

### 3

## Reconstrução de Malha por Minimização

Neste capítulo apresentaremos a formalização do problema de reconstrução da geometria de malhas seguindo a proposta de Sorkine (16) e o método computacional que usamos para resolvê-lo.

### 3.1

#### Descrição do Problema

Este trabalho propõe a reconstrução de uma malha a partir da sua conectividade e da geometria de alguns pontos. Dada uma malha com conectividade arbitrária e um subconjunto de pontos selecionados como de controle, construímos e resolvemos um sistema linear esparso representando a forma normal da minimização do Laplaciano. A solução do sistema linear corresponde a geometria dos vértices da malha, buscando uma distribuição suave e uniforme sobre a superfície.

A maioria dos métodos que representam uma superfície geométrica usa amplamente malhas poligonais baseada na geometria e na conectividade da malha. A primeira determina a localização de cada vértice no plano ou no espaço e a segunda determina como os vértices estão conectados formando os polígonos que descrevem a superfície. É natural imaginar que estas duas componentes da malha são independentes, isto é, muitas malhas distintas poderiam co-existir com uma geometria dada. Embora isso seja teoricamente verdade, várias aplicações para representação de curvas ou superfícies nos levam a acreditar que existe uma correlação entre as duas, principalmente se levarmos em consideração que boas triangulações fazem com que a geometria da malha seja suave e uniforme.

Uma questão interessante nesta tese é o fato de trabalharmos com uma malha, onde somente um subconjunto de vértices contém informações geométricas. Assim como Isenburg et al (8) e Sorkine et al (16), queremos mostrar que uma malha tem alguma forma natural, ou seja, mostramos que uma malha simples contém algumas informações geométricas não-triviais.

A solução usada neste trabalho é baseada na malha de mínimos quadrados (16). Propomos uma representação eficiente dos sistemas lineares en-

volvidos na minimização. Essa representação está embutida na malha, como descrito na seção 3.2. A minimização propriamente dita é feita pelo método do gradiente conjugado e foi implementada nesta mesma representação.

### 3.2

#### Formulação do Problema

Sejam  $M = (V, E, F)$  uma malha triangular com conectividade arbitrária,  $L$  a Matriz Laplaciana e  $L_S$  a Matriz Laplaciana Simétrica de  $M$ , tal que  $L$  e  $L_S$  são matrizes de ordem  $n \times n$  e  $n$  é o número de vértices da malha. Sejam  $x, y, z$  vetores de dimensão  $n$ . A regularidade e suavidade de  $M$  pode ser descrita como:

$L \cdot x = 0, L \cdot y = 0, L \cdot z = 0$ , para Matriz Laplaciana. Outra formulação é:

$L_S \cdot x = 0, L_S \cdot y = 0, L_S \cdot z = 0$ , para Matriz Laplaciana Simétrica.

Os vetores  $x, y, z$  são as soluções do sistemas, ou seja, armazenam as coordenadas dos  $n$  vértices da malha reconstruída.

A Figura 3.1 ilustra um exemplo de uma malha no plano. A Figura 3.2: ilustra as Matrizes: Laplaciana (a) e Laplaciana Simétrica (b).

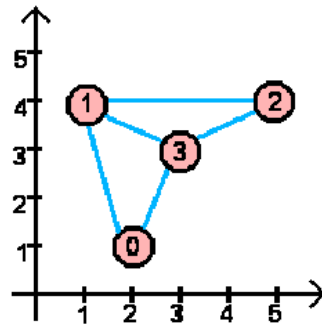


Figura 3.1: Malha no plano com 4 vértices

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

3.2(a): Matriz Laplaciana

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

3.2(b): Matriz Laplaciana Simétrica

Figura 3.2: Representação matricial da conectividade da malha ilustrada na Figura 3.1

Selecionando apenas  $m$  pontos como informação geométrica dada (pontos de controle) uma solução não-trivial pode ser encontrada.

Seja  $C = \{s_1, \dots, s_m\}$  o conjunto dos índices dos vértices selecionados como de controle, as equações restringindo as coordenadas dos pontos de

controle são adicionadas ao sistema linear:

$$F_{ij} = \begin{cases} 1 & \text{Se } j = s_i \in C \\ 0 & \text{nos outros casos} \end{cases}$$

O sistema linear se torna então retangular  $((n + m) \times n)$ :

$$A x = b, \text{ onde}$$

$$A = (L) F \text{ ou } A = (L)_S F, e$$

$$b_k = \begin{cases} 0 & \text{Se } k \leq n \\ x_{s_{k-n}} & \text{Se } n < k \leq n + m \end{cases}$$

A Figura 3.3 mostra o mesmo exemplo de malha ilustrado anteriormente, mas agora o vértice 3 de coordenada (3, 3) é selecionado como de controle implicando na modificação da construção do sistema linear ilustrado na Figura 3.4. A Figura 3.5 ilustra os vetores  $b_x$  e  $b_y$ .

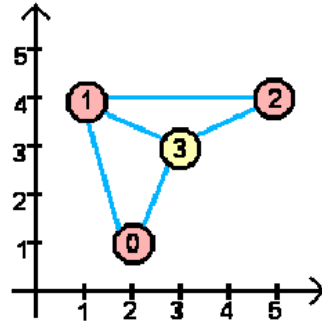


Figura 3.3: Malha no plano com 4 vértices. O vértice 3 de coordenada (3, 3) foi selecionado como ponto de controle

$$A = \begin{bmatrix} 1 & -\frac{1}{2} & 0 & -\frac{1}{2} \\ -\frac{1}{3} & 1 & -\frac{1}{3} & -\frac{1}{3} \\ 0 & -\frac{1}{2} & 1 & -\frac{1}{2} \\ -\frac{1}{3} & -\frac{1}{3} & -\frac{1}{3} & 1 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix}$$

3.4(a): Matriz Laplaciana

$$A_S = \begin{bmatrix} 2 & -1 & 0 & -1 \\ -1 & 3 & -1 & -1 \\ 0 & -1 & 2 & -1 \\ -1 & -1 & -1 & 3 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix}$$

3.4(b): Matriz Laplaciana Simétrica

Figura 3.4: Representação matricial da conectividade da malha ilustrada na Figura 3.3.

Observe que o exemplo de malha ilustrado não mostra necessariamente uma matriz esparsa, mas considerando que as malhas utilizadas neste trabalho possuem no mínimo cinquenta vértices, e as valências destes são em média iguais a seis, faz sentido dizer que a matriz A é uma matriz esparsa.

$$b_x = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 3 \end{bmatrix} \quad b_y = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 3 \end{bmatrix}$$

Figura 3.5: Vetores  $b_x$  e  $b_y$ . Note que todos os pontos que não são de controle estão na origem

### 3.2.1

#### Representação das Matrizes

Como foi visto anteriormente a escolha da estrutura de dados que armazena a matriz esparsa é crucial para eficiência do trabalho. Assim, somente os elementos da matriz esparsa diferentes de zero serão armazenados.

A representação da Matriz Laplaciana e da Matriz Laplaciana Simétrica, é baseada em três listas de vetores de números inteiros ( $val$ ,  $adj$  e  $cp$ ). O vetor  $val$  armazena a valência de cada vértice, portanto sua dimensão é igual ao número de vértices existentes na malha. O vetor  $adj$  armazena a vizinhança de cada vértice. O vetor  $cp$  armazena os índices dos vértices selecionados como de controle. A Figura 3.6 exemplifica os vetores  $val$ ,  $adj$  e  $cp$  para a matriz na Figura 3.3.

$$\begin{aligned} val &= [2; 3; 2; 3; ] \\ adj &= [1; 3; 0; 2; 3; 1; 3; 0; 1; 2; ] \\ cp &= [3] \end{aligned}$$

Figura 3.6: Representação da Matriz A da Figura 3.3 através de três listas de vetores  $val$ ,  $adj$  e  $cp$ .

### 3.2.2

#### Solução do Sistema Linear Esperso

Queremos resolver um sistema linear esperso por mínimos quadrados, ou seja, minimizar  $\|Ax - b\|$ , onde a matriz  $A$  é uma matriz retangular  $((n + m) \times n)$ , o que nos leva a uma aplicação de um método iterativo ao sistema de equações normais  $A^T A x = A^T b$ . Para evitar problemas quanto a

formação da matriz  $A^T A$ , a forma fatorada das equações normais  $A^T(b - Ax) = 0$  foi utilizada.

O método para solução do sistema linear utilizado foi o método do Gradiente Conjugado (ver seção 2.2.3), formulado para solucionar a forma fatorada das equações normais. Uma vez que os pontos de controle são definidos, três sistemas lineares são construídos, um para  $x$ , outro para  $y$ , e outro para  $z$ , e o algoritmo de solução do sistema pelo método do gradiente conjugado é então acionado três vezes. Optou-se por trabalhar com  $x$ ,  $y$  e  $z$  separadamente, pois não há garantias de que a solução do sistema convirja igualmente para cada coordenada.

---

**Algorithm 1** Algoritmo do gradiente conjugado

---

Sejam  $x^{(0)}$  uma aproximação inicial,  $r^{(0)} = b - Ax^{(0)}$ ,  $p^{(0)} = s^{(0)} = A^T r^{(0)}$ ,  $\gamma_0 = \|s^{(0)}\|_2^2$  e  $k$  um número inteiro.

Para  $k = 0, 1, 2, \dots$

  enquanto  $\gamma_k > \text{tolerância}$

$$q^{(k)} = Ap^{(k)};$$

$$\alpha_k = y_k / \|q^{(k)}\|_2^2;$$

$$x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)};$$

$$r^{(k+1)} = r^{(k)} - \alpha_k q^{(k)};$$

$$s^{(k+1)} = A^T r^{(k)};$$

$$\gamma_{k+1} = \|s^{(k+1)}\|_2^2;$$

$$\beta_k = \gamma_{k+1} / \gamma_k;$$

$$p^{(k+1)} = s^{(k+1)} + \beta_k p^{(k)}.$$

  fim-enquanto

fim-para

---

Este método requer o armazenamento de dois vetores de tamanho  $n$ :  $x$  e  $p$ ; e dois vetores de tamanho  $m$ :  $r$  e  $q$ . Cada iteração requer cerca de  $2nz(A) + 3n + 2m$  flops (floating points operations), onde  $nz(A)$  é o número de elementos diferentes de zero em  $A$ .

Por questões convenientes a aproximação inicial é sempre zero, ou seja,  $x^{(0)} = 0$ , e por consequência  $r^{(0)} = b$ . A tolerância adotada foi de 0,0000001.

Uma outra questão importante no nosso método de reconstrução é que os pontos de controle não são interpolados e sim aproximados, e com isso mantemos as condições de suavidade e de uniformidade da malha

Esta condição de suavidade pode ser alterada, fazendo com que os pontos de controle fiquem estáticos, sendo esta mais uma vantagem de representar  $A$  diretamente na estrutura de dados. A cada iteração os pontos de controle são ajustados para sua posição original, isto implica no aumento do número de passos e no tempo de processamento.

A implementação eficiente de um método iterativo depende substancialmente do desempenho do produto entre a matriz esparsa e um vetor ( $Av$  e  $A^T u$ ). Neste trabalho em especial a diferença em termos de implementação da Matriz Laplaciana da Matriz Laplaciana Simétrica está estritamente ligada a este produto. Os algoritmos destas operações estão apresentados abaixo.

### Matriz Laplaciana

O produto entre a matriz Laplaciana  $A$  e um vetor  $v$  ( $u = Av$ ) é então representado pelo seguinte algoritmo:

Sejam  $n$  o número de vértices existentes na malha,  $j$  o número de pontos de controle,  $val$ ,  $adj$ ,  $cp$  os vetores que armazenam a valência de cada vértice, a vizinhança de cada vértice e os índices dos pontos de controle respectivamente e  $k = 0$ :

---

**Algorithm 2** Produto da Matriz  $A$  (Laplaciana) por um vetor  $v$ :

---

```

para i = 0 até n
    u[i] = v[i]
    d = -1 / val[i];
    para j = 0 até val[i]
        u[i] = u[i] + (d * v[ adj[k] ] );
        k = k+1;
    fim-para
fim-para
para i = 0 até j
    u[ i + n ] = v[ ctrl[i] ] ;
fim-para

```

---

O produto entre a matriz transposta de  $A$  e o vetor  $u$  ( $v = A^T u$ ) também possui um algoritmo muito simples e necessita dos mesmos elementos.

---

**Algorithm 3** Produto da Matriz transposta de  $A$  (Laplaciana) por um vetor  $u$ :

---

```

para i = 0 até n
    v[i] = u[i];
    para j = 0 até val[i]
        d = -1 / val[ adj[k] ];
        v[i] = v[i] + (u[ adj[k] ] * d );
        k = k+1;
    fim-para
fim-para
para i = 0 até j
    v[cp[i] ] = v[cp[i]] + (u[i+n]) ;
fim-para

```

---

Os cálculos são simples e as informações da matriz são acessadas de maneira ordenada através da correlação dos vetores que a representam. Note que o próprio vetor *val* indica quanto deve ser percorrido do vetor *adj*.

### Matriz Laplaciana Simétrica

A simetria da matriz faz com que o produto da matriz por um vetor seja muito parecido com o produto da transposta da matriz, mudando somente no que se refere aos pontos de controle.

---

**Algorithm 4** Produto da matriz  $A$  (Laplaciana Simétrica) por um vetor  $v$ :

---

```

para i = 0 até n
    u[i] = val[i] * v[i];
    para j = 0 até val[i]
        u[i] = u[i] - ( v[ adj[k] ] );
        k = k+1;
    fim-para
fim-para
para i = 0 até j
    u[ i + n ] = v[ ctrl[i] ] ;
fim-para

```

---



---

**Algorithm 5** Produto da matriz transposta de  $A$  (Laplaciana Simétrica) por um vetor  $u$ :

---

```

para i = 0 até n
    u[i] = val[i] * v[i];
    para j = 0 até val[i]
        u[i] = u[i] - ( v[ adj[k] ] );
        k = k+1;
    fim-para
fim-para
para i = 0 até j
    v[cp[i] ] = v[cp[i]] + (u[i+n]) ;
fim-para

```

---

Embora as matrizes Laplaciana e Laplaciana Simétrica sejam muito parecidas em seus processos de formação, a grande diferença destes métodos é relacionada aos seus resultados e a velocidade de convergência, que serão expostos no capítulo destinado aos resultados.

### 3.2.3

#### Pontos de Controle

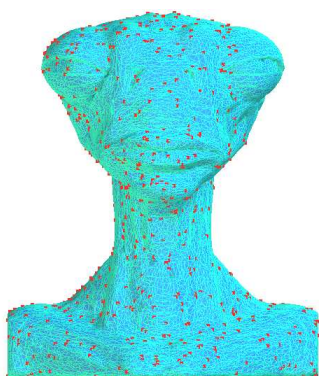
Para obtermos uma aproximação da superfície é necessário que os pontos de controle sejam determinados a partir de um critério com a finalidade

de trazer a malha reconstruída o mais próximo possível da malha original e minimizar o erro geométrico. Intuitivamente, os pontos de controle devem ser lugares estratégicos na superfície, como pontos significativos, nas extremidades e em lugares onde o detalhe geométrico é maior. Em virtudes destas considerações três critérios de seleção foram testados: Seleção Aleatória; Seleção por Intervalos e Seleção por Curvatura. Pela falta de base teórica para descrever a relação malha / geometria, esses critérios são apenas heurísticas. Por outro lado, resultados empíricos poderão contribuir para o melhor entendimento dessa relação.

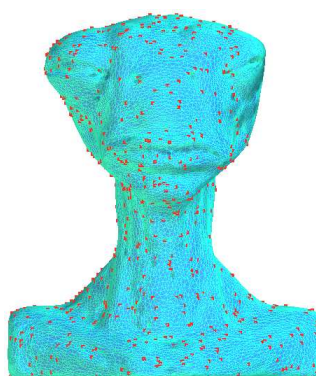
Inicialmente um percentual do número de vértices foi definido para determinar a dimensão do vetor  $cp$ , no entanto dependendo da necessidade este percentual pode ser alterado.

### Seleção Aleatória

A seleção aleatória consiste em sortear números inteiros entre zero e o número de vértices da malha. Este é um método de escolha rápido. No entanto se a malha for extremamente irregular há chances de que regiões com grande significância fiquem sem informação geométrica. Neste caso, esse método pode se tornar ineficiente (Figura 3.7).



3.7(a): Modelo Original



3.7(b): Modelo reconstruído

Figura 3.7: Modelo Alien com 19198 vértices com 5% dos pontos selecionados como de controle (em destaque). Note que a região do olho esquerdo do Alien quase não possui pontos para representá-la, este fato nos leva a um erro geométrico local maior.

### Seleção por Intervalos

A seleção por intervalos consiste em dividir o número de vértices da malha em intervalos e selecionar um percentual previamente determinado de pontos dentro de cada intervalo.

Para malhas com um número de vértices significativamente elevado a seleção por intervalos é mais rápida que a seleção aleatória. Caso a malha



possua os vértices em ordem aleatória, o que é raro na prática devido aos métodos de geração e compressão, dificilmente serão encontradas regiões sem algum ponto de controle que as represente (Figura 3.8), caso contrário os pontos de controle formarão faixas sobre a malha o que significa que determinadas regiões estarão bem representadas e outras não (Figura 3.9). Levando em consideração que há situações onde uma determinada região necessita de um número expressivo de pontos de controle esta questão seria um fator positivo, todavia este método não é capaz de medir qual região possui esta necessidade, podendo ocasionar um erro maior que o da seleção aleatória. Portanto, é um método rápido que funciona bem para muitos pontos de controle.

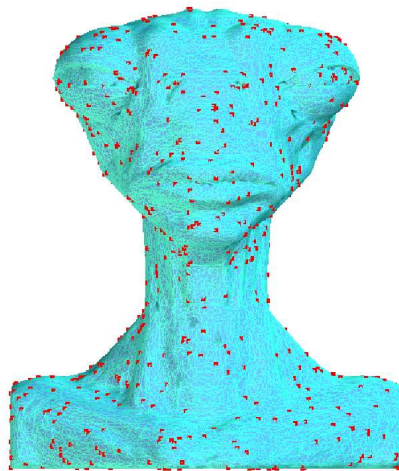


Figura 3.8: Note que os pontos ficam bem distribuídos, porém regiões como a boca e os olhos precisam de uma maior amostragem, pois possuem detalhes geométricos mais relevantes.

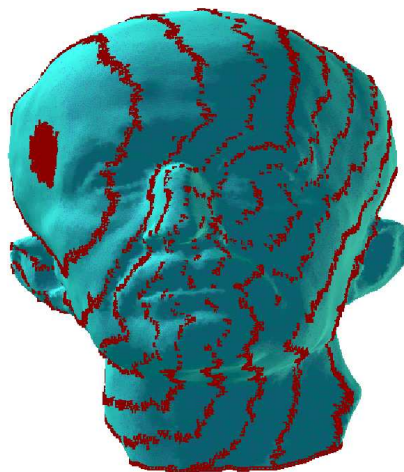


Figura 3.9: A região da orelha esquerda do modelo não está representada, o que acarreta na má aproximação da malha.

### Seleção por Curvatura

A seleção por curvatura também possui uma idéia muito simples, calcula-se primeiramente a curvatura gaussiana em cada vértice da malha, em seguida seleciona-se como pontos de controle aqueles que possuem maior curvatura em módulo. De maneira geral este método requer um tempo maior que os apresentados anteriormente já que primeiro é necessário computar a curvatura de cada vértice e em seguida ordená-los para enfim selecioná-los. Embora este método seja muito eficiente em encontrar pontos com informações geométricas significativas, dependendo da malha os pontos não ficam bem distribuídos (Figura 3.10). A curvatura média também foi estudada, no entanto este método requer um tempo de processamento maior e seleciona na média a mesma região que o método da curvatura gaussiana (Figura 3.11).

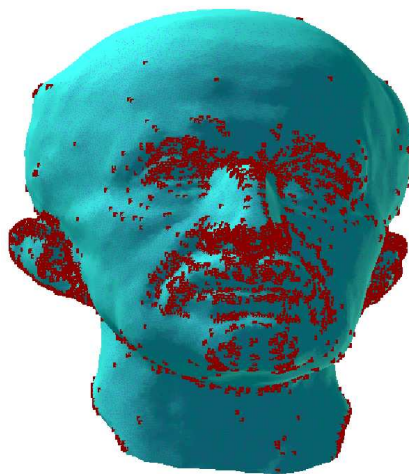


Figura 3.10: Modelo Max com 8% dos pontos selecionados como de controle. Note que os pontos ficam concentrados nas regiões das orelhas, nariz e boca já que estes possuem maior significância geométrica, ou seja, nos pontos de maior curvatura.

Para evitar o problema da concentração dos pontos de controle dois casos foram estudados, que serão apresentado a seguir.

**Caso 1:** O primeiro caso faz uma seleção dos pontos de controle a partir dos pontos de maior curvatura. Respeitando as seguintes restrições: quando o número de pontos de controle está entre dez e vinte e cinco por cento do número de vértices um ponto de controle fica distante pelo menos uma vizinhanças de outro ponto de controle (Figura 3.12 (b)); quando o número de pontos de controle é menor que dez por cento do número de vértices um ponto de controle fica distante pelo menos duas vizinhanças de outro ponto de controle (Figura 3.12 (a)).

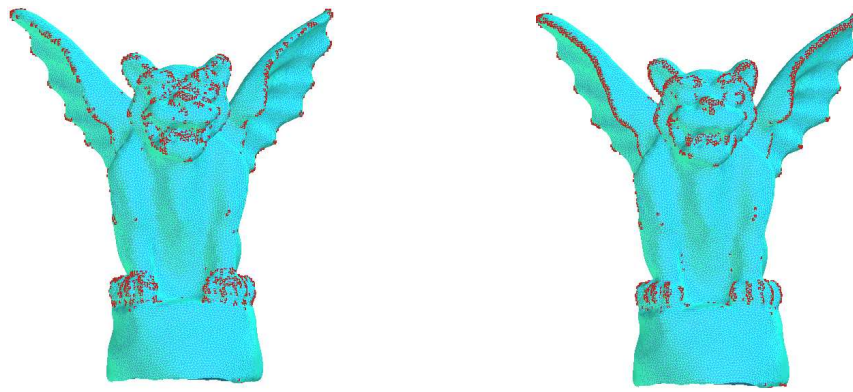
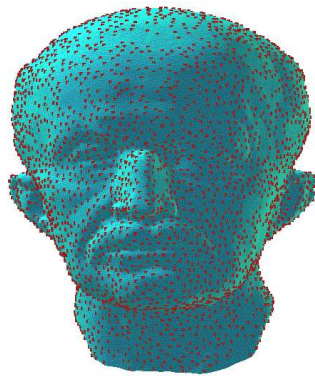


Figura 3.11: Modelo Gargoyle com 5% dos pontos selecionados como pontos de controle. No modelo da esquerda os pontos foram selecionados através do cálculo da curvatura gaussiana e no modelo da direita foram selecionados através do cálculo da curvatura média.



3.12(a): 8% dos pontos selecionados como de controle



3.12(b): 20% dos pontos selecionados como de controle

Figura 3.12: Modelo Max, onde (a) ilustra 8% dos pontos selecionados como de controle com a restrição de duas vizinhanças e (b) ilustra o modelo com 20% dos pontos selecionados como de controle com a restrição de uma vizinhança.

Quando o percentual de pontos de controle é realmente pequeno, 2% por exemplo, este método pode não ser tão bom, pois como a distribuição dos pontos de controle é feita a partir dos pontos de maior curvatura, regiões onde a curvatura é igual a zero ou próxima de zero ficam sem representantes (Figura 3.13).

**Caso 2:** O segundo caso consiste em selecionar os pontos de controle por amostragem em relação a uma distribuição de curvatura, de forma que grande parte do percentual dos pontos de controles seja de alta curvatura, e os demais pontos de controle possuam curvatura média e baixa. Este método de amostragem é conhecido como Importance Sampling (4).

Para realizarmos a amostragem foi necessária a construção de um vetor

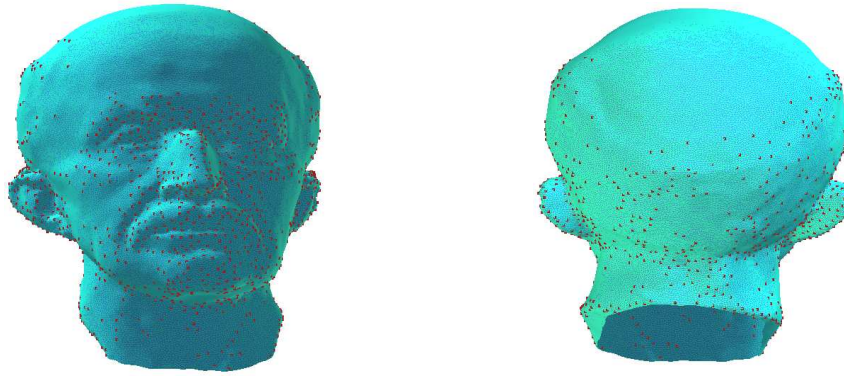


Figura 3.13: Modelo Max com 2% dos pontos selecionados como de controle. Note que há uma região grande sem pontos de controle para representá-la.

u de dimensão  $n$  tal que:

$$u_j = \frac{\sum_{i=0}^j \text{curvatura de } v_i}{\sum_{i=0}^n \text{curvatura de } v_i}$$

onde os  $v_i$ 's são vértices da malha e  $j = 1, \dots, n$

Para selecionarmos  $m$  pontos de controle, basta sortearmos  $m$  números entre 0 e 1 e marcarmos no vetor  $u$  os primeiros elementos (não marcados) maiores que os números selecionados.

Esta forma de seleção faz com que as regiões com maior detalhe geométrico possuam um número grande de pontos de controle e regiões de curvatura média e baixa possuam alguns pontos para representá-las, isto é, faz com que haja uma distribuição proporcional dos pontos de controle em relação às regiões de maior significância geométrica. (Figura 3.14).

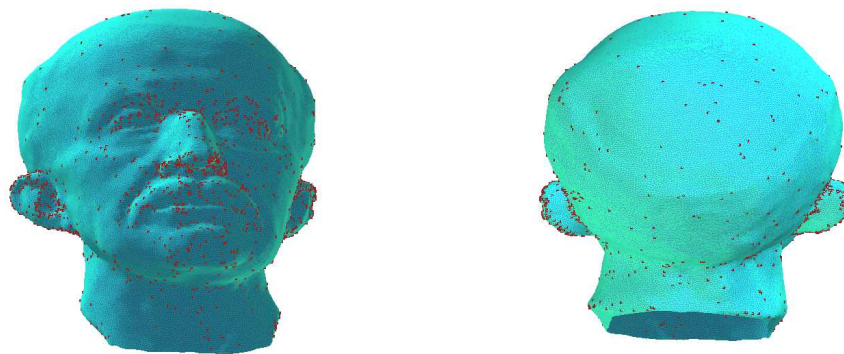


Figura 3.14: Modelo Max com 2% dos pontos selecionados como de controle. Note que há uma distribuição proporcional dos pontos de controle.