

UNIVERSIDADE DE SÃO PAULO  
Instituto de Ciências Matemáticas e de Computação

## **Reconstrução de Superfícies 3D em Rostos Humanos: uma abordagem Laplaciana**

Relatório final de iniciação científica referente  
ao processo FAPESP nº 2023/15751-3 para o  
período entre 01/01/2024 e 31/12/2024.

**Bolsista:** João Pedro Viguini Tolentino Taufner Correa

**Orientador:** João do Espírito Santo Batista Neto

Brasil

2024

# Sumário

<b>1</b>	<b>Introdução</b>	<b>3</b>
1.1	O projeto e o plano de trabalho	3
<b>2</b>	<b>Revisão Bibliográfica</b>	<b>4</b>
2.1	Características robustas na reconstrução de superfícies	4
2.2	Reconstrução Laplaciana de malhas	5
2.3	Ferramentas computacionais para reconstrução de superfícies	5
2.4	Comparando diferentes abordagens	5
<b>3</b>	<b>Atividades realizadas</b>	<b>6</b>
3.1	Extração da nuvem de pontos e seleção dos pontos âncora	7
3.2	Obtenção das características robustas	8
3.2.1	Curvas parabólicas	9
3.2.2	Curvas <i>ridges</i>	11
3.3	Criação da nova malha	11
3.4	Reconstrução da superfície	13
3.4.1	Escolha dos pontos âncora	13
3.5	Texturização da malha	14
3.6	Resultados obtidos	14
3.6.1	Curvas parabólicas	15
3.6.2	Curvas <i>ridges</i>	16
<b>4</b>	<b>Considerações finais</b>	<b>16</b>
4.1	Participação em eventos científicos	17
	<b>Referências</b>	<b>18</b>

# 1 Introdução

Este projeto de Iniciação Científica é parte integrante do projeto temático "Teoria de Singularidades e Aplicações à Geometria Diferencial, Equações Diferenciais e Visão Computacional" (FAPESP 2019/07316-0). Dentre as quatro linhas de atuação do projeto, uma diz respeito à reconstrução de faces humanas a partir de informações reduzidas do domínio. Este documento tem como objetivo apresentar as atividades realizadas pelo bolsista João Pedro Viguini Tolentino Taufner Correa, no período de janeiro de 2024 a dezembro de 2024, com um foco nas atividades de agosto em diante. Este projeto (FAPESP nº 2023/15751-3) intitula-se “Reconstrução de Superfícies 3D em Rostos Humanos: uma abordagem Laplaciana”.

## 1.1 O projeto e o plano de trabalho

A reconstrução de malhas a partir de um conjunto reduzido de pontos constitui o núcleo deste trabalho. O objetivo é reconstruir uma face humana (incluindo nariz, boca, olhos, entre outros elementos) a partir de uma versão simplificada desta, como uma caricatura representada por curvas simplificadas da face. Essas informações reduzidas são denominadas características robustas - termo cunhado por Ian Porteus - que são características que se mantêm quando a superfície se deforma (IZUMIYA et al., 2015). Elas serão extraídas da malha tridimensional de um rosto humano e, por meio do operador Laplaciano, utilizadas para reconstruir a superfície 3D.

As seguintes atividades foram realizadas durante o período de janeiro a dezembro de 2024:

- Cálculo da nuvem de pontos 3D de um rosto humano utilizando o Mediapipe (LUGARESI et al., 2019);
- Extração das curvas parabólicas e *ridges* (características robustas) pelo método desenvolvido por Danilo Marques;
- Geração da nova malha incluindo as características robustas.
- Reconstrução da malha a partir apenas das características robustas utilizando operadores Laplacianos, conforme sugerido por Sorkine (2006);
- Avaliar a qualidade da reconstrução variando a quantidade de pontos importantes empregados.

Para a realização dessas tarefas, primeiramente realizou-se estudos sobre a obtenção das curvas parabólicas e *ridges*, previamente calculadas. De posse das curvas, foi necessário construir uma nova malha que inclui os novos pontos. A partir dessa malha, a nova matriz de adjacência é utilizada para a aplicação do algoritmo de reconstrução.

O cronograma foi atualizado com frequência no decorrer da segunda metade do projeto, principalmente com relação à criação da nova malha antes da etapa de reconstrução. Essa etapa foi incluída após os testes realizados na primeira metade demonstrarem que a malha reconstruída não estava seguindo fielmente as especificações do projeto. Com essa correção, o cronograma seguiu a disposição da Tabela 1. Em negrito estão as atividades realizadas na segunda metade do projeto.

Atividades	Meses de trabalho					
	1º e 2º	3º e 4º	5º e 6º	7º e 8º	9º e 10º	11º e 12º
Estudo das técnicas de reconstrução de curvas	•	•				
Estudo das bibliotecas do MediaPipe		•	•			
Extração/Adaptação dos pontos importantes		•	•			
Redação do relatório parcial			•			
<b>Geração da nova malha</b>			•	•	•	
<b>Implementação da reconstrução 3D</b>				•	•	
<b>Avaliação e relatório final</b>					•	•

Tabela 1 – Cronograma de atividades para 12 meses de trabalho.

## 2 Revisão Bibliográfica

A reconstrução de superfícies tridimensionais é uma área de pesquisa amplamente estudada, com aplicações que abrangem desde a computação gráfica até o reconhecimento facial e a realidade aumentada. Nesta seção, apresentamos um panorama sobre os principais avanços, conceitos e trabalhos correlatos que fornecem a base para o desenvolvimento deste projeto.

### 2.1 Características robustas na reconstrução de superfícies

O uso de características robustas, como curvas parabólicas e *ridges*, tem se consolidado como uma abordagem eficaz na análise de superfícies. Bruce, Giblin e Tari (1996) destacaram a importância das *ridges* na detecção de padrões geométricos em superfícies, fornecendo uma base matemática para sua aplicação em visão computacional.

Curvas parabólicas, definidas como o conjunto de pontos onde a curvatura gaussiana muda de sinal, também desempenham um papel importante na análise geométrica.

## 2.2 Reconstrução Laplaciana de malhas

Os operadores Laplacianos têm sido amplamente utilizados em tarefas de processamento de malhas, conforme explorado por Sorkine et al. (2005). A representação diferencial possibilita capturar informações locais e globais da superfície, sendo fundamental para o refinamento e suavização das malhas reconstruídas. Este trabalho também fundamenta o uso de matrizes de adjacência para representar relações entre vértices e refinar a geometria da malha.

Na literatura recente, Fakhoury (2021) aplicou operadores Laplacianos na reconstrução de curvas bidimensionais utilizando características robustas, demonstrando a viabilidade de estender tal metodologia para o espaço tridimensional. Além disso, Angarola (2024) destacou a importância de otimizar os algoritmos de cálculo de curvaturas, especialmente para lidar com grandes volumes de dados em aplicações práticas.

## 2.3 Ferramentas computacionais para reconstrução de superfícies

Frameworks como o Mediapipe (LUGARESI et al., 2019) têm se mostrado ferramentas robustas para a extração de pontos faciais tridimensionais, permitindo a geração de nuvens de pontos com alta precisão.

## 2.4 Comparando diferentes abordagens

A reconstrução de curvas ou superfícies a partir de pontos de controle é de fundamental importância em áreas como computação gráfica, geometria computacional, visão computacional e processamento de imagens. Boa parte de soluções neste campo se baseiam em representações do tipo *spline* (SCHOENBERG, 1946). Técnicas baseadas em *spline* consistem basicamente de um processo de minimização da distância entre a curva de referência e o resultado obtido e são ditas interpolatórias, uma vez que a curva resultante passa por todos os pontos de controle.

Uma segunda categoria de reconstrução de curvas são as baseadas em aproximação em que não necessariamente a curva ou superfície é reconstruída passando pelos pontos de controle. Lee (2000) propõe um método baseado em mínimos quadrados combinado à uma árvore geradora mínima Euclidiana para, a partir de um conjunto de pontos não organizados, aproximar uma curva.

A técnica de reconstrução explorada neste trabalho é descrita por Sorkine (2006). A abordagem é baseada no operador de Laplace e representações diferenciais entre os vértices de uma dada vizinhança em uma malha de pontos. Em contraste com tradicional representação por coordenadas cartesianas globais, a representação diferencial de uma superfície

revela informações sobre a forma local da mesma, bem como o tamanho e orientação de detalhes locais. Além de prover informações que resultarão em uma reconstrução com maior preservação de detalhes, trata-se de um sistema linear, o que é computacionalmente eficiente.

### 3 Atividades realizadas

Nesta etapa, detalham-se as tarefas realizadas pelo bolsista durante o período de vigência do projeto. As seções 3.1 e 3.2 foram desenvolvidas exclusivamente durante a primeira metade do cronograma, enquanto as seções 3.3 a 3.6 se referem à segunda metade do projeto.

Na seção 3.1, é mostrado como a nuvem de pontos foi obtida utilizando o Mediapipe para gerar as curvas parabólicas e *ridges* utilizadas na reconstrução da superfície. Essas curvas representam as características robustas, que serão fundamentais para selecionar os pontos âncora na reconstrução.

Na seção 3.2, apresenta-se a extração das curvas parabólicas e *ridges* por meio do programa desenvolvido em MATLAB por Danilo Marques durante seu doutorado.

Na seção 3.3, mostramos como a nova malha foi criada com o intuito de obter a matriz de adjacência atualizada com os vértices da curva. Essa matriz de adjacência é utilizada pelo algoritmo de reconstrução para capturar informações a respeito da vizinhança de cada vértice.

Na seção 3.4, aplica-se a reconstrução da superfície por meio do operador Laplaciano - conforme sugerido por Sorkine (2006). O método consiste em calcular as coordenadas diferenciais dos pontos âncora para reconstruir as curvas que definem a superfície 3D.

A seção 3.5 apresenta a texturização da malha. Esse incremento na visualização permitiu avaliar melhor os resultados com as cores reais do rosto em questão. Isso fez a malha ficar mais fiel a um rosto humano, o que facilita a análise da qualidade da reconstrução.

Na última seção, discutem-se os resultados obtidos ao variar a quantidade de pontos âncora selecionados e o tipo de curva utilizada para a reconstrução. A qualidade da reconstrução foi avaliada qualitativamente levando em conta a fidelidade da malha reconstruída em relação à malha original, e quantitativamente, considerando o erro (Distância Euclidiana) entre a malha original e a malha reconstruída.

### 3.1 Extração da nuvem de pontos e seleção dos pontos âncora

Para a obtenção da nuvem de pontos tridimensionais do rosto, utilizou-se a ferramenta Mediapipe, que emprega modelos de aprendizado de máquina para detectar pontos de referência e expressões faciais a partir de imagens ou sequências de imagens (vídeos). A malha gerada servirá como ponto de partida para a obtenção das características robustas, bem como a nova matriz de adjacência.

O dado de entrada para a extração da nuvem de pontos foi uma imagem (PNG, JPG ou JPEG) de um rosto humano na posição frontal. Optou-se por escolher rostos com expressões faciais relativamente semelhantes para facilitar a comparação da malha original com a malha reconstruída entre cada rosto.

As imagens de entrada foram selecionadas do repositório público WIDER FACE: A Face Detection Benchmark (YANG et al., 2016).



Figura 1 – Nuvem de pontos.

Por meio da rotina de obtenção de pontos faciais do Mediapipe - *MediaPipe Face Landmarker* - o código fonte para essa tarefa extrai a nuvem de pontos 3D da imagem do rosto e salva esses pontos em um arquivo OBJ. Os modelos são capazes de detectar 468 pontos distribuídos pelo rosto. Arquivos OBJ são um formato padrão de arquivo de definição de geometria, utilizado para representar a superfície 3D de um objeto. Eles contêm informações sobre vértices e faces que compõem a geometria do objeto.

Este trecho do código mostra como a nuvem de pontos foi gerada:

```
1 # Extrai a nuvem de pontos usando o Face Landmark Detection (MediaPipe)
2 def extract_point_cloud(frame):
3     mp_face_mesh = mp.solutions.face_mesh
4     face_mesh = mp_face_mesh.FaceMesh()
```

```

5
6     rgb_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
7     points = []
8     results = face_mesh.process(rgb_frame)
9     if results.multi_face_landmarks:
10         for face_landmarks in results.multi_face_landmarks:
11             for i, landmark in enumerate(face_landmarks.landmark, start
=0):
12                 x, y, z = landmark.x, landmark.y, landmark.z
13                 points.append((x, y, z))
14     return points
15
16 # Extrai as faces que definem as conexoes de cada vertice
17 def extract_faces(points, filename):
18     faces = []
19     with open(filename, 'r') as file:
20         for line in file:
21             if line.startswith('f '):
22                 face = [int(x) for x in line.strip().split()[1:]]
23                 faces.append(face)
24     return faces
25
26 # Escreve os vertices e as faces no arquivo OBJ
27 def write_obj(vertices, faces, filename, image_path):
28     with open(filename, 'w') as f:
29         f.write(f"# {image_path}\n")
30         for vertex in vertices:
31             f.write(f"v {vertex[0]} {vertex[1]} {vertex[2]}\n")
32         for face in faces:
33             f.write(f"f {' '.join([str(idx+1) for idx in face])}\n")

```

O conjunto de vértices que compõem uma malha pode ser conectado de diferentes formas, resultando em diferentes topologias para o mesmo rosto. O MediaPipe, por exemplo, utiliza sempre as mesmas conexões entre os vértices, garantindo uma topologia consistente. No entanto, é importante destacar que diferentes topologias produzem resultados distintos, mesmo que o conjunto de vértices permaneça inalterado, pois a forma como os vértices se conectam altera as propriedades da malha resultante.

## 3.2 Obtenção das características robustas

São duas as principais características robustas utilizadas na reconstrução: curvas parabólicas e *ridges* (BRUCE; GIBLIN; TARI, 1996). O código fonte contendo o cálculo para obtenção dessas curvas foi desenvolvido em MATLAB, como parte do doutorado de Danilo Marques.



O dado de entrada para o programa é o arquivo OBJ contendo os pontos da face. Como resultado, tem-se os pontos âncora, sejam eles curvas parabólicas ou *ridges*, que servirão de base para a reconstrução.

As seções seguintes mostram como as características robustas foram extraídas.

### 3.2.1 Curvas parabólicas

A definição de curvas parabólicas está diretamente relacionada à curvatura gaussiana  $K$  de uma superfície, que pode ser expressa em termos dos coeficientes da primeira e da segunda forma fundamental. A curvatura gaussiana  $K$  é dada por:

$$K = \frac{eg - f^2}{EG - F^2}$$

onde  $e$ ,  $f$  e  $g$  são os coeficientes da segunda forma fundamental, e  $E$ ,  $F$  e  $G$  são os coeficientes da primeira forma fundamental. As curvas parabólicas são definidas como o conjunto de pontos de uma superfície onde a curvatura gaussiana  $K$  é igual a zero, ou seja:

$$P = \{p \in S : K(p) = 0\}$$

Substituindo a expressão da curvatura gaussiana, temos que as curvas parabólicas ocorrem nos pontos onde:

$$eg - f^2 = 0$$

Isso significa que, nos pontos parabólicos, a relação entre os coeficientes da segunda forma fundamental resulta em uma curvatura gaussiana nula. As curvas parabólicas, portanto, representam os locais na superfície onde a curvatura muda de sinal, ou seja, onde a superfície transita entre uma curvatura positiva (elíptica) e uma curvatura negativa (hiperbólica).

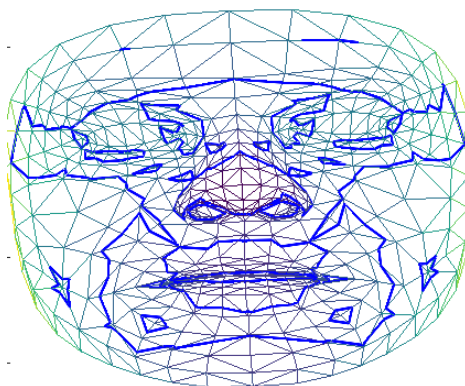


Figura 2 – Curvas parabólicas.

Essas curvas são importantes na análise geométrica de superfícies, pois indicam transições significativas na forma da superfície. Essas zonas de transição são preservadas quando a superfície se deforma de maneira suave (sem dobras ou rupturas). Portanto, as curvas parabólicas se tornam úteis para a reconstrução da superfície, visto que se configuram como uma característica robusta.

### 3.2.2 Curvas *ridges*

Seja  $S$  uma superfície parametrizada por  $\phi : U \rightarrow \mathbb{R}^3$ . Considere  $k_1$  e  $k_2$  as curvaturas principais, com  $k_1 > k_2$ , e  $d_1$  e  $d_2$  as direções principais correspondentes. Os gradientes das curvaturas principais são  $\nabla k_1$  e  $\nabla k_2$ .

- Um ponto é um **ridge azul** se  $\langle \nabla k_1, d_1 \rangle = 0$ .
- Um ponto é um **ridge vermelho** se  $\langle \nabla k_2, d_2 \rangle = 0$ .

O produto interno (ou produto escalar) ser zero significa que o gradiente da curvatura principal é ortogonal à direção principal, caracterizando um ponto onde a curvatura é extrema ao longo dessa direção.

Ou seja, os pontos *ridges* são aqueles onde a curvatura principal é extrema ao longo de sua própria linha de curvatura. O conjunto de todos esses pontos forma a curva *ridge* ou simplesmente *ridge*.

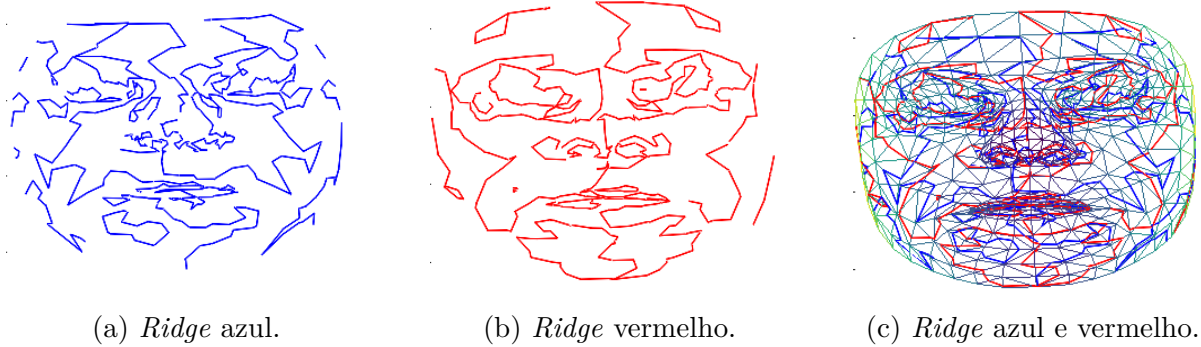


Figura 3 – Conjunto de imagens das *ridges* azul, vermelha e a combinação de ambas.

Ao obter as *ridges*, nota-se que elas compreendem muito mais pontos que as curvas parabólicas. Por isso, vale ressaltar que o código em MATLAB requer muito mais tempo para calcular as *ridges* em relação às curvas parabólicas. Grande parte desse atraso se deve à ausência de uma estrutura de dados apropriada para percorrer todos os pontos e calcular os dados necessários (curvatura gaussiana, curvatura média, etc). Uma implementação mais eficiente para calcular alguns desses elementos foi desenvolvida em C++ por Angarola (2024), cujo trabalho também faz parte deste projeto temático.

## 3.3 Criação da nova malha

Antes de aplicar o algoritmo de reconstrução, é necessário atualizar a malha original de modo a incluir os vértices resultantes das interseções com as curvas. A nova malha possui uma matriz de adjacência atualizada, que contém informações dos vértices vizinhos para os pontos âncora na reconstrução da superfície. Dessa forma, inserimos os pontos da

curva na malha original e recalculamos as conexões entre cada vértice, o que gera novas faces na malha. Como os pontos das curvas intersectam necessariamente as arestas da malha original (formada inicialmente por faces triangulares), temos os seguintes cenários de interseção:

- **0 interseções:** Nenhum ponto da curva intersecta a face da malha original. Nesse caso, a face permanece inalterada.
- **1 interseção:** Apenas um ponto da curva intersecta uma das arestas da face. O triângulo original é dividido em dois novos triângulos, com o ponto de interseção sendo conectado ao vértice oposto à aresta de interseção.
- **2 interseções:** Dois pontos intersectam a face. Nesse caso, ligamos os dois pontos e criamos um triângulo e um quadrilátero. Esse é o caso mais recorrente.
- **3 interseções:** Três pontos intersectam a face, cada um em uma aresta diferente. A face é dividida em quatro novos triângulos que conectam os pontos de interseção aos vértices originais da face.
- **4 ou mais interseções:** Embora raro, caso haja quatro ou mais interseções em uma única face, o triângulo original é mantido para simplificação. Por outro lado, o refinamento posterior é recomendado para tratamento adequado.

A figura abaixo ilustra os diferentes casos de interseções que podem ser encontrados.

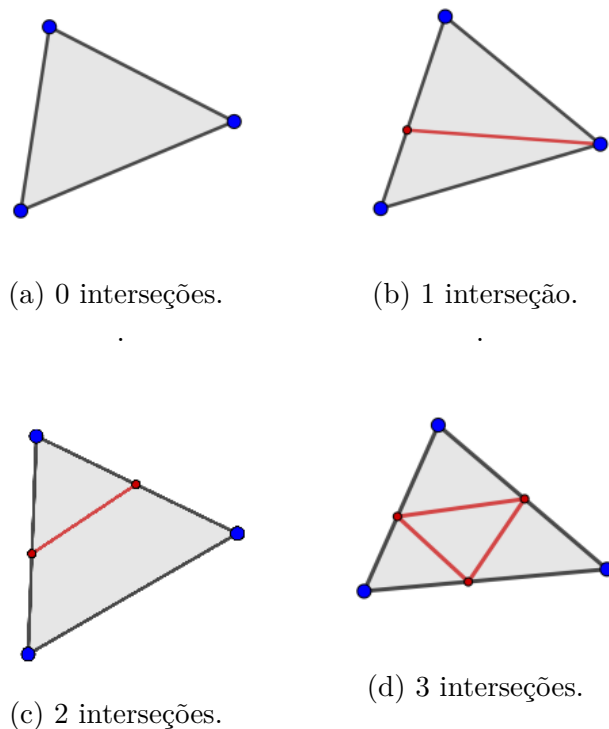


Figura 4 – Cenários de interseção da curva com a malha.

Esses cenários asseguram a subdivisão apropriada das faces triangulares. Para uma malha mais refinada, é necessário estender o tratamento para a divisão de outras formas poligonais, como quadriláteros e pentágonos.

Com a malha atualizada, é possível extrair a nova matriz de adjacência, que captura as relações entre os vértices da curva e os da malha. Esta matriz é essencial para a reconstrução da superfície, pois proporciona as informações necessárias para o algoritmo de reconstrução, o que garante uma melhor integração dos vértices da curva com a nova malha.

## 3.4 Reconstrução da superfície

Assumindo que os arquivos OBJ com os vértices das curvas parabólicas ou *ridges* já estejam calculados, o algoritmo pode ser resumido nas seguintes etapas:

- Seleciona-se a quantidade desejada de pontos âncora;
- Calcula-se a matriz de adjacência
- Geram-se os pesos de cada vértice (padrão para todos os vértices);
- Calculam-se as coordenadas diferenciais;
- Gera-se a matriz Laplaciana;
- Recupera-se as coordenadas cartesianas a partir das coordenadas diferenciais e obtêm-se a malha reconstruída.

Para organizar a ordem de execução de cada etapa do projeto, foi criado um repositório no GitHub<sup>1</sup>. Este repositório contém instruções de como executar o código, bem como fornece toda a base teórica (em inglês) de forma mais resumida.

### 3.4.1 Escolha dos pontos âncora

Para a etapa inicial da reconstrução, é necessário determinar a porcentagem de pontos âncora que serão utilizados. Foi considerado apenas um método para a seleção desses pontos: a função `linspace`, disponível na biblioteca NumPy. Essa função permite selecionar uma porcentagem de pontos de maneira que eles sejam linearmente espaçados com base em seus índices. A escolha de uma porcentagem de pontos âncora é fundamental para avaliar a qualidade da reconstrução conforme a quantidade de dados disponíveis diminui, possibilitando uma análise mais abrangente da robustez do processo de reconstrução.

<sup>1</sup> Disponível em <https://github.com/jpviguini/3D-mesh-reconstruction>. Acesso em 10 de dezembro de 2024.

### 3.5 Texturização da malha

Para avaliar melhor os resultados, realizamos a texturização da malha, mapeando as cores reais do rosto humano capturadas na imagem inicial para os vértices da malha 3D. Esse processo consiste em associar cada vértice da malha a um pixel correspondente na imagem original, utilizando as coordenadas normalizadas dos vértices para determinar as posições na textura. A partir dessas cores atribuídas aos vértices, as faces da malha são visualizadas com tons interpolados, proporcionando um aspecto realista.

A renderização foi realizada utilizando Mayavi, uma biblioteca Python para visualização 3D. Aplicamos também *Phong Shading* para melhorar a qualidade visual. O *Phong Shading* é uma técnica que utiliza interpolação de iluminação, fornecendo suavidade às superfícies e realçando os detalhes da textura e iluminação.

Dessa forma, a visualização final apresenta uma representação tridimensional da malha texturizada, com as cores alinhadas à imagem original e iluminação que realça os detalhes faciais.

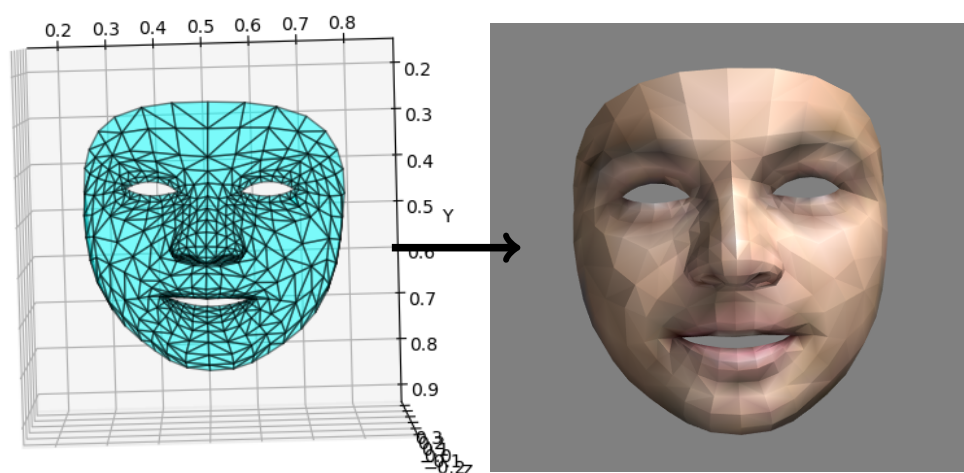


Figura 5 – Malha original texturizada

### 3.6 Resultados obtidos

A seguir serão apresentados os resultados da reconstrução da malha original na medida em que variamos a quantidade de pontos selecionados e o tipo de curva utilizada. Os testes foram realizados utilizando 100%, 50% e 20% dos pontos âncora.

### 3.6.1 Curvas parabólicas



(a) Malha original



(b) 20% dos pontos. Erro: 0.798.



(c) 50% dos pontos. Erro: 0.633.



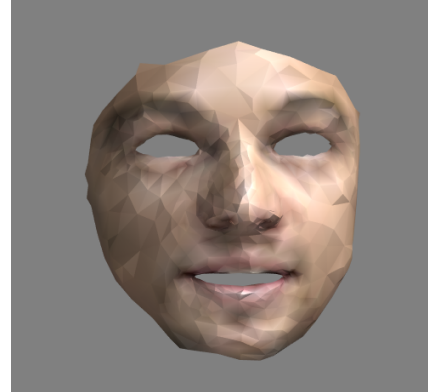
(d) 100% dos pontos. Erro: 0.515

Figura 6 – Reconstrução a partir de curvas parabólicas

### 3.6.2 Curvas *ridges*



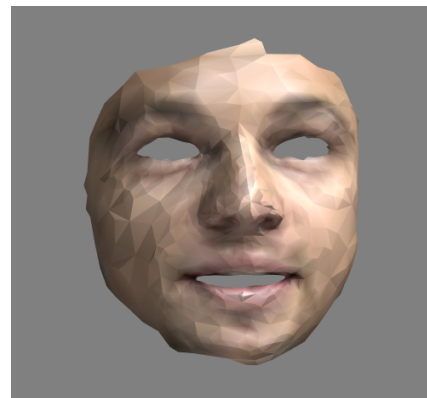
(a) Malha original



(b) 100% dos pontos (*ridge* azul).  
Erro: 0.510



(c) 100% dos pontos (*ridge* vermelha). Erro: 0.503



(d) 100% dos pontos (*ridges* azul e vermelha). Erro: 0.487

Figura 7 – Reconstrução a partir de *ridges*

## 4 Considerações finais

O objetivo deste relatório final foi apresentar as atividades realizadas pelo bolsista no período entre janeiro e dezembro de 2024, que abrangeram desde a extração de características robustas de rostos humanos até a aplicação do algoritmo de reconstrução. O andamento das atividades foi ligeiramente alterado em relação ao cronograma inicial, principalmente em razão da criação da nova malha antes de efetuar a reconstrução. Na primeira metade do projeto essa etapa não estava sendo considerada, o que nos levava a resultados incorretos, visto que a matriz de adjacência não era calculada considerando a interseção dos vértices da curva com a malha. A proposta de criar um *notebook* usando o Google Colab para organizar os algoritmos (mencionada no relatório parcial) foi abandonada



em razão da compatibilidade entre a ferramenta de visualização Mayavi e o Google Colab. Portanto, mantivemos apenas um repositório com instruções de como utilizar o código no Github.

Os resultados quantitativos e qualitativos sugerem que a utilização de curvas *ridges* proporciona uma maior fidelidade à malha original, especialmente em cenários com menos pontos âncora, em comparação às curvas parabólicas. Esses achados reforçam a importância de explorar diferentes características robustas na etapa de reconstrução.

O projeto abriu possibilidades para melhorias futuras, como a otimização do cálculo de características robustas e o desenvolvimento de estratégias para minimizar o erro visual na reconstrução conforme o número de pontos disponíveis diminui. Além disso, há de melhorar a forma de seleção de pontos âncora. Atualmente os pontos selecionados estão linearmente espaçados por índice. Ao escolher uma porcentagem de pontos, cada porção do rosto não está sendo igualmente considerada (por exemplo, não estamos selecionando 50% do nariz, 50% da boca, etc). Portanto, esse refinamento resultaria em uma seleção de pontos mais consistente. As melhorias aqui citadas serão fundamentais para consolidar a metodologia proposta e ampliar seu potencial de aplicação em outras áreas, como reconhecimento facial e compressão de imagem.

## 4.1 Participação em eventos científicos

Em outubro de 2024, este projeto foi apresentado no Simpósio Internacional de Iniciação Científica da USP (SIICUSP), realizado no Instituto de Ciências Matemáticas e de Computação (ICMC-USP). A apresentação incluiu um resumo detalhado do trabalho, abordando a motivação, a metodologia empregada e os resultados parciais alcançados até aquele momento. Essas informações foram organizadas em um pôster, utilizado como base para a exposição oral durante o evento.

O trabalho foi avaliado por dois especialistas, que analisaram tanto o conteúdo técnico do projeto quanto a qualidade da apresentação. Ao final, o feedback recebido foi altamente positivo, destacando pontos fortes da pesquisa e da comunicação, o que reforçou a relevância e o impacto do estudo.

# Referências

- ANGAROLA, M. P. *Curvature Estimation Using Machine Learning Algorithms*. 2024. Acessado em: 19-08-2024. Disponível em: <<https://github.com/MatheusPaivaa/CurvatureML>>.
- BRUCE, J. W.; GIBLIN, P. J.; TARI, F. Ridges, crests and sub-parabolic lines of evolving surfaces. *International Journal of Computer Vision*, 1996.
- FAKHOURY, A. *Image curves reconstruction by means of robust features*. 2021. Acessado em: 18-08-2024. Disponível em: <<https://github.com/andrefakhoury/image-curve-reconstruction>>.
- IZUMIYA, S. et al. *Differential Geometry From A Singularity Theory Viewpoint*. Singapura: World Scientific, 2015. 139 p.
- LEE, I.-K. Curve reconstruction from unorganized points. *Computer Aided Geometric Design*, Elsevier BV, v. 17, n. 2, p. 161–177, fev. 2000. Disponível em: <[https://doi.org/10.1016/S0167-8396\(99\)00044-8](https://doi.org/10.1016/S0167-8396(99)00044-8)>.
- LUGARESI, C. et al. Mediapipe: A framework for perceiving and processing reality. In: *Third Workshop on Computer Vision for AR/VR at IEEE Computer Vision and Pattern Recognition (CVPR) 2019*. [s.n.], 2019. Disponível em: <[https://mixedreality.cs.cornell.edu/s/NewTitle\\_May1\\_MediaPipe\\_CVPR\\_CV4ARVR\\_Workshop\\_2019.pdf](https://mixedreality.cs.cornell.edu/s/NewTitle_May1_MediaPipe_CVPR_CV4ARVR_Workshop_2019.pdf)>.
- SCHOENBERG, I. J. Contributions to the problem of approximation of equidistant data by analytic functions. part a. on the problem of smoothing or graduation. a first class of analytic approximation formulae. *Quarterly of Applied Mathematics*, American Mathematical Society (AMS), v. 4, n. 1, p. 45–99, abr. 1946. Disponível em: <<https://doi.org/10.1090/qam/15914>>.
- SORKINE, O. Differential representations for mesh processing. *Computer Graphics Forum*, European Association for Computer Graphics, v. 25, n. 4, p. 789–807, 2006.
- SORKINE, O. et al. Geometry-aware bases for shape approximation. *IEEE Transactions on Visualization and Computer Graphics*, Institute of Electrical and Electronics Engineers (IEEE), v. 11, n. 2, p. 171–180, mar. 2005. Disponível em: <<https://doi.org/10.1109/tvcg.2005.33>>.
- YANG, S. et al. Wider face: A face detection benchmark. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2016.