

UNIVERSIDADE FEDERAL FLUMINENSE

MÔNICA DE LOURDES SOUZA BATISTA

Simulação de emoções em faces humanas utilizando os
algoritmos de *bump mapping* e *morphing* implementados em
GPU

NITERÓI-RJ
2011

UNIVERSIDADE FEDERAL FLUMINENSE

MÔNICA DE LOURDES SOUZA BATISTA

Simulação de emoções em faces humanas utilizando os
algoritmos de *bump mapping* e *morphing* implementados em
GPU

Dissertação de Mestrado submetida ao
Programa de Pós-Graduação em
Computação da Universidade Federal
Fluminense como requisito parcial para a
obtenção do título de Mestre. Área de
concentração: Computação Visual e
Interfaces.

Orientador:
Prof. Esteban Walter Gonzalez Clua, D.Sc.

NITERÓI-RJ
2011

Simulação de emoções em faces humanas utilizando os algoritmos de *bump mapping* e *morphing* implementados em GPU

Mônica de Lourdes Souza Batista

Dissertação de Mestrado submetida ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do título de Mestre. Área de concentração: Computação Visual e Interfaces.

Aprovada por:

Prof. Esteban Walter Gonzalez Clua, IC-UFF
Presidente

Prof. Anselmo Antunes Montenegro, IC-UFF

Prof. Ricardo Farias, COPPE-UFRJ

Niterói, 22 de Março de 2011.

*Aos meus pais Batista e Fátima,
e ao meu irmão Thiago.*

Agradecimentos

Primeiramente, agradeço a Deus, pois Ele é o maior responsável por todas as minhas conquistas.

Agradeço aos meus pais, Batista e Fátima, e ao meu irmão, Thiago, pelo apoio que sempre me deram. Por me incentivarem e acreditarem em mim. Agradeço também pela força que eles me deram durante todos os anos de minha vida.

Agradeço também a minha amiga Vitória, que me acompanha desde os oito anos de idade. Obrigada pela força e por torcer por mim.

Ao meu orientador Esteban Clua que sempre me ajudou e sempre esteve do meu lado. Pela paciência que teve comigo durante o mestrado, pelo apoio e por acreditar em mim.

Ao Anselmo Montenegro, que foi mais que um professor, é um grande amigo. Não tenho palavras para agradecer tudo o que ele fez por mim no mestrado.

Agradeço a professora Aura Conci por ter me mostrado o quanto sou capaz e por ter me inspirado cada vez mais na concretização de meus objetivos.

Não poderia me esquecer de agradecer também as professoras Simone Martins e Ana Cristina Bicharra. Elas me apoiaram muito.

Ao Martinhon por ter estudado comigo horas e horas por dia, por tirar minhas dúvidas e pela paciência que teve comigo quando eu não conseguia entender aquelas contas.

Não podia deixar de agradecer ao Sérgio Muinhos que me acompanha desde o segundo período da graduação. Orientou-me em projeto de iniciação científica e na monografia. É um grande amigo e que está sempre comigo em todos os momentos da minha vida.

Agradeço a Teresa e a Ângela pela amizade e por confiarem em mim.

Ao artista Fernando Ribeiro que fez todas as imagens e mapas de normal que precisei utilizar na minha dissertação.

Aos meus grandes amigos Marcelo Zamith, Leonardo Motta, Roger Resmini, Thales Sabino, Cintia Caetano, Pedro Martins, Victor Fernández e Carlos Henrique (o Carlão). Agradeço pelos vários momentos em que passamos juntos e pelos estudos em grupo.

Agradeço a Aline Ramires e a Amanda Ramos. Elas foram minhas companheiras de apartamento quando morei em Niterói. Grandes amigas que me acompanharam no decorrer do mestrado. Obrigada por todos os momentos felizes que passamos juntas e por cuidarem de mim.

Resumo

A percepção humana está acostumada a captar emoções e expressões em todo relacionamento com pessoas. Assim, para criar personagens virtuais convincentes e realistas, a computação gráfica e os jogos precisam elaborar métodos capazes de gerar simulações que arremetem a percepção do espectador ou jogador com semelhante percepção. Entretanto, enquanto gerar animações de corpos rígidos e até mesmo de articulações sejam problemas computacionais relativamente triviais de serem resolvidos em tempo real, a animação facial oferece uma série de desafios.

Neste trabalho é feita uma abordagem para reforçar a percepção da emoção de personagens mediante o tratamento e animação das rugas de uma face. Para tanto, lança-se mão de técnicas de *bump mapping* para a representação das rugas, juntamente com técnicas de *warping* e *morphing* para a animação das mesmas.

Tendo como objetivo a inserção destas animações em jogos, este trabalho propõe e implementa um processo totalmente baseado nas GPUs.

O trabalho também apresenta um estudo e uma revisão bibliográfica sobre o tema de animação facial, de forma a estabelecer um *framework* e roteiro de expressões guias para situações genéricas.

Palavras-chave: *Bump mapping*, *morphing*, emoções, expressões faciais, rugas.

Abstract

Human perception is used to capture emotions and expressions in all relationships with people. Thus, to create compelling and realistic virtual characters, computer graphics and games must devise methods to generate simulations that rush to the perception of the viewer or player with a similar perception. However, while generating animations of rigid bodies and joints are even relatively trivial computational problems to be solved in real-time facial animation offers a number of challenges.

In this work, an approach to enhance the perception of emotion through the characters and animation treatment of wrinkles on a face. To do so, we applied the technique of bump mapping for the representation of wrinkles, along with techniques of warping and morphing to animate them.

Aiming to insert these animations in games, this work proposes and implements a process based entirely on GPUs.

The work also presents a study and a literature review on the subject of facial animation, in order to establish a framework and roadmap of guiding expressions for generic situations.

Keywords: Bump mapping, morphing, emotions, facial expressions, wrinkles

Sumário

1. Introdução.....	14
1.1 Motivação.....	15
1.2 Problema.....	15
1.3 Contribuição desta dissertação.....	16
2. Emoções.....	18
2.1 Introdução.....	18
2.2 Emoção.....	18
2.2.1 Teoria de emoções.....	19
2.3 Emoções básicas.....	20
2.4 Anatomia facial.....	27
2.4.1 Ossos.....	27
2.4.2 Músculos.....	28
2.4.2.1 Músculos ao redor do globo ocular.....	29
2.4.2.2 Músculos do nariz.....	30
2.4.2.3 Músculos da boca.....	30
2.5 Animação facial.....	32
2.5.1 História da animação facial.....	32
2.5.2 Modelagem e animação facial.....	34
3. Morphing.....	37
3.1 Introdução.....	37
3.2 Estágios do <i>morphing</i>	38
3.2.1 Especificação.....	38
a) Especificação por partição.....	38
b) Especificação por características.....	39
3.2.2 Deformação.....	39
a) Deformação por malha.....	40
b) Deformação por malhas de triângulos.....	42
c) Deformação por campo.....	42
d) Deformação por funções de base radial.....	44
3.2.3 Transformação de tonalidades.....	45
3.2.4 Controle de transição.....	47
4. Deformação da superfície para rugas.....	48
4.1 Introdução.....	48
4.2 O <i>bump mapping</i>	48
4.3 Métodos para armazenamento dos modelos de perturbação da normal.....	50
4.4 Sistemas de coordenadas.....	51
5. Morphing paralelo na GPU.....	53
5.1 CUDA.....	53
5.2 <i>Morphing</i> paralelo de texturas 2D em tempo real.....	54
5.2.1 Estrutura de dados e estratégia de alocação.....	55
5.2.2 Estrutura processual.....	55
5.3 Etapas para a execução do <i>kernel</i>	56

5.3.1 Alocação de memória na GPU.....	56
5.3.2 Transferência de dados da CPU para a GPU.....	56
5.3.3 Execução do <i>kernel</i>	57
5.3.4 Desalocação de memória.....	58
5.4 Passagem do resultado do contexto CUDA para o contexto gráfico.....	58
5.5 <i>Bump mapping</i> na GPU.....	59
5.5.1 Os <i>shaders</i>	60
6. Resultados.....	63
6.1 Modelagem.....	63
6.2 Animação facial.....	64
6.3 Máquina utilizada.....	65
6.4 Resultados.....	66
7. Conclusão e trabalhos futuros.....	68
8. Referências bibliográficas.....	70

Lista de Figuras

Figura 1.1: Algumas expressões faciais no jogo <i>Heavy Rain</i>	15
Figura 2.1: Expressões básicas: (a) tristeza, (b) raiva, (c) alegria, (d) medo, (e) desgosto e (f) surpresa.....	21
Figura 2.2: Expressão de tristeza.....	22
Figura 2.3: Expressão de raiva.....	23
Figura 2.4: Expressão de alegria.....	24
Figura 2.5: Expressão de medo.....	25
Figura 2.6: Expressão de desgosto.....	26
Figura 2.7: Expressão de surpresa.....	26
Figura 2.8 – Vista frontal e lateral do esqueleto da cabeça.....	28
Figura 2.9 – Músculos da expressão facial, vista frontal.....	29
Figura 2.10: Músculos da boca.....	31
Figura 2.11: Animação utilizando interpolação linear.....	33
Figura 2.12: Detalhes da malha facial.....	36
Figura 3.1: Carro se transformando em tigre.....	37
Figura 3.2: Especificação por partição.....	39
Figura 3.3: Deformação por malha.....	41
Figura 3.4: Malha não deformada e malha deformada, respectivamente.....	42
Figura 3.5: Vetores iniciais e vetores finais.....	43
Figura 3.6: Interpolação das tonalidades das imagens.....	46
Figura 3.7: Interpolação de cores (Cross-Dissolve).....	46
Figura 4.1: Resultado da aplicação de uma perturbação em uma superfície.....	49

Figura 4.2a: <i>Bump mapping em um personagem</i>	49
Figura 4.2b: <i>Bump mapping em uma parede de pedras</i>	50
Figura 4.3: Coordenadas x, y e z contendo informações da direção do vetor normal da superfície.....	51
Figura 4.4: Matriz utilizada para a transformação.....	52
Figura 5.1: Modelo de execução do CUDA.....	54
Figura 5.2: Diagrama da passagem do resultado do contexto CUDA para o contexto gráfico.....	59
Figura 6.1: Malha 3D criada pelo artista Fernando.....	63
Figura 6.2: Mapa de normal da expressão de bravo.....	64
Figura 6.3: Mapa de normal da expressão de assustado.....	65
Figura 6.4: Resultado da simulação para cada taxa (t) de transição.....	66

Lista de Tabelas

Tabela 6.1: Tempo gasto em cada etapa do <i>kernel</i> e o tempo gasto com o <i>bump mapping</i> na máquina 1.....	67
Tabela 6.2: Tempo gasto em cada etapa do <i>kernel</i> e o tempo gasto com o <i>bump mapping</i> na máquina 2	67
Tabela 6.3: Tempo gasto em cada etapa do <i>kernel</i> e o tempo gasto com o <i>bump mapping</i> na máquina 3	67

Capítulo 1

Introdução

A simulação de expressões faciais em computação gráfica é algo fundamental para manter o realismo em animações e ambientes interativos. Enquanto grandes avanços foram feitos e conquistados nos modelos de iluminação, permitindo que imagens realistas pudessem ser sintetizadas em tempo real, há ainda muito a ser desenvolvido no que se refere ao movimento e animação de personagens. Em se tratando de movimentos faciais, há vários fatores que agravam mais ainda o problema, devido à complexidade intrínseca dos músculos, sua elasticidade e o grande número de combinações que pode haver.

Com o incremento de poder de processamento gráfico e o aumento de número de polígonos capazes de serem tratados, é possível representar personagens mais detalhados. Entretanto, apenas seu aspecto geométrico não é suficiente para o aumento correspondente de seu realismo. Movimentos naturais tornam-se absolutamente fundamentais para que haja um realismo completo da representação e da credibilidade.

Esta dissertação apresenta uma proposta de animação facial desenvolvida para simular expressões faciais em tempo real, usando uma combinação da técnica de *warping* e *bump mapping*.

A presente dissertação de mestrado é uma extensão do trabalho proposto por Zamith et al (ZAMITH et al, 2009), onde foi apresentada uma abordagem paralela em GPU do algoritmo de *morphing*.

O método de *warping* é usado para a etapa de animação, criando interpolações de imagens que correspondem às expressões faciais básicas. Neste trabalho utiliza-se uma estratégia de *warping* em GPU, primeiramente proposta por Zamith et al (ZAMITH et al, 2009), possibilitando que sejam criadas imagens interpoladas em taxas interativas. O *bump mapping*, introduzido por Blinn em 1978, é utilizado num estágio seguinte para aplicar os resultados interpolados sobre uma malha que representa uma face. Esta etapa também é feita na GPU, porém dentro de um contexto gráfico. Neste trabalho também é proposta uma estratégia para que todo o processo se mantenha inteiramente na GPU, evitando que haja cópia de dados entre CPU e GPU.

1.1 Motivação

A animação facial é uma área da computação gráfica que tem como objetivo simular, de uma maneira convincente, como a face humana se comporta. Atualmente a animação facial abrange uma grande área de aplicações, dentre elas, jogos, filmes, televisão e medicina. Este problema não é trivial dentro da computação gráfica, pois envolve muitas restrições, sendo necessário adotar um conjunto de técnicas e algoritmos complexos (ROSÁRIO, 2008). Desta forma, há ainda muita pesquisa e diversos problemas a serem resolvidos nessa área (RODRIGUES, 2007). Este trabalho se situa dentro deste contexto, propondo uma estratégia para tornar o processo viável em tempo real e ao mesmo tempo tornar o processo de animação o mais transparente possível para o artista.

A Figura 1.1, com imagens extraídas do jogo *Heavy Rain*, ilustra a importância do tema, mostrando como expressões faciais podem influenciar na criação de personagens convincentes (MORETI, 2006).



Figura 1.1: Algumas expressões faciais no jogo *Heavy Rain*

Fonte: (HEAVYRAINPS3, 2010).

1.2 O problema

Um dos principais desafios encontrados nos jogos eletrônicos é que as cenas devem ser sintetizadas em tempo real, ao contrário dos filmes que podem levar muito tempo.

Produzir expressões faciais virtuais com o maior nível de realismo possível e com o menor número de polígonos é um dos principais desafios da área de jogos. O constante aumento do poder de processamento das Unidades de Processamento Gráfico (GPU) permite criar faces virtuais mais refinadas e detalhadas, mas é importante ao mesmo tempo desenvolver métodos que possibilitem a animação ser eficiente e igualmente rápida.

As rugas das faces, bem como seus movimentos, exercem um papel importante na animação de faces e na sua representação convincente. Enquanto a inserção de rugas foi bastante explorada mediante a técnica de *bump mapping*, poucos trabalhos foram propostos com o objetivo de animá-las, uma vez que é necessário um volume grande de processamento para todas as imagens. Este trabalho propõe uma abordagem focada na animação de rugas e marcas de pele dos personagens, reforçando outros processos de animação facial, que também são necessários de serem usados.

1.3 Contribuição desta dissertação

Essa dissertação tem como principal objetivo o desenvolvimento de um sistema de animação facial para a simulação de emoções em tempo real. Neste trabalho foca-se a animação e manipulação de rugas de cada expressão facial, sendo estas modeladas pelos mapas de normal, aplicadas posteriormente numa malha utilizando o *bump mapping*. A animação do movimento das rugas é feito através do algoritmo de *morphing*, tomando como base imagens que contém expressões chaves e básicas.

Assim sendo, pode-se dizer que as principais contribuições do trabalho são:

- Realizar um levantamento bibliográfico detalhado sobre o tema, listando diversas estratégias pertinentes e que podem vir a ser combinadas com o modelo proposto;
- Apresentar uma estratégia que utiliza rugas para reforçar as expressões faciais tradicionais;
- Utilizar o método de *morphing* baseado em GPUs para gerar imagens interpoladas em tempo real;
- Apresentar uma estratégia para otimizar a troca de contexto CUDA para contexto gráfico, minimizando o tráfego de dados entre GPU e CPU.
- Apresentar uma proposta de imagens de rugas primitivas para gerar diversas combinações de situações de emoção.

A dissertação apresenta os seguintes capítulos: além desta introdução, é apresentado no capítulo 2 um estudo sobre emoções, as seis emoções básicas, anatomia facial e animação facial. No capítulo 3, é apresentado o conceito de *morphing* e suas fases. No capítulo 4, são mostrados conceitos de *bump mapping* e a perturbação da normal. No capítulo 5, é apresentado o algoritmo proposto de *morphing* paralelo de texturas 2D em tempo real, a estrutura de dados e estratégia de alocação, a estrutura processual, o *kernel*, as etapas do *kernel*, a estratégia de passagem do resultado do Cuda para o *shader* e o *bump mapping* na GPU. O capítulo 6 apresenta os resultados alcançados, a modelagem e a animação facial. Finalmente, o capítulo 7 apresenta a conclusão e os trabalhos futuros.

Capítulo 2

Emoções

2.1 Introdução

A interação entre pessoas é uma atividade que faz parte do cotidiano de todos. Olhar para o rosto de uma pessoa é uma tarefa comum, quando se realiza esta comunicação. Assim sendo, o ser humano possui uma importante habilidade para reconhecer faces e entender as emoções que elas transmitem (MARIN, 2006).

Assim como a expressão oral, a expressão facial também permite que os seres humanos partilhem suas emoções, pois ela é capaz de transmitir sentimentos que acompanham e complementam as palavras.

Neste capítulo será apresentado um resumo sobre conceitos de emoções, anatomia da face e técnicas de modelagem e animação facial.

2.2 Emoção

A Computação Afetiva é a área da computação que estuda os aspectos que influenciam as emoções. Dentro desta área existe uma linha de pesquisa que trabalha com simulação de emoções (FROZZA, 2009).

As expressões faciais despertam a atenção de psicólogos e cientistas há muitas décadas. Desde o século XIX, pesquisadores de diversas áreas, como psicólogos, filósofos e neurocientistas tentam definir o que é a emoção (ZAGALO, 2004), não havendo um consenso.

Segundo Rodrigues (2007), duas perguntas se fazem presentes entre as diversas teorias que se referem a emoção:

- Por que as pessoas em uma situação particular experimentam uma determinada emoção e não outra?
- Por que diferentes pessoas não experimentam a mesma emoção quando vivenciam a mesma situação?

Para Damásio (1996), emoção significa tornar um sentimento exterior. Quando uma pessoa se emociona, ela externaliza a emoção através do corpo e das expressões faciais, enquanto o sentimento é algo que fica no interior da pessoa.

As emoções podem ser expressas por seis maneiras: reações expressivas (sorrisos, dentes trincados, testa franzida); reações fisiológicas (batimentos cardíacos acelerados, lágrimas, calores e rosto vermelho); comportamentos instrumentais (correr, juntar as mãos); comportamentos instrumentais situacionais (digitar uma tecla com força); cognições (pensamento de injustiça, sentir impotência diante de problemas); e finalmente, sentimentos que integram fenômenos fisiológicos e cognitivos (sentir um aperto no peito, lágrimas nos olhos e a lembrança do evento que gerou tais sensações) (FROZZA, 2009).

2.2.1 Teoria de emoções

Diferentes modelos foram desenvolvidos pelos pesquisadores para explicar o conceito de emoção.

O primeiro modelo de emoção foi proposto por Platão por volta de 430 a.c. Segundo Platão, a alma é composta por cognição, emoção e motivação. A emoção é algo que confunde e que retira a razão humana (RODRIGUES, 2007).

Aristóteles (384-322 a.c.) questionou a divisão que Platão fez da alma e sugeriu a interação desses componentes. Aristóteles propôs que as emoções são positivas e interessantes e que elas surgem a partir de experiências que já foram vividas pelo ser humano (JAQUES e VICARI, 2005).

Por volta de 1600, Descartes insistiu na visão dualista que separa a mente do corpo e do cérebro. Seu modelo consiste em uma abordagem cognitiva para a emoção onde é estabelecida a idéia de dualidade, afirmando que o corpo psicológico e a mente possuem diferentes funções (RODRIGUES, 2007).

Em 1872, Darwin realizou vários estudos da expressão de emoções na face, corpo e voz. Em suas pesquisas Darwin verificou que fenômenos emocionais, principalmente expressão, podem ser encontrados em diferentes culturas. Darwin e Ekman (EKMAN, 1999) concluíram que existe um conjunto de expressões faciais que são universais, ou seja, que representam a mesma emoção em diferentes culturas. Este estudo deu origem ao surgimento da teoria das emoções básicas. Alguns teóricos que seguem esse modelo acreditam na existência de um conjunto de emoções básicas tais como raiva, alegria e tristeza (JAQUES e VICARI, 2005).

Willian James, psicólogo e filósofo norte-americano, baseou-se na abordagem fisiológica. Ele propôs que o corpo muda e esta mudança leva ao sentimento de emoção. Este modelo faz com que pesquisas e teorias levem em consideração a importância de expressões faciais e corporais em uma emoção (JAQUES e VICARI, 2005).

Atualmente, Paul Ekman é um dos principais pesquisadores que estudam a área de emoção. Ekman provou, através de realização de diversas pesquisas, que existem seis emoções básicas, conhecidas como emoções puras que são a tristeza, raiva, alegria, medo, desgosto e surpresa (EKMAN, 1999; RODRIGUES, 2007).

A psicologia das emoções propõe que as emoções são resultado de uma avaliação cognitiva (*appraisal*) de situações e eventos antecedentes. Os psicólogos cognitivistas estão interessados no *appraisal* e nos eventos que disparam as emoções. Esta abordagem, conhecida como abordagem cognitiva das emoções, influencia vários trabalhos em computação afetiva. Ortony, Clore e Collins criaram uma teoria baseada na abordagem cognitiva das emoções. Conhecida como modelo OCC, esta teoria explica a origem das emoções e descreve os processos cognitivos que ativam cada uma delas (JAQUES e VICARI, 2005).

2.3 Emoções básicas

Atualmente, pesquisadores realizam diversos estudos e pesquisas sobre as expressões faciais. Estudiosos da área estão dando uma atenção à universalização das expressões faciais através das diferentes culturas (RODRIGUES, 2005).

Pesquisadores e psicólogos concordam na teoria de Ekman, e afirmam que existem seis categorias universais de expressões faciais, dentre elas: tristeza, raiva, alegria, medo, desgosto e surpresa. Dentro de cada uma dessas categorias pode existir uma variedade de “intensidades” das expressões faciais e algumas variações nos seus detalhes. Cada uma dessas categorias possui características, principalmente nos componentes faciais olhos e boca, e nos locais onde as rugas se formam, conforme ilustrado na Figura 2.1 (RODRIGUES, 2005).

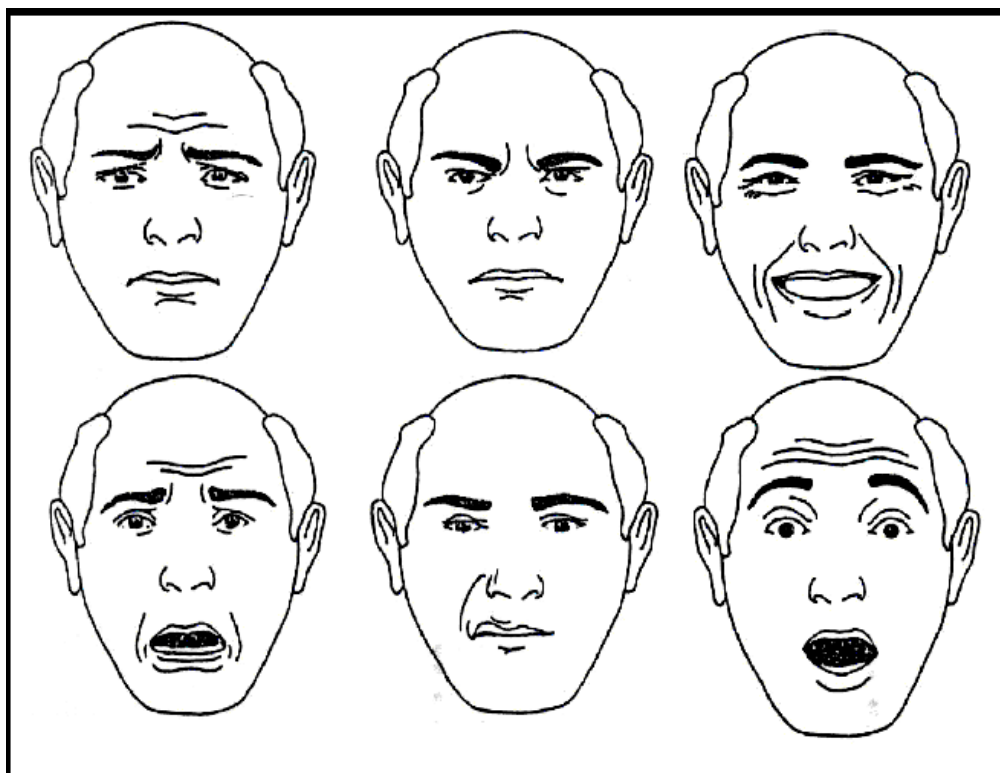


Figura 2.1: Expressões básicas: (a) tristeza, (b) raiva, (c) alegria, (d) medo, (e) desgosto e (f) surpresa.

Fonte: (RODRIGUES, 2005)

Alguns pesquisadores concluíram que existem também as emoções secundárias. Essas emoções são aprendidas em sociedade e estão relacionadas com a vida do homem enquanto ser social. As emoções secundárias ou sociais são: vergonha, ciúme, culpa e orgulho (LESSA, 2008).

Segundo Adamatti (2003), muitos pesquisadores afirmam que não existe um conjunto de emoções básicas porque muitas emoções podem ser distintas e igualmente básicas. Para esses pesquisadores existem algumas perguntas que precisam ser respondidas:

- Como sustentar que existam emoções básicas que possam ser universais?
- Estas emoções consideradas básicas formam emoções complexas, ou mistura de emoções?
- Emoções não básicas podem ter sido desenvolvidas antes das emoções básicas?

Neste trabalho utilizaremos a teoria de Ekman para a construção de todas as emoções. A seguir, lista-se e descrevem-se características destas emoções básicas.

Tristeza

O sentimento de tristeza é o oposto do sentimento de alegria. Geralmente ele está associado com depressão, sofrimento ou dor (RODRIGUES, 2007).

Na expressão facial de tristeza simples, as partes internas das sobrancelhas são movidas em direção a região superior da face, ficando elevadas. A pele e o tecido macio abaixo das sobrancelhas são puxados para cima da pálpebra superior. Os olhos são levemente fechados por causa de uma pressão para baixo do tecido sobre a pálpebra e também por causa do movimento para cima da pálpebra inferior. Na tristeza simples, a boca fica relaxada.

As rugas associadas à tristeza incluem dobras horizontais na testa entre as sobrancelhas, linhas verticais são traçadas entre as sobrancelhas, dobras oblíquas surgem sobre as pálpebras superiores e uma dobra no formato de sorriso surge baixo do lábio inferior, conforme ilustrado na Figura 2.2.

A tristeza tem várias intensidade e variações, incluindo o choro de boca aberta, o choro de boca fechada, tristeza oprimida, quase choro e tristeza desesperadora. Essas variações podem incluir sobrancelhas totalmente abaixadas, olhos firmemente fechados, boca aberta no formato de quadrado e saliência no queixo.



Figura 2.2: Expressão de tristeza

Fonte: (BAPTISTA, 2005)

Raiva

Emoção relacionada ao ódio, à ira e ao rancor. Este sentimento toma conta do ser humano quando ele é contrariado por algo ou alguém (RODRIGUES, 2007).

Na expressão facial de raiva simples, os cantos internos das sobrancelhas são puxados para baixo e aproximados um do outro. O limite inferior da sobrancelha fica no

mesmo nível que a pálpebra superior. O olho fica amplamente aberto, mas a pressão da fronte rebaixada faz com que o branco dos olhos não seja mostrado sobre a íris.

A boca é fechada com o lábio superior levemente comprimido ou em formato de trapézio. As rugas para a raiva incluem dobras horizontais sobre as pálpebras superiores e linhas verticais entre as sobrancelhas. As rugas e as demais características da expressão facial de raiva podem ser observadas na Figura 2.3.

Variações de raiva incluem a raiva com gritos e austeridade. Essas variações podem incluir lábios fortemente comprimidos com uma saliência no queixo ou a boca aberta com o lábio superior dando ar de desdém e o lábio inferior reto, mostrando tanto a dentição superior quanto a inferior.



Figura 2.3: Expressão de raiva

Fonte: (BAPTISTA, 2005)

Alegria

Esta emoção é caracterizada por sentimentos de agrado, gratidão e satisfação. Estão associados com alegria o bem-estar, encanto, paz interior, saúde, segurança e contentamento (RODRIGUES, 2007).

Na expressão facial de alegria simples, as sobrancelhas ficam relaxadas. A pálpebra superior é baixada levemente e a pálpebra inferior é empurrada para cima pela bochecha. Os olhos ficam brilhantes e a boca fica alongada com os cantos puxados na direção das orelhas.

Se a boca estiver fechada, os lábios ficam finos e fortemente pressionados contra o maxilar ou mandíbula. Se a boca estiver aberta, o lábio superior fica reto, mostrando a dentição superior, o lábio inferior fica horizontalizado no meio e forma um ângulo ascendente perto dos cantos.

Na expressão de alegria, formam-se rugas, conhecidas como “pés de galinha” nos cantos dos olhos, uma dobra embaixo da pálpebra inferior, covinhas e uma profunda dobra nasolabial do nariz ao queixo é formada, conforme ilustrado na Figura 2.4.

Variações na expressão de alegria incluem gargalhada exagerada, gargalhada, sorriso de boca aberta, sorriso, sorriso abafado, sorriso melancólico, sorriso ávido, sorriso de aceitação, sorriso malicioso, sorriso debochado, sorriso de olhos fechados, sorriso falso e gargalhada falsa.

Sorriso falso e gargalhada falsa são indicados pela diminuição dos “pés de galinha” nos cantos dos olhos e pela presença de leves dobras abaixo das pálpebras inferiores ou pela ausência das mesmas.



Figura 2.4: Expressão de alegria

Fonte: (BAPTISTA, 2005)

Medo

O sentimento de medo corresponde a uma grande inquietação diante de um perigo real ou imaginário, de uma ameaça, susto, pavor, temor, terror. O medo é um mecanismo de sobrevivência e geralmente ocorre como resposta a um estímulo negativo específico (RODRIGUES, 2007).

Na expressão facial de medo, as sobrancelhas são elevadas e puxadas para perto uma da outra e as partes internas são presas para cima. Os olhos ficam tensos e em alerta. A boca pode ficar levemente aberta e esticada para os lados. As rugas associadas ao medo incluem dobras horizontais acima das sobrancelhas, linhas verticais entre as sobrancelhas, covinhas sobre as sobrancelhas e dobras oblíquas sobre as pálpebras superiores (RODRIGUES, 2007).

Na sensação de temor ou preocupação, os lábios são apertados juntos firmemente, sendo que as margens dos lábios desaparecem. Aparece uma saliência abaixo do lábio inferior e acima do queixo.

Na sensação de terror, os olhos e a boca ficam amplamente abertos. O lábio superior fica relaxado enquanto o lábio inferior fica esticado, expondo de forma estreita a dentição inferior. Dobras no formato de traços aparecem dos lados do lábio inferior.

A Figura 2.5 ilustra as regiões onde as rugas são formadas e as demais características, descritas acima, para a expressão facial de medo.



Figura 2.5: Expressão facial de medo

Fonte: (BAPTISTA, 2005)

Desgosto

Esta emoção associa-se com coisas que são percebidas como sujas ou intragáveis. É um sentimento de incompatibilidade, aversão e nojo (RODRIGUES, 2007).

No desgosto, as sobrancelhas ficam relaxadas. As pálpebras ficam relaxadas e levemente fechadas. A pálpebra superior é elevada gerando um ar de desdém, normalmente assimétrico. A dobra nasolabial fica aprofundada ao longo do nariz assimetricamente.

Os olhos ficam quase totalmente fechados na maior parte das vezes. O lábio superior é elevado podendo mostrar a dentição superior. O lábio inferior é levemente empurrado para cima.

Surgem linhas verticais entre as sobrancelhas, “pés de galinha” e dobras na pálpebra inferior. Surgem também rugas nos cantos internos dos olhos através da ponte do nariz, além de uma saliência no queixo, conforme visto na Figura 2.6.



Figura 2.6: Expressão de repúdio

Fonte: (BAPTISTA, 2005)

Surpresa

Esta emoção está relacionada com algo que surpreende, como por exemplo, um acontecimento imprevisto ou um prazer inesperado (RODRIGUES, 2007).

Na expressão de surpresa as sobrancelhas são elevadas o mais alto possível. As pálpebras superiores ficam abertas o mais amplamente possível e as pálpebras inferiores ficam relaxadas. A boca fica aberta sem tensão muscular em um formato oval. Na testa rugas horizontais são formadas se estendendo ao longo de toda a sobrancelha, conforme mostrado na Figura 2.7.



Figura 2.7: Expressão de surpresa

Fonte: (BAPTISTA, 2005)

2.4 Anatomia facial

A face humana possui uma complexa estrutura. Apesar dos seres humanos possuírem características muito semelhantes, existem alguns atributos e algumas particularidades que os diferenciam, como por exemplo, texturas, cores, linhas de expressão, dentre outros (LUCENA, 2002).

Para modelar uma face com realismo é necessário conhecer a anatomia facial que é composta por ossos, cartilagens, músculos, nervos, vasos sanguíneos, glândulas, tecidos e pele (LUCENA, 2002).

2.4.1 Ossos

A anatomia da cabeça é dividida em dois planos imaginários, o mediano e o coronal. O mediano é o plano vertical que divide o corpo em duas partes iguais: esquerda e direita (MARTINI, 2006).

Já o coronal é um plano vertical e perpendicular ao mediano, que divide o corpo humano em uma parte anterior e outra parte posterior (BAPTISTA, 2005).

Duas partes formam a estrutura óssea da cabeça: o crânio e o esqueleto facial. O crânio armazena e protege o cérebro. O esqueleto facial contém grande número de ossos e se divide em: superior, intermediária e inferior. O esqueleto facial superior é formado pelas cavidades orbitais, pelo osso frontal e pelos ossos nasais. O esqueleto facial intermediário é formado pela cavidade nasal e pelo maxilar. Finalmente, o esqueleto facial inferior corresponde à região da mandíbula (MARTINI, 2006). A Figura 2.8 ilustra a vista frontal e lateral do esqueleto da cabeça.

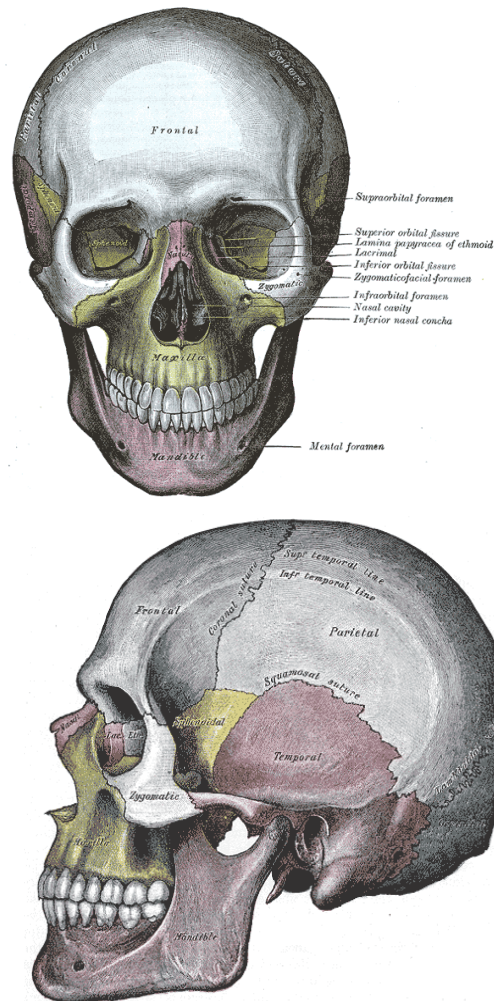


Figura 2.8 – Vista frontal e lateral do esqueleto da cabeça

Fonte: (MARTINI, 2006)

2.4.2 Músculos

Os músculos da face são responsáveis pela alimentação, mastigação, fonação e piscar de olhos. Estes músculos, em uma extremidade, estão situados abaixo da pele e constituem uma camada quase única, onde os feixes de fibras de um músculo são unidos aos de outros e nos locais de inserção é comum estarem entrelaçados. Na outra extremidade estão ligados aos ossos (MARTINI, 2006).

São os músculos mais delicados e mais fracos do corpo humano e a sua contração faz com que a área da pele à qual estão fixados se movimente, produzindo assim,

depressões em forma de linhas perpendiculares à direção das fibras dos músculos. Essas depressões, com passar do tempo, se transformam em rugas (MARTINI, 2006).

Quando os músculos estão relaxados, os espaços vazios são ocupados pelos tecidos de gordura e as transições angulares são suavizadas, possibilitando assim, a visualização do formato geral do crânio (BAPTISTA, 2005).

A Figura 2.9 mostra a vista frontal dos músculos da expressão facial.

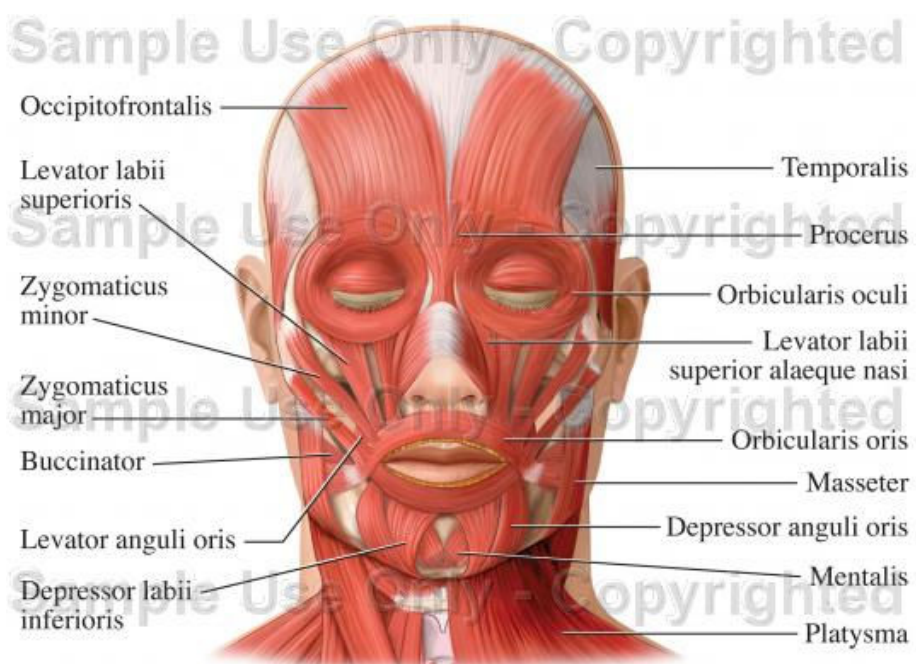


Figura 2.9 – Músculos da expressão facial, vista frontal.

Fonte: (MARTINI, 2006)

2.4.2.1 Músculos ao redor do globo ocular

Segundo Martini (2006), existem três músculos que se encontram na região ocular. São eles: músculo orbicular do olho, músculo corrugador do supercílio e músculo levantador da pálpebra superior.

O músculo orbicular (*orbicularis oculi*) do olho envolve o olho em fibras concêntricas. A parte orbital é responsável pelo fechamento forçado das pálpebras e a parte central é responsável pelo piscar involuntário. Desempenha um importante papel na proteção do olho, pois ele fecha firmemente as pálpebras, evitando sujeira e claridade

solar, além de prevenir contato de objetos contra o próprio olho. É o responsável pelo aparecimento de rugas conhecidas como “pés-de-galinha”.

O músculo corrugador do supercílio (*corrugator supercilli*) é um músculo horizontal que inicia no osso frontal e termina na extremidade lateral do supercílio. Este músculo é o responsável por causar rugas verticais na testa quando as sobrancelhas são empurradas para baixo.

O músculo levantador da pálpebra superior (*levator palpebrae superioris*) se encontra acima da caixa ótica e avança até o final da pálpebra superior. Quando ele está relaxado, realiza o fechar dos olhos.

2.4.2.2 Músculos do nariz

Os músculos do nariz são um pouco rudimentares, entretanto eles agem contraindo e dilatando as fossas nasais. Os músculos do nariz são: músculo prócero, músculo nasal e o músculo levantador do lábio superior e da asa do nariz (BAPTISTA, 2005).

O músculo prócero (*procerus*) é pequeno, tem início no osso nasal e é vizinho do orbicular do olho e do frontal. Encontra-se inserido na pele entre os supercílios. Auxilia na redução do efeito dos raios solares sobre a face (BAPTISTA, 2005).

O músculo nasal (*nasalis*) tem origem óssea na base do processo alveolar próximo à abertura piriforme, que é a região da face onde fica a cartilagem que separa as duas metades da cavidade nasal (MARTINI, 2006).

Já o músculo levantador do lábio superior e da asa do nariz (*levator labii superioris alaeque nasi*) é um músculo longo, pouco espesso, estende-se do processo frontal da maxila ao nível do ângulo do olho até o lábio superior (BAPTISTA, 2005).

2.4.2.3 Músculos da boca

Os músculos da boca são muito importantes na expressão facial e na fala. A Figura 2.10 mostra os principais músculos da boca que são: músculo orbicular da boca, músculo bucinador, músculo levantador do lábio superior, músculo zigomático maior, músculo zigomático menor, músculo levantador do ângulo da boca, músculo abaixador do ângulo da boca, músculo abaixador do lábio inferior, músculo risório e o músculo mentoniano. Alguns músculos são responsáveis por abrir os lábios, enquanto outros são responsáveis por fechar os lábios (BAPTISTA, 2005).

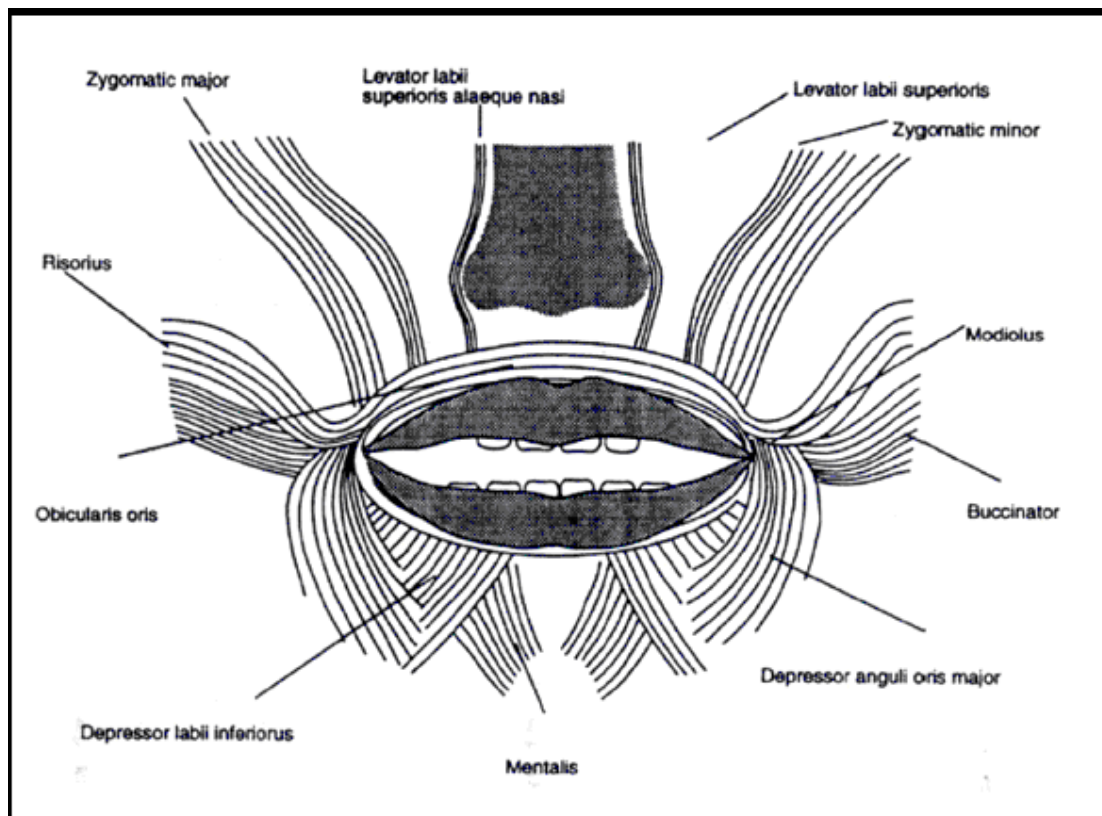


Figura 2.10: Músculos da boca

Fonte: (BAPTISTA, 2005)

O músculo orbicular da boca (*orbicularis oris*) é o músculo que está contido nos lábios. Ele produz um total controle sobre os lábios, auxiliando assim a mastigação e a fala (BAPTISTA, 2005).

O músculo bucinador (*buccinator*) é o músculo da bochecha. Este músculo possui duas funções: manter a bochecha distendida durante as fases de abertura e fechamento da boca (para que ela não dobre e seja ferida pelos dentes) e empurrar os alimentos até os dentes, auxiliando assim, a mastigação (MARTINI, 2006).

O músculo levantador do lábio superior (*levator labii superioris*) se encontra entre o músculo levantador do lábio superior e da asa do nariz e o zigomático menor.

O músculo zigomático maior (*zygomaticus major*) é conhecido como músculo do riso. Ele possui a ação de levar para cima e para fora a linha de junção, dando uma formação arqueada à boca (MARTINI, 2006).

O músculo zigomático menor (*zygomaticus minor*) está situado ao lado do levantador do lábio superior. Possui a função de auxiliar no movimento dos elevadores do lábio, permitindo que os dentes maxilares fiquem à mostra (BAPTISTA, 2005).

O músculo levantador do ângulo da boca (*levator anguli oris*) é um músculo curto e bem desenvolvido. Ele se encontra em uma posição mais profunda que os demais e está fixado na fossa canina, além de estar totalmente coberto pelo levantador do lábio superior. Ele acentua o ângulo da boca permitindo que os dentes sejam expostos (MARTINI, 2006).

O músculo abaixador do ângulo da boca (*depressor anguli oris*) é um músculo muito superficial que cobre um pouco o abaixador do lábio inferior e do *bucinador*. Este músculo permite que a linha de junção da boca para baixo seja puxada, aprofundando assim o canto dos lábios (BAPTISTA, 2005).

O músculo abaixador do lábio inferior (*depressor labii inferioris*) tem sua origem logo acima da linha de origem do abaixador do ângulo da boca. A função deste músculo é descer o lábio inferior (BAPTISTA, 2005).

O músculo risório (*risorius*) se encontra no canto da boca e é um músculo fino. Este músculo puxa o ângulo da boca lateralmente e durante a contração, produz uma “cavinha” na bochecha. Ele é muito conhecido como o “músculo do sorriso” (BAPTISTA, 2005).

O músculo mentoniano (*mentalis*) está situado em um espaço triangular formado pelas margens mediais dos músculos abaixadores do lábio inferior. Este músculo eleva a pele do queixo e vira o lábio inferior para fora, permitindo, por exemplo, o movimento para beber (MARTINI, 2006).

2.5 Animação facial

Muitas pesquisas sobre modelagem e animação facial vêm sendo realizadas nos últimos anos. A modelagem trabalha na construção de malhas e a animação facial trabalha com técnicas para gerar e animar imagens da face humana.

2.5.1 História da animação facial

As pesquisas na área de animação facial tiveram início em 1972, com o trabalho de Frederick Parke: *Computer Generated Animation of Faces*. Parke modelou a face humana

no computador. Para isso ele utilizou técnicas de fotometria e de interpolação linear. Para a modelagem, ele marcou vários pontos de referência em uma face e fotografou a mesma em diferentes pontos de vista. Para a animação, ele utilizou a técnica de interpolação, interpolando a posição de determinados vértices no rosto inicial até a posição dos vértices na nova expressão (WEY, 1999). A Figura 2.11 ilustra uma animação utilizando interpolação linear.



Figura 2.11: Animação utilizando interpolação linear

Fonte: (WEY, 1999)

A técnica de interpolação linear apresenta bons resultados em modelos que não possuem muita expressividade. À medida que a expressividade aumenta, o animador precisa trabalhar com toda a face para cada nova expressão criada. Para simplificar a criação de expressões faciais, Parke desenvolveu modelos paramétricos da face humana. A face humana foi dividida em determinadas áreas para que o animador pudesse controlar os parâmetros referentes a cada área e, em seguida, modelar novas expressões (MARTINI, 2006).

Em 1985, (BERGERON e LACHAPELLE, 1985) aprimoraram o trabalho de Parke. Eles fizeram um personagem animado em três dimensões, onde este personagem transmitia seu sentimento através de expressões faciais. Bergeron e Lachapelle utilizaram técnicas de interpolação linear e parametrização, onde, dada uma expressão facial neutra, novas expressões eram criadas com a alteração de determinados vértices do modelo. Assim, surgiu a técnica de soma de vetores dos vértices. Esta técnica consiste em realizar o cálculo das distâncias dos vértices da face na expressão neutra e nas outras expressões. Somando e subtraindo estes vértices foi possível reunir várias expressões em uma mesma face simultaneamente (WEY, 1999).

Em 1987, Nadia Magnenat Thalmann propôs o modelo de simulação dos músculos da face, elevando assim o nível dos trabalhos de animação facial. Este modelo reproduz a ação dos músculos através de técnicas de deformação de formas livres. As expressões são divididas em pequenas ações e, para cada ação, é determinado o movimento que os “pseudomúsculos” devem fazer (MARTINI, 2006).

Neste mesmo ano, o pesquisador Keith Waters propôs um sistema de simulação de músculos baseado em modelos físicos. Foram dados valores de “massa” aos vértices dos polígonos que compõe a face e a ação dos músculos foi simulada por molas, caracterizando bem a ação dos músculos (MARTINI, 2006).

No ano de 1991, Nadia Magnenat Thalmann desenvolveu e aprimorou uma técnica, semelhante à de Parke, para animação facial. Esta técnica utilizou o conceito de animação em camadas, onde as camadas inferiores determinam como a animação é feita e as camadas superiores determinam o que deve ser feito (WEY, 1999).

Atualmente, as pesquisas na área de modelagem e animação facial estão direcionando seus esforços na geração de modelos tridimensionais foto-realistas. Através de técnicas como fotometria, o modelo em três dimensões é obtido através da extração de parâmetros obtidos por imagens de vídeo. As imagens ainda contribuem para o mapeamento das texturas, tornando o modelo mais realista (MARTINI, 2006).

2.5.2 Modelagem e animação facial

Para se desenvolver um modelo facial é necessário passar por duas etapas: modelagem da face (malha de um rosto) e animação. A etapa de modelagem da face consiste em desenvolver uma malha de polígonos de um rosto. Já na etapa de animação, é realizada a transição de uma expressão para a outra.

Existem várias técnicas de representação facial, dentre elas, representação volumétrica, de superfícies e a técnica de geração de novas faces a partir de faces existentes (RODRIGUES, 2005).

Na representação volumétrica, a modelagem da face é feita utilizando técnicas de representação volumétricas, como por exemplo, *Constructive Solid Geometric* (CSG) e *arrays* de elementos de volume (*voxels*). No CSG os objetos de interesse são representados por planos, cilindros e esferas. O *voxel* é muito utilizado para descrever estruturas anatômicas em imagens médicas (RODRIGUES, 2005).

A técnica de representação de superfícies é a mais utilizada para modelagem facial, pois as estruturas de superfície utilizadas permitem formatos de superfície e mudanças nos formatos quando necessário para as várias expressões faciais. As superfícies mais utilizadas são superfícies implícitas, superfícies paramétricas e superfícies poligonais (malhas) (RODRIGUES, 2005).

A técnica de geração de novas faces a partir de faces existentes consiste em criar novas faces tendo como entrada outra face (RODRIGUES, 2005).

No contexto de expressões faciais é muito importante pensar em como realizar a transição entre duas expressões faciais de forma que esta transição seja realista. Uma vez que a face foi modelada, o passo seguinte consiste de sua animação. Existem várias técnicas para a animação facial: interpolação, baseada em performance, parametrização direta, animação baseada em pseudomúsculos, animação baseada em músculos e simulação biomecânica (SANGOI, 2008).

A técnica de interpolação linear é a mais utilizada para animação facial. No caso de uma dimensão são dados dois valores e deseja-se determinar um valor intermediário, onde esse valor intermediário é especificado por um coeficiente de interpolação fracionária α , com α variando entre 0 e 1.

$$\text{valor} = (1 - \alpha) \text{valor}_1 + \alpha \text{valor}_2$$

Vale a pena lembrar que este conceito pode ser expandido para mais de uma dimensão, aplicando-se esse mesmo procedimento em cada uma das dimensões. Para superfícies poligonais, a interpolação é feita para cada vértice (RODRIGUES, 2005).

A técnica baseada em performance, também conhecida como parametrização *data-driven*, utiliza como parâmetros informações obtidas com equipamentos especiais de mensuração de ações humanas, ou seja, utiliza informações derivadas por uma medida de ações reais humanas para dirigir personagens sintéticos (SANGOI, 2008).

A parametrização direta tem como objetivo criar um modelo encapsulado que gera um intervalo grande de faces e de expressões faciais tendo como base o menor conjunto de parâmetros de controle possível (RODRIGUES, 2007).

Já a técnica de animação baseada em pseudo-músculos propõe que sejam simuladas as ações básicas dos músculos da face através do controle de alguns parâmetros (RODRIGUES, 2007).

A animação baseada em músculos utiliza a estrutura facial dos ossos, músculos, tecidos de conectividade e pele. Essa técnica possui a habilidade de manipular expressões faciais através de uma simulação das características dos músculos (RODRIGUES, 2007).

A simulação biomecânica representa as propriedades físicas da pele, conforme ilustrado na Figura 2.12. Nessa técnica, os músculos faciais são modelados como parâmetros de manipulação, prontos para desencadear a movimentação da face virtual. O tecido facial é representado por uma malha de sistemas massa-mola, que procura simular as propriedades elásticas da pele humana (KORTING, 2007).

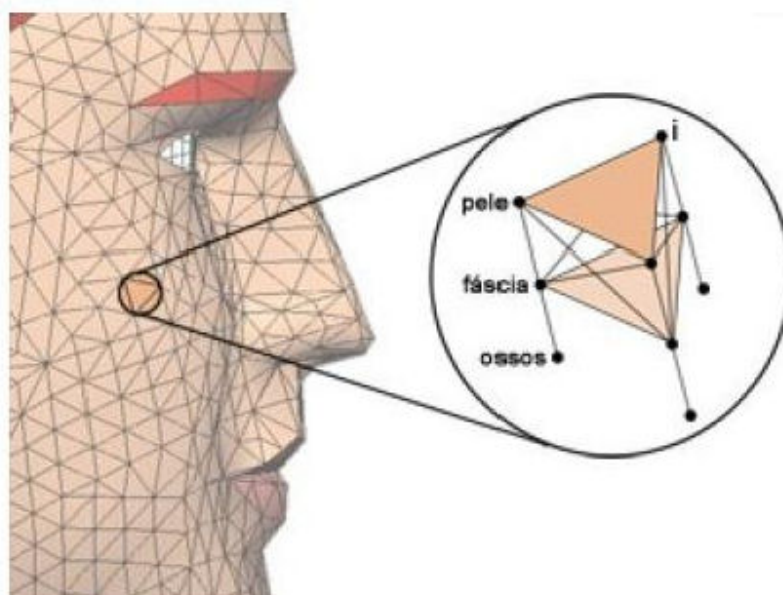


Figura 2.12: Detalhes da malha facial

Fonte: (SANGOI, 2008)

Capítulo 3

Morphing

Este capítulo apresenta uma breve revisão bibliográfica das técnicas conhecidas como *warping* e *morphing*, bem como as suas fases e as principais técnicas de deformação existentes.

3.1 Introdução

O *morphing* é uma técnica de processamento de imagens cujo objetivo é gerar uma animação que transforma, de forma gradual, uma imagem em outra (NOGUEIRA, PRATA e VELHO, 2003). Esta técnica é muito utilizada para efeitos visuais na indústria cinematográfica e de entretenimento. O clipe de *Michael Jackson* intitulado “*Black or White?*” é um exemplo de aplicação desta técnica, que a tornou bastante conhecida e popular. A Figura 3.1 ilustra o *morphing* de duas imagens, onde um carro se transforma em um tigre.



Figura 3.1: Carro se transformando em tigre.

Fonte: (SCHROEDER, 2007)

Para que uma imagem inicial seja gradualmente transformada em outra, é necessário realizar simultaneamente a deformação espacial e a transformação de tonalidades (SCHROEDER, 2007). Para realizar esta transformação é necessário definir elementos correspondentes em ambas as imagens fonte e destino, ou seja, pares de características devem ser destacados, como, por exemplo, uma boca de uma imagem com a boca da outra. Essas regiões podem ser marcadas com malhas de pontos, segmentos de retas, curvas ou pontos, sendo que cada uma dessas especifica um

atributo ou um ponto de referência. Este mapeamento de características corresponde à fase de “Especificação”, onde funções de mapeamento são criadas para definir uma relação espacial entre os pontos chaves nas duas imagens. Essas funções de mapeamento correspondem ao processo de deformação e são utilizadas para interpolar as posições dos atributos correspondentes nas imagens durante esta fase do processo de *morphing* (HIGA, 2008).

3.2 Estágios do *morphing*

Segundo Schroeder (2007), o *morphing* de imagens é composto de 4 estágios: especificação, deformação, transformação de tonalidades e controle de transição. Esses estágios serão detalhados a seguir.

3.2.1 Especificação

Conforme dito anteriormente, a especificação é o estágio onde pares de características são destacados, tanto na imagem fonte como na imagem de destino.

Para que o *morphing* alcance os resultados desejados é necessária a especificação, por parte do usuário. Ainda não existem, no momento, técnicas totalmente automáticas (DARSA, 1994), mesmo porque esta etapa envolve um certo toque artístico. Mas para especificação em faces, já existem tentativas de especificação automática. Uma maneira de especificação automática pode ser encontrada no trabalho de (ZANELLA e FUENTES, 2004).

Existem algumas técnicas de especificação, dentre elas, especificação por partição e especificação por características.

a) Especificação por partição

Esta técnica de especificação utiliza malhas. Nesta abordagem, duas malhas, com a mesma estrutura topológica, são superpostas sobre a imagem, sendo que cada malha define uma partição do domínio de uma imagem (SILVA, 1994).

A Figura 3.2 mostra um exemplo da especificação por partição, onde duas malhas são utilizadas para fazer a especificação, uma malha na imagem inicial e outra malha na imagem final.

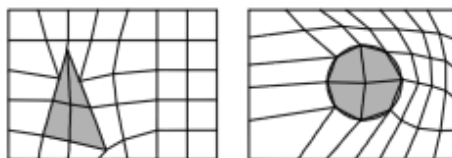


Figura 3.2: Especificação por partição

Fonte: (DARSA, 1994)

b) Especificação por características

Esta especificação é realizada por primitivas que podem ser pontos, segmentos de reta, grades, curvas, etc., exigindo atenção especial do animador no posicionamento preciso das mesmas. Estas primitivas são posicionadas na imagem sobre as características que elas irão mapear e realizar a devida correspondência, como por exemplo, regiões da face (para imagens com face) como olhos, sobrancelhas, nariz e boca (SCHROEDER, 2007).

As primitivas precisam ser especificadas tanto na imagem inicial, quanto na final. Além disso, tem que haver uma correspondência entre as características da imagem inicial e da final, ou seja, se existe uma primitiva na imagem inicial sobre a boca, então deverá existir a mesma primitiva sobre a boca da imagem final (SCHROEDER, 2007).

3.2.2 Deformação

Esta fase é composta pela deformação das imagens através do alinhamento das características mapeadas na fase de Especificação. É através da especificação de atributos que é possível realizar uma correspondência entre as imagens, permitindo que o *morphing* ocorra (SCHROEDER, 2007).

A deformação ou *warping* consiste em manipular uma imagem fazendo com que as formas presentes na imagem sejam deformadas de acordo com as especificações da fase anterior, ou seja, da fase de especificação (HIGA, 2008).

O *warping* de imagens é a principal técnica de processamento de imagens empregada no *morphing* e é a etapa mais complexa, pois a qualidade e a performance da visualização depende desta fase. No *morphing*, as especificações das duas imagens são alinhadas para que a transição entre elas seja a mais suave possível. O resultado da deformação depende da quantidade de características que foram especificadas: quanto maior a quantidade, melhor o resultado do *warping* (SCHROEDER, 2007).

A deformação realiza a transformação de pontos entre as duas imagens. Esta transformação é definida com o objetivo de converter cada *pixel* da imagem original para o *pixel* correspondente na imagem final. Com o mapeamento que foi especificado no estágio anterior, a deformação de imagens promove o mapeamento entre essas duas imagens a partir dessas especificações (HIGA, 2008).

Matematicamente, o *warping* de uma imagem consiste em uma função $w: U \rightarrow W$, onde $U, W \subset \mathbb{R}^2$. A função w transforma a posição de um ponto pertencente a uma imagem inicial U , produzindo uma nova imagem, que será chamada de imagem final W . A deformação de imagens pode ser entendida como o deslocamento de um *pixel* de uma imagem inicial para outra posição. O resultado obtido ao deslocar todos os *pixels* da imagem inicial gera uma imagem final.

A deformação final obtida varia, de acordo com a função do método de interpolação que será utilizado. Esta interpolação se encarregará de preencher as lacunas da imagem, com pontos não mapeados diretamente no estágio anterior.

Existem vários métodos para se realizar a deformação, dentre eles: deformação por malha, deformação por malhas de triângulos, deformação por campo, deformação por funções de base radial, entre outros.

a) Deformação por malha

Um dos primeiros métodos de *warping* descritos na literatura é o método de Deformação por Malha, proposto por George Wolberg. Este método foi utilizado pela empresa *Industrial Light Magic* no filme *Willow* em 1988 e, mais tarde foi usado em muitos outros filmes (WOLBERG, 1994).

Este método de deformação se baseia em aplicar uma função de transformação em duas imagens: uma imagem origem chamada de I_0 e uma imagem destino chamada de I_1 . Após definir as imagens que serão utilizadas no processo, são identificadas correspondências entre as imagens com o intuito de se estabelecer um critério para a

deformação. Neste algoritmo, essa correspondência é feita por uma grade em ambas as imagens, conforme mostrado na Figura 3.3, onde cada linha da grade das imagens origem e destino representam a mesma característica como, por exemplo, olhos, boca e nariz. A correspondência de características corresponde aos pontos de interseção das linhas das grades. Desta forma, a quantidade de pontos de interseção tem que ser a mesma em ambas as imagens. Cada ponto de interseção da grade da imagem origem é deslocado para a posição correspondente do ponto de interseção da imagem destino e as células das grades são então deformadas de tal forma que acompanhem este deslocamento dos pontos de interseção. Essa deformação pode ocorrer em várias etapas, gerando vários quadros de imagens intermediários, até conseguir ter a mesma forma da imagem destino.

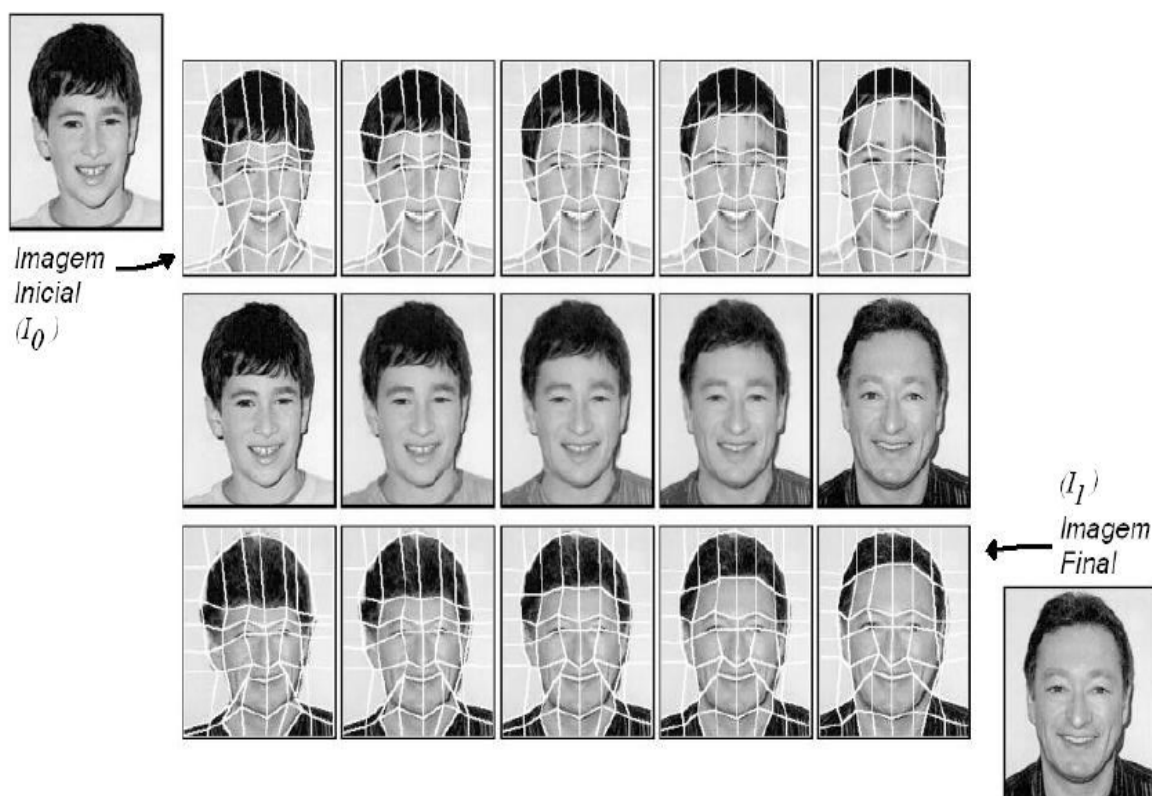


Figura 3.3: Deformação por malha

Fonte: (SCHROEDER, 2007).

A grade associada à imagem original I_0 será chamada de M_0 e a da imagem destino I_1 será chamada de M_1 , sendo que as posições da grade M_1 estão relacionadas com as posições da grade M_0 . Juntas, M_0 e M_1 definem uma função de transformação espacial

onde todos os pontos de I_0 são mapeados em I_1 . Vale a pena lembrar que as grades devem ser topologicamente iguais, permitindo que os pontos de interseção das grades se desloquem se necessário (SCHROEDER, 2007).

b) Deformação por malhas de triângulos

Nesta técnica, a deformação é descrita por duas malhas triangulares: uma para definir um sistema de coordenadas não deformado, e a outra para definir uma deformação deste sistema. As duas malhas precisam ter a mesma estrutura topológica para que a correspondência entre pares de triângulos esteja bem definida. Um mapeamento precisa ser estabelecido entre os interiores de cada par de triângulos correspondentes para que exista um mapeamento de sistemas de coordenadas global.

A Figura 3.4 mostra uma malha de triângulos não deformada e uma outra malha de triângulos deformada.

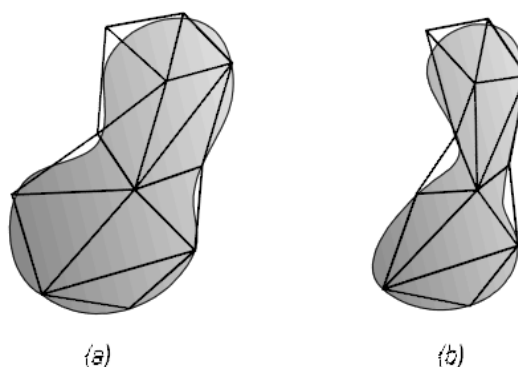


Figura 3.4: Malha não deformada e malha deformada, respectivamente

Fonte: (DARSA, 1994)

c) Deformação por campo

A técnica de *warping* por campo foi proposta inicialmente por *Thaddeus Beier e Shawn Neely* (BEIER e NEELY, 1992). Os ajustes das características na deformação por malha são muito trabalhosos, porque todos os pontos da malha devem ser correlacionados. A deformação por campo facilita a interface do usuário, utilizando segmentos de reta ou vetores na fase de especificação de características (HIGA, 2008). Estes segmentos elementos são posicionados sobre características das imagens e

determinam a área de deformação das mesmas. Uma das características desse algoritmo é o fato de que ele utiliza menos pontos do que o *morphing* por malha.

Dependendo do número de vetores e do seu posicionamento, um vetor pode influenciar na área de deformação de outro vetor. Na Figura 3.5, o vetor \overrightarrow{PQ} é posicionado sobre uma característica da imagem inicial I_0 , da mesma forma que o vetor $\overrightarrow{P'Q'}$ é posicionado sobre esta mesma característica da imagem final I_1 . O vetor \overrightarrow{PQ} é definido sobre as características de interesse. O ponto X é o *pixel* que será analisado e conforme a posição dos vetores iniciais e finais (\overrightarrow{PQ} e $\overrightarrow{P'Q'}$), o *pixel* poderá ter sua localização na imagem destino alterada para outra localização ou não (HIGA, 2008).

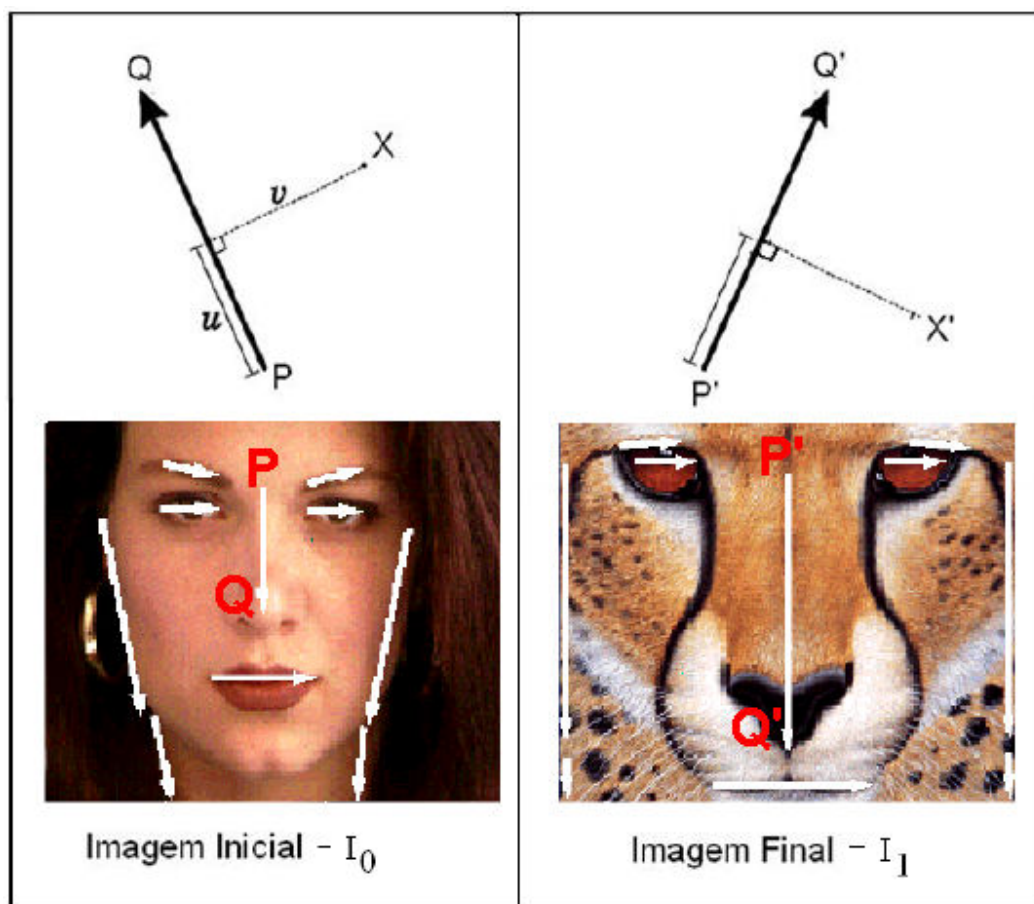


Figura 3.5: Vetores iniciais e vetores finais

Fonte: (SCHROEDER, 2007)

O cálculo da nova localização do *pixel* na imagem final utiliza as coordenadas dos pontos X, P, P', Q e Q' e algumas medidas que podem ser observadas na Figura 3.5.

Essas medidas são v e u onde: v, é a distância do ponto X ao ponto que corta o vetor \overrightarrow{PQ} através de uma linha perpendicular a esse vetor, representando a distância na direção perpendicular do vetor ao ponto; u é a distância medida pelo ponto P até o local onde a linha perpendicular que passa pelo ponto X corta o vetor. A distância u é também chamada de distância ao longo do vetor. Esses valores são calculados para se obter a nova localização do pixel X, que será chamado de X' na imagem deformada. u, v e X' são calculadas da seguinte maneira (ZAMITH et al, 2009):

$$u = \frac{(X - P)(Q - P)}{\|Q - P\|^2}$$

$$v = \frac{(X - P)(Q - P)^\perp}{\|Q - P\|}$$

$$X' = P' + u(Q' - P') + \frac{v(Q' - P')^\perp}{\|Q' - P'\|}$$

Os pontos denotados por um apóstrofe são os pontos da imagem destino. A notação $\|...\|$ expressa o cálculo da distância euclidiana entre dois pontos e a notação $(...)^{\perp}$ significa o vetor $(-y, x)$ do vetor (x, y) (SCHROEDER, 2007).

Para a implementação deste trabalho foi utilizado o *warping* por campo, onde todos os cálculos acima foram executados em paralelo. A implementação, a estrutura de dados e a estrutura processual são detalhadas no Capítulo 5.

d) Deformação por funções de base radial

Este método foi inicialmente proposto por *Nur Arad*. As características são especificadas por pontos, tanto na imagem inicial quanto na imagem final, estabelecendo assim a correspondência de elementos entre as imagens e, para a deformação das imagens, o método utiliza funções de base radial (ARAD e REISFELD, 1995).

Uma função de base radial é uma transformação que realiza um mapeamento de uma imagem para outra, à partir de um conjunto de pontos, chamados pontos de controle, que descrevem a posição de uma característica na imagem. Cada ponto na imagem inicial tem somente um ponto correspondente na imagem final, resultando numa

seqüência de pares de controle. Portanto, são informados pares de pontos de controle \vec{p}_i e \vec{q}_i na imagem inicial e final, onde $\vec{p}_i = (x_i, u_i)$ é a coordenada no estado inicial e $\vec{q}_i = (u_i, v_i)$ é a coordenada no estado final, ou seja, o lugar onde a coordenada inicial deverá estar após a transformação. Portanto, uma função de base radial é uma transformação que descreve uma relação espacial entre estes pontos de controle correspondentes, ou seja, mapeia um conjunto de n pontos de controle \vec{p}_i , da imagem inicial em pontos correspondentes \vec{q}_i da imagem deformada ($i = 1, \dots, n$). Esse mapeamento é representado por funções definidas como uma combinação de funções de base radialmente simétricas, cada uma centrada em um ponto de controle. Após essas funções serem encontradas, elas são usadas em todo o domínio da imagem para realizar a deformação na imagem inteira (SCHROEDER, 2007).

As funções de base radial apresentam um comportamento facilmente controlável, já que não existem restrições nas posições dos pontos de controle sobre as imagens. Esse tipo de função também permite realizar deformações tanto locais como deformações globais (SCHROEDER, 2007).

3.2.3 Transformação de tonalidades

Transformação de tonalidades é a fase do *morphing* em que as tonalidades das imagens, geradas no decorrer da deformação, são interpoladas, conforme mostrado na Figura 3.6. Com a aplicação de uma função de *warping* o processo de *morphing* cria uma seqüência de *frames*, onde uma imagem inicial se transforma em outra imagem final. O número de *frames* é que define o tempo da animação. Esse número de *frames* é muito importante porque é através destes *frames* que as imagens serão deformadas gradualmente. No *morphing*, tanto a imagem inicial pode ser deformada para a imagem final, como a imagem final é deformada para a imagem inicial. Portanto, as funções de deformação são utilizadas tanto na imagem inicial, como na imagem final. Depois de ocorrer a deformação, obtém-se imagens intermediárias. Estas imagens intermediárias precisam passar pelo processo de transformação de tonalidades conhecido na literatura como *cross-dissolve* (SCHROEDER, 2007).

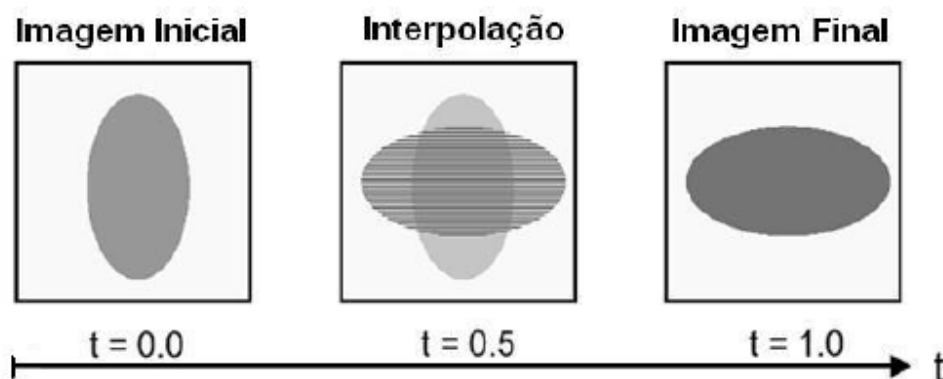


Figura 3.6: Interpolação das tonalidades das imagens

Fonte: (SCHROEDER, 2007)

Matematicamente, dadas duas imagens I_1 e I_2 , um *cross-dissolve* é simplesmente uma transformação a um parâmetro I_t que produz uma nova imagem, ou seja, um quadro do *morphing*. A equação abaixo mostra o cálculo do *cross-dissolve*.

$$I_t(x,y) = (1 - C_t) I_1(x,y) + C_t I_2(x,y) = I_1(x,y) + C_t [I_2(x,y) - I_1(x,y)]$$

Sendo que C_t varia de 0 a 1, logo com a variação de C_t , uma animação é produzida, onde I_1 se transforma em I_2 (SILVA, 1994).

A Figura 3.7 mostra o processo de transformação de tonalidade. As imagens (a) e (e) são, respectivamente, as imagens iniciais e finais. As imagens (b), (c) e (d) são as imagens resultantes da transformação de tonalidades.

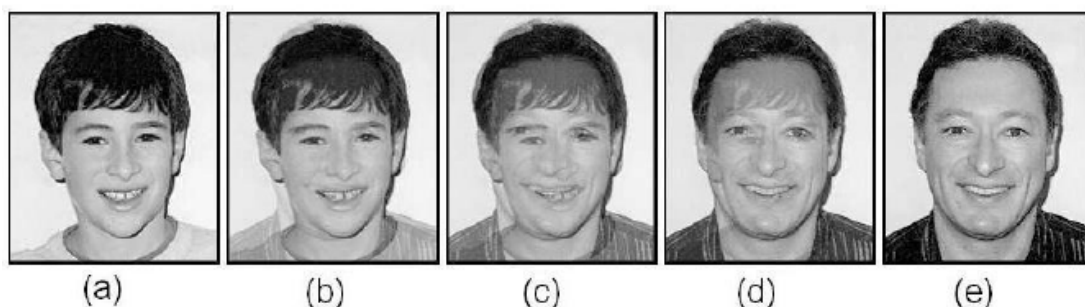


Figura 3.7: Interpolação de cores (*Cross-Dissolve*)

Fonte: (SCHROEDER, 2007)

3.2.4 Controle de transição

A taxa de transição no *morphing* permite que as características especificadas na imagem sejam deformadas em diferentes tempos. A taxa de transição é o tempo C_t utilizado na equação anterior. Esta mesma taxa de transição é aplicada a todos os pontos para gerar a imagem intermediária I_t (KORTING, 2007).

O tempo C_t é que determina a taxa da transição entre as imagens I_0 e I_1 , sendo que a taxa de transição é um valor que varia de 0 a 1. Para valores de t próximo de zero, as imagens intermediárias se parecem com a imagem inicial I_0 . Já para t próximo de 1, as imagens intermediárias ficam parecidas com a imagem final I_1 (SCHROEDER, 2007).

Capítulo 4

Deformação da superfície para as rugas

4.1 Introdução

Diversas técnicas foram desenvolvidas para dar mais realismo às imagens geradas pelo computador. Um componente que permite um aspecto maior de realismo em uma imagem é a aplicação de textura, que pode ser bidimensional ou tridimensional, sendo que ambas podem ser vistas na superfície externa do objeto. Estes dois tipos de textura podem ser gerados de duas formas: procedural ou não procedural. As não procedurais podem ser definidas por imagens digitalizadas ou qualquer outro tipo de informação que não seja resultado de um processo algorítmico. Já as procedurais, são texturas geradas matematicamente através da execução de um algoritmo que faz com que os padrões sejam gerados por procedimentos baseados em métodos que utilizam como argumentos de entrada as coordenadas cartesianas do ponto do objeto em que é aplicada a textura e, como resultado, retornam um valor correspondente ao ponto especificado. Muitos métodos foram desenvolvidos para fazer a aplicação de texturas.

Para este trabalho foi adotada a técnica de *bump mapping* para a representação das rugas. Esta técnica é vantajosa para esta finalidade, pelo fato de não requerer um grande número de polígonos para representar pequenas deformações em superfícies, como é o caso das rugas da pele. Como a percepção da rugosidade em uma superfície é resultado direto da forma como a luz interage com ela, o *bump mapping* realiza pequenas modificações na direção do vetor normal da superfície antes de ser calculada a iluminação.

4.2 O *bump mapping*

Bump mapping é uma técnica que faz com que a superfície tenha uma aparência áspera, enrugada, dentada ou perfurada. É uma técnica que produz efeitos visuais muito realistas sem modificar a geometria do objeto (BLINN, 1978). Ela provoca perturbações no vetor normal usada para o cálculo da iluminação de cada ponto da superfície de um

polígono. Ao se perturbar o vetor normal, o ângulo de incidência da luz é alterado, simulando assim, a existência de uma superfície irregular (CASTRO, 2004).

A Figura 4.1 ilustra a superfície P onde a geometria do objeto é suave em sua extensão. Para simular irregularidades nesta superfície, sem que seja necessário alterá-la diretamente, aplica-se uma função de perturbação F nas normais ao longo da superfície e o resultado obtido é a superfície perturbada P' .

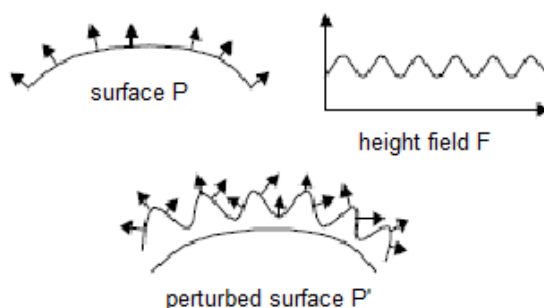


Figura 4.1: Resultado da aplicação de uma perturbação em uma superfície (Akenine-Moller 2008)

Percebe-se que a geometria da superfície permanece inalterada, mas as normais ao longo da superfície não são mais suaves, gerando assim, um efeito de irregularidade ao se aplicar o modelo de iluminação.

A Figura 4.2 mostra dois exemplos da aplicação do *bump mapping*.

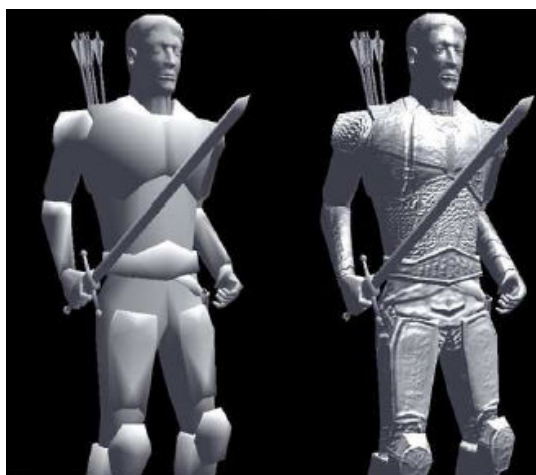


Figura 4.2a: *Bump mapping* em um personagem



Figura 4.2b: *Bump mapping* em uma parede de pedras
(NEHE, 2010)

4.3 Métodos para armazenamento dos modelos de perturbação da normal

Existem várias maneiras de se armazenar os dados que serão usados para deformar as normais. Neste trabalho foi utilizado o mapa de normal.

Um mapa de normal é uma textura em 2D que contém informações sobre a direção da normal da superfície em cada pixel. Esta textura está no padrão RGB, mas o valor armazenado representa a normal em cada pixel, ou seja, as coordenadas do vetor normal x , y e z são armazenadas como cores no padrão RGB, onde a coordenada x da normal corresponde canal R, a coordenada y corresponde ao valor de G e a coordenada z ao canal B, conforme ilustrado na Figura 4.3.

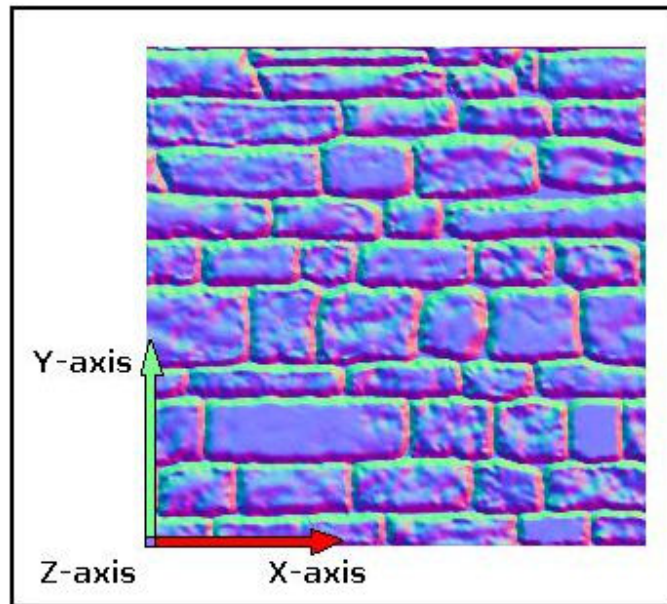


Figura 4.3: Coordenadas x, y e z contendo informações da direção do vetor normal da superfície.
(NEHE, 2010)

Percebe-se que o mapa de normal fica com uma cor azulada. Isso se deve ao fato de que a normal da superfície aponta para fora da textura, ou seja, para o eixo z que corresponde ao canal B (SALGADO, 2006).

A direção da normal é definida por três coordenadas, x, y e z, sendo que cada coordenada está no intervalo de $[-1, 1]$. Entretanto, na textura RGB, os canais estão no intervalo de $[0, 1]$. Desta maneira, é necessário realizar uma transformação do intervalo de $[0, 1]$ para $[-1, 1]$ (POZZER, 2009). Segundo (NVIDIA, 2010), a equação para realizar esta conversão é:

$$\text{direction} = 2.0 * (\text{colorValue} - 0.5)$$

Com a equação acima é possível ler e interpretar a direção da normal em cada *pixel* do mapa de normal.

4.4 Sistemas de coordenadas

Para que o *bump mapping* funcione para qualquer tipo de geometria aplicada, ou seja, independente da curvatura ou da orientação da superfície, é necessário realizar uma

transformação. Com esta transformação, o mapa de normais pode ser aplicado em qualquer parte da superfície do objeto (EVANGLISTA e SILVA, 2007).

É necessário que o vetor de luz e o observador compartilhem o mesmo sistema de coordenadas do vetor normal do mapa de normais, pois o cálculo de iluminação deve ser feito com a superfície e a fonte de luz no mesmo espaço, seja no espaço do objeto, ou no espaço do mundo ou no espaço da tangente (CONCI e AZEVEDO, 2003). Existem duas maneiras de colocar a fonte de luz e o observador no mesmo espaço do vetor normal. A primeira maneira é transformar o vetor de luz e o observador para o espaço tangente e a segunda maneira é transformar todas as normais da superfície do mapa de normal para o espaço do objeto (AKENINE-MOLLER, 2008). A primeira opção é a melhor, pois somente a fonte de luz e o observador serão convertidos. Já na segunda opção, todas as normais do mapa de normal seriam convertidas para o espaço objeto. Logo é melhor e menos custoso transformar o vetor de luz e o observador para o sistema de coordenadas do mapa de normais. O sistema de coordenadas do mapa de normal é chamado de espaço da tangente (FERNANDO e KILGARD, 2003).

Espaço tangente é um sistema de coordenada que corresponde a uma superfície plana. Este espaço é composto por três vetores (T, B, N), onde T corresponde à tangente, B à binormal e N à normal. A tangente e a binormal são vetores no plano e a normal (do mapa de normal) é perpendicular aos vetores tangente e binormal (JOUVIER, 2010).

Para converter o vetor de luz, que está no espaço objeto, para o espaço tangente é necessário construir a matriz que contém os três vetores (T, B e N) e multiplicar o vetor de luz por esta matriz. A Figura 4.4 mostra a matriz que será utilizada para a transformação (VILLAR, 2010).

$$\text{mat3 TBN} = \begin{Bmatrix} T_x & T_y & T_z \\ B_x & B_y & B_z \\ N_x & N_y & N_z \end{Bmatrix}$$

Figura 4.4: Matriz utilizada para a transformação

Fonte: (VILLAR, 2010)

Capítulo 5

Morphing paralelo na GPU

Com o advento das GPUs, diversas tarefas tipicamente executadas na CPU, puderam passar a ser feitas na GPU, desde que sejam capazes de ser paralelizadas numa arquitetura *stream based*. Assim, tarefas que eram inviáveis de serem executadas em tempo real em plataformas de baixo custo, puderam se tornar viáveis, mediante este recurso. Neste trabalho utiliza-se uma proposta de *morphing* de imagens paralela na GPU, de maneira a tornar a animação facial viável para aplicações interativas.

5.1 CUDA

Em novembro de 2007 a nVidia lançou a arquitetura CUDA com o objetivo de possibilitar a programação genérica em placas gráficas (NVIDIA, 2010).

No CUDA, a GPU é vista como um co-processador de dados paralelos, chamando a GPU de dispositivo (*device*) e a CPU de *host* (NVIDIA, 2010).

A Figura 5.1 mostra alguns conceitos importantes existentes nesta plataforma, dentre eles, *threads*, *blocks*, *grids* e *kernels*. *Threads* são as unidades de execução paralela em uma GPU. Elas são agrupadas em *blocks*, onde as *threads* que pertencem a um mesmo *block* podem sincronizar sua execução e compartilhar um mesmo espaço de memória (*shared memory*). Um conjunto de *blocks* representa um *grid*. E um *kernel* consiste no código que é executado por cada *thread*. Cada chamada a um *kernel* precisa especificar uma configuração contendo o número de *blocks* em cada *grid*, o número de *threads* em cada *block* e opcionalmente a quantidade de memória compartilhada a ser alocada (NVIDIA, 2010).

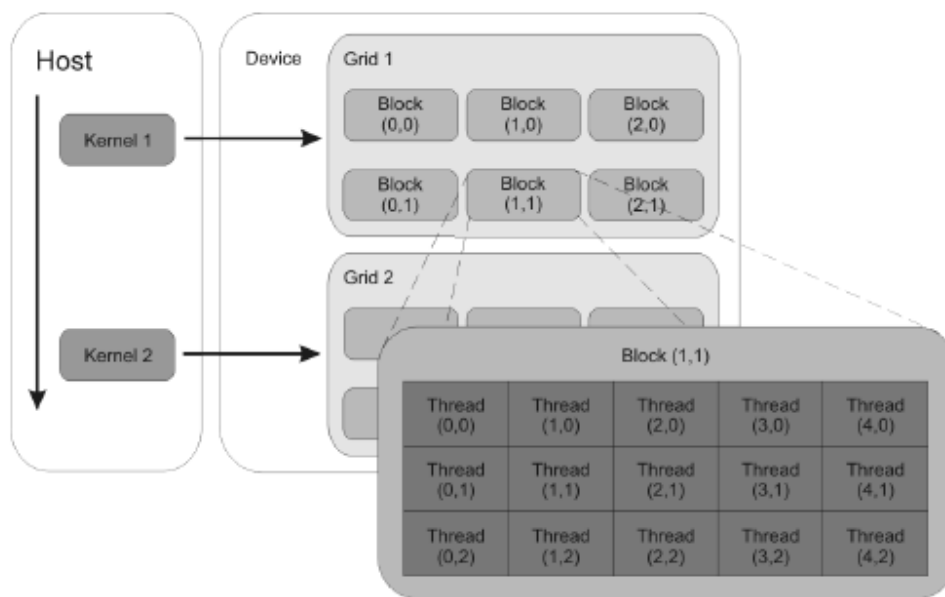


Figura 5.1: Modelo de execução do CUDA

Fonte: NVIDIA, 2010

Sua sintaxe se parece com a linguagem C, com algumas extensões. Estas extensões definem o escopo de uma função ou variável. Os escopos permitidos são *host*, *global* e *device*. O *host* define que uma função ou variável só pode ser utilizada ou executada na CPU, o *global* define que a função deve ser executada na GPU, mas que pode ser invocada pela CPU, e o *device* define que a função ou variável pode apenas ser invocada na GPU. Extensões também foram criadas para informar que uma variável faz parte da memória compartilhada e desta forma, estará acessível apenas a *threads* de um mesmo *block* (NVIDIA, 2010).

CUDA pode ser utilizada em conjunto com a ferramenta *Microsoft Visual Studio*, através da configuração das ferramentas de CUDA no *Visual Studio*. Para a implementação do algoritmo de *morphing* paralelo foi utilizado o *Microsoft Visual C++ 2008*.

5.2 *Morphing* paralelo de texturas 2D em tempo real

O algoritmo de *morphing* paralelo proposto por Zamith (ZAMITH et al, 2009) transforma imagens e texturas em tempo real na GPU. A implementação foi feita em

CUDA para explorar as características de paralelismo das GPUs atuais. O algoritmo proposto processa diversos pixels simultaneamente.

5.2.1 Estrutura de dados e estratégia de alocação

Os dados de entrada são representados por cinco matrizes: duas matrizes representam a textura fonte e a textura destino, outras duas contém as linhas guias e a última é uma matriz temporária e utilizada para carregar a interpolação das primeiras matrizes que são usados na reconstrução do *warping*, antes do *blending*.

As matrizes são alocadas dinamicamente no início da aplicação na memória global do dispositivo gráfico. As texturas e as suas linhas guias são transferidas da CPU para a GPU antes do processo começar e permanecem lá até que todo o processamento seja feito, evitando assim, que dados sejam transferidos desnecessariamente, pois a cada novo *frame* a textura produzida é enviada diretamente para o estágio de *render* sem sair da placa gráfica.

5.2.2 Estrutura processual

O algoritmo foi subdividido em quatro etapas, implementadas como quatro funções que definem a comunicação entre o *host* e o dispositivo gráfico: (1) alocação de memória na GPU; (2) transferência de dados da CPU para a GPU; (3) execução do *kernel* e (4) desalocação de memória. A função *kernel*, na etapa três, é invocada em cada etapa de *rendering*. Nesta etapa, um *buffer* é registrado na GPU, usando as funções do CUDA SDK. Ao utilizar este *buffer*, é possível manter o resultado computado dentro da GPU para que este resultado possa estar diretamente disponível para as etapas de *rendering*, sem a necessidade de uma comunicação extra com a CPU.

A transferência de dados e as funções de alocação de GPU são chamadas antes do estágio de *rendering*. Outros conjuntos de dados podem ser transferidos sem custos adicionais se eles possuírem o mesmo tamanho da área alocada. Caso não possuam o mesmo tamanho, é necessário executar os passos quatro, um e dois em sequência se outros tamanhos são considerados.

5.3 Etapas para a execução do *kernel*

Conforme comentado anteriormente, o algoritmo foi subdividido em quatro etapas, implementadas como quatro funções. Estas etapas são mostradas e explicadas abaixo.

5.3.1 Alocação de memória na GPU

Na aplicação principal, são criadas e lidas todas as texturas. Foram gerados quatro espaços de texturas: a textura inicial, a final, a textura interpolada, que a princípio foi inicializada recebendo a textura inicial e, finalmente a quarta textura que é uma textura de cor cinza que será utilizada pelo *bump mapping*.

Para a implementação do *bump mapping* foi utilizado a multi-textura, sendo assim, foram necessárias duas texturas: a textura de cor cinza (quarta textura) e o mapa de normal (terceira textura). A terceira textura, que é na verdade um espaço de textura que recebe o resultado da interpolação, foi criada e ativada como GL_TEXTURE1. Vale a pena lembrar que o resultado do *morphing* é um mapa de normal. E a quarta textura foi criada e ativada como GL_TEXTURE0.

Depois que todas as texturas foram criadas e ativadas, é necessário alocar memória na GPU para as mesmas. Para isso é necessário utilizar a função *cudamalloc*.

Esta função, executada pela CPU, foi utilizada para alocar memória na GPU para a textura inicial, final, linhas guias, para a variável v (esta variável v representa a distância do ponto X ao ponto que corta o vetor \overrightarrow{PQ} através de uma linha perpendicular a esse vetor) e para a variável u (a variável u representa a distância medida pelo ponto P até onde a linha perpendicular que passa pelo ponto X corta o vetor).

5.3.2 Transferência de dados da CPU para a GPU

Depois da alocação de memória na GPU para as imagens, inicial e final, e linhas guias, a GPU já se encontra pronta e com memória alocada para receber esses dados. O passo seguinte consiste em transferir os dados que estão na CPU para a GPU. Esta etapa também é executada pela CPU.

5.3.3 Execução do Kernel

O *kernel* se encarrega de executar o *morphing* propriamente dito, pois ele possui o código que será executado em paralelo pela GPU. Para que a execução do *kernel* comece, é necessário que todas as *matrizes* contendo as texturas e as linhas guias já tenham sido transferidas da CPU para a GPU.

Com essas matrizes e com os dados das texturas na GPU, inicia-se a execução do *kernel*. O *kernel* é responsável por ler os dados das texturas, os dados das linhas guias, calcular as distâncias, realizar a interpolação entre a textura inicial e a textura final e carregar o resultado da interpolação em uma matriz temporária para que este resultado possa estar diretamente disponível para as etapas de *rendering*. Todos os cálculos necessários para a interpolação entre as texturas são realizados em paralelo.

O resultado da interpolação, que é carregada em uma matriz temporária, é enviado para um *buffer* para que ele fique dentro da GPU. Desta maneira, o resultado fica diretamente disponível para as etapas de *rendering*.

O *kernel* foi dividido em três funções distintas. No final de cada uma dessas funções, as *threads* são sincronizadas.

A primeira função, chamada de *MakeInterpolationGPUKernel*, é responsável pelo cálculo da interpolação. São passadas como parâmetros as linhas guias de cada textura (textura A e textura B) e a taxa de transição, que varia de 0 a 1.

A segunda função, chamada de *MakeMorphiGPU0*, é responsável por calcular os valores das variáveis u e v . Esta função é subdividida em duas sub-funções: *FindU* e *FindV*. A sub-função *FindU* é responsável por encontrar a distância medida pelo ponto P até onde a linha perpendicular que passa pelo ponto X corta o vetor. Já a sub-função *FindV* é responsável por encontrar a variável v que é a distância do ponto X ao ponto que corta o vetor \overrightarrow{PQ} através de uma linha perpendicular a esse vetor, representando a distância na direção perpendicular do vetor ao ponto.

Por fim, a terceira função, chamada de *MakeMorphiGPU1*, é responsável por calcular a nova localização do *pixel* X.

5.3.4 Desalocação de memória

Finalmente a última etapa, desalocação de memória. Esta etapa é responsável por liberar a memória, ou seja, liberar a GPU invocando o comando *free* sobre as texturas e linhas guias.

5.4 Passagem do resultado do contexto CUDA para o contexto gráfico

Como dito anteriormente, as texturas e as suas linhas guias são transferidas para a GPU antes de o processo começar e permanecem lá até que todo o processamento seja feito, evitando assim, que dados sejam transferidos desnecessariamente, pois a cada novo *frame* a textura produzida é enviada diretamente para o estágio de render sem sair da placa gráfica.

O *kernel* é invocado a cada etapa do *rendering* e um *buffer* é registrado na GPU, usando as funções do CUDA SDK, para receber o resultado computado pelo CUDA. Ao utilizar este *buffer*, é possível manter o resultado computado dentro da GPU para que este resultado possa estar diretamente disponível para as etapas de *rendering*. Dessa forma, o resultado armazenado neste *buffer* é aplicado como multi-textura na malha do rosto, ativado como *texture1* e enviado para o *shader* como um *sampler*. É neste momento que ocorre a integração dos algoritmos de *morphing* e de *bump mapping*.

Com o resultado enviado para o *shader* é iniciada a execução do *bump mapping*. A Figura 5.2 mostra o diagrama da passagem do resultado do contexto CUDA para o contexto gráfico.

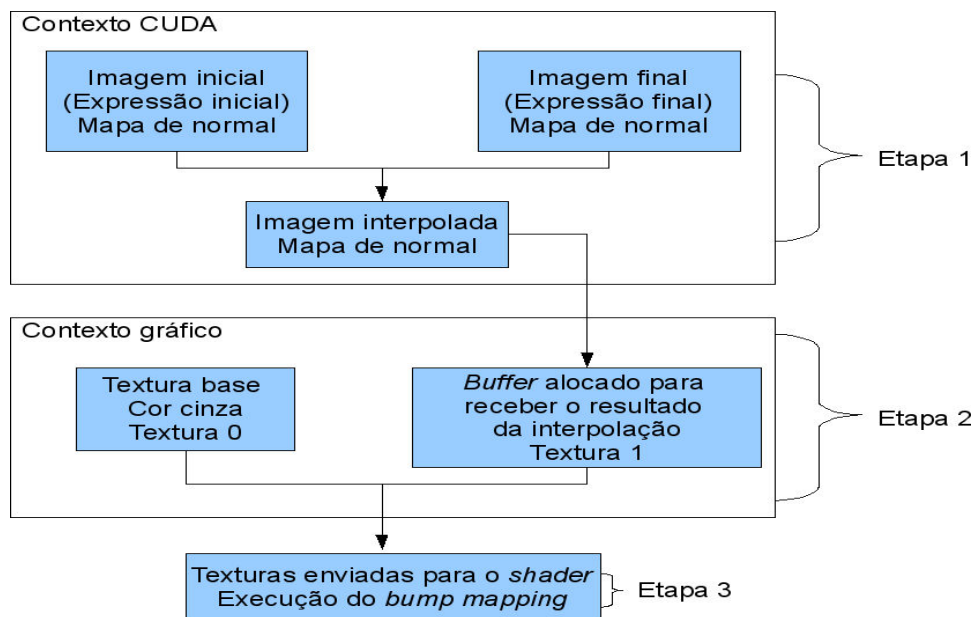


Figura 5.2: Diagrama da passagem do resultado do contexto CUDA para o contexto gráfico.

5.5 Bump mapping na GPU

O *bump mapping* será utilizado em cada etapa da visualização para simular as rugas, ou seja, as rugas das expressões faciais serão colocadas como *bump mapping* sobre a superfície da face.

Existem quatro etapas para implementar o *bump mapping*, as quais são descritas a seguir (DREIJER, 2010):

1. Calcular o vetor da luz e transformar o mesmo do espaço objeto para o espaço tangente;
2. Ler o vetor normal do mapa de normal e converter do intervalo de $[0,1]$ para $[-1,1]$;
3. Calcular a cor difusa final através do produto escalar entre o vetor de luz e a normal e multiplicar com a cor da luz e da cor do material da superfície;
4. Repetir o processo para todos os pixels da superfície dada.

Para a implementação do *Bump Mapping* foi utilizada a linguagem GLSL (*OpenGL Shading Language*). GLSL é uma extensão da linguagem OpenGL. Foi introduzida ao padrão OpenGL na especificação 2.0 sendo utilizada para o desenvolvimento de *shaders*. O GLSL é baseada na linguagem C, pois possuem a mesma sintaxe. Alguns recursos do

C foram modificados e novos tipos de dados foram acrescentados para permitir a programação de GPU (*Graphics Processing Units* (EVANGELISTA e SILVA, 2007)).

5.5.1 Os *shaders*

Nesta seção são apresentados o *vertex shader* e o *fragment shader* do *bump mapping*. O *vertex shader* é responsável pela conversão do vetor de luz do espaço objeto para o espaço tangente. É também neste *shader* que a posição corrente do vértice é transformada do espaço objeto para o *clip space*, através da multiplicação das matrizes *modelview* e *projection* com a posição do vértice. Abaixo é mostrado o *vertex shader* do *bump mapping*.

```
attribute vec3 tangent;
attribute vec3 binormal;
varying vec3 position;
varying vec3 lightvec;

void main()
{
    // Transforma o vértice corrente do espaço objeto para o clip space
    gl_Position = ftransform();
    gl_TexCoord[0] = gl_TextureMatrix[0] * gl_MultiTexCoord0;
    position = gl_Vertex.xyz;
    mat3 TBNMatrix = mat3(tangent, binormal, gl_Normal);

    // Calcula o vetor de luz
    lightvec = gl_LightSource[0].position.xyz - position;

    // Transforma o vetor de luz do espaço objeto para o espaço tangente
    lightvec *= TBNMatrix;
}
```

O *fragment shader* para este trabalho é dado pelo código abaixo.

```
uniform sampler2D imgSemBump;
uniform sampler2D mapaNormal;
uniform vec3 CAMERA_POSITION;

varying vec3 position;
varying vec3 lightvec;

void main()
{
    //As normais do mapa de normal está no intervalo de [0, 1]. É necessário que as
    //normais estejam no intervalo de [-1,1]. Transformando ([0, 1] → [-1, 1])
    vec3 norm = normalize(texture2D(mapaNormal, gl_TexCoord[0].st).rgb * 2.0 - 1.0);
    vec3 baseColor = texture2D(imgSemBump, gl_TexCoord[0].st).rgb;

    //Quando se normaliza o vetor de luz, o vetor é interpolado linearmente através da
    //superfície. Isso faz com que o comprimento do vetor será alterado à medida que
    //ele é interpolado.
    float dist = length(lightvec);
    vec3 lightVector = normalize(lightvec);
    float nxDir = max(0.0, dot(norm, lightVector));

    //Calculo da componente difusa
    vec4 diffuse = gl_LightSource[0].diffuse * nxDir;
    float specularPower = 0.0;

    if (nxDir != 0.0)
    {
        vec3 cameraVector = normalize(CAMERA_POSITION - position.xyz);
        vec3 halfVector = normalize(lightVector + cameraVector);
        float nxHalf = max(0.0, dot(norm, halfVector));
        specularPower = pow(nxHalf, gl_FrontMaterial.shininess);
    }
}
```

```
vec4 specular = gl_LightSource[0].specular * specularPower;  
gl_FragColor = gl_LightSource[0].ambient + (diffuse * vec4(baseColor.rgb, 1.0)) + specular;  
}
```

Capítulo 6

Resultados

Nesta seção são mostradas as etapas de modelagem, de animação facial e os resultados obtidos da simulação de emoções utilizando a técnica proposta. A representação de atributos como a utilização de texturas, também será mostrada. A etapa de modelagem consiste da criação do modelo 3D utilizado e a etapa de animação consiste da escolha do algoritmo utilizado para a animação.

6.1 Modelagem

Como dito anteriormente, essa etapa corresponde à criação do modelo 3D utilizado na simulação. A malha 3D, desenvolvida pelo artista Fernando Ribeiro, foi gerada no *Blender* e exportada para o formato de dados 3D *OBJ*. A malha contém 234 vértices, 249 coordenadas de textura e 222 faces. A Figura 6.1 ilustra a malha 3D.

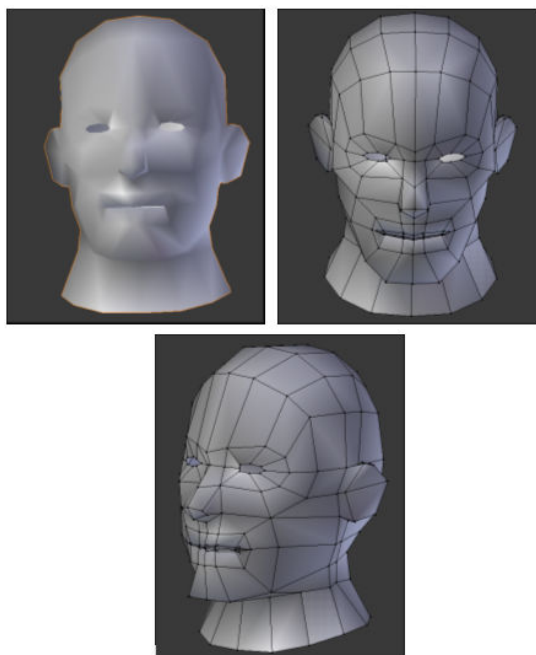


Figura 6.1: Malha 3D criada pelo artista Fernando

O sistema desenvolvido lê o arquivo *OBJ*, carrega e renderiza a malha. O formato OBJ é um arquivo simples, que contém os vértices, as coordenadas de textura e as faces dos polígonos.

6.2 Animação facial

A etapa de animação facial é a etapa onde é realizada a transição entre duas expressões faciais de forma que esta transição seja realista. Uma vez que a face foi modelada, o passo seguinte consiste em sua animação. Como visto no Capítulo 2, existem várias técnicas para a animação facial: interpolação, baseada em *performance*, parametrização direta, animação baseada em pseudomúsculos, animação baseada em músculos e simulação biomecânica. Este trabalho não trata da animação extrínseca da malha, mas apenas acrescenta detalhes nas rugas. Qualquer outra técnica de animação de malha pode vir a ser usada em conjunto com a que propomos.

As texturas utilizadas pelo *morphing* são os mapas de normais que contém os detalhes das rugas de cada expressão. Cada expressão chave é modelada por um mapa de normais específico.

As texturas também foram feitas pelo artista Fernando Ribeiro. O formato escolhido foi o PPM e o tamanho é de 256x256 pixels. As Figuras 6.2 e 6.3 mostram o mapa de normal das expressões de bravo e de assustado. Repare as rugas na testa e as sobrancelhas em ambos os mapas de normal.

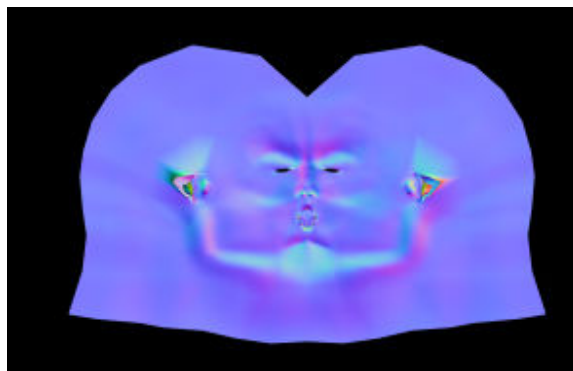


Figura 6.2: Mapa de normal da expressão de bravo

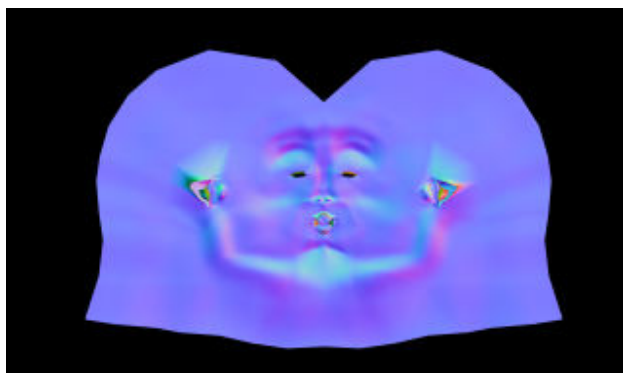


Figura 6.3: Mapa de normal da expressão de assustado

Para a especificação dos pontos de controle, foram utilizados os *FAPs* do padrão MPEG-4. Este padrão propõe uma definição de parâmetros de controle para uma animação facial. Os *FAPs* são os pontos que representam um ponto característico e que são responsáveis pelo movimento deste ponto na face, produzindo a animação facial. A interpolação é feita utilizando esses pontos para a animação. Conforme o padrão MPEG-4 propõe, foram especificados 84 pontos de controle que são utilizados pelo *morphing* para a animação facial. Estes pontos foram especificados em um arquivo texto, que é lido pelo *morphing*.

6.3 Máquina utilizada

Para essa simulação foram utilizadas três máquinas diferentes com a seguinte configuração: Na máquina 1 foi utilizada uma placa de vídeo "GeForce GTX 480", com memória global de 1GB.

Na máquina 2 foi utilizada uma placa de vídeo "GeForce 9800 GTX/9800 GTX+", memória global de 512Mb e 16 multiprocessadores.

Já na máquina 3 foi utilizada uma placa de vídeo "GeForce 8800 GTS", com memória global de 320Mb e com 12 multiprocessadores.

6.4 Resultados

Para testar a técnica proposta, foi feita uma simulação da transição da expressão de raiva para a expressão de surpresa. A expressão de raiva corresponde à imagem inicial (I_0) e a expressão de surpresa à imagem final (I_1).

A figura abaixo mostra os resultados da simulação para cada taxa (t) de transição, sendo que esta taxa varia de 0 a 1. Para t perto de zero, as imagens intermediárias se parecem com a expressão facial da textura original. Na medida em que se aumenta este parâmetro, a face adquire expressões intermediárias e não totalmente previstas e modeladas pelo artista.

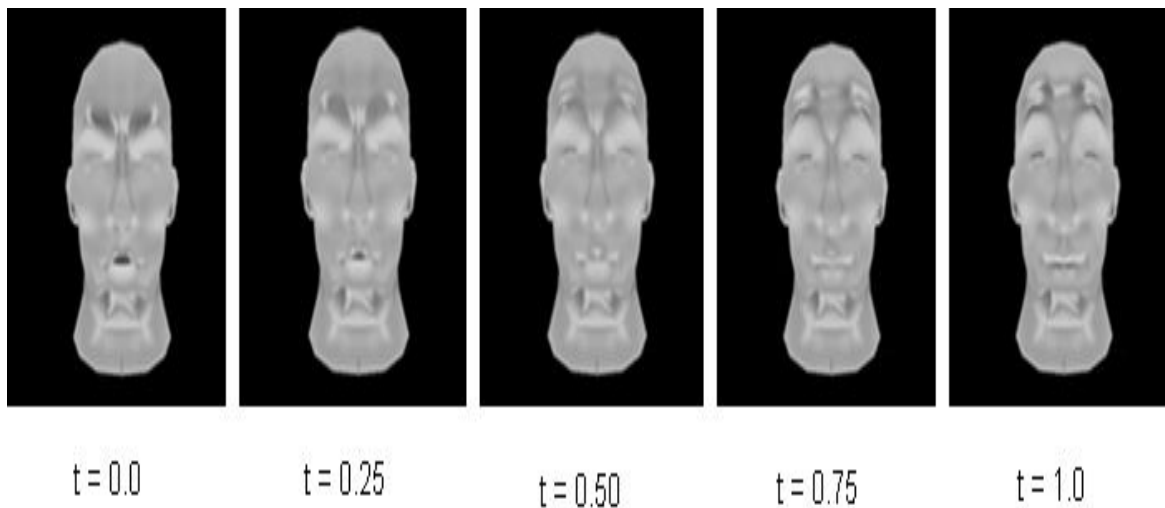


Figura 6.4: Resultado da simulação para cada taxa (t) de transição

Durante a simulação, foram medidos os tempos gastos em cada etapa do *kernel* e o tempo gasto com o *bump mapping* para cada máquina utilizada. A Tabela 6.1 mostra estes tempos na primeira máquina, a Tabela 6.2 mostra o tempo gasto na segunda máquina e a Tabela 6.3 mostra o tempo gasto na terceira máquina.

Etapa do Kernel	Tempo gasto em ms
[1] Alocação de memória na GPU	11,67 ms
[2] Transferência de dados da CPU para a GPU	0,90 ms
[3] Execução do <i>kernel</i>	31,81 – 32,05 ms
<i>Bump mapping</i>	0,03 ms
Tempo Total	44,41 ms

Tabela 6.1: Tempo gasto em cada etapa do *kernel* e o tempo gasto com o *bump mapping* na máquina 1

Etapa do Kernel	Tempo gasto em ms
[1] Alocação de memória na GPU	16,17 ms
[2] Transferência de dados da CPU para a GPU	0,58 ms
[3] Execução do <i>kernel</i>	96,56 ms
<i>Bump mapping</i>	0,1 ms
Tempo Total	113,41 ms

Tabela 6.2: Tempo gasto em cada etapa do *kernel* e o tempo gasto com o *bump mapping* na máquina 2

Etapa do Kernel	Tempo gasto em ms
[1] Alocação de memória na GPU	1,72 ms
[2] Transferência de dados da CPU para a GPU	0,51 ms
[3] Execução do <i>kernel</i>	429 - 442 ms
<i>Bump mapping</i>	0,45 ms
Tempo Total	431,68 ms

Tabela 6.3: Tempo gasto em cada etapa do *kernel* e o tempo gasto com o *bump mapping* na máquina 3

Capítulo 7

Conclusão e trabalhos futuros

Este trabalho apresentou o sistema de animação facial desenvolvido para simular expressões faciais em tempo real. Foram mostradas as técnicas utilizadas e os algoritmos implementados. Foi apresentada também uma estratégia que utiliza rugas para reforçar as expressões faciais tradicionais, o método de *morphing* baseado em GPUs para gerar imagens interpoladas em tempo real e uma estratégia para otimizar a troca de contexto CUDA para contexto gráfico, minimizando o tráfego de dados entre GPU e CPU.

De acordo com os resultados obtidos, foi possível perceber a animação das rugas através da simulação da transição da expressão de raiva para a expressão de surpresa. A simulação apresentou expressões detalhadas e realistas utilizando um número pequeno de polígonos. Logo, com a utilização do *bump mapping*, as rugas foram criadas, e com a utilização do algoritmo de *morphing*, as rugas foram animadas.

Para que a simulação fosse realista foi realizado um estudo sobre emoções e expressões faciais. Foram estudadas também as características de cada expressão facial, ou seja, onde as rugas se formam para cada expressão.

A estratégia utilizada para otimizar esta troca de contexto CUDA para o contexto gráfico foi possível por causa da utilização do *buffer* que foi registrado na GPU. Este *buffer* recebe o resultado computado pelo CUDA e este resultado permanece dentro da GPU, estando diretamente disponível para as etapas de *rendering*.

Durante a execução da simulação, o tempo gasto em cada etapa do *kernel* foi medido. De acordo com a análise dos tempos obtidos em cada etapa, percebe-se que a etapa 3, Execução do *kernel*, foi a que consumiu mais tempo para ser executada. Sendo assim, tem-se um gargalo neste ponto. Logo, este trabalho pode ser aprimorado no que diz respeito ao tempo gasto para a execução do *kernel*, pois esta etapa pode ser otimizada, melhorando assim, o desempenho. Uma possível solução para esse problema seja não alterar a posição de todos os pixels, e sim, somente daqueles que deverão ter a sua posição alterada.

Alguns trabalhos futuros são sugeridos abaixo:

- Exportar o resultado obtido na simulação para uma *engine* de jogo, para que a transição gradual de uma emoção em outra, seja aplicado nas faces dos personagens.

- Otimização do *kernel* proposto neste trabalho, visto que ele consumiu bastante tempo para ser executado.
- Utilização do sistema proposto para a animação de rugas em roupas.

Capítulo 8

Referências bibliográficas

- [Adamatti, 2003] Adamatti, D. F. *Afrodite: Ambiente de simulação baseado em agentes com emoções*. 2003. Dissertação de Mestrado. Universidade Federal do Rio Grande do Sul.
- [Akenine-Moller et al, 2008] Akenine-Moller, T., Haines, E. e Hoffman, N.; *Real-Time Rendering*, Third Edition, AK Peters, Massachusetts. 2008
- [Amoroso, 2008] Amoroso, D.; *O que é Pixel Shader*. 2008. Disponível em <http://www.baixaki.com.br/info/811-o-que-e-pixel-shader-.htm> Acesso em 15/02/2010
- [Arad and Reisfeld, 1995] Arad, N.; Reisfeld, D. *Image warping using few anchor points and radial functions*. Computer Graphics Forum, v. 14, n. 1, p. 35–46, 1995.
- [Baptista, 2005] Baptista, S. P. *Estudo Preliminar de Animação Facial Realista para o Ensino de Deficientes Auditivos*. 2005. Dissertação de Mestrado. Universidade Federal Fluminense
- [Bayer, 2010] Bayer Schering Pharma. *O que é a pele?* Disponível em: <http://www.bayerscheringpharma.com.br/site/homem/dermatologia/oqueeapele.fss> Acesso em 27/09/2010.
- [Beier and Neely, 1992] Beier, T.; Neely, S. *Feature-based image metamorphosis*. In: SIGGRAPH '92: Proceedings of the 19st annual conference on Computer graphics and interactive techniques. New York, NY, USA: ACM Press, 1992. v. 26, n. 2, p. 35–42. ISSN 0097-8930.
- [Bergeron and Lachapelle, 1985] Bergeron P. e Lachapelle P. (1985). *Controlling facial expressions and body movements in the computer generated animated short 'Tony de Peltrie'*. In SigGraph '85 Tutorial Notes, Advanced Computer Animation Course..
- [Blinn, 1978] Blinn, J. F. *Simulation of Wrinkled Surfaces*, In Proceedings SIGGRAPH 78, pp. 286-292, 1978
- [Capovilla, 2006] Capovilla, G. H.; *Ambiente de visualização e interação para simulações neurais*. Dissertação de mestrado. Universidade Metodista de Piracicaba. 2006
- [Carvalho et al, 2009] Carvalho, A., Farias, T. Teixeira, J. M., Teichrieb, V. Kelner, J. *Aplicando Model-Driven Development à Plataforma GPGPU*. In: WSCAD-SSC 2009: X SIMPÓSIO EM SISTEMAS COMPUTACIONAIS. 2009
- [Carvalho, 2010] Carvalho, B. M.; *Iluminação e Sombreamento*. Universidade Federal do Rio Grande do Norte. 2010

- [Castro, 2004] Castro, T. A. L.; *Realidade virtual e engenharia civil: Detecção de interferências entre projetos de edificações em 3D*. Dissertação de mestrado. 2004
- [Clua, 2010] Clua, E. W. G.; *Arquitetura de GPU's*. Disponível em <http://www.ic.uff.br/~esteban>. Acesso em 15/02/2010
- [Conci and Azevedo, 2003] Conci, A.; Azevedo, E.; *Computação Gráfica: Teoria e Prática*, Elsevier. 2003
- [Damásio, 1996] Damásio, A. *O erro de descartes: emoção, razão e o cérebro humano*. São Paulo: Companhia da Letras, 1996.
- [Damásio, 2003] Damásio, A., *Ao Encontro de Espinoza*, trad. P.E.A., Pub. Europa-América, Lisboa, 2003
- [Darsa, 1994] Darsa, L. *Deformação e metamorfose de objetos gráficos*. Dissertação de Mestrado. PUC-Rio. 1994
- [Darwin, 1872] Darwin, C., *The Expression of the Emotions in Man and Animals*. D. Appleton and Company: New York (1872)
- [Ekman, 1992] Ekman, P., *An argument for basic emotions*. Cognition and Emotion, 6, 1992
- [Ekman, 1993] Ekman, P., *Facial expression of emotion*. American Psychologist, 48, 384-392, 1993
- [Ekman, 1999] Ekman, P., *Basic Emotions*. In Dalglish, T. and Power, M., (Eds.), *Handbook of Cognition and Emotion*, Sussex, U.K., 1999
- [Faustino et al, 2006] Faustino, J., Cláudio, A. P., Carmo, M. B., *Faces - Biblioteca de Expressões Faciais*. In: II Conferência Nacional em Interação Pessoa-Máquina, Interação 2006, artigo curto, pp 139-142, Braga, 16-18 de Outubro de 2006.
- [Fernandes and Santos, 2009] Fernandes, A. R.; Santos, L. P.; *Transformações geométricas*. Universidade do Minho. 2009
- [Frijda, 1986] Frijda, N.H., *The Emotions, Studies in Emotion and Social Interaction*. 1986. Cambridge University Press, Cambridge.
- [Frezza et AL, 2009] Frezza, R.; Silva, A.; Lux, B.; Cruz, M.; Borin, M. *Dóris 3D: Agente Pedagógico baseado em Emoções*. In: XX Simpósio Brasileiro de Informática na Educação, 2009, Florianópolis. XX Simpósio Brasileiro de Informática na Educação. Florianópolis: Universidade Federal de Santa Catarina, 2009. p. 1-10.
- [Gamedev, 2010] Gamedev. *Pixel Shader no XNA*. Disponível em: <http://www.gamedevbr.com/tags/programacao/> Acesso em 15/02/2010
- [Heavyrainps3, 2010] Heavy Rain. Disponível em: <http://www.heavyrainps3.com/> Acesso em 23/10/2010

- [Higa, 2008] Higa, R. S. *Uma proposta de morphing utilizando técnicas de interpolação de formas e média morfológica*. Dissertação de Mestrado. Unicamp. 2008
- [James, 1884] James, William. *What is an Emotion?* First published in *Mind*, 9, 188-205, (1884), Disponível em: <<http://psychclassics.asu.edu/James/emotion.htm>> Acessado em 25/09/2010
- [Jaques and Vicari, 2005] Jaques, P.; Vicari, R. M. *Estado da Arte em Ambientes Inteligentes de Aprendizagem que Consideram a Afetividade do Aluno*. Informática na Educação, UFRGS: Porto Alegre, v. 8, n. 1, p. 15-38, 2005.
- [JogosGamers, 2010] JogosGamers. *Os melhores graficos dos Games - Especial Gold - 01*. Disponível em < <http://www.jogosgamers.com/2010/03/os-melhores-graficos-dos-game-especial.html> > Acesso em 23/10/2010.
- [Jouvier, 2010] Jouvier, J.; *Tangent Space*. Disponível em <http://jerome.jouvie.free.fr/OpenGL/Lessons/Lesson8.php> Acesso em 15/02/2010
- [Körting, 2007] Körting, T. S. *Animador tridimensional de faces*. Dissertação de mestrado. Universidade Federal do Rio Grande. Engenharia de Computação. 2007
- [Lessa, 2008] Lessa, J. *A importância das emoções na comunicação interpessoal mediada por Tecnologia*. 2008. Encuentro Latinoamericano de Diseño. Argentina.
- [Lima and Braun, 2008] Lima, D. S.; Braun, H.; *Exibição de terrenos em tempo real: Uma abordagem a terrenos com larga escala geométrica*. Dissertação de mestrado. Pontifícia Universidade Católica do Rio Grande do Sul. 2008
- [Lucena, 2002] Lucena, P. S. *Expressive Talking Heads: Um Estudo de Fala e Expressão Facial em Personagens Virtuais*. 2002. Dissertação de Mestrado. DI/PUC-Rio, Rio de Janeiro.
- [Marin, 2006] Marin, L. O. *Métodos estatísticos no reconhecimento de faces*. RESI-Revista Eletrônica de Sistemas de Informação, Edição 8, No2, 2006. Universidade Federal de Santa Catarina.
- [Martini, 2006] Martini, M. *Proposta da modelagem de faces virtuais baseadas em um modelo 3D genérico*. 2006. Monografia. Centro Universitário Feevale.
- [Moreti, 2006] Moreti, M. P. *Animação facial*. 2006. Monografia apresentada ao Instituto de Matemática e Estatística – IME.
- [Neto, 2009] Neto, J. R. P.; *Framework para geração de personagens humanóides 3D para jogos computacionais*. 2009
- [Nogueira et al, 2003] Nogueira, A., Prata, E., Velho, L. *Visualização de Objetos Tridimensionais Baseada em Interpolação Projetiva*. In Proceedings of I Workshop de Trabalhos de Iniciação Científica em Computação Gráfica e Processamento de Imagens, 2003. WICCGPI'2003.

- [Nvidia, 2010] Nvidia. *Cuda – Computed Unified Device Architecture – programming guide. Technical report*. NVIDIA, 2010
- [Oliveira, 2010] Oliveira, D. T.; *Uma abstração de alto nível para programação do processador gráfico*. Dissertação de mestrado. Pontifícia Universidade Católica do Rio de Janeiro. 2003. Disponível em <http://www.ideais.com.br/XMLTalk/site/pub/tese3.pdf> Acesso em 15/02/2010
- [Paul, 2009] Paul.; *Simple Bump Mapping: Rotating, diffuse lit torus*. Disponível em: <http://www.paulsprojects.net/tutorials/simplebump/simplebump.html> Acesso em: 14/02/2010 Pozzer, C. T.; *OpenGL Shading Language*. 2009
- [Rodrigues, 2005] Rodrigues, P. S. L., *Modelos Avancados de Animacao Facial: O Estado da Arte*. 2005. Relatório técnico. Pontifícia Universidade Católica do Rio de Janeiro.
- [Rodrigues, 2007] Rodrigues, P. S. L. *Um sistema de geração de expressões faciais dinâmicas em animações faciais 3D com processamento de fala*. Tese de doutorado. Pontifícia Universidade Católica do Rio de Janeiro. 2007
- [Rosário, 2008] Rosário, S. V. P. G. *Facial Emoticons: Reprodução de Informação Associada a Expressões Faciais por Via do seu Reconhecimento*. Dissertação de mestrado. Universidade Técnica de Lisboa. 2008
- [Salgado, 2006] Salgado, A. V.; *Simulação visual em tempo real de ondas oceânicas utilizando a GPU*. Dissertação de mestrado. Universidade Federal Fluminense. 2006
- [Sangoi, 2008] Sangoi, S. A. *Aumento do realismo na dramatização de storytelling com a criação de personagens que demonstrem emoções através de expressões faciais*. 2008. Monografia. Universidade Federal de Santa Maria.
- [Santos, 2007] Santos, B. S.; *Modelos de iluminação: Métodos de surface-rendering*. Universidade de Aveiro. 2007
- [Santos, 2009] Santos, M. C. *Renderização de cenas tridimensionais interativas em computadores com recursos gráficos limitados*. Dissertação de Mestrado. Universidade Federal do Paraná. 2009
- [Schroeder, 2007] Schroeder, G. N. *Morphing aplicado ao envelhecimento de imagens faciais*. Dissertação de Mestrado. Unicamp. 2007
- [Silva, 1994] Silva, B. C. *Deformação e metamorfose de imagens digitais*. Dissertação de Mestrado. PUC-Rio. 1994
- [Souza, 2010] Souza, G. S.; *Espectro de luz e modelos de reflexão*. 2010
- [Tonietto, 2010] Tonietto, L.; *Iluminação e Sombreamento*. Disponível em <http://www.inf.unisinos.br/~ltonietto/jed/cgr/iluminacao.pdf>. Acesso em 13/02/2010
- [Villar, 2010] Villar, J. R.; *OpenGL Shading Language Course. Chapter 4 – Advanced Shaders*. Disponível em: http://www.opengl.org/sdk/docs/tutorials/TyphoonLabs/Chapter_4.pdf

Acesso em 15/02/2010

[Wangenheim, 2010] Wangenheim, A. V; *Visualização reslística em 3D, z-buffering e raytracing*. Disponível em <http://www.inf.ufsc.br/~awangenh/CG/raytracing/iluminacao.html>. Acesso em: 13/02/2010

[Waters, 1987] WATERS, Keith.; *A muscle model for animation threedimensional facial expression*. SIGGRAPH'87 Proceedings of the 14th annual conference on computer graphics and interactive techniques, ACM Press, 1987, p.1724.

[Wey, 1999] Wey, J. D. R. *Interface: Um sistema de animação interativa de rostos humanos*. 1999. Dissertação de Mestrado. Escola Politécnica da Universidade de São Paulo.

[Wolberg, 1994] WOLBERG, G. *Digital Image Warping*. Los Alamitos, CA, USA: IEEE Computer Society Press, 1994. ISBN 0818689447.

[Zagalo, 2001] Zagalo, N., Branco, V., Barker, A., 2004, *Emoção e Suspense no Storytelling Interactivo*, in Actas, Games2004 - Workshop Entretenimento Digital e Jogos Interactivos, Lisboa, Portugal, (pp.75-84) - Best paper prize

[Zamith, 2009] Zamith, M., Clua, E. W. G., Montenegro, A., PASSOS, E., LEAL, R., CONCI, A. *Real time feature-based parallel morphing in GPU applied to texture-based animation*. In: 16th International Workshop on Systems, Signals and Image Processing, 2009, Chalkida, Grécia. IEEE Proceedings of the 16th International Workshop on Systems, Signals and Image Processing. London : IEEE - Region 8, 2009. v. 16. p. 145-150.

[Zanella and Fuentes, 2004] Vittorio Zanella and Olac Fuentes, *An Approach to Automatic Model-Based Morphing of Face Images in Frontal View*, Proceedings of 2004 Mexican International Conference on Artificial Intelligence (MICAI), Mexico City, Mexico, Lecture Notes in Artificial Intelligence 2972, April 2004.