos. De maneira geral, em *softwares* de animação, os animadores precisam projetar manualmente o movimento de seus personagens em cada quadro-chave (*keyframe*). A animação automática de humanos virtuais é, portanto, facilmente justificável. Técnicas como a cinemática inversa (que faz o cálculo da posição das juntas de um objeto articulado) e modelos de animação baseados em física (dinâmica) podem produzir resultados realistas e simplificar o trabalho do animador. No caso das faces, em que a maioria dos movimentos não depende de articulações, é necessário fazer a manipulação da malha tridimensional através de técnicas como a aplicação de funções que deformam um ponto específico e sua vizinhança, ou ainda a modelagem de pseudo-músculos, que conduzem a deformação de regiões da face. Mais recentemente, surgiram técnicas que fazem a captura dos movimentos de pessoas reais e tentam reproduzi-los e/ou adaptá-los nos humanos virtuais.

Esta seção apresenta uma visão geral sobre técnicas de animação facial, de maneira a contextualizar as decisões tomadas para o modelo e os termos utilizados.

2.1 Faces 3D Animáveis

O trabalho considerado pioneiro em modelagem e animação de faces no computador é o de Frederick Parke (55). Parke fez um estudo de faces e expressões faciais em voluntários, que tiveram seu rosto pintado com uma malha poligonal, a fim de determinar um layout “ótimo” dos vértices, como mostrado na Figura 2.1. Satisfeito com a topologia encontrada no experimento, ele utilizou um método fotogramétrico para recriar a malha pintada no rosto em uma malha 3D. Com este estudo, Parke descobriu que a configuração ótima dos vértices e arestas é a que segue as linhas dos músculos faciais. Esta pode ser considerada uma heurística básica que continua sendo utilizada até hoje pelos artistas e softwares de modelagem 3D para produzir faces animáveis realistas.

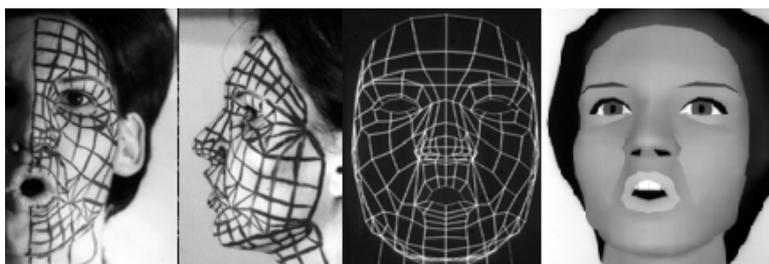


Figura 2.1: Imagens dos experimentos de Parke e sua topologia “ótima” de face (55)

2.2 Técnicas de Animação Facial

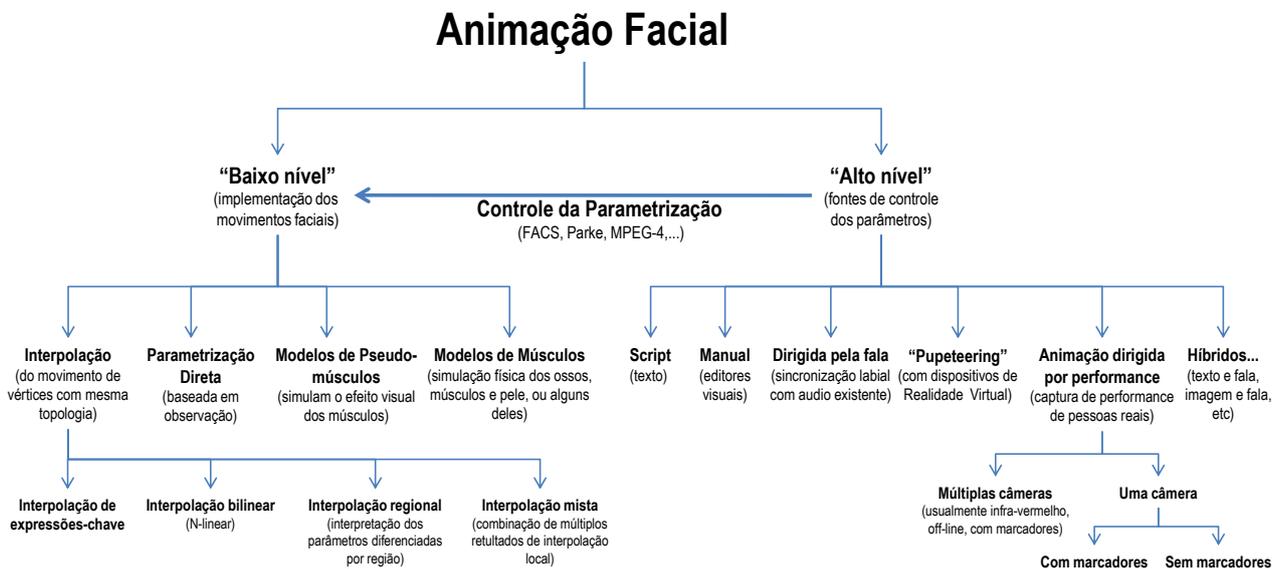


Figura 2.2: Técnicas de animação facial paramétricas (taxonomia de Pandzic (51)).

O estudo de técnicas de parametrização facial, que produzam resultados realistas em aplicações de animação computacional, é uma área ativa de pesquisa desde o trabalho pioneiro de Parke (56). A Figura 2.2 mostra a taxonomia apresentada por Pandzic *et al.* (51), que divide as técnicas de parametrização de animação facial em dois ramos:

- **Baixo-nível**, que incluem os modelos computacionais que fazem a mudança da geometria e/ou aparência da face no tempo, baseada no conjunto de parâmetros; e
- **Alto-nível**, que incluem os modelos e métodos que fazem a codificação das ações faciais desejadas para o conjunto de parâmetros, tentando capturar a essência dos movimentos em termos de intensidades, direções e velocidade.

As próximas seções apresentam alguns modelos de parametrização, com destaque ao padrão MPEG-4, que foi o escolhido para o desenvolvimento deste trabalho. Logo após, é apresentada uma visão geral de modelos para deformação das faces 3D (modelos “baixo-nível”) e das técnicas “alto-nível”, com enfoque na investigada neste trabalho, que é a Animação Dirigida por Performance. Como fontes mais completas sobre técnicas de animação facial, recomenda-se a leitura de *surveys* como (36; 23; 30).

2.3 Parametrização

Um bom conjunto de parâmetros faciais (que também é um tópico de pesquisas) é essencial para produzir a variabilidade e controle das animações. Uma parametrização ideal

deve especificar qualquer expressão facial possível através da combinação de valores independentes (89). Na literatura, encontram-se algumas abordagens de parametrização da face, para a descrição dos movimentos faciais, descritas a seguir (51).

O sistema FACS (*Facial Action Coding System*), proposto por Ekman (29), codifica as expressões faciais humanas através da determinação dos conjuntos de músculos responsáveis pelas mudanças faciais. Nesse sistema, são definidas Unidades de Ação (*Action Units*, AUs), representadas pela contração e relaxamento dos músculos faciais (ou conjunto de músculos). É um sistema inicialmente desenvolvido para estudos psicológicos, para avaliar as expressões faciais humanas. Porém, pela sua abordagem sistemática, fundamentada nas atividades musculares e descrição detalhada dos efeitos visuais de cada AU, o FACS torna-se atrativo para os pesquisadores de animação facial, sendo usado como base da parametrização em vários trabalhos (61; 77).

Outro tipo de parametrização que tem uma grande influência nos trabalhos de animação facial é o desenvolvido por Parke (56). Parke desenvolveu seus parâmetros faciais baseados na observação e análise da face em movimento, e dividiu os parâmetros em parâmetros de expressão (para animação) e de conformação (para modelagem). Apesar do seu modesto conjunto de parâmetros, seu modelo consegue gerar uma variedade de faces e expressões. O modelo de Parke é usado por pesquisadores em trabalhos como (17; 24).

Magnet-Thalmann *et al.* (78) propõe uma parametrização baseada em pseudo-músculos, chamada *Abstract Muscle Actions* (AMA), inspirada no sistema FACS. Os AMAs simulam a ação de um ou mais músculos, não independentes, cuja ordem de execução é importante. O esquema compreende dois controles de alto-nível (para visemas e expressões), compostos por AMAs de baixo-nível.

Kalra *et al.* (38) introduz uma abordagem para animação facial baseada em camadas de abstração. Os elementos principais de sua parametrização são os chamados *Minimal Perceptive Actions* (MPAs), que descrevem os efeitos visuais de uma ou mais ações musculares, que podem ser combinados em diferentes níveis de abstração. Essas “ações mínimas” incluem ações como, por exemplo, “levantar da sobrancelha direita” e “boca fazendo bico”, que recebem valores normalizados de 0 a 1 para MPAs unidirecionais e -1 a 1 para bidirecionais. Esses valores máximos e mínimos são dependentes da implementação e do modelo 3D em particular, portanto não é um esquema portátil de uma face para outra.

Em 1999, foi lançado o padrão MPEG-4¹ de Animação Facial, que é um esforço de centenas de pesquisadores e engenheiros de padronizar os parâmetros de animação facial para a comunicação e integração de conteúdos de multimídia. No processo de criação do padrão, o grupo inicialmente adotou a proposta dos MPAs de Kalra. No decorrer do processo, foram definidos o conjunto de parâmetros de animação e o esquema de normalização que os torna “portáveis” conforme mostra a próxima seção. O MPEG-4 ainda provê

¹O MPEG-4 é um padrão ISO/IEC pertencente à família MPEG (Moving Picture Experts Group)

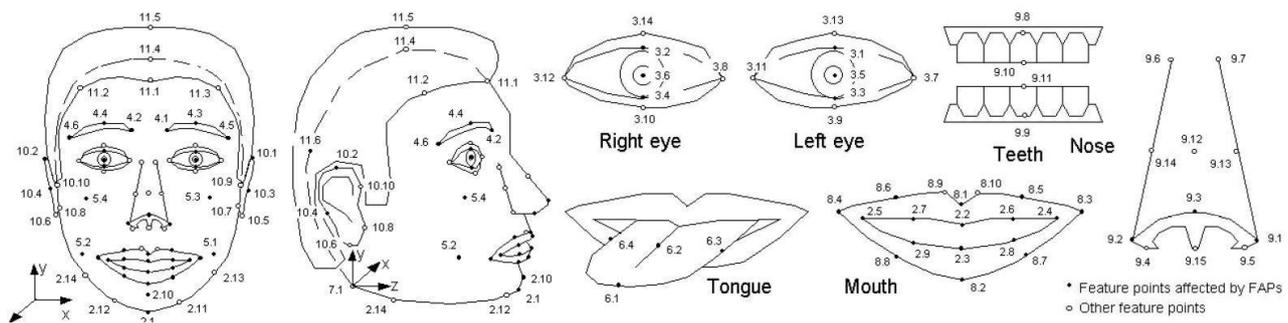


Figura 2.3: MPEG-4 Feature Points (FPs)

esquemas de codificação e compressão para o envio dos dados pela rede, que não serão abordados nesse trabalho.

2.3.1 O Padrão MPEG-4 de Animação Facial

O padrão MPEG-4 de Animação Facial especifica um conjunto de 84 pontos característicos (*Feature Points* – FPs) localizados na malha da face. A Figura 2.3 mostra esses pontos característicos. Um subconjunto desses pontos atua como pontos de controle para os 68 parâmetros de animação (*Facial Animation Parameters*, FAPs), também definidos pelo padrão.

Os dois primeiros FAPs descrevem ações em alto-nível (6 expressões faciais e 14 visemas) e os restantes lidam com regiões específicas da face, descrevendo ações mais baixo-nível como “levantar o canto direito dos lábios” e “fechar a pálpebra superior esquerda”. Os FAPs são codificados como valores numéricos, que são normalizados por um conjunto de unidades baseadas nas distâncias entre alguns pontos característicos principais da face, chamadas FAPU (*Facial Animation Parameter Units*). A Figura 2.4 e a Tabela 2.1 apresentam essas distâncias e unidades. Com essa normalização, é possível animar faces com diferentes tamanhos, proporções e número de polígonos. Uma descrição mais completa dos FAPs e suas respectivas FAPUs pode ser vista no Anexo A desta dissertação.

Para a determinação da FAPU, o modelo da face deve estar em estado “neutro”, ou seja:

- olhar na direção do eixo z ;
- todos os músculos da face devem estar relaxados;
- as pálpebras devem estar tangentes à íris;
- a pupila deve medir um terço do diâmetro da íris;
- os lábios devem estar em contato e os cantos dos lábios na mesma altura;

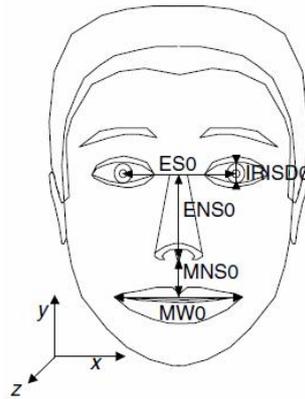


Figura 2.4: FAPU

Tabela 2.1: Definição da FAPU (51)

$IRISD_0$	Diâmetro da íris (por definição, é igual a distância entre as pálpebras superior e inferior)	$IRISD = \frac{IRISD_0}{1024}$
ES_0	Separação entre os olhos	$ES = \frac{ES_0}{1024}$
ENS_0	Separação entre o nariz e os olhos	$ES = \frac{ENS_0}{1024}$
MNS_0	Separação entre o nariz e a boca	$MNS = \frac{MNS_0}{1024}$
MW_0	Largura da boca	$MW = \frac{MW_0}{1024}$
AU	Unidade de ângulo	$10^{-5}rad$

- a boca deve estar fechada, de maneira que os dentes superiores e inferiores estejam em contato;
- a língua deve estar plana na horizontal, e com a ponta tocando o limite entre os dentes superiores e inferiores.

Gerar uma animação baseada em FAPs consiste em prover, para cada frame de animação, a variação dos valores dos FAPs. Para cada frame, tem-se então uma *stream* de valores dos FAPs, que é processada pela aplicação para gerar as animações na face, e pode ser enviada pela rede ou salva em arquivos para posterior animação.

É importante ressaltar que o padrão MPEG-4 apenas sugere os parâmetros envolvidos na animação de faces, e não os métodos para deformá-las. Tendo um conjunto de valores de parâmetros, é necessário deformar os vértices da face para produzir a animação. Por exemplo, no padrão MPEG-4, cada FAP atua sobre um FP, que por sua vez influencia os vértices de sua vizinhança (que dependem da topologia da face e não são especificados pelo padrão), produzindo uma deformação na malha poligonal. Isso significa que cada valor de FAP é escalado pela sua FAPU para se obter o deslocamento do seu ponto de

controle FP, e os vértices de sua zona de influência podem ser deformados pela aplicação de diferentes técnicas (técnicas “Baixo-nível” na Figura 2.2). A Figura 2.5 (à direita) mostra a atuação do FAP *close_top_left_eyelid*, utilizando-se uma função gaussiana sobre o FP e sua vizinhança.

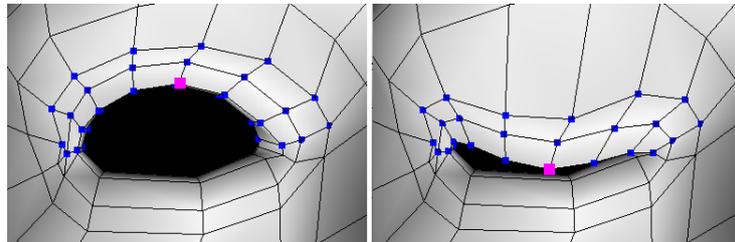


Figura 2.5: Exemplo de deformação na pálpebra da face 3D, mostrando o ponto de controle (FP, em rosa) e a sua zona de influência (pontos azuis), que sofre a deformação.

A informação sobre a FAPU, FPs e suas zonas de influência de um modelo 3D são os chamados Parâmetros de Deformação Facial (*Face Definition Parameters – FDP*), , normalmente descritos em arquivos de configuração da face. O FDP pode incluir, opcionalmente, informações sobre a deformação de cada FAP. O padrão MPEG-4 chama essa informação de FAT (*Facial Animation Table*) . Cada modelo pode ter sua própria FAT, que pode ser modelada por artistas, como a proposta de Garchery e Thalmann (32). Outra proposta é indicar no FDP do modelo a região de influência (índices dos vértices) dos FPs e a indicação de qual função de deformação deve ser aplicada para cada FAP que atua sobre aquele FP (7).

Segundo (51), para um modelo de animação facial ser conforme o padrão, ele deve ser capaz de interpretar todos os FAPs como na sua especificação. Portanto, o modelo de face deve ter um número de vértices pelo menos correspondente aos pontos característicos necessários para sua animação. O padrão especifica que o modelo de face deve ter no mínimo 50 vértices. No entanto, recomenda-se que, para alcançar resultados visuais satisfatórios, o modelo tenha pelo menos 500 vértices.

Assim, para construir uma face animada conforme o padrão, é necessário seguir 4 passos (51):

1. Construir um modelo de face e definir a localização dos FPs, de acordo com a Figura 2.3 e as localizações especificadas pelo padrão (51);
2. Para cada FAP, deve ser definido como os FPs devem se mover. Para a maioria dos FPs, o padrão define somente a movimentação em uma dimensão. É deixado a critério do desenvolvedor associar a esses movimentos outras dimensões.
3. Depois que a movimentação de cada FP é definida para cada FAP, deve-se definir como esse movimento afeta os vértices da vizinhança

4. Para expressões e visemas, o padrão define apenas dicas qualitativas sobre como eles deveriam ser projetados. Esses FAPs devem ser desenvolvidos com cuidado, porque eles serão geralmente usados para a geração de animações visualmente aparentes.

2.4 Modelos para a Deformação da Face

Nos modelos paramétricos de animação facial, os vértices da superfície (“pele”) do modelo 3D são governados por um conjunto de parâmetros de controle (como os FPs no MPEG-4). As técnicas “alto-nível” fornecem esses parâmetros, indicando, por exemplo, **o quanto** (intensidade) e **em que direção** eles modificam a face em determinados instantes de tempo. Tendo esses parâmetros, as técnicas “baixo-nível” especificam **como** a superfície poligonal da face vai deformar: como esses parâmetros de controle vão mover os vértices da face. Esse controle pode ser feito através de diferentes métodos de interpolação, que atuam diretamente sobre a superfície da face (89; 74; 26), baseada em observações (54), ou ainda em estruturas auxiliares que simulam a ação de músculos na superfície da pele (83; 11), podendo ser baseados em Física, simulando a elasticidade do tecido da pele(16; 88) e interações multi-camadas, simulando ossos, músculos e pele (45; 40; 91).

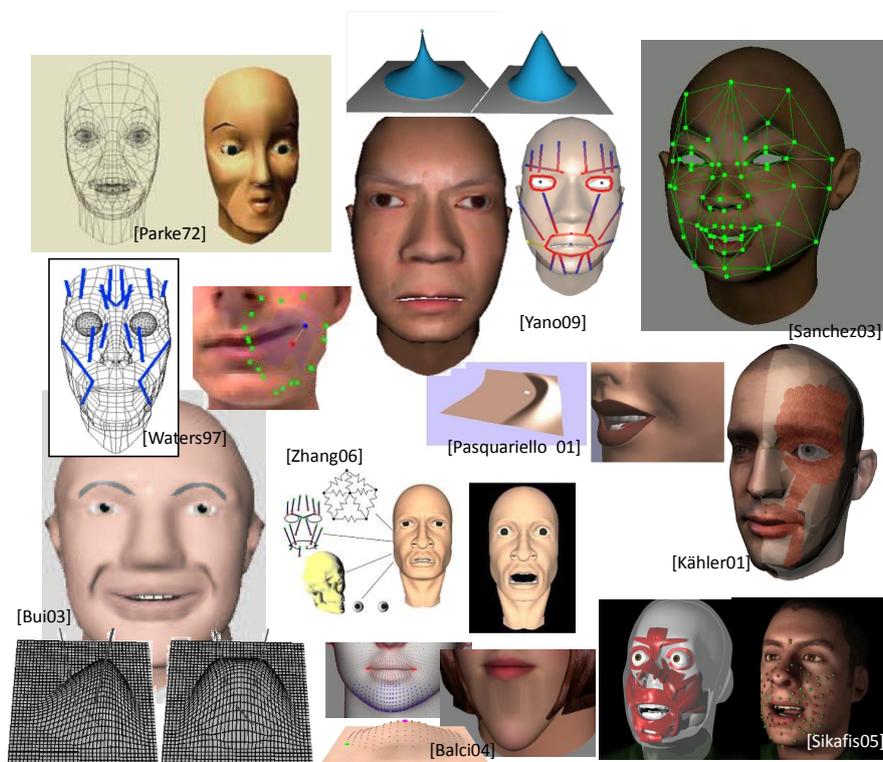


Figura 2.6: Painel ilustrando algumas das abordagens de geração de deformação de trabalhos de pesquisa.

Outro aspecto importante na geração de animações de face convincentes são as rugas

de expressão. Segundo Oat (50), as rugas de expressão ajudam a desambiguar certas expressões. Encontram-se dois tipos de abordagem para a geração das rugas: *baseada em geometria*, em que as deformações são feitas através da movimentação dos vértices da malha poligonal (58; 11); e *baseada em textura*, em que as deformações são geradas através da perturbação das normais dos vértices (utilizando-se técnicas de *bump-mapping* e *normal mapping* (9)), sendo aplicadas somente na textura do modelo (58; 50; 74; 27). Essa abordagem possui como vantagem a geração de resultados visuais adequados mesmo quando o modelo possui poucos vértices.

2.5 Modelos para Controle dos Parâmetros

De acordo com a taxonomia de Pandzic (Figura 2.2), as técnicas “alto-nível” são os modelos que fazem a obtenção e controle dos parâmetros de animação. Essas abordagens vão desde a edição manual em uma ferramenta gráfica de edição, passando por modelos que recebem *scripts* de ações em alto-nível e as transformam em parâmetros de animação (discutidos no próximo capítulo); animações dirigidas por fala (17; 35; 19; 92; 68), onde há o processamento do som e extração dos parâmetros de animação; animações dirigidas por dispositivos de hardware que atuam como “fantoques” mecânicos, como o Waldo (81; 75); até animações dirigidas por performance, onde são capturados movimentos humanos e transformados em parâmetros de animação.

Na Animação Facial Dirigida por Performance, procura-se mapear os movimentos capturados de pontos característicos da face de pessoas reais para modelos 3D (85). É notável o quanto esse tipo de técnica tem sido visada na indústria de entretenimento, sendo oferecido atualmente por empresas, como a Image Metrics², que grava a performance dos atores, sem qualquer marcação no rosto, e produz animações faciais de alta qualidade visual (veja Figura 2.7, primeira figura à esquerda), sendo este serviço utilizado para produção de filmes e animações pré-definidas em jogos.

Filmes recentes, como o AVATAR, também fazem a captura do movimento da face dos atores para gerar as animações, normalmente em ambientes controlados, dispositivos de captura de alta tecnologia e marcações no rosto (como mostra a Figura 2.7, figura do meio³ e à direita⁴). No entanto, esses processos utilizados na indústria do cinema geralmente não produzem as animações em tempo real (não há a exigência), e tão pouco com processos automatizados. No caso do filme AVATAR, a tecnologia por eles usada permite uma pré-visualização dos movimentos dos atores em tempo real, de maneira simplificada, mas os dados faciais são gravados para edição posterior por artistas.

Um típico sistema de captura de face é geralmente construído com múltiplas câmeras

²<http://www.image-metrics.com>

³Fontes: Wikipedia ([http://en.wikipedia.org/wiki/Avatar_\(2009_film\)](http://en.wikipedia.org/wiki/Avatar_(2009_film)))

⁴Fonte: VizWorld (<http://www.vizworld.com/2009/12/art-motion-capture-avatar/>)



Figura 2.7: Exemplo de um quadro da animação produzida pela empresa Image Metrics e imagens da produção do filme AVATAR.

calibradas e marcadores pelo rosto da pessoa, como em (22). É interessante ainda citar o uso de scanners 3D para a captura, como no recente trabalho de Waise *et al.* (84), que gera animações de qualidade em tempo real. Essas abordagens possuem a vantagem de capturar os dados faciais de forma precisa, porém não utilizam dispositivos de baixo custo que pudessem, atualmente, ser usados em aplicações com avatares.

Outras abordagens utilizam apenas uma câmera e marcadores no rosto (41; 26), sendo essa a configuração do estado-da-arte das pesquisas em trabalhos mais voltados para animação em tempo-real para avatares. Avanços em algoritmos de CV têm permitido o rastreamento de alguns pontos sem marcadores, como no trabalhos de (10; 77). Os trabalhos que utilizam uma câmera e CV para animação em tempo-real são descritos na Seção 3.2. Vê-se que a geração de animações fidedignas em tempo real, sem o uso de marcadores ou outros métodos intrusivos de captura e ambientes controlados, é um desafio que está no topo do estado-da-arte das pesquisas.

3. Trabalhos Relacionados

Este capítulo apresenta o levantamento dos trabalhos relacionados com os objetivos do modelo proposto. Primeiro, são apresentados trabalhos de descrição de *frameworks* de animação facial, que procuram prover uma plataforma desejável para o desenvolvimento de pesquisas e aplicações. Logo após, são apresentados trabalhos que usam, como fonte de controle dos parâmetros de animação, modelos de Animação Facial Dirigida por Performance, fazendo a geração das animações em tempo real.

3.1 *Frameworks* de Animação Facial

Conjuntos de parâmetros como o FACS e o MPEG-4 provêm uma base um tanto abstrata e “baixo-nível” para os animadores (52). Por isso, existem vários esforços para produzir uma abordagem paramétrica um pouco mais “alto-nível”, de maneira efetiva. Esse tipo de trabalho envolve diferentes níveis de abstração, e são propostas interfaces para o controle da geração das animações, que vão de linguagens de descrição a ferramentas visuais de edição.

Através de uma linguagem de descrição, o usuário pode definir um cenário completo de animação, não manipulando diretamente quadros-chave, mas por descrição textual, onde expressões e ações são os seus blocos de construção básicos (47). Na literatura, encontram-se várias linguagens de descrição, propostas por pesquisadores para descrever ações do corpo e face dos personagens. A maioria delas é baseada em XML (por exemplo, AML, APML, CML, VHML, SMIL e SMIL-Agent (42; 13; 2; 48; 70; 49)) e proveêm diferentes canais de controle e níveis de abstração, como fala, expressões faciais, ou especificação direta de alguns parâmetros de animação.

Outros trabalhos descrevem metodologias e modelos para o desenvolvimento de seus *frameworks* ou sistemas para a geração de animações faciais, procurando prover uma plataforma desejável para o desenvolvimento de pesquisas na área e aplicações.

Perlin (60), em seu sistema *Responsive Face*¹, apresenta uma abordagem multi-camadas para a representação de expressões faciais. Primeiro, foram definidos parâmetros “baixo-nível” para a descrição de movimentos, contendo um número discreto de graus de liberdade. A idéia é que, através da combinação desses parâmetros podem ser construídos movimentos mais “alto-nível”, que por sua vez também podem ser combinados, obtendo camadas mais altas de abstração. O modelo em si é bastante simples, porém expressivo, construído com base em observações. Além disso, por ser um sistema disponível, torna-se uma plataforma de animação que pode ser estendida e usada em outros trabalhos, como

¹ *Applet* Java disponível em <http://mrl.nyu.edu/~perlin/facedemo/>

(69; 67).

Cassel *et al.* (14) propõe um sistema chamado BEAT (*Behavioral Expression Animation Toolkit*), que permite aos animadores entrarem com um texto a ser dito pelo personagem virtual, e obter comportamentos não-verbais sincronizados com a fala (descrição textual) enviada, através de um modelo que faz a análise pragmática do discurso e um conjunto de regras que definem os comportamentos de acordo com essa análise. Esses parâmetros comportamentais e de síntese da fala são fornecidos em forma de uma linguagem de descrição, que pode ser interpretada por diferentes sistemas de animação.

O trabalho de Wang (82) é mais focado à síntese das animações, descrevendo uma metodologia para a construção de um sistema de animação facial com expressões faciais e sincronização labial usando componentes acessíveis no mercado, como o software *Face-Gen Modeller*² para a geração das malhas-chave de face (para cada visema) e a *Microsoft Speech SDK*³ para a API de fala.

Outro trabalho que descreve a metodologia de um sistema para a criação de faces animáveis é o de Haber *et al.* (35). O modelo de animação por eles proposto é baseado na simulação física da interação entre estruturas de caveira, músculos e pele. O sistema por eles proposto permite a edição de faces, ajustando as malhas poligonais a essas estruturas auxiliares, e a geração de falas sincronizadas por meio da descrição de fonemas. As animações geradas podem ser renderizadas em tempo-real (interpolação dos *keyframes* gerados pelo sistema).

Cosi (18) propõe um *toolkit* de animação facial implementado em MATLAB⁴, criado principalmente para acelerar o processo de criação do agente conversacional LUCIA (20). Para a determinação das trajetórias dos pontos de controle da animação, é utilizado um equipamento especializado de *motion capture*. A tradução para os parâmetros de animação do padrão MPEG-4 se dá por ajuste manual na interface do programa de rastreamento, e para a síntese das animações de boca, é utilizado um modelo de coarticulação labial descrito pelos autores em (19). Apesar dessa abordagem parecer promissora, ela é construída sobre um equipamento de hardware específico e software proprietário.

DiPaola e Arya (25) propõem um *framework* de animação facial, também compatível com o padrão MPEG-4, chamado iFace, cujos binários estão disponíveis na *web*⁵. O iFace permite a criação de expressões não-verbais através de uma linguagem de *scripting* chamada FML (Face Modeling Language) (5), que permite a descrição de ações faciais sequenciais e paralelas. As ações faciais incluem fala, expressões faciais, movimentos de cabeça e os parâmetros MPEG-4 diretamente. Atualmente, o iFace está sendo utilizado para estudos de animação comportamental com o objetivo de alcançar a associação entre as ações faciais

²www.facegen.com/modeller.htm

³<http://www.microsoft.com/speech/download/sdk51>

⁴<http://www.mathworks.com/>

⁵<http://ivizlab.sfu.ca/research/iface/>

e traços de personalidade, criando um conjunto de parâmetros mais alto-nível (6).

Balci *et al.* (8) projetaram o Xface⁶, um conjunto de ferramentas de código aberto para a criação de *talking heads* usando MPEG-4 ou animação por *keyframes*. O controle da animação por key-frames é feito pela linguagem de *scripting* SMIL-Agent (49). Isso significa que, para usar este modo de animação, é necessário preparar diversas malhas com poses diferentes para cada modelo de face que pretende-se usar, e a biblioteca faz a interpolação dessas malhas-chave. O suporte dado para o modo de animação que segue o padrão MPEG-4 inclui:

- uma biblioteca principal, que inclui as funções principais de animação;
- um editor gráfico, que permite carregar modelos de face neutra e configurar seu FDP (setar os FPs e vizinhança, FAPU, funções de deformação para cada FAP), salvo como um arquivo XML;
- uma aplicação simples que interpreta arquivos com *streams* de FAPs e gera animações em tempo-real (*player*)
- uma aplicação de programa cliente, que permite o controle do *player* pela rede.

O Xface é usado em outros trabalhos de pesquisa, como (39), (87) e (68), sendo este último estende o editor gráfico, criando uma ferramenta de autoria que permite a geração de expressões faciais baseadas em um modelo emocional e fala sincronizada.

3.2 Animação Facial Dirigida por Performance em Tempo Real

Em modelos de Animação Dirigida por Performance, o mapeamento a ser feito entre os dados obtidos e os parâmetros de animação é um problema complexo, no qual os pesquisadores têm proposto diferentes métodos. Uma questão que se impõe é a transformação de movimentos contínuos, na realidade, em dados discretos, nos personagens virtuais.

No trabalho de Chai *et al.* (15), um pequeno conjunto de parâmetros é extraído de vídeos e combinado com dados de arquivos de *motion capture*. A principal contribuição desse modelo é o método que faz a associação do movimento, com poucos pontos de controle, associados com as movimentações de alta qualidade extraídas de um banco de dados de arquivos de *motion capture*.

Buenaposada *et al.* (10) estimam parâmetros de animação em alto-nível (contrações de músculos, movimentos de olhos, abertura das pálpebras, movimentação da mandíbula e contrações dos lábios) de uma sequência de imagens de face, sem marcadores, através de

⁶<http://xface.itc.it/>

um *tracker* que utiliza o modelo estatístico de aprendizagem baseado em aparência (*appearance based*). Para as animações, os autores utilizam uma versão modificada do modelo de Parke e Waters (57) que é feita em tempo real, de acordo com os parâmetros identificados. Eles utilizam animações da face 3D também para o treino do modelo estatístico. Entretanto, a adaptação para um usuário novo é parcialmente manual, pelo fato de que não foi estudado como fazer a correspondência entre as imagens do usuário e as expressões faciais do modelo 3D. A Figura 3.1 mostra alguns resultados desse trabalho.

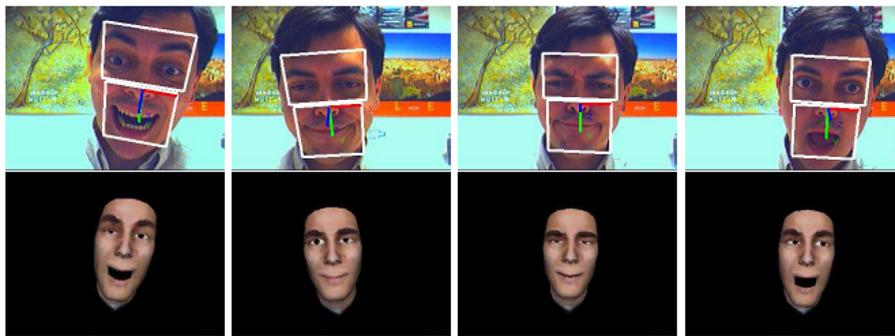


Figura 3.1: Exemplo de resultados obtidos pelo trabalho de animação dirigida por performance de Buenaposada (10)

Khanam *et. al* (39) descrevem um modelo que procura reproduzir expressões faciais detectadas de vídeos automaticamente. O modelo propõe uma técnica para melhorar a síntese das expressões faciais detectadas, através da identificação do seu contexto. As expressões faciais são modeladas como *streams* de parâmetros de animação MPEG-4. O contexto do vídeo é processado por um sistema baseado em lógica difusa, que encontra qual a expressão que precisa ser adicionada aos parâmetros da expressão facial detectada, para melhorar sua “naturalidade”.

No trabalho de Tang *et al.* (77), os parâmetros de animação (ângulos de rotação, distâncias de translação da cabeça e um conjunto de pesos de 12 unidades de ação, baseadas do sistema FACS (29)) são estimados de um vídeo usando um algoritmo de rastreamento. Os parâmetros estimados são traduzidos para os parâmetros de animação do padrão MPEG-4 (valores de FAPs) através da resolução de um problema de otimização linear. O mapeamento é feito através de uma tabela de animação (FAT) para determinar para cada FAP um conjunto de deslocamentos dos FPs. Figura 3.2 mostra algumas etapas do modelo e um exemplo de resultado aplicado a uma face 3D.

Quax *et al.* (31; 64) apresentam uma metodologia para a extração de dados relacionados com emoções de vídeos em tempo real e sua aplicação em representações estilizadas de avatares. Na etapa de visualização, os autores optaram por uma metodologia 2D estruturada, uma vez que a face na qual os movimentos faciais capturados são aplicados podem ser desenhos de cartuns 2D ou baseados em filmagens reais (Figura 3.3). O estágio de

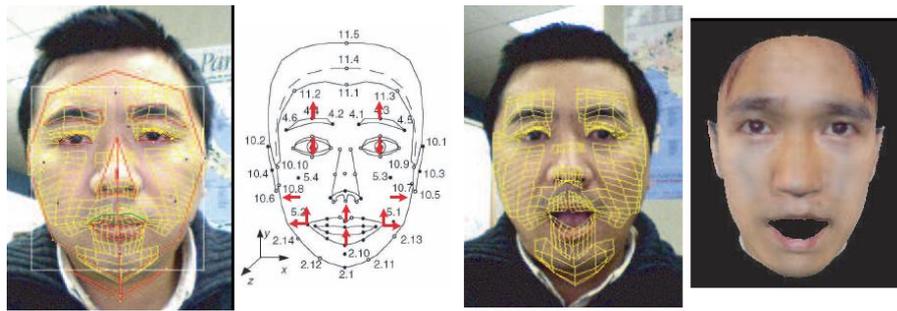


Figura 3.2: Da esquerda para a direita: inicialização do algoritmo de rastreamento, as unidades de ação mapeadas para MPEG-4 FPs, um exemplo rastreamento e sua animação correspondente (77).

modelagem demanda bastante trabalho manual para o ajuste de cada componente facial como um Modelo Hierárquico de Visualização (*Hierarchical Display Model*, HDM) e criação de poses extremas em vários pontos de vista. O mapeamento entre a pessoa real e o avatar é feito pela representação das coordenadas como um conjunto de pontos característicos do padrão MPEG-4 e combinando eles com os parâmetros de animação (FAPs). Um processo intermediário é usado para interpolar a informação dos quadros-chave que são enviados entre os usuários.

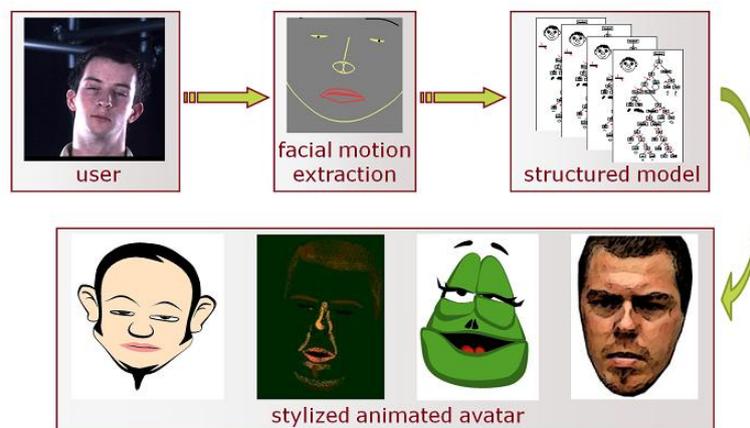


Figura 3.3: Uma visão geral das etapas do modelo de Quax (31), mostrando, abaixo, exemplos de resultados visuais obtidos.

Um outro trabalho relacionado a este é a tese de Sanchez (74), que trabalha com métodos de mapeamento de dados capturados por dispositivos de *Motion Capture* (MoCap) facial para deformações dos modelos de face 3D, como mostra a Figura 3.4. Neste trabalho, procura-se gerar automaticamente e em tempo real as animações de face, baseada em dados capturados de pessoas reais. As maiores contribuições da tese foram dois métodos de deformação de superfícies, chamados *Planar Bones* (73) e *Bèzier-Induced Deformation of Surfaces* (BIDS) (72). Esses métodos, segundo o autor, constituem num *framework* adequado para a tradução de expressões faciais para modelos poligonais, para quando o



Figura 3.4: Exemplo de resultado da tese de Sanchez (74)

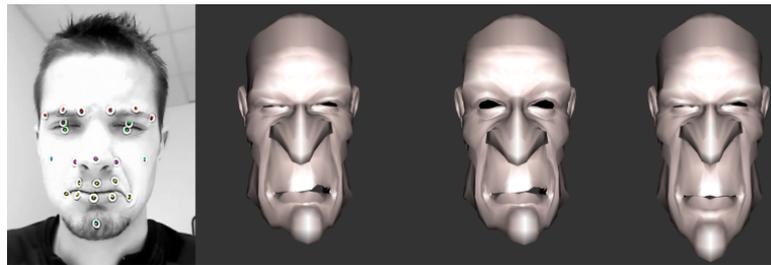


Figura 3.5: Exemplo de resultado do trabalho de Dutreve (26)

modelo de expressão correspondente é baseado no movimento espacial de uma “nuvem de pontos” de marcadores. Além disso, eles propõem uma técnica para o mapeamento dos dados de MoCap para a malha facial, através de funções de base radial (71). Eles também propõem métodos de geração de rugas por *normal-mapping*. Porém este último modelo requer que a captura do “mapa de rugas” de uma pessoa seja feito previamente.

O trabalho de Dutreve (26) (Figura 3.5) apresenta uma metodologia que considera, como entrada para a geração de animações, apenas um pequeno conjunto de pontos característicos. Estes pontos característicos são selecionados manualmente na inicialização, e uma rede de funções de base radial (*Radial Basis Functions – RBFs*) é treinada para fazer a correspondência desses pontos no modelo-fonte (vídeo do usuário) e no modelo-destino (face 3D). As zonas de influência são determinadas considerando as distâncias dos vértices aos pontos de controle, o que nem sempre produz bons resultados e pode requerer ajustes manuais.

3.3 Contextualização do Trabalho no Estado-da-arte

Essa seção apresenta a contextualização desse trabalho com o estado-da-arte levantado nesse capítulo e no anterior.

3.3.1 Modelo de Parametrização

Em relação à parametrização, o modelo proposto neste trabalho optou pelo padrão MPEG-4 de Animação Facial. Como será visto no próximo capítulo, o modelo faz a inclusão

de alguns parâmetros de animação, seguindo o modelo de definição dos FAPs. Isso faz com que o modelo não seja completamente compatível com o padrão MPEG-4 por definição. No entanto, tecnicamente, esse acréscimo não alterou os parâmetros originais. Portanto, os parâmetros gerados compatíveis com o padrão podem ser enviados para outros sistemas compatíveis com o padrão.

3.3.2 Modelo de Deformação de Faces

Considerando as técnicas de “baixo-nível” para a geração das deformações na face do avatar, optou-se em usar interpolações locais baseadas nos trabalhos de (89; 26; 7), sem a necessidade de estruturas auxiliares, como músculos.

3.3.3 Técnicas de Controle

Considerando técnicas de animação facial “alto-nível” para o controle dos avatares, este trabalho focou-se Controle por *Script* e Animação Dirigida por Performance. Para o controle de animações faciais em alto-nível, foi desenvolvido um modelo que faz a geração dos parâmetros de animação através de uma linguagem de *script*. Esse modelo foi inspirado principalmente nos trabalhos (25; 8; 68).

Sobre o estado-da-arte em PDA, pode-se notar que os pesquisadores atualmente estão concentrando seus esforços na complexa tarefa do mapeamento entre os dados capturados de pessoas reais e os parâmetros de animação. Alguns trabalhos (por exemplo, o de Tang (77)) estão mais focados nas abordagens de rastreamento dos componentes faciais, tentando imitar diretamente a performance do ator. Outros trabalhos lidam com a “tradução” de linguagens de representação, como o de Quax (64), considerando que um avatar pode ser um personagem não realista (cartum).

O modelo proposto neste trabalho utiliza uma abordagem semelhante à de (26) (animação por um conjunto de pontos de controle), porém utilizando a parametrização baseada no padrão MPEG-4 de animação facial no mapeamento e considerando que, dos pontos característicos rastreados podem ser extraídas informações que, utilizando-se um modelo de relacionamento entre os FAPs, seja possível suprir alguns parâmetros que não são fornecidos por CV e deformar de maneira coerente todas as regiões da face.

3.3.4 *Framework*

Com relação ao uso de um *framework* já existente (cuja investigação constitui um dos objetivos específicos do trabalho, com viés tecnológico, porém indispensável para a condução das pesquisas), com o levantamento na Seção 3.1 e verificação de sua disponibilidade para a comunidade científica, verificou-se que poucos deles (e alguns deles, em parte) encontram-se disponíveis (Responsive Face, LUCIA, iFace e Xface). Como a maioria dos

trabalhos apresentados tratam-se de trabalhos de pesquisa com diferentes focos, optou-se, inicialmente, em utilizar o Xface apenas como uma *engine* para a geração de animação facial compatível com o padrão MPEG-4.

O modelo começou a ser desenvolvido utilizando-se as bibliotecas principais de animação do Xface (7), estendendo a sua aplicação *player*. No entanto, apesar do Xface fornecer subsídios para os fins da pesquisa (código aberto e abstração de classes), ele apresenta algumas limitações técnicas, a saber, problemas com suas dependências e estruturação de classes complexa. Essas limitações foram perceptíveis ao longo do desenvolvimento do modelo de geração de animações por *script*, porém foi no desenvolvimento do modelo de controle por PDA que surgiu a necessidade de algumas alterações no modelo de animação facial, o que exigiria reestruturações no *core* do Xface. Nessa etapa, surgiu a necessidade do acréscimo dos parâmetros de animação e alterações/melhorias no modelo de deformação do modelo 3D, uma vez que o XFace utiliza a abordagem de apenas um ponto de controle por região, e pesos pré-definidos manualmente para sua função de deformação de cada modelo 3D. Conseqüentemente, com as limitações técnicas apresentadas, os acréscimos ao modelo de animação demandariam um muito tempo, o que tornaria inviável a continuação da pesquisa dentro do cronograma previsto.

Optou-se, então, em construir um protótipo com um modelo de animação facial próprio, que recebe os parâmetros de animação e realiza as deformações na face, conforme apresenta o Capítulo 4. É importante ressaltar que essa decisão de implementação não teve impacto no modelo de geração dos parâmetros de animação, uma vez que o Xface era usado apenas para as deformações do modelo 3D e visualização das animações. Parte deste trabalho, que já havia sido publicado como um *framework* em (66), utilizava o Xface, mas o modelo do controle para geração dos parâmetros de animação (no caso, por *script*) é independente do Xface. Portanto, se arquivos gerados com os parâmetros de animação forem “rodados” no Xface ou na plataforma de animação posteriormente desenvolvida, haverá diferenças apenas na maneira como os parâmetros são interpretados para a geração das deformações.

3.3.5 Avaliação

Com relação à avaliação, Computação Gráfica é certamente uma das áreas da computação mais difíceis de avaliar seus modelos. O problema normalmente está em como responder a seguinte questão: “o quão realista é o resultado que o modelo produz?”, sendo que, muitas vezes, não existem maneiras efetivas de fazer comparações utilizando dados quantitativos do mundo real. Thalmann e Thalmann (47) sugerem que a avaliação do realismo seja feita de acordo com o seu campo de aplicação: animação ou simulação. No contexto deste trabalho, o foco está em produzir animações convincentes, compreensíveis

aos usuários. Por esse motivo, foi realizada uma avaliação qualitativa com usuários, apresentada no Capítulo 5.

4. Modelo

Este capítulo apresenta o modelo desenvolvido, cujo objetivo é prover animações de face personalizadas e expressivas para avatares. O modelo permite duas opções de controle das animações:

1. Por *scripts* de ações faciais: visando que a aplicação pode oferecer diferentes opções de expressões faciais, audio e/ou chat, foi proposto um modelo que integra técnicas da literatura e gera animações sincronizadas com a fala, expressões faciais e comportamento dos olhos;
2. Dirigida por performance: através de dados capturados de uma *webcam* comum, foi proposto um modelo que gera animações que permitem ao avatar “refletir” a movimentação da face do usuário.

A arquitetura geral do modelo é apresentada na Figura 4.1. Ele está dividido em três etapas principais: a etapa de *Geração/Visualização da Animação Facial* a partir dos *parâmetros de animação* gerados, e duas correspondentes aos modelos das opções de controle do avatar abordados, *Controle por Script* e *Controle por Animação Dirigida por Performance*, que são os geradores dos parâmetros de animação. A descrição dessas etapas principais e seus modelos, como mostrados na Figura 4.1, são descritos nas próximas seções.

4.1 Geração e Visualização da Animação Facial

O modelo de *Geração das Deformações na Face 3D* recebe os *parâmetros de animação* de um dos modelos de controle e gera as deformações na geometria de uma *face 3D parametrizada*, que é composta pela *malha poligonal do avatar* e pelos *parâmetros de descrição do avatar* (apresentados na Seção 4.1.1). A Figura 4.2 apresenta a arquitetura geral do modelo, que inclui:

- Uma etapa de inicialização, em que é feito o cálculo das unidades de medida utilizadas para escalar os parâmetros de animação (*Cálculo da FAPU*, apresentado na Seção 4.1.3);
- O *Deslocamento dos FPs* (pontos de controle) de acordo com o valor dos parâmetros de animação, apresentado na Seção 4.1.4;
- A *Deformação das Regiões* de influência, de acordo com o deslocamento sofrido por seu(s) ponto(s) de controle, também apresentado na Seção 4.1.4;

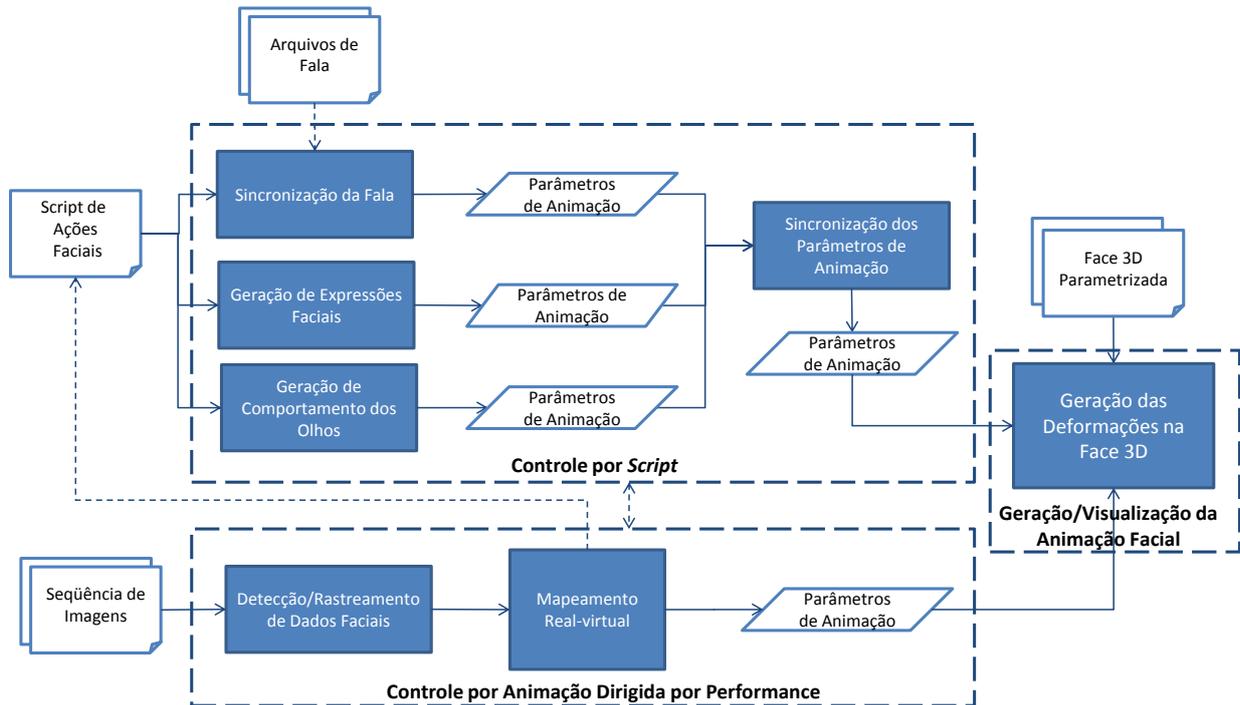


Figura 4.1: Arquitetura geral do modelo, mostrando suas principais etapas e modelos de geração das animações.

- A *Visualização* da animação, que inclui as técnicas de *rendering* da cena. No modelo, utiliza-se o padrão do OpenGL(86).

4.1.1 Parâmetros de Descrição

O modelo gera as deformações a partir dos dados referentes à geometria do avatar (malha poligonal) e dos seus seguintes parâmetros de descrição, via arquivo de configuração:

- **Pontos de Controle:** devem ser informados os índices dos vértices do modelo 3D correspondentes aos FPs do padrão MPEG-4. O ideal é que sejam fornecidos todos os FPs que possuam um parâmetro de animação associado, caso contrário, não serão geradas deformações nos pontos/regiões da face não especificados. O conjunto mínimo de pontos a serem informados são os que são usados para se fazer o cálculo da FAPU do avatar (Seção 4.1.3). O processo de captura dos índices dos FPs é manual. Figura 4.3 apresenta exemplos de faces e a localização dos seus FPs, baseada nas indicações do padrão MPEG-4.
- **Regiões de Influência:** correspondem às áreas (conjunto de vértices) que são influenciadas pelos pontos de controle. Devem ser informados os índices dos vértices da região e o conjunto de um ou mais FPs que a influencia. Neste trabalho, essas regiões também foram definidas manualmente, como mostra a Figura 4.4.

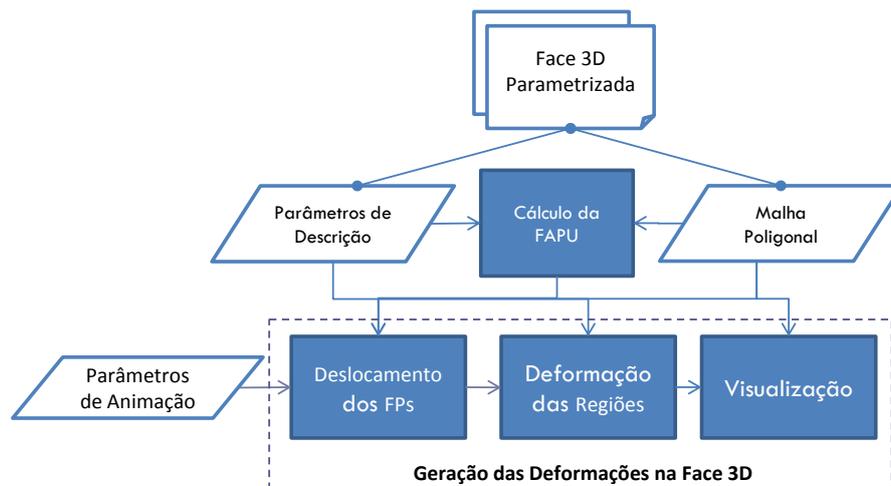


Figura 4.2: Etapas do modelo de *Geração/Visualização da Animação Facial*

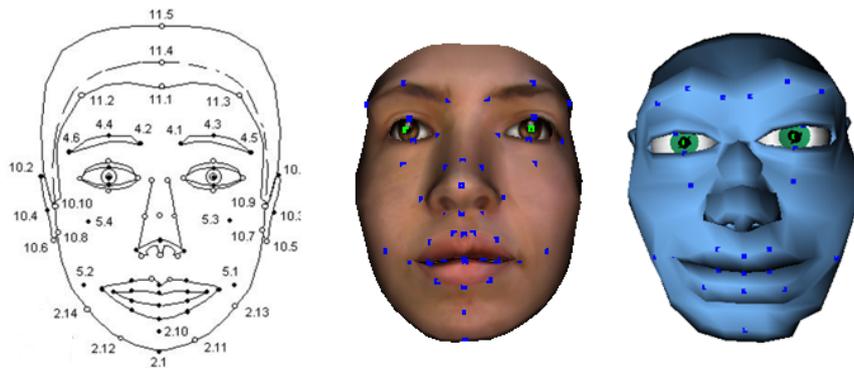


Figura 4.3: Exemplos de faces parametrizadas, mostrando a localização dos Pontos de Controle.

4.1.2 Os Parâmetros de Animação

O modelo utiliza o conjunto de parâmetros do padrão MPEG-4 (FAPs). O padrão MPEG-4 descreve, para alguns *feature points* (FPs), movimentos nas direções dos eixos x , y e z . Não necessariamente o padrão define o movimento em todos os três eixos para cada FP. Por exemplo, o FP 2.2, localizado no meio do lábio superior, possui os FAPs *LOWER_T_LIP* (movimento no eixo y) and *PUSH_T_LIP* (movimento no eixo z) associados. Mas não há um FAP para descrever o movimento no eixo x , que pode ocorrer frequentemente. De acordo com o segundo item dos passos para a construção de uma face animada por (51) na Seção 2.3.1, movimentos em que o padrão descreve em apenas uma dimensão devem ser interpretados pelo modelo de animação, deixando a cargo do desenvolvedor/pesquisador a associação com outras dimensões. Na abordagem adotada neste trabalho, a falta desses parâmetros possui um forte impacto na tentativa de se fazer a reflexão dos movimentos reais das pessoas, uma vez que esses movimentos existem (como é o caso do movimento no eixo x do FP 2.2) e, no caso da abordagem Dirigida por

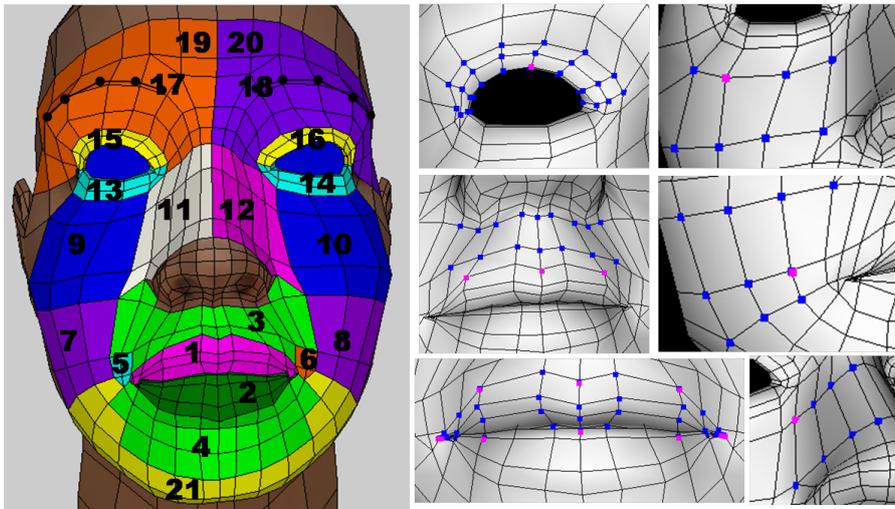


Figura 4.4: Regiões de controle definidas para as faces 3D.

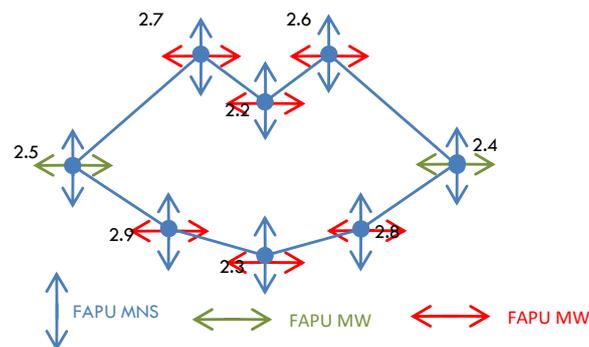


Figura 4.5: Os 8 pontos característicos da parte interna dos lábios, seus parâmetros de animação na direção dos eixos x e y e seus FAPUs correspondentes. Os parâmetros de animação em vermelho correspondem aos que foram incluídos aos parâmetros-padrão MPEG-4 (verdes e azuis).

Performance, pode se obter os dados sobre esse movimento, não necessitando de regras de associação. Por essa razão, o conjunto de parâmetros de animação deste trabalho propõe a inclusão de novos parâmetros, completando os movimentos nos eixos que não são contemplados pelo padrão MPEG-4. O Anexo A apresenta os parâmetros acrescentados em função do modelo de PDA, sendo que esse conjunto ainda pode ser estendido.

É importante observar que a inclusão desses parâmetros requer a escolha de uma FAPU que os quantifique. Para isso, seguiu-se a lógica empregada nos outros FAPs do padrão MPEG-4. Por exemplo, para a boca, movimentos no eixo x utilizam o FAPU MW (Mouth Width), e em y e z , o FAPU MNS (Mouth-Nose Separation). A Figura 4.5 mostra um exemplo da inclusão desses parâmetros (considerando apenas os eixos x e y) nos FPs da parte interna dos lábios.

4.1.3 Cálculo da FAPU

Dada como entrada do modelo a face 3D parametrizada, a primeira etapa é a determinação das unidades de medida dos parâmetros de animação, a FAPU, como descrita no padrão MPEG-4. As distâncias $FAPU_0 = \{IRISD_0, ES_0, ENS_0, MNS_0, MW_0\}$ são calculadas usando as posições dos vértices assinalados como FPs (equações 4.1 a 4.5), e depois são normalizadas como sugere o padrão MPEG-4 Tabela 2.1.

$$IRISD_0 = |FP_{3.1y} - FP_{3.3y}| \quad (4.1)$$

$$ES_0 = |FP_{3.5x} - FP_{3.6x}| \quad (4.2)$$

$$ENS_0 = |FP_{3.5y} - FP_{9.15y}| \quad (4.3)$$

$$MNS_0 = |FP_{9.15y} - FP_{2.2y}| \quad (4.4)$$

$$MW_0 = |FP_{8.3x} - FP_{8.4x}| \quad (4.5)$$

4.1.4 Deslocamentos dos FPs e Deformação das Regiões

O deslocamento dos pontos de controle (FPs ativos) se dá pelo valor do parâmetro de animação j do conjunto de FAPs (FAP_j) escalado pela FAPU do avatar (correspondente ao FAP_j):

$$\Delta FP_j = FAP_j \cdot FAPU_{avatar}(FAP_j) \quad (4.6)$$

Após deslocar os pontos de controle, cada vértice da região de influência do FP são deformados de acordo com uma função de deformação. A estratégia adotada para fazer a deformação das regiões de influência foi a seguinte:

- para regiões influenciadas por apenas um FP, aplica-se uma função de cosseno (explicada a seguir);
- para regiões contornadas por FPs (como os lábios superior e inferior) e a linha das sobrancelhas, optou-se em usar a função de base radial (explicada a seguir);
- para regiões influenciadas por mais de um FP (mas não contornadas por eles), optou-se em fazer a média da função de cosseno aplicada a cada FP.

É interessante ressaltar que essa estratégia é a que está sendo adotada no momento, no protótipo implementado. A justificativa é que essa configuração produz resultados visuais adequados. No entanto, pode-se adotar estratégias diferentes (por exemplo, utilizando funções de base radial em outras zonas, por exemplo), apenas modificando o arquivo de Parâmetros de Descrição do Avatar.

Função de Cosseno

Funções de cosseno (*Raised Cosine Functions*, ou *RCFs*) são encontradas em alguns trabalhos da literatura (19; 7) como uma maneira suave e rápida de deformar regiões influenciadas por um ponto de controle.

Seja $V_j = \{\vec{v}_1, \vec{v}_2, \dots, \vec{v}_n\}$ o conjunto de vértices do FP_j . A função de cosseno atua sobre o FP_j e cada vértice p (\vec{v}_p) de sua zona de influência de acordo com a Equação 4.7 (7),

$$F(\vec{v}_p) = \left[1 + \cos\left(\pi \cdot \frac{d_p}{d_{max}}\right) \right] \cdot w \cdot \Delta FP_j \quad (4.7)$$

onde d_p é a distância entre o vértice \vec{v}_p e o FP_j e d_{max} é a distância do vértice mais distante do FP_j na região de influência. Figura 4.6 mostra a aplicação de uma RCF em uma zona de influência circular, nos eixos x e y . Neste trabalho, w é uma constante (0, 5), que garante que, no FP, o valor de $F(\vec{v}_p)$ retorne 1, fazendo uma normalização do deslocamento dos outros pontos em relação ao ΔFP_j (Equação 4.6). É importante ressaltar que, em outros trabalhos, como o de Balci *et al.* (7), este w constitui um valor que é atribuído manualmente para cada FAP, durante a configuração dos parâmetros de definição de cada face, de maneira a tornar mais suaves as deformações. Porém, com esses valores, não se garante que no FP haverá o deslocamento desejado conforme o modelo de mapeamento apresentado na seção anterior (além de tornarem o processo manual), e conseqüentemente, isso impacta no comportamento dos outros pontos da região.



Figura 4.6: Ilustração dos elementos de uma região para a aplicação de uma RCF. O ponto rosa representa o ponto de controle (FP) e o verde é um dos pontos mais distantes do FP (já que a zona é circular nesse exemplo).

Função de Base Radial

Funções de Base Radial (*Radial Basis Functions*, ou *RBFs*) são bastante usadas em Computação Gráfica para aproximação e interpolação de superfícies. No presente trabalho, é utilizada a abordagem de Noh *et al.* (89) (que propõe a animação de faces usando RBFs), baseando-se também na explicação do trabalho de Dutreuve *et al.* (26) (que utiliza as RBFs para o mapeamento dos espaços de parâmetros real e virtual). Em ambos os trabalhos, utiliza-se a função RBF multi-quadrática.

Para fazer a deformação de uma região da face usando uma RBF, é necessário definir regiões da face compostas por um conjunto de pontos característicos e os outros vértices

da região de influência. Essas regiões são chamadas de GDEs (*Geometry Deformation Elements*) como mostra a Figura 4.7 (89). O conjunto de pontos característicos pode ter dois tipos de pontos, conforme Noh *et al.* (89) :

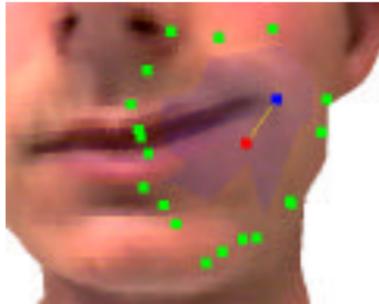


Figura 4.7: Ilustração dos elementos de uma região para a aplicação de um RBF, chamadas de GDE por (89). O ponto vermelho representa o ponto de controle na sua posição inicial, enquanto o azul representa o ponto de controle após o deslocamento. Os pontos verdes são os pontos-âncora, que delimitam o GDE.

- pontos de controle: um ou mais pontos que sofrerão os deslocamentos;
- pontos-âncora: são das bordas da região, que permanecem estáticos, e são necessários no treino da RBF, para delimitar a região;

A definição dos limites do GDE pode ser feita automaticamente, através da definição de algum critério de distância, como em (89) ou manualmente (pontos das bordas da região manualmente escolhida). Neste trabalho, as GDEs foram definidas com os pontos de controle (os FPs da região), uma vez que utilizou-se as RBFs na região dos lábios, onde os FPs delimitam a sua região (como mostrado anteriormente na Figura 4.4).

Seja $S = (\vec{s}_1, \vec{s}_2, \dots, \vec{s}_N)$ o conjunto de pontos característicos inicial ou “fonte” (*source*, correspondente à face em repouso) de um GDE. O primeiro passo é o treino da RBF a partir desse conjunto. Após esse treino, para cada conjunto de pontos característicos “alvo” (*target*, contendo as novas posições dos pontos característicos deslocados) $T = (\vec{t}_1, \vec{t}_2, \dots, \vec{t}_N)$, são calculados o conjunto de pesos $W = (w_1, w_2, \dots, w_N)$ os quais, aplicados à função RBF multi-quadrática $F(\vec{s}_j)$ para os outros pontos j da região de influência, determinam sua nova posição ($\vec{t}_j = F(\vec{s}_j)$). É usada uma RBF para cada eixo. A seguir são apresentados os cálculos de cada uma das etapas (26):

- **Treino**

Para o treino da RBF, é necessário determinar a matriz H , que contém a função multi-quadrática aplicada a todos os pares elementos do conjunto S , conforme a Equação 4.8

$$H_{ij} = h(\|\vec{s}_j - \vec{s}_i\|) = \sqrt{(\|\vec{s}_j - \vec{s}_i\|)^2 + sc_j^2} \quad (4.8)$$

Onde $sc_j^2 = \min_{j \neq i}(\|\vec{s}_j - \vec{s}_i\|)$ e trata-se de um coeficiente de “rigidez” (*stiffness*) sugerido por Eck (28) para suavizar deformações onde os pontos característicos estão muito esparsos e reforçar onde eles estão muito próximos.

- **Determinação do vetor de pesos**

Tendo a matriz H e o conjunto $T_x = (t_1^x, t_2^x, \dots, t_N^x)$ com as novas posições dos pontos característicos (t_j^x é a coordenada x de \vec{t}_j), calcula-se o vetor de pesos do sistema no eixo x , como mostrado na Equação 4.9:

$$T_x = H \cdot W_x \therefore W_x = H^{-1}T_x \quad (4.9)$$

- **Determinação das novas posições**

Uma vez treinada a RBF para cada eixo, a nova posição para cada ponto \vec{t}_j de cada um dos outros pontos \vec{s}_j da zona de influência é obtido pela função $F(\vec{s}_j)$ da Equação 4.10, que representa a RBF propriamente dita:

$$\vec{t}_j = F(\vec{s}_j) = \sum_{i=1}^N \vec{w}_i \cdot h(\|\vec{s}_j - \vec{s}_i\|) \quad (4.10)$$

4.2 Controle por Script

Esta seção apresenta a descrição detalhada o modelo que faz o *Controle por Script* dos avatares (Figura 4.1). A principal entrada para esse modo de geração das animações é um (ou mais) arquivos de *script* contendo a descrição de uma ou mais ações faciais. A linguagem de *descrição* desenvolvida é chamada FDL (Face Description Language), portanto, seus arquivos serão chamados no decorrer do texto como arquivos FDL. O modelo interpreta as ações faciais descritas nos arquivos FDL e gera animações de acordo com sua especificação. A linguagem compreende seqüências de três tipos de ações faciais: *fala*, *expressões faciais* e *comportamentos dos olhos*.

Cada um desses tipos de ações faciais são descritos de forma independente (a coerência entre o *timeline* dessas ações pode ser feita manualmente ou através de um modelo emocional que os correlacione, por exemplo), e são processados nas seguintes etapas: *Sincronização da Fala*, *Geração de Expressões Faciais* e *Geração de Comportamento dos Olhos*. Essas etapas provêm os valores de cada parâmetro de animação (FAP) correspondente à ação facial desejada, ao longo do tempo. Depois disso, é feita a *Sincronização*

dos *Parâmetros de Animação*, etapa que recebe os valores dos FAPs e resolve possíveis conflitos entre eles (como a combinação entre expressões faciais e visemas). Essa etapa retorna então as *streams* de parâmetros de animação final, que são processadas pelo modelo de *Geração/Visualização da Animação Facial*, podendo ser executadas com diferentes modelos de face 3D.

4.2.1 Geração de Comportamento dos Olhos

O modelo de *Geração de Comportamento dos Olhos*, cuja arquitetura está ilustrada na Figura 4.8, contém a implementação do modelo de Queiroz *et al.* (65) para geração automática de animações dos olhos. O modelo provê um conjunto de comportamentos dos olhos que podem ser usados com diferentes estados afetivos. O modelo utiliza um *modelo estatístico* (de Lee *et al.* (44)) como o motor de geração dos movimentos sacádicos e cria diferentes olhares (que compõem a *Base de Dados Comportamental*) através de modificação nos parâmetros do olhar considerados pelo modelo estatístico, como direção, magnitude e intervalo entre os movimentos. Movimentos sacádicos são movimentos rápidos de ambos os olhos de uma posição para outra (44).

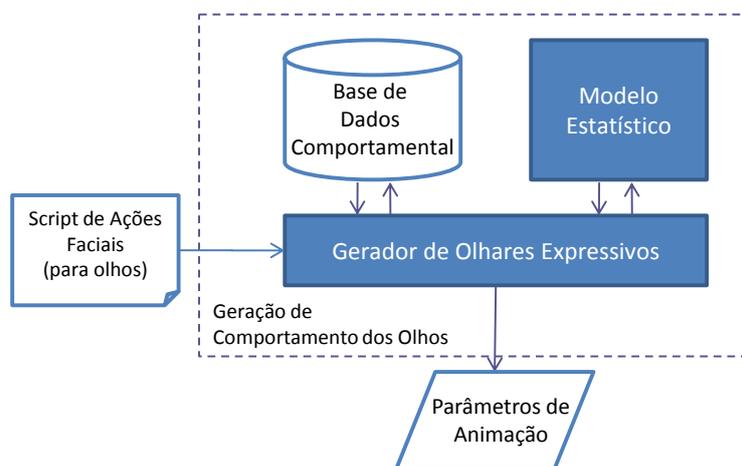


Figura 4.8: Etapas do modelo de *Geração de Comportamento dos Olhos*

Nos *scripts* FDL, os comportamentos dos olhos são descritos como ações em alto-nível. Quando combinados com expressões faciais, eles contribuem no aumento da expressividade e engajamento na comunicação. O *Gerador de Olhares Expressivos* recebe as ações faciais correspondentes aos olhos e retorna os parâmetros de animação para cada olhar gerado de acordo com o comportamento de olhos especificado, além de prover movimentos de cabeça e pálpebras de acordo com as regras do modelo (65).

4.2.2 Geração das Expressões Faciais

A tarefa do modelo de *Geração de Expressões Faciais* (Figure 4.9) é mais simples: ele associa as ações de expressões faciais com um conjunto de expressões cujos parâmetros de animação foram previamente definidos (na *Base de Dados Emocional*). O *Gerador de Expressões Faciais* recebe as ações faciais de expressões, busca os parâmetros na base de dados e gera as *streams* de parâmetros ao longo do tempo, de acordo com a duração definida no *script*, por interpolação linear.

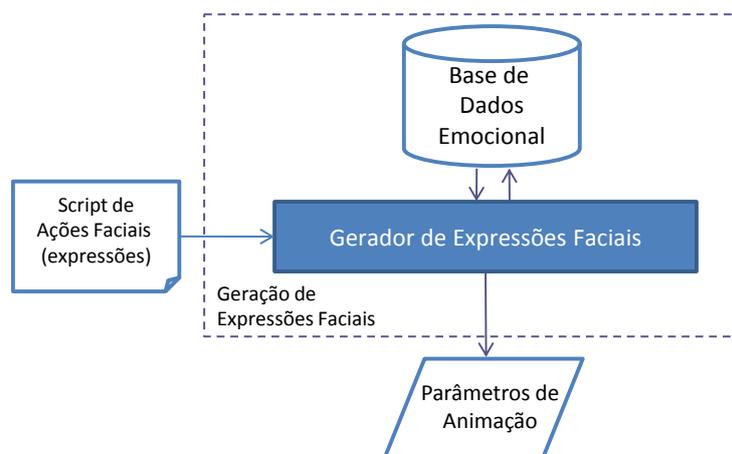


Figura 4.9: Arquitetura do modelo de *Geração de Expressões Faciais*

4.2.3 Sincronização da Fala

O modelo de *Sincronização da Fala* (Figura 4.10) segue a metodologia descrita em Rodrigues (68). Basicamente, o modelo recebe como entrada um arquivo de áudio, contendo a fala do personagem e um arquivo auxiliar contendo sua descrição textual. Para a *Extração dos Fonemas*, é usada a ferramenta CSLU Toolkit¹, que alinha o arquivo de áudio e a descrição textual e provê um arquivo com os fonemas ditos ao longo do tempo. Este arquivo é a entrada para a etapa de *Mapeamento Fonema-Visema*, que associa os fonemas às suas representações visuais (visemas). O CSLU Toolkit gera 41 fonemas da língua inglesa, que são mapeados para os 14 visemas especificados pelo padrão MPEG-4, de acordo com o mapeamento proposto em (68) (ver Anexo B). Com essa etapa, obtém-se o *timeline* completo dos visemas, cujo mapeamento para os parâmetros de animação é feito pela etapa de *Mapeamento Visema-Parâmetros de Animação*. Os parâmetros de animação dos visemas estão previamente definidos, assim como as expressões faciais. Nessa etapa é gerada a animação conforme as durações de cada visema e a transição entre eles, por interpolação linear.

¹<http://cslu.cse.ogi.edu/toolkit/>

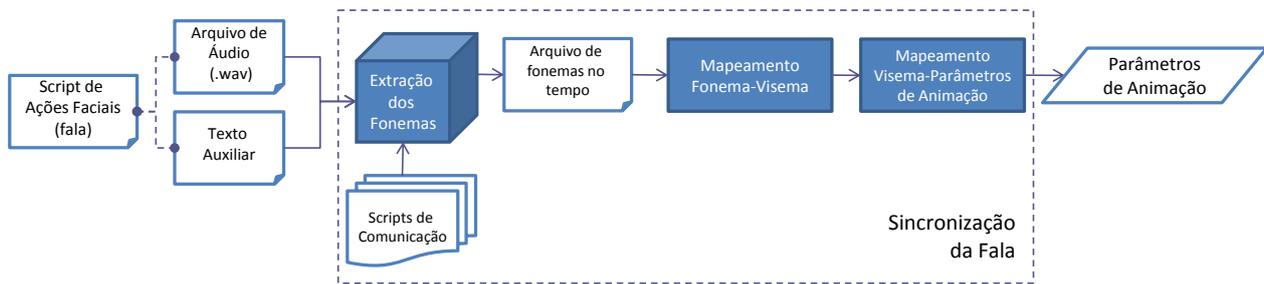


Figura 4.10: Arquitetura do modelo de *Sincronização da Fala*

4.2.4 Sincronização dos Parâmetros de Animação

Após as três etapas de processamento das ações faciais descritas no arquivo FDL, a etapa de *Sincronização dos Parâmetros de Animação* recebe os valores dos FAPs retornados e resolve possíveis conflitos entre eles. O modelo faz o tratamento de três tipos de conflitos:

1. Conflito Visema-expressão

Esse conflito ocorre principalmente na região da boca, quando as expressões faciais incluem movimentos na boca. Para resolver esse conflito, é feita a combinação dos valores dos FAPs gerados pelos modelos de expressão facial e sincronização da fala através de uma soma ponderada. Pyun *et al.* (63) propõe que os pesos para diferentes visemas e expressões variem de acordo com sua “importância”. Esses pesos poderiam ser incluídos como parâmetros adicionais no *script* de ações faciais ou em uma “tabela de importâncias”, definidas por um artista. No modelo, são usados os mesmos pesos para todas as combinações de expressões e visemas, como mostrado na Equação 4.11. Cada FAP (FAP_j) é gerado pela soma ponderada dos parâmetros de expressão (E_j) e visema (V_j).

$$FAP_j = \frac{E_j + 3V_j}{4} \quad (4.11)$$

2. Conflito Olhos-expressão/Cabeça-expressão

O modelo de *Geração de Comportamento dos Olhos* gera movimentos de pálpebras e cabeça relacionados com a direção e deslocamento do olhar (65). Mas algumas expressões faciais definem seus valores para as pálpebras e cabeça. Nesses casos, também é proposta uma soma ponderada para resolver o conflito (25% para o parâmetro de animação do comportamento dos olhos e 75% os da expressão).

3. Controle da Cabeça

Durante o processamento das ações faciais, alguns comportamentos dos olhos geram movimentos de cabeça, como descrito anteriormente. Se isso acontece, o movimento é mantido. Se não há movimentos da cabeça, então é utilizada uma pequena perturbação nos parâmetros de animação da cabeça usando o algoritmo de *Perlin noise* (59) para gerar suaves movimentos, de maneira a produzir animações mais realistas.

4.2.5 Facial Description Language (FDL)

A FDL é uma extensão de um trabalho anterior (65), em que as ações faciais (no caso, ações dos olhos) foram codificadas como um conjunto de comandos em *scripts* Lua, chamada GDL (Gaze Description Language). A FDL, portanto, encapsula a GDL e provê a inclusão de ações de expressões faciais e fala. Optou-se em usar Lua ao invés de uma sintaxe baseada em XML por duas razões: (1) a sintaxe da Lua é mais “limpa” (o que torna mais clara a leitura por pessoas) e sua estrutura ajuda a entender intuitivamente a seqüência de ações; e (2) é uma poderosa maneira de se incorporar funcionalidades no modelo/*framework*, uma vez que a linguagem comunica-se de maneira bi-direcional com a linguagem C/C++ (37) (pensando-se em futuras extensões).

A FDL, assim como a GDL, possui dois propósitos principais:

1. Incluir ações faciais ao modelo, através da especificação de seus parâmetros de animação (como nas bases de Dados Comportamental e Emocional dos modelos de geração de comportamento dos olhos e expressões faciais);
2. Descrever uma ou mais ações em seqüência, que constituem no *storyboard* para a geração das animações.

Um *script* de *storyboard* de ações contém informação sobre a fala (nomes dos arquivos de som e texto) e dos arquivos que contém a seqüência de expressões faciais e de olhar, como mostra o exemplo abaixo. Caso não exista um desses componentes, deve ser atribuído o valor “none” às variáveis. Se a saída desejada é um arquivo contendo os parâmetros de animação ao longo do tempo, deve-se especificar o nome do arquivo de saída.

```
FDL = {  
  speakSound = "hastalavista.wav",  
  speakText = "hastalavista.txt",  
  expressionScript = "expressions.sb",  
  eyesScript = "eyes.sb",  
  output = "animation.fap",  
}
```

Uma seqüência de expressões faciais pode ser descrita como no exemplo abaixo (arquivo “expressions.sb”), que informa as expressões da *Base de Dados Emocional* desejadas ao longo do tempo e sua duração.

```
expressions = {
{"joy", 100},
{"sadness", 100},
{"surprise", 100},
{"anger", 50},
{"disgust", 50},
{"fear", 100},
{"trust", 100},
{"tongueout", 100},
{"inlove", 100},
{"worry", 100},
{"sleepy", 100},
}
```

De forma similar, é descrita a seqüência de comportamentos dos olhos, como visto no exemplo abaixo (arquivo “eyes.sb”) que inclui os comportamentos da GDL (65) e o movimento de “perseguição” dos olhos (*pursuit*) adicionado em um dos estudos de caso desenvolvido (Seção 5.2).

```
eyes = {
{"lookTo", "left", 17.0, "yes", 50},
{"default", "talking", 0.9, 500},
{"concentration", 0.01, 50},
{"discomfort", 0.99, 0.6, 500},
{"distress", 0.5, 400},
{"ironic", 0.5, 50},
{"pursuit", "right", 17.0, "yes", 50},
}
```

Como mostrado no primeiro exemplo de *script* FDL, para a geração da fala são especificados um arquivo de som e sua descrição textual (transcrição da fala em texto). Após o processo de mapeamento fonema-visema, o modelo pode gerar um arquivo FDL como o exemplo abaixo, que é processado pela etapa de *Mapeamento Visema-Parâmetros de Animação*. Alternativamente, esse tipo de *script* pode ser armazenado e usado em aplicações com falas-padrão (otimizando a aplicação, não necessitando o processamento dos arquivos de fala repetidamente), ou mesmo refinado manualmente por algum artista. Seguindo a mesma sintaxe dos outros arquivos, é descrita a seqüência de visemas ao longo do tempo.

```

Sentence = {
{"silence", 11},
{"A", 6},
{"kg", 1},
{"Q", 1},
{"nl", 3},
{"sz", 3},
{"I", 6},
{"A", 2},
{"silence", 27},
}

```

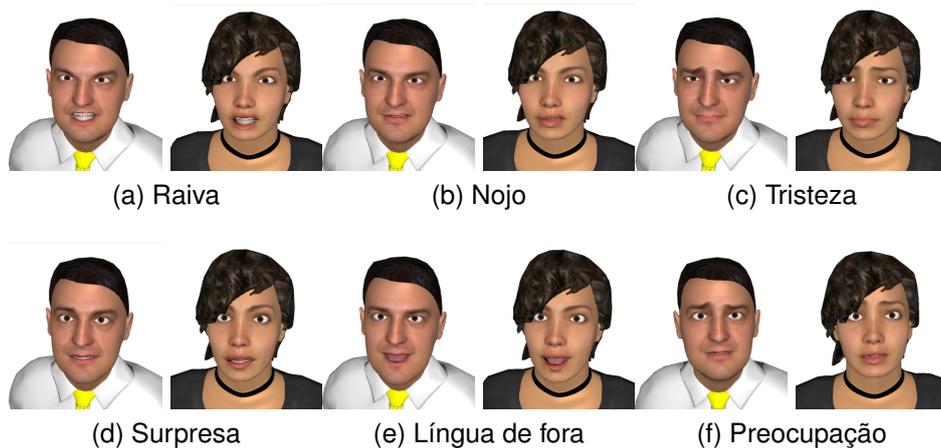


Figura 4.11: Exemplo de animações geradas através de algumas ações faciais da FDL, executadas por dois personagens diferentes.

A Figura 4.11 apresenta exemplos de animações geradas a partir de *scripts* FDL, que podem ser aplicadas a diferentes personagens. Outros resultados podem ser vistos na Seção 5.2. Vê-se que é possível controlar os avatares através de um variado conjunto de comandos de ações faciais, de maneira explícita. Com a forma de controle apresentada, pode-se criar animações personalizadas no sentido de criar qualquer fala (em língua inglesa), combinada ou não com expressões faciais e comportamentos dos olhos. No entanto, nessa abordagem não é possível gerar automaticamente a personalização das ações faciais em si. Por exemplo todos os modelos irão sorrir da mesma maneira, a menos que, com um trabalho manual, especifique-se diferentes tipos de sorriso através de FAPs em um editor gráfico como o do trabalho de Rodrigues (68) ou a ferramenta de edição do Xface (7).

4.3 Controle por Animação Dirigida por Performance

A arquitetura do modelo de *Controle por Animação Dirigida por Performance* está ilustrada na Figura 4.12, correspondente às etapas *Detecção/Rastreamento dos Dados Faciais* e *Mapeamento Real-Virtual* da Figura 4.1 apresentada no início do capítulo. O modelo leva em consideração dois tipos de informação, que são providas pela etapa de *Rastreamento dos Pontos*:

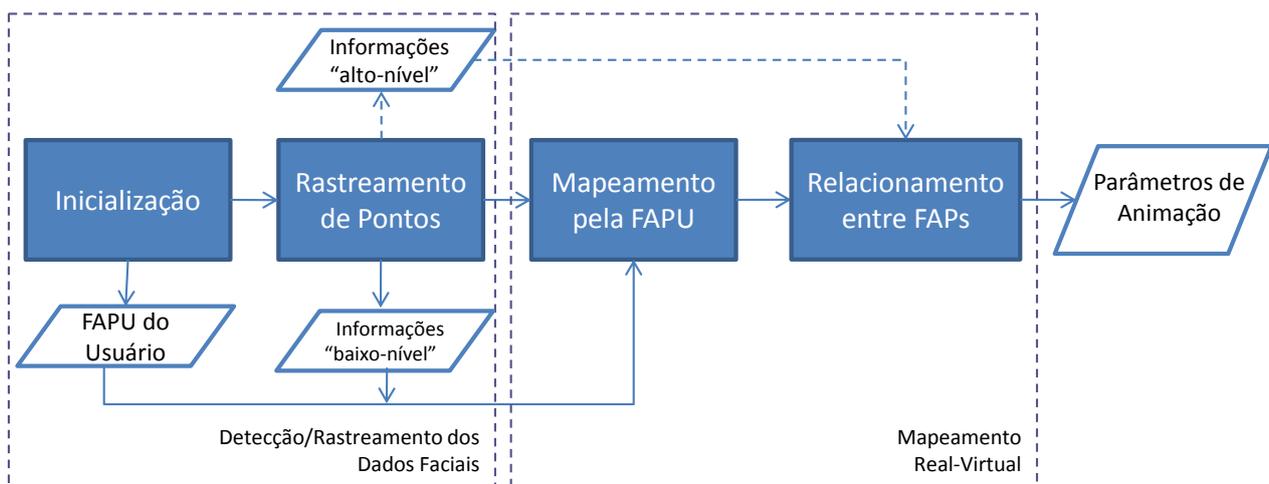


Figura 4.12: Arquitetura do modelo de Controle por Animação Dirigida por Performance

- **Informações “baixo-nível”**

São as informações numéricas, como por exemplo a nova posição dos pontos característicos rastreados em um *frame*. Esses dados são mapeados para os pontos característicos do avatar na etapa de *Mapeamento pela FAPU*, como mostra a Seção 4.3.2;

- **Informações “alto-nível”**

São as informações semânticas sobre expressões faciais (estado afetivo) ou ações faciais mais específicas sobre alguns componentes, como por exemplo, estados da boca: “fechada”, “aberta”, “sorrindo”, “fazendo bico”, etc. Essas informações servem de informações complementares para o *Relacionamento entre FAPs* (Seção 4.3.3).

4.3.1 Rastreamento dos Pontos

O modelo parte do pressuposto que os algoritmos de Visão Computacional vão detectar e/ou rastrear um conjunto de pontos característicos da face do usuário. Esses pontos característicos serão diretamente associados aos FPs do padrão MPEG-4. Assim, o mapeamento se dá através da determinação dos deslocamentos dos FPs em unidades de FAPU

(da pessoa real e do modelo 3D), associando-se aos seus devidos parâmetros de animação (Seção 4.3.2).

A etapa de *Rastreamento de Pontos* faz o rastreamento de marcadores localizados na face do usuário, através da técnica de fluxo ótico. Essa técnica calcula uma aproximação do campo de vetores velocidade a partir do movimento da intensidade dos pixels da imagem em uma sequência de vídeo (53). No modelo, utiliza-se a proposta de Lucas-Kanade-Tomasi (LKT) (46). Para dar início ao rastreamento, é necessária uma etapa de *Inicialização* do algoritmo, clicando-se no centro dos marcadores.

Os pontos rastreados correspondem a um sub-conjunto de FPs (pontos vermelhos na Figura 4.13). Para esses pontos, o mapeamento dos deslocamentos ao longo do tempo é feito de maneira direta, utilizando-se as unidades de FAPU (*Mapeamento pela FAPU*). Os pontos azuis da Figura 4.13 correspondem aos pontos em que o mapeamento é feito através de uma relação estabelecida entre eles e os FAPs dos pontos rastreados (*Relacionamento entre FAPs*).

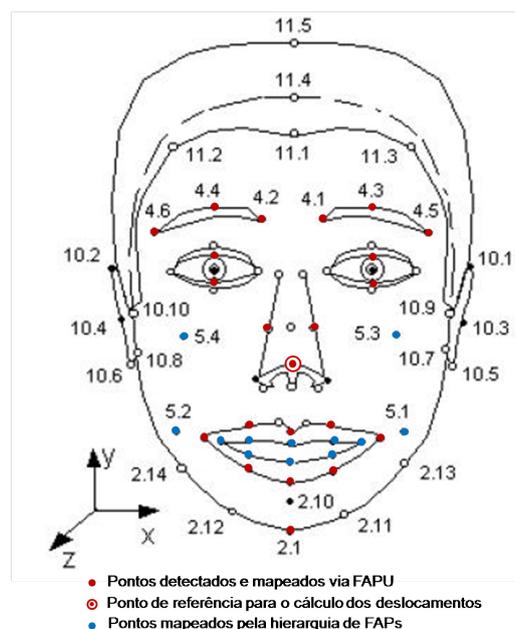


Figura 4.13: Conjunto de pontos característicos utilizado, conformes ao padrão MPEG-4. Os pontos vermelhos são os que são mapeados diretamente pelo *Mapeamento pela FAPU*, e os pontos azuis são determinados pelo *Relacionamento entre FAPs* estabelecido no modelo.

4.3.2 Mapeamento pela FAPU

Seja $T_{ini} = \{\vec{t}_1, \vec{t}_2, \dots, \vec{t}_n\}$ o conjunto de pontos característicos determinados manualmente na *Inicialização*, conforme os pontos vermelhos da Figura 4.13. Com alguns desses pontos, calcula-se a FAPU da pessoa no primeiro frame da captura ($FAPU_{usuário}$), como

no exemplo (Figura 4.14). Esses pontos devem corresponder aos pontos apresentados na Seção 4.1.3 (para o cálculo do FAPU do avatar). Como alguns pontos não possuem marcadores correspondentes, é necessário fazer a aproximação dos seguintes FPs, conforme as equações 4.12 a 4.15:

$$FP_{3.5} = (FP_{3.1x}, (FP_{3.1y} - FP_{3.3y})/2) \quad (4.12)$$

$$FP_{3.6} = (FP_{3.2x}, (FP_{3.2y} - FP_{3.4y})/2) \quad (4.13)$$

$$FP_{2.2} = (FP_{8.1x}, (FP_{8.1y} - FP_{8.2y})/2) \quad (4.14)$$

$$FP_{9.15} = (FP_{9.3x}, (FP_{9.3y} - FP_{2.2y})/2) \quad (4.15)$$

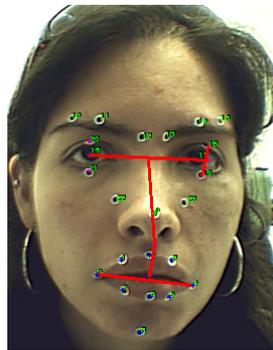


Figura 4.14: Exemplo de FAPU do usuário, determinada na inicialização da captura.

Um dos pontos característicos, o correspondente à ponta do nariz \vec{r} em cada passo tempo i (\vec{r}_i), é considerado o ponto de referência para o cálculo dos deslocamentos de todos os outros pontos, por não possuir significativos deslocamentos em relação ao restante do conjunto de pontos rastreados. A partir de \vec{r}_{ini} (ponto \vec{r} no primeiro *frame*) e os pontos \vec{t}_j de T_{ini} , obtém-se então um conjunto de distâncias de referência $D_{ini} = \{d_1, d_2, \dots, d_n\}$, $d_j = |\vec{t}_j - \vec{r}_{ini}|$.

Para cada passo de tempo, obtém-se o conjunto dos pontos rastreados T_i pelo algoritmo LKT, cujas distâncias em relação a \vec{r}_i são calculadas, obtendo-se o conjunto dos deslocamentos D_i .

O modelo assume que a cabeça do usuário não vai se mover. Até o momento, nos experimentos, procurou-se trabalhar com a face em uma distância constante da câmera, de forma a não alterar a escala da face projetada. Caso a pessoa se aproxime ou distancie da câmera, seria necessário reescalar a FAPU e o conjunto de referenciais D_{ini} . Uma alternativa simples para isso seria, utilizando um detector de face robusto, reescalar a FAPU e D_{ini} de acordo com o raio da face detectada. O modelo também não considera rotações na cabeça. Uma vez que podem ocorrer pequenas variações na movimentação da cabeça durante a captura, em cada passo de tempo é aplicada uma correção, baseada na diferença

entre os pontos r_{ini} e r_i , na determinação de D_i .

Para a determinação dos valores do conjunto de parâmetros de animação (FAPs) em um *frame*, é feita então a diferença entre D_i e D_{ini} , normalizada pela FAPU do usuário.

$$FAP_{s_i} = \frac{D_i - D_{ini}}{FAPU_{usuario}} \quad (4.16)$$

O sinal dos FAPs devem ser dados de acordo com as direções especificadas no conjunto de parâmetros. Então, para a geração das animações no avatar, esses valores são escalados pela FAPU do avatar em seus FPs e são feitas as deformações do restante dos vértices da face, conforme a equação 4.6 e a função de deformação correspondente.

4.3.3 Relacionamento entre FAPs

Optou-se em utilizar um conjunto não muito numeroso de pontos para o rastreamento, visando uma futura substituição do algoritmo com marcadores por algum algoritmo de detecção/rastreamento mais robusto, que não necessite marcadores e inicialização (detecção) manual. Por isso, um critério para a escolha dos pontos foi aqueles em que já existem alguns trabalhos de detecção automática na bibliografia. Além disso, em algumas áreas (como a parte interna dos lábios) é inviável/incômoda a colocação de marcadores. Outro critério para a escolha dos pontos rastreado foi a “independência muscular” deles, baseada em observações. Por esse motivo, julgou-se necessário também o rastreamento dos dois pontos laterais da pirâmide nasal.

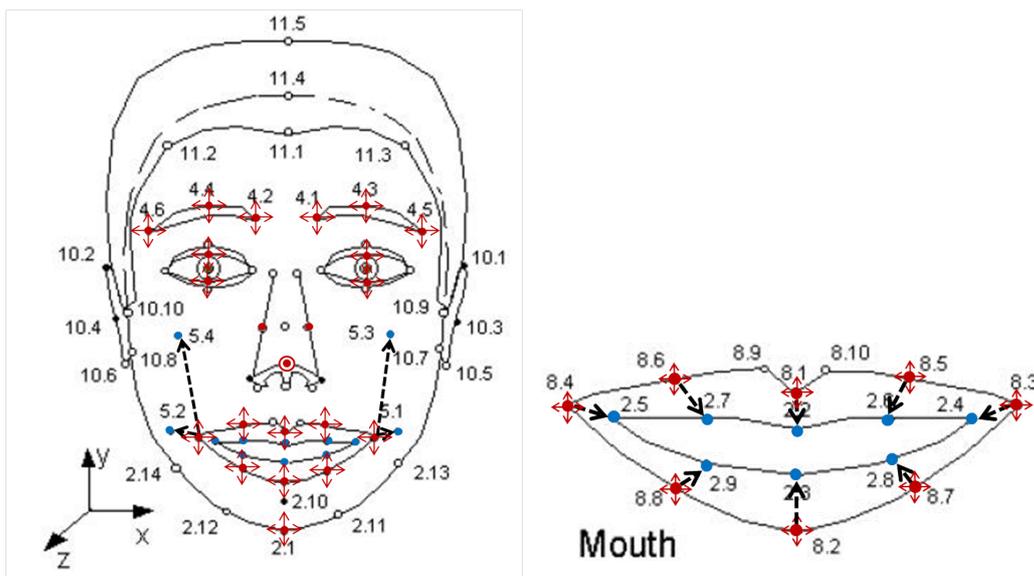


Figura 4.15: Relacionamento hierárquico dos FAPs.

A etapa de *Relacionamento entre FAPs* estima os valores de FAPs que não são obtidos via o mapeamento pela FAPU apresentado anteriormente. A idéia básica desse relacio-

namento é estabelecer uma hierarquia de FPs e seus FAPs, como se fossem “pseudo-músculos”, baseados em observações (Figura 4.15). Sejam dois FPs relacionados, um rastreado por CV (FP_R) e outro que não é rastreado (FP_N). É feita uma associação entre os FAPs desses FPs, normalmente entre os que possuem o movimento na mesma dimensão (as regras de associação são definidas por observação). O valor de um FAP_N é determinado em função do seu relacionamento com o FAP_R , de acordo com a Equação 4.17:

$$FAP_N = c_{RN} \cdot FAP_R \quad (4.17)$$

Onde c_{RN} é o intensificador que relaciona o FAP_R e o FAP_N , definido por observações. Dessa maneira, o *Relacionamento entre FAPs* possui duas tarefas principais:

- Estimar valores de deslocamento de FPs que não são rastreados, como os FPs das bochechas e da parte interna dos lábios. Nessas situações, propôs-se um relacionamento hierárquico que pode ser visto pelas setas pontilhadas pretas na Figura 4.15.

Para os FPs das bochechas, optou-se em associá-los com os FPs dos cantos da boca. E para cada FP da parte interna dos lábios, associou-se o FP correspondente da parte externa. O relacionamento entre os FAPs desses FPs é dado pela dimensão da direção do movimento. O valor dos c_{RN} para os FAPs das bochechas com movimento na dimensão x é 0.5 e na dimensão y é 0.25. Para os FAPs relacionados dos lábios, definiu-se $c_{RN} = 1$.

- Estimar valores de alguns FAPs com movimentos no eixo z , não considerado pelo mapeamento, que só lida com a informação bi-dimensional das imagens e gera valores para os FAPs com direções do movimento em x e y . Nesse caso, não é necessário relacionar dois FPs diferentes, mas o valor de um FAP com direção no eixo x ou y com do do eixo z .

No momento, o modelo faz o mapeamento do FAP *THRUST_JAW* (deslocamento em profundidade da mandíbula) com o FAP *OPEN_JAW* (abertura da mandíbula), com $c_{RN} = 0.5$.

Relacionamentos gerados por informações “alto-nível”

Outro tipo de relacionamento considerado nessa etapa é através da definição de algumas regras, a partir de informações “alto-nível” obtidas durante o rastreamento. No momento, como o modelo não provê esse tipo de informação, foram feitos alguns estudos preliminares considerando alguns estados de boca que devem ser identificados manualmente no protótipo do modelo (como “fazendo bico” e “sorriso”), uma vez que existe a variação na espessura dos lábios nessa situação, e as informações vindas de CV apenas os deslocamentos dos pontos externos da boca. Nesse caso, o valor do intensificador c_{RN} é baseado

no estado de boca setado. A Figura 4.16 mostra o impacto visual desse tipo de informação: na imagem do centro, é mostrado o resultado da animação gerada sem a informação do estado da boca, e à direita, foi indicado o estado “fazendo bico”, e então foi definido um alargamento dos pontos internos da boca, utilizando os valores -1.5 e 0.5 como c_{RN} dos três FAPs internos dos lábios superior e inferior, respectivamente. Além disso, são associados os FAPs com movimento em z $PUSH_T_LIP$ e $PUSH_B_LIP$, utilizando-se os valores 2.0 e 0.5 , respectivamente, para o c_{RN} .

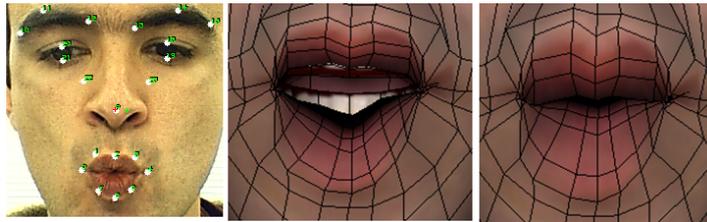


Figura 4.16: Exemplo do uso de informações “alto-nível” para melhorar a animação: à esquerda, a pose de usuário, que foi rotulada (manualmente) para o estado de boca “fazendo bico”. No centro, a animação gerada sem levar em consideração o estado de boca (relacionamento entre FAPs padrão). À direita, a animação gerada levando em consideração o estado de boca.

A Figura 4.17 mostra um exemplo de animação gerada seguindo a metodologia apresentada. Outros resultados podem ser vistos na Seção 5.3, assim como uma avaliação qualitativa do modelo.

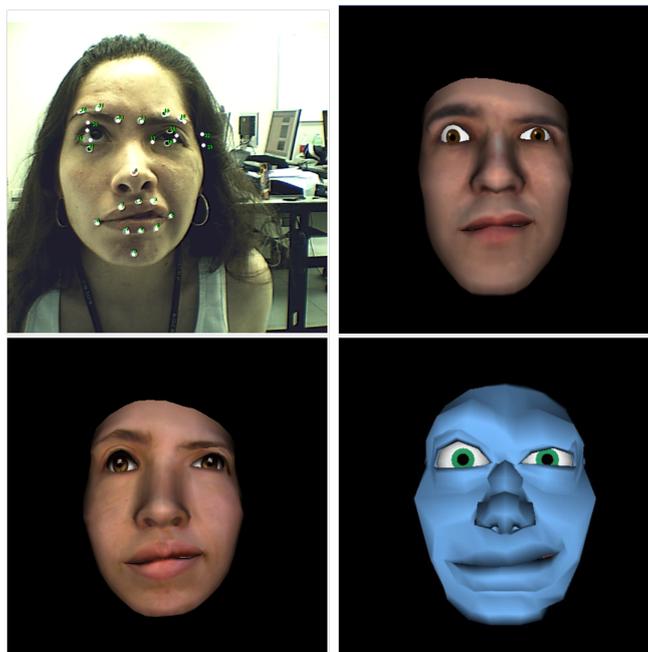


Figura 4.17: Exemplo de animação gerada pelo modelo via PDA.

4.4 Algumas Considerações

A criação do modelo contou com três etapas principais de pesquisa e desenvolvimento, correspondentes aos modos de controle e síntese da animação facial:

1. Desenvolvimento de uma plataforma de animação facial adequada e extensível para o desenvolvimento de pesquisas

Inicialmente, foram escolhidos o modelo de parametrização das ações faciais e as técnicas “baixo-nível” para implementá-las (geração das deformações). Optou-se em seguir a parametrização baseada no padrão MPEG-4 de Animação Facial (houve, no entanto, algumas extensões, apresentadas na Seção 4.1.2).

Para a geração das deformações, optou-se em usar, inicialmente, funções de cosseno, como em (7). Para zonas com pontos de controle muito próximos e correlacionados (como os dos lábios), optou-se em utilizar uma única zona de influência e fazer a interpolação dos pontos através de uma função de base radial (89).

2. Desenvolvimento de uma estratégia de controle de ações em alto nível, via linguagem de *script*

A fim de oferecer uma opção de parâmetros mais “alto-nível” que os FAPs do padrão MPEG-4, foi desenvolvido um modelo de controle que inclui uma linguagem de *scripting* de ações faciais, também facilmente extensível. Foram incorporados alguns modelos da literatura para a geração de fala sincronizada (68) e comportamentos dos olhos (65), combinados com um pequeno conjunto de expressões faciais definido em termos de FAPs, e acessíveis via linguagem de *scripting*.

3. Desenvolvimento de um modelo de Animação Dirigida por Performance

Na Seção 4.3 é proposto um modelo que, a partir da captura de um conjunto de pontos-chave da face do usuário, gera animações verdadeiramente diferenciadas no avatar, procurando refletir os sinais expressivos da face do usuário. O modelo gera os movimentos com base no deslocamento dos pontos capturados, medidos em termos de FAPU e traduzidos para o conjunto de parâmetros de animação (FAPs estendidos). Dessa maneira, o mapeamento para quaisquer modelos 3D devidamente parametrizados conforme o esquema apresentado na Seção 4.1 se torna imediato.

O modelo proposto pode ser visto como um *framework* para o desenvolvimento de aplicações e pesquisas em Animação Facial, que permite a inclusão de técnicas para detecção de eventos e a geração de animações faciais interativamente. Isso pode se dar através da inclusão de diferentes técnicas de RV e CV, que diretamente gerem *streams* de FAPs ou arquivos FDL (por exemplo, pela inclusão de um modelo emocional que responde a eventos

do usuário ou mesmo modelos de processamento de linguagem natural que gerem fala por texto e/ou áudio). Ao mesmo tempo, vê-se que é possível a extensão dos componentes ilustrados nos diagramas de arquitetura dos modelos, de maneira a incluir métodos para melhoria das animações, como outras técnicas de deformação ou manipulação das texturas, para a geração de movimentos específicos e rugas de expressão. Focando no modelo que faz a geração de animações dirigidas por performance, vê-se que ele pode ser estendido através da inclusão de algoritmos de Visão Computacional para detecção e/ou *tracking* mais robusto (por exemplo, sem marcadores ou incluir novas *features*, como a movimentação da cabeça) e melhorias no mapeamento usuário-avatar.

5. Resultados

O modelo de animação facial descrito no capítulo anterior permite a geração de animação de faces em tempo-real por duas técnicas de controle: por *script* ou dirigida por performance. As próximas seções apresentam algumas informações técnicas do protótipo desenvolvido, resultados visuais das animações geradas, algumas aplicações desenvolvidas como “estudos-de-caso” para a exploração do modelo de controle por *script* e uma avaliação feita com usuários para avaliar qualitativamente o modelo de controle por PDA.

5.1 O Protótipo

Para validação do modelo, foi construído um protótipo, que pode ser considerado um *framework* para o desenvolvimento de pesquisas e aplicações. O protótipo foi desenvolvido em C/C++, usando a API gráfica OpenGL (86). Durante a primeira etapa da pesquisa, utilizou-se os recursos do Xface para a geração das animações faciais, visualização e interface. Depois, como explicado anteriormente na Seção 3.3.4, optou-se em implementar uma biblioteca própria para geração das animações. As animações geradas pelo modelo podem ser salvas em arquivos contendo as *streams* dos FAPs a cada frame, como mostra a Figura 5.1. Esse arquivo segue o formato encontrado nos trabalhos de (19) (LUCIA) e (7) (Xface). Portanto, os parâmetros de animação gerados podem ser processados e as animações, visualizadas nos sistemas citados. Porém, a geração das animações será conforme o modelo de deformação de cada aplicação, o que pode gerar resultados visuais diferentes.

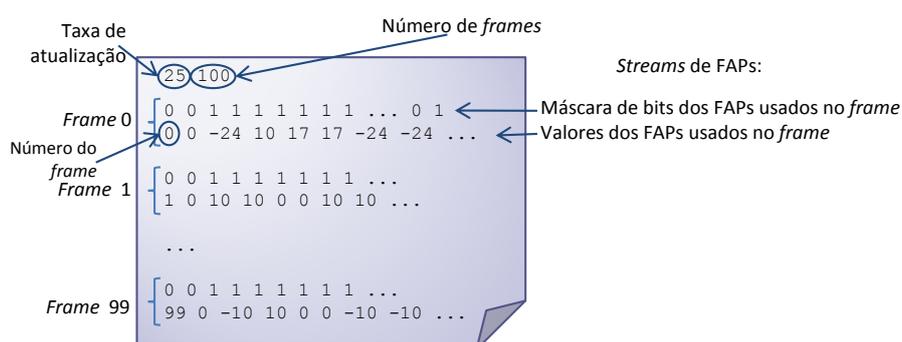


Figura 5.1: Esquema de um arquivo FAP.

Nesse contexto, uma consideração importante a se fazer é que os modelos de controle dos parâmetros de animação fornecem como saída *streams* de FAPs (MPEG-4 + FAPs acrescidos) no modelo, conforme a Seção 4.1.2. Atualmente, a *stream* de parâmetros gerada pelo modelo de PDA possui 84 parâmetros de animação (68 FAPs e 16 novos).

O processamento da fala se dá através da comunicação via *scripts* com os aplicativos do CSLU *Toolkit*. Para a FDL, foi utilizada a linguagem LUA e uma biblioteca que faz a comunicação entre C++ e Lua (37), chamada Diluculum¹. Foram utilizados os algoritmos de visão computacional da biblioteca OpenCV².

No estágio atual, o protótipo permite 3 tipos de entrada: *scripts FDL*, arquivos com animações FAPs já geradas e vídeo. Os algoritmos usados para o rastreamento dos pontos característicos exigem o uso de marcadores e inicialização manual (clique nos pontos), para uma captura estável. Foram feitas também algumas integrações com ferramentas do grupo de pesquisa, usando algoritmos de detecção sem marcadores (para um conjunto menor de pontos característicos). Essa integração permite a geração usando diretamente a *web cam*. Porém, como esses algoritmos ainda estão em fase de estudo, o resultado da detecção é ainda um pouco instável. A Figura 5.2 mostra exemplos da integração do modelo em um protótipo de ambiente colaborativo desenvolvido pelo grupo de pesquisa.

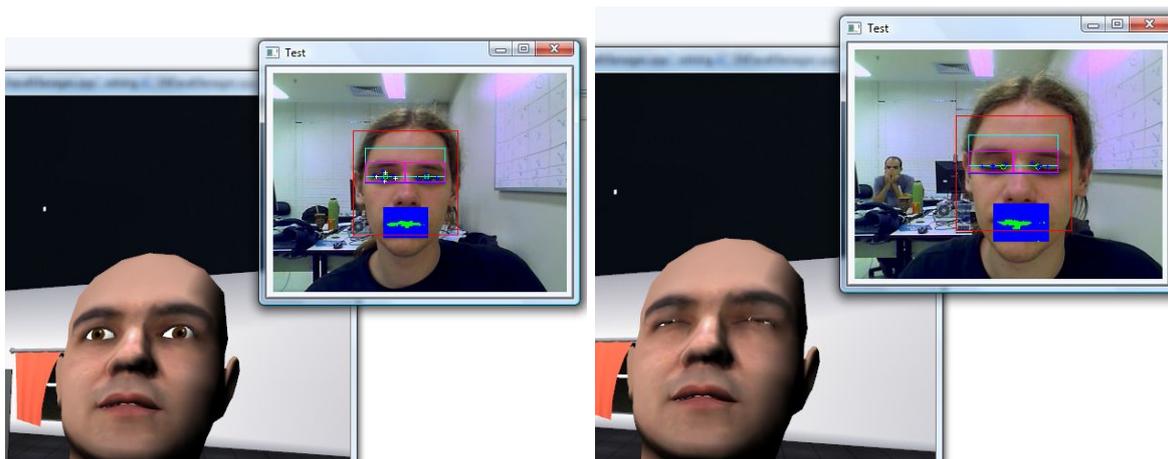


Figura 5.2: Exemplos de animações geradas em tempo-real, capturadas diretamente da *webcam*, utilizando-se algoritmos de detecção que não exigem marcadores em um protótipo de ambiente colaborativo.

5.2 Controle por Script: Estudos de Caso

A fim de se investigar a capacidade de geração de animações variadas pelo modelo de controle por *scripts*, foram desenvolvidos alguns “estudos de caso”, ou seja, aplicações e produções pequenas com alguns objetivos diferentes:

- A exploração da robustez da FDL em termos de: “o quão usável ela é para produzir diálogos expressivos” e “se as animações produzidas são compreensíveis ao público

¹<http://www.stackedboxes.org/Projects/Diluculum/>

²<http://www.opencv.org>

geral”, via a elaboração das cenas de animação de um filme técnico para uma conferência³, sem qualquer retoque por artistas. Todos os diálogos da personagem virtual foram gerados automaticamente por *scripts FDL*. A edição dos *scripts*, no entanto, foi feita manualmente. A Figura 5.3 mostra alguns *snapshots* do filme;



Figura 5.3: Snapshots of the “Tracking is Believing” video.

- Alguns testes preliminares de geração de animações interativamente, usando algoritmos simples de CV (detector de face, detector de sorriso, por exemplo) para detectar eventos e mapear para ações de alto-nível, disparando animações pré-gravadas ou gerando *scripts* de ações em tempo real (Figura 5.4 a), c) e d) ilustra as aplicações interativas desenvolvidas), a saber:
 - Uma aplicação que explora a geração de animações de olhos interativamente, chamada *Follower Eyes*. Nessa aplicação, os olhos do personagem virtual seguem a face detectada do usuário, utilizando o algoritmo de detecção de faces do OpenCV (80). Para essa aplicação, foi criado o movimento de “perseguição” dos olhos (*pursuit*) e adicionado à FDL. O movimento de perseguição ocorre quando os olhos seguem um objeto, voluntariamente ou não. Esses movimentos são um pouco diferentes dos movimentos sacádicos (providos pelo modelo estatístico): eles são mais suaves, contínuos e com uma latência menor (44). Esse movimento dos olhos, diferentemente dos comportamentos gerados pelo modelo de Geração de Comportamento dos Olhos, que usa o modelo estatístico de (44) para a obtenção dos seus parâmetros, tem seus parâmetros (direção, ângulo de rotação e duração) determinados pela posição estimada da face detectada.
 - Uma aplicação que explora a interatividade através do disparo de animações geradas previamente (por *scripts FDL* editados manualmente) com falas expressivas, em que o personagem virtual (Diretor da Faculdade de Informática da PU-

³Vídeo vencedor da categoria técnica do Festival de Vídeos do SIBGRAPI 2008. Encontrado em <http://fellowsheep.wait4.org/mestrado/videos/TrackingIsBelieving.html>



Figura 5.4: *Snapshots* das aplicações desenvolvidas com o *framework*: a) *Follower Eyes*, em que o personagem virtual segue a face detectada do usuário com os olhos; b) Gerador de *ground truth*, para avaliação de algoritmos de CV; c) Diretor Virtual, em que o personagem virtual saúda e informa os usuários de acordo com a detecção e permanência da face; e d) Espelho Virtual, em que são detectadas algumas ações como sorriso e direção horizontal dos olhos e a personagem virtual “imita” a ação (um estudo preliminar de PDA).

CRS) saúda e apresenta informações sobre a faculdade de acordo com a detecção de dois tipos de evento: detecção (para as saudações) e permanência da face do usuário por algum tempo (informações). Para a detecção da face, também foi utilizado o detector do OpenCV.

- Uma aplicação que explorou, de maneira preliminar, a geração de animações dirigidas por performance através da detecção de eventos correspondentes a ações faciais de alto-nível (dados fornecidos por algoritmos experimentais do grupo de pesquisas), chamada de “Espelho Virtual”. Nessa aplicação, as ações faciais detectadas são: direção horizontal do olhar (direita, centro e esquerda), status da boca (sorrindo, aberta ou fechada) e rotação da cabeça em y e z . Na Figura 5.5, as as ações em que o algoritmo fornece uma intensidade (distância numérica em pixels) são mostradas nas setas azuis. Esses valores são mapeados para FAPs (através de regras de associação definidas empiricamente). As ações faciais em

que o algoritmo retorna somente um estado (“sorrindo”, por exemplo) são representadas pelas setas verdes na Figura 5.5. Nessas ações, o mapeamento para FAPs foi estabelecido com base em valores predefinidos (ações que integram as bases de dados dos modelos de *Geração de Comportamentos dos Olhos* e *Geração de Expressões Faciais*), sendo as animações geradas em tempo-real.

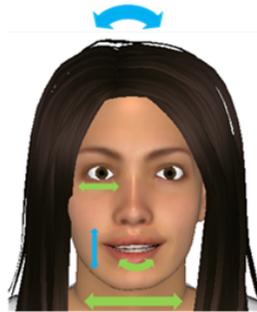


Figura 5.5: Ações faciais detectadas na aplicação “Espelho Virtual”, que constituem os parâmetros alto-nível de animação.

- Desenvolvimento de uma interface para a geração de ground-truth sintético, a partir do modelo de geração das deformações e scripts FDL (Figura 5.4 b)).

Para maiores detalhes técnicos desses estudos de caso, recomenda-se a leitura de (66)⁴. Em (66), o modelo de controle por *script* proposto é apresentado como um *framework* extensível para o desenvolvimento de aplicações com Animação Facial. O desenvolvimento desses estudos de caso mostraram que o modelo é capaz de produzir uma variedade de animações e ser integrado em aplicações interativas usando esse abordagem de controle de ações faciais alto-nível apresentada no capítulo anterior.

5.3 Controle por PDA: Avaliação com Sujeitos

O protótipo desenvolvido permite o controle por Animação Dirigida por Performance, que recebe como entrada um vídeo de pessoa com marcadores na face e necessita de uma etapa de inicialização manual, onde a pessoa clica no centro dos marcadores. Depois dessa etapa, o algoritmo de rastreamento obtém os deslocamentos dos pontos, que são mapeados para o avatar conforme o explicado no capítulo anterior. O protótipo ainda permite, se o usuário desejar, a geração dessas animações interativamente, de maneira a corrigir possíveis erros de rastreamento. Dessa maneira, é possível gerar animações mais acuradas, para situações em que não é exigido tempo-real, como o caso da avaliação apresentada a seguir. A Figura 5.6 mostra um exemplo de quadros de animação geradas pelo modelo, executadas por três personagens diferentes⁵.

⁴Vídeo técnico de (66) em <http://fellowsheep.wait4.org/mestrado/videos/VideoSBGames2008.html>

⁵Alguns vídeos de resultados podem ser vistos em <http://fellowsheep.wait4.org/mestrado/videos>

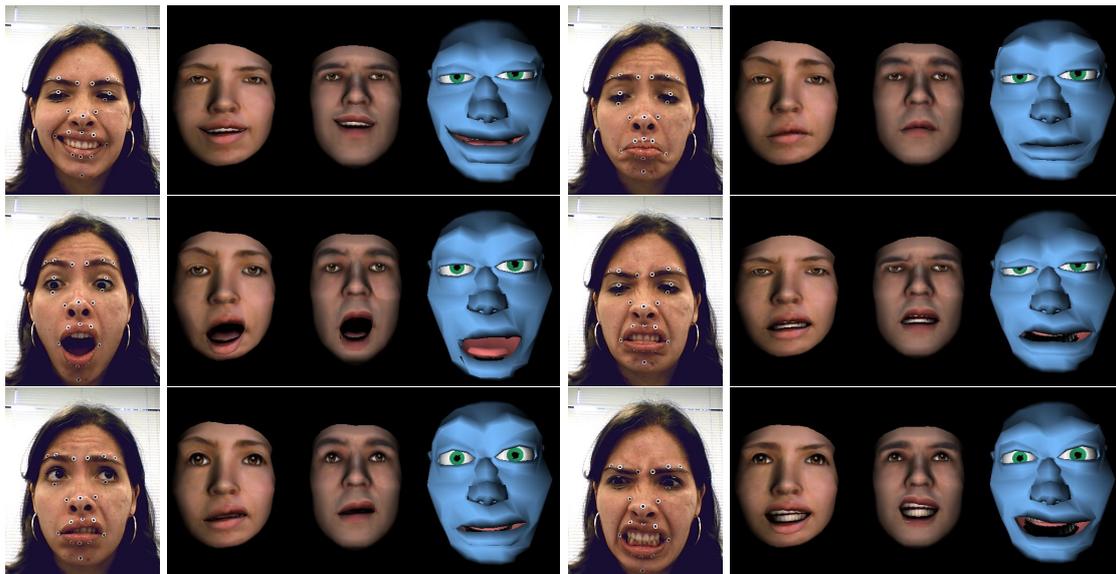


Figura 5.6: Resultado gerado pelo modelo para uma seqüência de expressões faciais executadas por um usuário, em diferentes modelos de face.

Esta seção apresenta uma avaliação feita com usuários, com o objetivo de fazer uma análise preliminar para verificar se o modelo desenvolvido consegue refletir a expressão da pessoa real, de maneira perceptível ao usuário. A avaliação foi elaborada de maneira a investigar três situações principais:

1. Identificação da expressão da pessoa real no avatar (imagens);
2. Impacto do uso do avatar correspondente à pessoa real (na forma) ou de um avatar diferente na identificação da imagem ou animação da com performance;
3. Identificação da expressão da pessoa real no avatar em vídeos (animação).

5.3.1 Metodologia

Para fazer a avaliação alcançando um número satisfatório de voluntários e um público variado, optou-se em fazer um questionário na internet. Para a elaboração das animações, contou-se com a participação de 5 voluntários, que gravaram vídeos seguindo uma seqüência de ações pré-definida. Esses voluntários serão chamados de atores no decorrer desse capítulo.

Atores e Avatares Utilizados

A Figura 5.7 mostra os atores e seus respectivos avatares, obtidos através do *software* Facegen⁶, que gera modelos de face 3D baseados em fotografias. Para os fins da avalia-

⁶<http://www.facegen.com/modeller.htm>

ção, também foram utilizados três avatares diferentes dos atores, sendo dois deles também gerados pelo Facegen e um deles um personagem cartum disponível na internet⁷.



Figura 5.7: À esquerda, os 5 atores voluntários para a gravação das performances e seus respectivos avatares, abaixo de cada um deles. À direita, os três avatares não correspondentes aos atores, sendo um deles (o azul, abaixo) um personagem cartum disponível na internet.

Gravação da Performance e Geração das Animações

Para a obtenção dos dados para a geração das animações, foi gravada a performance dos 5 atores, seguindo uma seqüência de expressões faciais pré-definida. As expressões faciais solicitadas foram as 6 expressões básicas de Ekman (29), a saber: *alegria*, *tristeza*, *surpresa*, *nojo*, *medo* e *raiva*. A Figura 5.8 mostra quadros da atuação de dois atores, correspondentes a cada uma das expressões e o mapeamento para o seu avatar correspondente.

Foram selecionados, manualmente dos vídeos, os *frames* considerados mais significativos (visualmente expressivos) dessas expressões faciais, e gerou-se a expressão de cada ator em todos os avatares⁸. Foram geradas as animações completas da expressão de *alegria* de cada ator, também com todos os avatares⁹. O processo de geração dos parâmetros de animação para cada vídeo dos atores foi feito apenas uma vez (gerando um arquivo FAP), junto com a animação de um avatar, e a geração das animações dos outros avatares foi feita através do arquivo FAP gerado.

Com esses dados, foram selecionadas algumas imagens e videos para o questionário.

⁷<http://jasonpierce.animadillo.com/resource/ludwig/ludwig.html>

⁸O resultado dessa etapa pode ser conferido em <http://fellowsheep.wait4.org/mestrado/quiz/TabelaExpressoes.html>

⁹Disponível em <http://fellowsheep.wait4.org/mestrado/quiz/TabelaVideosFeliz.html>

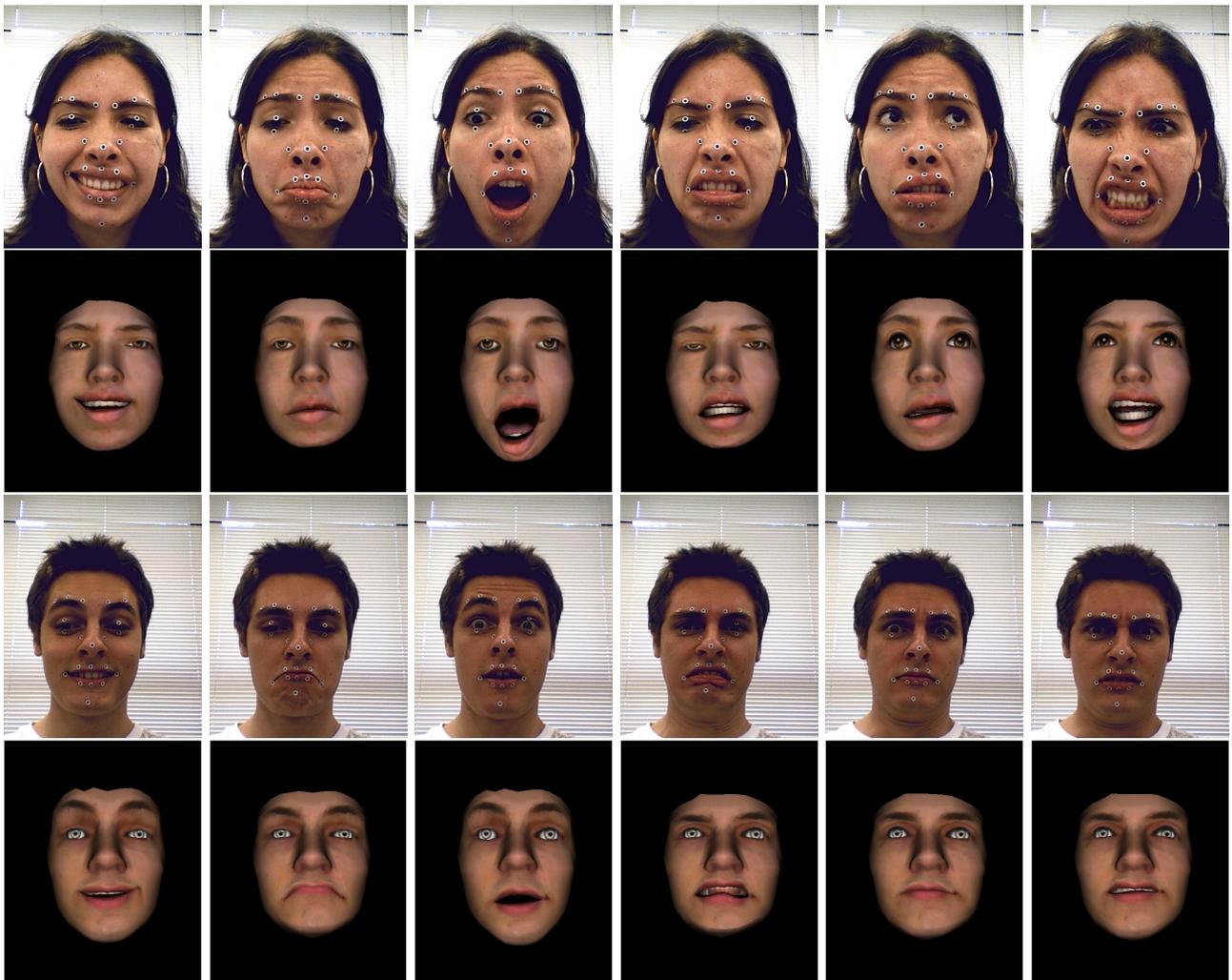


Figura 5.8: Um exemplo de *frames* capturados, com a performance de dois atores executando as expressões faciais solicitadas. Abaixo de cada um deles, o frame correspondente ao seu avatar executando a expressão.

Questionário

O questionário¹⁰ foi dividido em 3 partes, com 20 questões objetivas no total (obrigatórias), e 1 parte com uma questão dissertativa (opcional). As três partes do questionário foram elaboradas da seguinte maneira:

- **Parte 1**

Composta por 8 questões, o objetivo é verificar se os voluntários conseguem identificar a expressão do personagem virtual correspondente ao personagem real, em imagens. Exemplos das questões podem ser vistos na Figura 5.9. Nas primeiras 4 questões, utilizou-se o avatar correspondente ao ator e nas 4 seguintes, um avatar diferente. Em cada questão, existe uma das opções em que o avatar está executando

¹⁰Disponível em <http://felloosheep.wait4.org/mestrado/quiz/>

a expressão do ator em questão, e nas outras opções, o avatar está executando a mesma expressão capturada, de atores diferentes.

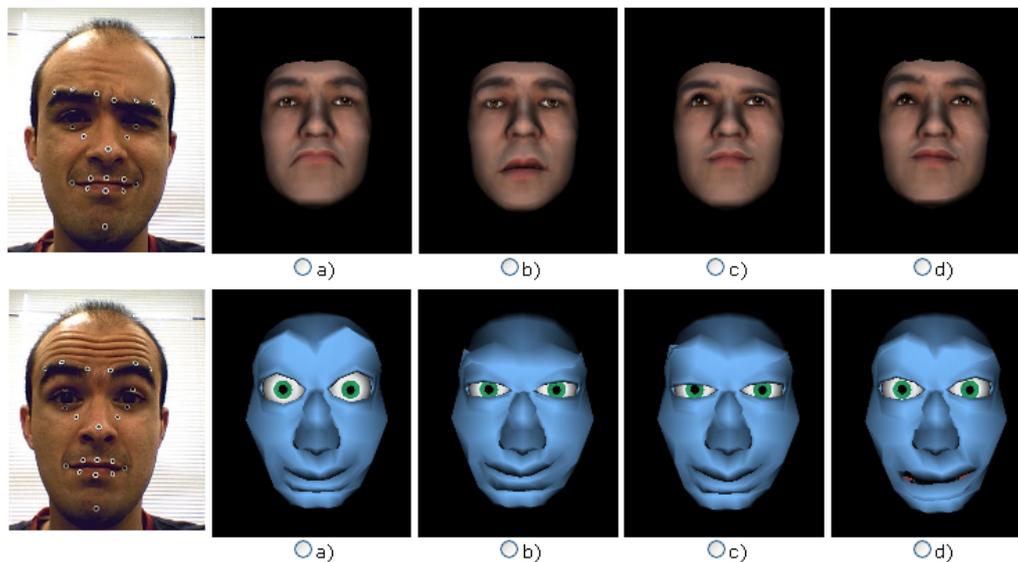


Figura 5.9: Exemplos de questões da Parte 1 do questionário. Acima, uma questão em que a face do avatar corresponde à do ator. Abaixo, uma questão em que a face do avatar não é correspondente à do ator. O usuário deve identificar a imagem em que o avatar está executando a performance do ator real.

- **Parte 2**

Composta por 7 questões, o objetivo é verificar se os voluntários conseguem associar corretamente cada uma das imagens de personagens virtuais com uma das pessoas reais, de acordo com a expressão facial deles. Um exemplo de questão pode ser visto na Figura 5.10. Nessa etapa, foi utilizado um avatar diferente de todos os atores.

- **Parte 3**

Composta por 5 questões, o objetivo é o mesmo da Parte 1, só que usando vídeos. Exemplos das questões podem ser vistos na Figura 5.11. Assim como na Parte 1, em cada questão, existe uma das opções em que o avatar está executando a animação do ator em questão, e nas outras opções, o avatar está executando a mesma animação capturada, de atores diferentes.

A questão dissertativa e opcional deixa livre para o voluntário colocar suas considerações técnicas e impressões que teve a respeito da avaliação e das animações.

5.3.2 Análise dos Resultados

O questionário foi respondido por 53 voluntários, no período de uma semana. A fim de verificar o impacto da avaliação num grupo variado de pessoas, dividiu-se (através dos

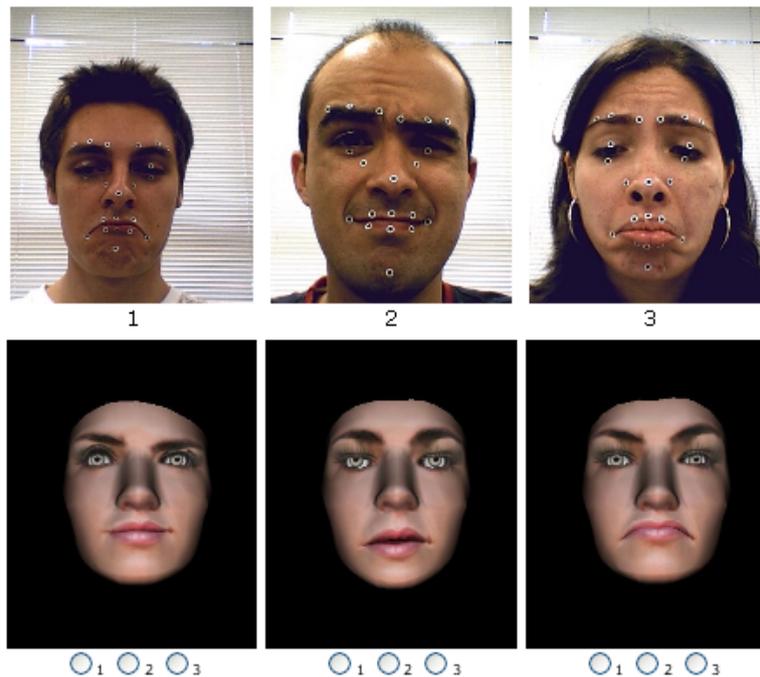


Figura 5.10: Exemplo de questão da Parte 2 do questionário, em que utiliza-se um avatar não correspondente aos atores e o usuário precisa associar para cada ator real uma expressão do avatar.

dados fornecidos sobre a pessoa) o conjunto de voluntários nos seguintes grupos:

- Grupo de pessoas que fazem parte da comunidade acadêmica do curso de Computação (Grupo **COMP**, com 44 pessoas). Desse grupo, fêz-se ainda uma subdivisão, considerando:
 - Grupo de pessoas que trabalham com Computação Gráfica ou Processamento de Imagens (Grupo **CGPI**, 28 pessoas)
 - Grupo de pessoas que não trabalham com Computação Gráfica ou Processamento de Imagens (Grupo **NCGPI**, 16 pessoas)
- Grupo de pessoas que não fazem parte da comunidade acadêmica do curso de Computação (Grupo **NCOMP**, 9 pessoas)
- Grupo de pessoas que conhecem pelo menos um ator da avaliação (Grupo **CA**, 46 pessoas)
- Grupo de pessoas que não conhecem nenhum ator da avaliação (Grupo **NCA**, 7 pessoas)

A Tabela 5.1 mostra as médias e desvios-padrão da porcentagem de acertos dos voluntários por grupo de questões, considerando também os grupos acima mencionados. A

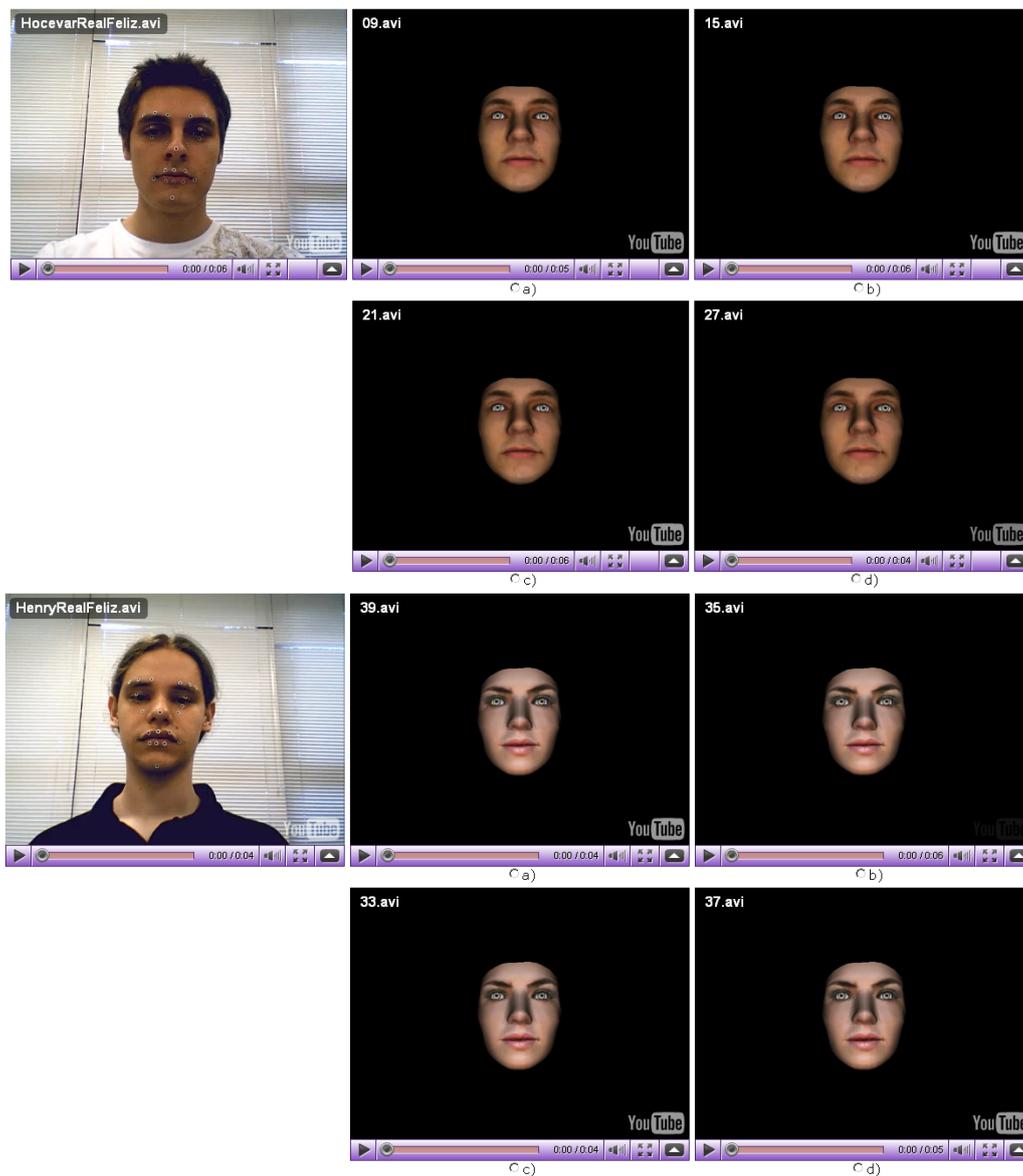


Figura 5.11: Exemplos de questões da Parte 3 do questionário. Acima, uma questão em que a face do avatar corresponde à do ator. Abaixo, uma questão em que a face do avatar não é correspondente à do ator. O usuário deve identificar o vídeo em que o avatar está executando a performance do ator real.

Figura 5.12 mostra o gráfico das médias obtidas. No geral, percebe-se que os grupos acertaram mais as partes 1 (sub-questões 1 a 4) e 2, e tiveram menos acerto na Parte 3 (questões com vídeos), nas duas subdivisões de questões. Nota-se também uma média menor de acertos na Parte 3 pelos grupos de usuários que não fazem parte da comunidade de Ciência da Computação e que não conhecem nenhum dos atores. De fato, na questão dissertativa, muitos voluntários disseram ter achado mais difícil a Parte 3, uma vez que era necessário assistir as animações sem sincronia com o vídeo da pessoa real, e que ao rodar separadamente cada vídeo ficava difícil perceber as diferenças.

Tabela 5.1: Médias e desvios-padrão dos acertos pelos voluntários por partes do questionário.

Grupos/Avaliação		Geral	Parte 1 (1-4)	Parte 1 (5-7)	Parte 2	Parte 3 (1-3)	Parte 3 (4-5)
Todos	Média	72.79%	79.63%	63.43%	78.11%	67.28%	61.11%
	Desvio-padrão	9.61%	17.56%	16.62%	11.62%	27.84%	34.57%
Grupo COMP	Média	74.24%	80.11%	63.64%	79.52%	71.21%	62.50%
	Desvio-padrão	9.29%	17.54%	16.58%	11.49%	26.50%	32.57%
Grupo NCOMP	Média	65.12%	77.78%	61.11%	71.53%	44.44%	50.00%
	Desvio-padrão	8.23%	19.54%	18.16%	11.13%	23.57%	43.30%
Grupo CGPI	Média	72.72%	77.68%	63.39%	78.06%	73.81%	60.71%
	Desvio-padrão	9.89%	17.13%	17.32%	11.37%	26.23%	31.50%
Grupo NCGPI	Média	75.69%	81.25%	67.19%	80.89%	64.58%	59.38%
	Desvio-padrão	8.15%	19.36%	11.97%	11.25%	28.46%	37.50%
Grupo CA	Média	72.83%	78.80%	62.50%	78.16%	68.84%	63.04%
	Desvio-padrão	10.01%	17.45%	16.46%	12.46%	27.58%	34.05%
Grupo NCA	Média	71.83%	85.71%	67.86%	78.23%	52.38%	42.86%
	Desvio-padrão	7.76%	19.67%	18.90%	5.11%	26.23%	34.50%

Relacionando-se as médias de alguns grupos de voluntários (Figura 5.13), pode-se fazer algumas observações:

- Os usuários da área da Computação acertaram, em média, mais questões que os que não são da área;
- Ainda que com pouca diferença, os usuários da computação que não trabalham com CG e PI, acertaram mais do que os que trabalham, exceto na Parte 3.
- Aparentemente, o fato de alguns usuários conhecerem os sujeitos avaliados no teste, não fez diferença relevante nos resultados obtidos. A diferença das médias foi maior também na Parte 3, porém, observa-se pela Tabela 5.1 que os desvios-padrão nessa etapa são maiores também, o que não permite maiores considerações sem testes estatísticos de comparação de médias.

A Figura 5.14 apresenta as médias dos grupos pelas partes do questionário separadamente. Algumas observações podem ser feitas:

- A Parte 1, nas sub-questões 1 a 4 (avatar correspondente ao ator) obtiveram maior acerto que as sub-questões 5-7 (avatar diferente), sendo os valores de 80% e 63%, na média.
- Na Parte 3 do formulário, o grupo que mais acertou em média foi o grupo de pessoas que trabalham com CG ou PI.

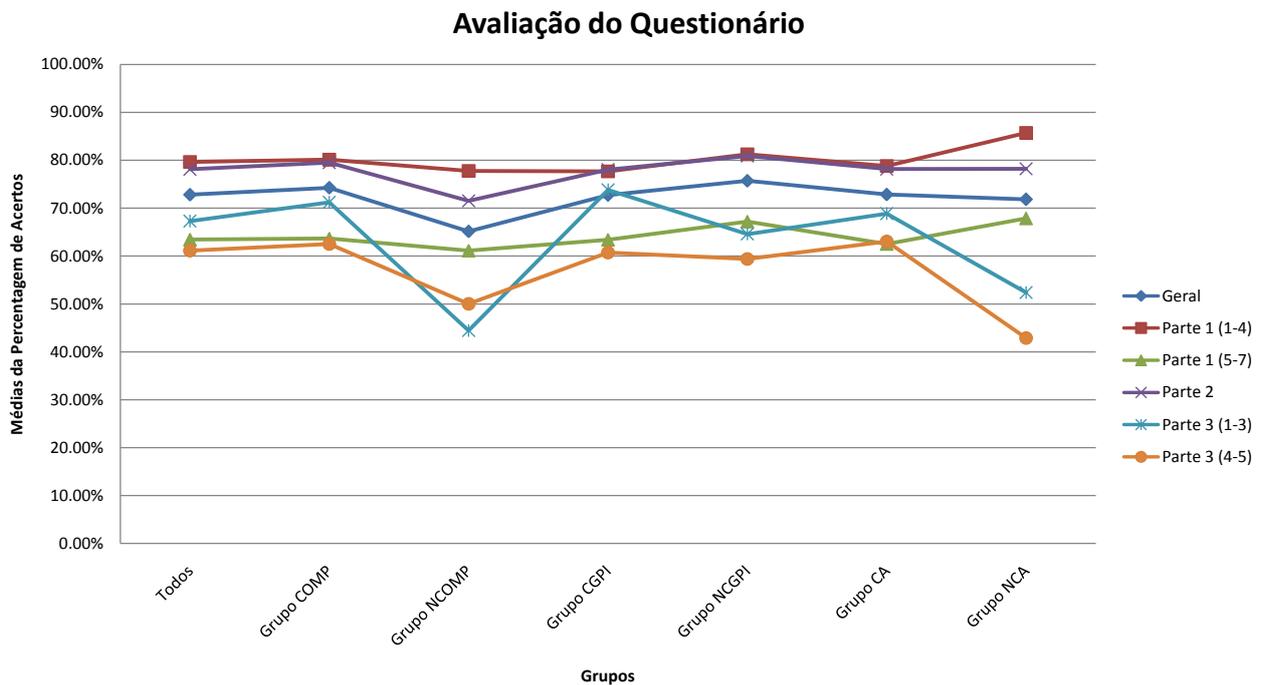


Figura 5.12: Médias dos acertos dos grupos de voluntários por partes do questionário.

Como conclusões principais, baseadas nessa análise preliminar, pode-se citar:

- Em geral, a média dos acertos foi maior nas questões em que os atores foram representados pelo seu avatar de forma correspondente (Parte 1, sub-questões 1 a 4 e Parte 3, sub-questões 1 a 3);
- A parte 3 do questionário (avaliação com vídeos) foi a que obteve as menores taxas na média de acertos, como já comentado anteriormente;
- Conhecer ou não os atores aparentemente não apresentou impacto na média de acertos.

É importante ressaltar que essa análise é bastante preliminar. Para maiores conclusões, devem ser realizados testes de significância a fim de estabelecer comparações estatisticamente válidas entre os dados obtidos.

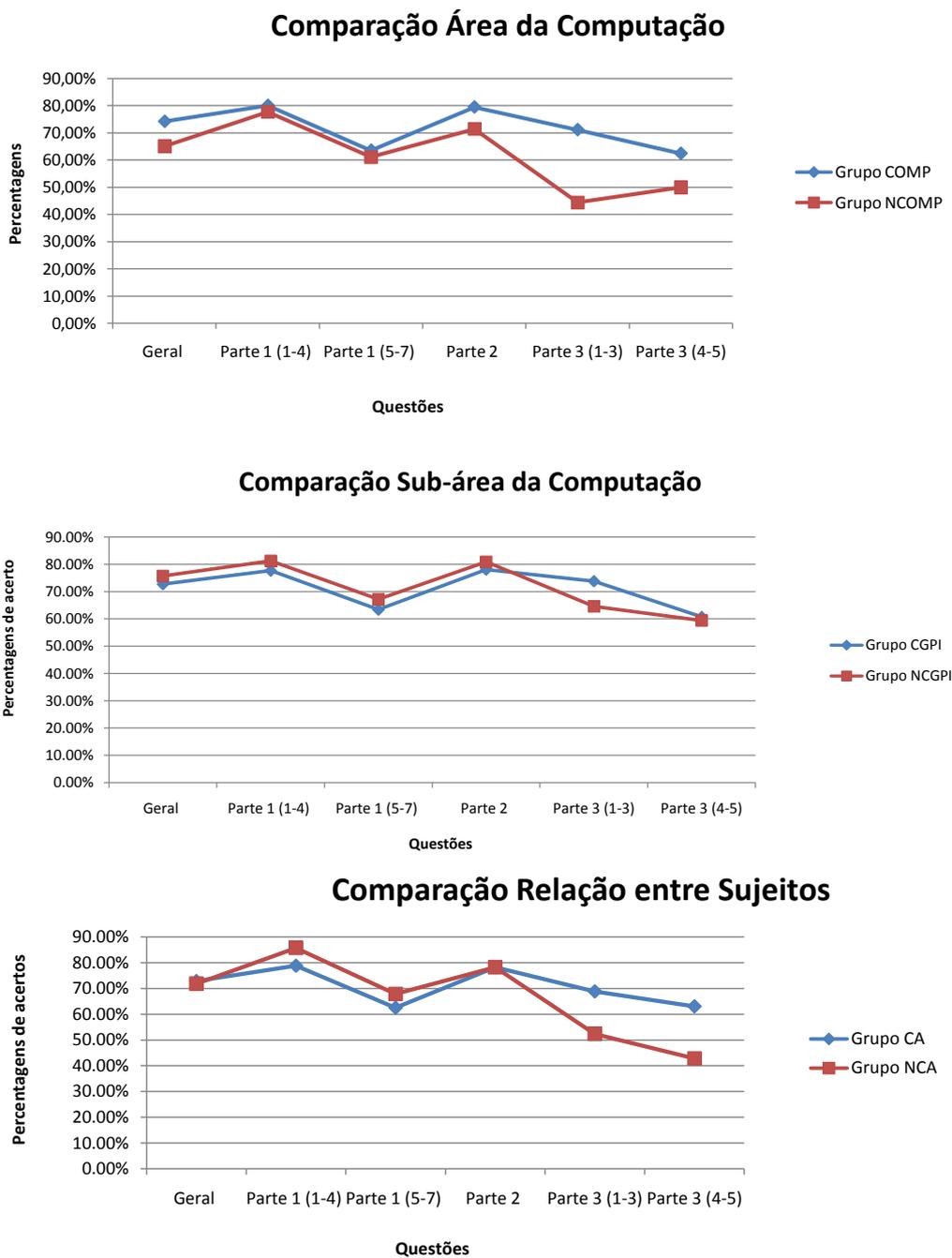


Figura 5.13: Comparação das médias entre os grupos de usuários que são o os que não são da Computação, dos que trabalham ou não com CG ou PI e dos que conhecem ou não os atores, respectivamente.

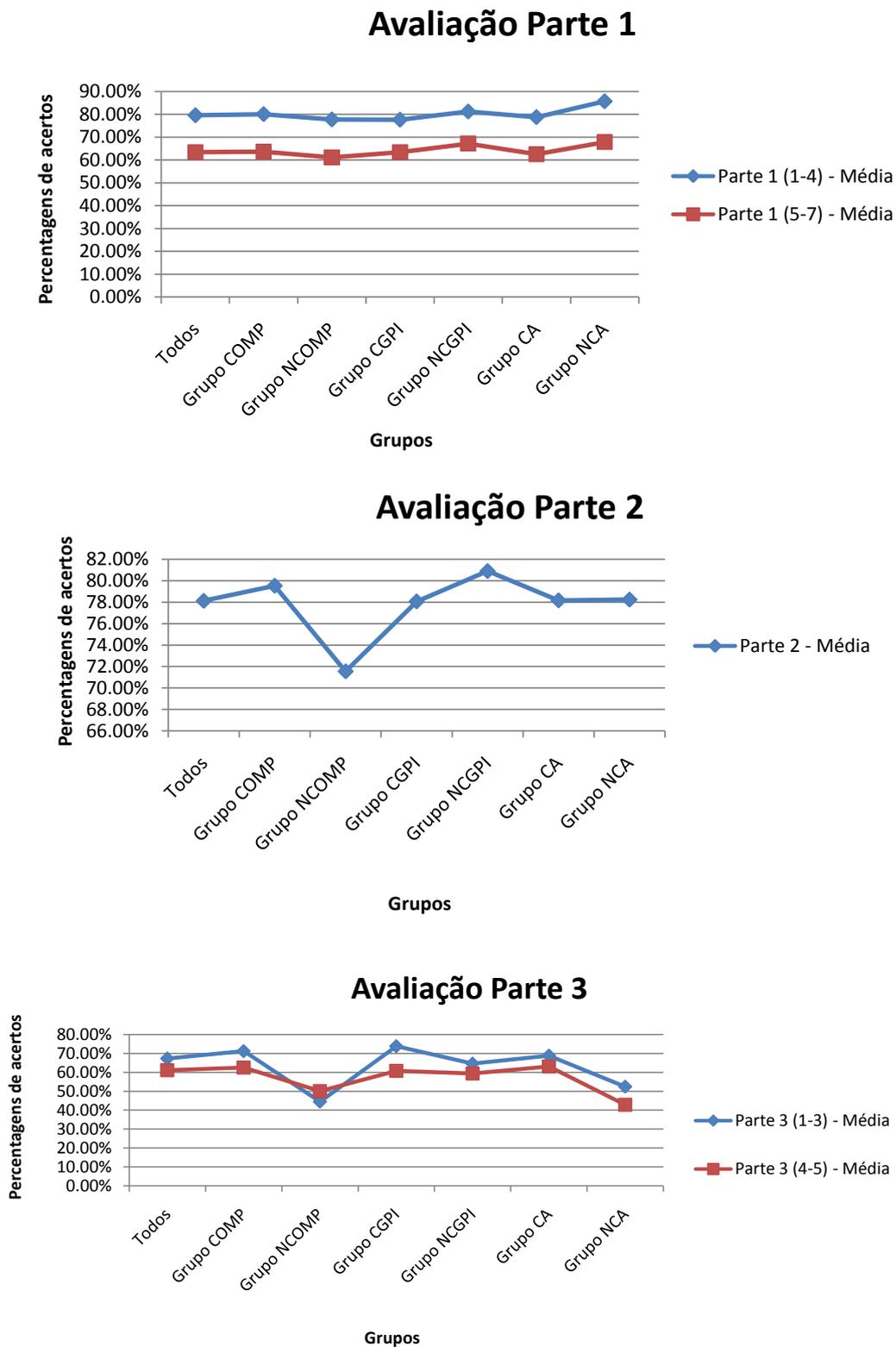


Figura 5.14: Médias das porcentagens de acertos dos grupos de usuários por cada parte do questionário.

6. Considerações Finais

Este trabalho apresentou um modelo cujo objetivo é prover animações de face personalizadas e expressivas para avatares, através da escolha de duas formas de controle: por *script* ou por Animação Dirigida por Performance.

O modelo de animação por *script* permite a geração de animações variadas, incluindo comportamentos verbais e não-verbais (fala, expressões faciais e comportamentos de olhos). Os comportamentos são descritos em alto-nível como ações faciais em uma linguagem de descrição chamada FDL. Falando em termos de “animações personalizadas”, a estratégia de controle por *script* permite a geração de animações personalizadas no sentido de que, através de um *script* FDL, editado manualmente ou suprido por um modelo que o gere baseado na detecção de eventos da aplicação, elas serão únicas, geradas em tempo real. A personalização nesse caso, está da descrição das ações, definindo o comportamento do personagem/avatar. No entanto, como comentado na Seção 4.4, no momento em que se indicar para o personagem “sorrir”, ele sempre sorrirá conforme os parâmetros de sorriso definidos no modelo de Geração de Expressões Faciais.

O estudo de caso do modelo de controle por *script* chamado “Espelho Virtual”, apresentado no capítulo anterior, mostra que, usando essa abordagem é possível se obter uma Animação Dirigida por Performance baseada em ações de alto-nível, porém sem a personalização dos movimentos, como na proposta do modelo de controle por Animação Dirigida por Performance. Nessa aplicação, a identificação de um pequeno conjunto de estados alto-nível (olhos e boca abertos ou fechados, sorriso, direção horizontal do olhar) gera animações que, apesar de “imitar” o que o usuário está fazendo, não reflete as características de sua movimentação.

Como trabalhos futuros para melhorias do modelo, pode-se citar o acréscimo de um modelo de emoções que permita a geração de expressões combinadas, como o de (68) e de uma modelo que faça a geração dos parâmetros de animação especificados nas ações faciais baseado em dicas de personalidade (baseado nas idéias do trabalho de (6)), como a geração de um “sorriso tímido”, por exemplo.

Tecnicamente, é uma contribuição desse trabalho a descrição de uma metodologia de construção de uma plataforma de animação facial usando ferramentas livres ou de código aberto, integrando alguns modelos da bibliografia capazes de produzir uma combinação substancial de ações faciais em animações de boa qualidade visual, comparada com o estado da arte das pesquisas.

O modelo de Animação Dirigida por Performance proposto permite a geração de animações, em tempo-real, que refletem as ações faciais do usuário, capturadas por uma *webcam*. Novamente falando em termos de “animação personalizada”, esse modelo gera

animações únicas, uma vez que é baseada diretamente nos dados de pessoas reais, que se movimentam de forma diferente umas das outras. Comparando com a aplicação “Espelho Virtual” citada acima e relacionando com os trabalhos apresentados na Seção 3.2, vê-se os modelos de Animação Dirigida por Performance podem ter diferentes objetivos: “imitar” a performance do usuário de forma geral, em alto-nível (o usuário sorri e o avatar sorri com o mesmo sorriso que o modelo produz para todos), traduzir a performance do ator em outra linguagem de representação (por exemplo, realista em cartum, como na proposta de (64) – o usuário sorri e o seu avatar cartum sorri como um cartum) ou tentar refletir exatamente a performance do usuário (o usuário sorri e o avatar sorri do mesmo jeito que o usuário).

No estágio atual do modelo de PDA, os algoritmos usados para o rastreamento dos pontos característicos exigem o uso de marcadores, para uma captura estável. É importante frisar que o foco da pesquisa é o mapeamento dos dados obtidos por CV para a da face virtual, portanto, o algoritmo de rastreamento usado pode ser substituído futuramente por outro mais robusto, que não necessite marcadores. Da mesma forma, como mostra o estudo preliminar de relacionamento entre FAPs baseado em informações “alto-nível” (Seção 4.3.3), é de interesse que algoritmos que forneçam essas informações sejam acrescentados, para que essa etapa do modelo seja viabilizada no processo de reflexão do usuário em tempo-real.

Ainda como trabalhos futuros para melhorias desse modelo de controle, pode-se citar uma integração com os modelos de *Geração de Expressões Faciais*, *Sincronização da Fala* e *Geração de Comportamentos dos Olhos* do modelo controle por *script* para fornecer dados que auxiliem no relacionamento entre os FAPs, quando fornecida uma informação de alto-nível pelos algoritmos de detecção/rastreamento. Haveria, portanto, a necessidade de um mapeamento das informações alto-nível identificadas e fornecidas por CV para as ações faciais do modelo de controle por *script* e um modelo de integração/combinção dos parâmetros fornecidos por esses modelos.

A respeito do modelo de *Geração/Visualização da Animação Facial*, pode-se dizer que é adequado para a geração em tempo-real das animações. Porém trata-se de um modelo bastante simples, que poderia ser melhorado com a inclusão de outras técnicas de deformação existentes na literatura, como as apresentadas na Seção 2.4, tanto para automatizar o processo de especificação dos pontos de controle e zonas de influência como para a geração de deformações mais realistas específicas de de cada região do rosto, como rugas de expressão.

Sobre a avaliação feita com usuários apresentada no capítulo anterior, percebe-se que, no geral, os usuários tiveram uma taxa de acertos satisfatória no questionário (média geral 73%), o que indica que é possível perceber a reflexão dos movimento dos atores nos avatares. É necessário, no entanto, realizar alguns testes estatísticos nos dados obtidos, a fim de se ter conclusões estatisticamente válidas sobre os pontos em que se mostrou a intenção

de investigar.

Falando de um modo geral, o modelo é promissor e gera resultados variados, “personalizados” (dentro da proposta dos modos de controle investigados) e visualmente atraentes em tempo real, alcançando satisfatoriamente os objetivos apresentados nessa dissertação.

Referências Bibliográficas

- [1] Arafa, Y.; Mamdani, A. “Multi-modal embodied agents scripting”. *Multimodal Interfaces, IEEE International Conference on 0*, 2002, 454.
- [2] Arafa, Y.; Mamdani, A. “Scripting embodied agents behaviour with cml: character markup language”. In *IUI '03: Proceedings of the 8th international conference on Intelligent user interfaces*, New York, NY, USA, 2003, ACM, pp. 313–316.
- [3] Arellano, D.; Varona, J.; Perales, F. J. “Generation and visualization of emotional states in virtual characters”. *Comput. Animat. Virtual Worlds 19*, 3-4, 2008, 259–270.
- [4] Arya, A.; DiPaola, S. “Face modeling and animation language for mpeg-4 xmt framework”. *Multimedia, IEEE Transactions on 9*, 6, Oct. 2007, 1137–1146.
- [5] Arya, A.; DiPaola, S. “Face modeling and animation language for mpeg-4 xmt framework”. *Multimedia, IEEE Transactions on 9*, 6, Oct. 2007, 1137–1146.
- [6] Arya, A.; Jefferies, L. N.; Enns, J. T.; DiPaola, S. “Facial actions as visual cues for personality: Research articles”. *Comput. Animat. Virtual Worlds 17*, 3-4, 2006, 371–382.
- [7] Balci, K. “Xface: Mpeg-4 based open source toolkit for 3d facial animation”. In *AVI '04: Proceedings of the working conference on Advanced visual interfaces*, New York, NY, USA, 2004, ACM Press, pp. 399–402.
- [8] Balci, K.; Not, E.; Zancanaro, M.; Pianesi, F. “Xface open source project and smile-agent scripting language for creating and animating embodied conversational agents”. In *MULTIMEDIA '07: Proceedings of the 15th international conference on Multimedia*, New York, NY, USA, 2007, ACM, pp. 1013–1016.
- [9] Blinn, J. F. “Simulation of wrinkled surfaces”. *SIGGRAPH Comput. Graph. 12*, 3, 1978, 286–292.
- [10] Buenaposada, J. M.; Muñoz, E.; Baumela, L. “Performance driven facial animation by appearance based tracking”. 2005, pp. 476–483.
- [11] Bui, T. D.; Heylen, D.; Nijholt, A. “Improvements on a simple muscle-based 3d face for realistic facial expressions”. In *CASA '03: Proceedings of the 16th International Conference on Computer Animation and Social Agents (CASA 2003)*, Washington, DC, USA, 2003, IEEE Computer Society, p. 33.

- [12] Byun, M.; Badler, N. I. "Facemote: qualitative parametric modifiers for facial animations". In *SCA '02: Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, New York, NY, USA, 2002, ACM, pp. 65–71.
- [13] Carolis, B. D.; Pelachaud, C.; Poggi, I.; Steedman, M. "Apml, a mark-up language for believable behavior generation". *H. Prendinger, ed., Life-like Characters.*, 2004, 65–85.
- [14] Cassell, J.; Vilhjálmsón, H. H.; Bickmore, T. "Beat: the behavior expression animation toolkit". In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, New York, NY, USA, 2001, ACM, pp. 477–486.
- [15] Chai, J.-x.; Xiao, J.; Hodgins, J. "Vision-based control of 3d facial animation". In *SCA '03: Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, Aire-la-Ville, Switzerland, Switzerland, 2003, Eurographics Association, pp. 193–206.
- [16] Chen, C.; Prakash, E. C. "Physically based facial expression synthesizer with performance analysis and gpu-aided simulation". In *CyberGames '06: Proceedings of the 2006 international conference on Game research and development*, Murdoch University, Australia, Australia, 2006, Murdoch University, pp. 171–176.
- [17] Cohen, M. M.; Massaro, D. W. "Synthesis of visible speech". *Behaviour Research Methods, Instruments & Computers* 22, 2, 1990, 260–263.
- [18] Cosi, P.; Drioli, C.; Tesser, F.; Tisato, G. "Interface toolkit: a new tool for building ivas". 75–87.
- [19] Cosi, P.; Fusaro, A.; Tisato, G. "Lucia a new italian talking-head based on a modified cohen-massaro's labial coarticulation model". In *Proceedings of Eurospeech 2003, Geneva, Switzerland, 2003*, 2003, vol. 3, pp. 2269–2272.
- [20] Cosi, P.; MagnoCaldognetto, E.; Tisato, G. "Emotional talking head: The development of "lucia"". In *CD Proceedings Workshop "Toni Mian"*, 2007.
- [21] Cosker, D.; Borkett, R.; Marshall, D.; Rosin, P. L. "Towards automatic performance-driven animation between multiple types of facial model". *Computer Vision, IET* 2, 3, 2008, 129–141.
- [22] Deng, Z.; Neumann, U.; Lewis, J. P.; Kim, T.-Y.; Bulut, M.; Narayanan, S. "Expressive facial animation synthesis by learning speech coarticulation and expression spaces". *IEEE Transactions on Visualization and Computer Graphics* 12, 6, 2006, 1523–1534.

-
- [23] Deng, Z.; Noh, J. "Computer facial animation: A survey". *Data-Driven 3D Facial Animation*. Springer London, 2007, ch. 1, pp. 1–28. This is a full INBOOK entry.
- [24] DiPaola, S. "Extending the range of facial types". *Journal of Visualization and Computer Animation* 2, 4, 1991, 129–131.
- [25] DiPaola, S.; Arya, A. "A framework for socially communicative faces for game and interactive learning applications". In *Future Play '07: Proceedings of the 2007 conference on Future Play*, New York, NY, USA, 2007, ACM, pp. 129–136.
- [26] Dutreve, L.; Meyer, A.; Bouakaz, S. "Feature points based facial animation retargeting". In *VRST '08: Proceedings of the 2008 ACM symposium on Virtual reality software and technology*, New York, NY, USA, 2008, ACM, pp. 197–200.
- [27] Dutreve, L.; Meyer, A.; Bouakaz, S. "Real-Time Dynamic Wrinkles of Face for Animated Skinned Mesh". In *ISVC' 09: 5th International Symposium on Visual Computing*, Nov. 2009.
- [28] Eck, M. "Interpolation methods for reconstruction of 3d surfaces from sequences of planar slices". *CAD und Computergraphik* 13, 5, 1991, 109–120.
- [29] Ekman, P.; Friesen, W. "Facial action code system". Consulting Psychologists Press, Inc., Palo Alto, CA, 1978.
- [30] Ersotelos, N.; Dong, F. "Building highly realistic facial modeling and animation: a survey". *Vis. Comput.* 24, 1, 2008, 13–30.
- [31] Fiore, F.; Quax, P.; Vanaken, C.; Lamotte, W.; Reeth, F. "Conveying emotions through facially animated avatars in networked virtual environments". 222–233.
- [32] Garchery, S.; Magnenat-Thalmann, N. "Designing mpeg-4 facial animation tables for web applications". In *Multimedia Modeling 2001*, May 2001, pp. 39–59.
- [33] García-Rojas, A.; Gutiérrez, M.; Thalmann, D. "Simulation of individual spontaneous reactive behavior". In *AAMAS '08: Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems*, Richland, SC, 2008, International Foundation for Autonomous Agents and Multiagent Systems, pp. 143–150.
- [34] Guye-Vuillème, A. *Simulation of nonverbal social interaction and small groups dynamics in virtual environments*. PhD thesis, Lausanne, 2004.
- [35] Haber, J.; Kähler, K.; Albrecht, I.; Yamauchi, H.; Seidel, H. P. "Face to face: From real humans to realistic facial animation". In *Proceedings Israel-Korea Binational Conference on Geometrical Modeling and Computer Graphics*, 1991, pp. 37–46.

- [36] Haber, J.; Terzopoulos, D. "Facial modeling and animation". In *GRAPH '04: Proceedings of the conference on SIGGRAPH 2004 course notes*, New York, NY, USA, 2004, ACM Press, p. 6.
- [37] Ierusalimschy, R. "Programming in lua". Lua.org, Rio de Janeiro, Brazil, 2003.
- [38] Kalra, P.; Mangili, A.; Magnenat-thalmann, N.; Thalmann, D. "Smile: A multilayered facial animation system". In *In T.L. Kunii, editor, Modeling in Computer Graphics*, 1991, Springer-Verlag, pp. 189–198.
- [39] Khanam, A.; Mufti, M. "Intelligent expression blending for performance driven facial animation". *Consumer Electronics, IEEE Transactions on* 53, 2, May 2007, 578–584.
- [40] Kähler, K.; Haber, J.; Seidel, H.-P. "Geometry-based muscle modeling for facial animation". In *GRIN'01: No description on Graphics interface 2001*, Toronto, Ont., Canada, Canada, 2001, Canadian Information Processing Society, pp. 37–46.
- [41] Kim, T.-Y.; Bulut, M. "Expressive facial animation synthesis by learning speech coarticulation and expression spaces". *IEEE Transactions on Visualization and Computer Graphics* 12, 6, 2006, 1523–1534. Member-Zhigang Deng and Member-Ulrich Neumann and Member-J. P. Lewis and Senior Member-Shrikanth Narayanan.
- [42] Kshirsagar, S.; Magnenat-Thalmann, N. "A multilayer personality model". In *SMART-GRAPH '02: Proceedings of the 2nd international symposium on Smart graphics*, New York, NY, USA, 2002, ACM, pp. 107–115.
- [43] Lee, S. P. *Facial Animation System with Realistic Eye Movement Based on a Cognitive Model for Virtual Agents*. PhD thesis, University of Pennsylvania, 2002.
- [44] Lee, S. P.; Badler, J. B.; Badler, N. I. "Eyes alive". In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, New York, NY, USA, 2002, ACM Press, pp. 637–644.
- [45] Lee, Y.; Terzopoulos, D.; Walters, K. "Realistic modeling for facial animation". *Computer Graphics 29*, Annual Conference Series, 1995, 55–62.
- [46] Lucas, B. D.; Kanade, T. "An iterative image registration technique with an application to stereo vision". In *IJCAI'81: Proceedings of the 7th international joint conference on Artificial intelligence*, San Francisco, CA, USA, 1981, Morgan Kaufmann Publishers Inc., pp. 674–679.
- [47] Magnenat Thalmann, M.; Thalmann, D. "An Overview of Virtual Humans". In *Handbook of Virtual Humans*. John Wiley, 2004, pp. 1–25.

-
- [48] Marriot, A. "Vhml-virtual human markup language". In *Talking Head Technology workshop, OZCHI 2001 Conference Computer-Human Interaction Special Interest Group's (CHISIG)*, 2001, pp. 20–22.
- [49] Not, E.; Balci, K.; Pianesi, F.; Zancanaro, M. "Synthetic characters as multichannel interfaces". In *ICMI '05: Proceedings of the 7th international conference on Multimodal interfaces*, New York, NY, USA, 2005, ACM, pp. 200–207.
- [50] Oat, C. "Animated wrinkle maps". In *SIGGRAPH '07: ACM SIGGRAPH 2007 courses*, New York, NY, USA, 2007, ACM, pp. 33–37.
- [51] Pandzic, I. S.; Forchheimer, R., Eds. "Mpeg-4 facial animation: The standard, implementation and applications". John Wiley & Sons, Inc., New York, NY, USA, 2003.
- [52] Paradiso, A. "An algebra for combining mpeg-4 compliant facial animations". In *Proc. Int. Workshop Lifelike Animated Agents: Tools, Affective Functions, and Applications*, 2002, H. Prendinger [Online].
- [53] Paragios, N.; Chen, Y.; Faugeras, O. "Handbook of mathematical models in computer vision". Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.
- [54] Parke, F. "Parameterized models for facial animation". *IEEE Computer Graphics and Applications* 2, 9, 1982, 61–68.
- [55] Parke, F. I. "Computer generated animation of faces". In *ACM'72: Proceedings of the ACM annual conference*, New York, NY, USA, 1972, ACM, pp. 451–457.
- [56] Parke, F. I. *A parametric model for human faces*. PhD thesis, 1974.
- [57] Parke, F. I.; Waters, K. "Computer facial animation". A. K. Peters, Ltd., Natick, MA, USA, 1996.
- [58] Pasquariello, S.; Pelachaud, C. "Greta: A simple facial animation engine". In *In Proc. of the 6th Online World Conference on Soft Computing in Industrial Applications*, 2001.
- [59] Perlin, K. "An image synthesizer". *SIGGRAPH Comput. Graph.* 19, 3, 1985, 287–296.
- [60] Perlin, K. "Layered compositing of facial expression". In *ACM SIGGRAPH - Technical Sketch*, 1997.
- [61] Poggi, I.; Pelachaud, C. "Signals and meanings of gaze in animated faces". In *Language, Vision and Music*, 2002, Mc Kevitt, Paul, Seán Ó Nualláin and Conn Mulvihill, pp. 133–144.

- [62] Poznanski, M.; Thagard, P. "Changing personalities: towards realistic virtual characters". *Journal of Experimental & Theoretical Artificial Intelligence* 17, 2005, 1523–1534.
- [63] Pyun, H.; Kim, Y.; Chae, W.; Kang, H. W.; Shin, S. Y. "An example-based approach for facial expression cloning". In *SCA '03: Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, Aire-la-Ville, Switzerland, Switzerland, 2003, Eurographics Association, pp. 167–176.
- [64] Quax, P.; Di Fiore, F.; Lamotte, W.; Van Reeth, F. "Efficient distribution of emotion-related data through a networked virtual environment architecture". *Computer Animation and Virtual Worlds* 9999, 9999, 2009, n/a+.
- [65] Queiroz, R. B.; Barros, L. M.; Musse, S. R. "Providing expressive gaze to virtual animated characters in interactive applications". *Comput. Entertain.* 6, 3, 2008, 1–23.
- [66] Queiroz, R. B.; Cohen, M.; Musse, S. R. "An extensible framework for interactive facial animation with facial expressions, lip synchronization and eye behavior". *Comput. Entertain.* 7, 4, 2009, 1–20.
- [67] Rodrigues, P. S.; Queiroz, R. B.; Barros, L. M.; Feijó, B.; Velho, L.; Musse, S. R. "Automatically generating eye motion in virtual agents". In *Proceedings of the IX Symposium on Virtual and Augmented Reality (SVR2007)*, 2007, pp. 84–91.
- [68] Rodrigues, P. S. L. *Um Sistema de Geração de Expressões Faciais Dinâmicas em Animações Faciais 3D com Processamento de Fala*. PhD thesis, Pontifícia Universidade Católica do Rio de Janeiro, 2007.
- [69] Rodrigues, P. S. L.; Feijó, B.; Velho, L. "Expressive talking heads: uma ferramenta com fala e expressão facial sincronizadas para o desenvolvimento de aplicações interativas". In *IX Simpósio Brasileiro de Sistemas Multimídia e WEB - WebMídia 2003*, 2003.
- [70] Rutledge, L. "Smil 2.0: Xml for web multimedia". *Internet Computing, IEEE* 5, 5, Sep/Oct 2001, 78–84.
- [71] Sanchez, M.; Edge, J.; King, S.; Maddock, S. "Use and re-use of facial motion capture data". In *Proceedings of Vision, Video, and Graphics Conference*, 2003, pp. 1–8.
- [72] Sanchez, M.; Edge, J.; Maddock, S. "Realistic performance-driven facial animation using hardware acceleration". Tech. rep., Department of Computer Science Research Memorandum CS-04-10, University of Sheffield, 2004.

-
- [73] Sanchez, M.; Maddock, S. "Planar bones for mpeg-4 facial animation". In *TPCG '03: Proceedings of the Theory and Practice of Computer Graphics 2003*, Washington, DC, USA, 2003, IEEE Computer Society, p. 81.
- [74] Sanchez, M. A. *Techniques for Performance-based, real-time Facial Animation*. PhD thesis, The University of Sheffield, 2006.
- [75] Sturman, D. J. "Computer puppetry". *IEEE Comput. Graph. Appl.* 18, 1, 1998, 38–45.
- [76] Su, W.-P.; Pham, B.; Wardhani, A. "High-level control posture of story characters based on personality and emotion". *Second Australasian conference on Interactive entertainment 23*, 2005, 179 – 186.
- [77] Tang, H.; Huang, T. "Mpeg4 performance-driven avatar via robust facial motion tracking". pp. 249–252.
- [78] Thalmann, M. N.; Primeau, E.; Thalmann, D. "Abstract muscle action procedures for human face animation". *Visual Computer* 3, 1988, 290–297.
- [79] Vilhjálmsón, H.; Cantelmo, N.; Cassell, J.; Chafai, N. E.; Kipp, M.; Kopp, S.; Mancini, M.; Marsella, S.; Marshall, A. N.; Pelachaud, C.; Ruttkay, Z.; Thórisson, K. R.; van Welbergen, H.; van der Werf, R. J. "The behavior markup language: Recent developments and challenges". In *Intelligent Virtual Agents*. Springer Link, 2007.
- [80] Viola, P.; Jones, M. "Rapid object detection using a boosted cascade of simple features". *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on 1*, 2001, 1–511–1–518 vol.1.
- [81] Walters, G. "The story of waldo c. graphic". In *ACM SIGGRAPH 89: Course Notes*, 1989.
- [82] Wang, A.; Emmi, M.; Faloutsos, P. "Assembling an expressive facial animation system". In *Sandbox '07: Proceedings of the 2007 ACM SIGGRAPH symposium on Video games*, New York, NY, USA, 2007, ACM, pp. 21–26.
- [83] Waters, K. "A muscle model for animation three-dimensional facial expression". In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, New York, NY, USA, 1987, ACM, pp. 17–24.
- [84] Weise, T.; Li, H.; Van Gool, L.; Pauly, M. "Face/off: live facial puppetry". In *SCA '09: Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, New York, NY, USA, 2009, ACM, pp. 7–16.

- [85] Williams, L. "Performance-driven facial animation". In *SIGGRAPH '90: Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, New York, NY, USA, 1990, ACM, pp. 235–242.
- [86] Woo, M.; Davis; Sheridan, M. B. "Opengl programming guide: The official guide to learning opengl, version 1.2". Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.
- [87] Wu, Z.; Zhang, S.; Cai, L.; Meng, H. M. "Real-time synthesis of chinese visual speech and facial expressions using mpeg-4 fap features in a three-dimensional avatar". In *INTERSPEECH-2006*, 2006.
- [88] Yano, K.; Harada, K. "A facial expression parameterization by elastic surface model". *Int. J. Comput. Games Technol.* 2009, 2009, 1–11.
- [89] yong Noh, J.; Fidaleo, D.; Neumann, U. "Animated deformations with radial basis functions". In *VRST '00: Proceedings of the ACM symposium on Virtual reality software and technology*, New York, NY, USA, 2000, ACM, pp. 166–174.
- [90] Zammito, V.; DiPaola, S.; Arya, A. "A methodology for incorporating personality modeling in believable game characters". In *4th Intl Conf on Games Research and Development (CyberGames)*, 2008, p. 2.
- [91] Zhang, Y.; Prakash, E. C.; Sung, E. "Real-time physically-based facial expression animation using mass-spring system". *Computer Graphics International Conference 0*, 2001, 0347.
- [92] Zoric, G.; Smid, K.; Pandzic, I. "Automated gesturing for virtual characters: speech-driven and text-driven approaches". *Image and Signal Processing and Analysis, 2005. ISPA 2005. Proceedings of the 4th International Symposium on*, Sept. 2005, 295–300.

A. Conjunto de Parâmetros de Animação

Este anexo apresenta, inicialmente, a definição de todos os FAPs do padrão MPEG-4 de Animação Facial. A Tabela A.1 mostra a relação de todos eles, sua função e agrupamento, por área do rosto. A seguir, a Tabela A.2 mostra os grupos ao qual os FAPs pertencem. Após são apresentados os novos FAPs propostos no modelo (Tabela A.3).

Tabela A.1: Definições dos FAPs. Fonte: (43)

#	Nome do FAP	Descrição do FAP	(U)ni ou (B)idirecional	Movimento Positivo	Grupo
1	viseme	um conjunto de valores para representação dos visemas	n/a	n/a	1
2	expression	um conjunto de valores para a representação de expressões faciais	n/a	n/a	1
3	open_jaw	deslocamento vertical da mandíbula	U	para baixo	2
4	lower_t_midlip	deslocamento vertical da parte média do lábio superior interno	B	para baixo	2
5	raise_b_midlip	deslocamento vertical da parte média do lábio inferior interno	B	para cima	2
6	stretch_l_cornerlip	deslocamento horizontal do canto do lábio esquerdo interno	B	esquerda	2
7	stretch_r_cornerlip	deslocamento horizontal do canto do lábio direito interno	B	direita	2
8	lower_t_lip_lm	deslocamento vertical do ponto central entre o canto esquerdo e a parte média do lábio superior interno	B	para baixo	2
9	lower_t_lip_rm	deslocamento vertical do ponto central entre o canto direito e a parte média do lábio superior interno	B	para baixo	2

10	raise_b_lip_lm	deslocamento vertical do ponto central entre o canto esquerdo e a parte média do lábio inferior interno	B	para cima	2
11	raise_b_lip_rm	deslocamento vertical do ponto central entre o canto direito e a parte média do lábio inferior interno	B	para cima	2
12	raise_l_cornerlip	deslocamento vertical do canto do lábio esquerdo interno	B	para cima	2
13	raise_r_cornerlip	deslocamento vertical do canto do lábio direito interno	B	para cima	2
14	thrust_jaw	deslocamento em profundidade da mandíbula	U	para frente	2
15	shift_jaw	deslocamento lado a lado da mandíbula	B	direita	2
16	push_b_lip	deslocamento em profundidade da parte média do lábio inferior interno	B	para frente	2
17	push_t_lip	deslocamento em profundidade da parte média do lábio superior interno	B	para frente	2
18	depress_chin	movimento de compressão para cima do queixo (como na expressão da tristeza)	B	para cima	2
19	close_t_l_eyelid	deslocamento vertical da pálpebra superior esquerda	B	para baixo	3
20	close_t_r_eyelid	deslocamento vertical da pálpebra superior direita	B	para baixo	3
21	close_b_l_eyelid	deslocamento vertical da pálpebra inferior esquerda	B	para cima	3
22	close_b_r_eyelid	deslocamento vertical da pálpebra inferior direita	B	para cima	3
23	yaw_l_eyeball	rotação horizontal do olho esquerdo	B	esquerda	3
24	yaw_r_eyeball	rotação horizontal do olho direito	B	para cima	3

25	pitch_l_eyeball	rotação vertical do olho esquerdo	B	para baixo	3
26	pitch_r_eyeball	rotação vertical do olho direito	B	para baixo	3
27	thrust_l_eyeball	deslocamento em profundidade do olho esquerdo	B	para frente	3
28	thrust_r_eyeball	deslocamento em profundidade do olho direito	B	para frente	3
29	dilate_l_pupil	dilatação da pupila esquerda	B	crescente	3
30	dilate_r_pupil	dilatação da pupila direita	B	crescente	3
31	raise_l_i_eyebrow	deslocamento vertical da sobrancelha interna esquerda	B	para cima	4
32	raise_r_i_eyebrow	deslocamento vertical da sobrancelha interna direita	B	para cima	4
33	raise_l_m_eyebrow	deslocamento vertical da sobrancelha média esquerda	B	para cima	4
34	raise_r_m_eyebrow	deslocamento vertical da sobrancelha média direita	B	para cima	4
35	raise_l_o_eyebrow	deslocamento vertical da sobrancelha externa esquerda	B	para cima	4
36	raise_r_o_eyebrow	deslocamento vertical da sobrancelha externa direita	B	para cima	4
37	squeeze_l_eyebrow	deslocamento horizontal da sobrancelha esquerda	B	direita	4
38	squeeze_r_eyebrow	deslocamento horizontal da sobrancelha direita	B	esquerda	4
39	puff_l_cheek	deslocamento horizontal da bochecha esquerda	B	esquerda	5
40	puff_r_cheek	deslocamento horizontal da bochecha direita	B	direita	5
41	lift_l_cheek	deslocamento vertical da bochecha esquerda	U	para cima	5
42	lift_r_cheek	deslocamento vertical da bochecha direita	U	para cima	5
43	shift_tongue_tip	deslocamento horizontal da ponta da língua	B	direita	6
44	raise_tongue_tip	deslocamento vertical da ponta da língua	B	para cima	6

45	thrust_tongue_tip	deslocamento em profundidade da ponta da língua	B	para frente	6
46	raise_tongue	deslocamento vertical da língua	B	para cima	6
47	tongue_roll	enrolar a língua em forma de "U"	U	côncavo para cima	6
48	head_pitch	rotação vertical da cabeça	B	para baixo	7
49	head_yaw	rotação horizontal da cabeça	B	esquerda	7
50	head_roll	rotação da cabeça em direção aos ombros (inclinar para os lados)	B	direita	7
51	lower_t_midlip_o	deslocamento vertical da parte média do lábio superior externo	B	para baixo	8
52	raise_b_midlip_o	deslocamento vertical da parte média do lábio inferior externo	B	para cima	8
53	stretch_l_cornerlip_o	deslocamento horizontal do canto do lábio esquerdo externo	B	esquerda	8
54	stretch_r_cornerlip_o	deslocamento horizontal do canto do lábio direito externo	B	direita	8
55	lower_t_lip_lm_o	deslocamento vertical do ponto central entre o canto esquerdo e a parte média do lábio superior externo	B	para baixo	8
56	lower_t_lip_rm_o	deslocamento vertical do ponto central entre o canto direito e a parte média do lábio superior externo	B	para baixo	8
57	raise_b_lip_rm_o	deslocamento vertical do ponto central entre o canto esquerdo e a parte média do lábio inferior externo	B	para cima	8

58	raise_b_lip_rm_o	deslocamento vertical do ponto central entre o canto esquerdo e a parte média do lábio inferior externo	B	para cima	8
59	raise_l_cornerlip_o	deslocamento vertical do canto esquerdo externo do lábio	B	para cima	8
60	raise_r_cornerlip_o	deslocamento vertical do canto direito externo do lábio	B	para cima	8
61	stretch_l_nose	deslocamento horizontal do lado esquerdo do nariz	B	esquerda	9
62	stretch_r_nose	deslocamento horizontal do lado direito do nariz	B	esquerda	9
63	raise_nose	deslocamento vertical da ponta do nariz	B	para cima	9
64	band_nose	deslocamento horizontal da ponta do nariz	B	direita	9
65	raise_l_ear	deslocamento vertical da orelha esquerda	B	para cima	10
66	raise_r_ear	deslocamento vertical da orelha direita	B	para cima	10
67	pull_l_ear	deslocamento horizontal da orelha esquerda	B	esquerda	10
68	pull_r_ear	deslocamento horizontal da orelha direita	B	direita	10

Tabela A.2: Grupos dos FAPs. Fonte: (43)

Grupo	Número de FAPs
1: visemas e expressões	2
2: mandíbula, queixo, lábio superior e inferior interno, cantos dos lábios interno e parte central dos lábios	16
3: olhos, pupilas e pálpebras	12
4: sobrancelhas	8
5: bochechas	4
6: língua	5
7: cabeça	3
8: posições externas dos lábios	10
9: nariz	4
10: orelhas	4

O modelo propõe a inclusão de novos parâmetros para suprir os movimentos nos eixos que não são contemplados pelos FAPs do padrão MPEG-4. Até o momento, foram adicionados os seguintes (Tabela A.3):

Tabela A.3: Definições dos novos FAPs.

#	Nome do FAP	Descrição do FAP	(U)ni ou (B)idirecional	Movimento Positivo	FAPU
69	STRETCH_T_LIP_LM	deslocamento horizontal do FP 2.6	B	para esquerda	MW
70	STRETCH_T_LIP_RM	deslocamento horizontal do FP 2.7	B	para direita	MW
71	STRETCH_B_LIP_LM	deslocamento horizontal do FP 2.8	B	para esquerda	MW
72	STRETCH_B_LIP_RM	deslocamento horizontal do FP 2.9	B	para direita	MW
73	STRETCH_T_LIP_LM_O	deslocamento horizontal do FP 8.5	B	para esquerda	MW
74	STRETCH_T_LIP_RM_O	deslocamento horizontal do FP 8.6	B	para direita	MW
75	STRETCH_B_LIP_LM_O	deslocamento horizontal do FP 8.7	B	para esquerda	MW
76	STRETCH_B_LIP_RM_O	deslocamento horizontal do FP 8.8	B	para direita	MW
77	STRETCH_T_LIP	deslocamento horizontal do FP 2.2	B	para direita	MW
78	STRETCH_B_LIP	deslocamento horizontal do FP 2.3	B	para direita	MW
79	STRETCH_T_LIP_O	deslocamento horizontal do FP 8.1	B	para direita	MW
80	STRETCH_B_LIP_O	deslocamento horizontal do FP 8.2	B	para direita	MW
81	NOSE_R_UP	deslocamento vertical do FP 9.14	B	para cima	MNS
82	NOSE_L_UP	deslocamento vertical do FP 9.13	B	para cima	MNS
83	NOSE_R_STRETCH	deslocamento horizontal do FP 9.14	B	para direita	MW
84	NOSE_L_STRETCH	deslocamento horizontal do FP 9.13	B	para esquerda	MW

B. Mapeamento Fonema-Visema

Visema #	Fonemas	Exemplos
0	nenhum	na
1	p,b,m	put, bed, mill
2	f,v	far, voice
3	T,D	think, that
4	t,d	Tip, doll
5	k,g	call, gas
6	tʃ,dʒ,S	chair, join, she
7	s,z	Sir, zeal
8	n,l	Lot, not
9	r	red
10	A:	car
11	e	bed
12	ɪ	Tip
13	Q	top
14	U	book

Tabela B.1: Relação Fonema-Visema pelo padrão MPEG-4 (51)

Tabela B.2: Mapeamento Fonemas-Visemas MPEG-4 proposto por Rodrigues (68)

Fonema	Exemplo	Visema #	Fonema	Exemplo	Visema #
i:	beet	12	l	bit	12
E	bet	11	@	bat	13
^	above	13	u	boot	14
U	book	14	&	above	10
A	father	10	3r	bird	9
&r	butter	9	ei	bay	12
al	bye	10	oi	boy	13
iU	few	12	aU	about	10
oU	boat	14	l	lent	8
9r	rent	9	j	yes	12
w	went	14	m	me	1
n	knee	8	N	sing	8
f	fine	2	T	thigh	3
s	sign	7	S	assure	6
h	hope	9	v	vine	2
D	thy	3	z	resign	7
Z	azure	7	ph	pan	1
th	tan	4	kh	can	5
b	ban	1	d	dan	4
g	gander	5	th_(writer	4
d_(rider	4	tS	church	6
dZ	judge	6			