5
DynaFeX: Uma Ferramenta de Edição e Exibição de Expressões Faciais Dinâmicas

Como apresentado no Capítulo 3, esta tese propõe um modelo, denominado *VeeM* (*Virtual emotion-to-expression Model*), para geração de novas emoções (chamadas de "emoções derivadas") e para visualização de expressões faciais. De acordo com o que foi discutido na Seção 3.6, cada emoção possui uma expressão facial associada e cada expressão é representada por um vetor, denominado $\vec{fev}$ (*facial expression vector*). O $\vec{fev}$ armazena o valor do deslocamento de cada ponto que define a face para que se obtenha a expressão facial desejada.

Este capítulo apresenta a ferramenta desenvolvida para incorporar o *VeeM* e as emoções geradas a partir da especificação do hipercubo emocional. A ferramenta possui o nome *DynaFeX* (*Dynamic Facial eXpression*) e está implementada na linguagem de programação C++ (Stro97). Sua implementação faz uso do pacote gráfico *wxWidgets* (SmHC05). A *DynaFeX* tem como um dos seus objetivos mapear vetores $\vec{fev}$ em uma face tridimensional realista conforme com o padrão MPEG-4, respeitando o sincronismo dos aspectos emocionais e da fala da personagem definidos pelo usuário.

Como já motivado ao longo deste documento, um dos desafios encontrados na área de animação facial é a junção de expressões faciais com a fala para ter a animação de uma personagem virtual com esses elementos sincronizados. Na literatura de animação facial, esse tipo de personagem é comumente chamada de *talking head*. Minimamente, três fatores definem um *talking head* de sucesso, expressando naturalidade na animação facial construída (PaWa96):

- Movimento labial adequado na sincronização da fala;
- Expressão facial não-verbal relacionada com a face, como levantar as sobrancelhas, piscar, mexer a cabeça; e
- Expressão facial das emoções.

Qualquer um desses pontos que falte na animação sintetizada faz com que a animação seja percebida como "sem vida", entediante e artificial. Com isso, a ferramenta *DynaFeX* busca contemplar esse três fatores e também incorporar

os fenômenos afetivos, mencionados na Seção 3.5, para se ter uma animação facial mais próxima do natural.

Este capítulo está estruturado da seguinte forma. A Seção 5.1 oferece uma visão geral de como a ferramenta implementada está organizada. As cinco seções seguintes abordam os módulos existentes na ferramenta *DynaFeX*: fornecimento de arquivos de entrada (Seção 5.2), geração da estrutura fonética da fala (Seção 5.3), definição e refinamento de emoções para geração de expressões faciais (Seção 5.4), combinação dos FAPs (Seção 5.5) e execução da animação facial com sincronismo da fala e dos aspectos emocionais (Seção 5.6). A Seção 5.7 descreve a incorporação dos fenômenos afetivos (Seção 3.5) na ferramenta e a Seção 5.8 finaliza este capítulo apresentando sistemas desenvolvidos ao longo deste doutorado onde personagens virtuais com expressões faciais foram utilizados.

5.1 Visão Geral da DynaFeX

DynaFeX é uma ferramenta que tem o propósito de permitir que animadores profissionais e usuários comuns construam animações faciais de forma simples. A ferramenta incorpora um modelo rico de emoções para geração de expressões faciais, almejando proporcionar uma fácil interação. A Figura 5.1 ilustra a estruturação da ferramenta, onde as "setas" indicam possíveis *pipelines* de execução através dos principais módulos.

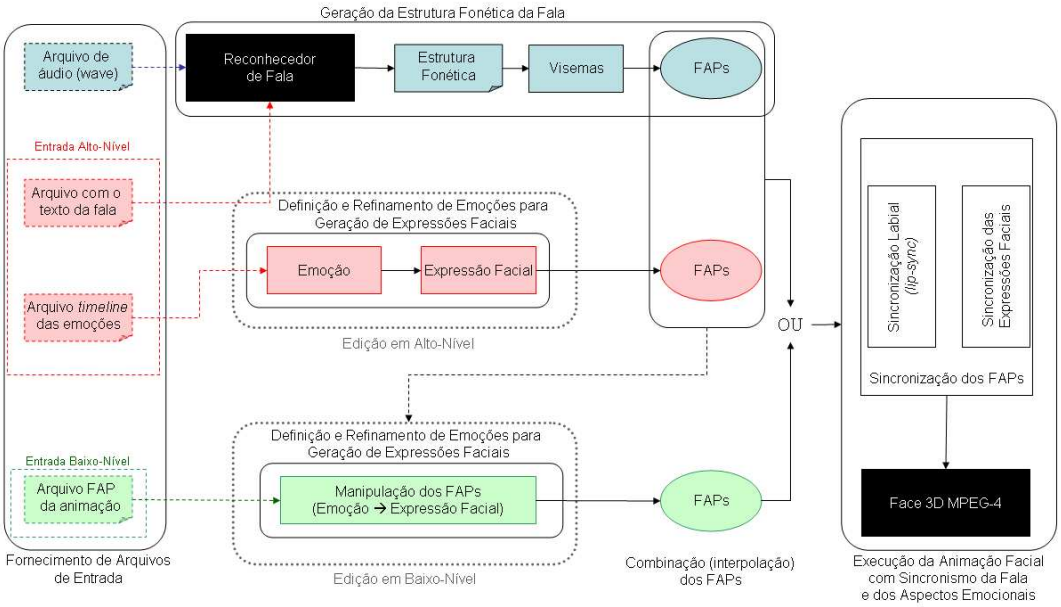

Figura 5.1: Visão geral da ferramenta *DynaFeX*.

Como ilustrado na figura e comentado anteriormente, a ferramenta está estruturada em cinco módulos principais:

- *Fornecimento de Arquivos de Entrada*: módulo em que os arquivos de entrada são tratados, quando fornecidos. Além do arquivo de áudio contendo a fala da personagem, a ferramenta trabalha com dois tipos possíveis de entrada de dados, uma com arquivos descritos em alto-nível ("linguagem textual") e outra com arquivos descritos em baixo-nível (FAPs MPEG-4). A Seção 5.2 explica o funcionamento completo desse módulo.
- *Geração da Estrutura Fonética da Fala*: esse módulo é responsável por tratar o arquivo de áudio fornecido como entrada e extrair as informações fonéticas necessárias para o sincronismo labial. A Seção 5.3 descreve o funcionamento desse módulo.
- *Definição e Refinamento de Emoções para Geração de Expressões Faciais*: esse módulo é responsável pela criação de novas emoções e pelo refinamento das emoções definidas para geração da expressão facial em cada quadro (ou grupo de quadros) da animação. Esse módulo pode receber como entrada o "arquivo *timeline* das emoções" ou o usuário pode, interativamente, definir as emoções usando a ferramenta. A Seção 5.4 explica o funcionamento de todos os submódulos dessa etapa.
- *Combinação dos FAPs*: uma vez gerada a estrutura baixo-nível (FAPs) de cada um dos módulos citados anteriormente, esse módulo é responsável por combinar (interpolar) os valores para definir a estrutura única de FAPs para cada quadro da animação. A Seção 5.5 descreve o procedimento executado para combinação dos vários FAPs.
- *Execução da Animação Facial com Sincronismo da Fala e dos Aspectos Emocionais*: módulo responsável pela sincronização da fala e dos componentes faciais. É nesse módulo que o usuário dispara a apresentação da animação e obtém o resultado completo. A Seção 5.6 detalha o funcionamento desse módulo.

É importante ressaltar que para a construção de uma animação facial na ferramenta *DynaFeX*, o usuário não precisa passar por todos esses módulos. Cenários distintos existem para a especificação de uma animação facial através de diferentes "combinações" dos módulos disponíveis. Alguns dos elementos que podem acarretar em diferentes cenários são enumerados a seguir:

1. O arquivo contendo a fala da personagem é um dado opcional. É possível gerar uma animação com sincronismo da fala ou uma animação onde sejam visualizadas apenas as expressões geradas.

2. O usuário pode especificar arquivos de entrada para iniciar a animação. Esses arquivos de entrada podem ser especificados de duas formas: "formato alto-nível (textual)" e "formato baixo-nível (FAPs)". Esses arquivos não são obrigatórios, podendo a especificação da animação facial ser feita desde o princípio, de forma interativa, e depois salva para uma nova edição/execução .

3. É possível fazer um refinamento das emoções já carregadas pelos arquivos de entrada como também especificar novas emoções para um dado instante de tempo ou quadro da animação. No caso do refinamento, dependendo do formato especificado na entrada, tem-se uma "edição em alto-nível" ou uma "edição em baixo-nível". A geração de novas emoções para a construção da animação facial pode também ocorrer nos dois níveis de interação.

Partindo para uma visão mais detalhada da *DynaFeX*, a implementação da ferramenta está estruturada segundo a especificação do *pattern Mediator* (Gamm95). Esse padrão de projeto tem o propósito de concentrar o conhecimento do sistema em uma classe, uma vez que a ferramenta *DynaFeX* é composta por uma grande quantidade de classes, várias delas adaptadas de outros sistemas, conforme ficará mais claro ao longo deste capítulo. O padrão também serve para intermediar a comunicação entre as classes de interface, as classes do modelo de dados e as classes da lógica de execução, conforme ilustrado pela Figura 5.2. A ferramenta disponibiliza uma interface interativa para manipulação de dados do usuário. Cada interação do usuário dispara uma chamada ao mediador que verifica e direciona o tratamento para a área da lógica responsável por esse tratamento. A lógica retorna a solução para o mediador que, para a maioria das ações, aplica esse resultado na face para visualização.

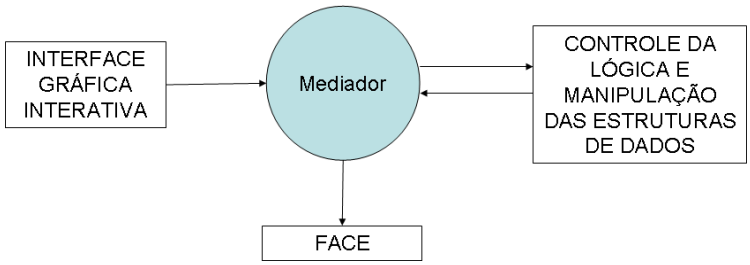

Figura 5.2: Padrão de projeto aplicado na *DynaFeX*.

5.2 Fornecimento de Arquivos de Entrada

Este módulo da *DynaFeX* é responsável por interpretar arquivos de dados fornecidos pelo usuário como entrada. As saídas produzidas por este módulo são entradas para outros módulos, como sinalizado pelos *pipelines*, ilustrados na Figura 5.1.

O arquivo que descreve a fala da personagem em modo textual simples (*plain text*) é uma das possíveis entradas especificadas neste módulo. Os outros arquivos que a ferramenta espera receber são classificados de acordo com a sua especificação. Existe uma entrada alto-nível onde os arquivos são descritos em uma linguagem mais fácil de manipular e mais próxima de uma descrição textual. A outra possibilidade é uma entrada baixo-nível, onde os arquivos são descritos em uma linguagem mais próxima de ser executada. Ambos os tipos de entrada têm o propósito de fornecer uma descrição temporal da animação facial.

5.2.1 Arquivo de Áudio

Ao definir uma animação facial, o usuário pode escolher incorporar ou não uma fala à personagem. A opção de não inserir a fala pode ser interessante em casos onde o usuário queira verificar se as expressões faciais estão traduzindo a emoção esperada ou, por exemplo, usar a ferramenta em uma aplicação para treinamento de leitura labial.

Se o usuário optar por incorporar uma fala à animação, deve ser fornecido um arquivo de áudio no formato *wave*[1]. Como trabalho futuro, a ferramenta pode ser estendida para funcionar com outros formatos de áudio de entrada, como *mp3*, *mp4* etc.

Quando presente, o arquivo de áudio serve como entrada para o módulo da *DynaFeX* responsável por reconhecer e extrair as informações da fala da personagem, conforme comentado na próxima seção.

5.2.2 Arquivos para Entrada Alto-Nível: Texto da Fala e Timeline da Emoção

Ao fornecer arquivos como entrada, o usuário precisa definir se seus arquivos estão descritos em um alto-nível ou em um baixo-nível de abstração. No caso do tratamento dos arquivos descritos em alto-nível, a *DynaFeX*

[1]Especificação do formato *wave* disponível em *http://ccrma.stanford.edu/courses/422/projects/WaveFormat/* (acesso em 04/Nov/2007).

necessita de um processamento mais elaborado, principalmente se o arquivo de áudio estiver presente para geração da animação facial.

Considerando o fornecimento do arquivo de áudio contendo a fala da personagem, é necessário que, além do áudio, o usuário forneça também um arquivo contendo o texto correspondente à fala. Esses dois arquivos são a base para uma importante fase, "reconhecedor de fala" (Seção 5.3.1), e são fundamentais para geração da animação facial com a fala sincronizada.

Além do arquivo com o texto da fala, o usuário pode fornecer um arquivo contendo uma descrição temporal do estado emocional esperado para a personagem. A Figura 5.3 exemplifica o formato dos arquivos esperados para a entrada no modo alto-nível. Como pode ser observado, o arquivo opcional de emoções é descrito como uma lista de tuplas $< instante\_inicial >, < instante\_final >, < nome\_emocao1 >, < intensidade\_emocao1 >, < nome\_emocao2 >, < intensidade\_emocao2 >$, onde $< instante\_inicial >$ e $< instante\_final >$ descrevem os instantes absolutos em que uma determinada emoção deve ser expressada e suspensa, respectivamente, e a emoção é descrita através da combinação de duas emoções básicas ($< nome\_emocao1 >$ e $< nome\_emocao2 >$), dadas com as respectivas intensidades ($< intensidade\_emocao1 >$ e $< intensidade\_emocao2 >$), que devem variar no intervalo [0, 1] [2]. Os parâmetros $< nome\_emocao2 >$ e $< intensidade\_emocao2 >$ são opcionais, permitindo que, alternativamente, o usuário especifique um comportamento baseado em uma única emoção. O mesmo efeito é obtido se o usuário especificar uma das emoções como sendo o estado de neutralidade (emoção natural).

Nos instantes de tempo em que a personagem não possui emoções associadas, a mesma permanece no estado natural, podendo, no entanto, ter um movimento labial associado à fala, assim como movimentos de olhos e cabeça. Assim, a ausência do arquivo de descrição temporal da emoção faz com que a animação esteja sempre no estado de neutralidade da emoção.

5.2.3 Arquivos para Entrada Baixo-Nível: FAPs da Animação

No caso de uma entrada baixo-nível, o usuário deve carregar um arquivo no formato *fap*. O arquivo deve conter, na primeira linha, a taxa de quadros por segundo e o número total de quadros. Em seguida, o arquivo descreve o valor que cada um dos FAPs definidos pelo padrão MPEG-4 deve assumir em cada quadro da animação. Para cada quadro, primeiro segue-se uma linha

[2] O intervalo de intensidades é normalizado para os limites válidos dos valores de cada FAP.

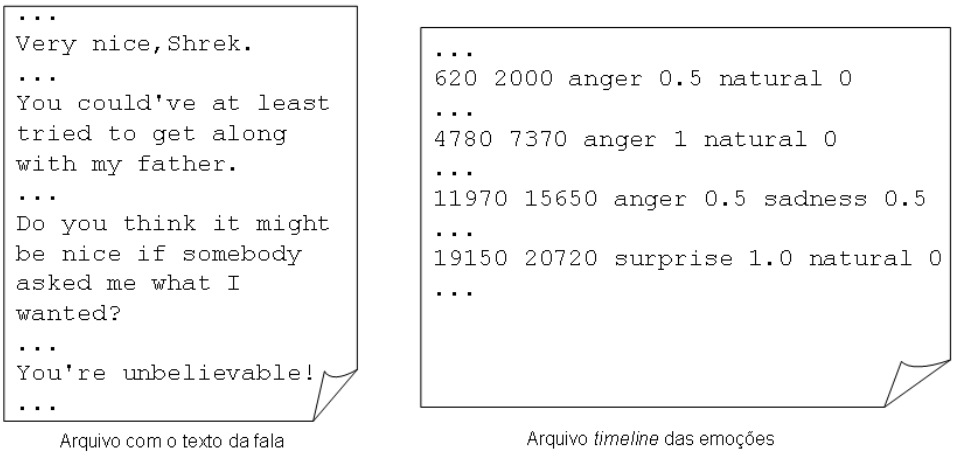

Figura 5.3: Exemplo de descrição textual e de emoção.

contendo a máscara dos 68 FAPs. Nessa linha, valores iguais a 1 indicam que o FAP deve ter seu valor considerado, enquanto valores iguais a 0 indicam que o FAP correspondente deve permanecer com o valor neutro naquele quadro. A linha de máscara é seguida pelos valores propriamente ditos dos FAPs para o quadro em questão. Apenas os FAPs que possuem máscara igual a 1 são especificados (os demais são omitidos, economizando espaço no arquivo). A Figura 5.4 exemplifica o formato através de um trecho de um arquivo *fap*. Como o arquivo *fap* descreve todos os quadros da animação, esse arquivo, juntamente com o arquivo de áudio (quando houver), é suficiente para descrever o *script* da animação facial com a fala e os aspectos emocionais sincronizados, não sendo necessários o texto com a descrição da fala e o *script* de emoção.

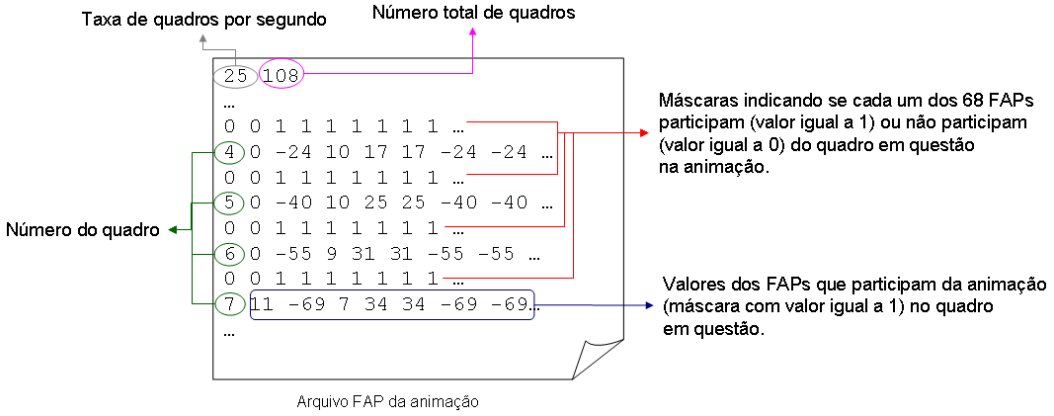

Figura 5.4: Exemplo de arquivo *fap*.

5.3 Geração da Estrutura Fonética da Fala

Este segundo módulo da ferramenta *DynaFeX* é ativado pelo arquivo de áudio contendo a fala da personagem juntamente com o arquivo textual que descreve a fala. Sendo assim, este módulo só é utilizado nas animações que possuem áudio associado e têm como entrada os dados em alto-nível de abstração. A Figura 5.5 ilustra a estruturação interna deste módulo e as subseções seguintes descrevem cada um dos componentes em maiores detalhes.

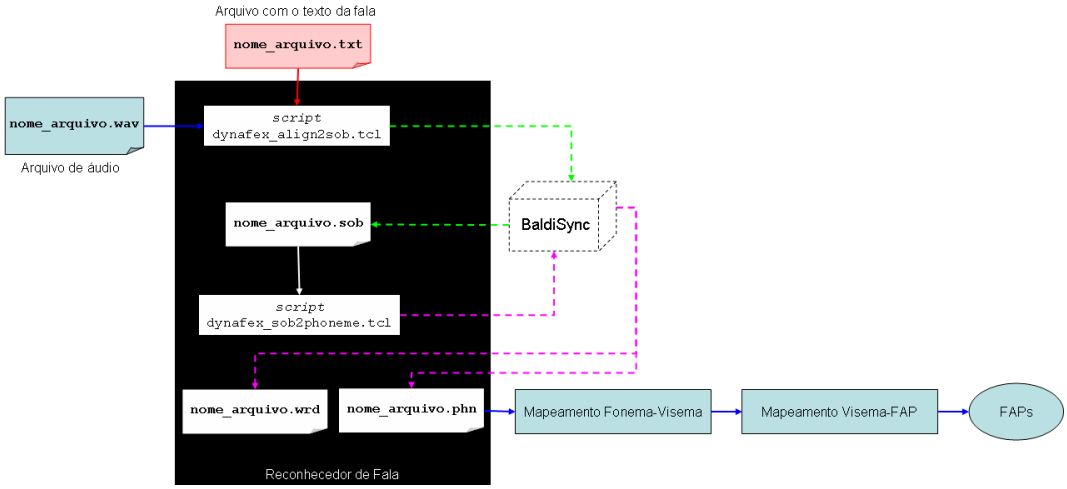

Figura 5.5: Módulo de estrutura fonética da fala.

5.3.1 Reconhecedor de Fala

O componente "reconhecedor de fala" é responsável por extrair, a partir do áudio e do arquivo que descreve a fala da personagem, as informações fonéticas para geração do *script* da animação. Para realização do reconhecimento da fala foi incorporada à *DynaFeX* um módulo externo, chamado *Baldisync* (Cole99), proveniente de um conjunto de ferramentas para síntese e reconhecimento de áudio.

Para obter a saída desejada, a entrada é submetida a dois *scripts*, como ilustrado na Figura 5.5. Ambos os *scripts* foram implementados utilizando a linguagem TCL [3] e são extensões de *scripts* definidos pelo *Baldisync* para evitar que o usuário tenha que interagir diretamente com a interface gráfica do *Baldisync*.

O primeiro *script*, *dynafex_align2sob.tcl*, recebe como entrada o áudio e o texto da fala (formatos *wave* e *txt*, respectivamente) e gera como saída um arquivo no formato *sob*, que é um formato binário contendo tanto os dados

[3]Especificação descrita em *http://wiki.tcl.tk/* (acesso em 07/Nov/07).

do áudio como as informações de transcrição. Internamente, o *script* chama o módulo *Baldisync* para fazer o reconhecimento da fala e um alinhamento temporal dos fonemas extraídos com o áudio da fala, objetivando atingir uma precisão no sincronismo labial.

O arquivo *sob* é a entrada para a execução do segundo *script*, denominado *dynafex_sob2phoneme.tcl*. Esse *script* entrega o arquivo *sob* para o *Baldisync* que, por sua vez, transforma as informações do arquivo *sob* em dois arquivos mais simples de serem interpretados por uma ferramenta externa: um arquivo contendo os fonemas da fala da personagem e outro contendo as palavras da fala da personagem. Ambos os arquivos são descritos de forma semelhante: uma lista de tuplas $< instante\_inicial >, < instante\_final >, < nome\_fonema >$ para o arquivo de fonemas (extensão *.phn*) e uma lista de tuplas $< instante\_inicial >, < instante\_final >, < palavra\_textual >$ para o arquivo de palavras (extensão *.wrd*). A Figura 5.6 apresenta exemplos dos dois arquivos gerados nessa fase para o arquivo de entrada ilustrado na Figura 5.3.

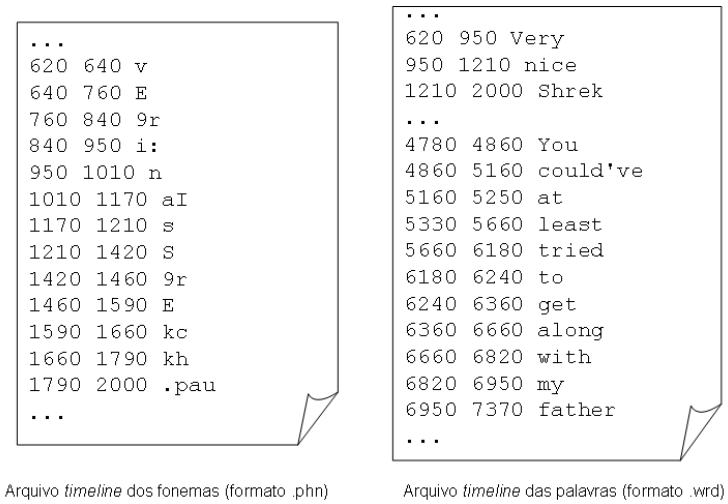

Figura 5.6: Arquivos de saída do reconhecedor de fala.

O arquivo contendo a descrição *timeline* dos fonemas é a única entrada atualmente necessária para o próximo componente deste módulo: "mapeamento fonema-visema". Na atual implementação, o arquivo de palavras não é utilizado, mas pode ser aproveitado em versões futuras para estender a interface e permitir uma interação do usuário diretamente com o texto da fala.

5.3.2 Mapeamento de Fonemas para Visemas

A área de lingüística classifica os sons da fala em um número abstrato de categorias, chamadas de fonemas. Cada língua (idioma) possui um conjunto

próprio de fonemas, não havendo restrições ou limitações para essa quantidade.

A ferramenta *DynaFeX* opera atualmente apenas com os fonemas da língua inglesa. Basicamente, a razão para essa escolha é a universalidade do idioma inglês e o fato do *Baldisync* não oferecer suporte para o idioma português. Também não foi possível encontrar reconhecedores para a língua portuguesa que oferecessem suporte semelhante ao fornecido pelo *Baldisync* (a pesquisa e as ferramentas para o idioma português brasileiro ainda são incipientes (NeSS05)). Um terceiro aspecto relevante para a escolha é o uso do padrão MPEG-4, que especifica os visemas tendo como referência o idioma inglês (Seção 4.5).

Em termos de avaliação, como a maioria das ferramentas para animação facial com fala foram desenvolvidas para o inglês, a análise comparativa da *DynaFeX* não é comprometida por essa escolha. Todavia, entende-se que um importante trabalho futuro é a incorporação de suporte à língua portuguesa. Cabe observar que a ferramenta *DynaFeX* foi projetada para que novos idiomas possam ser facilmente acrescentados, bastando, para isso, obter o suporte por parte da ferramenta de extração dos fonemas, e modificar as bases de fonemas e de mapeamento para visemas utilizadas. Uma possível referência para inclusão de uma base de visemas para o idioma português é o trabalho (MaMV06).

O inglês americano, por exemplo, tem cerca de 41 fonemas, embora esse número varie de acordo com o dialeto do interlocutor e do sistema lingüístico usado na classificação. O módulo *Baldisync* gera um subconjunto desses possíveis 41 fonemas e, conseqüentemente, esse subconjunto é utilizado pela ferramenta *DynaFeX*[4].

Cada um dos fonemas na *DynaFeX* precisa ser mapeado em um visema do padrão MPEG-4 (Seção 4.5) para visualização durante a fala. Esse mapeamento é feito com base na Tabela 5.1.

Internamente, a Tabela 5.1 é armazenada como uma estrutura de dados de tabela *hash* onde a chave é o fonema gerado a partir do reconhecedor da fala. O valor armazenado para cada chave da tabela corresponde ao identificador do visema. Esse último serve como entrada para o mapeamento na representação baixo-nível de FAPs.

5.3.3

[4]Os fonemas que estão sendo considerados na ferramenta *DynaFeX* foram retirados da especificação de fonemas definida em *http://cslu.cse.ogi.edu/tutordemos/SpectrogramReading/ipa/ipahome.html* (acesso em 06/Nov/07).

Tabela 5.1: Mapeamento dos fonemas para visemas.

Fonema	Exemplo	Id do Visema	Fonema	Exemplo	Id do Visema
i:	*beet*	12	I	*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
aI	*bye*	10	¿i	*b{oy*	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			

Mapeamento de Visemas para FAPs

Como apresentado na Seção 4.5, o padrão MPEG-4 define um conjunto de 15 visemas (incluindo o silêncio) para representação dos fonemas. O MPEG-4 opta por agrupar os fonemas cujos posicionamentos dos lábios são similares, definindo uma base de visemas *muitos-para-um* (vários fonemas podem ser representados por um mesmo visema).

Uma vez definidos os 15 visemas que compõem a base MPEG-4, o passo seguinte é representá-los através dos parâmetros do padrão: os FAPs. Uma representação direta e trivial é obtida através do FAP 1, pertencente ao grupo 1 de FAPs. O grupo 1 é chamado de grupo de alto-nível, pois o FAP 1 recebe um valor entre $[0, 14]$ referente à identificação do visema que a face deve renderizar no instante da fala. Contudo, a representação em alto-nível do visema precisa ser traduzida em uma descrição baixo-nível dos pontos de controle da face (FAPs dos grupos 2 a 10). Apesar dessa necessidade, o padrão não especifica esse mapeamento. Como conseqüência, é uma importante contribuição desta tese esse trabalho de tradução.

Com relação aos 9 grupos de FAPs de baixo-nível, os grupos 2 e 8 referem-se, especificamente, à região da boca em um modelo facial MPEG-4. Mais especificamente, o grupo 2 define 16 FAPs que atuam nas áreas do maxilar, queixo, lábio inferior interno, canto dos lábios e meio dos lábios, e o grupo 8 define 10 FAPs que movimentam as porções externas dos lábios. Sendo assim, cada visema é especificado através da valoração desses 26 FAPs.

A Figura 5.7 ilustra os 15 visemas especificados pelo padrão MPEG-4 e renderizados pela face tridimensional na ferramenta *DynaFeX*. Os valores dos FAPs (parâmetros baixo-nível adotados na *DynaFeX*) para cada um dos visemas a fim de que seja atingida a expressão facial ilustrada na Figura 5.7 são descritos no Apêndice C.

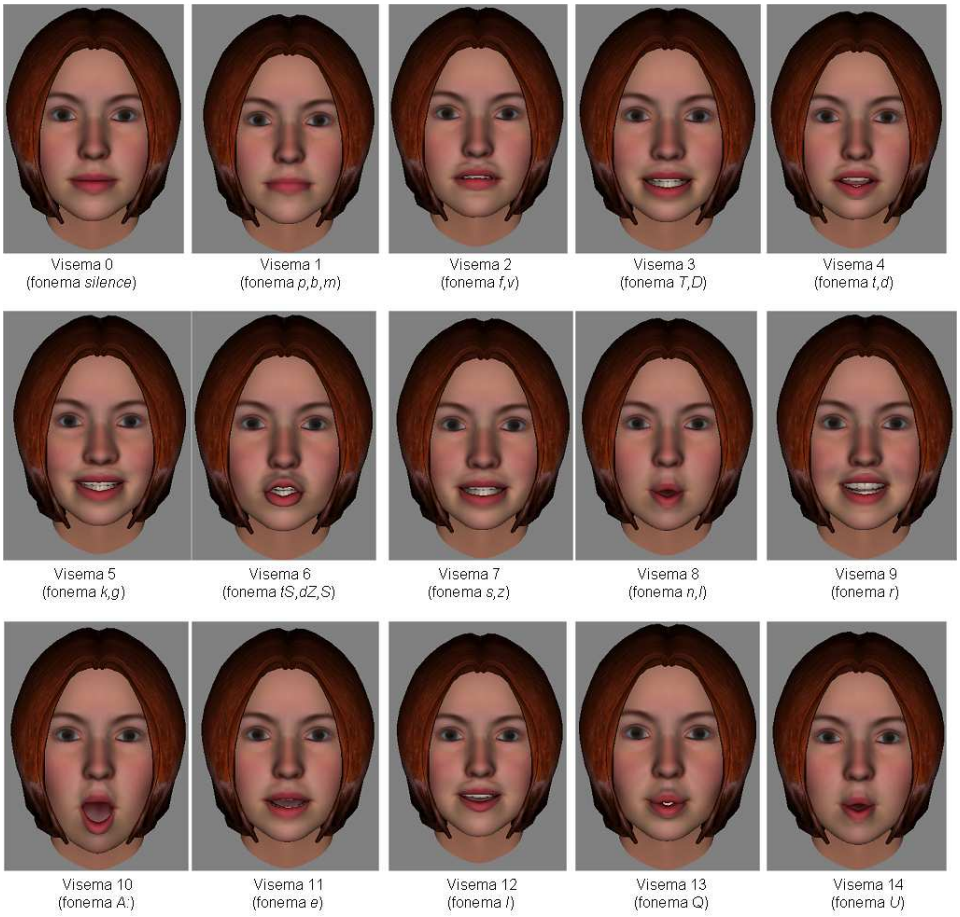

Figura 5.7: Expressão facial dos 15 visemas da *DynaFeX*.

Analogamente ao mapeamento fonema-visema, o mapeamento de visemas para FAPs de baixo-nível é armazenado como uma estrutura de dados de tabela *hash*, onde a chave é o identificador numérico do visema (saída do mapeamento fonema-visema). O valor armazenado para cada chave da tabela corresponde a um vetor armazenando os 26 FAPs que compõem a região da boca. Cada posição do vetor contém o valor que o FAP correspondente deve

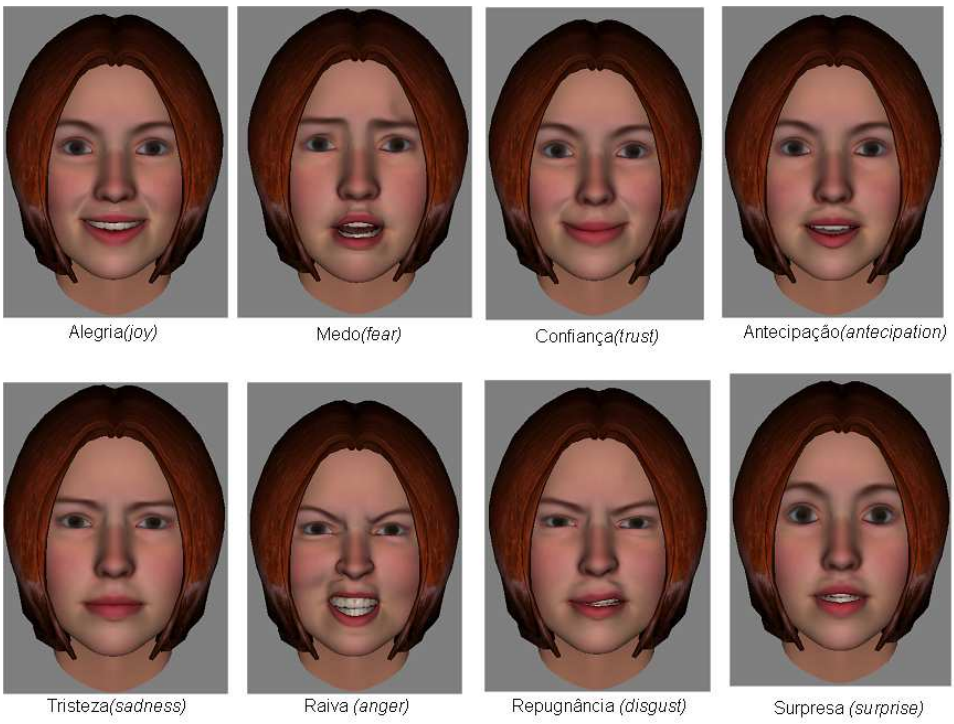

Figura 5.8: Expressões puras do *VeeM* na *Dynafex*.

assumir para formar o visema.

5.4 Definição e Refinamento de Emoções para Geração de Expressões Faciais

O terceiro módulo da ferramenta *DynaFeX* apresentado neste capítulo caracteriza-se por ser um módulo interativo que permite ao usuário definir emoções básicas ou combinar emoções, gerando emoções derivadas, para posterior visualização através das expressões faciais. A Figura 5.8 ilustra as oito emoções básicas especificadas (visualizadas) através de valores para os FAPs definidos na *DynaFeX*.

Este módulo pode assumir duas formas de manipulação para especificação das emoções: a "edição em alto-nível" e a "edição em baixo-nível", conforme descrito nas próximas subseções.

5.4.1 Edição em Alto-Nível: Mapeamento de Emoção para Expressões Faciais

A edição é considerada em alto-nível se o animador interage diretamente com as emoções, não precisando manipular os FAPs do padrão MPEG-4. Apesar do grupo 1 do padrão dedicar um FAP (FAP 2) para expressões faciais (Tabela 4.1), a *DynaFeX* não usa esse FAP, pois o FAP 2 padronizado pelo MPEG-4 trabalha apenas com as seis expressões universais, enquanto que a

DynaFeX permite que o usuário visualize todas as possíveis emoções originadas através do *VeeM*. A Figura 5.9 ilustra a interface de manipulação que o usuário possui para especificação, em alto-nível, de uma emoção a ser visualizada.

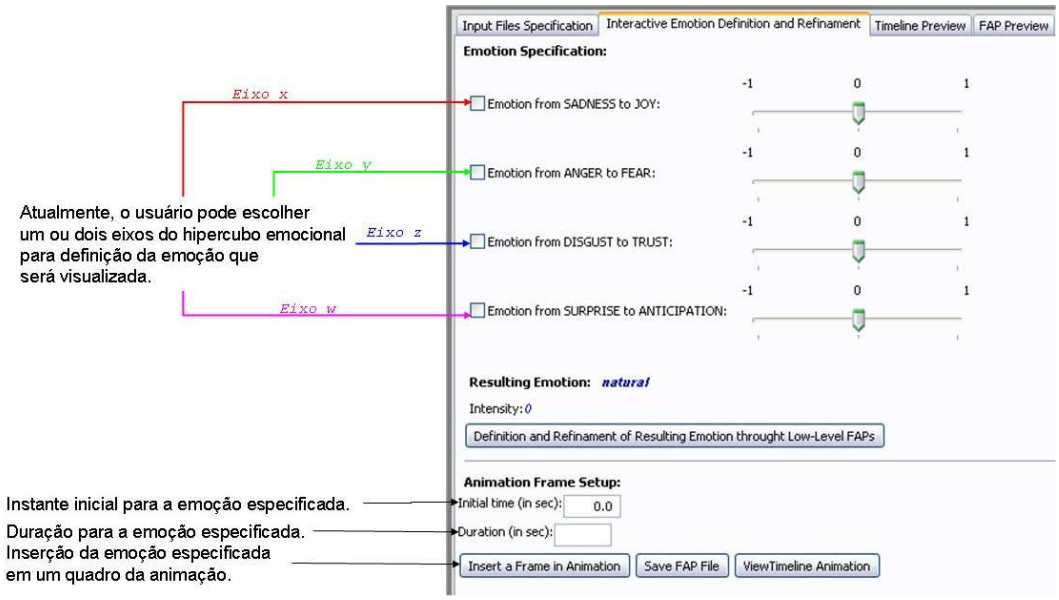

Figura 5.9: Edição alto-nível.

A manipulação em alto-nível permite que o usuário trabalhe com dois cenários: refinamento da emoção e definição da emoção, ambos podendo ser especificados para quadros da animação. O refinamento da emoção ocorre quando o usuário fornece, através do arquivo *timeline* das emoções (Figura 5.3), uma configuração inicial de qual emoção (pura ou derivada) a personagem deve expressar para cada intervalo de tempo (intervalo de quadros da animação). A partir do arquivo com a definição temporal das emoções, o usuário pode refinar o tempo de visualização da emoção, como também sua intensidade. É possível ainda que emoções sejam substituídas.

A definição da emoção considera que o usuário não forneceu o arquivo *timeline* das emoções. Nesse caso, o usuário pode especificar a emoção através da manipulação dos eixos de emoção, especificando também o instante inicial (campo *Initial time*) e a duração (campo *Duration*), como indicado na Figura 5.9. A definição ocorre, efetivamente, quando o usuário solicita à ferramenta a inserção do grupo de quadros na animação (através do botão *Insert Frame(s) in Animation* na Figura 5.9).

Assim como ocorre com os visemas, um mapeamento entre a especificação alto-nível da expressão para a sua representação de acordo com os FAPs de baixo-nível é necessária. Esse mapeamento também é realizado por este submódulo e os valores exatos utilizados para as expressões puras são descritos no Apêndice C.

Para converter a especificação de uma emoção pura em valores dos FAPs, é feita uma normalização da intensidade com base nos limites que cada FAP pode assumir para o eixo da emoção em questão. Tem-se que $\forall j, j \in N$ e $j \in [3, 68], FAP_j = min_j + \alpha_e * (max_j - min_j)$, onde α_e é a intensidade da emoção pura E, e $[min_j, max_j]$ define, para a emoção E, o intervalo de valores que podem ser atribuídos ao FAP_j.

No Apêndice C são apresentadas duas tabelas que especificam os limites para os FAPs de cada uma das emoções dos quatro eixos utilizados na implementação do *VeeM*. A Tabela C.1 determina os valores máximo e mínimo de cada FAP para as quatro emoções dos eixos x e y do hipercubo emocional, enquanto a Tabela C.2 o faz para as quatro emoções dos eixos z e w.

Os intervalos adotados na *DynaFeX* são baseados na proposta de Tsapatsoulis et al. (Tsap02). É importante mencionar que a referência (Tsap02) sugere apenas o intervalo para as seis expressões universais de Ekman, sendo uma contribuição da *DynaFeX* a especificação dos intervalos para as expressões faciais das emoções "confiança" (*trust*) e "antecipação" (*anticipation*).

A tradução para FAPs é feita antes de iniciar a apresentação da animação, ou se o usuário escolhe mudar o modo de edição de alto-nível para baixo-nível. A última situação normalmente ocorre quando o usuário deseja ter um controle mais fino sobre os quadros, conforme explicado a seguir.

5.4.2 Edição em Baixo-Nível: Manipulação Direta dos FAPs

Na edição baixo-nível o usuário especifica as emoções para os quadros da animação utilizando, diretamente, os valores dos FAPs necessários para a emoção em questão. De forma análoga à edição alto-nível, a *DynaFeX* permite que o usuário edite emoções existentes ou defina novas emoções.

O refinamento da emoção ocorre quando o usuário fornece o arquivo FAP da animação, exemplificado na Figura 5.4. Para cada quadro da animação, a expressão formada a partir do arquivo de entrada pode ser modificada alterando o valor dos FAPs. A Figura 5.10 ilustra a manipulação baixo-nível para os FAPs da região da boca.

A definição de uma nova emoção ocorre com o usuário especificando um valor para cada FAP. De forma análoga à definição em alto-nível, o usuário deve especificar o instante inicial e a duração e, posteriormente, solicitar a inserção dos quadros com os valores dos FAPs especificados.

Um ponto interessante da edição baixo-nível é a liberdade que ela oferece ao usuário. É possível especificar poses além das expressões faciais geradas

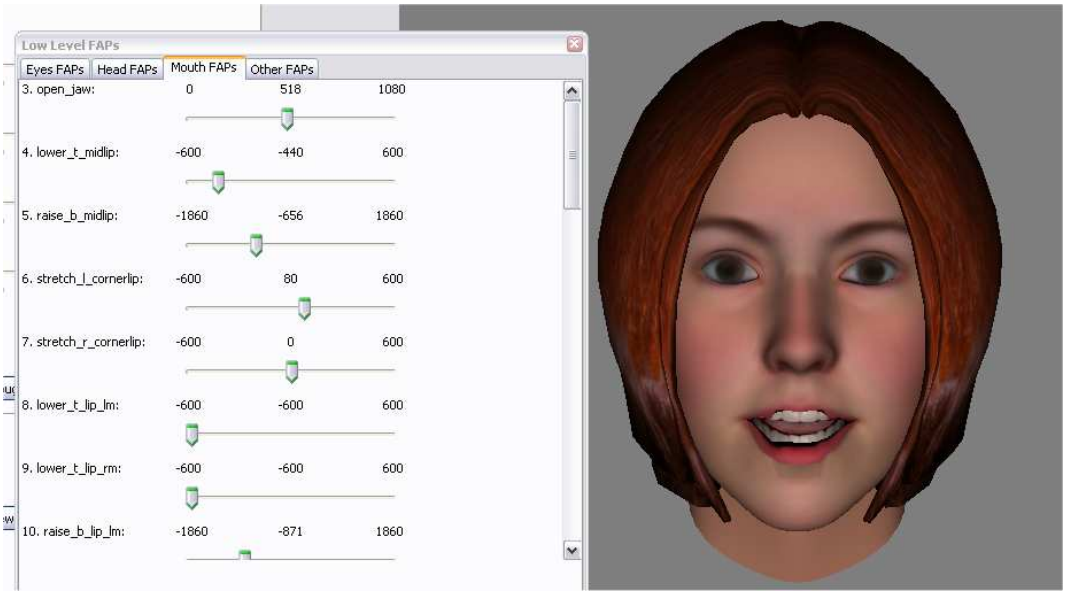

Figura 5.10: Edição baixo-nível.

a partir do *VeeM*. Por exemplo, o usuário pode acentuar as expressões da personagem nos quadros da animação conferindo uma expressão mais caricata.

5.5 Combinação dos FAPs

Como descrito pelos possíveis *pipelines* da Figura 5.1, os valores para os FAPs gerados nos diferentes módulos precisam ser combinados (interpolados) para definir o valor de cada FAP em cada quadro da animação (como descrito no formato de arquivo ilustrado na Figura 5.4). Para cada quadro da animação, a combinação de FAPs é feita em dois estágios:

- Escolha da expressão facial para a emoção;
- Escolha da expressão facial a partir da emoção resultante e do visema.

A Figura 5.11 esquematiza a organização dos estágios e as entradas e saídas de cada estágio no módulo de combinação de FAPs.

O primeiro estágio pode receber como entrada nenhum vetor de FAPs, um único vetor de FAPs ou dois vetores de FAPs. Qualquer que seja a entrada, o resultado são dois vetores de dimensão 68: um vetor contendo a máscara dos FAPs e um outro com os valores propriamente ditos dos FAPs resultante.

Se nenhum vetor é fornecido como entrada, o resultado do primeiro estágio é o vetor de FAPs para o estado natural e uma máscara contendo o valor 0 para os 26 FAPs da região da boca (Apêndice B) e para o FAP 1 (visema), e o valor 1 para os demais FAPs. O objetivo dessa configuração

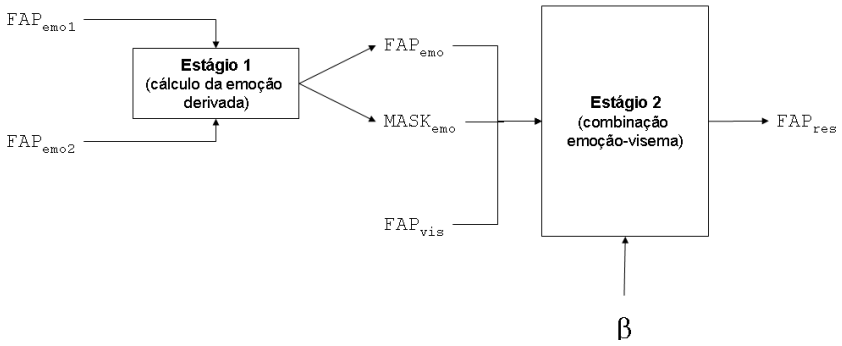

Figura 5.11: Módulo de combinação de FAPs.

é fazer com que a face permaneça no estado natural, tendo a sua expressão influenciada apenas pelos FAPs dos visemas.

No caso em que um único vetor de FAPs é fornecido, o resultado do primeiro estágio é o próprio vetor de FAPs e uma máscara contendo o valor 1 para todos os FAPs, com exceção do FAP 1 (visema).

Quando dois vetores de FAPs são entregues para o primeiro estágio, ocorre a combinação para construção da emoção derivada. A combinação é obtida calculando-se a média entre os valores de cada FAP, com exceção dos FAPs 1 e 2, que são os FAPs de alto-nível. No FAP 1 é atribuído o valor 0, uma vez que esse FAP descreve o visema. Já no FAP 2, que descreve a emoção, é atribuído o valor 0, pois esse valor não interfere na animação, uma vez que apenas os FAPs de baixo-nível são considerados pelo módulo de sincronização da *DynaFeX*.

Se os dois vetores de FAPs estiverem presentes na entrada, a máscara gerada como saída é idêntica à criada no caso de só haver um vetor de entrada.

Fenômenos afetivos não estão implementados na versão atual da *DynaFeX*, mas quando presentes devem influenciar no cálculo de combinação realizado nesse primeiro estágio.

O segundo estágio da combinação gera o vetor de FAPs resultante a partir da saída do primeiro estágio (vetor de FAPs de emoção e máscara de FAPs de emoção), do vetor de FAPs do visema e de um parâmetro de contribuição do visema, denotado por β, com $0 \leq \beta \leq 1$.

Antes de aplicar a regra de combinação visema-emoção, o segundo estágio gera uma máscara para o vetor de FAPs do visema. Havendo um vetor de FAPs para o visema, o vetor da máscara gerado possui valor igual a 1 para o FAP 1 e os 26 FAPs relacionados à região da boca (Apêndice B), e 0 para os demais FAPs. Caso não haja um vetor de FAPs de visema especificado na entrada do segundo estágio, a máscara é criada com todos os elementos do vetor tendo valor igual a 0.

A regra para combinação é simples. Denotando por FAP_{emo} e $MASK_{emo}$, respectivamente, o vetor de FAPs e o vetor da máscara da emoção resultante; FAP_{vis} e $MASK_{vis}$, respectivamente, o vetor de FAPs do visema e a máscara criada no segundo estágio; e β o índice de contribuição do visema, tem-se que, para cada índice i do vetor de FAPs resultante FAP_{res}:

- se $(MASK_{vis} = 0) \Rightarrow FAP_{res_i} = MASK_{emo_i} * FAP_{emo_i}$
- se $((MASK_{vis} \neq 0) \wedge (MASK_{emo} = 0)) \Rightarrow FAP_{res_i} = MASK_{vis_i} * FAP_{vis_i}$
- se $((MASK_{vis} \neq 0) \wedge (MASK_{emo} \neq 0)) \Rightarrow FAP_{res_i} = (1 - \beta) * MASK_{emo_i} * FAP_{emo_i} + \beta * MASK_{vis_i} * FAP_{vis_i}$

Pela regra, é possível observar que o valor $\beta = 1$ corresponde à estratégia de combinar visemas e emoções sobrepondo os visemas e ignorando a influência da emoção na geração da expressão facial na região da boca. O valor β pode ser ajustado na ferramenta para realização de análises qualitativas do resultado visual obtido, ficando como sugestão para um trabalho futuro.

5.6 Execução da Animação Facial com Sincronismo da Fala e dos Aspectos Emocionais

O quinto e último módulo da ferramenta *DynaFeX* a ser descrito é o módulo de apresentação, responsável pela execução da animação facial sintetizada a partir dos dados provenientes dos outros módulos. Esse módulo, além de mapear os valores dos FAPs armazenados para cada instante (ou quadros) da animação na malha da face, permite também que o usuário armazene no formato baixo-nível (arquivo *fap*) o *script* da animação. A vantagem do módulo permitir a gravação (armazenamento) dos FAPs é poder, posteriormente, recarregar a mesma animação e refiná-la, se desejável, utilizando o modo baixo-nível de operação.

A geração do *script* de animação no formato baixo-nível é um procedimento bastante simples, uma vez que a apresentação já utiliza os dados nesse formato. Basta ao usuário solicitar a operação de "salvar o arquivo FAP" (equivalente ao botão *"Save FAP File"* na Figura 5.9). A saída da operação é um arquivo no formato ilustrado na Figura 5.4.

O procedimento de apresentação é um pouco mais elaborado. Uma vez que os FAPs foram unificados (combinados) em uma única estrutura de dados, como descrito na Seção 5.5, a *DynaFeX* tem a responsabilidade de transicionar os valores dos FAPs em cada quadro sincronizadamente com a

fala da personagem (quando presente) e de forma bastante suave, procurando fornecer uma naturalidade e maior proximidade de realismo na animação final.

A modelagem facial da *DynaFeX* é reusada de um outro projeto código-aberto (*open source*) denominada *Xface*[5] (Balc04) (Balc05). A ferramenta *Xface* foi projetada para ser uma ferramenta para desenvolvimento de agentes conversacionais tridimensionais, com sua malha modelada segundo a especificação do padrão MPEG-4. A ferramenta é formada por quatro pacotes, um básico e três aplicações independentes construídas sobre esse pacote básico:

- *XfaceCore* é a biblioteca básica, orientada a desenvolvedores. Todas as outras aplicações da ferramenta *Xface* utilizam essa biblioteca.
- *XfaceEd* é um editor que fornece uma interface para gerar malhas MPEG-4 a partir de modelos 3D estáticos;
- *XfacePlayer* é o apresentador ("tocador") das animações definidas em arquivos no formato *fap* com suporte a sincronismo da exibição com áudios capturados; e
- *XfaceClient*, que é usada como controladora de comunicação para transmissão de apresentações MPEG-4 através de uma rede.

Todas as aplicações da *Xface* são implementadas na linguagem de programação C++ (Stro97). O projeto oferece uma ferramenta que favorece a extensibilidade (Balc04). Das três aplicações da *Xface*, a *XfaceEd* (Balc05) foi detalhadamente estudada e estendida para que alguns de seus módulos fossem incorporados à *DynaFeX*.

Seguindo o padrão MPEG-4 de animação facial, a *XfaceEd* permite que usuários especifiquem os "pontos de definição facial" (FPD) e as "unidades dos parâmetros da animação" (FAPU). A ferramenta permite também a definição da zona de influência de cada ponto característico e como essa influência é propagada entre os vértices vizinhos. A ferramenta especifica os 15 visemas do MPEG-4 e as seis expressões universais de Ekman (Seção 2.2) utilizando a linguagem VRML e arquivos no formato *wrl* [6]. Internamente esses valores são mapeados para os FAPs MPEG-4.

O componente principal responsável pelo controle da execução da animação na ferramenta *Xface* é modelado pela classe *FaceView*. Originalmente, essa classe oferecia um único método para receber como parâmetro o caminho de um arquivo de FAPs e gerar a animação a partir da leitura desse arquivo, chamado *previewFAP*. Para permitir a interação do usuário na fase

[5] *Download* da ferramenta disponível em *http://xface.itc.it/* (acesso em 16/Nov/2007).

[6] Informações sobre a linguagem VRML e o formato *wrl* disponíveis em *http://www.w3.org/MarkUp/VRML/* (acesso em 17/Nov/2007).

de edição e apresentação de uma animação a partir de dados em memória, que não necessariamente estivessem gravados em um arquivo, a classe foi estendida. Foi criado um método chamado *previewFacialAnimation*, que recebe como parâmetro um ponteiro para uma estrutura de dados de FAPs.

Ao iniciar a execução da animação, uma *thread* para apresentação do áudio é iniciada em paralelo com a *thread* utilizada para controlar a face. A *thread* responsável pelo controle da face (método *previewFacialAnimation*) sincroniza a FAP com o áudio consultando o relógio da máquina e utilizando a informação da freqüência de quadros da animação (número de quadros por segundo). A cada iteração, verifica-se o tempo decorrido a partir do início da animação e calcula-se o quadro correspondente. Se o quadro for diferente do anterior, ele é buscado na estrutura de FAPs recebida como parâmetro, já como resultado da combinação (Seção 5.5), e entregue para o método *previewFAP*, também da classe *FaceView*. Esse método gera uma chamada para que os FAPs sejam mapeados na malha de pseudo-músculos da ferramenta e a nova expressão seja renderizada na tela. Esses dois últimos métodos não foram alterados, sendo reusados do projeto *Xface*.

5.7 Implementação dos Fenômenos Afetivos e das Expressões Não-Verbais

Adicionalmente, a fala é acompanhada por expressões faciais não-verbais. Essas expressões têm o propósito de enriquecer a animação facial final através de elementos que valorizam a fala da personagem, enfatizando partes importantes de uma sentença, muitas vezes, facilitando seu entendimento. Por exemplo, o movimento das sobrancelhas pode servir para acentuar palavras importantes ou partes das sentenças, para estruturar uma fala ou marcar uma sentença como uma pergunta.

Existem várias abordagens para incorporar expressões faciais não-verbais relacionadas à fala. Por exemplo, sistemas baseados em aprendizagem (Bran99) (LeBB02) são treinados para gerar animações faciais a partir da fala que incluem expressões faciais não-verbais. Uma outra abordagem são os sistemas baseados em *script* (Pear86) (IpCh96) (Karl91). Esses sistemas deixam a sincronização das expressões faciais não-verbais para o usuário. Já os métodos baseados em regras geram expressões não-verbais a partir da análise de conteúdo, da estrutura da pronúncia e do estado do diálogo (Cass94) (PeBS96), a partir da análise do sinal da fala (AlHS02) ou a partir de uma saída intermediária acoplada a um sistema TTS (*Text-to-Speech*) (Albr02).

O *VeeM* propõe uma forma de incorporar as expressões não-verbais e os fenômenos afetivos. Em particular, um estudo da expressão não-verbal relacionada ao movimento dos olhos foi desenvolvido (Rodr07). A movimentação dos olhos leva em consideração, entre outros fatores, a emoção que a personagem está sentindo e se o movimento é aleatório ou atencional (focalizado). Adicionalmente, a expressão não-verbal relacionada à movimentação da cabeça e o fenômeno afetivo referente ao papel da personagem durante a conversação são definidos de forma dependente do movimento dos olhos.

Para validar o estudo, uma ferramenta interativa, anterior à *DynaFeX* foi implementada. A ferramenta permite que o usuário forneça como entrada arquivos descrevendo movimentos dos olhos e da cabeça, emoções e o papel na conversação que a personagem assume ao longo da animação (ouvindo ou falando). A aplicação desse estudo concentra-se principalmente no movimento dos olhos, como já comentado e ilustrado através da interface gráfica da ferramenta na Figura 5.12. Outro aspecto importante a ser mencionado é que essa ferramenta utiliza um modelo 2D para os olhos, extraído do *Expressive Talking Heads* (Luce02) (Rodr03). Manualmente, os resultados foram refletidos em um modelo 3D, mas fica como trabalho futuro aplicar o estudo desenvolvido em (Rodr07) na face MPEG-4 da *DynaFeX*.

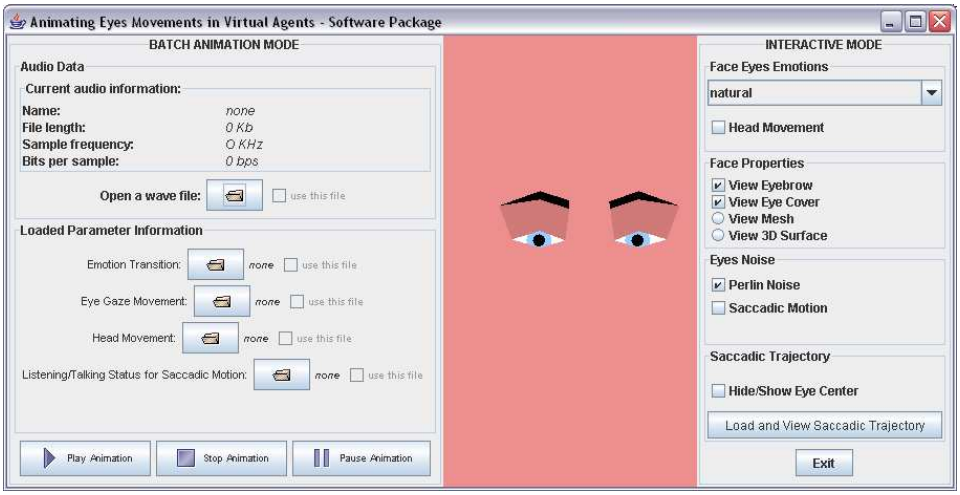

Figura 5.12: Interface gráfica para movimento dos olhos.

Como discutido em (Rodr07), o movimento dos olhos tem um papel fundamental na comunicação verbal e não-verbal entre as pessoas. O movimento da cabeça, em certas situações, é uma conseqüência do movimento dos olhos. No momento que se deseja olhar para algum lugar ou alguma coisa, a cabeça involuntariamente se movimenta.

Uma personagem, em uma conversação, pode assumir dois papéis: estar falando ou estar ouvindo. O papel por ela assumido vai interferir na velocidade

que seus olhos se movimentam, ou seja, na velocidade da pupila, na freqüência do piscar etc.

Por fim, o movimento do olhos pode ser caracterizado por ser direcionado (com foco ou atencional) ou aleatório. Dependendo da emoção, os olhos possuem um movimento específico e uma configuração visual. Em algumas emoções, os olhos estão mais aberto, em outras, mais fechados, por exemplo. A Figura 5.13 ilustra o resultado de algumas configurações dos olhos expressando emoções diferentes em uma personagem bidimensional e, equivalentemente, em uma personagem tridimensional (as emoções na personagem 2D e 3D são equivalentes, da esquerda para direita e de cima para baixo: natural, desconfiada, feliz, surpresa, com raiva e com medo.). Os detalhes desses resultados podem ser encontrados no trabalho (Rodr07).

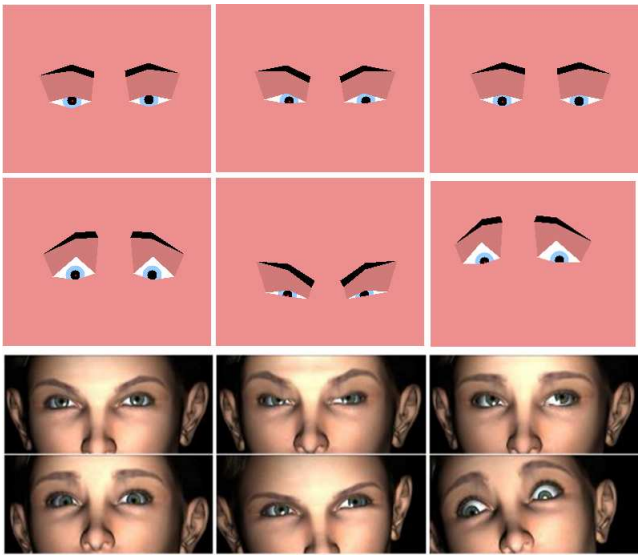

Figura 5.13: Configuração dos olhos para algumas emoções.

Um aspecto importante quando se deseja ter uma animação facial realista (natural e convincente) é a dinâmica das expressões faciais. O ser humano dificilmente fica parado, mesmo quando ele está em silêncio, pensativo ou em uma conversação (tanto no papel de ouvinte como de falante). Como já mencionado ao longo deste documento, os movimentos dos componentes faciais podem estar relacionados à emoção que está sendo vivenciada ou podem ter um corportamento completamente aleatório.

A partir dessas características, o trabalho (Rodr07) também analisou esses dois cenários. O trabalho considera dois tipos de perturbações que podem ocorrer sobre a movimentação da pupila em uma conversação: a existência de um ruído aleatório e a existência de uma mudança "brusca" nesse movimento. A Figura 5.14 retrata esses dois tipos de perturbações, à esquerda tem-se o

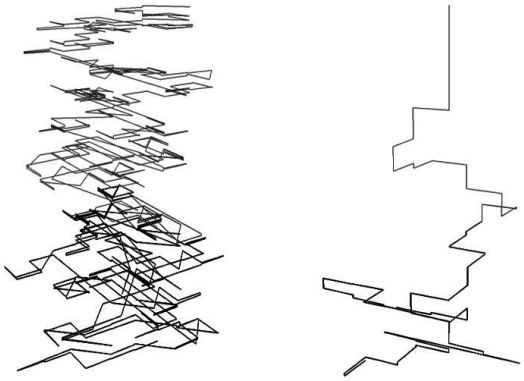

Figura 5.14: Análise comparativa do movimento dos olhos.

movimento da pupila em função do tempo usando o *Perlin noise* (Perl97) e à direita tem-se o movimento da pupila usando o movimento sacádico (LeBB02).

A partir do arquivo que descreve o movimento dos olhos temporalmente durante a animação ("*Eye Gaze Movement*" na Figura 5.12), a pupila pode respeitar apenas o movimento direcionado (atencional ou com foco) descrito pelo arquivo, ou pode atrelar às informações uma das perturbações (ruídos) citadas acima. O *Perlin noise* (Perl97) é um ruído aleatório (contínuo), que quando aplicado aos olhos perturba o movimento da pupila, respeitando apenas a configuração visual da emoção (abertura dos olhos e posicionamento das sobrancelhas, por exemplo).

Já o movimento sacádico caracteriza-se por ser um movimento "brusco" (mudança forte, não-contínua). Respeitando o arquivo que descreve o posicionamento dos olhos, no momento que o movimento sacádico é aplicado há uma mudança rápida da pupila para alcançar a nova posição definida. O estudo do comportamento sacádico dos olhos em (Rodr07) é baseado no modelo estatístico desenvolvido por (LeBB02). Um resultado interessante é obtido quando se combina ambos os movimentos: o movimento sacádico respeitando o comportamento descrito pelo arquivo de entrada e o *Perlin noise* para alguns movimentos como o piscar, proporcionando uma maior naturalidade na animação facial, conforme demonstram os estudos com usuários descritos em (Rodr07).

5.8 Estudos de Casos

Antes do desenvolvimento da ferramenta *DynaFeX*, a pesquisa relacionada a esta tese teve como base de implementação a ferramenta "Expressive Talking Heads" (*ETHs*) (Luce02) (RoFV03). *ETHs* é uma ferramenta de apresentação de animação facial 2D implementada na linguagem de programação

Java [7] e baseada no trabalho de (Perl97). Diferente da *DynaFeX*, o *ETHs* trabalha com síntese de áudio, recebendo com entrada o texto marcado com expressões e especificações de movimentos não-verbais e gerando como saída o áudio sintetizado e a animação das expressões faciais sincronizadas com a fala.

Utilizando o *ETHs*, duas integrações da ferramenta foram feitas com aplicações externas, utilizando as facilidades de uma personagem virtual falante com expressões faciais. Apesar de realizados com outra implementação, os trabalhos exploram vários dos recursos implementados com a *DynaFeX*, podendo em um trabalho futuro as integrações serem implementadas também utilizando a face da *DynaFeX*.

O primeiro estudo de caso consistiu na junção da ferramenta "Expressive Talking Heads" com o formatador "HyperProp" para controle de apresentações hipermídia (RoRS01) (RoSo03). O sistema resultante (Rod04a) (Rod04b) estabelece uma base que pode ser explorada no desenvolvimento de classes importantes de aplicações *Web* e de TV interativa, tais como ensino a distância, jogos, entretenimento, jornalismo, teleconferência etc.

Um aspecto relevante na apresentação de documentos hipermídia são os diferentes tipos de conteúdo que podem estar presentes. Quanto maior a diversidade de tipos de mídia e de formatos, maior o potencial de expressividade para os autores. O suporte aos diferentes formatos está relacionado com as ferramentas de exibição (decodificadores, descompressores, renderizadores etc.) existentes na plataforma de execução dos documentos.

Integrando o *ETHs* ao formatador *HyperProp* como um exibidor de conteúdos textuais passou a ser possível a apresentação desses conteúdos para pessoas portadoras de deficiências visuais, e também a introdução de avatares em documentos multimídia, como teleconferência, aplicações de educação a distância, telemedicina, jogos eletrônicos, telejornalismo, comércio eletrônico etc. A potencialidade de síntese de áudio e de face com emoção foi explorada e o formatador hipermídia trouxe como benefício o suporte à sincronização temporal e espacial do avatar falante com outros conteúdos (objetos de mídia). A Figura 5.15 ilustra a arquitetura de integração das duas ferramentas.

Um dos exemplos desenvolvidos na integração do *ETHs* e do *HyperProp* utilizou a face como apresentadora virtual de um noticiário esportivo, como ilustrado na Figura 5.16.

A adoção do formatador *HyperpProp* como base da implementação do *middleware* do Sistema Brasileiro de TV Digital e a adoção do padrão MPEG-4 como padrão de codificação de vídeo para esse sistema, faz com que a

[7] Informações e tutorais sobre a linguagem Java podem ser encontrados no *site http://java.sun.com/* (acesso em 11/Nov/2007).

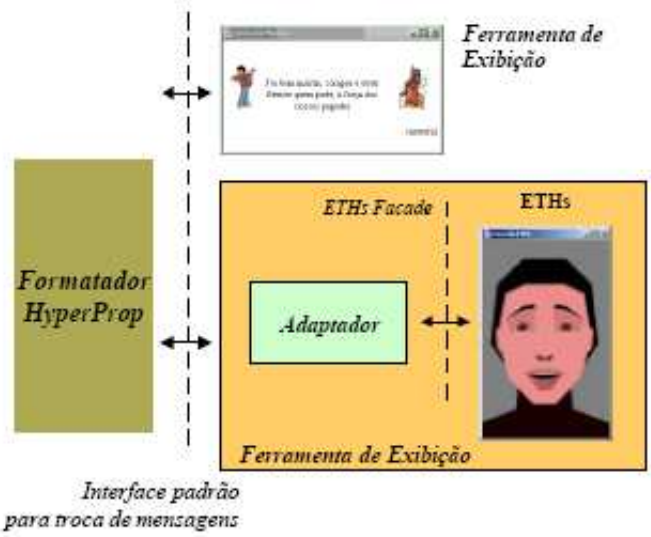

Figura 5.15: Integração das ferramentas ao formatador *HyperProp*.

Figura 5.16: Exemplo de execução.

integração da ferramenta *DynaFeX* com esse ambiente de execução hipermídia possa ser bastante interessante no desenvolvimento de futuras aplicações de entretenimento digital. Nesse sentido, a experiência obtida com a integração com o *ETHs* certamente ajudará nesse possível trabalho futuro.

Na mesma linha de integração da ferramenta *ETHs* com outros sistemas, um segundo estudo de caso foi desenvolvido incorporando o modelo de face 2D com expressão facial e fala sincronizadas a um sistema para geração interativa de histórias, denominado "LOGTELL" (Cial05). Nesse trabalho, a face exportada pelo *ETHs* atua como uma narradora das histórias criadas pelo usuário (Rodr06).

O *LOGTELL* (Cial05) é baseado em modelagem e simulação. O sistema procura capturar a lógica de um gênero de histórias através de um modelo de lógica temporal, sendo capaz de guiar a geração de enredos de histórias através da simulação combinada com a intervenção (interação) do usuário. O

LOGTELL não foca apenas nas diferentes formas de se contar uma história, mas também na criação dinâmica dos enredos. O modelo é composto por eventos típicos e regras de inferência de objetivos.

O trabalho (Rodr06) descreve a arquitetura que incorpora um narrador virtual com expressões emocionais sincronizadas com a fala. A Figura 5.17 ilustra a arquitetura criada na junção do *ETHs* e o *LOGTELL*.

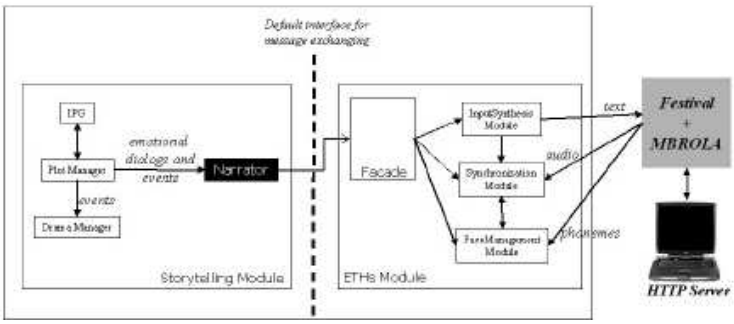

Figura 5.17: Arquitetura da junção do *ETHs* com o *LOGTELL*.

O ambiente integrado do *ETHs* com o *LOGTELL* tem a capacidade de construir, apresentar e narrar diferentes histórias a partir de gêneros variados. O exemplo trabalhado em (Rodr06) é baseado no gênero "espadas e dragões" (*Swords-and-Dragons context*), onde heróis, vítimas e vilões interagem em um cenário 3D ocupado por castelos e igrejas. O narrador é capaz de contar, usando uma perspectiva em terceira pessoa e com a emoção apropriada, cada cena da história, como ilustra a Figura 5.18. A história pode ser modificada em qualquer momento através da intervenção do usuário. Uma utilização interessante desse estudo de caso que pode ser investigada em um outro trabalho futuro é em um jogo participativo (*participatory game*, onde o jogador é co-autor do enredo (*plot*)).

Como pode ser observado, existe um grande conjunto de aplicações que podem ser enriquecidos com o uso de faces construídas através de sistemas com o *ETHs* e a ferramenta *DynaFeX*.

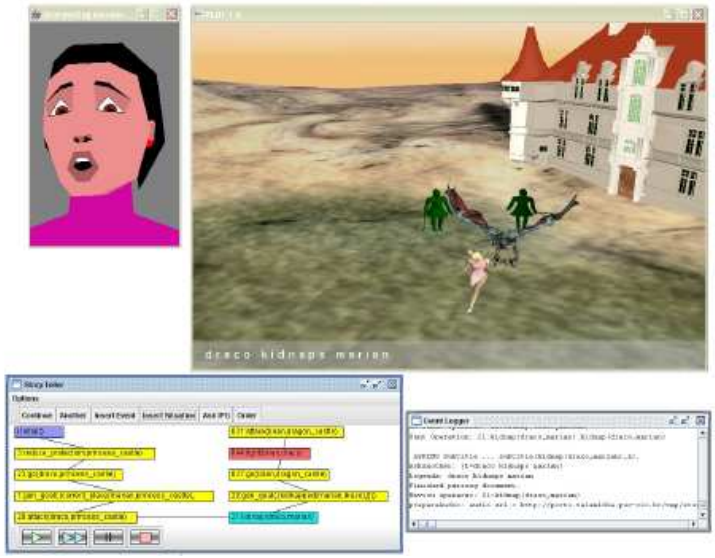

Figura 5.18: Cena renderizada no ambiente de narração.