

UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

Modelagem Tridimensional de Rostos Por Características Robustas

Pedro Regattieri Rocha

Dissertação de Mestrado do Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-C²MC)

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Pedro Regattieri Rocha

Modelagem Tridimensional de Rostos Por Características Robustas

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Mestre em Ciências – Ciências de Computação e Matemática Computacional. *EXEMPLAR DE DEFESA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientador: Prof. Dr. João do Espírito Santo Batista Neto

USP – São Carlos
Junho de 2023

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

R672m Rocha, Pedro Regattieri
Modelagem tridimensional de rostos por
características robustas / Pedro Regattieri Rocha;
orientador João do Espírito Santo Batista Neto. --
São Carlos, 2023.
52 p.

Dissertação (Mestrado - Programa de Pós-Graduação
em Ciências de Computação e Matemática
Computacional) -- Instituto de Ciências Matemáticas
e de Computação, Universidade de São Paulo, 2023.

1. Computação Gráfica. 2. Face. 3. Malha. I. do
Espírito Santo Batista Neto, João, orient. II.
Título.

Pedro Regattieri Rocha

**Three-dimensional Modeling of Faces Utilizing Robust
Characteristics**

Dissertation submitted to the Instituto de Ciências Matemáticas e de Computação – ICMC-USP – in accordance with the requirements of the Computer and Mathematical Sciences Graduate Program, for the degree of Master in Science. *EXAMINATION BOARD PRESENTATION COPY*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Prof. Dr. João do Espírito Santo Batista Neto

**USP – São Carlos
June 2023**

RESUMO

ROCHA, P. R. **Modelagem Tridimensional de Rostos Por Características Robustas**. 2023. 52 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2023.

A modelagem tridimensional de faces humanas é parte integral de uma gama de aplicações computacionais, de animações e entretenimento até programas de segurança e reconhecimento facial, sendo que programas desta natureza lidam com uma representação virtual de faces humanas de alguma forma durante a sua execução. E assim como existe uma variedade de aplicações que utilizam essas representações, também existem diferentes maneiras de gerar estas faces utilizando diferentes técnicas de computação gráfica, como por exemplo modelagem tridimensional de malhas ou métodos baseados em pontos. Alguns destes métodos requerem uma única imagem do rosto a ser modelado, enquanto outros métodos requerem fotos de diferentes ângulos para criar um modelo mais detalhado. Certos métodos produzem o modelo da face apenas recebendo um conjunto de imagens como entrada do usuário, enquanto outros requerem que o usuário realize passos adicionais durante a execução do método. O objetivo deste projeto de pesquisa é o desenvolvimento de um método que, dado uma foto como entrada, extraí um conjunto de pontos importantes, referentes a pontos anatômicos extraídos de um rosto humano específico, modificando então uma malha tridimensional triangular de forma que esta malha poligonal se assemelhe ao rosto cujas características foram extraídas sem que sejam necessárias mais informações ou comandos do usuário. Este método, implementado em uma instância de teste em Python, utiliza princípios de modelagem de malhas como o uso de δ -coordenadas para permitir a modificação de uma malha inicial de forma controlada a partir das informações recebidas como entrada. As malhas geradas por este método mantém o formato e as cores do rosto base, com testes realizados constatando a robustez das malhas geradas e manipuladas assim como a consistência na velocidade de propagação de modificações pela malha, independente do número de vizinhos do ponto ou do tamanho da modificação realizada.

Palavras-chave: Modelagem Tridimensional, Faces Humanas, Curvas, Pontos, Computação Gráfica.

ABSTRACT

ROCHA, P. R. **Three-dimensional Modeling of Faces Utilizing Robust Characteristics**. 2023. 52 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2023.

The modeling of human faces is an integral part of an array of computer applications, from animations and entertainment to security and facial recognition programs, in that programs of this nature use a virtual representation of human faces at some point during their execution. And in the same way there are a variety of applications that use these representations, there also are different computer graphics techniques to generate such representations, for example three-dimensional mesh modeling methods or point-based methods. Some of these methods require a single image of the face that will be modeled, whilst others may require several photographs from different angles to create a more detailed model. Certain methods produce a model of a face using only the set of photographs, while others may require further user input during their execution. The objective of this research project is the development of a method that, given a set of curves or important points that refer to important anatomic features extracted from a specific human face as input, modifies a three-dimensional quadrilateral mesh so that the resulting polygonal mesh resembles the specific human face whose features were extracted. This method, implemented in a test instance in Python, utilizes Three-dimensional modeling principles such as the use of δ -coordinates to manipulate an initial mesh according to information received as input. The meshes generated by this method maintain the features and color of the original face, with tests confirming the robustness of the created and edited meshes as well as the consistency of the speed at which changes are propagated through the mesh, independently of the number of neighbours of the edited point or how far it was moved.

Keywords: Three-Dimensional modeling, Human faces, Curves, Points, Computer Graphics.

LISTA DE ILUSTRAÇÕES

Figura 1 – Malha "Suzanne", representando a cabeça de um orangotango, criada por Willem-Paul van Overbruggen em 2002	22
Figura 2 – Uma malha quadrilateral em forma de telhado	23
Figura 3 – Uma malha cujas faces são quatro quadrados que será utilizada como exemplo.	25
Figura 4 – Exemplo do processo de edição utilizando o operador de Laplace-Beltrami em um conjunto point-based.	29
Figura 5 – Exemplo do funcionamento da câmara de perspectiva virtual. Fonte: MediaPipe Face Mesh.	30
Figura 6 – Exemplo de Detecção de Landmarks com o FaceMesh. Fonte: Google for Developers.	31
Figura 7 – Diagrama do método proposto	33
Figura 8 – Busto de Max Planck com as curvas ridge desenhadas sobre a sua superfície, as curvas ridge extraídas do busto, e uma superfície reconstruída a partir destas curvas. Ohtake, Belyaev e Seidel (2004)	34
Figura 9 – Exemplo de foto que é utilizada como entrada para o método. Vista frontal do rosto.	35
Figura 10 – Exemplo de execução do Face Mesh para a imagem 9	35
Figura 11 – Exemplo de malha inicial, gerada a partir dos pontos obtidos na imagem 10.	36
Figura 12 – Exemplo de execução do Face Mesh para a imagem 9, com os landmarks 78, 228 e 358 marcados em amarelo, verde e azul respectivamente.	41
Figura 13 – Malha após as modificações intermediárias.	42
Figura 14 – Malha após as modificações finais.	42
Figura 15 – Foto utilizada como entrada no segundo teste.	43
Figura 16 – Exemplo de execução do Face Mesh para a imagem 15, com os landmarks 308, 50 e 13 marcados em amarelo, verde e azul respectivamente.	44
Figura 17 – Malha inicial do segundo teste, gerada a partir da foto 16.	45
Figura 18 – Malha após as modificações intermediárias.	45
Figura 19 – Malha após as modificações finais.	46
Figura 20 – O modelo canônico da malha utilizado pelo Face Mesh. Este modelo ilustra as conexões entre os landmarks. Fonte: Developers.google	48
Figura 21 – Um exemplo de uma malha com o formato de uma cabeça humana genérica. Fonte: Mad Mouse Design, 2007	49

LISTA DE TABELAS

Tabela 1 – Tabela com os resultados do primeiro experimento.	43
Tabela 2 – Tabela com os resultados do segundo experimento.	43

SUMÁRIO

1	INTRODUÇÃO	15
1.1	Objetivo	16
1.2	Hipótese	16
1.3	Organização da Monografia	16
2	TRABALHOS RELACIONADOS	19
3	FUNDAMENTOS TEÓRICOS	21
3.1	Malhas Poligonais	21
3.2	Operador de Laplace-Beltrami	23
3.2.1	<i>Edição de Malhas</i>	27
3.3	MediaPipe Face Mesh	29
4	METODOLOGIA	33
4.1	Extração de Pontos Importantes	33
4.2	Criação da Malha Inicial	36
4.3	Modelagem / Manipulação da Malha	37
4.4	Materiais	37
4.5	Considerações Éticas	38
5	RESULTADOS	39
5.1	Configuração dos testes	39
5.1.1	<i>Primeiro Teste</i>	40
5.1.2	<i>Segundo Teste</i>	41
6	CONCLUSÕES	47
6.1	Trabalhos Futuros	48
	REFERÊNCIAS	51

INTRODUÇÃO

A modelagem de rostos humanos é utilizada como parte de uma gama de aplicações, como por exemplo, aplicativos de reconhecimento facial, programas voltados à modelagem artística ou mesmo na criação de avatares em aplicações de entretenimento (KITTLER *et al.*, 2016).

Malhas poligonais são uma ferramenta versátil para aplicações em computação gráfica, que podem ser modeladas por um grafo bidirecional não orientado conexo que descreve uma forma geométrica através da posição de seus vértices e das conexões como apresentado por Botsch *et al.* (2007).

Através de métodos de modelagem de malhas é possível modificar malhas poligonais para que sejam obtidas uma variedade de formas, desde objetos até seres vivos como plantas e animais.

Uma das inúmeras formas que podem ser aproximadas por meio destas técnicas é o rosto humano, que, por sua vez, em decorrência de pequenas diferenças anatômicas entre cada pessoa, possui uma infinidade de configurações possíveis Richmond *et al.* (2018). Através de técnicas como malhas dinâmicas e o uso do operador de Coordenadas Laplacianas, é possível alterar a configuração dos vértices de uma malha sem modificar o número de arestas Soares (2007).

Desta maneira, modificações em um vértice são propagadas aos seus vizinhos, de forma a continuamente modificar a malha para se assemelhar ao rosto humano. Malhas poligonais podem tomar a forma de qualquer objeto euclidiano bidimensional ou tridimensional, sendo que os rostos humanos não são exceção. Ao modificar uma malha tridimensional, é possível fazer com que ela pareça com uma máscara.

Utilizando informações das curvas características que descrevem a anatomia do rosto de uma determinada pessoa é então possível modificar uma malha para que ela se torne análoga a uma máscara da face desta determinada pessoa. Por meio das curvas que caracterizam um

determinado rosto, é possível então modificar a configuração dos vértices de uma malha poligonal para simular estas curvas, e, desta forma, modelar este dado rosto (KENT; MARDIA; WEST, 1996).

1.1 Objetivo

O projeto de pesquisa tem por objetivo propor e testar um método que, dado um conjunto de pontos que descrevem as características de um rosto humano, gera uma malha poligonal tridimensional para modelar este rosto e então permitir a edição desta, de forma a propagar quaisquer modificações realizadas em um ponto deste conjunto entre seus vizinhos.

Sendo possível a modelagem de rostos humanos únicos a partir de seus pontos característicos, o objetivo do projeto, para uma descrição mais específica, é desenvolver um método que, dado uma imagem frontal de um rosto humano fornecida como entrada, extrai os pontos que descrevem o rosto exibido na imagem, crie uma malha tridimensional inicial utilizando estes pontos para se assemelhar ao rosto fornecido. Adicionalmente, após a modelagem automática da malha a partir dos pontos fornecidos, será permitido ao usuário modificar os pontos da malha, e estas mudanças são propagadas a pontos vizinhos. Após cada iteração de modificação de um dado ponto, a malha será gerada novamente. Desta forma, os objetivos deste projeto de pesquisa são, a rigor:

- Implementar um método para modelagem de faces humanas a partir de uma imagem fornecida pelo usuário.
- Testar o funcionamento do método e aperfeiçoar a qualidade das malhas produzidas.
- Permitir que o usuário observe o resultado das modificações realizadas.

1.2 Hipótese

O conjunto de pontos e curvas características extraídos de um rosto humano pode prover informações visuais suficientemente significantes para gerar uma malha capaz de ser identificada como o rosto de um indivíduo específico, sem que seja necessário que haja interação de um usuário do método após o fornecimento da entrada.

A edição da malha, realizada pelo usuário, pode ser feita pontualmente, com modificações sendo propagadas na malha sem afetar as arestas existentes.

1.3 Organização da Monografia

A dissertação está organizada em 6 capítulos, a contar com esta introdução. O Capítulo 2 descreve alguns dos principais trabalhos relacionados a área de modelagem facial. O capítulo

3 lida com os fundamentos teóricos e técnicas que serão utilizados durante o desenvolvimento do método, cuja metodologia será apresentada no capítulo 4. O capítulo 5 exhibe os resultados encontrados durante os testes realizados, e uma breve discussão dos mesmos enquanto no capítulo 6 são realizadas as conclusões, analisando possíveis aplicações.

TRABALHOS RELACIONADOS

O uso e aplicações de coordenadas diferenciais na edição de malhas, em específico o operador de Laplace-Beltrami, está documentado no artigo “Laplacian Mesh Processing” (SORKINE, 2006). O artigo detalha e explica os passos para a criação e transformação da matriz laplaciana e a operação para obter as coordenadas cartesianas a partir dela. O artigo também detalha uma gama de aplicações deste método, de especial relevância para o projeto a parte de edição de malhas e interpolação de formas, em que são descritas técnicas para preservar a orientação dos detalhes quando alterações são realizadas sobre uma parte específica da malha. Quando uma parte do rosto for modelada de acordo com a curva característica, os impactos sobre as outras partes do rosto, com suas próprias curvas características, devem ser minimizados.

Em “Differential Coordinates for Interactive Mesh Editing” (LIPMAN *et al.*, 2004), um dos artigos relacionados ao artigo “Laplacian Mesh Processing”, os autores descrevem o processo de edição de malhas focando em regiões de interesse. A escolha da região de interesse adequada é crucial para evitar que as transformações executadas levem a resultados diferentes dos esperados.

A tese “Movimento de Malhas e Remalhamento de Malhas Superficiais” (SOARES, 2007) explora os usos de malhas dinâmicas, e estressa que esta técnica minimiza a necessidade de remalhamento exceto no caso de grandes deformações, porém precauções devem ser tomadas para evitar que as malhas apresentem comportamentos inválidos, como a geração de polígonos com áreas negativas. O uso de malhas permite que modificações sejam propagadas pelos vértices vizinhos ao vértice modificado, idealmente permitindo a modelagem de curvas suaves e simplificando a execução do algoritmo para modificação da malha.

No artigo “3D Morphable Face Models and Their Applications”, Kittler *et al.* (2016) exploram o uso de modelos faciais morfáveis 3D, e mencionam as aplicações deste tipo de modelo para neutralizar expressões faciais e auxiliar no processo de reconhecimento facial de imagens 2D. O artigo explora tanto o uso de malhas para replicar o formato da face como o uso

de texturas para replicar as características RGB de cada ponto na malha, embora, no contexto deste projeto, a primeira parte é de especial relevância pois lida com o formato da malha.

Em “3D Face Mesh Modeling for 3D Face Recognition”, [A-Nasser, Mohammad e Mohame \(2009\)](#) propõem um método para reconhecimento facial que utiliza três pontos de características faciais para alinhar um modelo facial genérico a uma imagem que deseja ser replicada. O modelo, que utiliza malhas triangulares, então é deformado de acordo com os dados para aproximar melhor o modelo utilizado. Após a etapa de modelagem, ocorrem os testes de reconhecimento, garantindo a similaridade dos modelos obtidos na etapa anterior.

Outro trabalho que lida com a criação de rostos 3D através de uma única imagem é “3D Face Reconstruction From Single 2D Image Using Distinctive Feature” ([AFZAL et al., 2020](#)). O método utiliza extração de características e cálculo de profundidade para extrair características faciais e gerar uma imagem tridimensional ao adicionar profundidade ao modelo do rosto. Embora o método não lide com malhas, o resultado final é um conjunto de imagens 3D do rosto original.

Em “Detailed 3D Human Body Reconstruction from a Single Image Based on Mesh Deformation” [Zhang e Xiao \(2021\)](#) também lidam com o método de deformação de malha utilizado por [A-Nasser, Mohammad e Mohame \(2009\)](#), porém o objetivo é reconstruir o corpo humano inteiro, e não apenas o rosto. O resultado é uma malha cuja posição e formato é fiel ao corpo conforme ele está posicionado na imagem, podendo replicar até detalhes da roupa sendo utilizada na foto.

No artigo “Automatic system for virtual human reconstruction with 3D mesh multi-texturing and facial enhancement” [Pagés, Berjón e Morán \(2013\)](#) propõem um método para criar atlas de textura para modelos humanos em jogos virtuais a partir de fotos, assim como duas possíveis melhorias para especificamente aprimorar a parte facial das malhas tridimensionais dos modelos gerados, uma melhoria passiva e outra baseada em técnicas ativas como iluminação.

FUNDAMENTOS TEÓRICOS

Neste capítulo serão introduzidos os fundamentos teóricos, detalhando brevemente cada tópico da seguinte maneira: inicialmente será feita uma breve introdução sobre Malhas Poligonais, explicando então o uso do operador de Laplace-Beltrami e contextualizando o uso do mesmo em processos de edição de malhas. Finalmente, encerraremos o capítulo com a exposição de uma ferramenta cujo uso é importante no contexto deste trabalho para a extração de pontos importantes a partir de uma foto ou imagem de um rosto.

3.1 Malhas Poligonais

Seja $\mathbf{M} = (\mathbf{V}, \mathbf{A}, \mathbf{F})$ uma malha poligonal descrita por \mathbf{V} , um conjunto de vértices; \mathbf{A} , um conjunto de arestas entre os vértices de \mathbf{V} e \mathbf{F} , o conjunto das faces de \mathbf{M} descritas por \mathbf{V} e \mathbf{A} . \mathbf{M} pode ser categorizada pelos polígonos descritos pelas conexões de \mathbf{V} e \mathbf{A} , ou seja, as faces do conjunto \mathbf{F} . Por exemplo, se o polígono formado for um triângulo, a malha é triangular. Porém, quadriláteros ou outros polígonos convexos simples também podem ser utilizados. O padrão predominante para modelagem de formas geométricas em aplicações que utilizam malhas bidimensionais ou tridimensionais é o uso de malhas triangulares e quadrilaterais, devido a sua relativa simplicidade comparado a malhas com mais vértices por face. O modelo a ser utilizado neste projeto consiste no uso de malhas triangulares, utilizado também pela ferramenta que será detalhada ao final deste capítulo.

Para descrever uma malha poligonal, dois elementos são necessários: a) uma lista de vértices contendo as coordenadas de cada ponto \mathbf{v}_i no eixo \mathbf{X} e eixo \mathbf{Y} , caso a malha seja bidimensional, adicionando a coordenada do eixo \mathbf{Z} para uma malha tridimensional. Uma maneira de armazenar estas coordenadas é manter os valores em duas ou três matrizes, cujas células contêm a transformação de um dado vértice em um dos planos em relação à origem do sistema; b) uma lista de adjacências, indicando quais os vértices vizinhos \mathbf{v}_j de cada vértice \mathbf{v}_i .

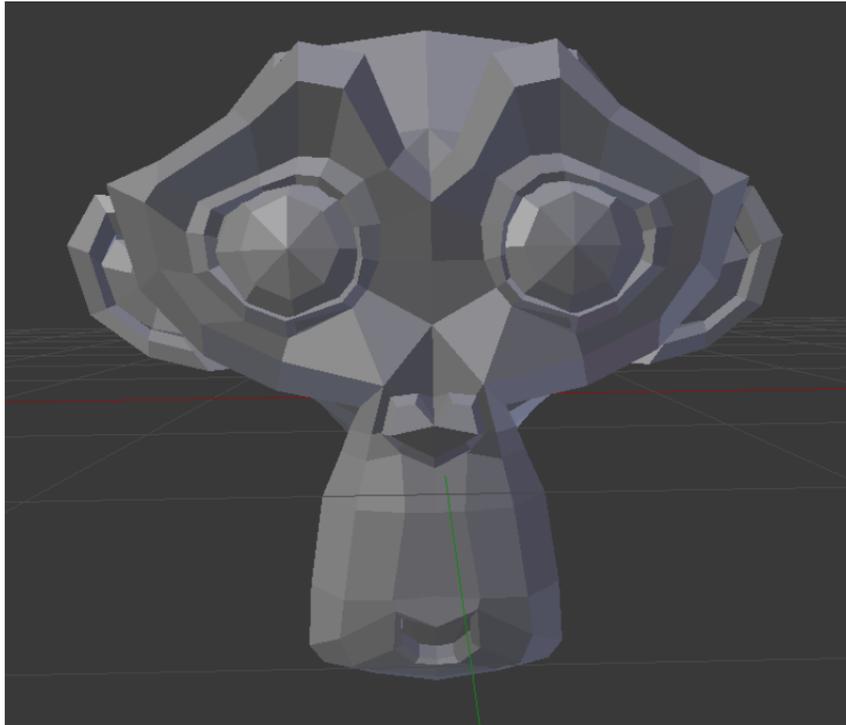


Figura 1 – Malha "Suzanne", representando a cabeça de um orangotango, criada por Willem-Paul van Overbruggen em 2002

Esta lista de adjacências também pode ser descrita por uma matriz, no caso uma única matriz quadrada em que cada linha i representa um vértice i da malha e cada coluna j possui valores que indicam se um vértice v_j é vizinho de v_i ou não, por exemplo, 1 se o nó for vizinho, 0 caso ele não possua uma aresta. É importante notar que, conforme uma malha poligonal se torna cada vez mais detalhada (com mais vértices, arestas e faces) o número de arestas conectando um único vértice a outros tende a continuar o mesmo. O resultado disto é que a matriz de adjacências se torna esparsa, ou seja, a maior parte dos elementos desta matriz tem valor 0.

A malha ilustrada na figura 2 foi gerada em Octave utilizando três matrizes quadradas com 441 elementos cada, sendo que cada elemento corresponde à posição nos eixos tz , ty e tx de cada vértice. O resultado é uma malha com 400 faces. A matriz de adjacência desta malha é esparsa, uma vez que cada vértice está conectado no máximo a 4 outros, e no mínimo a 2 outros vértices vizinhos, de forma que, para cada linha da matriz de adjacência de tamanho 441 x 441, existem ao menos 437 valores iguais a zero.

Devido à presença consistente de malhas em aplicações de computação gráfica, diversas linguagens de programação possuem soluções e bibliotecas dedicadas à geração de malhas tridimensionais a partir de conjuntos de pontos e faces, nuvens de pontos ou interpretando arquivos de formatos dedicados a armazenar dados de malhas (e.g. arquivos .obj ou .off), auxiliando os processos de geração e visualização de malhas. Algumas linguagens, como Python, possuem uma variedade de soluções, como PyVista (SULLIVAN; AL, 2019), Open3D (TEAM, 2019) e Plotly (PLOTLY, 2013), sendo que Plotly também possui implementações em outras

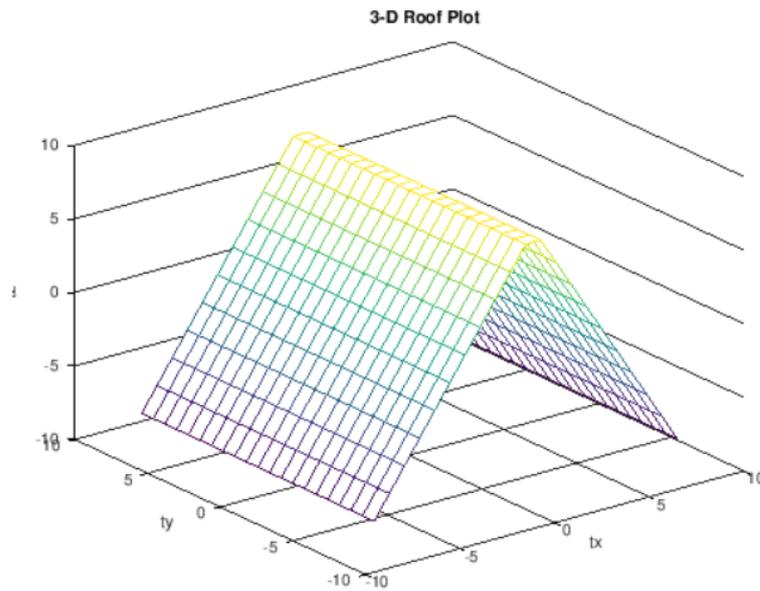


Figura 2 – Uma malha quadrilateral em forma de telhado

linguagens como R, Javascript e MATLAB®.

3.2 Operador de Laplace-Beltrami

O operador de Laplace-Beltrami é um operador diferencial de segunda ordem definido como o divergente do gradiente de uma função duplamente derivável f . Este é utilizado em uma gama de contextos, como campos elétricos, campos gravitacionais, fluxo de fluidos e propagação de ondas, mas possui especial relevância para o projeto no contexto de geometria diferencial pois, ao trabalhar no espaço euclidiano, o Laplaciano é dado pela soma das segundas derivadas parciais da função com base em cada uma das variáveis independentes, permitindo a operação sobre funções definidas em subvariedades no espaço euclidiano (SORKINE, 2006).

A Coordenada Diferencial, também chamada de δ -coordenada de um dado vértice \mathbf{v}_i de \mathbf{V} , é definida como a diferença entre a coordenada cartesiana de \mathbf{v}_i , cujas duas ou três componentes estão armazenadas nas matrizes descritas na seção anterior, e o centro de massa dos seus vizinhos imediatos, \mathbf{v}_j .

A fórmula das δ -coordenadas de um dado vértice \mathbf{v}_i , dado em termos do número N_j de vizinhos \mathbf{j} do vértice \mathbf{i} e das coordenadas cartesianas de cada um de seus vizinhos \mathbf{v}_j é dada por

$$\delta_i = (\delta_i(x), \delta_i(y), \delta_i(z)) = \mathbf{v}_i - \frac{1}{N_j} \sum_{j \in N(i)} \mathbf{v}_j. \quad (3.1)$$

Como a malha \mathbf{M} é a aproximação linear por partes de uma superfície suave (i.e. um rosto humano), podemos considerar que as δ -coordenadas são a discretização do operador de

Laplace-Beltrami. Com base nisto, podemos escrever o vetor de δ -coordenadas como sendo

$$\delta_i = \frac{1}{N_j} \sum_{j \in N(i)} (\mathbf{v}_i - \mathbf{v}_j). \quad (3.2)$$

Uma possibilidade é adicionar um peso \mathbf{w}_{ij} para cada aresta na lista de adjacências da malha, de forma que a interação entre \mathbf{v}_i e cada vizinho tenha um comportamento particular. Neste caso, a fórmula do vetor de δ -coordenadas é

$$\delta_i = \frac{1}{N_j} \sum_{j \in N(i)} \mathbf{w}_{ij} * (\mathbf{v}_i - \mathbf{v}_j). \quad (3.3)$$

A transformação do vetor de coordenadas cartesianas ao vetor de δ -coordenadas pode ser representada em forma de matriz. Seja \mathbf{Adj} a matriz de adjacências da malha, descrita anteriormente, ou seja, uma matriz quadrada de $n * n$ elementos em que cada elemento \mathbf{Adj}_{ij} da matriz é 1 se há uma aresta em A que conecta o vértice i ao vértice j , e 0 caso contrário. Neste caso, a diagonal principal de \mathbf{Adj} é composta por zeros já que um vértice não pode ser vizinho de si mesmo.

Em seguida definimos a matriz diagonal \mathbf{D} em que cada elemento de \mathbf{D}_{ii} é o grau do vértice i , ou seja, o número de vizinhos do vértice. Na malha da imagem 2, por exemplo, este número seria 2, 3 ou 4. A matriz \mathbf{L} , que representa a matriz transformação das coordenadas cartesianas para as coordenadas diferenciais, é dada por

$$\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1} \mathbf{Adj} \quad (3.4)$$

onde \mathbf{I} é a matriz identidade, \mathbf{D}^{-1} é a inversa da matriz \mathbf{D} e \mathbf{Adj} é a matriz de adjacência. O Laplaciano Topológico da malha \mathbf{M} é a versão simétrica de \mathbf{L} , \mathbf{Ls} , que é mais simples de calcular computacionalmente que \mathbf{L} . A fórmula de \mathbf{Ls} é

$$\mathbf{Ls} = \mathbf{D} - \mathbf{Adj} \quad (3.5)$$

e, portanto, cada célula pode ser calculada como $\mathbf{Ls}_{ij} = N_j$, se $i = j$, $\mathbf{Ls}_{ij} = -1$, caso exista uma aresta entre os vértices i e j , e $\mathbf{Ls}_{ij} = 0$, caso contrário.

Consideremos como exemplo a Figura 3. Uma malha M tridimensional quadrilateral simples, com 9 elementos em V , 12 elementos em A e 4 elementos em F . As matrizes \mathbf{D} , \mathbf{Adj} e \mathbf{Ls} de M são respectivamente:

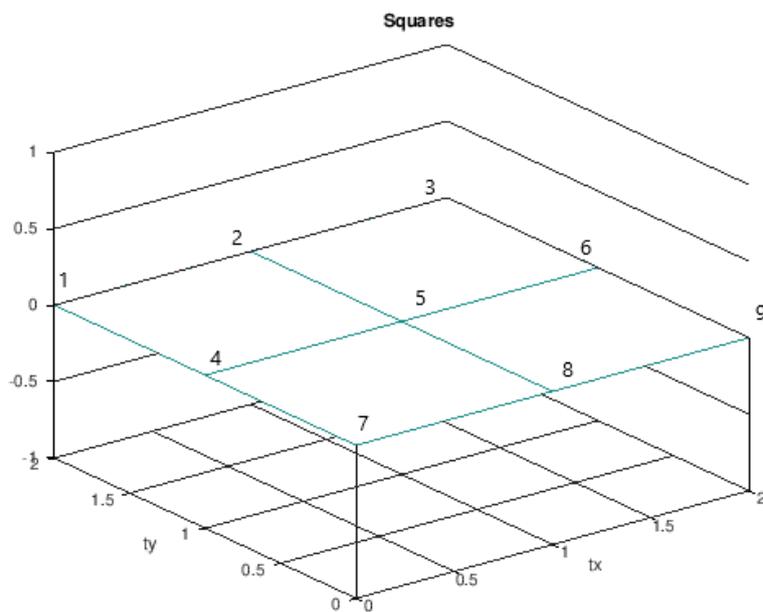


Figura 3 – Uma malha cujas faces são quatro quadrados que será utilizada como exemplo.

$$D = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}$$

$$Adj = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

$$Ls = \begin{bmatrix} 2 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 3 & -1 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & -1 & 4 & -1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & -1 & 3 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 & 3 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 2 \end{bmatrix}$$

A partir do laplaciano topológico, o cálculo das δ -coordenadas a partir das coordenadas cartesianas se dá através da multiplicação de \mathbf{Ls} por cada uma das coordenadas relativas à dimensão da malha M .

$$\mathbf{Lsx} = \delta(x) \quad (3.6)$$

$$\mathbf{Lsy} = \delta(y) \quad (3.7)$$

$$\mathbf{Lsz} = \delta(z) \quad (3.8)$$

Os valores obtidos para $\delta(x)$, $\delta(y)$ e $\delta(z)$ no caso da figura 3 são, respectivamente, $[-1, 0, 1, -1, 0, 1, -1, 0, 1]$, $[1, 1, 1, 0, 0, 0, -1, -1, -1]$ e $[0,0,0,0,0,0,0,0,0]$.

Após as transformações serem concluídas, é necessário converter as δ -coordenadas novamente em coordenadas cartesianas absolutas. Para tal, não é possível simplesmente fazer o processo inverso, pois $x = \mathbf{Ls}^{-1}\delta(x)$ é indefinido, assim como nos eixos y e z . Isso ocorre pois \mathbf{L} e \mathbf{Ls} são singulares e as coordenadas diferenciais são invariantes à translação. Para realizar a transformação, é necessário fixar a posição de alguns vértices, que servirão como âncoras. Fixando os nós 1 e 5 (no exemplo da figura 3), essas âncoras ficarão no final do vetor de δ -coordenadas de forma e o sistema linear gerado será:

$$\bar{L}x = (\delta(x) \cdot c(x)') \quad (3.9)$$

$$\bar{L}y = (\delta(y) \cdot c(y)') \quad (3.10)$$

$$\bar{L}z = (\delta(z) \cdot c(z)') \quad (3.11)$$

onde $c(x, y, z)$ é a coordenada absoluta nos eixos cartesianos dos vértices âncora em ordem. Por exemplo, no caso do eixo x a matriz é:

$$\bar{L} = \begin{bmatrix} 2 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 3 & -1 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & -1 & 4 & -1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & -1 & 3 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 & 3 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 2 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

e o vetor $\delta(x) \cdot c(x)'$ é $[-1, 0, 1, -1, 0, 1, -1, 0, 1, 0, 1]$ e o vetor \mathbf{x} possui as novas posições de \mathbf{x} em coordenadas cartesianas.

Como discutido previamente, é possível ponderar os pesos das δ -coordenadas de forma a modificar o comportamento das arestas de A . No caso padrão, o comportamento de todas as arestas é idêntico, ou seja, o peso delas é constante e, portanto, pode ser simplificado. Com algumas modificações é possível alterar os pesos de cada uma das arestas individualmente, a partir de qualquer princípio que seja interessante adotar para otimizar a qualidade de aproximação durante a transformação.

3.2.1 Edição de Malhas

As técnicas a serem investigadas para alterar a malha \mathbf{M} com o intuito de simular um rosto baseiam-se no Operador de Laplace-Beltrami e coordenadas diferenciais (δ -coordenadas). Adicionalmente, pode-se explorar os princípios de Malhas Dinâmicas para que as arestas de \mathbf{A} simulem o comportamento de molas.

Ambas estas técnicas visam a alteração de malhas localmente, sem a alteração no número de elementos em \mathbf{V} , \mathbf{A} ou \mathbf{F} . Definindo a forma como as coordenadas laplacianas funcionam, o próximo passo é aplicá-las para a edição de malha, assim como definir de qual forma a malha será modificada.

A edição de malhas utilizando o laplaciano se dá de forma similar à recuperação das coordenadas cartesianas a partir do laplaciano. Porém, além das coordenadas dos pontos de âncora, as posições dos pontos que serão editados também são concatenados a Ls e em $\delta(x, y, z)$, representados por \mathbf{I}_e e $e(x, y, z)'$ nas equações abaixo. Assim, o sistema a ser resolvido é dado por:

$$(Ls \cdot I_c \cdot I_e)x = (\delta(x) \cdot c(x)' \cdot e(x)') \quad (3.12)$$

$$(Ls \cdot I_c \cdot I_e)y = (\delta(y) \cdot c(y)' \cdot e(y)') \quad (3.13)$$

$$(Ls \cdot I_c \cdot I_e)z = (\delta(z) \cdot c(z)' \cdot e(z)') \quad (3.14)$$

Sendo I_c a matriz identidade dos pontos âncora escolhidos, $Ls + I_c$ é então a matriz \bar{L} utilizada na subseção anterior e I_e é a matriz identidade dos vértices sendo editados.

Retornando ao exemplo da figura 3, para editar o vértice 5, utilizando os vértices 2 e 8 como âncoras, a coordenada x do sistema a ser resolvido seria a matriz L_e abaixo junto ao vetor $\delta e(x) = [-1, 0, 1, -1, 0, 1, -1, 0, 1, 1, 1, 1.5]$

$$L_e = \begin{bmatrix} 2 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 3 & -1 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & -1 & 4 & -1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & -1 & 3 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 & 3 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 2 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Coordenadas Diferenciais, em específico Coordenadas Laplacianas, permitem que seja efetuada a deformação e manipulação local da malha de forma rápida e robusta, ao mesmo tempo que é preservada a orientação dos detalhes. Após criar uma matriz laplaciana, que tende a ser esparsa, e realizar as modificações nos vértices de interesse (pontos importantes) para simular as curvas características, o método dos mínimos quadrados é utilizado para resolver a matriz laplaciana e obter as coordenadas cartesianas a partir da representação diferencial. Um exemplo desse processo de edição pode ser observado na figura 4, sendo que o processo é análogo para malhas tridimensionais.

As curvas importantes são as curvas que descrevem as características anatômicas de um dado rosto. Por exemplo, curvas Ridge no rosto humano correspondem a características anatômicas marcantes [Kent, Mardia e West \(1996\)](#), como por exemplo a forma do nariz, olhos ou testa. Um exemplo dessas curvas pode ser observada nos rostos ilustrados na Figura 8, do Capítulo 4. Ao obter estas curvas a partir de uma foto de um rosto humano e inseri-las no programa, será possível, ao modelar estas curvas na malha inicial, obter uma aproximação do rosto fotografado.

A criação de uma Malha Dinâmica [Soares \(2007\)](#) consiste na inserção de molas virtuais em uma malha poligonal. Estas molas movem os vértices vizinhos quando um vértice é manipulado, com transformações sendo propagadas pelo sistema de molas para se adaptar à nova descrição de domínio. Estas molas também possuem o objetivo de impedir que formas geométricas inválidas sejam criadas durante a manipulação dos vértices.

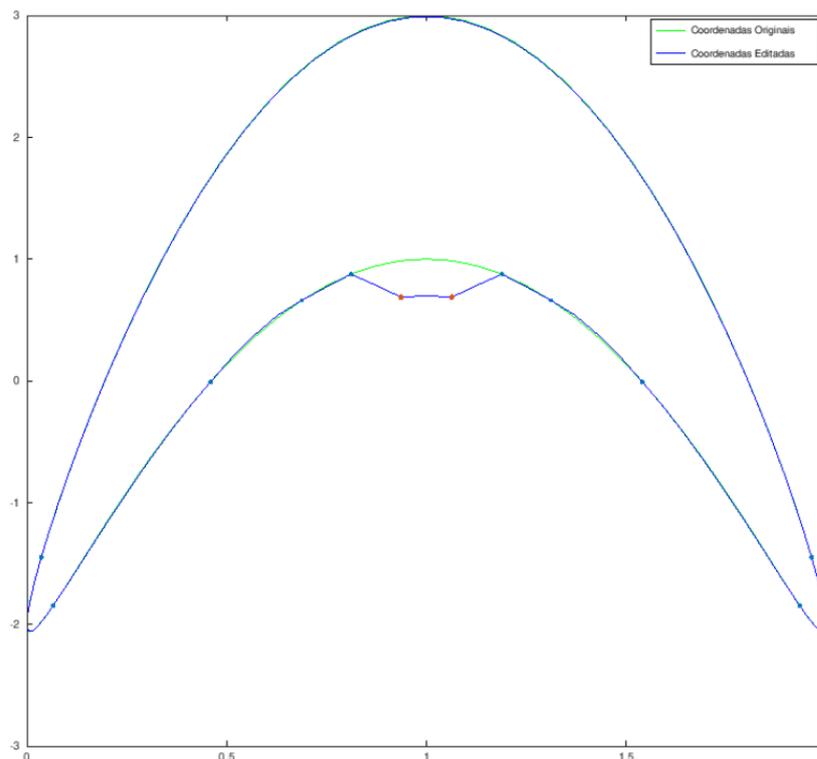


Figura 4 – Exemplo do processo de edição utilizando o operador de Laplace-Beltrami em um conjunto point-based.

3.3 MediaPipe Face Mesh

O projeto "Face Mesh" dentro do conjunto de projetos "MediaPipe" (GOOGLE, 2019) propõe uma metodologia para gerar malhas poligonais a partir de um conjunto de 468 pontos tridimensionais, denominados landmarks, extraídos de fotos ou vídeo, e então permitir que o usuário realize alterações de realidade aumentada em tempo real. Um exemplo destes landmarks dispostos sobre um rosto humano pode ser observado na imagem 10.

Face Mesh usa técnicas de aprendizado de máquina para inferir informações sobre a profundidade de um dado ponto, não sendo necessário equipamento especializado para obter dados tridimensionais. Através do uso de funções disponibilizadas pelo projeto é possível gerar uma malha tridimensional a partir de uma única foto. Estas malhas serão a base sobre a qual o método será aplicado, simplificando o processo de criação de malha para que seja possível focar na parte de modelagem e edição.

O pipeline de aprendizado de máquina utilizado pelo Face Mesh consiste em duas redes neurais profundas implementadas utilizando TensorFlow Lite (TENSORFLOW, 2016), um framework de deep learning open source que converte modelos pré-treinados, que trabalham em conjunto: a) uma rede responsável por detectar posições e elementos importantes no rosto (landmarks) e b) outra que utiliza estas informações para prever a superfície tridimensional do rosto, mesmo quando este não está completamente visível.

Ambas as redes neurais são modeladas utilizando grafos MediaPipe, sendo que o grafo da

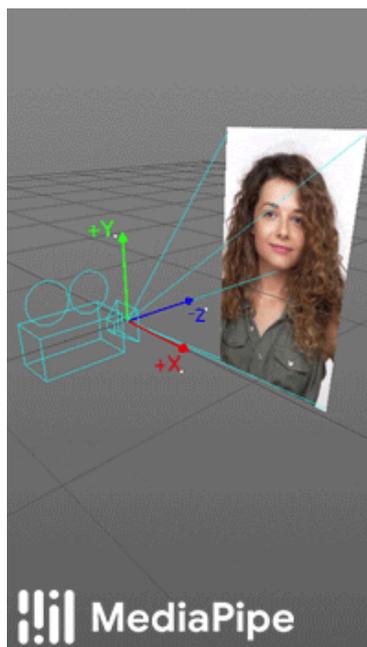


Figura 5 – Exemplo do funcionamento da câmera de perspectiva virtual. Fonte: MediaPipe Face Mesh.

pipeline e subgrafos associados estão disponibilizados pelo conjunto de projetos. No contexto do projeto, os resultados da primeira rede do pipeline servem de base para o processo de modelagem e edição proposto, enquanto a superfície predita pela segunda pipeline pode ser utilizada para comparar os resultados obtidos pelo método.

Dentre os modelos disponibilizados, é de especial relevância ao projeto o modelo de landmarks faciais, responsável por prever informações desconhecidas como as coordenadas e profundidade dos landmarks a partir de seu treinamento.

A rede de landmarks recebe um frame de vídeo ou uma imagem cortada ao redor da face, sem informações adicionais de profundidade e retorna a probabilidade de existir um ou mais rostos razoavelmente alinhados na imagem, mas, de maior interesse ao projeto de pesquisa, ela também retorna o conjunto de todos os pontos e suas posições no espaço 3D.

Para calcular a profundidade baseada em uma imagem bidimensional, o módulo de transformada facial utiliza um espaço métrico tridimensional para receber as primitivas necessárias para ser capaz de trabalhar com a face detectada na imagem como se ela fosse um objeto tridimensional. Este espaço ortonormal métrico coordenado 3D obedece a lei de Flemming, e nele existe uma câmera de perspectiva virtual localizada na origem do espaço, direcionada na direção negativa do eixo Z. A figura 5 ilustra este processo.

Diversos parâmetros desta câmera virtual podem ser manipulados, mas os parâmetros do Face Mesh são os mais próximos de uma câmera real possível, prezando primeiramente os resultados. Dentro do espaço tridimensional também existe um modelo canônico de rosto, um modelo tridimensional estático que segue a mesma topologia de 468 landmarks tridimensionais do modelo de Landmarks Faciais.

O propósito deste modelo é definir as unidades métricas do espaço métrico tridimensional e auxiliar no alinhamento de elementos virtuais de realidade aumentada com um rosto monitorado utilizando uma matriz de transposição. O Pipeline de Transformação, uma estrutura utilizada para realizar cálculos baseada em nó de um grafo, é um de dois componentes do Face Mesh, junto do Renderer de Efeitos. Porém, enquanto o Renderer de Efeitos está mais voltado a técnicas de realidade aumentada, o Pipeline de Transformação é responsável por realizar três tarefas a cada quadro de entrada, obedecendo a seguinte ordem:

- coordenadas de landmarks dos rostos detectados em tela são convertidos em coordenadas do espaço métrico tridimensional;
- a matriz de transposição de pose facial é estimada mapeando linearmente valores a partir do modelo canônico de rosto para o conjunto de landmarks do rosto em tempo de execução;
- uma malha do rosto é gerada, com as posições dos vértices do conjunto de landmarks do rosto em tempo de execução representando valores em coordenadas XYZ, enquanto as coordenadas de vértices de textura UV e a topologia triangular são herdadas do modelo canônico de rosto.

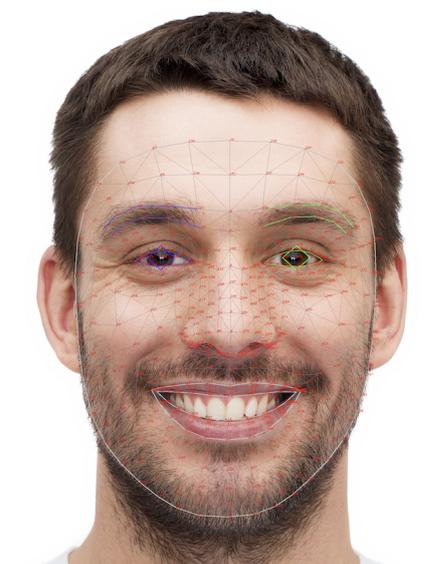


Figura 6 – Exemplo de Detecção de Landmarks com o FaceMesh. Fonte: Google for Developers.

A API do Face Mesh possui um número de argumentos, permitindo adaptabilidade na forma como este é utilizado. Estes argumentos determinam se o Face Mesh deve operar de forma contínua, rodando para cada frame de um vídeo de entrada, ou uma única vez em uma imagem estática, o número de faces máximo que devem ser detectadas a qualquer dado instante, se o Face Mesh deve utilizar um modelo mais computacionalmente custoso, mas que adiciona landmarks extras ao redor das íris de cada olho e distribui landmarks ao redor de áreas como lábios e olhos de maneira mais precisa, qual o grau de confiança que o modelo de detecção de

faces utilizado requer para considerar se uma detecção teve sucesso e, finalmente qual o grau mínimo de confiança necessário para o modelo considerar que ele foi capaz de monitorar um rosto em movimento. A figura 6 exibe um exemplo de como os landmarks ficam dispostos no rosto de uma pessoa. Os parâmetros utilizados para o projeto serão detalhados no capítulo 4.

O retorno da API é uma coleção em que o número de elementos é igual ao número de faces detectadas, e cada elemento é uma lista das posições em x, y e z de cada uma das 468 landmarks no espaço tridimensional. Os valores de x e y estão normalizados entre [0.0] e [1.0], correspondentes à largura e altura da imagem, enquanto o valor de z, representativo da profundidade da landmark, possui origem no centro da cabeça e decresce conforme se aproxima da câmera virtual utilizada para extrapolar as posições das landmarks.

A API do Face Mesh está disponível para C++, JavaScript, Android, iOS, Coral e, de especial relevância para o projeto, Python.

METODOLOGIA

O método proposto é composto de quatro etapas principais, como ilustrado no diagrama da figura 7

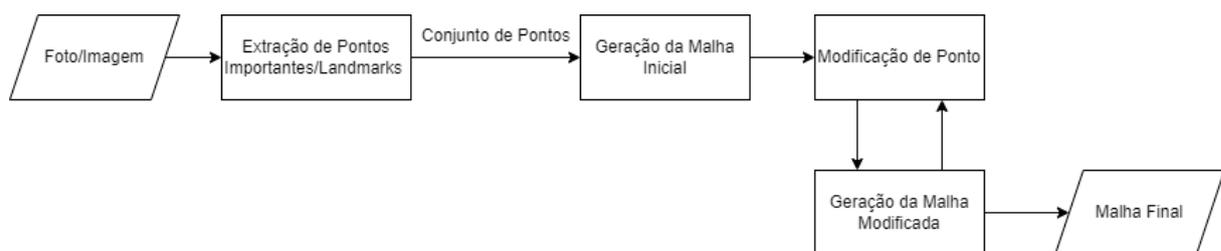


Figura 7 – Diagrama do método proposto

Inicialmente, o usuário fornece a imagem ou foto de um rosto a partir da qual será extraído o conjunto de pontos. Detalhes sobre estes pontos e as curvas da malha que descrevem as características anatômicas mais marcantes do rosto, bem como o processo de extração serão apresentados na seção 4.1. Uma malha tridimensional é então gerada. Este processo será detalhado na seção 4.2.

Subseqüentemente, a malha inicial é alterada nas etapas de “Modificação de Ponto” e “Geração de Malha Modificada” (seção 4.3) de acordo com informações fornecidas como entrada pelo usuário até que ele esteja satisfeito com as modificações realizadas, de forma que a malha resultante ao final do processo é um modelo da face utilizada como base deformada a partir das modificações realizadas pelo usuário.

4.1 Extração de Pontos Importantes

A primeira etapa do método consiste em extrair os 468 pontos importantes que serão utilizados utilizando a API do Face Mesh. Uma imagem ou foto como na figura 9 é utilizada, e o conjunto de pontos é extraído como na imagem de exemplo 10. Cada um desses pontos possui



Figura 8 – Busto de Max Planck com as curvas ridge desenhadas sobre a sua superfície, as curvas ridge extraídas do busto, e uma superfície reconstruída a partir destas curvas. Ohtake, Belyaev e Seidel (2004)

uma posição correspondente em espaço tridimensional, os valores nos eixos X, Y e Z que podem ser editadas pelo usuário em etapas seguintes.

Como mencionado anteriormente no Capítulo 3, a API do Face Mesh disponibilizar um conjunto de parâmetros que determinam o comportamento das aplicações. Alguns parâmetros importantes no contexto deste projeto são :

```
STATIC_IMAGE_MODE = true
MAX_NUM_FACES = 1 (default)
REFINE_LANDMARKS = false (default)
MIN_DETECTION_CONFIDENCE = 0.5 (default)
MIN_TRACKING_CONFIDENCE = 0.5 (default)
```

onde `STATIC_IMAGE_MODE` indica se a entrada é um stream de vídeo (`false`) ou uma imagem estática (`true`), `MAX_NUM_FACES` é o número máximo de faces a serem detectadas em uma dada imagem, `REFINE_LANDMARKS` indica se é necessário usar o modelo “Attention Mesh” para aumentar a acurácia das predições de landmarks em regiões mais importantes do rosto, além de aumentar o número de landmarks, adicionando landmarks ao redor da íris do rosto detectado, `MIN_DETECTION_CONFIDENCE` é o valor mínimo de confiança, entre 0.0 e 1.0 que o modelo de detecção de rostos precisa obter para determinar o que é um rosto válido e `MIN_TRACKING_CONFIDENCE` é o valor mínimo de confiança para o modelo de monitoração de landmarks.

O parâmetro `STATIC_IMAGE_MODE` é verdadeiro pois a aplicação do método utiliza apenas uma imagem como entrada. Da mesma forma, apenas interessa extrair uma única face por imagem, considerando as operações sendo executadas sobre a malha para o usuário. Por este motivo, `MAX_NUM_FACES` é 1, que coincide de ser o valor padrão da entrada. `REFINE_LANDMARKS` é falso, em parte pois a adição de landmarks para a íris é supérflua no contexto do projeto, e também para

manter a execução do método com uma performance superior. `MIN_DETECTION_CONFIDENCE` permanece 0.5, o valor padrão, pois como a entrada pretendida é a foto de um rosto, não há necessidade de modificar este parâmetro. Finalmente, `MIN_TRACKING_CONFIDENCE` é ignorado quando `STATIC_IMAGE_MODE` é verdadeiro, então não é necessário realizar alterações nesta entrada.



Figura 9 – Exemplo de foto que é utilizada como entrada para o método. Vista frontal do rosto.

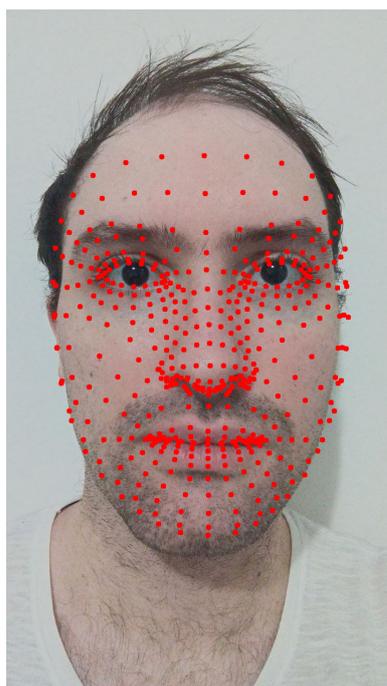


Figura 10 – Exemplo de execução do Face Mesh para a imagem 9



Figura 11 – Exemplo de malha inicial, gerada a partir dos pontos obtidos na imagem 10.

4.2 Criação da Malha Inicial

A partir dos pontos obtidos na etapa anterior, é inicializada a malha tridimensional que será editada durante a etapa de modelagem utilizando o conjunto de pontos e arestas recebidos como entrada. O número de vértices e as arestas entre eles são fixas, utilizando o mesmo padrão que o Face Mesh.

Para gerar a malha utilizando a biblioteca plotly é necessário enviar a posição dos pontos recebidos do Mediapipe, e, através do frozenset do face mesh que descreve todas as arestas do conjunto, obter todas as faces (i.e., triângulos) que compõe a malha. É possível gerar a malha sem enviar as informações de faces do Mediapipe, porém a malha resultante neste caso usa heurísticas tal qual triangulação de Delaunay para conectar os vértices. Considerando que uma das vantagens do método proposto é este não alterar as arestas existentes, é de maior interesse utilizar as arestas propostas pelo Face Mesh.

A partir dessas informações, podemos exibir a malha gerada pelo rosto da imagem de entrada, a partir da qual iremos realizar um ciclo de operações descritos na próxima seção.

Um exemplo de malha inicial está ilustrada na figura 11. Como o foco do projeto de pesquisa é o rosto, não está no escopo utilizar uma malha com o formato completo de uma cabeça humana, ou incluir na malha elementos como orelhas ou cabelo. O resultado é uma malha que se assemelha a uma máscara, um semi hemisfério, da testa até o queixo e lateralmente até antes das orelhas.

4.3 Modelagem / Manipulação da Malha

Utilizando a malha inicial genérica criada no passo anterior como base, o método de coordenadas laplacianas será utilizado para manipular a malha com base em uma função que permite que o usuário escolha um vértice e altere a sua posição. Considerando que curvas e pontos importantes descrevem os traços anatômicos mais marcantes de uma determinada face humana [Kent, Mardia e West \(1996\)](#), ao replicar estas curvas e pontos na malha ela irá adquirir os traços e feições desta face. O formato da malha será uma aproximação do formato do rosto, embora sem detalhes que não produzam curvas importantes que possam ser extraídas e fornecidas ao programa, como por exemplo pintas e cabelo facial.

Os pontos de maior curvatura nas faces humanas tendem a ser ao redor dos olhos, o nariz e a boca. Desta forma, convém dedicar especial atenção ao lidar com estas regiões, considerado que, devido à alta concentração de pontos nestas regiões, modificações em pontos centrais serão propagadas aos seus vizinhos.

Com o uso de δ -coordenadas, o processo de modelagem irá gradualmente propagar as modificações feitas em um vértice v_i aos vértices vizinhos, evitando picos e vales não naturais. O usuário informa qual o vértice que ele deseja alterar e o grau de propagação que ele deseja utilizar, assim como as novas coordenadas do ponto.

Grau 0 de propagação é simplesmente uma modificação na posição de dado ponto, sem afetar os vizinhos. Isso pode ser realizado facilmente sem o uso de δ -coordenadas, e foi adicionado para a realização de testes simples. Grau 1 de propagação, o padrão desta implementação, utiliza δ -coordenadas para afetar todos os vizinhos diretos do vértice v_i . Grau 2 de propagação, implementado para motivos de teste, faz com que, quando os vizinhos do vértice v_i são movidos, estes por sua vez afetem os seus próprios vizinhos.

Teoricamente é possível repetir este processo para graus maiores, porém a complexidade do método aumenta exponencialmente conforme cada mudança afeta ainda mais vizinhos. Por este motivo o grau 2 foi o maior grau implementado.

Após cada modificação realizada pelo usuário a malha pode ser recriada utilizando o mesmo processo de criação da malha inicial, porém, utilizando as novas posições dos pontos. Como mencionado anteriormente, não ocorre modificações nas arestas, desta maneira são aproveitadas as informações de faces obtidas anteriormente para gerar a malha. Após observar o resultado, o usuário pode novamente modificar um novo vértice na malha, repetindo estas duas etapas do processo.

4.4 Materiais

O único material requerido do usuário é uma imagem que ele deseja utilizar como base para a malha a ser editada. Realizar testes e modificações em parâmetros na aplicação baseada

no método requer a instalação de módulos e o uso de Python na máquina utilizada.

4.5 Considerações Éticas

Devido a questões éticas provenientes do uso de faces humanas de terceiros no projeto, as imagens utilizadas nos testes referenciados e exemplos neste capítulo são fotos do pesquisador. Imagens de terceiros foram utilizadas para obter indicadores como quais landmarks possuem maior variância de posições relativas em faces humanas, porém, para proteger a privacidade das pessoas cujas faces foram utilizadas para extrair estas métricas o conjunto de imagens não foi disponibilizado e as imagens não são exibidas.

RESULTADOS

Neste capítulo serão apresentados alguns resultados que, a partir de um conjunto de pontos importantes extraídos de uma foto, gera a malha de um rosto com formato similar ao utilizado como amostra. A malha então pode ser editada pelo usuário alterando-se a posição de qualquer um dos 468 vértices que a compõe, um por vez, porém sem restrição no número de vezes que esta operação pode ser realizada, sejam estas repetidas operações em um único vértice ou operações em vértices distintos.

5.1 Configuração dos testes

Os testes realizados empregam diferentes fotos do pesquisador. Para cada foto, uma malha foi gerada. A partir de cada malha, três vértices foram selecionados: a) um vértice com 3 arestas, o grau mínimo de vértices na malha; b) outro com 8 arestas, o grau máximo de vértices na malha e c) um vértice com 6 arestas, a moda dos graus da malha.

Para cada teste, os três tipos de vértices foram escolhidos arbitrariamente entre os vértices que atendiam cada um dos três critérios mencionados. Cada um destes vértices foi modificado duas vezes:

1. a primeira com uma modificação mínima, para avaliar a funcionalidade do método e computar o tempo necessário para gerar a malha novamente. Essas modificações foram feitas em sequência, para garantir que modificações são persistentes.
2. A segunda modificação foi realizada com valores extremos, para testar a robustez do método e das malhas geradas pelo mesmo.

O movimento realizado em cada vértice no eixo X , Y e Z foram os mesmos em cada teste, com o objetivo de manter consistência. Esses valores são, respectivamente, $(-0.005, -0.005, -0.005)$ no teste de modificação suave e $(0.1, 0.1, -0.1)$ no teste de modificação extrema.

Para cada um dos testes são ilustradas a imagem utilizada, a imagem com as landmarks assinaladas, imagens de cada malha em três momentos. São também descritos o momento de criação da malha inicial, após a primeira rodada de modificações e após a segunda rodada de modificações e finalmente uma tabela de resultados contendo o tempo de execução de cada uma das seguintes tarefas:

- Extração de Landmarks da foto
- Criação da malha inicial
- Modificação de vértice de grau 3
- Modificação de vértice de grau 6
- Modificação de vértice de grau 8
- Modificação de vértice de grau 3 com valores extremos
- Modificação de vértice de grau 6 com valores extremos
- Modificação de vértice de grau 8 com valores extremos

O tempo necessário para renderizar a malha não está sendo considerado como parte do tempo de criação de malha inicial e modificação de vértice.

5.1.1 Primeiro Teste

O primeiro teste foi realizado com a imagem apresentada na figura 9, uma expressão neutra. Na etapa de extração de landmarks, a posição dos pontos importantes para a geração da malha, assim como a cor destes pontos, foram armazenados, como dispostos na figura 12. Finalmente, a malha inicial 11 foi gerada com base nestas informações.

Os vértices escolhidos para este teste foram:

- 78, vizinho de 62, 95 e 191.
- 228, vizinho de 25, 31, 110, 117, 118 e 229.
- 358, vizinho de 266, 279, 294, 327, 331, 371, 423 e 429.

Após a primeira rodada de edições, a aparência da malha pode ser verificada na figura 13.

Após a segunda rodada de edições, a aparência da malha pode ser verificada na figura 14

O tempo necessário para realizar cada um destes passos pode ser observado na tabela 1.

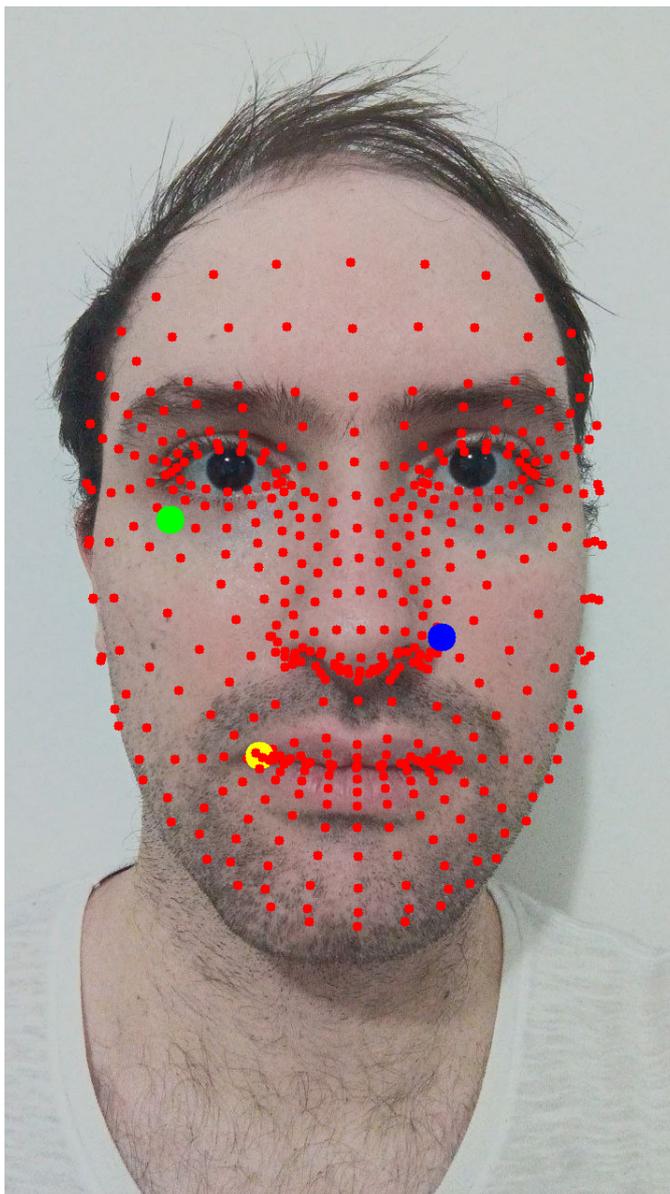


Figura 12 – Exemplo de execução do Face Mesh para a imagem 9, com os landmarks 78, 228 e 358 marcados em amarelo, verde e azul respectivamente.

5.1.2 Segundo Teste

O segundo teste foi realizado com a foto da figura 5, uma expressão exagerada. Na etapa de extração de landmarks, a disposição dos pontos foi dada conforme a foto na figura 5

Os vértices escolhidos para este teste foram:

- 308, vizinho de 292, 324 e 415.
- 50, vizinho de 101, 117, 118, 123, 187 e 205
- 13, vizinho de 12, 14, 38, 82, 87, 268, 312 e 317.



Figura 13 – Malha após as modificações intermediárias.



Figura 14 – Malha após as modificações finais.

Após a primeira rodada de edições, a aparência da malha pode ser verificada na figura 13.

Após a segunda rodada de edições, a aparência da malha pode ser verificada na figura 14. O tempo necessário para realizar cada um destes passos pode ser observado na tabela 2.

Etapa	Tempo de Execução (Segundos)
Extração de Landmarks	0.218615 seconds
Geração da Malha Inicial	0.003471
Modificação Vértice 78	0.399355
Modificação Vértice 228	0.364520
Modificação Vértice 358	0.366581
Modificação (Maior) Vértice 78	0.393232
Modificação (Maior) Vértice 228	0.387505
Modificação (Maior) Vértice 358	0.384684

Tabela 1 – Tabela com os resultados do primeiro experimento.



Figura 15 – Foto utilizada como entrada no segundo teste.

Etapa	Tempo de Execução (Segundos)
Extração de Landmarks	0.214413
Geração da Malha Inicial	0.007095
Modificação Vértice 308	0.362755
Modificação Vértice 50	0.389365
Modificação Vértice 13	0.401822
Modificação (Maior) Vértice 308	0.363234
Modificação (Maior) Vértice 50	0.374863
Modificação (Maior) Vértice 13	0.584279

Tabela 2 – Tabela com os resultados do segundo experimento.

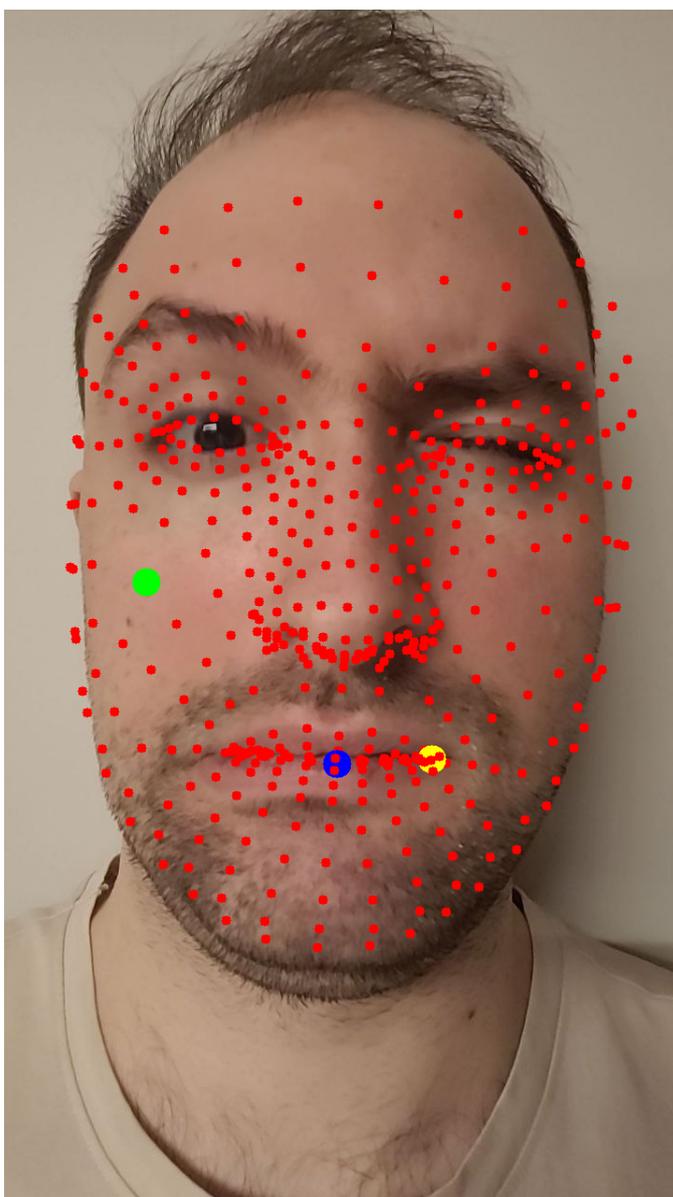


Figura 16 – Exemplo de execução do Face Mesh para a imagem 15, com os landmarks 308, 50 e 13 marcados em amarelo, verde e azul respectivamente.



Figura 17 – Malha inicial do segundo teste, gerada a partir da foto 16.



Figura 18 – Malha após as modificações intermediárias.



Figura 19 – Malha após as modificações finais.

CONCLUSÕES

As malhas geradas pelo método proposto são uma alternativa para realizar e propagar modificações pela malha quando a posição de um dos pontos específicos no rosto é alterada. Estas malhas preservam as suas arestas originais e resistem deformações, devido às características das δ -coordenadas utilizadas. Porém, devido à liberdade oferecida ao usuário do método, a malha pode ser deformada até perder a semelhança com a malha original e, portanto, o rosto original, porém contanto que estas alterações permaneçam dentro do escopo do programa e obedeçam as limitações de espaço em que a malha está submetida, isto é, as posições válidas no eixo X e Y para os pontos são valores entre 0.0 e 1.0, esta liberdade é oferecida por design, não levando em consideração a similaridade da malha resultante a um rosto real.

Os limites de largura e altura da malha serem iguais acaba por fazer com que as malhas geradas através do método se tornem mais achatadas. É possível observar a olho nu comparando, por exemplo, as figuras 9 e 11 que, embora o rosto utilizado como entrada seja mais ovalado, as malhas geradas a partir dos pontos extraídos são mais circulares. Observando o modelo canônico de rosto do Face Mesh em 20, que foi utilizado como base para a implementação do método, é possível perceber como ele é, de fato, arredondado.

De acordo com os testes realizados, o tempo de extração de landmarks e de geração de malha não possuem muita variância de acordo com as fotos, sendo mais sensíveis ao processamento do computador.

Da mesma forma, o tempo de modificação de vértice também foi relativamente estável, independente do grau do vértice sendo modificado ou do tamanho da modificação sendo realizada, sendo muito mais sensível ao núcleo em que o método está sendo executado.

Como o método não contempla uma interface de usuário, a aplicação não disponibiliza nenhuma UI para o usuário e toda interação é realizada rodando funções diretamente nas células do notebook da aplicação, uma vez que a criação e uso de interfaces não é uma necessidade para a execução das etapas, mas, caso seja necessário, uma interface simples pode ser criada para

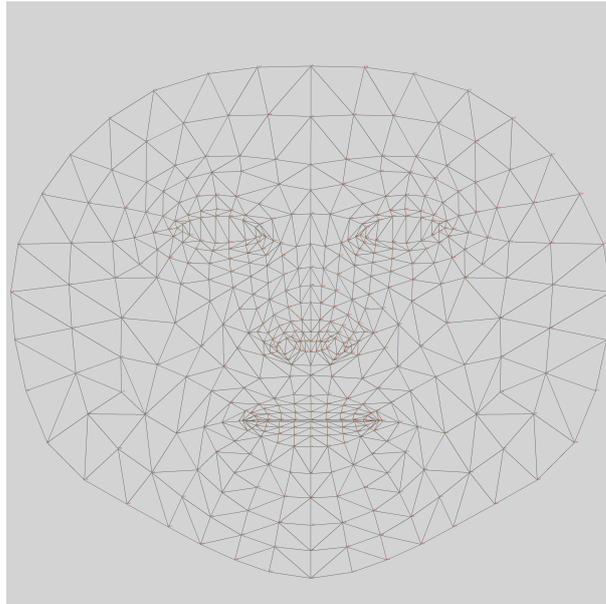


Figura 20 – O modelo canônico da malha utilizado pelo Face Mesh. Este modelo ilustra as conexões entre os landmarks. Fonte: Developers.google

permitir, por exemplo, que um usuário escolha um dos pontos e altere sua posição de maneira interativa.

6.1 Trabalhos Futuros

Seria possível expandir o contexto de trabalhos futuros para englobar não apenas rostos mas o resto da cabeça. Por exemplo, unindo a malha de rosto à um modelo tridimensional de uma cabeça humana, a combinação de ambas seria a representação da cabeça de um humano, como no exemplo da figura 21.

Uma possibilidade para otimizar o processo de modelagem seria utilizar um eixo de simetria vertical na malha, uma vez que simetria ocorre naturalmente em parte dos elementos da face humana, de forma que modificações feitas em um ponto de um lado da malha fossem espelhadas em um ponto simetricamente oposto na outra metade, com ambos os pontos formando um par de forma que, conforme um é modificado em relação ao eixo de simetria, o seu par espelharia esta modificação, porém há alguns problemas na proposta desta adição ao método. Embora a face humana seja, em termos gerais, simétrica, fatores como cicatrizes, deformações decorrentes de patologias e lesões em geral que afetam apenas um lado da face removem este fator de simetria. Para evitar que estas deformações sejam ignoradas ou duplicadas, é necessário modelar ambos os lados independentemente, ao invés de utilizar o espelhamento para simplificar o método. Ainda assim, permitir espelhamento como uma opção que pode ser utilizada baseada em uma entrada é uma possibilidade para pontos que possuam pares em posições simétricas do rosto.

Outra alteração que pode ser utilizada para aperfeiçoar o processo de modelagem,

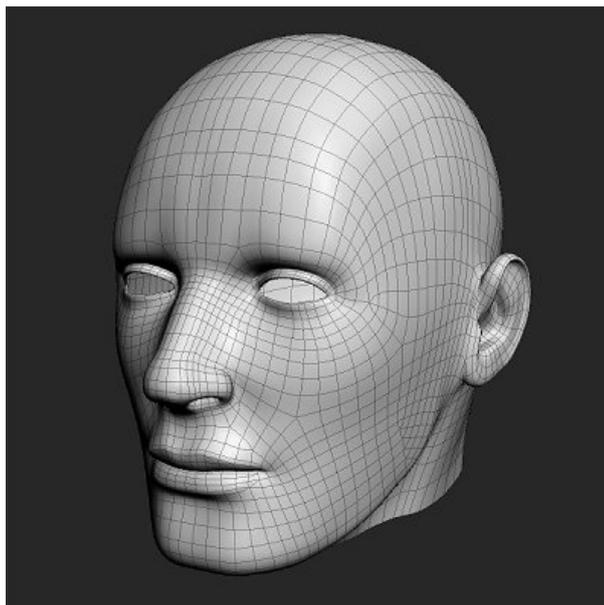


Figura 21 – Um exemplo de uma malha com o formato de uma cabeça humana genérica. Fonte: Mad Mouse Design, 2007

como discutido na seção 3, é o uso de pesos nas arestas durante o cálculo das coordenadas diferenciais. Por exemplo, ao utilizar arestas cujo peso é ditado pela lei de Hooke, e assim simular o comportamento de uma mola, onde a constante de resistência de uma dada aresta é inversamente proporcional ao tamanho desta aresta, é possível combinar aspectos de Malhas Dinâmicas com o uso de Coordenadas Laplacianas. Assim, arestas mais curtas são mais resistentes, e transferem transformações de forma direta de uma ponta à outra, enquanto arestas mais longas propagam transformações de forma mais branda ao vértice vizinho. Outros tipos de pesos podem ser utilizados, tanto para motivos de teste, para ver qual opção fornece a maior similaridade quando comparada ao modelo, como para outros motivos, como exagerar feições ou aumentar o detalhamento de certos traços faciais. Esta adaptação do método pode ser estudada para verificar o efeito dela no processo de edição da malha, permitindo o uso de pesos diferentes.

REFERÊNCIAS

A-NASSER, A.; MOHAMMAD, M.; MOHAME, A.-M. 3d face mesh modeling for 3d face recognition. In: *State of the Art in Face Recognition*. I-Tech Education and Publishing, 2009. Disponível em: <<https://doi.org/10.5772/6643>>. Citado na página 20.

AFZAL, H. M. R.; LUO, S.; AFZAL, M. K.; CHAUDHARY, G.; KHARI, M.; KUMAR, S. A. P. 3d face reconstruction from single 2d image using distinctive features. **IEEE Access**, Institute of Electrical and Electronics Engineers (IEEE), v. 8, p. 180681–180689, 2020. Disponível em: <<https://doi.org/10.1109/access.2020.3028106>>. Citado na página 20.

BOTSCH, M.; PAULY, M.; KOBELT, L.; ALLIEZ, P.; LÉVY, B.; BISCHOFF, S.; RÖSSL, C. Geometric modeling based on polygonal meshes. In: **ACM SIGGRAPH 2007 courses on - SIGGRAPH '07**. ACM Press, 2007. Disponível em: <<https://doi.org/10.1145/1281500.1281640>>. Citado na página 15.

GOOGLE. **MediaPipe Face Mesh**. 2019. Disponível em: <https://google.github.io/mediapipe/solutions/face_mesh.html>. Citado na página 29.

KENT, J.; MARDIA, K.; WEST, J. Ridge curves and shape analysis. In: **Proceedings of the British Machine Vision Conference 1996**. British Machine Vision Association, 1996. Disponível em: <<https://doi.org/10.5244/c.10.26>>. Citado nas páginas 16, 28 e 37.

KITTLER, J.; HUBER, P.; FENG, Z.-H.; HU, G.; CHRISTMAS, W. 3d morphable face models and their applications. In: **Articulated Motion and Deformable Objects**. Springer International Publishing, 2016. p. 185–206. Disponível em: <https://doi.org/10.1007/978-3-319-41778-3_19>. Citado nas páginas 15 e 19.

LIPMAN, Y.; SORKINE, O.; COHEN-OR, D.; LEVIN, D.; ROSSL, C.; SEIDEL, H.-P. Differential coordinates for interactive mesh editing. In: **Proceedings Shape Modeling Applications, 2004**. IEEE, 2004. Disponível em: <<https://doi.org/10.1109/smi.2004.1314505>>. Citado na página 19.

OHTAKE, Y.; BELYAEV, A.; SEIDEL, H.-P. Ridge-valley lines on meshes via implicit surface fitting. **ACM Transactions on Graphics**, Association for Computing Machinery (ACM), v. 23, n. 3, p. 609–612, ago. 2004. Disponível em: <<https://doi.org/10.1145/1015706.1015768>>. Citado nas páginas 9 e 34.

PAGÉS, R.; BERJÓN, D.; MORÁN, F. Automatic system for virtual human reconstruction with 3d mesh multi-texturing and facial enhancement. **Signal Processing: Image Communication**, Elsevier BV, v. 28, n. 9, p. 1089–1099, out. 2013. Disponível em: <<https://doi.org/10.1016/j.image.2013.07.001>>. Citado na página 20.

PLOTLY. **Plotly**. 2013. Disponível em: <<https://plotly.com/graphing-libraries/>>. Citado na página 22.

- RICHMOND, S.; HOWE, L. J.; LEWIS, S.; STERGIAKOULI, E.; ZHUROV, A. Facial genetics: A brief overview. **Frontiers in Genetics**, Frontiers Media SA, v. 9, out. 2018. Disponível em: <<https://doi.org/10.3389/fgene.2018.00462>>. Citado na página 15.
- SOARES, I. P. **Movimento de malhas e remalhamento de malhas superficiais**. Tese (Doutorado) — ICMC-USP, São Carlos, 2007. Citado nas páginas 15, 19 e 28.
- SORKINE, O. Differential representations for mesh processing. **Computer Graphics Forum**, European Association for Computer Graphics, v. 25, n. 4, p. 789–807, 2006. Citado nas páginas 19 e 23.
- SULLIVAN, B.; AL, A. K. et. **PyVista**. 2019. Disponível em: <<https://docs.pyvista.org/>>. Citado na página 22.
- TEAM, O. **Open3D**. 2019. Disponível em: <<https://pypi.org/project/open3d/#description>>. Citado na página 22.
- TENSORFLOW. **TensorFlow Lite**. 2016. Disponível em: <<https://www.tensorflow.org/lite/guide>>. Citado na página 29.
- ZHANG, S.; XIAO, N. Detailed 3d human body reconstruction from a single image based on mesh deformation. **IEEE Access**, Institute of Electrical and Electronics Engineers (IEEE), v. 9, p. 8595–8603, 2021. Disponível em: <<https://doi.org/10.1109/access.2021.3049548>>. Citado na página 20.

