

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

PEDRO HENRIQUE VIEIRA DE OLIVEIRA
AZEVEDO

UM SISTEMA DE REALIDADE AUMENTADA
PARA A VISUALIZAÇÃO DE OBJETOS
VIRTUAIS FACIAIS UTILIZANDO CÂMERAS
RGB-D

VITÓRIA-ES
2016

**PEDRO HENRIQUE VIEIRA DE OLIVEIRA
AZEVEDO**

**UM SISTEMA DE REALIDADE AUMENTADA
PARA A VISUALIZAÇÃO DE OBJETOS
VIRTUAIS FACIAIS UTILIZANDO CÂMERAS
RGB-D**

Dissertação apresentada ao Programa de Pós-Graduação em Informática do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para a obtenção de Título de Mestre em Informática.

Orientador: Prof. Dr. Edilson de Aguiar

Co-orientador: Prof. Dr. Thiago Oliveira dos Santos

VITÓRIA-ES

2016

Para fazer a ficha catalográfica, que ficará no lugar desta página, na versão final, ligue ou escreva para Arlete Franco

Bibliotecária/BC/UFES

franco@bc.ufes.br

27 4009-2405

Há um formulário específico para solicitar tal ficha, que é enviado à bibliotecária via e-mail

(ver www.ele.ufes.br/paginappgee.html)

PEDRO HENRIQUE VIEIRA DE OLIVEIRA AZEVEDO

**UM SISTEMA DE REALIDADE AUMENTADA PARA A
VISUALIZAÇÃO DE OBJETOS VIRTUAIS FACIAIS
UTILIZANDO CÂMERAS RGB-D**

Dissertação submetida ao programa de Pós-Graduação em Informática do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para a obtenção do Grau de Mestre em Informática.

Aprovada em 23 de Setembro de 2016.

COMISSÃO EXAMINADORA

Prof. Dr. Edilson de Aguiar
Universidade Federal do Espírito Santo
Orientador

Prof. Dr. Thiago Oliveira dos Santos
Universidade Federal do Espírito Santo
Co-orientador

Prof. Dr. Alberto Ferreira de Souza
Universidade Federal do Espírito Santo

Prof. Dr. Gilson Antônio Giraldi
Laboratório Nacional de Computação Científica

A Deus, pela graça de conseguir concluir este trabalho.

Aos meus pais, Ailton e Vanderléa, pelos inesquecíveis gestos de incentivo aos meus estudos e por todo o apoio financeiro para esta jornada.

Ao meu irmão, Gustavo, pela torcida para a conclusão deste trabalho.

À minha noiva, Ramoni, meu presente, por estar sempre ao meu lado e pelas incontáveis ajudas na realização deste trabalho.

Agradecimentos

Para não correr o risco da injustiça, agradeço de antemão a todos que de alguma forma passaram pela minha vida e contribuíram para a construção de quem sou hoje.

E agradeço, particularmente, à algumas pessoas pela contribuição direta na construção deste trabalho:

- À Universidade Federal do Espírito Santo (UFES), por me oferecer um aperfeiçoamento gratuito e de qualidade.
- Aos professores Edilson de Aguiar e Thiago Oliveira dos Santos, pela orientação ao longo deste caminho.
- Aos tios Carlos e Luciana por serem os grandes apoiadores da minha vida capixaba e pelos felizes e divertidos momentos vividos na Cob. 01.
- À todos os meus familiares e amigos de Petrópolis que de alguma forma deram uma palavra de incentivo ou apoio.
- Aos amigos e colegas do LCAD pelos cafés, bolos e conversas que tanto ajudaram a amenizar a jornada.
- Aos amigos Alexandro Ramos e Marcelo Carrion por toda a amizade e ajuda demonstrada durante o mestrado e a todos os colegas do LabOtim.
- Aos amigos-irmãos Daniel Gaspar e Fabrício Vilasboas por todo o companherismo, força e ajuda, apesar de toda a distância.
- Aos amigos para sempre Bruna, Bárbara, Dudu, Léo e Thiago por sempre estarem tão presentes nessa e em outras vivências.

- Aos professores do PPGI, que com seus conhecimentos compartilhados em sala de aula, ajudaram a completar o conhecimento necessário para a conclusão deste trabalho.
- À todas as pessoas que ajudaram na elaboração do banco de faces para este trabalho.

*“Você pode sempre sonhar,
e seus sonhos se tornarão realidade,
mas é você que tem que torná-los realidade.”*

(Michael Jackson)

Publicações

Como parte deste trabalho, foi desenvolvido e publicado o seguinte artigo, no período do mestrado.

Publicação de trabalho em anais de congresso:

- AZEVEDO, P.; SANTOS, T. O. ; DE AGUIAR, E. . **An Augmented Reality Virtual Glasses Try-On System**. 18th Symposium on Virtual and Augmented Reality, 2016, Gramado. XVIII Symposium on Virtual and Augmented Reality, 2016.

Resumo

Este trabalho apresenta um sistema para a correta visualização de objetos 3D, por exemplo, um óculos, na face de um usuário. Através da captura de uma imagem e das informações de profundidade de um usuário por uma câmera de baixo custo RGB-D, o Microsoft Kinect, nós aplicamos uma técnica de rastreamento facial que detectam pontos de interesse específicos na face. Esses pontos de interesse e a nuvem de pontos, reconstruídos através das informações de profundidade, são combinados para otimizar um modelo morfológico facial 3D para que este se ajuste o melhor possível à face do usuário. No final, nós otimizamos o objeto virtual 3D escolhido de maneira que ele se ajuste à forma do rosto do usuário. O último passo é projetar e renderizar o objeto virtual 3D na imagem original com uma melhor precisão e na escala apropriada. Nós validamos numericamente e visualmente a performance do nosso sistema baseado em um banco de dados de imagens e nuvens de pontos de faces de sujeitos masculinos e femininos de diversas idades e formas de face.

Palavras-chave: realidade aumentada, reconstrução facial, computação gráfica

Abstract

This work presents a virtual try-on system to correctly visualize 3D objects (e.g., glasses) in the face of a given user. By capturing the image and depth information of a user through a low-cost RGB-D camera, we apply a face tracking technique to detect specific landmarks in the facial image. These landmarks and the point cloud reconstructed from the depth information are combined to optimize a 3D facial morphable model that fits as good as possible to the user's head and face. At the end, we deform the chosen 3D objects from its rest shape to a deformed shape matching the specific facial shape of the user. The last step projects and renders the 3D object into the original image, with enhanced precision and in proper scale, showing the selected object in the user's face. We validate the performance of our system numerically and visually based on a facial database composed by images and point clouds of male and female subjects with different ages and face shapes.

Keywords: augmented reality, facial reconstruction, computer graphics

Lista de Figuras

1.1	Exemplo de um sistema de realidade aumentada.	18
1.2	Diversos tipos de face e seus diferentes detalhes e formas.	19
2.1	Técnica de animação de avatares em tempo real utilizada no filme <i>Avatar</i>	25
2.2	Processo de calibração do sistema da loja online DITTO.	28
2.3	Visualizações em diferentes ângulos dos óculos virtuais.	29
2.4	Em (a) temos calibração do sistema do espelho virtual que é realizada através da detecção dos olhos e em (b), (c) e (d) temos os diferentes ângulos de visualização do óculos virtual.	30
2.5	Em (a), (b) e (c) temos o ajuste fino dos pontos de interesse faciais detectados pelo sistema TAAZ e em (d) temos o resultado final após um processo de inclusão de maquiagem na imagem.	31
2.6	Pontos de interesse faciais manualmente marcados e malha triangular criada baseada nos pontos marcados.	32
2.7	Objetos virtuais e efeitos adicionados à face do usuário com o aplicativo <i>Snapchat</i>	33
3.1	Pontos de interesse faciais 2D detectados pelo método FaceTracker. Ao todo, detectamos 66 pontos de interesse divididos em 4 grupos: os pontos vermelhos são os pontos de contorno da face, os pontos brancos são os pontos de interesse do olho, os pontos verdes são pontos de interesse do nariz e os pontos azuis são os pontos de interesse internos e externos dos lábios.	36

3.2	Representação no espaço de uma nuvem de pontos S_i de tamanho n representada pelos pentágonos vermelhos e de uma nuvem de pontos P_i de tamanho m representada pelos círculos azuis. Os pentágonos vermelhos representam o grupo de treinamento e os círculos azuis representam o grupo de teste.	39
3.3	Dado um ponto em S_i com $k = 3$ queremos saber os P_i pontos mais próximos do ponto S_i	39
3.4	k P_i pontos correspondentes ao ponto S_i	40
3.5	Nuvens de pontos \mathbf{M} (em preto) \mathbf{D} (em vermelho) desalinhadas.	40
3.6	Nuvens de pontos \mathbf{M} (em preto) \mathbf{D} (em vermelho) devidamente alinhadas após a aplicação, a cada iteração, da matriz T no conjunto de pontos \mathbf{D}	44
3.7	Minimização da função $f(x) = x^2$ com $x_0 = 0.8$ para diferentes valores de δ	45
3.8	Em (a) temos a imagem de um crânio humano, em (b) os pontos de um crânio de um chimpanzé e em (c) temos o resultado após o processo de <i>thin-plate</i>	46
4.1	Visão geral do sistema de visualização de objetos virtuais 3D na face do usuário. Inicialmente geramos um modelo morfológico facial 3D masculino e feminino baseado em uma série de faces escaneadas de milhares de indivíduos. Como entrada do nosso sistema, temos uma imagem RGB e suas informações de profundidade capturadas através de uma câmera RGB-D de baixo custo além dos pontos de interesse faciais 2D e 3D. Essas informações de entrada são utilizadas para otimizar os parâmetros do nosso modelo morfológico virtual 3D para que esse se alinhe a forma do rosto do usuário. No final, deformamos o objeto virtual 3D baseado na deformação do modelo e o projetamos para 2D e geramos uma imagem da face do usuário com o respectivo objeto virtual.	51
4.2	Faces escaneadas em diferentes posições em \mathbb{R}^3	52
4.3	<i>Landmarks</i> 3D manualmente marcadas no modelo morfológico facial 3D.	53
4.4	O Microsoft Kinect Primeira Versão.	54

4.5	Nosso sistema de aquisição de dados. O Microsoft Kinect é utilizado para capturar uma imagem RGB e uma nuvem de pontos. Logo após, um rastreador de pontos de interesse facial é aplicado para detectar os <i>landmarks</i> na imagem RGB. Como resultado, nós geramos os <i>landmarks</i> 2D correspondendo à face na imagem RGB (a) e os <i>landmarks</i> 3D correspondendo à nuvem de pontos de entrada (b). Como podemos observar em (c), antes do processo de otimização, podemos visualizar o modelo morfológico facial 3D alinhado à nuvem de pontos através da matriz de transformação gerada entre os <i>landmarks</i> 3D manualmente marcados do modelo morfológico e os da nuvem de pontos.	56
4.6	KNN na configuração nuvem de pontos como alvo e modelo morfológico facial 3D como origem.	59
4.7	A entrada do processo de otimização são os modelos morfológicos faciais 3D masculinos e femininos, a nuvem de pontos e os seus respectivos pontos de interesse faciais 3D. Como saída, geramos o modelo morfológico facial 3D ajustado à face do usuário.	60
4.8	Dados os modelos morfológicos faciais 3D original e otimizado, aplicamos o método TPS e achamos uma transformação afim que representa a deformação ocorrida de S^{or} para S^{op} . Depois, qualquer conjunto de pontos que esteja no mesmo espaço que os modelos iniciais, se aplicado a eles essa transformação, sofrerão as deformações ocorridas.	61
4.9	Imagem capturada pela câmera RGB-D com dimensão de 640×480 e com o objeto virtual renderizado.	62
4.10	Imagem capturada pela câmera RGB-D com dimensão de 640×480 em (a) e imagem com dimensão de 512×512 cortada baseado no centro de massa dos pontos de interesse faciais em (b).	62
5.1	<i>Tuning</i> do sistema para ambos os gêneros com $nComp = 50$	66
5.2	<i>Tuning</i> do sistema para ambos os gêneros com $nComp = 100$	67

Lista de Figuras	10
<hr/>	
5.3 <i>Tuning</i> do sistema para ambos os gêneros com $nComp = 150$	67
5.4 <i>Tuning</i> do sistema para ambos os gêneros com $nComp = 200$	67
5.5 <i>Tuning</i> do sistema para ambos os gêneros com $nComp = 250$	68
5.6 <i>Tuning</i> do sistema para ambos os gêneros com $nComp = 300$	68
5.7 <i>Tuning</i> do sistema para ambos os gêneros com $nComp = 350$	68
5.8 <i>Tuning</i> do sistema para ambos os gêneros com $nComp = 400$	69
5.9 <i>Tuning</i> do sistema para ambos os gêneros com $nComp = 450$	69
5.10 <i>Tuning</i> do sistema para ambos os gêneros com $nComp = 500$	69
5.11 Oito pontos de interesse faciais utilizados para avaliar a precisão numérica do nosso sistema. As medidas são: (a) distância entre o meio dos olhos e o queixo, (b) distância entre os olhos, (c) tamanho do nariz e (d) distância entre as bochechas.	73
5.12 Os cinco modelos virtuais utilizados na elaboração dos resultados visuais deste trabalho. Em (a) temos um modelo de aro redondo, em (b) um modelo conhecido como tartaruga, em (c) um modelo esportivo, em (d) um modelo clássico e em (e) um modelo de aro mais grosso.	76
5.13 Em (a), (b) e (c) podemos observar a nuvem de pontos e seu respectivo modelo morfológico facial 3D otimizado para os indivíduos masculinos.	76
5.14 Em (a), (b) e (c) podemos observar a nuvem de pontos e seu respectivo modelo morfológico facial 3D otimizado para os indivíduos femininos.	77
5.15 Resultados visuais do nosso sistema. Indivíduos masculinos e femininos com diferentes tipos de óculos. A coluna à esquerda mostra o objeto virtual renderizado na imagem RGB, sem passar pelo processo de otimização, lado a lado com o objeto virtual após o processo de otimização. Na coluna da direita apresentamos a imagem da coluna da esquerda cortada na região dos olhos para uma melhor visualização do resultado.	79

-
- 5.16 Resultados visuais do nosso sistema. Indivíduos masculinos e femininos com diferentes tipos de óculos. A coluna à esquerda mostra o objeto virtual renderizado na imagem RGB, sem passar pelo processo de otimização, lado a lado com o objeto virtual após o processo de otimização. Na coluna da direita apresentamos a imagem da coluna da esquerda cortada na região dos olhos para uma melhor visualização do resultado. 80

Lista de Tabelas

5.1 Os parâmetros do nosso sistema são, número de componentes do modelo morfológico facial 3D (<i>nComp</i>) e o parâmetro que controla o passo do gradiente δ . Essa tabela mostra os intervalos de valores e o incremento de cada intervalo durante o processo de <i>tuning</i>	65
5.2 Menores erros médios encontrados para o conjunto de componentes dos indivíduos masculinos. Para cada componente, mostramos o valor δ que possui o menor erro médio e o seu respectivo tempo médio de processamento.	71
5.3 Menores erros médios encontrados para o conjunto de componentes dos indivíduos femininos. Para cada componente, mostramos o valor δ que possui o menor erro médio e o seu respectivo tempo médio de processamento.	72
5.4 Medidas para o conjunto de testes masculino. Para cada medida (primeira coluna), temos a distância entre os pontos de interesse faciais da nuvem de pontos (segunda coluna), representado por <i>mPontos</i> , e sua distância correspondente no modelo morfológico facial 3D otimizado (terceira coluna), representado por <i>mModelo</i> . Na última coluna temos a diferença entre as distâncias da segunda e terceira coluna, representado por <i>difPM</i>	74

5.5 Medidas para o conjunto de testes feminino. Para cada medida (primeira coluna), temos a distância entre os pontos de interesse faciais da nuvem de pontos (segunda coluna), representado por mPontos, e sua distância correspondente no modelo morfológico facial 3D otimizado (terceira coluna), representado por mModelo. Na última coluna temos a diferença entre as distâncias da segunda e terceira coluna, representado por difPM.	74
--	----

Sumário

1	Introdução	16
1.1	Motivação	18
1.2	Contribuição	20
1.3	Estrutura do Trabalho	20
2	Trabalhos Correlatos	22
2.1	Reconstrução Facial	22
2.2	Aplicações de Realidade Aumentada	25
2.3	Aplicações Comerciais com Realidade Aumentada Facial	27
3	Fundamentação Teórica	34
3.1	A Detecção dos <i>Landmarks</i>	35
3.2	Método de Análise de Componentes Principais	36
3.3	O Algoritmo dos K Vizinhos Mais Próximos	38
3.4	O Algoritmo Iterativo de Pontos Mais Próximos	40
3.5	A Otimização por Descida de Gradiente	44
3.6	<i>Spline</i> Placa-Fina	46
4	Sistema Proposto	50
4.1	Criação do Modelo Morfológico 3D	52
4.2	Aquisição de Dados	54
4.3	Otimização do Modelo Morfológico Facial 3D	56
4.4	A Deformação dos Objetos Virtuais 3D	60
4.5	A Composição da Imagem Final	61

Sumário	15
<hr/>	
5 Resultados	64
5.1 <i>Tuning</i> do Sistema	64
5.2 Experimentos Numéricos	72
5.3 Resultados Visuais	75
5.4 Limitações	81
6 Conclusão	82
6.1 Trabalhos Futuros	83
Referências Bibliográficas	84

Capítulo 1

Introdução

Através do uso de técnicas de Realidade Aumentada (RA), um sistema pode ser desenvolvido integrando elementos virtuais com visualizações do mundo real, como por exemplo, uma imagem capturada por uma câmera. Objetos virtuais 3D podem ser sobrepostos sob imagens reais e a sensação é de que eles fazem parte da cena (LIMA et al., 2015). Ao contrário da Realidade Virtual (RV), onde o usuário está completamente imerso num mundo virtual, a Realidade Aumentada permite, segundo (KIRNER, 2016), que o usuário veja, ouça, sinta e interaja com informações e elementos virtuais em um ambiente onde ele está apenas parcialmente imerso, ou seja, a Realidade Aumentada dispensa o uso de dispositivos de custo mais elevado como um *Oculus Rift* ou uma caverna digital (CAVE). A Realidade Aumentada tem o poder de dar vida a uma imagem, produto ou até mesmo a uma vitrine. Desta forma, a atração dos consumidores por determinada marca ou produto tende a crescer pois o uso da Realidade Aumentada proporciona uma nova forma de interação (MARKETING, 2014).

Para compor um sistema de Realidade Aumentada são necessários os quatro itens: 1 - computador ou algum dispositivo capaz de executar um aplicativo como celulares ou tablets; 2 - uma câmera, ou seja, qualquer dispositivo capaz de tirar uma fotografia; 3 - um marcador, ou seja, o local onde o objeto virtual será renderizado. Este marcador pode ser um *QR code*, uma face, como no caso deste trabalho ou qualquer objeto do mundo real que o programador

defina; 4 - um aplicativo adequado, ou seja, um aplicativo devidamente programado para lidar com as informações de entrada e a renderização do objeto virtual.

Ao longo do tempo, a face humana vem sendo estudada nas mais diferentes áreas, como por exemplo, na interação homem-máquina, entretenimento, medicina e segurança (BOLKART; WUHRER, 2015). Nos dias atuais, algumas aplicações e produtos combinam Realidade Aumentada com modelos de face. Em (ALMEIDA et al., 2015), por exemplo, o usuário tira uma foto de sua face através de um dispositivo móvel e depois de um processo de reconhecimento de pontos de interesse da face, é possível a aplicação de uma maquiagem virtual na foto. Como exemplo de uso para fins comerciais, temos a empresa DITTO (DITTO, 2016), uma loja online de óculos que permite ao usuário experimentá-los virtualmente antes de comprá-los. Por fim, podemos citar o Snapchat (SNAPCHAT, 2016) um aplicativo vastamente conhecido nos dias atuais e com versões para Android e IOS onde através de um rastreamento de face em tempo real, somos capazes de aplicar efeitos especiais como por exemplo orelhas de gatos, olhos grandes, etc.

Este trabalho propõe um sistema para a visualização de objetos virtuais na face do usuário. Inicialmente geramos um modelo morfológico de face 3D baseado em um banco de faces escaneadas de milhares de indivíduos. Utilizamos como entrada do nosso sistema uma imagem RGB e as informações de profundidade de uma câmera RGB-D de baixo custo, o Microsoft Kinect. Depois nós detectamos os pontos de interesse faciais 2D e 3D, também conhecido na literatura como *landmarks*. Em seguida, nós otimizamos os parâmetros do nosso modelo morfológico facial 3D para que este se alinhe aos *landmarks* e a nuvem de pontos gerada pelas informações de profundidade da câmera RGB-D. Essa otimização garante o melhor alinhamento do modelo ao rosto do usuário.

Diferentemente dos trabalhos e produtos previamente relatados, nosso sistema modela a face do usuário e os objetos virtuais no espaço 3D e em seguida projeta-os para 2D e, desta forma, compomos uma imagem do usuário com os objetos virtuais. Essas diferenças nos permitem lidar com situações desafiadoras como, por exemplo, a aplicação de efeitos

luminosos ao objeto virtual bem como a obtenção de medidas faciais reais que são calculadas pelas distâncias entre os pontos de interesse faciais em 3D capturados pelo nosso sistema. As situações apresentadas anteriormente podem ser difíceis de serem gerenciadas em 2D. A performance do nosso sistema é verificada através de um banco de faces e nuvens de pontos composto de usuários do sexo masculino e feminino de diversas idades e formatos de faces. Finalmente, mostramos numericamente e visualmente que, o ajuste do modelo morfológico de face 3D garante uma melhor precisão de posicionamento e escala dos objetos virtuais.

1.1 Motivação

Um sistema de realidade aumentada é um sistema que cria uma visualização de uma cena real, juntando um objeto virtual gerado pelo computador, incluindo todas as suas propriedades tridimensionais e a cena em questão, conforme podemos observar na Figura 1.1. A ideia é que a medida que o usuário se mova pela cena real o objeto virtual pareça realmente fazer parte da cena.



Figura 1.1: Exemplo de um sistema de realidade aumentada.¹

Além de fazer parte da cena real, dependendo do sistema desenvolvido, o objeto virtual

¹Disponível em: <<http://fillip.pro/portfolios/augmented-reality/>>. Acesso em julho, 2016.

pode interagir virtualmente com os objetos reais de uma maneira natural. De uma forma geral, a realidade aumentada melhora a performance e percepção do usuário com o mundo a sua volta (VALLINO, 1998).

O objetivo principal da realidade aumentada é o de criar um sistema onde o usuário não consiga distinguir a diferença entre o mundo real e o mundo virtual criado baseado na imagem do mundo real.

A face humana é uma parte do corpo desafiadora para várias áreas da ciência e também para a Realidade Aumentada devido à grande complexidade de seus detalhes e formas, conforme podemos observar na Figura 1.2.



Figura 1.2: Diversos tipos de face e seus diferentes detalhes e formas.²

Desta forma, muitas pessoas, por questão de estética ou até mesmo vaidade gostariam que fosse possível a inserção de itens em suas faces de forma que os mesmos além de parecerem quase reais, estejam devidamente ajustados ao formato de suas respectivas faces. Outra função da realidade aumentada, é a de evitar arrependimentos e perda de tempo com erros desnecessários com a utilização de aplicativos onde seja possível, por exemplo, a pré-

²Disponível em: <<http://www.timesofmalta.com/articles/view/20140922/world/why-human-faces-lookso-different.536732>>. Acesso em julho, 2016.

visualização de maquiagens, diferentes tipos de cabelo ou até mesmo tatuagens na face do usuário. Em âmbito comercial a realidade aumentada, além de fazer com que o usuário muitas vezes se cativasse com uma determinada marca ou empresa, tem como objetivo criar experiências virtuais que tragam algum teor de realismo para o cliente como, por exemplo, um *e-commerce* de uma ótica, onde, com o auxílio da realidade aumentada é possível a utilização de um provador virtual dos óculos vendidos por esta empresa.

1.2 Contribuição

Este trabalho propõe um visualizador de objetos virtuais 3D na face do usuário, como por exemplo, óculos, bigodes, etc. Nós reconstruímos um modelo de face 3D com técnicas de otimização para ajustá-lo ao rosto do usuário. Como resultado, os objetos virtuais 3D que serão projetados na face, podem ser previamente personalizados pelo usuário e posteriormente, visualizados com mais realismo.

1.3 Estrutura do Trabalho

Este trabalho está dividido da seguinte forma:

- No Capítulo 2, faremos uma breve revisão bibliográfica que vai desde o começo da utilização de modelos morfológicos faciais 3D, passando por alguns trabalhos que mesclam o uso de modelos morfológicos faciais 3D e Realidade Aumentada e por fim, apresentamos alguns produtos comerciais relacionados ao tópico do trabalho.
- No Capítulo 3 abordaremos a metodologia estudada na elaboração deste trabalho, ou seja, toda a base teórica para o desenvolvimento deste trabalho.

-
- No Capítulo 4 faremos uma apresentação de cada parte do sistema, apresentando alguns métodos utilizados e informações relevantes para a melhor compreensão deste trabalho.
 - No Capítulo 5 falaremos sobre a calibração do sistema bem como sobre os resultados numéricos e visuais.
 - Finalmente no Capítulo 6 apresentaremos uma breve conclusão sobre o sistema desenvolvido e proporemos trabalhos futuros.

Capítulo 2

Trabalhos Correlatos

Neste capítulo iremos descrever alguns trabalhos correlatos à esta dissertação. Começaremos apresentando trabalhos que falam sobre as várias formas para realizar uma reconstrução facial utilizando modelos morfológicos faciais 3D. Na segunda parte deste capítulo, apresentaremos alguns trabalhos que mesclam o uso de Realidade Aumentada e modelos morfológicos faciais e não faciais. Na terceira e última parte deste capítulo apresentaremos alguns produtos relacionados ao tópico do trabalho.

2.1 Reconstrução Facial

O uso de modelos morfológicos 3D para modelar faces humanas tem uma vasta gama de aplicações como por exemplo, visão computacional, análise facial, reconhecimento de face além de computação gráfica e animações. Esses modelos representam a face humana através de malhas poligonais 3D e podem ser criados baseado em imagens 2D, conceito introduzido por (VETTER; POGGIO, 1997) e (LANITIS; TAYLOR; COOTES, 1997). Estes modelos também podem ser criados baseados em escaneamentos 3D da face, como foi apresentado por (BLANZ; VETTER, 1999) através da utilização de um *scanner* a laser. Este último trabalho citado introduziu a utilização da análise dos componentes principais (JOLLIFFE,

1986) afim de reduzir as redundâncias das faces escaneadas e representar os dados da melhor forma possível.

A principal área de atuação em pesquisas que envolvem o estudo de modelos morfológicos faciais 3D é o de análise por síntese, proposto por (BLANZ; VETTER, 1999) onde, dado um modelo 3D e uma imagem 2D, os parâmetros 2D da imagem são capturados e servem de entrada para o ajuste do modelo morfológico facial 3D à face da imagem 2D (KITTLER et al., 2016).

O processo de ajuste do modelo morfológico facial 3D a uma face, é uma das várias formas de se reconstruir uma face e possui uma série de dificuldades como por exemplo, estimação da posição da cabeça, iluminação e representação de expressões faciais (ARELLANO; DAHYOT, 2013) e ao longo dos anos diversas técnicas vem sendo apresentadas para solucioná-las. (HUBER et al., 2016) apresenta uma técnica para lidar com o problema do posicionamento da cabeça na imagem. Primeiramente um modelo morfológico facial 3D é construído e seus devidos pontos de interesse faciais são marcados. Os pontos de interesse faciais da imagem 2D são detectados e, baseado nos pontos do modelo e da imagem, a pose do modelo facial 3D é estimada, ou seja, a posição da câmera na cena. O modelo é então ajustado à face e é devidamente texturizado. (PIOTRASCHKE; BLANZ, 2016) afirma que muitas vezes não é possível reconstruir um modelo facial com somente uma imagem pois a mesma pode apresentar muitas irregularidades como por exemplo, iluminação, oclusões, expressões faciais ou até mesmo uma pose onde poucos pontos de interesse faciais possam ser reconhecidos. Para contornar o problema, o autor trabalha com um conjunto de imagens. A face é dividida em três segmentos: olhos, nariz e boca. Desta forma, os pontos de interesse faciais são devidamente reconhecidos e uma média ponderada de suas distâncias é realizada para garantir uma melhor qualidade ao modelo reconstruído. Através da detecção de bordas e pontos de interesse faciais da face também é possível reconstruir um modelo facial e essa é a solução descrita por (BAS et al., 2016) e por (LEE et al., 2003).

Existe ainda a área de reconstrução facial de alta qualidade que são utilizados em jogos

de computador e filmes em 3D. Essa área se baseia em escaneamentos faciais utilizando *scanners* a laser (ALEXANDER et al., 2013) (ALEXANDER et al., 2009) ou através de imagens estéreo (GHOSH et al., 2011) (BEELER et al., 2011). Essas técnicas, apesar de gerarem modelos com uma alta precisão, demandam muito trabalho de pós-processamento para gerar o modelo morfológico. Em contrapartida, nesta área, problemas como a estimação de expressões faciais ou a geração de modelos morfológicos faciais 3D não-rígidos são devidamente contornados devido à qualidade dos dados para a geração do modelo.

Outra área abrangente é a de reconstrução de modelos morfológicos faciais 3D com expressões faciais, também conhecidos como modelos de face multilineares. (AMBERG; PAYSAN; VETTER, 2009) propõe um método onde dado um modelo multilinear reconstruído de um banco de faces escaneadas, devidamente texturizado com o rosto do usuário, é possível alterar os parâmetros do modelo como tamanho do rosto e expressões faciais. (ZHANG; WEI, 2012) propõe utilizar um modelo multilinear de imagens 2D onde, dado dois indivíduos com a mesma expressão facial, é possível transferir uma sequência de expressões faciais entre eles. Já (VLASIC et al., 2005) utiliza um modelo de face linear para mesclar expressões de diferentes indivíduos e alterar a expressão de um usuário de treino em um vídeo.

A animação facial de avatares em tempo real, também conhecida como *facial motion capture*, é uma técnica bastante difundida em filmes da atualidade devido a evolução da mesma, conforme podemos observar na Figura 2.1. Nesta técnica, uma câmera filma o rosto do usuário, que possui vários pontos de referência marcados em sua face. Esses pontos possuem correspondências em um modelo facial 3D e, em tempo real, as expressões feitas pelo usuário são animadas no modelo facial 3D.



Figura 2.1: Técnica de animação de avatares em tempo real utilizada no filme *Avatar*.¹

Na área da saúde podemos citar o trabalho de (OLIVEIRA-SANTOS et al., 2013) onde baseado em um conjunto de imagens de face em diferentes ângulos, alguns pontos de interesse faciais são detectados e servem como parâmetros para a reconstrução de um modelo facial 3D. Este modelo reconstruído pode ser alterado em tempo real afim de prever o resultado de intervenções estéticas faciais.

2.2 Aplicações de Realidade Aumentada

Com o advento das câmeras de baixo custo, tanto as RGB, que nos oferecem uma imagem colorida quanto as RGB-D, que nos oferecem além da imagem colorida as informações de profundidade da cena, a comunidade científica se aproveitou deste fato para mesclar o mundo virtual e o mundo real. (HAUSWIESNER; STRAKA; REITMAYR, 2013) propõe um provador virtual de roupas baseado apenas em renderização de imagens. Já (YANG et al., 2015) propõe um provador virtual de calçados onde o posicionamento do calçado é baseado em alguns pontos de referência nos pés do usuário em conjunto com as informações de profundidade provenientes de uma câmera RGB-D.

¹Disponível em: <<https://www.fxguide.com/fxpodcasts/fxpodcast-dr-mark-sagar/>>. Acesso em agosto, 2016.

Na área de inserção de objetos na face do usuário, (PENG, 2015) utiliza a câmera de um *smartphone* e em tempo real aplica o classificador *Haar-cascade* para detectar a face do usuário. Com a face detectada, uma máscara é projetada sobre a face do usuário e através de botões virtuais, é possível mudar as expressões faciais da máscara.

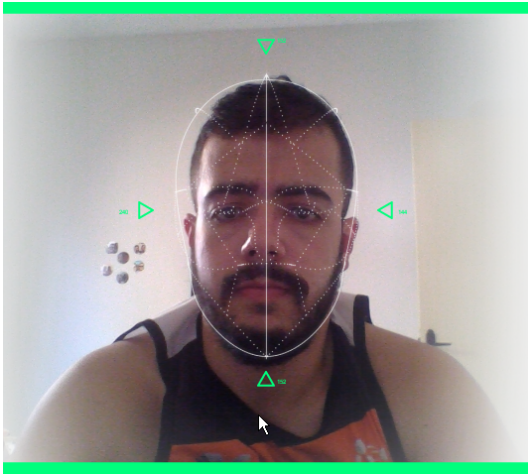
Alguns trabalhos mesclam o uso de modelos morfológicos 3D e realidade aumentada e este é um tópico de interesse nas mais diversas áreas de pesquisa. No segmento da saúde, (BRAGA et al., 2012) propõe um sistema para auxiliar uma cirurgia ortodôntica combinando um escaneamento de face e um raio-X da face e como resultado, um modelo 3D de face com o possível resultado da cirurgia é apresentado ao usuário.

A área cosmética pode também se beneficiar dessas aplicações. (WANG et al., 2011) propõe um *framework* de maquiagem similar ao sistema apresentado neste trabalho. Baseado em um banco de imagens de face 2D, alguns pontos de interesse faciais ou *landmarks* são detectados em cada imagem, esses pontos são projetados para \mathbb{R}^3 e um modelo morfológico facial 3D é reconstruído. Depois deste processo, a forma do modelo morfológico é ajustada baseados nos pontos de interesse detectados e como resultado o modelo morfológico pode ser texturizado com a face do usuário e a maquiagem é aplicada. Em comparação, este trabalho gera um modelo morfológico 3D baseado em uma coleção de escaneamentos de face 3D. Em nosso trabalho nós também deformamos e projetamos os objetos virtual 3D a fim de compor a imagem final.

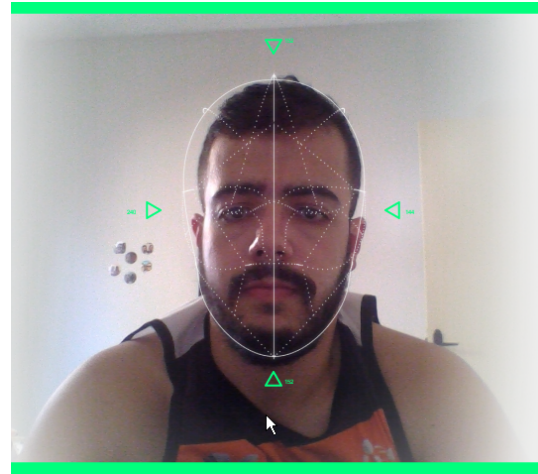
Outros dois trabalhos semelhantes a estes foram propostos por (HUANG; YANG; CHU, 2012) e (YUAN et al., 2011). Ambos os trabalhos propõem um provador interativo de óculos utilizando um modelo morfológico 3D. O alinhamento dos óculos à face do usuário são baseados em parâmetros como posição da cabeça, rotação da cabeça e tamanho dos olhos. Em comparação, nós otimizamos a forma da face e deformamos o objeto subjacente para corresponder com a forma ajustada da face. Desta forma, acreditamos que é possível renderizar o objeto com melhor precisão pois o mesmo se ajusta à cada usuário.

2.3 Aplicações Comerciais com Realidade Aumentada Facial

Nos dias de hoje, alguns produtos de caráter comercial relacionados ao nosso sistema estão disponíveis. Por exemplo, DITTO (DITTO, 2016) é uma loja online que permite ao usuário experimentar virtualmente os óculos que são vendidos no site. O usuário é guiado por um processo de calibração onde, após o usuário alinhar a sua face a um campo pré-estabelecido um pequeno vídeo é gravado onde o usuário deve lentamente rotacionar a cabeça para esquerda e para a direita. Além disso é necessário que o usuário tire uma foto com um objeto retangular acima de sua sobrancelha para a demarcação do limite dos olhos. Este processo pode ser observado na Figura 2.2.



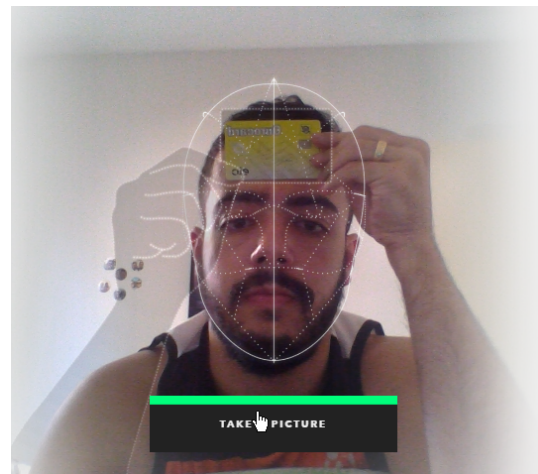
(a) Centralização da face do usuário em um campo pré-estabelecido.



(b) Início da gravação do vídeo de calibração.



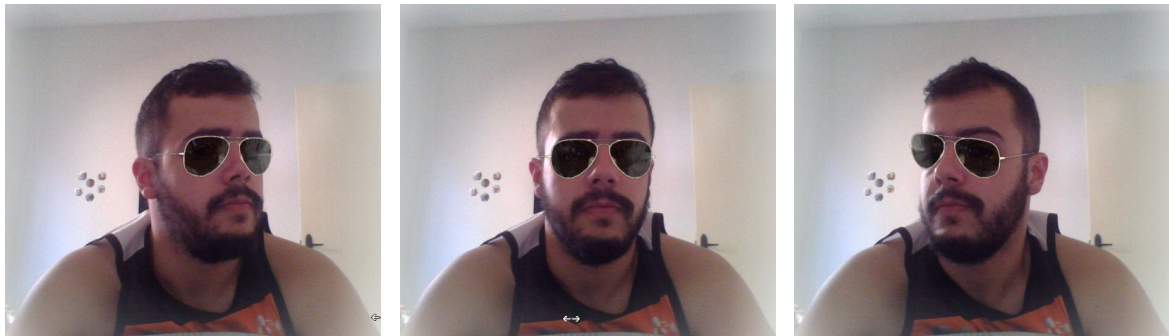
(c) Final da gravação do vídeo de calibração.



(d) Objeto retangular para demarcar a altura da sobrancelha.

Figura 2.2: Processo de calibração do sistema da loja online DITTO.

Após a calibração do sistema o usuário pode visualizar alguns óculos pré-estabelecidos pelo site em diversos ângulos conforme podemos observar na Figura 2.3. Não é necessário recalibrar o sistema para experimentar outros tipos de óculos.



(a) Lateral direita.

(b) Frontal.

(c) Lateral esquerda.

Figura 2.3: Visualizações em diferentes ângulos dos óculos virtuais.

O espelho virtual Ray-Ban (RAY-BAN, 2016) é outro sistema de realidade aumentada similar que permite ao usuário experimentar óculos. Com um sistema de calibração baseado apenas na detecção dos olhos, basta que o usuário posicione o rosto na área demarcada e rapidamente os olhos são detectados e o óculos é posicionado na face do usuário. O processo de calibração e visualização pode ser observado na Figura 2.4.



(a) Calibração do sistema.



(b) Lateral direita.



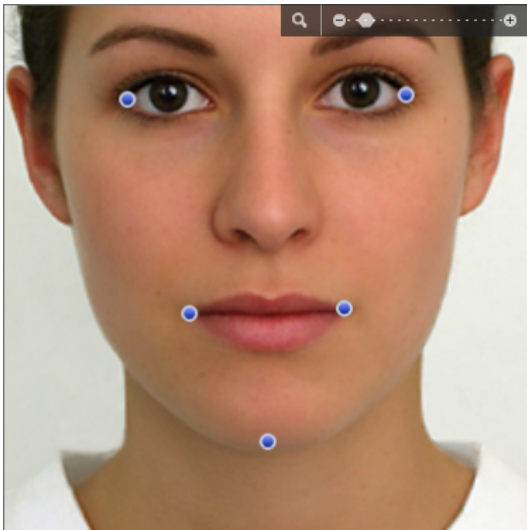
(c) Lateral esquerda.



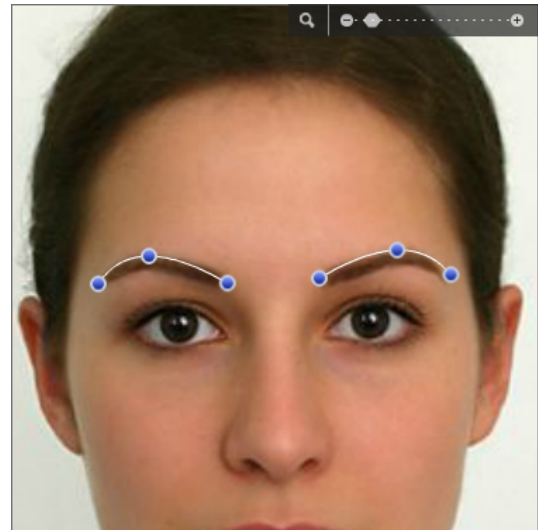
(d) Frontal.

Figura 2.4: Em (a) temos calibração do sistema do espelho virtual que é realizada através da detecção dos olhos e em (b), (c) e (d) temos os diferentes ângulos de visualização do óculos virtual.

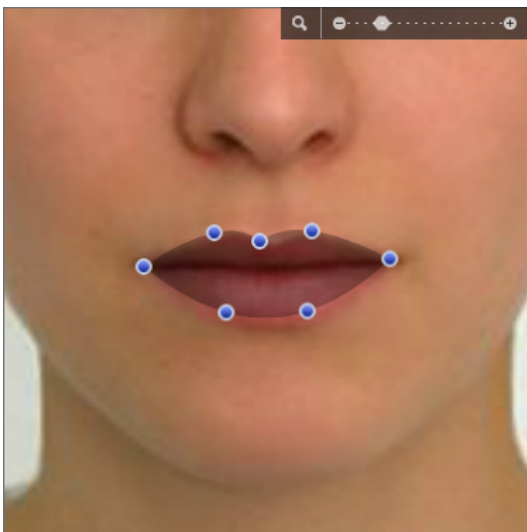
Além da área de provedores virtuais específicos para óculos, um dos ramos que se beneficiam do poder da realidade aumentada é o da beleza. Um exemplo é o website TAAZ (TAAZ, 2016) onde o usuário carrega um foto de sua face e após um longo processo de ajuste fino dos pontos de interesse faciais detectados pelo sistema, o usuário consegue aplicar em sua foto os mais diversos tipos de maquiagens, lentes de contato, batons, dentre outros. O processo de ajuste fino e o resultado final podem ser observados na Figura 2.5.



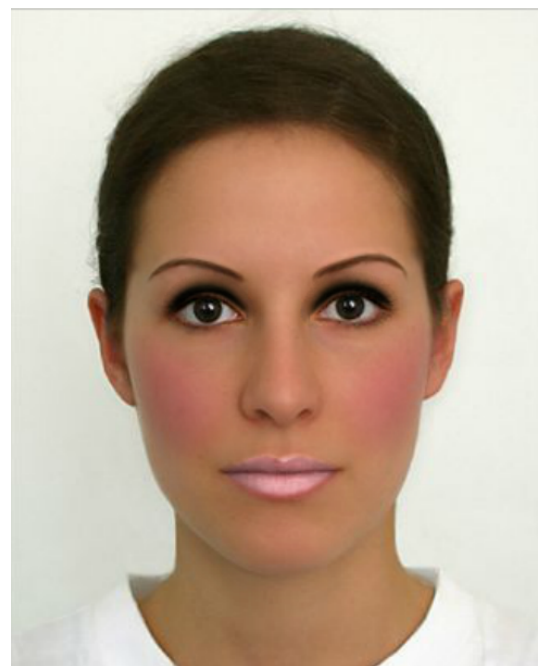
(a) Ajuste dos olhos, boca e queixo.



(b) Ajuste da sobrancelha.



(c) Ajuste dos lábios.



(d) Resultado final com a maquiagem aplicada.

Figura 2.5: Em (a), (b) e (c) temos o ajuste fino dos pontos de interesse faciais detectados pelo sistema TAAZ e em (d) temos o resultado final após um processo de inclusão de maquiagem na imagem.

Por fim, outra área que vem avançando rapidamente nos últimos anos são os aplicativos de realidade aumentada para *smartphones*. Um dos aplicativos mais significativos nesta área é o *Snapchat* (SNAPCHAT, 2016). Nele, baseados em milhares de imagens de treino de face

com os pontos de interesse faciais manualmente marcados, uma malha triangular é criada baseada nesses pontos conforme podemos observar na Figura 2.6 e uma série de objetos e efeitos podem ser adicionados à face do usuário conforme podemos observar na Figura 2.7.

Em comparação aos sistemas apresentados nesta seção, ao invés de trabalharmos com processamento de imagens para a aplicação dos objetos virtuais na face do usuário, nós utilizamos a nuvem de pontos capturada por um câmera RGB-D, reconstruímos a face do usuário em 3D, ajustamos o objeto virtual ao modelo facial reconstruído do usuário e somente após este processo nós convertemos o objeto virtual de 3D para 2D e o aplicamos sobre a imagem. Desta forma nós garantimos que os objetos virtuais fiquem mais fiéis à forma do rosto do usuário. Além disso, nosso sistema não necessita de calibração manual, ou seja, o processo de captura dos dados e reconstrução facial é realizado de forma automática.

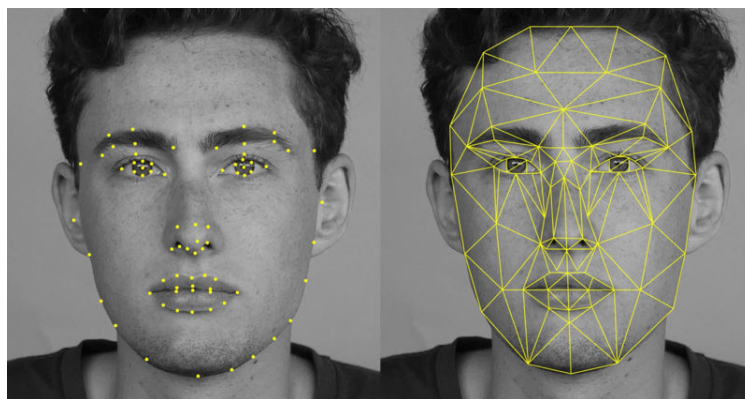


Figura 2.6: Pontos de interesse faciais manualmente marcados e malha triangular criada baseada nos pontos marcados.²

²Disponível em: <<http://petapixel.com/2016/06/30/snapchats-powerful-facial-recognition-technology-works/>>. Acesso em agosto, 2016.



Figura 2.7: Objetos virtuais e efeitos adicionados à face do usuário com o aplicativo *Snapchat*.³

³Disponível em: <<http://thisfairytalelife.com/why-i-love-snapchat/>>. Acesso em agosto, 2016.

Capítulo 3

Fundamentação Teórica

Neste capítulo vamos abordar a metodologia teórica que embasa todas as etapas do nosso sistema. Iniciaremos descrevendo a detecção dos pontos de interesse faciais ou *landmarks*, etapa importante do trabalho pois com esses pontos, conseguimos realizar alinhamentos de nuvens de pontos em diferentes locais no espaço. Em seguida, descreveremos o método de análise dos componentes principais, que neste trabalho é usado para a criação do modelo morfológico facial 3D que será ajustado à forma da face do usuário. Na sequência, abordaremos o algoritmo dos k vizinhos mais próximos, que neste trabalho nos auxilia na obtenção da correspondência entre diferentes nuvens de pontos. O próximo tópico é o algoritmo iterativo de pontos mais próximos, que é um algoritmo onde, dados duas nuvens de pontos ou os *landmarks*, estima uma matriz de transformação que as alinha. Em seguida temos a otimização por descida de gradiente, etapa onde os pesos do modelo morfológico facial 3D são otimizados fazendo com que este se ajuste à face do usuário. Por fim, descrevemos sobre o método da *spline* placa fina onde, dados dois conjuntos de pontos devidamente relacionados, achamos uma transformação que represente a deformação ocorrida entre ambos.

3.1 A Detecção dos *Landmarks*

Nós detectamos importantes *landmarks* 2D na imagem RGB de entrada utilizando o algoritmo *FaceTracker* (SARAGIH, 2013). Este algoritmo é baseado em um método de ajuste de um modelo deformável onde este modelo, que é parametrizado, é registrado de tal forma que seus pontos de interesse faciais correspondam consistentemente aos do usuário. Este método é caracterizado por uma aproximação linear, denominada distribuição de pontos do modelo (em inglês, *Point Distribution Model - PDM*) e descrita por (COOTES; TAYLOR, 1992). Nele, modelamos as variações de formas não-rígidas, por exemplo, a face humana, de maneira linear através de uma transformação não-rígida que alinha um modelo 3D a uma imagem RGB da seguinte forma:

$$x_i = sR(\bar{x}_i + \phi_i q) + t \quad (3.1)$$

onde x_i é a localização 2D do i -ésimo ponto de interesse facial e s, R, t, q é o conjunto de parâmetros do modelo onde s é uma escala global, R é uma rotação, t uma translação e q é uma série de parâmetros não-rígidos. \bar{x}_i é a localização média do i -ésimo ponto de interesse facial do PDM em referência à imagem RGB e por fim, ϕ_i é uma submatriz de variação de bases do ponto de interesse facial i (SARAGIH; LUCEY; COHN, 2011).

Utilizando o algoritmo *FaceTracker*, 66 pontos de interesse faciais são registrados, como mostrado na Figura 3.1, divididos em 4 categorias: 27 *landmarks* de contorno de face, 12 *landmarks* para os olhos (6 para cada um), 09 *landmarks* para o nariz, 18 *landmarks* para a boca.



Figura 3.1: Pontos de interesse faciais 2D detectados pelo método FaceTracker. Ao todo, detectamos 66 pontos de interesse divididos em 4 grupos: os pontos vermelhos são os pontos de contorno da face, os pontos brancos são os pontos de interesse do olho, os pontos verdes são pontos de interesse do nariz e os pontos azuis são os pontos de interesse internos e externos dos lábios.

3.2 Método de Análise de Componentes Principais

O método de Análise de Componentes Principais (em inglês, *Principal Components Analysis - PCA*), também conhecido por **transformada Karhunen-Loève (transformada KLT)** ou **transformada Hotelling**, foi proposto em 1901 por Karl Pearson (PEARSON, 1901) e a melhor referência atual para o método é a de Jolliffe (JOLLIFFE, 1986). Trata-se basicamente de uma transformação na qual se busca um sistema de coordenadas ortogonal que melhor represente os dados de maneira que os eixos coordenados (componentes principais) sejam computados dependendo das direções com maior variância dos dados. O PCA extrai características reduzindo as redundâncias e desta forma, é possível fazer uma melhor representação dos dados.

Neste trabalho, o PCA nos permitiu a criação de um modelo morfológico facial 3D masculino e outro feminino baseado em uma série de faces escaneadas.

Para exemplificar o método, vamos supor um conjunto de faces escaneadas M contendo

m vetores de dados x^l de tamanho n , logo, temos:

$$\begin{aligned} x^1 &= (x_1^1, x_2^1, \dots, x_n^1) \\ x^2 &= (x_1^2, x_2^2, \dots, x_n^2) \\ &\vdots \\ x^m &= (x_1^m, x_2^m, \dots, x_n^m) \end{aligned}$$

O primeiro passo do PCA é achar um vetor médio \bar{x} que contém a média de cada coordenada do banco de faces S . O vetor \bar{x} é dado por:

$$\bar{x} = (\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n) = \left(\frac{x_1^1 + \dots + x_1^m}{m}, \dots, \frac{x_n^1 + \dots + x_n^m}{m} \right) \quad (3.2)$$

Depois, subtraímos todos os vetores do conjunto inicial pelo vetor de média \bar{x} e criamos a matriz de dados $M_{m,n}$:

$$M_{m,n} = \begin{pmatrix} x_1^1 - \bar{x}_1 & \dots & x_n^1 - \bar{x}_n \\ \vdots & \ddots & \vdots \\ x_1^m - \bar{x}_1 & \dots & x_n^m - \bar{x}_n \end{pmatrix}$$

Cada linha da matriz $M_{m,n}$ é um vetor representando a diferença entre o vetor original e o vetor de média \bar{x} .

O passo seguinte do PCA é a criação da matriz de covariância. Essa matriz é definida por:

$$C = S^T S / m \quad (3.3)$$

Os valores da matriz C (c_{ij}) são uma medida que representam o quanto as i -ésimas e j -ésimas coordenadas estão de acordo com os seus valores médios. Desta forma, se temos

um valor positivo para c_{ij} , ambas as dimensões crescem juntas, caso contrário, elas crescem em proporções inversas. Se esse valor é igual a zero, as dimensões são independentes.

A matriz de covariância C possui a seguinte configuração: ela é simétrica e possui valores reais. Desde forma, ela possui n autovetores e autovalores. Os autovetores são mutualmente ortogonais uns aos outros. Sendo assim, eles podem ser usados como o vetor base a esse espaço dimensional.

Para realizarmos o cálculo desses autovetores e autovalores, aplicamos diretamente na nossa matriz de dados M a decomposição dos valores singulares (SVD) (BLANZ, 2000) e desta forma, não precisamos criar a matriz de covariância C . Usando o SVD, a matriz de dados M pode ser fatorizada como:

$$M = U * S * V^T \quad (3.4)$$

onde, as colunas de V são os autovetores de $M^T * M$, as colunas de U são os autovetores de $M * M^T$ e os autovalores de $M^T * M$

Desta forma, quando aplicamos o algoritmo SVD diretamente na matriz de dados M , seus autovetores serão a matriz V e os autovalores serão a matriz S e assim, conseguimos criar um modelo PCA dado o conjunto M de diferenças da média. O modelo PCA é composto do vetor de média, dos autovetores e dos autovalores.

3.3 O Algoritmo dos K Vizinhos Mais Próximos

O algoritmo dos k vizinhos mais próximos (em inglês, *K Nearest Neighbour - KNN*) é utilizado nas mais diversas áreas devido a sua simplicidade de implementação. Nele, dado um grupo de treinamento e um grupo de teste, conseguimos baseados nos exemplos de treino, classificar os objetos que estão mais próximos a ele no espaço de características.

Para exemplificar o funcionamento do *KNN*, vamos supor uma nuvem de pontos S_i representada pelos pentágonos vermelhos com n pontos e uma nuvem de pontos P_i representada pelos círculos azuis com m pontos, queremos encontrar as correspondências entre os pontos conforme podemos observar na Figura 3.2.

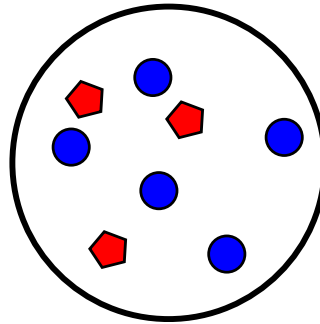


Figura 3.2: Representação no espaço de uma nuvem de pontos S_i de tamanho n representada pelos pentágonos vermelhos e de uma nuvem de pontos P_i de tamanho m representada pelos círculos azuis. Os pentágonos vermelhos representam o grupo de treinamento e os círculos azuis representam o grupo de teste.

A correspondência entre os pontos é achada baseada na menor distância entre o ponto alvo e sua origem. Por exemplo, dado um ponto S_i , marcado com uma interrogação e $k = 3$, queremos saber quais são os 3 P_i pontos mais próximos do ponto S_i como podemos observar na Figura 3.3.

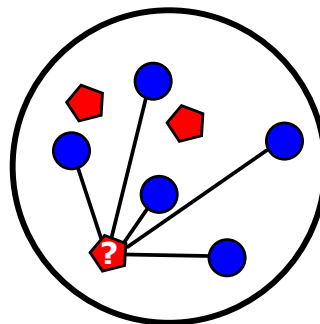


Figura 3.3: Dado um ponto em S_i com $k = 3$ queremos saber os P_i pontos mais próximos do ponto S_i .

Todas distâncias entre os pontos S_i e os pontos P_i são iterativamente calculadas e os k pontos mais próximos de S_i são considerados seus pontos correspondentes conforme podemos observar na Figura 3.4.

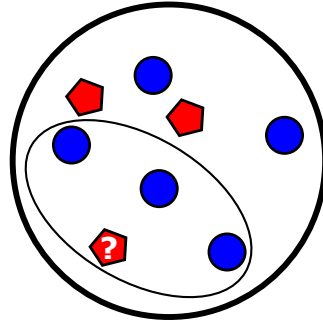


Figura 3.4: k P_i pontos correspondentes ao ponto S_i .

3.4 O Algoritmo Iterativo de Pontos Mais Próximos

O algoritmo iterativo de pontos mais próximos (em inglês, *Iterative Closest Point - ICP*), descrito por (BESL; MCKAY, 1992), tem como objetivo geral estimar uma rotação e uma translação, a cada iteração, que minimizem as distâncias entre nuvens de pontos.

Dado como entrada duas nuvens de pontos \mathbf{M} e \mathbf{D} , respectivamente as nuvens preta e vermelha da Figura 3.5 ambas com pontos definidos em \mathbb{R}^3 , cada uma com N_m e N_d pontos respectivamente, precisamos estimar transformações de forma que possamos alinhar da melhor forma possível o conjunto de pontos de \mathbf{D} ao conjunto modelo \mathbf{M} .

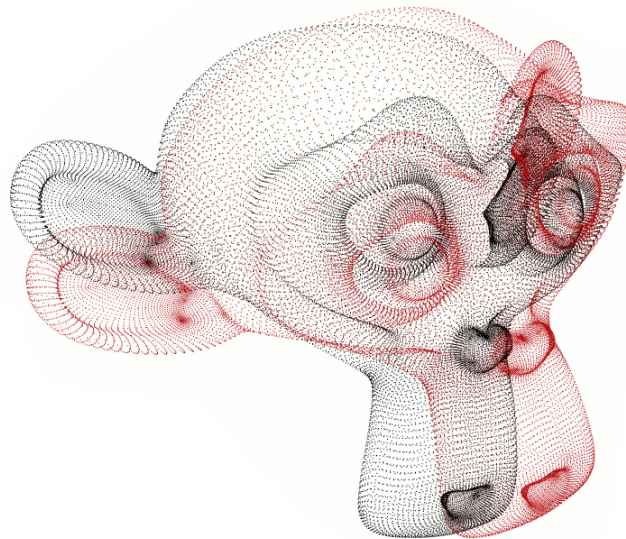


Figura 3.5: Nuvens de pontos \mathbf{M} (em preto) \mathbf{D} (em vermelho) desalinhadas.¹

Inicialmente, precisamos determinar a correspondência de pontos entre os dois conjuntos. Para tal, utilizamos o classificador *KNN* que tem seus fundamentos abordados no Capítulo 3.3. Em linhas gerais, para acharmos as correspondências, utilizamos a distância euclidiana, logo, para cada $\vec{d} \in \mathbf{D}$ definimos como ponto correspondente o ponto $\vec{m} \in \mathbf{M}$ de menor distância euclidiana.

Seja a função de distância:

$$d(\vec{d}, M) = \min_{\vec{m} \in M} (||\vec{m} - \vec{d}||),$$

Podemos denotar por \mathbf{Y} o conjunto de pontos resultantes e C como um operador de ponto mais próximo. Logo, temos que:

$$\mathbf{Y} = C(\mathbf{D}, \mathbf{M}).$$

Devemos salientar que cada ponto $\vec{d}_i \in \mathbf{D}$ possui um correspondente $\vec{r}_i \in \mathbf{Y}$. Sendo assim, ambos os conjuntos possuem o mesmo número N_p de pontos.

O próximo passo do algoritmo é a definição de uma matrix \mathbf{T} de transformação 4×4 que contenham os valores de rotação e translação para que \mathbf{M} se alinhe com \mathbf{D} . Para isso utilizamos o conceito de quartérnios.

Um quartérnio é um vetor em \mathbb{R}^4 que representa um arco em \mathbb{R}^3 (FILHO, 2010). A unidade quartérnio é dada pela quádrupla:

$$\vec{q}_r = [q_0 \ q_1 \ q_2 \ q_3]^t, \quad q_0 \geq 0 \quad e \quad q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1 \quad (3.5)$$

Definido o vetor \vec{q}_r , associamos à ele um vetor $\vec{q}_t = [q_4 \ q_5 \ q_6]^t$ de translação. Desta forma, temos um vetor completo de registro de estado definido por $\vec{q} = [\vec{q}_r | \vec{q}_t]$. Para chegarmos a este vetor, precisamos realizar a operação quartérnio de mínimos quadrados (\mathbf{Q}) que é

¹Disponível em: <http://pointclouds.org/documentation/tutorials/interactive_icp.php>. Acesso em agosto, 2016.

definida por:

$$\vec{q} = \mathbf{Q}(\mathbf{M}, \mathbf{Y})$$

Para encontrarmos o operador \mathbf{Q} , precisamos definir os centróides $\vec{\mu}_m$ de \mathbf{M} e $\vec{\mu}_y$ de \mathbf{Y} . Logo, temos que:

$$\vec{\mu}_m = \frac{1}{N_p} \sum_{i=0}^{N_p} \vec{m}_i \quad \vec{\mu}_y = \frac{1}{N_p} \sum_{i=0}^{N_p} \vec{y}_i \quad (3.6)$$

Definidos os centróides, calculamos a matriz de covariância cruzada de \mathbf{M} e \mathbf{Y} que é dada por:

$$\Sigma_{my} = \frac{1}{N_p} \sum_{i=0}^{N_p} (\vec{m}_i \vec{y}_i^t) - (\vec{\mu}_m \vec{\mu}_y^t) \quad (3.7)$$

A partir da equação 3.7, definimos uma matriz $\mathbf{A} = \Sigma_{my} - \Sigma_{my}^t$ e utilizamos os componentes cíclicos desta para formarmos um vetor coluna $\Delta = [A_{23} \ A_{31} \ A_{12}]^t$. Precisamos também definir a soma da diagonal principal da matriz de covariância Σ , em matemática definida como traço. Esta é definida por:

$$tr(\Sigma_{my}) = \Sigma_{11} + \Sigma_{22} + \Sigma_{33} \quad (3.8)$$

Através das equações 3.6, 3.7 e 3.8 podemos definir a matriz de operador de mínimos quadrados $\mathcal{Q}(\Sigma_{my})$ como:

$$\mathcal{Q}(\Sigma_{my}) = \begin{bmatrix} tr(\Sigma_{my}) & \Delta^t \\ \Delta & \Sigma_{my} + \Sigma_{my}^t - tr(\Sigma_{my})I_3 \end{bmatrix}$$

onde I_3 é a matriz identidade 3×3 .

O quartérnio de rotação \vec{q}_r é descrito como o autovetor associado ao maior autovalor de $\mathcal{Q}(\Sigma_{my})$, que é também considerado uma rotação ótima (YAMADA et al., 2013). Com estas

informações, construímos a matriz de rotação do quatérnio \vec{q}_r :

$$\mathbf{R}(\vec{q}_r) = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & q_0^2 + q_2^2 - q_1^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 + q_3^2 - q_1^2 - q_2^2 \end{bmatrix}$$

Utilizando os centróides da equação 3.6 e a matriz $\mathbf{R}(\vec{q}_r)$, calculamos o vetor de translação, que é dado por:

$$\vec{q}_t = \mu_y - \mathbf{R}(\vec{q}_r)\mu_d$$

Finalmente, montamos a matriz final de transformação 4×4 :

$$T = \begin{bmatrix} \mathbf{R}(\vec{q}_r)_{11} & \mathbf{R}(\vec{q}_r)_{12} & \mathbf{R}(\vec{q}_r)_{13} & \vec{q}_{t1} \\ \mathbf{R}(\vec{q}_r)_{21} & \mathbf{R}(\vec{q}_r)_{22} & \mathbf{R}(\vec{q}_r)_{23} & \vec{q}_{t2} \\ \mathbf{R}(\vec{q}_r)_{31} & \mathbf{R}(\vec{q}_r)_{32} & \mathbf{R}(\vec{q}_r)_{33} & \vec{q}_{t3} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A matriz T é aplicada a todos os pontos do conjunto \mathbf{D} .

O algoritmo é repetido até que atinja um número máximo de iterações ou a medida de erro ultrapassar um limite pré-estabelecido. Podemos observar na Figura 3.6 as nuvens de pontos \mathbf{M} e \mathbf{D} devidamente alinhadas.

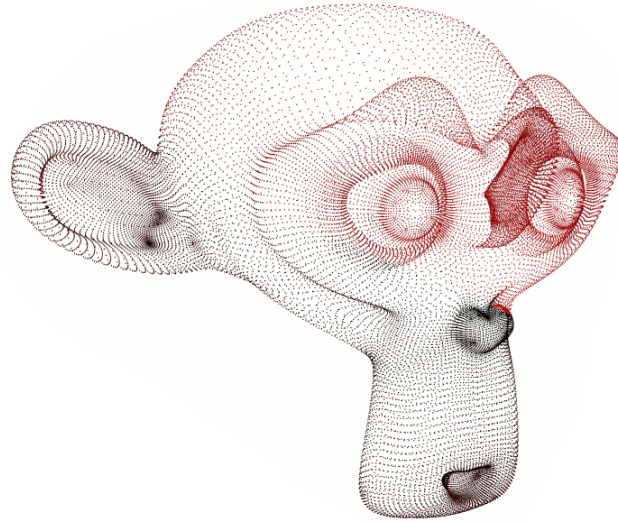


Figura 3.6: Nuvens de pontos \mathbf{M} (em preto) \mathbf{D} (em vermelho) devidamente alinhadas após a aplicação, a cada iteração, da matriz T no conjunto de pontos \mathbf{D} .²

3.5 A Otimização por Descida de Gradiente

A otimização por descida de gradiente (em inglês, *Gradient Descent Optimization*) é um algoritmo de primeira ordem, por somente calcular a derivada primeira de funções, desenvolvido para que o mínimo local de uma função seja achado. Nós começamos o método dando uma estimativa inicial para a solução e calculamos o gradiente daquela função no ponto. Depois de calculado o gradiente, nós saltamos com a solução na direção oposta ao resultado do gradiente e repetimos o processo até que o gradiente convirja para o mais próximo possível de zero.

Vamos supor que queremos achar o mínimo de uma função $f(x)$ qualquer. Então, damos um valor inicial x_0 para x e calculamos a derivada primeira de f (∇f). Intuitivamente, o gradiente resultará na inclinação da curva onde x e a sua respectiva direção apontará para o crescimento da função. Como queremos achar o mínimo valor, mudamos o valor de x para a

²Disponível em: <http://pointclouds.org/documentation/tutorials/interactive_icp.php>. Acesso em agosto, 2016.

direção oposta:

$$x_{k+1} = x_k - \delta \nabla f(x_k) \quad (3.9)$$

O valor δ é um valor que geralmente varia entre 0 e 1, dependendo da função, que garante a estabilidade e que faz com que o algoritmo realize pequenos saltos afim de achar o valor ótimo. Uma escolha correta para δ garante que $f(x_{k+1}) \leq f(x_k)$.

Para exemplificar a otimização por descida de gradiente, vamos definir uma função $f(x) = x^2$ com $x_0 = 0.8$. Vamos observar o comportamento da minimização da função mencionada com os seguintes valores para δ : 0 (Figura 3.7a), 0.1 (Figura 3.7b), 0.4 (Figura 3.7c) e 0.9 (Figura 3.7d).

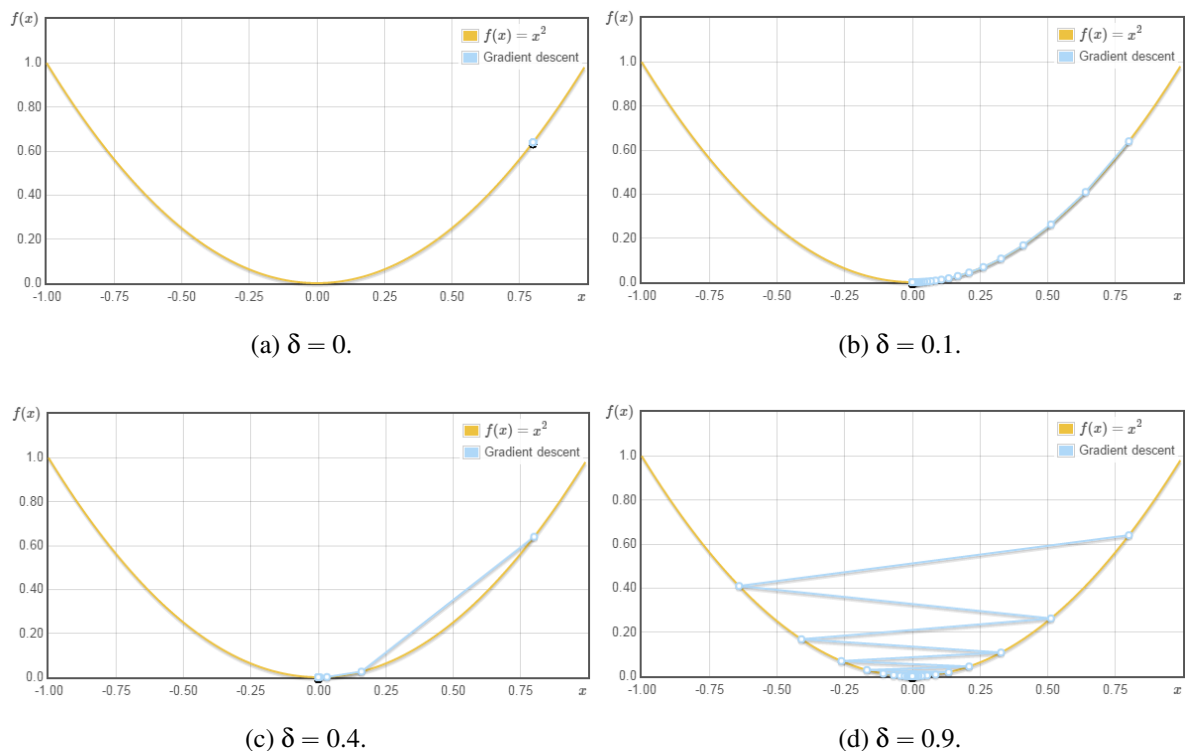


Figura 3.7: Minimização da função $f(x) = x^2$ com $x_0 = 0.8$ para diferentes valores de δ .

Conforme podemos observar, para $\delta = 0$ não houve minimização pois não houveram saltos. Com $\delta = 0.1$ as soluções parciais encontradas seguem suavemente a curva da função com pequenos saltos até atingir o valor mínimo, com $\delta = 0.4$ as soluções parciais já não são

tá suaves e os saltos das soluções parciais já não são tão pequenos e para $\delta = 0.9$ as soluções parciais não são suaves e os saltos são em zigue-zague.

3.6 Spline Placa-Fina

A Spline Placa-Fina (em inglês, *Thin-Plate Spline (TPS)*) foi proposta por (BOOKSTEIN, 1989) e faz alusão a minimização da energia de uma curvatura em uma placa de metal. Neste trabalho, usamos a TPS para criar mapas de interpolação entre dois conjuntos de pontos, ou seja, geramos funções de mapeamento que relacionam os pontos de um conjunto ao outro. Por exemplo, dado o conjunto de pontos A que é o sistema de origem e o conjunto de pontos B que é o sistema de destino, devidamente relacionado um ao outro, devemos achar uma transformação afim que represente a deformação dos pontos de A com relação aos pontos de B . Na Figura 3.8 temos em (a) a radiografia de um crânio humano e em (b) os pontos que representam um crânio de um chimpanzé. Os pontos de (a) e (b) estão todos relacionados e após o processo the *thin-plate* temos como resultado (c) que são os pontos de (a) aplicados à deformação de (b).

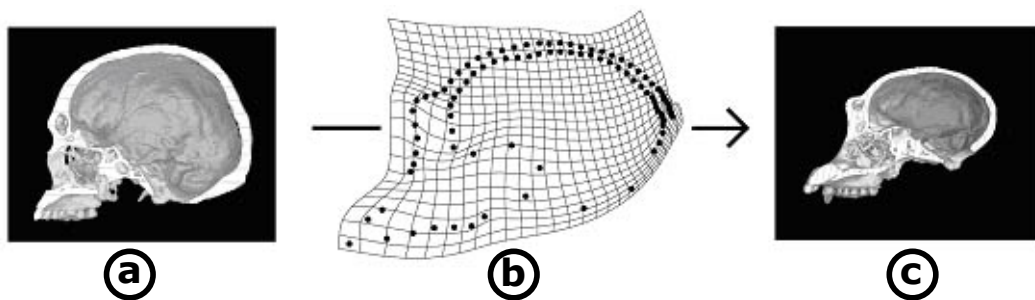


Figura 3.8: Em (a) temos a imagem de um crânio humano, em (b) os pontos de um crânio de um chimpanzé e em (c) temos o resultado após o processo de *thin-plate*.

A formulação proposta por (BOOKSTEIN, 1989) é para problemas bidimensionais porém, neste trabalho aplicamos este método em um conjunto de pontos tridimensionais. Desta forma, vamos considerar que temos dois conjuntos de pontos S_{or} e S_{op} e que seus pontos possuem uma relação entre si. O primeiro conjunto, que é o sistema de origem é dado por

$S_{or} = \{P_1, P_2, \dots, P_n\}$ onde $P_n = (x_n, y_n, z_n)$ e o segundo conjunto, que é o sistema de destino é dado por $S_{op} = \{V_1, V_2, \dots, V_n\}$ onde $V_n = (x_n, y_n, z_n)$ e n é o número de pontos de S_{or} e S_{op} . Para acharmos a matriz de transformação que represente a deformação dos pontos ocorridos de S_{or} para S_{op} , nós definimos as matrizes:

$$K = \begin{bmatrix} 0 & U(r_{12}) & \cdots & U(r_{1n}) \\ U(r_{21}) & 0 & \cdots & U(r_{2n}) \\ \vdots & \vdots & \ddots & \vdots \\ U(r_{n1}) & U(r_{n2}) & \cdots & 0 \end{bmatrix}_{n \times n} \quad (3.10)$$

onde K é a matriz que representa as funções de base radiais $U(r)$ onde $r_{ij} = |P_i - P_j|$, ou seja, a distância entre os pontos i e j e $U(r_{ij}) = r_{ij}^2 \log r_{ij}^2$.

Construímos também uma matriz P que é composta pelos P_i pontos do sistema de origem S_{or} :

$$P = \begin{bmatrix} 1 & x_1 & y_1 & z_1 \\ 1 & x_2 & y_2 & z_2 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & y_n & z_n \end{bmatrix}_{n \times 4} \quad (3.11)$$

Depois de definidas as matrizes K e P , compomos uma matriz L que contém a matriz K com as funções de base radial do sistema de origem S_{or} , com a matriz de pontos P em sua

forma original e transposta e uma matriz de zeros de dimensão 4×4 :

$$L = \left[\begin{array}{c|c} K & P \\ \hline P^T & O \end{array} \right]_{(n+4) \times (n+4)} \quad (3.12)$$

Definimos uma matriz V que é composta pelos V_i pontos do sistema de destino S_{op} acrescida de quatro colunas com valores nulos:

$$V = \left[\begin{array}{cccccccc} x_1 & x_2 & \cdots & x_n & 0 & 0 & 0 & 0 \\ y_1 & y_2 & \cdots & y_n & 0 & 0 & 0 & 0 \\ z_1 & z_2 & \cdots & z_n & 0 & 0 & 0 & 0 \end{array} \right]_{3 \times (n+4)} \quad (3.13)$$

Dadas as matrizes L e V criamos uma matriz R que é a multiplicação da inversa da matriz L pela matriz V . A matriz R contém os pesos e os coeficientes da função de mapeamento TPS.

$$R = L^{-1}V^T = \begin{bmatrix} w_{x1} & w_{y1} & w_{z1} \\ w_{x2} & w_{y2} & w_{z2} \\ \vdots & \vdots & \vdots \\ w_{xn} & w_{yn} & w_{zn} \\ a_{x1} & a_{y1} & a_{z1} \\ a_{xx} & a_{yx} & a_{zx} \\ a_{xy} & a_{yy} & a_{zy} \\ a_{xz} & a_{yz} & a_{zz} \end{bmatrix} \quad (3.14)$$

Cada coluna da matriz R contém w_{xi} , w_{yi} e w_{zi} , $i = 1, 2, \dots, n$ que são os respectivos pesos da função de mapeamento TPS e a_{xj} , a_{yj} e a_{zj} , $j = 1, x, y, z$ que são os coeficientes que compõem a função TPS. As funções de TPS são definidas como:

$$\begin{aligned} f_x(x, y, z) &= a_{x1} + a_{xx}x + a_{xy}y + a_{xz}z + \sum_{i=1}^n w_{xi}U(|P_i - (x, y, z)|) \\ f_y(x, y, z) &= a_{y1} + a_{yx}x + a_{yy}y + a_{yz}z + \sum_{i=1}^n w_{yi}U(|P_i - (x, y, z)|) \\ f_z(x, y, z) &= a_{z1} + a_{zx}x + a_{zy}y + a_{zz}z + \sum_{i=1}^n w_{zi}U(|P_i - (x, y, z)|). \end{aligned} \quad (3.15)$$

Definidas as funções TPS, qualquer conjunto de pontos que estejam no mesmo espaço que S_{or} e S_{op} quando aplicadas às funções TPS $f_x(x, y, z)$, $f_y(x, y, z)$ e $f_z(x, y, z)$, serão deformados baseados nos pesos e coeficientes definidos na matriz R .

Capítulo 4

Sistema Proposto

Neste capítulo vamos apresentar o nosso sistema de visualização de objetos virtuais na face do usuário desde a aquisição dos dados até a renderização da imagem final. Como podemos ver na Figura 4.1, nós utilizamos um modelo morfológico facial 3D gerado a partir de uma série de faces escaneadas. A entrada do nosso sistema é uma imagem e sua informação de profundidade correspondente capturado a partir de uma camera RGB-D de baixo custo, o Microsoft Kinect. Essa entrada é processada e geramos os pontos de interesse faciais ou *landmarks* 2D e 3D. Utilizando esses dados de entrada e o modelo morfológico facial 3D, nós otimizamos os parâmetros do modelo morfológico facial 3D utilizando o método da descida de gradiente para que este se ajuste à forma do rosto do usuário. Depois deste processo, objetos virtuais 3D são deformados baseado nas alterações ocorridas entre o modelo morfológico facial inicial e o otimizado. Finalmente geramos uma imagem composta do objeto virtual acima da imagem original do usuário.

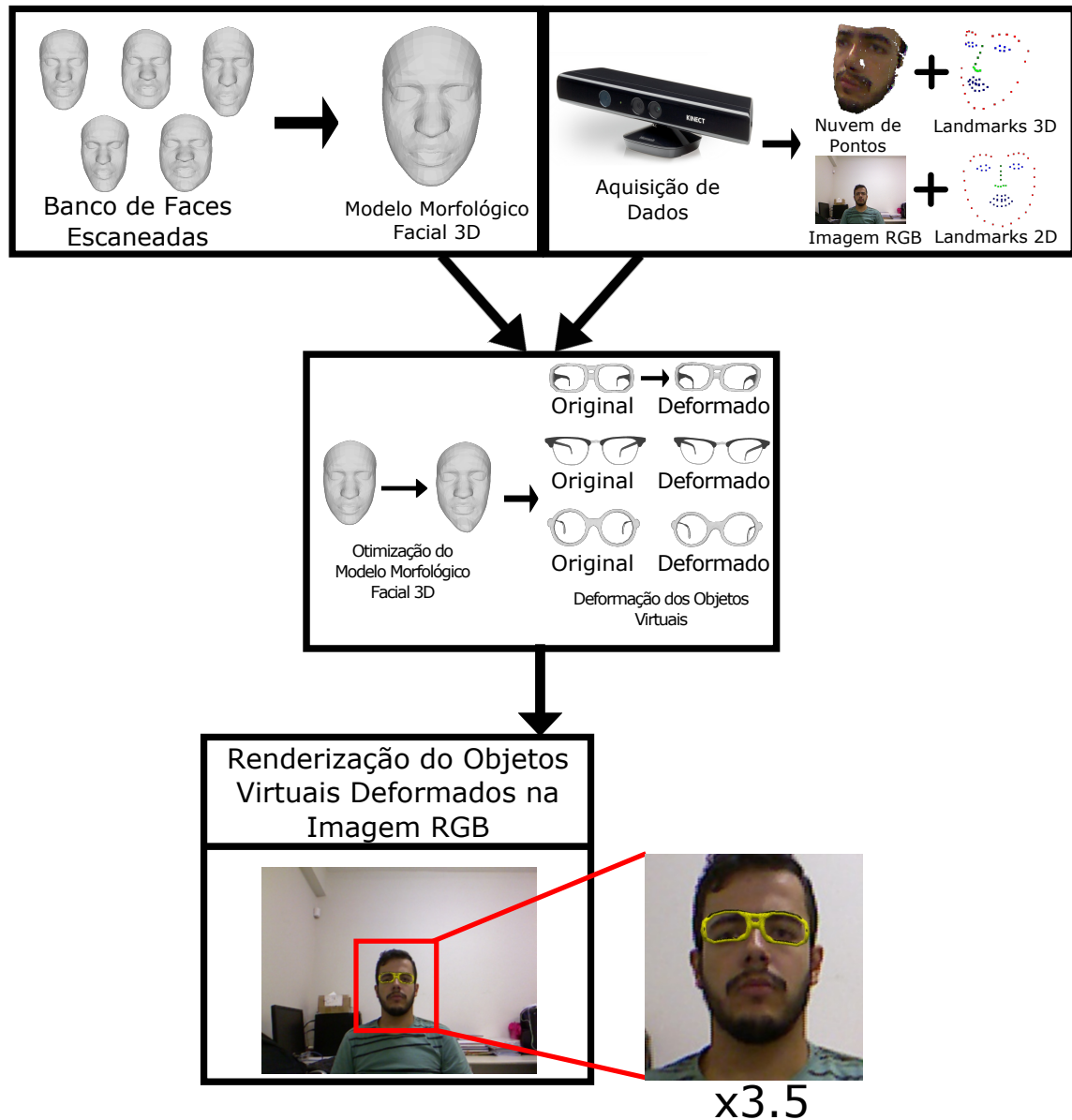


Figura 4.1: Visão geral do sistema de visualização de objetos virtuais 3D na face do usuário. Inicialmente geramos um modelo morfológico facial 3D masculino e feminino baseado em uma série de faces escaneadas de milhares de indivíduos. Como entrada do nosso sistema, temos uma imagem RGB e suas informações de profundidade capturadas através de uma câmera RGB-D de baixo custo além dos pontos de interesse faciais 2D e 3D. Essas informações de entrada são utilizadas para otimizar os parâmetros do nosso modelo morfológico virtual 3D para que esse se alinhe a forma do rosto do usuário. No final, deformamos o objeto virtual 3D baseado na deformação do modelo e o projetamos para 2D e geramos uma imagem da face do usuário com o respectivo objeto virtual.

4.1 Criação do Modelo Morfológico 3D

Baseado em um banco de dados de faces escaneadas 3D (ROBINETTE et al., 2002) (BLACKWELL et al., 2002), com expressão neutra, contendo 1.122 faces masculinas e 1.262 faces femininas registradas, nós geramos os seus respectivos modelos morfológicos faciais 3D. Todas as faces escaneadas da base de dados (ROBINETTE et al., 2002) (BLACKWELL et al., 2002) são registradas antecipadamente. O primeiro problema que precisamos lidar é o fato de todas essas faces registradas estarem em diferentes posições em \mathbb{R}^3 como podemos observar na Figura 4.2.

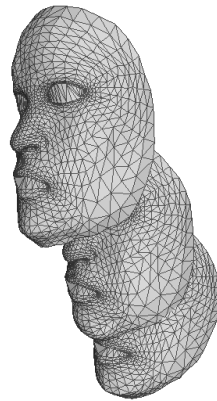


Figura 4.2: Faces escaneadas em diferentes posições em \mathbb{R}^3 .

Para contornar o problema acima mencionado, aplicamos a análise procrustes (GOWER, 1975), que tem como objetivo alinhar todas as faces escaneadas utilizando como ponto de alinhamento a primeira face registrada no banco de faces, que no caso da Figura 4.2 é a face mais ao topo. Desta forma, vamos supor que S seja o conjunto com as n faces escaneadas S_i , logo, $S = S_0, S_1, S_2, \dots, S_{n-1}$ e, $M = S_0$, ou seja, a primeira face escaneada do banco. A análise procrustes nos fornece uma matriz de translação e rotação que alinha as faces S_{i+1} à face de ponto de alinhamento M .

Após o processo de alinhamento dos modelos, aplicamos o método da análise dos componentes principais (PCA) para gerar os modelos morfológicos faciais 3D como descrito

em (BLANZ; SCHERBAUM; SEIDEL, 2007) e explicado em detalhes no Capítulo 3.2. Usando o modelo criado pelo PCA, qualquer modelo de face pode ser reconstruído como:

$$facemodel = S + \sum_{i=1}^{nComp} s_i * \beta_i \quad (4.1)$$

onde S é o modelo de face médio, $nComp$ é o número de componentes ou coeficientes do modelo. Esse número é dado pela quantidade de faces escaneadas dada para a geração do PCA. Para o caso do modelo morfológico facial masculino, podemos ter até 1.122 componentes e para o feminino 1.262. s_i são as componentes ou coeficientes e β_i são os pesos que influenciam cada componente.

Como última etapa do processo de criação, manualmente marcamos os *landmarks* 3D no modelo morfológico facial 3D, como podemos ver na Figura 4.3, para que o mesmo corresponda aos *landmarks* 2D detectados pelo nosso rastreador de face. O modelo morfológico criado contém 1411 vértices e 2768 faces triangulares.

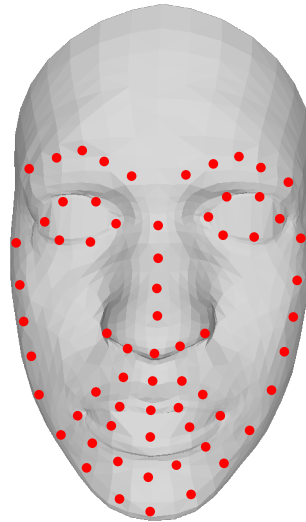


Figura 4.3: *Landmarks* 3D manualmente marcadas no modelo morfológico facial 3D.

4.2 Aquisição de Dados

O processo de aquisição de dados consiste em capturar os dados de entrada do nosso sistema de Realidade Aumentada. A aquisição foi realizada através de uma câmera RGB-D de baixo custo, o Microsoft Kinect. O advento do Microsoft Kinect permitiu a interação de pessoas com jogos eletrônicos sem a necessidade de um controle. Isso despertou o interesse de pesquisadores e entusiastas das áreas de ciência da computação, engenharia elétrica e robótica. O baixo custo do sensor e as suas capacidades abriam as portas para novas formas de interação com diferentes sistemas(CORREIA, 2013).

Na figura 4.4 podemos observar o Kinect.



Figura 4.4: O Microsoft Kinect Primeira Versão.¹

O Microsoft Kinect é composto por:

1. Um emissor de raios infravermelho e uma câmera infravermelho. Essa câmera captura imagens com resolução de 1200×960 pixels à 30 fps (NETWORK, 2010) e estas sofrem um processo de *downsample*, ou seja, o tamanho da amostra é redimensionado mantendo sempre o enésimo fator de redução(n), neste caso, com $n = 11$ gerando uma imagem infravermelho de 640×480 pixels;
2. Uma câmera RGB que captura imagens com resolução de 640×480 pixels à 30 fps . É possível setar a câmera para o modo de alta resolução, capturando imagens com resolução de 1280×1024 pixels à 10 fps (CRUZ; LUCIO; VELHO, 2012);

¹Disponível em: <<http://microsoft-careers.com/content/hardware/hardware-story-kinect/>>. Acesso em julho, 2016.

3. Um motor que faz a movimentação do Microsoft Kinect para cima e para baixo e um acelerômetro que determina a sua orientação (NETWORK, 2010);
4. O Microsoft Kinect possui ao longo de seu hardware quatro microfones para a captura de áudio fazendo com que seja possível além de gravar áudio, determinar a localização da origem do som (NETWORK, 2010).

O Microsoft Kinect nos fornece em tempo real uma imagem RGB e as informações de profundidade ou nuvem de pontos de cada frame, ambas com resolução inicial de 640×480 e necessárias para execução do nosso sistema. Além dessas informações, rastreamos e capturamos para cada frame os pontos de interesse faciais ou *landmarks* 2D para a imagem e 3D para a nuvem de pontos. Conforme mencionado anteriormente, a imagem RGB e as informações de profundidade possuem a mesma dimensão porém, conforme podemos observar na Figura 4.4 a câmera RGB e a de profundidade não estão alinhadas. Para salvarmos os pontos de interesse faciais em 3D utilizamos um método fornecido pelo *framework* de desenvolvimento do Microsoft Kinect onde, dado um ponto em 3D, nos é retornado um ponto em 2D. Se o ponto retornado for igual à algum ponto de interesse facial em 2D nós salvamos o respectivo ponto em 3D.

Na Figura 4.5 podemos observar as etapas do nosso sistema de aquisição de dados. Em resumo ele nos permite visualizar e salvar os seguintes dados: a imagem do usuário e seus respectivos *landmarks* em 2D como podemos observar na Figura 4.5(a), a nuvem de pontos e os *landmarks* em 3D como vemos na Figura 4.5(b). É importante frisar que como o que nos interessa é somente a face do usuário, realizamos um corte na nuvem de pontos baseado nos *landmarks* 3D que delimitam o contorno da face. A nuvem de pontos cortada contém em média 6000 pontos ao invés dos 300.000 pontos da nuvem de pontos inicial. Por último podemos visualizar na Figura 4.5(c) o modelo morfológico facial 3D alinhado à nuvem de pontos. A matriz de transformação que realiza este alinhamento também é salva.

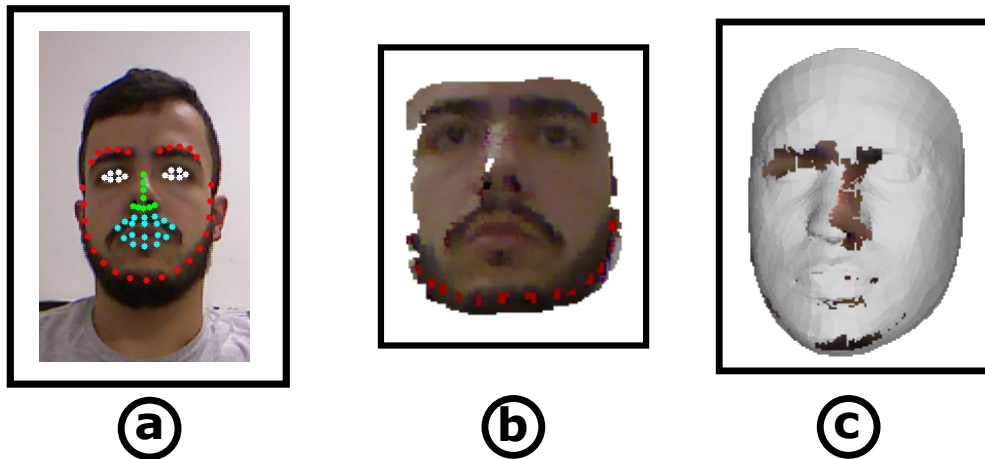


Figura 4.5: Nosso sistema de aquisição de dados. O Microsoft Kinect é utilizado para capturar uma imagem RGB e uma nuvem de pontos. Logo após, um rastreador de pontos de interesse facial é aplicado para detectar os *landmarks* na imagem RGB. Como resultado, nós geramos os *landmarks* 2D correspondendo à face na imagem RGB (a) e os *landmarks* 3D correspondendo à nuvem de pontos de entrada (b). Como podemos observar em (c), antes do processo de otimização, podemos visualizar o modelo morfológico facial 3D alinhado à nuvem de pontos através da matriz de transformação gerada entre os *landmarks* 3D manualmente marcados do modelo morfológico e os da nuvem de pontos.

4.3 Otimização do Modelo Morfológico Facial 3D

O processo de otimização do modelo morfológico facial 3D consiste em ajustá-lo à nuvem de pontos de entrada de um determinado usuário. Nós otimizamos os parâmetros do modelo morfológico facial 3D aplicando o método da descida de gradiente, conforme descrito no Capítulo 3.5. Em suma, essa otimização consiste em achar pesos β que aplicados às componentes do modelo morfológico facial 3D gere um modelo de face que se aproxime ao máximo da nuvem de pontos de entrada. Considere a função de energia D dada por

$$D = \frac{\sum_{i=1}^n \sqrt{(f_i(S) - P_i)^2}}{n} \quad (4.2)$$

onde $f_i(S)$ representa um ponto no modelo morfológico facial 3D gerado S que corresponde ao i -ésimo ponto na nuvem de pontos P , P_i representa o i -ésimo ponto da nuvem de pontos

de entrada P e n representa a quantidade de pontos da nuvem de pontos de entrada. Como vimos na Seção 4.1 e na Seção 4.2, o modelo morfológico facial 3D possui 1411 pontos e a nuvem de pontos possui aproximadamente 6000 pontos. Mais adiante apresentaremos o método utilizado para correlacionar pontos do modelo com a nuvem de pontos representado na função de energia D pela função f .

Nós utilizamos a aproximação por descida de gradiente para achar os pesos que precisam ser aplicados ao modelo morfológico facial 3D para minimizar a distância entre S e P . O algoritmo começa com um conjunto de pesos iniciais β_i que são iterativamente atualizados afim de minimizar a função de energia D . O pseudo código é mostrado no Algoritmo 1.

Algorithm 1 Gradient Descent Algorithm

```

1: function GRADDESC()
2:    $\delta \leftarrow \text{setDeltaValue}()$ 
3:   while  $\text{numIter} \leq \text{maxIter}$  OR  $\text{error} \leq \text{lastError}$  do
4:      $\text{transformationMatrix} \leftarrow \text{getTransformationMatrix}()$ 
5:      $\text{ALIGNPOINTCLOUDS}(\text{transformationMatrix})$ 
6:      $\text{indexKNN} \leftarrow \text{getClosestPoints}()$ 
7:     for  $i \leftarrow 1$  to  $nComp$  do
8:        $\text{gradient}[i] \leftarrow \text{getModelDistance}(\text{indexKNN}, \delta)$ 
9:     end for
10:    for  $j \leftarrow 1$  to  $nComp$  do
11:       $\text{weight}[j] \leftarrow \text{setWeight}(\text{gradient}[j])$ 
12:    end for
13:     $\text{model} \leftarrow \text{applyWeightOnModel}(\text{weight})$ 
14:     $\text{error} \leftarrow \text{getError}()$ 
15:  end while
16:  return error
17: end function

```

Primeiramente precisamos iniciar um valor para *delta* (δ) conforme podemos observar na linha 2 do Algoritmo 1. Esse valor indica a escala do passo do gradiente. Intuitivamente, se δ é inicializado com um valor pequeno, serão necessários mais iterações para que o algoritmo venha a convergir e podemos ficar presos a um mínimo local. Por outro lado, se δ é inicializado com um valor maior, os resultados da otimização serão mais rápidos porém com o custo de uma convergência ruim. Para achar o melhor valor para δ , foram rodados uma série de experimentos de *tuning* do sistema que serão descritos no Capítulo 5.1.

A cada iteração do algoritmo, nós achamos uma matriz de transformação para alinhar o modelo morfológico facial 3D com a nuvem de pontos conforme podemos observar nas linhas 4 e 5 do Algoritmo 1. Esse alinhamento garante uma melhor minimização da nossa função de energia. Para a realização desta tarefa, utilizamos uma transformação rígida, descrita na Seção 3.4, que é calculada baseada nos *landmarks* 3D do modelo morfológico facial 3D e da nuvem de pontos (RUSU; COUSINS, 2011).

O modelo morfológico facial 3D e a nuvem de pontos possuem números de pontos diferentes conforme mencionamos anteriormente e para que seja possível calcular a função de energia, precisamos achar as correspondências entre os pontos, como podemos ver na linha 6 do Algoritmo 1. Para acharmos essas correspondências, utilizamos a aproximação de pontos dos vizinhos mais próximos (MUJA; LOWE, 2009), descrito no Capítulo 3.3, onde K representa a quantidade de vizinhos desejados.

Neste trabalho, nós aplicamos a técnica *KNN* na seguinte configuração: nuvem de pontos como alvo e o modelo morfológico facial 3D como origem, ou seja, queremos saber quais pontos do modelo morfológico facial S correspondem aos pontos da nuvem de pontos P , como podemos observar na Figura 4.6. Conforme mencionado no Capítulo 4.2, a nuvem de pontos possui aproximadamente 6000 pontos e o modelo morfológico facial 3D apenas 1411 pontos o que gera uma equivalência de aproximadamente 4 pontos para 1. Desta forma, quando setamos o modelo morfológico facial 3D como origem, estamos aumentando a sua dimensionalidade, ou seja, 1 ponto do modelo poderá corresponder a 4 pontos na nuvem de

pontos. Após achadas as correspondências entre os pontos, conseguimos calcular a função de energia D pois os pontos estão devidamente relacionados.

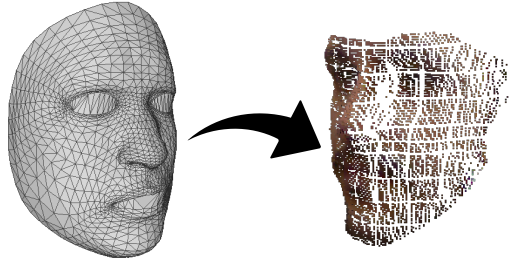


Figura 4.6: KNN na configuração nuvem de pontos como alvo e modelo morfológico facial 3D como origem.

O próximo passo do nosso algoritmo é gerar os valores de gradiente para cada componente do nosso modelo morfológico facial 3D conforme podemos ver nas linhas de 7 a 9 do Algoritmo 1. O cálculo do gradiente é dado por:

$$gradient_i = D_{i+} - D_{i-} \quad (4.3)$$

onde $gradient_i$ é o vetor de gradiente, D_{i+} e D_{i-} são as distâncias euclidianas médias entre S e P para cada componente do modelo morfológico facial 3D. Em D_{i+} , os pesos β de S são perturbados por $+\delta$ e em D_{i-} , os pesos β de S são perturbados por $-\delta$.

Após calculado o vetor de gradientes, nós atualizamos o vetor de pesos β como podemos ver nas linhas de 10 a 12 do Algoritmo 1. Essa atualização é dada por:

$$\beta_{i+1} = \beta_i * \delta * gradient \quad (4.4)$$

Finalmente os respectivos pesos β são aplicados à sua respectiva componente no modelo morfológico facial 3D e o erro é atualizado como mostrado nas linhas 13 e 14 do Algoritmo 1, respectivamente.

Todos esses passos são repetidos até que o número de iterações seja menor que um número de iterações pré-definido ou enquanto o erro continue a decrescer. Como saída do processo de otimização, nós obtemos a versão otimizada do modelo morfológico facial 3D, ou seja, a versão ajustada à nuvem de pontos de entrada como podemos ver na Figura 4.7.

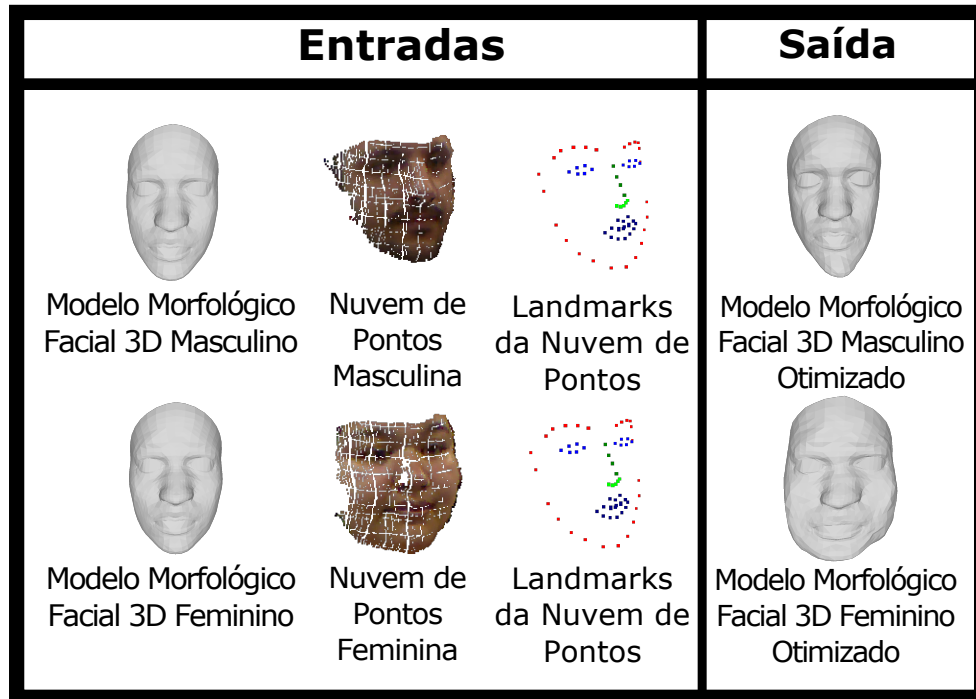


Figura 4.7: A entrada do processo de otimização são os modelos morfológicos faciais 3D masculinos e femininos, a nuvem de pontos e os seus respectivos pontos de interesse faciais 3D. Como saída, geramos o modelo morfológico facial 3D ajustado à face do usuário.

Depois de feito o processo de otimização, os objetos virtuais 3D são deformados baseados em uma função de aproximação de base radial, para que o objeto se ajuste com mais precisão à forma do rosto do usuário.

4.4 A Deformação dos Objetos Virtuais 3D

Dados os modelos morfológicos faciais 3D original (S^{or}) e otimizado (S^{op}), o processo de deformação consiste em deformar o objeto virtual 3D (alinhado ao modelo original de

face) baseado nas mudanças ocorridas de S^{or} para S^{op} conforme podemos observar na Figura 4.8. Este processo é realizado aplicando o método TPS (BOOKSTEIN, 1989) e descrito na Seção 3.6. Resumidamente precisamos posicionar manualmente os objetos virtuais 3D ao modelo morfológico facial 3D utilizando um software qualquer de modelagem 3D. Em seguida, são computados pesos, conforme descrito no conjunto de equações 3.15, para a função de base radial e utilizamos estes pesos para deformar os objetos virtuais. Como resultado nós geramos um modelo 3D do objeto que está melhor ajustado à S^{op} .

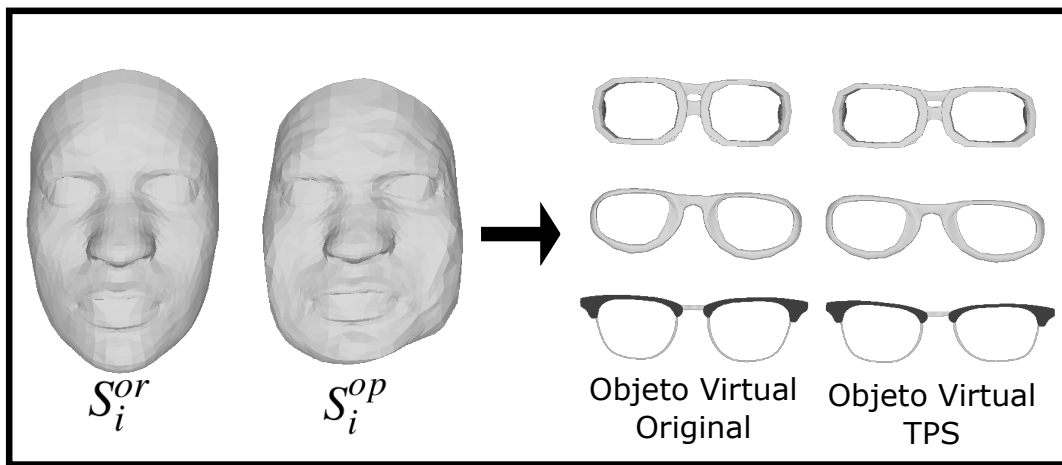


Figura 4.8: Dados os modelos morfológicos faciais 3D original e otimizado, aplicamos o método TPS e achamos uma transformação afim que representa a deformação ocorrida de S^{or} para S^{op} . Depois, qualquer conjunto de pontos que esteja no mesmo espaço que os modelos iniciais, se aplicado a eles essa transformação, sofrerão as deformações ocorridas.

4.5 A Composição da Imagem Final

A etapa da composição da imagem final consiste em renderizar a imagem RGB capturada da câmera RGB-D com o objeto virtual devidamente deformado e melhor alinhado e/ou escalado ao rosto do usuário. Como mencionado no Capítulo 4.2, a imagem capturada possui a dimensão de 640×480 e quando renderizamos o objeto virtual na face do usuário, não conseguimos observá-lo detalhadamente, conforme podemos observar na Figura 4.9.



Figura 4.9: Imagem capturada pela câmera RGB-D com dimensão de 640×480 e com o objeto virtual renderizado.

Como o foco deste trabalho é a visualização de objetos virtuais na face do usuário, optamos por realizar um corte na imagem para visualizarmos somente a área da face. Para realizarmos este corte, consideramos o centro de massa dos pontos de interesse faciais onde, depois de calculado, definimos uma distância do centro para as extremidades da face de forma que a área de corte seja quadrada e redimensionamos a imagem para 512×512 , conforme podemos observar na Figura 4.10.



Figura 4.10: Imagem capturada pela câmera RGB-D com dimensão de 640×480 em (a) e imagem com dimensão de 512×512 cortada baseado no centro de massa dos pontos de interesse faciais em (b).

O último passo da etapa de composição da imagem é a de renderização do objeto virtual

deformado em cima da imagem RGB. Para isso, nós projetamos o objeto virtual de \mathbb{R}^3 para \mathbb{R}^2 , onde dados os pontos de interesse faciais em 3D e em 2D capturados pelo nosso sistema de aquisição de dados, descrito no Capítulo 4.2, nós resolvemos um sistema linear que resulta em uma matriz de transformação \mathbf{T} que leva qualquer ponto de \mathbb{R}^3 para \mathbb{R}^2 . Como resultado, temos um objeto virtual em 2D que consiste de vértices, faces, normais e mapa UV.

O passo seguinte é renderizar o modelo 2D projetado sobre a imagem. Nós desenvolvemos um esquema de rasterização onde iteramos sobre a imagem RGB checando se o *pixel* da imagem esta dentro de alguma face do objeto virtual em 2D. Se sim, nós determinamos um peso para cada vértice dentro do triângulo utilizando coordenadas baricêntricas. Esses pesos também são utilizados para aplicarmos os efeitos de textura e iluminação no objeto.

Por fim, a imagem final é composta por um mapeamento de textura, ou seja, mapeamos cada ponto UV do mapa de textura do objeto virtual para a imagem final. Além disso aplicamos efeitos de luz ambiente e difusa ao objeto virtual em questão. Cada pixel a ser renderizado é dado por:

$$pixel_{l_i} = pixel_i * (I_a K_a + I_p K_d (N \cdot L)) \quad (4.5)$$

onde $pixel_{l_i}$ é o *pixel* corrente a ser iluminado, $pixel_i$ é o *pixel* corrente sem nenhum efeito de iluminação, I_a e I_p são as intensidades de luz ambiente e da fonte de luz, respectivamente, K_a e K_d são os componentes ambiente e difusos, respectivamente, N é o vetor normal e L é a posição da luz. Os resultados visuais são apresentados no Capítulo 5.3.

Capítulo 5

Resultados

Neste capítulo iremos apresentar nossos resultados numéricos e visuais. Inicialmente descreveremos o *tuning* do sistema, ou seja, etapa onde achamos os melhores parâmetros de otimização do modelo morfológico facial 3D. Na sequência apresentaremos os experimentos numéricos, etapa onde verificamos a performance do nosso sistema baseado em algumas medidas faciais. Por fim apresentamos os resultados visuais, ou seja, o objeto virtual deformado renderizado sobre a imagem RGB.

5.1 *Tuning* do Sistema

O processo de *tuning* do sistema consiste em rodar o nosso sistema variando dois parâmetros do processo de otimização: o número de componentes do modelo morfológico facial 3D ($nComp$) e o parâmetro delta (δ) que controla o passo do gradiente, conforme podemos observar na Tabela 5.1, afim de achar o melhor conjunto de parâmetros que minimizem a nossa função de energia D . Nosso conjunto de *tuning* consiste em um banco de faces com 14 indivíduos divididos em 7 do sexo masculino e 7 do sexo feminino. Esse banco de faces foi capturado com a câmera RGB-D Microsoft Kinect e possui pessoas das mais diferentes idades e formas de face.

Tabela 5.1: Os parâmetros do nosso sistema são, número de componentes do modelo morfológico facial 3D ($nComp$) e o parâmetro que controla o passo do gradiente δ . Essa tabela mostra os intervalos de valores e o incremento de cada intervalo durante o processo de *tuning*.

Parâmetro	Intervalo	Incremento
$nComp$	50 a 500	50
δ	0 a 1	0,1

O parâmetro $nComp$, que representa número de coeficientes do nosso modelo morfológico facial 3D foi variado de 50 a 500 componentes. Um número baixo de componentes captura menos detalhes faciais o que custa menos poder computacional. Já um número alto de componentes podem capturar mais detalhes faciais porém, será necessário maior poder de processamento.

O valor de δ altera o passo do gradiente durante o processo de otimização, conforme mencionamos no Capítulo 4.3. Se o valor δ é setado com um valor pequeno, serão necessários um número maior de iterações para a convergência do algoritmo de otimização. Em contrapartida, se δ é setado com um valor alto, os passos do gradiente serão muito grandes e podem ser instáveis.

Neste trabalho nós realizamos um processo de *tuning* para cada gênero (masculino e feminino). Para cada um nós variamos os parâmetros de *tuning* de acordo com a Tabela 5.1 e calculamos a distância entre o modelo morfológico facial 3D e a nuvem de pontos para cada indivíduo do conjunto de *tuning* do sistema. Em seguida, nós calculamos a distância média entre todos os indivíduos do mesmo sexo:

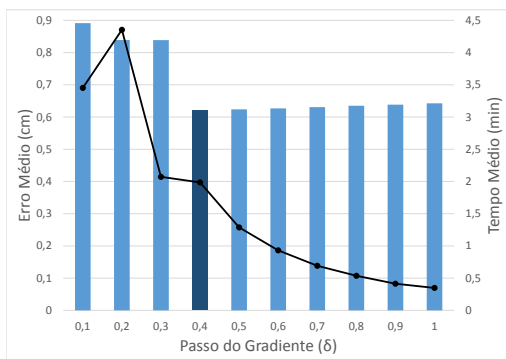
$$DM = \frac{\sum_{j=1}^{nInd} \left(\frac{\sum_{i=1}^n \sqrt{(f_i(S_j) - P_{ij})^2}}{n} \right)}{nInd} \quad (5.1)$$

onde DM é a distância média entre o modelo de face reconstruído e a nuvem de pontos,

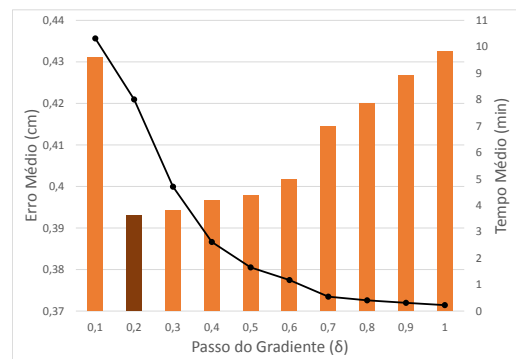
$nInd$ é o número de indivíduos do conjunto de *tuning*, n é o número de pontos da nuvem de pontos P , $f_i(S_j)$ representa um ponto no modelo morfológico facial 3D gerado S do j -ésimo indivíduo que corresponde ao i -ésimo ponto na nuvem de pontos P do j -ésimo indivíduo e P_{ij} é o i -ésimo ponto da nuvem de pontos do j -ésimo indivíduo.

Por padrão o Microsoft Kinect captura seus pontos utilizando a unidade de comprimento do Sistema Internacional, ou seja, o metro. Para apresentar valores condizentes com os valores reais da face, aplicamos a conversão de metros para centímetros para a apresentação dos resultados de *tuning* e para os resultados numéricos apresentados na Seção 5.2.

Nos gráficos a seguir, iremos apresentar para cada conjunto de componentes dos modelos morfológicos faciais 3D masculino e feminino, seguindo o intervalo e o incremento apresentado na Tabela 5.1, os erros médios obtidos para cada valor δ variado (gráfico em barra) e seu respectivo tempo médio de otimização (gráfico em linha). As cores em destaque no gráfico representam o menor erro encontrado para um determinado *delta*.

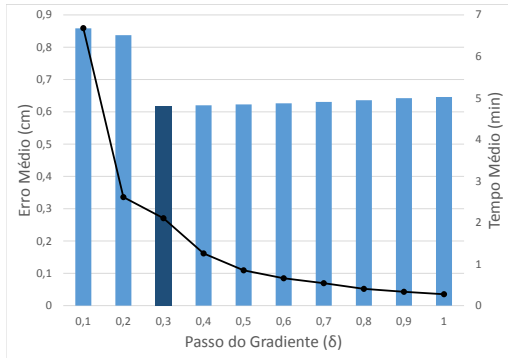


(a) Conjunto masculino.

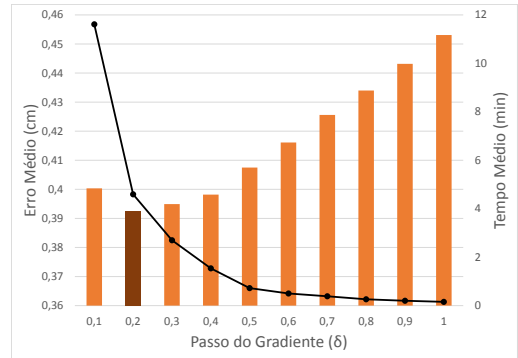


(b) Conjunto feminino.

Figura 5.1: *Tuning* do sistema para ambos os gêneros com $nComp = 50$.

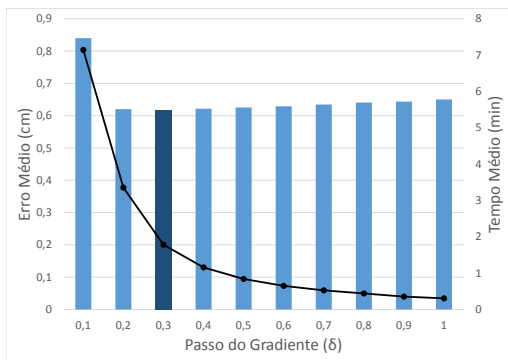


(a) Conjunto masculino.

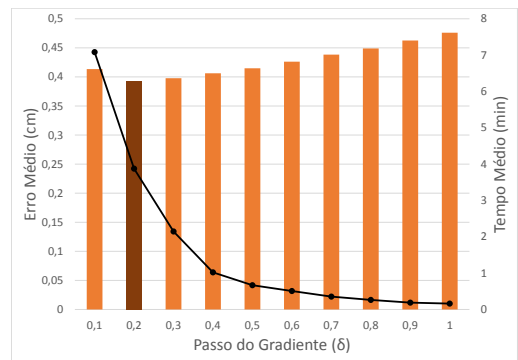


(b) Conjunto feminino.

Figura 5.2: *Tuning* do sistema para ambos os gêneros com $nComp = 100$.

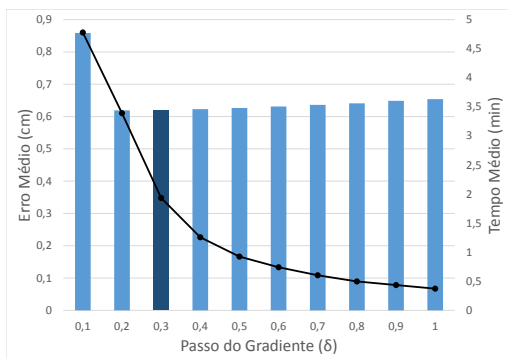


(a) Conjunto masculino.

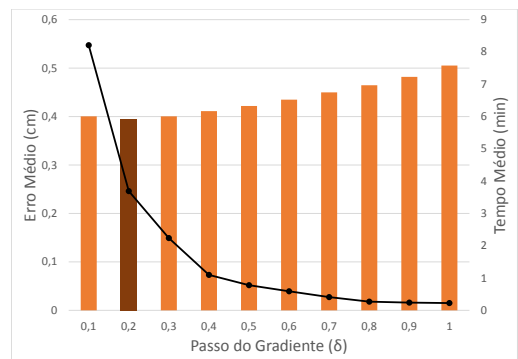


(b) Conjunto feminino.

Figura 5.3: *Tuning* do sistema para ambos os gêneros com $nComp = 150$.

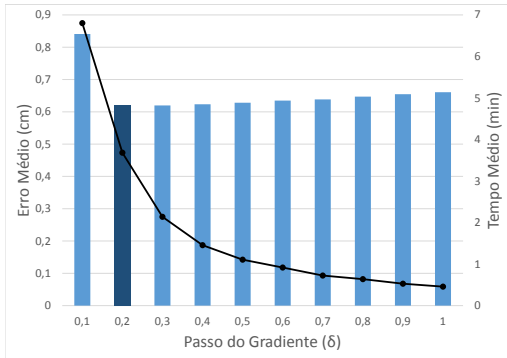


(a) Conjunto masculino.

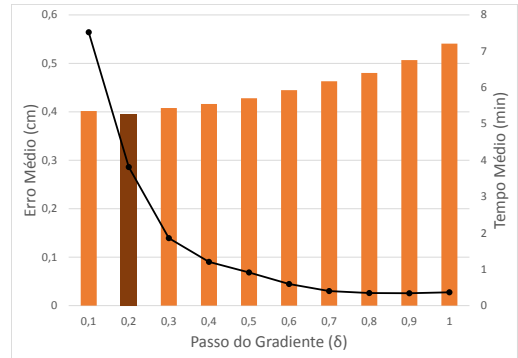


(b) Conjunto feminino.

Figura 5.4: *Tuning* do sistema para ambos os gêneros com $nComp = 200$.

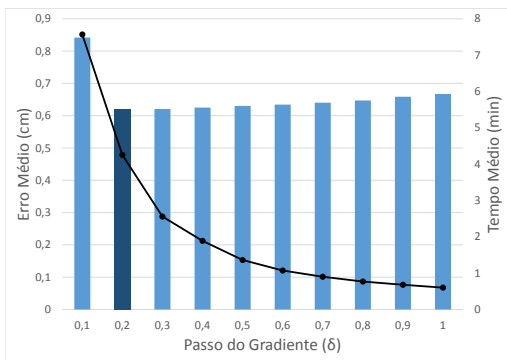


(a) Conjunto masculino.

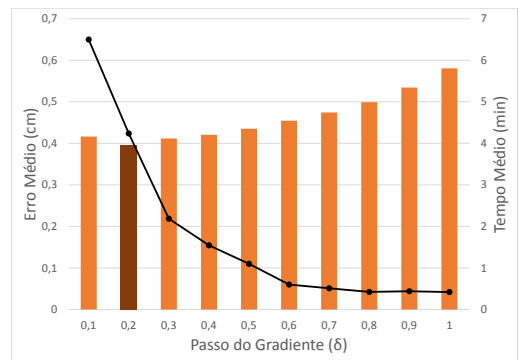


(b) Conjunto feminino.

Figura 5.5: *Tuning* do sistema para ambos os gêneros com $nComp = 250$.

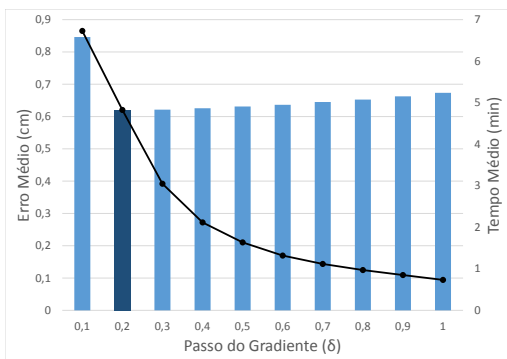


(a) Conjunto masculino.

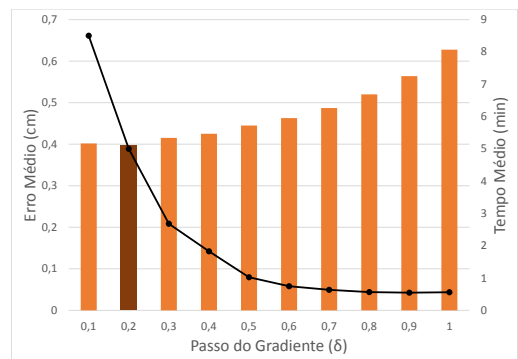


(b) Conjunto feminino.

Figura 5.6: *Tuning* do sistema para ambos os gêneros com $nComp = 300$.

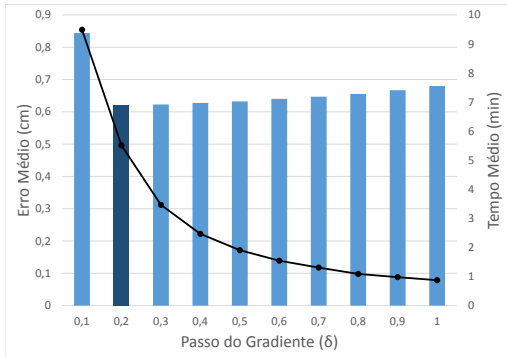


(a) Conjunto masculino.

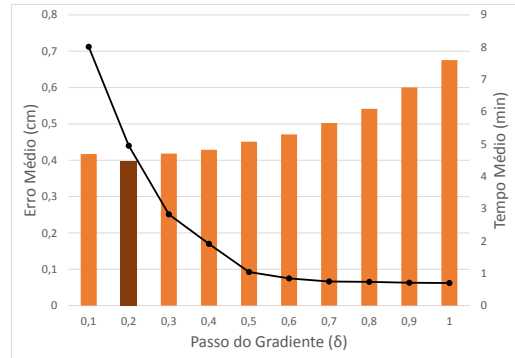


(b) Conjunto feminino.

Figura 5.7: *Tuning* do sistema para ambos os gêneros com $nComp = 350$.

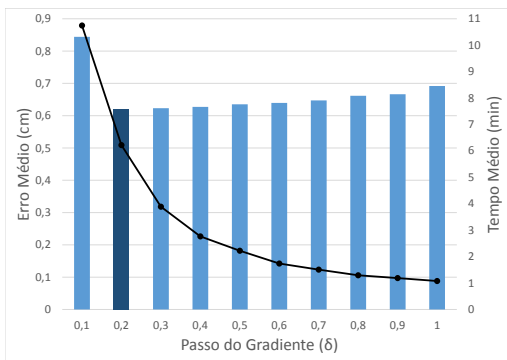


(a) Conjunto masculino.

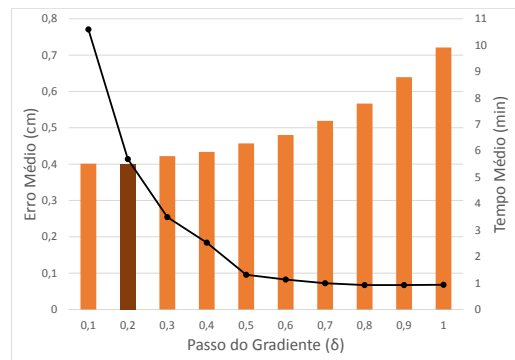


(b) Conjunto feminino.

Figura 5.8: *Tuning* do sistema para ambos os gêneros com $nComp = 400$.

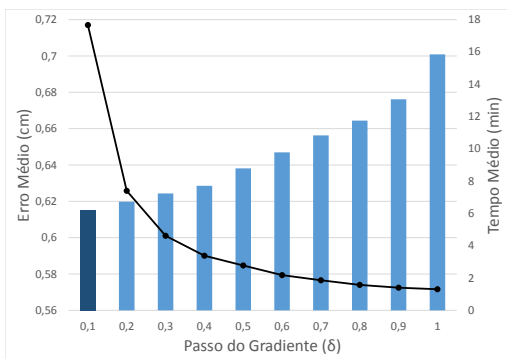


(a) Conjunto masculino.

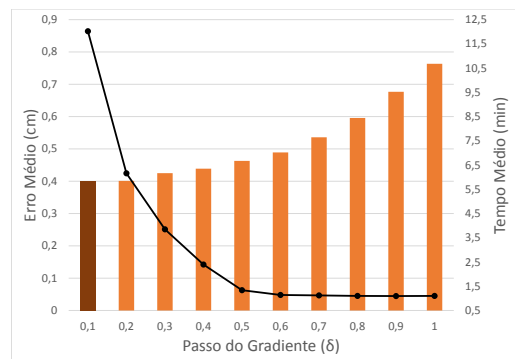


(b) Conjunto feminino.

Figura 5.9: *Tuning* do sistema para ambos os gêneros com $nComp = 450$.



(a) Conjunto masculino.



(b) Conjunto feminino.

Figura 5.10: *Tuning* do sistema para ambos os gêneros com $nComp = 500$.

Analisando os gráficos apresentados, podemos concluir que valores δ baixos, ou seja, valores no intervalo de 0,2 à 0,4, apresentaram os menores erros médios. Podemos observar também que a medida que aumentamos o valor δ o erro médio sobe em todos os casos e o tempo médio de otimização diminui. Conseqüentemente esses resultados nos leva a escolher um desses valores para gerarmos os resultados visuais do nosso sistema apesar de, conforme mencionado no Capítulo 4.3, quanto menor o δ maior será o número de iterações para que a otimização possa convergir.

As Tabelas 5.2 e 5.3 mostram os menores erros médios encontrados para cada conjunto de componentes, juntamente com seu respectivo valor δ e o tempo médio de processamento, para os indivíduos masculinos e femininos, respectivamente. O menor erro médio está marcado em cinza em ambas as tabelas. Podemos observar que variações de milésimos no erro médio, resulta em diferenças significativas no tempo de processamento. Desta forma, nós optamos por escolher o conjunto de parâmetros *nComp* e δ que apresentaram, dentre os menores erros médios obtidos, o menor tempo médio de otimização.

Na Tabela 5.2 temos marcado em azul os parâmetros que apresentaram o menor tempo médio e na Tabela 5.3 os parâmetros que representam o menor tempo médio estão marcados em laranja.

Tabela 5.2: Menores erros médios encontrados para o conjunto de componentes dos indivíduos masculinos. Para cada componente, mostramos o valor δ que possui o menor erro médio e o seu respectivo tempo médio de processamento.

<i>nComp</i>	δ	Erro Médio (em cm)	Tempo Médio (em min)
50	0,4	0,620513	1,98472
100	0,3	0,617995	2,10479
150	0,3	0,618432	1,78284
200	0,3	0,618752	1,93368
250	0,2	0,619595	3,6811
300	0,2	0,619939	4,25331
350	0,2	0,619822	4,82472
400	0,2	0,61931	5,51794
450	0,2	0,619516	6,22108
500	0,1	0,615366	17,6647

Tabela 5.3: Menores erros médios encontrados para o conjunto de componentes dos indivíduos femininos. Para cada componente, mostramos o valor δ que possui o menor erro médio e o seu respectivo tempo médio de processamento.

$nComp$	δ	Erro Médio (em cm)	Tempo Médio (em min)
50	0,2	0,393049	8,0085
100	0,2	0,392522	4,59527
150	0,2	0,393701	3,87657
200	0,2	0,394618	3,69109
250	0,2	0,395178	3,81389
300	0,2	0,396099	4,2372
350	0,2	0,397205	5,00018
400	0,2	0,398836	4,95
450	0,2	0,399561	5,69113
500	0,1	0,400739	12,021

5.2 Experimentos Numéricos

Os experimentos numéricos foram realizados para verificar com que precisão o nosso sistema consegue ajustar o modelo morfológico facial 3D das nuvens de pontos dos nossos usuários de teste. Essas medidas são geradas através da seleção de alguns pontos de interesse faciais na nuvem de pontos juntamente com os seus correspondentes no modelo morfológico

facial 3D.

Utilizando esses pontos de interesse faciais, nós calculamos quatro medidas da face que são dadas pela distância euclidiana entre os pontos de interesse faciais na nuvem de pontos e seus correspondentes no modelo morfológico facial 3D, conforme podemos observar na Figura 5.11. Essas medidas foram escolhidas pois apresentam as grandes variações da forma da face.

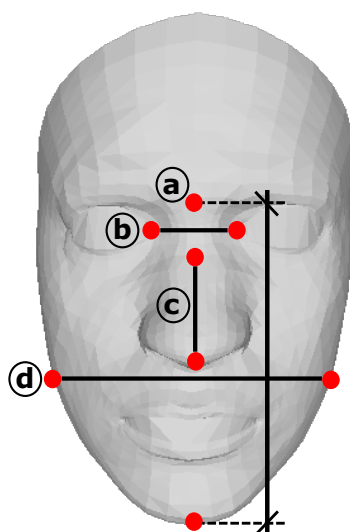


Figura 5.11: Oito pontos de interesse faciais utilizados para avaliar a precisão numérica do nosso sistema. As medidas são: (a) distância entre o meio dos olhos e o queixo, (b) distância entre os olhos, (c) tamanho do nariz e (d) distância entre as bochechas.

Nas Tabelas 5.4 e 5.5 podemos observar os resultados das medidas para o conjunto de teste dos indivíduos masculinos e femininos, respectivamente. Na primeira coluna temos as letras, que representam cada medida conforme podemos observar na Figura 5.11. Na sequência temos a medida entre os pontos de interesse faciais da nuvem de pontos (mPontos). Em seguida temos a distância correspondente do modelo morfológico facial 3D após o processo de otimização (mModelo3D) e por fim, na última coluna, a diferença entre as medidas da nuvem de pontos e do modelo morfológico facial 3D otimizado (difPM).

Tabela 5.4: Medidas para o conjunto de testes masculino. Para cada medida (primeira coluna), temos a distância entre os pontos de interesse faciais da nuvem de pontos (segunda coluna), representado por mPontos, e sua distância correspondente no modelo morfológico facial 3D otimizado (terceira coluna), representado por mModelo. Na última coluna temos a diferença entre as distâncias da segunda e terceira coluna, representado por difPM.

Medida	mPontos	mModelo	difPM
(a)	10,62 cm	10,78 cm	0,16 cm
(b)	7,02 cm	7,87 cm	0,85 cm
(c)	5,04 cm	5,05 cm	0,01 cm
(d)	10,50 cm	10,50 cm	0,0 cm

Tabela 5.5: Medidas para o conjunto de testes feminino. Para cada medida (primeira coluna), temos a distância entre os pontos de interesse faciais da nuvem de pontos (segunda coluna), representado por mPontos, e sua distância correspondente no modelo morfológico facial 3D otimizado (terceira coluna), representado por mModelo. Na última coluna temos a diferença entre as distâncias da segunda e terceira coluna, representado por difPM.

Medida	mPontos	mModelo	difPM
(a)	9,25 cm	8,60 cm	0.65 cm
(b)	6,78 cm	6,80 cm	0.02 cm
(c)	5,37 cm	5,10 cm	0,27 cm
(d)	10.19 cm	10.19 cm	0,0 cm

Os resultados numéricos apresentados mostram que em geral, nosso sistema é capaz de ajustar o modelo morfológico facial 3D à face do usuário corretamente. O melhor ajuste em ambos os casos foi a distância entre as bochechas (medida (d)). Acreditamos que por ser uma

área onde, na maioria das aquisições de face, a quantidade de ruídos provenientes da nuvem de pontos é menor, e o algoritmo *KNN* consegue aproximar os pontos do modelo morfológico facial 3D com os da nuvem de pontos com maior precisão. O restante das diferenças para ambos os sexos são pequenas. No conjunto de treino masculino o pior ajuste foi a medida entre os olhos (medida (b)), confirmando o problema de ser uma área que, em alguns casos, acaba sendo capturada pela câmera RGB-D com muitos ruídos. No conjunto de teste feminino o pior ajuste foi a do meio dos olhos até o queixo (medida (a)), sendo novamente uma medida que tem como ponto inicial a área dos olhos, ou seja, uma área ruidosa.

Comparando as medidas apresentadas do conjunto de testes masculino, Tabela 5.4, e do conjunto de testes feminino, Tabela 5.5, podemos dizer que, em geral, o conjunto masculino apresentou um melhor ajuste entre o modelo morfológico facial 3D e a nuvem de pontos, o que pode ser visualmente observado nas Figuras 5.13 e 5.14 na Seção 5.3. De uma forma geral, baseando-se em nossos resultados numéricos, podemos dizer que o nosso sistema é capaz de reconstruir corretamente a forma da face humana dado uma nuvem de pontos como entrada.

5.3 Resultados Visuais

Os resultados visuais foram gerados para demonstrar a performance do nosso sistema utilizando os parâmetros ótimos achados durante o processo de *tuning* do sistema. O nosso grupo de indivíduos de teste possui 9 indivíduos de cada sexo e nós utilizamos como objeto virtual neste trabalho diferentes tipos de óculos, conforme podemos observar na Figura 5.12. Todos os objetos virtuais estão sob a licença *Creative Commons* 3.0 e nos é permitido compartilhar e adaptar os modelos originais desde que dado o crédito aos autores originais.

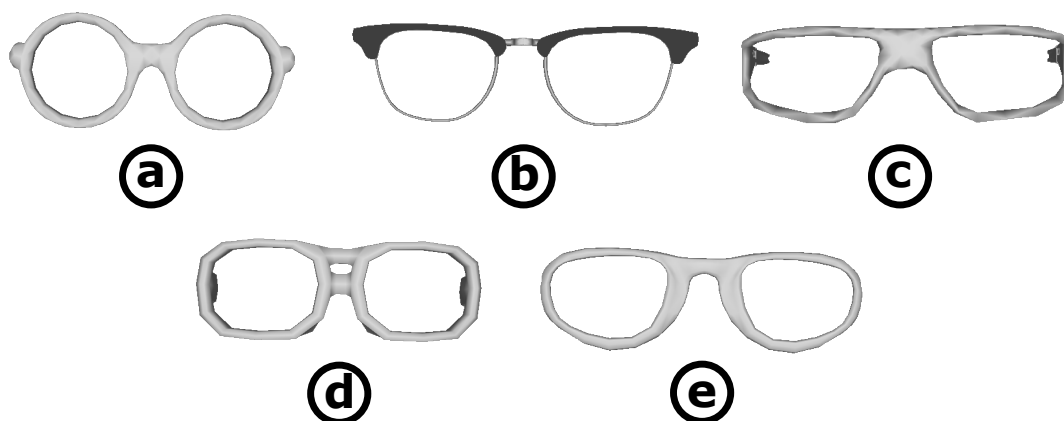


Figura 5.12: Os cinco modelos virtuais utilizados na elaboração dos resultados visuais deste trabalho. Em (a) temos um modelo de aro redondo, em (b) um modelo conhecido como tartaruga, em (c) um modelo esportivo, em (d) um modelo clássico e em (e) um modelo de aro mais grosso¹.

As Figuras 5.13 e 5.14 mostram a nuvem de pontos e seu respectivo modelo morfológico facial 3D otimizado para alguns indivíduos do sexo masculino e feminino respectivamente. Conforme podemos observar, o processo de otimização utilizando os parâmetros encontrados no processo de *tunning* aproxima o modelo morfológico da nuvem de pontos.

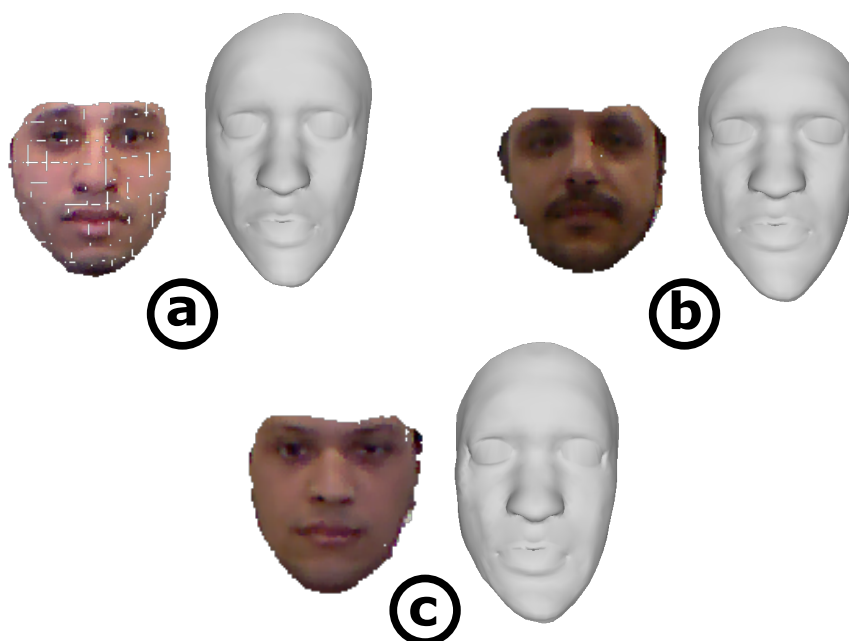


Figura 5.13: Em (a), (b) e (c) podemos observar a nuvem de pontos e seu respectivo modelo morfológico facial 3D otimizado para os indivíduos masculinos.

¹Todos os modelos disponíveis em: <<http://www.blendswap.com/>>. Acesso em agosto, 2016.

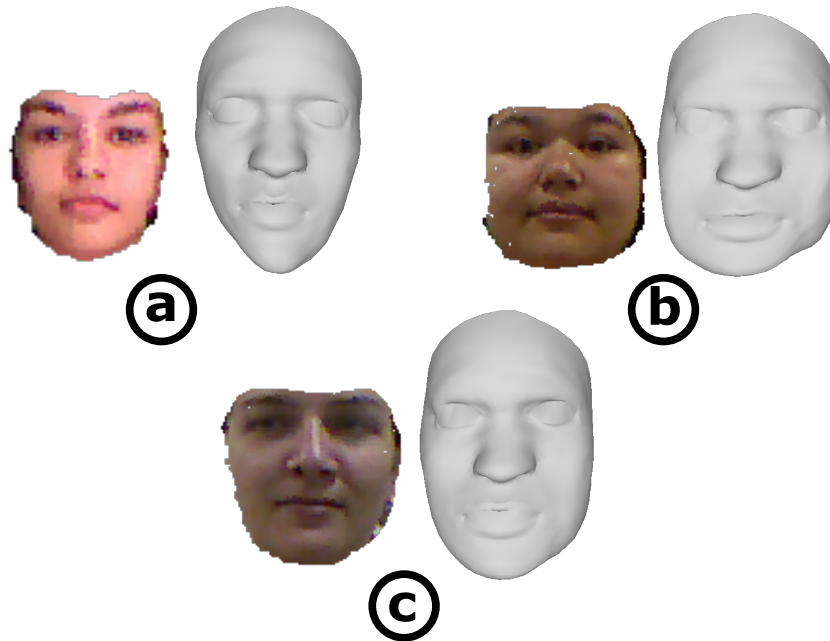


Figura 5.14: Em (a), (b) e (c) podemos observar a nuvem de pontos e seu respectivo modelo morfológico facial 3D otimizado para os indivíduos femininos.

Nas Figuras 5.15 e 5.16 podemos observar alguns resultados visuais, ou seja, a imagem RGB com o óculos renderizado sobre ela, para indivíduos masculinos e femininos. A coluna à esquerda mostra a imagem RGB com o óculos renderizado sobre ela sem sofrer o processo de deformação baseado nas mudanças ocorridas entre o modelo morfológico facial 3D original e otimizado, conforme abordado no Capítulo 4.4. Em seguida temos a imagem RGB com os óculos renderizado após o processo de deformação. Para uma melhor visualização do processo de deformação do objeto virtual na face do usuário, apresentamos na coluna da direita os mesmos resultados focando apenas na região próximo aos olhos.

Na primeira linha da Figura 5.15 o óculos começa levemente rotacionado à direita. Uma das sobrancelhas da usuária está mais visível que a outra. Após o processo de otimização, o problema da rotação do óculos foi resolvido e o óculos está melhor posicionado no rosto do usuário. Na segunda linha temos um óculos de tamanho desproporcional ao rosto do usuário. Podemos observar no lado direito da imagem original recortada que a armação passa a linha da face do usuário e após a otimização o óculos está melhor escalado dada a forma do rosto do usuário. Em sequência temos um usuário com um óculos que não segue o contorno do seu

rosto e após o processo de otimização o óculos está melhor adaptado ao formato do rosto. Na quarta linha podemos observar o óculos levemente rotacionado à direita e após o processo de otimização o óculos está melhor posicionado no rosto do usuário. Por fim, na quinta linha podemos observar que o óculos está melhor ajustado ao contorno do rosto do usuário no modelo otimizado.

Na primeira linha da Figura 5.16 podemos observar na imagem original recortada que o óculos, apesar de bem posicionado, está um pouco alto na face da usuária e após o processo de otimização o óculos está melhor posicionado. Na linha seguinte o óculos está reto na imagem com o modelo original e na imagem com o modelo otimizado, o óculos está levemente rotacionado para a direita ficando mais suave dado as formas do rosto do usuário. Na terceira linha, mostramos que nosso sistema lida também com faces que não estão retas. Podemos observar que o mesmo começa bem posicionado e após o processo de otimização podemos observar um leve ajuste no lado esquerdo do óculos. Em seguida temos um usuário onde o óculos começa bem posicionado mas o mesmo não segue a linha do seu rosto e após o processo de otimização podemos observar o óculos ajustado ao rosto do usuário. Na quinta e última linha o óculos começa bem posicionado porém, podemos observar no lado esquerda da imagem original recortada que a armação passa a linha da face do usuário e após a otimização o óculos está melhor escalado dada a forma do rosto do usuário.

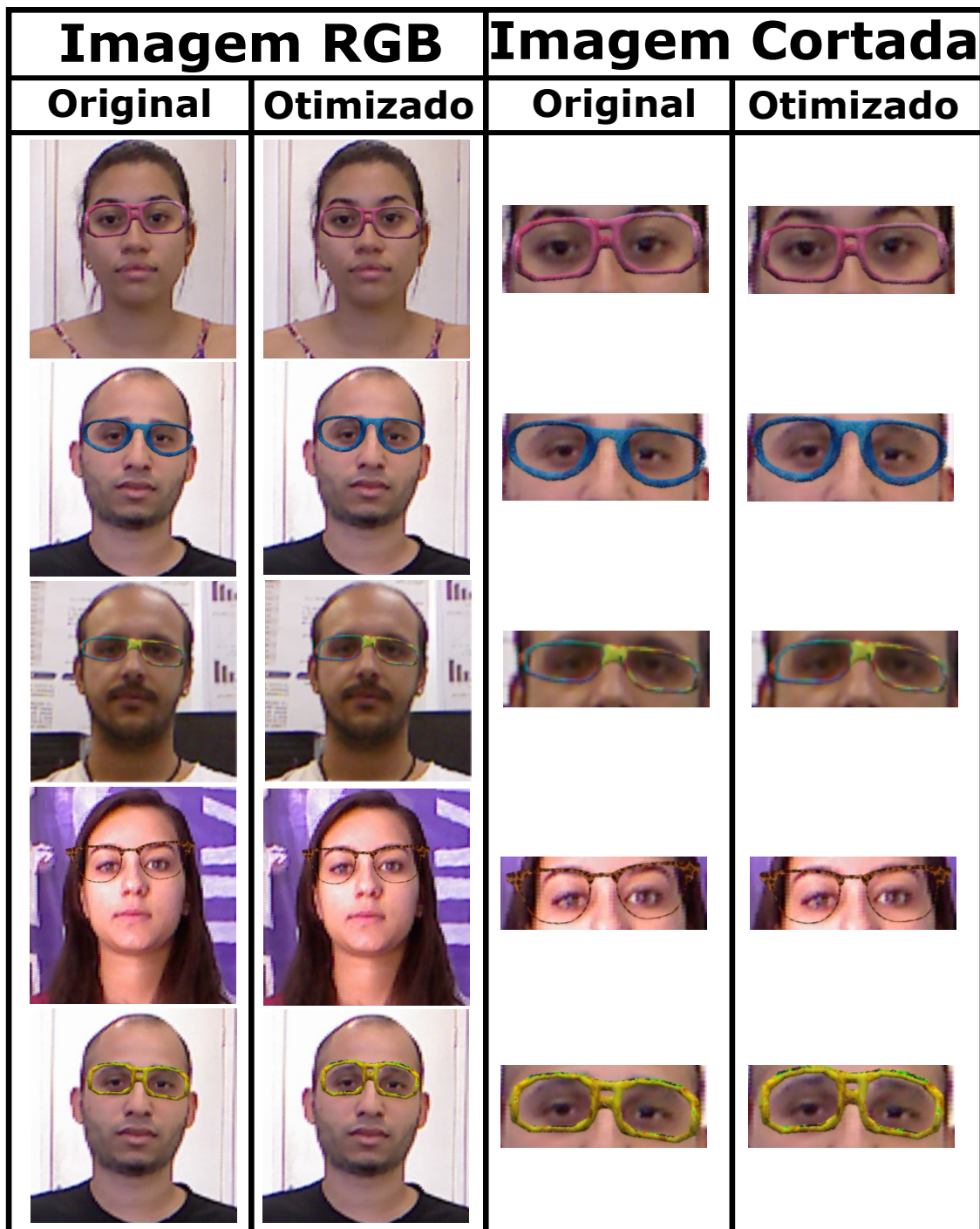


Figura 5.15: Resultados visuais do nosso sistema. Indivíduos masculinos e femininos com diferentes tipos de óculos. A coluna à esquerda mostra o objeto virtual renderizado na imagem RGB, sem passar pelo processo de otimização, lado a lado com o objeto virtual após o processo de otimização. Na coluna da direita apresentamos a imagem da coluna da esquerda cortada na região dos olhos para uma melhor visualização do resultado.

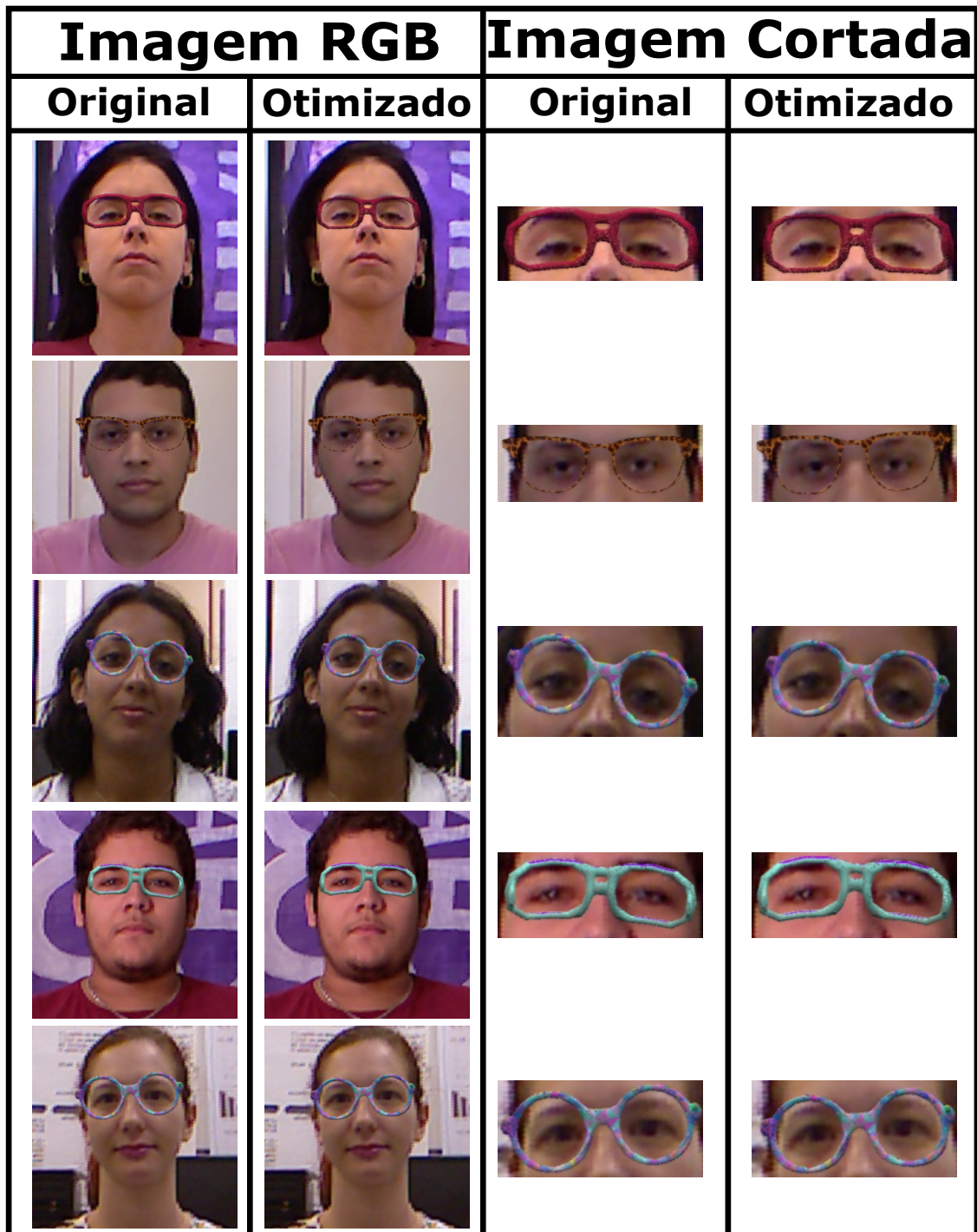


Figura 5.16: Resultados visuais do nosso sistema. Indivíduos masculinos e femininos com diferentes tipos de óculos. A coluna à esquerda mostra o objeto virtual renderizado na imagem RGB, sem passar pelo processo de otimização, lado a lado com o objeto virtual após o processo de otimização. Na coluna da direita apresentamos a imagem da coluna da esquerda cortada na região dos olhos para uma melhor visualização do resultado.

5.4 Limitações

Apesar dos resultados promissores do trabalho apresentado, nosso sistema ainda apresenta algumas limitações. Inicialmente podemos citar a renderização dos objetos virtuais sobre a imagem RGB. Para aumentarmos a sensação de realismo é precisamos melhorar a renderização e com as técnicas de processamento de imagem que utilizamos neste trabalho, supomos não ser possível realizar essas melhorias.

Em relação a captura dos dados, nosso sistema não lida com os ruídos provenientes da nuvem de pontos capturada pela câmera RGB-D e também não tratamos problemas de oclusões na face do usuário provenientes, por exemplo, de cabelo. Sendo assim, para que possamos capturar a face, principalmente nos usuários femininos, precisamos solicitar que o cabelo esteja preferencialmente preso ou afastado da face. O nosso sistema lida apenas com faces posicionadas de frente para a câmera RGB-D, ou seja, cabeças rotacionadas não terão um boa otimização nem uma boa renderização do óculos.

O nosso modelo morfológico facial 3D possui uma quantidade pequena de vértices e achamos que a aproximação dos pontos utilizando o algoritmo KNN de 1 ponto do modelo para 4 na nuvem de pontos é uma correspondência alta e que acaba influenciando no erro final da otimização. Acreditamos que um modelo morfológico com mais vértices resultaria em uma otimização ainda melhor.

O tempo de otimização do nosso sistema ainda é alto para um sistema onde o objetivo final é a Realidade Aumentada, ou seja, precisamos implementar outros métodos de otimização para verificarmos se a otimização do modelo morfológico facial 3D é melhorada.

O objeto virtual escolhido para a apresentação dos resultados visuais este trabalho foram diferentes tipos de óculos, usualmente aplicado em trabalhos como se fosse um modelo rígido. Porém, podemos citar como aplicação prática, um empreendimento virtual de uma ótica onde óculos sejam feitos sob medida para o usuário baseado na otimização de um modelo morfológico facial 3D o que nos garante medidas reais da face humana.

Capítulo 6

Conclusão

Neste trabalho foi apresentado um sistema que permite a visualização de objetos virtuais corretamente posicionados e devidamente ajustados para que acompanhe as formas do rosto do usuário. Para isso, dado um banco de faces escaneadas de ambos os sexos, criamos um modelo morfológico facial 3D para cada sexo utilizando o método de análise dos componentes principais, o que nos permite criar modelos faciais parametrizados. Capturamos um banco de faces através de uma câmera RGB-D contendo uma nuvem de pontos da região facial do usuário, uma imagem RGB e 66 pontos de interesse faciais tanto em 2D quanto em 3D. Dado o modelo morfológico facial 3D e a nuvem de pontos, nós utilizamos a otimização por descida de gradiente para otimizar os pesos para os componentes do modelo fazendo com que o mesmo se aproximasse da forma do rosto do usuário. Após o processo de otimização, nós aplicamos o método TPS no modelo morfológico facial 3D original e no otimizado afim de achar funções de mapeamento que relacionem um conjunto de pontos ao outro. Desta forma conseguimos ajustar os objetos virtuais de forma com que eles se encaixem melhor no modelo morfológico facial 3D. Por fim, nós renderizamos o objeto virtual na imagem RGB.

Nossos experimentos mostraram que a otimização por descida de gradiente se comporta melhor com valores δ pequenos conforme mostrado na parte de *tuning* do sistema. Nossos resultados numéricos mostraram que baseado nas medidas escolhidas, a otimização fez

com que o modelo morfológico facial 3D se aproximasse com um bom grau de precisão em comparação com as medidas da nuvem de pontos. Por fim, nossos resultados visuais demonstraram que a otimização por descida de gradiente melhora o posicionamento e escala do objeto virtual na face do usuário.

6.1 Trabalhos Futuros

Como trabalho futuro, propomos utilizar um sensor com melhor resolução ou ainda trabalhar com nuvens de pontos densas. Desta forma, acreditamos que a otimização ficará ainda melhor e com resultados mais precisos. Também propomos implementar outros métodos de otimização, como a otimização por enxame de partículas (PSO) e o algoritmo de Gauss-Newton, e comparar com a nossa solução atual.

Propomos também como trabalho futuro aplicar a técnica de *subsampling* no modelo morfológico facial 3D, ou seja, aumentar o número de vértices do modelo, o que originaria um modelo capaz de detalhar com mais precisão as formas do rosto.

Afim de melhorar o realismo da composição da imagem final, propomos utilizar alguma biblioteca convencional de visualização, como por exemplo, o OpenGL. Finalmente, gostaríamos de testar o nosso sistema com outros tipos de objetos virtuais, como por exemplo, bigodes, bonés e tatuagens.

Referências Bibliográficas

ALEXANDER, O.; FYFFE, G.; BUSCH, J.; YU, X.; ICHIKARI, R.; JONES, A.; DEBEVEC, P.; JIMENEZ, J.; DANVOYE, E.; ANTIONAZZI, B. et al. Digital ira: Creating a real-time photoreal digital actor. In: ACM. *ACM SIGGRAPH 2013 Posters*. [S.l.], 2013. p. 1.

ALEXANDER, O.; ROGERS, M.; LAMBETH, W.; CHIANG, M.; DEBEVEC, P. The digital emily project: photoreal facial modeling and animation. In: ACM. *Acm siggraph 2009 courses*. [S.l.], 2009. p. 12.

ALMEIDA, D. R. O. D.; GUEDES, P. A.; SILVA, M. M. O. D.; SILVA, A. L. Buarque Vieira e; LIMA, S. D. M.; PAULO, J.; TEICHRIEB, V. Interactive makeup tutorial using face tracking and augmented reality on mobile devices. In: IEEE. *Virtual and Augmented Reality (SVR), 2015 XVII Symposium on*. [S.l.], 2015. p. 220–226.

AMBERG, B.; PAYSAN, P.; VETTER, T. Weight, sex, and facial expressions: On the manipulation of attributes in generative 3d face models. In: SPRINGER. *International Symposium on Visual Computing*. [S.l.], 2009. p. 875–885.

ARELLANO, C.; DAHYOT, R. Robust bayesian fitting of 3d morphable model. In: ACM. *Proceedings of the 10th European Conference on Visual Media Production*. [S.l.], 2013. p. 9.

BAS, A.; SMITH, W. A.; BOLKART, T.; WUHRER, S. Fitting a 3d morphable model to edges: A comparison between hard and soft correspondences. *arXiv preprint arXiv:1602.01125*, 2016.

BEELER, T.; HAHN, F.; BRADLEY, D.; BICKEL, B.; BEARDSLEY, P.; GOTSMAN, C.; SUMNER, R. W.; GROSS, M. High-quality passive facial performance capture using anchor frames. In: ACM. *ACM Transactions on Graphics (TOG)*. [S.l.], 2011. v. 30, n. 4, p. 75.

BESL, P. J.; MCKAY, N. D. A method for registration of 3-d shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, v. 14, n. 2, p. 239–256, 1992. Disponível em: <<http://dblp.uni-trier.de/db/journals/pami/pami14.html#BeslM92>>.

BLACKWELL, S.; ROBINETTE, K. M.; BOEHMER, M.; FLEMING, S.; KELLY, S.; BRILL, T.; HOEFERLIN, D.; BURNSIDES, D. *Civilian American and European Surface Anthropometry Resource (CAESAR). Volume 2: Descriptions*. [S.l.], 2002.

BLANZ, V. Automatische rekonstruktion der dreidimensionalen form von gesichtern aus einem einzelbild. 2000.

BLANZ, V.; SCHERBAUM, K.; SEIDEL, H.-P. Fitting a morphable model to 3d scans of faces. In: IEEE. *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. [S.l.], 2007. p. 1–8.

BLANZ, V.; VETTER, T. A morphable model for the synthesis of 3d faces. In: ACM PRESS/ADDISON-WESLEY PUBLISHING CO. *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. [S.l.], 1999. p. 187–194.

BOLKART, T.; WUHRER, S. 3d faces in motion: Fully automatic registration and statistical analysis. *Computer Vision and Image Understanding*, Elsevier, v. 131, p. 100–115, 2015.

BOOKSTEIN, F. L. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, IEEE, n. 6, p. 567–585, 1989.

BRAGA, R. F.; SILVA, U. L. da; IWANO, T. M.; SCHERER, D. Estudo comparativo de tool-kits de realidade virtual e aumentada visando aplicação educacional. In: *Anais do Workshop de Desafios da Computação Aplicada à Educação*. [S.l.: s.n.], 2012. p. 138–147.

COOTES, T. F.; TAYLOR, C. J. Active shape models-’smart snakes’. In: *BMVC92*. [S.l.]: Springer, 1992. p. 266–275.

CORREIA, M. M. *Reconhecimento de Elementos Gestuais*. Dissertação (Mestrado) — Universidade do Porto, 2013.

CRUZ, L.; LUCIO, D.; VELHO, L. Kinect and rgbd images: Challenges and applications. In: *Graphics, Patterns and Images Tutoriais (SIBGRAPI-T), 2012 25th SIBGRAPI Conference on*. [S.l.: s.n.], 2012. p. 36–49.

DITTO. *Designer Glasses & Sunglasses*. 2016. Disponível em: <<https://www.ditto.com/>>.

FILHO, J. L. R. de S. *Alinhamento de nuvens de pontos adquiridos através de digitalizador câmera-projetor com luz estruturada*. 2010. Monografia (Bacharel em Ciência da Computação), UFJF (Universidade Federal de Juiz de Fora), Juiz de Fora, Brasil.

GHOSH, A.; FYFFE, G.; TUNWATTANAPONG, B.; BUSCH, J.; YU, X.; DEBEVEC, P. Multiview face capture using polarized spherical gradient illumination. *ACM Transactions on Graphics (TOG)*, ACM, v. 30, n. 6, p. 129, 2011.

GOWER, J. C. Generalized procrustes analysis. *Psychometrika*, v. 40, n. 1, p. 33–51, 1975. ISSN 1860-0980. Disponível em: <<http://dx.doi.org/10.1007/BF02291478>>.

HAUSWIESNER, S.; STRAKA, M.; REITMAYR, G. Virtual try-on through image-based rendering. *IEEE transactions on visualization and computer graphics*, IEEE, v. 19, n. 9, p. 1552–1565, 2013.

HUANG, S.-H.; YANG, Y.-I.; CHU, C.-H. Human-centric design personalization of 3d glasses frame in markerless augmented reality. *Advanced Engineering Informatics*, Elsevier, v. 26, n. 1, p. 35–45, 2012.

- HUBER, P.; HU, G.; TENA, R.; MORTAZAVIAN, P.; KOPPEN, W. P.; CHRISTMAS, W.; RÄTSCH, M.; KITTLER, J. A multiresolution 3d morphable face model and fitting framework. In: *Proceedings of the 11th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*. [S.l.: s.n.], 2016.
- JOLLIFFE, I. *Principal component analysis*. Springer-Verlag, 1986.
- KIRNER, C. *Usando Realidade Aumentada em Publicidade*. 2016. Disponível em: <<http://www.ckirner.com/download/artigos/RA-Publicidade.html>>.
- KITTLER, J.; HUBER, P.; FENG, Z.-H.; HU, G.; CHRISTMAS, W. 3d morphable face models and their applications. In: SPRINGER. *International Conference on Articulated Motion and Deformable Objects*. [S.l.], 2016. p. 185–206.
- LANITIS, A.; TAYLOR, C. J.; COOTES, T. F. Automatic interpretation and coding of face images using flexible models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, IEEE, v. 19, n. 7, p. 743–756, 1997.
- LEE, J.; MOGHADDAM, B.; PFISTER, H.; MACHIRAJU, R. *Silhouette-based 3d face shape recovery*. AK Peters, 2003.
- LIMA, J. P.; ROBERTO, R.; TEICHRIB, V.; MARQUES, G. Study about natural feature tracking for augmented reality applications on mobile devices. In: IEEE. *Virtual and Augmented Reality (SVR), 2015 XVII Symposium on*. [S.l.], 2015. p. 7–14.
- MARKETING, G. *10 Examples of Augmented Reality in Retail*. 2014. Disponível em: <<http://www.creativeguerrillamarketing.com/augmented-reality/10-examples-augmented-reality-retail/>>.
- MUJA, M.; LOWE, D. G. Fast approximate nearest neighbors with automatic algorithm configuration. In: *International Conference on Computer Vision Theory and Application VISSAPP'09*. [S.l.]: INSTICC Press, 2009. p. 331–340.
- NETWORK, M. D. *Kinect for Windows Sensor Components and Specifications*. 2010. <<https://msdn.microsoft.com/en-us/library/jj131033.aspx>>.
- OLIVEIRA-SANTOS, T.; BAUMBERGER, C.; CONSTANTINESCU, M.; OLARIU, R.; NOLTE, L.-P.; ALARAIBI, S.; REYES, M. 3d face reconstruction from 2d pictures: First results of a web-based computer aided system for aesthetic procedures. *Annals of Biomedical Engineering*, v. 41, n. 5, p. 952–966, 2013. ISSN 1573-9686. Disponível em: <<http://dx.doi.org/10.1007/s10439-013-0744-3>>.
- PEARSON, K. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, Taylor & Francis, v. 2, n. 11, p. 559–572, 1901.
- PENG, H. Application research on face detection technology based on opencv in mobile augmented reality. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, v. 8, n. 4, p. 249–256, 2015.

PIOTRASCHKE, M.; BLANZ, V. Automated 3d face reconstruction from multiple images using quality measures. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2016. p. 3418–3427.

RAY-BAN. *Ray-Ban Virtual Mirror*. 2016. Disponível em: <<http://www.ray-ban.com/usa/virtual-mirror>>.

ROBINETTE, K. M.; BLACKWELL, S.; DAANEN, H.; BOEHMER, M.; FLEMING, S. *Civilian American and European Surface Anthropometry Resource (CAESAR), Final Report. Volume 1. Summary*. [S.l.], 2002.

RUSU, R. B.; COUSINS, S. 3D is here: Point Cloud Library (PCL). In: *IEEE International Conference on Robotics and Automation (ICRA)*. Shanghai, China: [s.n.], 2011.

SARAGIH, J. M.; LUCEY, S.; COHN, J. F. Deformable model fitting by regularized landmark mean-shift. *International Journal of Computer Vision*, Springer, v. 91, n. 2, p. 200–215, 2011.

SARAGIH, K. M. J. *10 Examples of Augmented Reality in Retail*. 2013. Disponível em: <<http://facetracker.net/>>.

SNAPCHAT. *Snapchat*. 2016. Disponível em: <<https://www.snapchat.com/>>.

TAAZ. *The Brains Behind the Beauty*. 2016. Disponível em: <<http://www.taaz.com/>>.

VALLINO, J. R. *Interactive augmented reality*. Tese (Doutorado) — University of Rochester, 1998.

VETTER, T.; POGGIO, T. Linear object classes and image synthesis from a single example image. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, IEEE, v. 19, n. 7, p. 733–742, 1997.

VLASIC, D.; BRAND, M.; PFISTER, H.; POPOVIĆ, J. Face transfer with multilinear models. In: ACM. *ACM Transactions on Graphics (TOG)*. [S.l.], 2005. v. 24, n. 3, p. 426–433.

WANG, P.; TONG, X.; DU, Y.; LI, J.; HU, W.; ZHANG, Y. Augmented makeover based on 3d morphable model. In: ACM. *Proceedings of the 19th ACM international conference on Multimedia*. [S.l.], 2011. p. 1569–1572.

YAMADA, F. A. A.; CEJNOG, L. W. X.; DEMBOGURSKI, R. A.; VIEIRA, M. B.; SOUZA, R. Reconstrução de objetos 3d utilizando estruturas de indexação espacial com o microsoft kinect. *Universidade Federal de Juiz de Fora*, 2013.

YANG, Y.-I.; YANG, C.-K.; LIAO, X.-L.; CHU, C.-H. Virtual try-on of footwear in augmented reality using rgb-d cameras. In: IEEE. *Industrial Engineering and Engineering Management (IEEM), 2015 IEEE International Conference on*. [S.l.], 2015. p. 1072–1076.

YUAN, M.; KHAN, I. R.; FARBIZ, F.; NISWAR, A.; HUANG, Z. A mixed reality system for virtual glasses try-on. In: ACM. *Proceedings of the 10th International Conference on Virtual Reality Continuum and Its Applications in Industry*. [S.l.], 2011. p. 363–366.

ZHANG, Y.; WEI, W. A realistic dynamic facial expression transfer method. *Neurocomputing*, Elsevier, v. 89, p. 21–29, 2012.