



**Universidade Federal de Alagoas**

**Programa de Pós-Graduação em Matemática**

**DISSERTAÇÃO DE MESTRADO**

**Simplificação de malhas triangulares  
baseada no diagrama de Voronoi  
intrínseco**

**Douglas Cedrim Oliveira**

Rio São Francisco

DOUGLAS CEDRIM OLIVEIRA

Simplificação de malhas triangulares baseada no  
diagrama de Voronoi intrínseco

Dissertação de Mestrado na área de concentração de Computação Gráfica submetida em 18 de Fevereiro de 2011 à Banca Examinadora, designada pelo Colegiado do Programa de Pós-Graduação em Matemática da Universidade Federal de Alagoas, como parte dos requisitos necessários à obtenção do grau de mestre em Matemática.

**Orientador:** Prof. Dr. Dimas Martínez Morera

Maceió  
2011

**Catálogo na fonte  
Universidade Federal de Alagoas  
Biblioteca Central  
Divisão de Tratamento Técnico**

**Bibliotecária Responsável: Helena Cristina Pimentel do Vale**

O48s Oliveira, Douglas Cedrim.  
Simplificação de malhas triangulares baseada no diagrama de Voronoi intrínseco  
/ Douglas Cedrim Oliveira. – 2011.  
66 f.

Orientador: Dimas Martínez Morera.  
Dissertação (mestrado em Matemática) – Universidade Federal de Alagoas.  
Instituto de Matemática. Maceió, 2011.

Bibliografia: f. 66.

1. Voronoi, Diagrama de. 2. Malhas intrínsecas – Amostragem. 3. Simplificação  
de malhas. I. Título.

CDU: 514.772.2:004.92

# Simplificação de malhas triangulares baseada no diagrama de Voronoi intrínseco

Douglas Cedrim Oliveira

Dissertação de Mestrado na área de concentração de Computação Gráfica submetida em 18 de Fevereiro de 2011 à Banca Examinadora, designada pelo Colegiado do Programa de Pós-Graduação em Matemática da Universidade Federal de Alagoas, como parte dos requisitos necessários à obtenção do grau de mestre em Matemática.

Banca Examinadora:

---

Prof. Dr. Dimas Martínez Morera (orientador)

---

Prof. Dr. Adelailson Peixoto

---

Prof. Dr. Thales Vieira

---

Prof. Dr. Thomas Lewiner

# Agradecimentos

À minha família, em particular meus pais Yêda Lúcia e Edson Duarte, meus irmãos Daniela Cedrim e Daniel Cedrim e tios, pelo constante apoio e carinho.

Ao meu orientador, Prof. Dr. Dimas Martínez pelo seu apoio, comprometimento e pelas várias reuniões que foram de suma importância para o desenvolvimento deste trabalho.

Ao Prof. Dr. Adailson Peixoto por suas sugestões e ajuda na compreensão de alguns tópicos ao longo do trabalho.

Ao Prof. Dr. Thales Vieira e Prof. Dr. Thomas Lewiner por aceitarem compor a banca examinadora deste trabalho e por suas valiosas sugestões.

A todos da secretaria, mesmo Márcia e Silvinha que não mais estão lá, pela ajuda.

Àqueles que fizeram e fazem parte do Laboratório de Computação Gráfica - que já são muitos - ora por sua amizade, ora pelos momentos de descontração e aprendizado, pelos incentivos e sugestões durante toda a elaboração deste trabalho. Em especial aos amigos Allan, Renata, Cabral, Fabiane, Ailton, Michel e Adriano (Karlinha, integrante café-com-leite).

A todos os companheiros de Mestrado, pois através da amizade, companheirismo e compartilhamento de conhecimento, momentos importantes e difíceis ao longo do curso tornaram-se menos difíceis e foram concluídos.

Aos docentes do Instituto de Matemática que tanto contribuíram na minha formação acadêmica. Em especial aos professores Enoch Apaza, Feliciano Vitório, Dimas Martínez e Adailson Peixoto, pelo convívio durante e após as disciplinas ministradas por eles que cursei ao longo do curso.

Além dos citados, aos meus amigos que estiveram ao meu lado nesses anos de mestrado. Suas importantes e sábias palavras.

À Alice por seu amor, carinho, fé, paciência e por aceitar me dividir com outra: a dissertação.

Agradeço à CAPES pelo apoio financeiro durante o curso.

# Resumo

Nesta dissertação, estudamos o processo de simplificação de malhas triangulares, caracterizando-o com suas particularidades.

Discutimos uma adaptação para superfícies triangulares do método de simplificação baseado em uma cobertura de Voronoi proposto por Peixoto [2002]. Além disso, utilizamos o método *Fast Marching* como uma nova métrica e diferentes estratégias para seleção de vértices da malha simplificada, como a seleção por curvatura.

A simplificação ocorre a partir de um diagrama de Voronoi intrínseco à malha. Discutimos algumas condições necessárias para que a partir do dual desse diagrama, obtenha-se uma malha sem singularidades que seja equivalente a malha original.

**Palavras-chave:** simplificação de malhas, simplificação de malhas triangulares, diagrama de Voronoi intrínseco, subamostragem de malhas

# Abstract

In this dissertation, we study the triangular mesh simplification process, describing its main characteristics.

We discuss an adaptation for triangular meshes of a mesh simplification process based on Voronoi coverage proposed by [Peixoto \[2002\]](#). Moreover, we use Fast Marching Method as a distance function over the mesh and some different strategies for simplified mesh vertices selection, like curvature based selection.

The simplification process is done by constructing an intrinsic Voronoi diagram over the original mesh. We discuss some necessary conditions to obtain a mesh, as Voronoi dual, without any singularities and topologically equivalent to the original mesh.

**Keywords:** mesh simplification, triangular mesh simplification, intrinsic Voronoid diagram, mesh coarsening, mesh subsampling

# Lista de Figuras

1.1	À esquerda um <i>vértice singular</i> e à direita uma <i>aresta singular</i> . . . . .	11
1.2	Redundância de informação em triangulações. . . . .	13
1.3	O uso de diferentes níveis de detalhe do modelo <i>bunny</i> [Luebke et al., 2003].	14
2.1	Método que não preserva topologia. . . . .	17
2.2	Diferentes níveis de detalhamento de acordo com o observador [Hoppe, 1997].	19
2.3	$H(A, B) = \ v\ $ e $H(B, A) = \ w\ $ . . . . .	21
2.4	Em (a), superfases da malha de um crânio. Já em (b) a malha original e o resultado da simplificação. . . . .	23
2.5	Classificação dos vértices para decimação. . . . .	24
2.6	Exemplo do processo de simplificação por envelopes. . . . .	25
2.7	Elipsóides de erro $E(x)$ . . . . .	25
2.8	Colapso de aresta $\{v_i, v_j\}$ . . . . .	26
2.9	Junção de vértices não adjacentes. . . . .	26
2.10	Possíveis operações sobre o complexo simplicial que compõe a malha. . . .	27
2.11	Operação de colapso de arestas e separação de vértices. . . . .	28
2.12	Exemplo de malha $2D$ simplificada através de agrupamento espacial. . . .	29
3.1	Exemplo de uma cobertura de discos sobre o modelo <i>Bimba Con Nastrino</i> . . . .	32
3.2	À esquerda, $N_1(v)$ em cinza claro e em cinza escuro a região de Voronoi associada à $v$ . À direita, os ângulos utilizados na discretização pela cotangente. . . . .	34
3.3	Diferentes aproximações da geodésica entre dois pontos. À esquerda, um possível resultado por Dijkstra e à direita utilizando FMM [Morera, 2006]. . . . .	35
3.4	Detalhes da discretização do FMM. . . . .	36
3.5	Disco gerado sobre o modelo <i>Bimba con Nastrino</i> com a distância através do método <i>Fast-Marching</i> . . . . .	37
3.6	Triangulação dual degenerada. . . . .	39

---

3.7	Violação da primeira condição. . . . .	41
3.8	Violações da segunda condição da propriedade da bola fechada. . . . .	41
3.9	Violação da terceira condição. . . . .	42
3.10	Geração das células de Voronoi. . . . .	43
3.11	Exemplo de uma cobertura de Voronoi sobre o modelo <i>Bimba con Nastrino</i> . . . . .	44
3.12	Geração do dual ao diagrama de Voronoi. Orientação através dos triângulos característicos. . . . .	46
4.1	Interface da aplicação desenvolvida . . . . .	47
4.2	Diferentes simplificações do modelo <i>Bimba con Nastrino</i> . . . . .	50
4.3	Colorizações através da distância de Hausdorff da malha <i>Bimba con Nastrino</i> , com as malhas simplificadas exibidas na figura 4.2. . . . .	51
4.4	Diferentes simplificações do modelo <i>Camel</i> . . . . .	52
4.5	Colorizações através da distância de Hausdorff da malha <i>Camel</i> , com as malhas simplificadas exibidas na figura 4.4. . . . .	53
4.6	Alguns resultados mostrando as malhas originais e diferentes simplificações. . . . .	54
4.7	Mais alguns resultados mostrando as malhas originais e diferentes simplificações. . . . .	55
4.8	Um exemplo de utilização de diferentes níveis de detalhes. . . . .	56

# Lista de Algoritmos

1	Área de Voronoi generalizada $A_{mixed}(v)$ . . . . .	34
2	Pseudo código da geração da cobertura de discos. . . . .	38
3	Pseudo código da HaViolacaoCelula(c). . . . .	42
4	Pseudo código da geração da cobertura de Voronoi. . . . .	43
5	Pseudo código do refinamento da cobertura de discos e de Voronoi. . . . .	44
6	Pseudo código da geração da malha simplificada. . . . .	45

# Sumário

1	INTRODUÇÃO	10
1.1	Definições preliminares . . . . .	10
1.2	Porque simplificar? . . . . .	12
1.3	Objetivo . . . . .	15
1.4	Estrutura do trabalho . . . . .	15
2	SIMPLIFICAÇÃO: UMA VISÃO GERAL	16
2.1	Caracterização . . . . .	16
2.2	Avaliação de erro na simplificação . . . . .	19
2.3	Algumas abordagens para simplificação . . . . .	22
3	UM MÉTODO DE SIMPLIFICAÇÃO POR AGRUPAMENTO	30
3.1	Cobertura de discos . . . . .	30
3.1.1	Seleção de pontos . . . . .	31
3.1.2	Métricas . . . . .	34
3.1.3	Geração da Cobertura . . . . .	37
3.2	Diagrama de Voronoi intrínseco . . . . .	38
3.2.1	Boa formação do diagrama de Voronoi . . . . .	39
3.2.2	Geração do diagrama de Voronoi . . . . .	42
3.3	Grafo dual e malha simplificada . . . . .	43
3.3.1	Geração da malha . . . . .	44
4	RESULTADOS	47
5	CONSIDERAÇÕES FINAIS	57
	REFERÊNCIAS	59

# 1. INTRODUÇÃO

EM computação gráfica, um modelo digital corresponde a uma representação virtual de um objeto, seja ele alguma forma do mundo físico ou algo puramente abstrato. Esses modelos podem ser criados através de um artista ou usuário utilizando algum programa de modelagem (*CAD - Computer Aided Design*), de forma procedural, a partir de dados adquiridos de ressonâncias magnéticas ou tomografias, métodos sísmicos ou até mesmo dados provenientes de *scanners 3D*.

Esses modelos podem ser representados através de um conjunto de pontos, uma iso-superfície definida em um determinado volume ou um conjunto de polígonos. A representação poligonal por triângulos será abordada neste trabalho. Ela possui várias vantagens como simplicidade matemática, sistema de coordenadas associado e constitui uma boa aproximação linear por partes do modelo. Além disso, as principais placas gráficas possuem instruções nativas para trabalhar com triângulos, o que aumenta consideravelmente o desempenho de processamento das malhas. É possível obter uma representação poligonal a partir de uma representação por pontos [Kobbelt and Botsch, 2000] ou volumétrica [Lorensen and Cline, 1987; Lewiner et al., 2003].

## 1.1 Definições preliminares

Algumas definições apresentadas abaixo ajudam a caracterizar uma representação poligonal de um modelo.

**Definição 1.1** (Triangulação). *Uma triangulação é um conjunto de vértices, arestas e faces  $(V, A, F)$ , onde a intersecção de quaisquer triângulos  $T, S \in F$ , satisfaz uma das condições abaixo:*

- *É vazia,*
- *É um vértice comum a  $T$  e  $S$ ,*
- *É uma aresta comum a  $T$  e  $S$ .*

Ao longo do texto, *triangulação* pode ser encontrada apenas como *malha de triângulos*, *malha* ou *variedade discreta* apenas por uma questão de simplicidade. Quando a malha possui um grande número de triângulos (mais de 100K triângulos, por exemplo) ela é dita *densa*. *Complexidade da malha* pode ser vista como o custo computacional para seu armazenamento ou processamento.

De um ponto de vista contínuo, uma superfície regular define-se através de abertos na superfície que sejam homeomorfos a discos topológicos. No caso discreto, a regularidade se dá através de vizinhanças de vértices que definem uma *variedade discreta regular*.

Apesar da *topologia* ser um conceito bem mais amplo [Munkres, 1988], caso não seja feita nenhuma menção explícita, ele será utilizado como a descrição da conectividade da malha de triângulos, ou seja, a descrição das relações de incidência nos elementos da malha. Essas relações incluem: para cada aresta, os vértices e faces que nela incidem; as relações entre cada face e seus vértices e arestas e para cada vértice as arestas e faces incidentes nele.

Para muitos algoritmos em computação gráfica, é desejável que as triangulações sejam localmente homeomorfas a um disco (ou semi-disco, no bordo), constituindo uma *variedade discreta regular*, apesar que na prática isso nem sempre acontece. Na figura 1.1, mostramos dois exemplos que ilustram esse problema [Botsch et al., 2007].

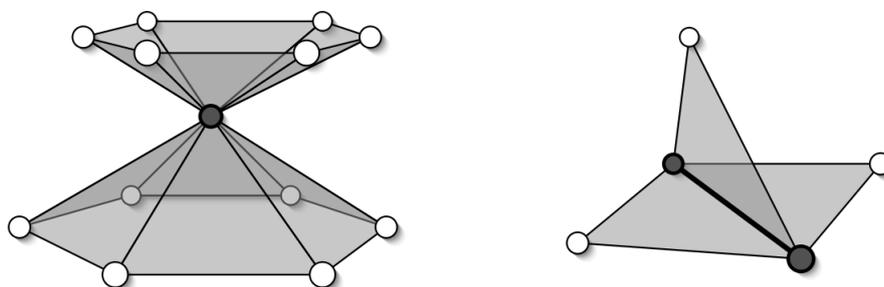


Figura 1.1: À esquerda um *vértice singular* e à direita uma *aresta singular*.

## 1.2 Porque simplificar?

Uma idéia inicial é que para um maior detalhamento de um modelo, as malhas criadas devam ser densas, com o objetivo de tornar sua representação mais fiel. Muitas vezes, o resultado disso pode ser uma malha de triângulos super-amostrada. Assim, uma questão interessante é investigar se essas malhas podem ser simplificadas de modo que ainda constituam uma boa representação desse modelo.

De que forma esses modelos simplificados podem ser úteis?

A seguir, serão expostos alguns pontos a se considerar quanto a essas indagações.

### Redução de complexidade

Quanto maior a quantidade de dados, maior será o custo computacional para efetuar algum processamento sobre eles. Apesar de todo o avanço no hardware disponível para processamento gráfico, onde uma única placa gráfica efetua mais de  $10^{12}$  operações de ponto flutuante (*TFlops*) com precisão simples, cenas compostas com malhas muito densas ainda são um problema para aplicações em tempo real. Dessa forma, a redução da complexidade da malha pode constituir um fator essencial para que o processo de renderização <sup>1</sup> da cena não seja comprometido.

### Redundância

Uma superfície representada por uma malha densa pode possuir informação geométrica ou topológica desnecessária, do ponto de vista que não contribui com nenhuma informação para a aplicação final. Pode existir um modelo equivalente mais simples que não possua diferenças geométricas com o primeiro. Na figura 1.2, mostramos duas triangulações. Na primeira há uma quantidade muito maior de vértices que na segunda. Dependendo da aplicação, ambas malhas podem representar uma mesma superfície plana e perceptualmente em nada diferir. Porém, o fazem no contexto da quantidade de informação necessária para representá-las.

### Nível de detalhe - *LOD*

Uma forma de diminuir os custos de processamento de uma determinada cena é utilizando variações do nível de detalhe (*level of detail - LOD*) de um determinado modelo

---

<sup>1</sup>Renderização é utilizado como um estrangeirismo da palavra inglesa *rendering*.

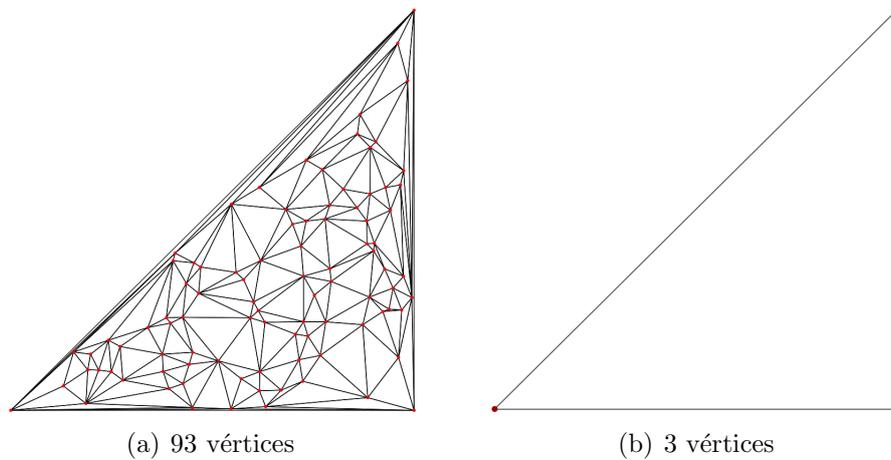


Figura 1.2: Redundância de informação em triangulações.

[Clark, 1976]. Essa técnica é comum em vários contextos. Um exemplo é quando a resolução de amostragem é muito maior que a resolução de exibição, ou seja, várias faces representadas por apenas um *pixel*. Isso pode ocorrer quando o objeto situa-se a uma distância muito grande do observador, fazendo com que ele possua um tamanho relativo muito pequeno na cena. Como ilustrado na figura 1.3, isso leva a idéia de adaptar a resolução dessa malha com o seu tamanho em cena, diminuindo sua complexidade.

## Transmissão progressiva

Uma outra situação possível decorre da possibilidade de transmitir malhas através de um determinado canal de dados, por exemplo a Internet. Modelos muito complexos requerem um considerável espaço para armazenamento e conseqüentemente transmissão, dificultando esse tipo de operação. A idéia de criar modelos com diferentes níveis de complexidade, permite definir um processo que comece com uma malha base densa e efetue a simplificação até obter uma malha suficientemente menos complexa, de forma que sejam criados níveis (malhas) intermediários. Com isso, a transmissão pode ser feita inicialmente com a malha simplificada e os detalhes da malha, ou seja, as etapas intermediárias geradas, são transmitidas de forma progressiva [Hoppe, 1996; Cheng, 2008]. Um exemplo prático pode ser visto em navegadores *web* ao carregarem grandes imagens ou no Google Body [Google, 2010].

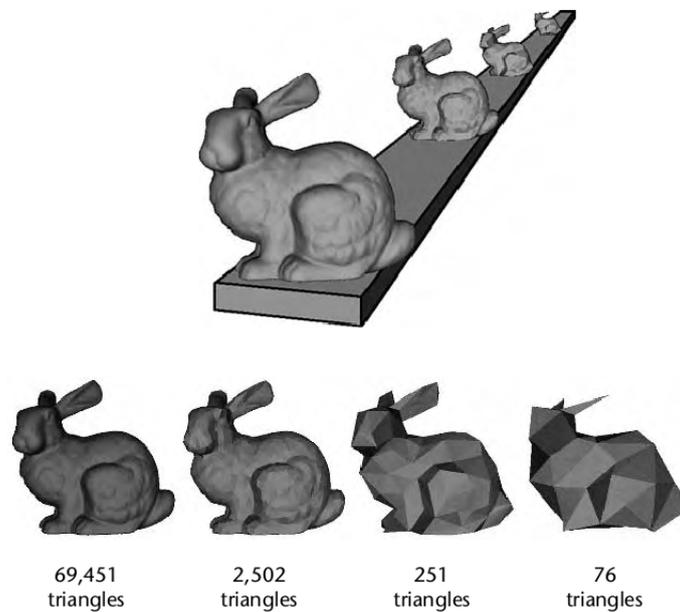


Figura 1.3: O uso de diferentes níveis de detalhe do modelo *bunny* [Luebke et al., 2003].

## Compressão com perdas

Quando a complexidade de uma malha de triângulos é dada pelo tamanho necessário para armazená-la em memória, o processo de reduzir a complexidade pode ser visto como uma compressão com perdas, controlada pelo fator de redução da malha original.

## Detecção de colisão

Ao trabalhar com animação ou com aplicativos que possam efetuar a simulação de algum ambiente físico (ambientes de realidade virtual, jogos), surge a necessidade de detectar quando dois objetos diferentes colidem entre si. Aproximações grosseiras através de volumes envolventes são utilizadas para representar o objeto [Akenine-Moller et al., 2002]. Porém, dependendo do objeto, a precisão alcançada pode ficar muito aquém do satisfatório. Uma forma de contornar esse problema é utilizar uma versão suficientemente simplificada da malha original para os cálculos de colisão.

## Elementos finitos

O processo de simulação numérica de vários problemas pode utilizar a discretização de equações contínuas, através dos métodos de elementos finitos, por exemplo. Uma etapa prévia à simulação consiste na geração de uma triangulação do domínio, onde a convergência das discretizações está relacionada com a densidade do domínio e as particularidades do processo a ser simulado [Date et al., 2005].

A simplificação da malha seguindo critérios controlados de erro serve para reduzir o tempo necessário para a simulação.

## 1.3 Objetivo

Este trabalho tem por objetivo:

- Adaptar o método de simplificação determinístico e baseado em agrupamento de vértices proposto por Peixoto [2002], para malhas densas de triângulos,
- Avaliar como a introdução de outras métricas e estratégias para seleção de vértices modifica a malha simplificada,
- Garantir que a malha simplificada seja topologicamente equivalente a malha original [Peixoto, 2002; Boubekur and Alexa, 2009; Edelsbrunner and Shah, 1994].

## 1.4 Estrutura do trabalho

A parte que segue do trabalho está estruturada da seguinte forma:

- **Capítulo 2:** Neste capítulo é feita uma caracterização do problema de simplificação, incluindo uma visão geral de alguns trabalhos relacionados.
- **Capítulo 3:** Neste capítulo discutimos o processo de simplificação de malhas que é a base deste trabalho, desde a seleção e agrupamento de pontos até a criação da malha resultante
- **Capítulo 4:** Neste capítulo apresentamos vários resultados obtidos com a implementação do método proposto e sua avaliação com diferentes tipos de malhas
- **Capítulo 5:** Neste capítulo são feitas as considerações finais e a discussão de alguns possíveis trabalhos futuros

## 2. SIMPLIFICAÇÃO: UMA VISÃO GERAL

**N**A literatura pode-se encontrar vários artigos que se propõem a analisar os diversos métodos existentes para simplificação de malhas [Cignoni et al., 1997; Heckbert and Garland, 1997; Luebke et al., 2003]. Em geral, definir o melhor método de simplificação corresponde a definir o mais adequado a um determinado objetivo. Ao longo desta seção, serão explicitadas algumas características em que os trabalhos e suas triangulações resultantes diferem.

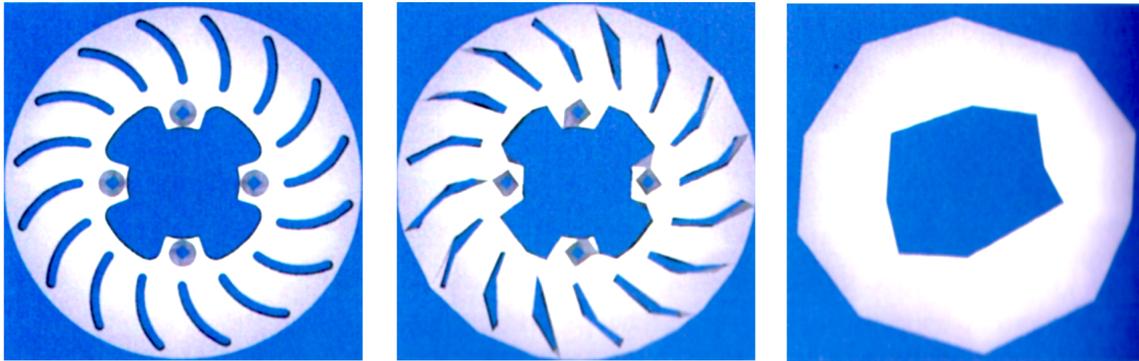
### 2.1 Caracterização

Esta seção tem por objetivo organizar as diferentes classes de algoritmos para simplificação de malhas de acordo com suas principais características.

#### Topologia

Um algoritmo que *preserva topologia*, não altera a conectividade regular da malha. Mantém o gênero, ou seja, não cria novos buracos ou fecha os inerentes à superfície. Além disso, preserva a quantidade de componentes conexas. Devido a isso, a fidelidade visual da malha simplificada geralmente é boa. Entretanto, o processo de simplificação acaba ficando limitado, principalmente em superfícies com gêneros maiores. Quando a malha não representa uma variedade discreta regular, há algoritmos que são *tolerantes à topologia*, ou seja, regiões de singularidades da malha não são alteradas.

Um algoritmo que *não preserva topologia* não possui garantias de preservar o gênero da superfície, nem suas componentes conexas. Observe na figura 2.1, que essa estratégia possibilita simplificações maiores da malha, mas com o preço de diferir bem mais, visualmente ou por medidas numéricas, da malha original.



(a) 4736 triângulos e 21 buracos. (b) 1006 triângulos e 21 buracos. (c) 46 triângulos e 1 buraco.

Figura 2.1: Método que não preserva topologia.

## Seleção de vértices

No processo de geração da malha simplificada é necessário definir, baseado na malha original, quais serão os seus vértices. Assim, eles podem ser um *subconjunto de vértices da malha original*, um *deslocamento desse subconjunto*, ou como pontos obtidos através de um processo de *remalhamento*.

Em geral, esse último possibilita a geração de malhas de maior qualidade, tanto no controle da conectividade da malha simplificada, na reamostragem adaptativa em regiões de interesse (e.g. alta curvatura), quanto na área dos triângulos [Alliez et al., 2008; Hormann et al., 2001], o que é uma boa característica em processamentos numéricos, como elementos finitos. Porém, o processo de *remalhamento* possui alguns problemas inerentes [Alliez et al., 2008], tais como:

- Erros de aproximação à geometria da malha original,
- *Aliasing*.

Além disso, nem sempre é adequado. Tome como exemplo uma malha que originalmente possui um campo escalar definido em cada vértice. Após o processo, o campo escalar na malha resultante teria de ser reconstruído e com isso, perder-se-iam informações ou erros seriam introduzidos nos valores dos novos campos.

## Quantificação do processo

O processo de simplificar uma malha envolve também a escolha de um critério de otimização. Em geral, as principais abordagens encontradas na literatura são:

- Simplificar uma malha para um determinado número de faces previamente estabelecido, minimizando o erro  $\epsilon$  de aproximação entre as malhas,
- Simplificar uma malha para um determinado erro de aproximação  $\epsilon$  entre as malhas previamente estabelecido, minimizando o número de faces da malha simplificada,
- Simplificar de acordo com um tamanho máximo determinado a ser ocupado pela malha em memória ou tempo para sua obtenção,
- Simplificar seguindo critérios indiretos na qualidade da malha resultante.

## Estático, incremental

Quando o objetivo final é apenas reduzir a complexidade de uma malha muito densa, em um único estágio, o processo de simplificação diz-se *estático*. Uma aplicação direta é *LOD* discreto [Luebke et al., 2003], onde para cada nível há um modelo simplificado com um determinado número de faces através de uma etapa de pré-processamento.

Por outro lado, a simplificação pode ser obtida de forma *incremental*, ou seja, um vértice, face ou aresta é removido em cada passo de acordo com algum critério quantitativo. Uma possível aplicação é *LOD* contínuo [Luebke et al., 2003], comumente observado em jogos digitais, onde o modelo simplificado desejado é gerado na execução.

## Adaptatividade ao observador

Alguns processos são ditos *independentes do observador*, ou seja, reduzem os detalhes de uma maneira quase uniforme ao longo da malha. São bem comuns por sua simplicidade de implementação. No caminho oposto, a simplificação pode ser obtida através de uma *adaptação* ou *dependência do observador*, ou seja, ao volume de visão de uma câmera. Assim, para uma determinada região contida nesse volume e com sua projeção visível, há uma maior quantidade de triângulos, ao passo que as outras regiões podem ser bem mais simplificadas. Na figura 2.2 mostramos um exemplo dessa adaptatividade.

É comum a associação de *dependência do observador* à *anisotropia* e a *independência do observador* à *isotropia*.

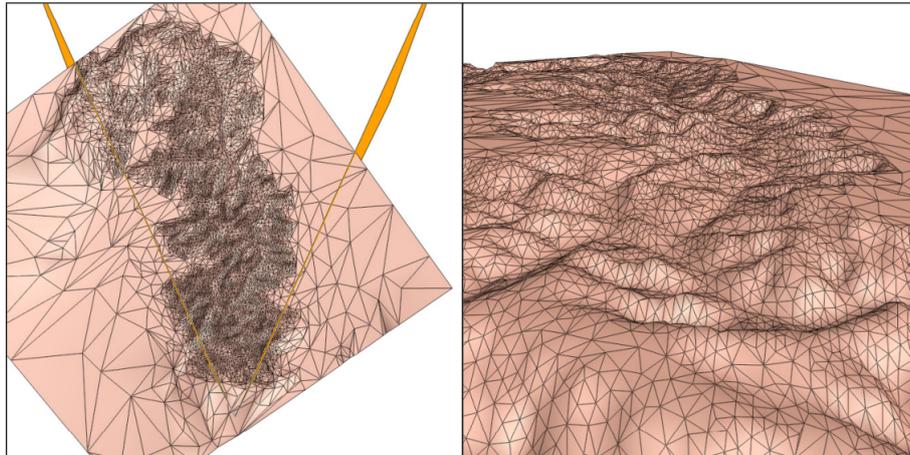


Figura 2.2: Diferentes níveis de detalhamento de acordo com o observador [Hoppe, 1997].

## 2.2 Avaliação de erro na simplificação

Antes de analisar as diferentes formas de se medir o erro em um processo de simplificação é importante entender o que o erro significa e o porquê de medi-lo [Luebke et al., 2003].

- **Critério de avanço e parada:** Dependendo do algoritmo, em cada etapa do processo de simplificação é utilizada uma otimização, onde dentre as possíveis operações é escolhida a que minimize o erro entre a malha densa e a simplificada. É determinado também, se ainda pode ser efetuada alguma operação na malha.
- **Avaliar os resultados:** Além de malhas simplificadas serem utilizadas para aplicações em tempo real, algumas aplicações necessitam de um controle da precisão do modelo simplificado, como análise de elementos finitos ou detecção de colisão.
- **Níveis de detalhe:** Ao se utilizar níveis de detalhe diferentes, é necessário escolher o mais adequado entre eles através de suas estimativas de erro em coordenadas de tela, satisfeitas para um valor limite de erro.

A estimativa do erro de aproximação entre as superfícies, que pode ser efetuada tanto *a priori*, ou seja, como critério de avanço e parada, quanto *a posteriori*, é uma boa forma de quantificar o quão a malha simplificada é parecida com a malha original, pois ajuda a prevenir que haja grandes discrepâncias entre as malhas. Ainda assim, pode *não ser calculado*, onde o usuário define a taxa de simplificação, por exemplo [Cignoni et al., 1997].

Apesar de diferenças na geometria serem uma estimativa muito comum, há também formas de calcular os erros de aproximação através de medições no espaço de atributos do

objeto [Luebke et al., 2003]. Note que os atributos de avaliação podem ser combinados com o objetivo de melhorar o resultado.

- Erro geométrico
  - Distância de Hausdorff,
  - Normais,
  - Espaço de tela.
- Erro no espaço de atributos
  - Cor,
  - Coordenadas de textura.

A seguir, será abordado apenas a *distância de Hausdorff* por ter sido o principal critério utilizado na avaliação dos resultados deste trabalho. É fácil encontrar na literatura uma análise mais detalhada dos outros pontos [Luebke et al., 2003].

## Erro geométrico

Quando a avaliação do erro se dá no espaço do objeto, é possível encontrar uma classificação de vários algoritmos que depende de quais elementos geométricos são utilizados na construção das correspondências [Luebke et al., 2003]. Essas correspondências podem vir de alguma das seguintes maneiras:

- *Vértice-Vértice*,
- *Vértice-Plano*,
- *Vértice-Superfície*,
- *Superfície-Superfície*.

## Distância de Hausdorff

Utilizada em diversas aplicações que vão desde topologia a visão computacional [Huttenlocher and Rucklidge, 1993], passando por processamento de imagens [Alhichri and Kamel, 2002], a *distância de Hausdorff* é utilizada também em simplificação de superfícies como uma forma de quantificar a diferença entre a malha original e a malha simplificada,

uma vez que, em síntese, ela mede o quão distante entre si estão dois conjuntos de pontos. Para isso, basta observar que a geometria de ambas é definida através de seus respectivos vértices, que nada mais são que conjuntos de pontos no espaço.

**Definição 2.1** (Distância de Hausdorff). *Sejam  $A$  e  $B$  dois subconjuntos não vazios em um espaço métrico  $(M, d)$ . A função  $H : A \times B \rightarrow \mathbb{R}$ , com*

$$H(A, B) = \sup_{a \in A} \{ \inf_{b \in B} d(a, b) \} \quad (2.1)$$

mede a proximidade de  $A$  e  $B$ . A distância de Hausdorff  $d_H$  é definida como

$$d_H(A, B) = \max\{H(A, B), H(B, A)\} \quad (2.2)$$

Formalmente, a expressão 2.1 não define uma métrica, pois  $H(A, B) \neq H(B, A)$  (ver figura 2.3). Apesar disso, é bastante útil em algumas ocasiões. Não é difícil encontrar as expressões 2.1 e 2.2 como *distância de Hausdorff* e *distância de Hausdorff simétrica*, respectivamente [Gotsman et al., 1998]. Assim, essa notação será utilizada ao longo do texto. Na figura 2.3 é exibido um exemplo da distância de Hausdorff entre duas superfícies.

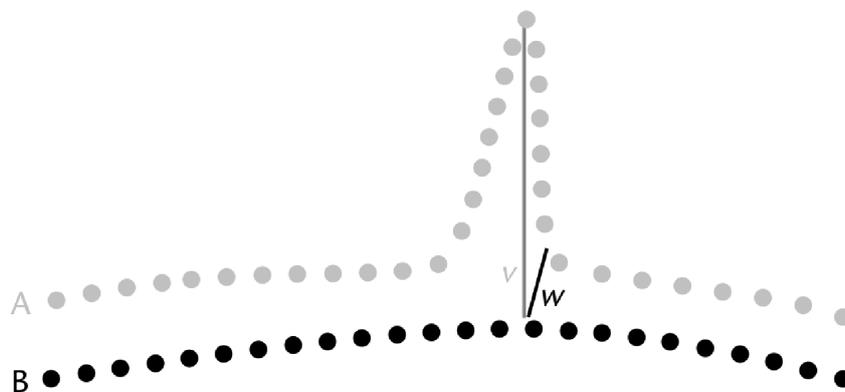


Figura 2.3:  $H(A, B) = \|v\|$  e  $H(B, A) = \|w\|$ .

## Ferramenta *Metro*

Um problema inerente a estimativas de erro entre a superfície original e a simplificada é a variedade de abordagens possíveis, onde algumas delas dependem do tipo de algoritmo utilizado para realizar a simplificação. Com isso, passa a ser desejável um método de

avaliação mais geral, onde particularidades entre cada método não afetem a avaliação das malhas resultantes.

Com esse objetivo, a ferramenta *Metro* possibilita medir a *distância de Hausdorff* entre duas superfícies, utilizando a abordagem Vértice-Superfície. A idéia principal do algoritmo é considerar uma amostragem em pequenas regiões da malha original, e a partir daí calcular uma distância média entre esses pontos e a superfície simplificada [Cignoni et al., 1998].

## 2.3 Algumas abordagens para simplificação

A seguir apresentamos algumas abordagens do problema de simplificação de malhas. Adicionalmente, várias outras podem ser encontradas na literatura [Eck et al., 1995; Gross et al., 1996; Vieira et al., 2004; Heckbert and Garland, 1997; Alliez et al., 2008; Cignoni et al., 1997].

### Junção de faces

Conjuntos de faces coplanares, ou quase coplanares são agrupadas em polígonos maiores que são então retriangulados, de forma a reduzir a quantidade de faces.

### Superfícies [Kalvin and Taylor, 1996]

Antes de mais nada, é necessário definir um *retalho* de uma superfície, que consiste em um conjunto de faces adjacentes em uma determinada região da superfície. Os vértices que situam-se no perímetro desse retalho definem o bordo do retalho. Assim, uma *superfície* é o polígono formado através desse bordo. Na figura, 2.4 é mostrado um exemplo de superfícies.

Em cada etapa de criação das superfícies, é determinado um plano  $P = \{(a, b, c) | ax + by + z = d\}$  como uma aproximação linear da superfície. A partir disso, em linhas gerais a simplificação se dá em três etapas:

- **Criação das superfícies:** Inicialmente, uma face é selecionada como semente e adicionada a uma superfície. Para cada face  $f_b$  adjacente a essa superfície são efetuados três testes. O primeiro checa se todos os vértices de  $f_b$  distam de um erro controlado  $\epsilon$  de  $P$ . O segundo teste checa se o ângulo entre as normais de  $f_b$  e  $P$  é menor que um valor limite  $\theta_{max}$ . O terceiro checa se  $f_b$  não cruza a superfície de modo a alterar sua topologia.

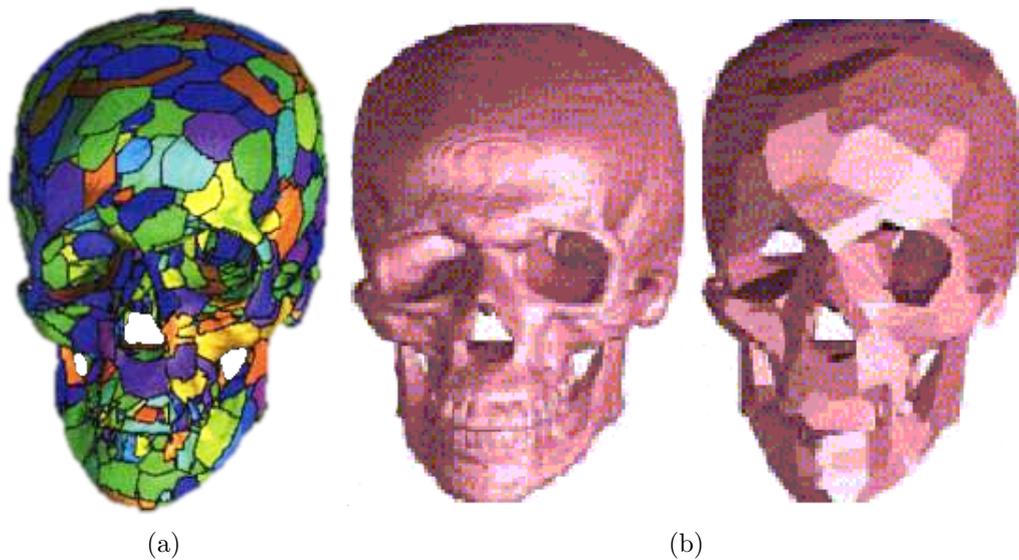


Figura 2.4: Em (a), superfases da malha de um crânio. Já em (b) a malha original e o resultado da simplificação.

- **Criação de superarestas:** Após a definição das superfases, há uma otimização nas arestas que as delimitam, de modo a formar um bordo simplificado (*superaresta*).
- **Triangulação da superfase:** A partir disso, a superfase delimitada pelas superarestas é projetada em  $P$ , o polígono obtido da projeção é transformado em um polígono monótono e então obtida sua triangulação.

Este método preserva topologia, é aplicável a qualquer modelo e possui limite de erro global.

Observe que os critérios de erros adotados foram tanto o geométrico *Vértice-Plano* quando no espaço de atributos, no caso das *normais*.

## Decimação

Em geral, elimina de forma iterativa vértices, faces ou arestas, que não satisfizerem um determinado critério local e efetua uma triangulação da região removida.

### Decimação de malhas triangulares [Schroeder et al., 1992]

O algoritmo é definido em três etapas:

A primeira etapa classifica os vértices como: *simples*, *singular*, *de bordo*, *de aresta interior* ou *de canto*. Quando um vértice possui duas arestas características é dito *de aresta*

*interior*. Quando possui mais de duas é dito de *canto*. Uma aresta  $e$  é dita característica se o seu ângulo diedral é maior que um valor limite  $\delta$ .

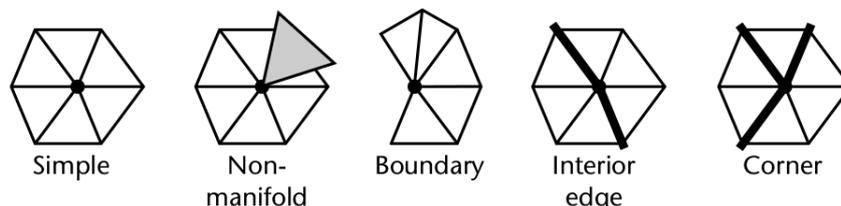


Figura 2.5: Classificação dos vértices para decimação.

Vértices singulares e de bordo não são deletados da malha, ao passo que os outros são candidatos.

A segunda etapa avalia se os vértices candidatos satisfazem determinados critérios para remoção. Assim, para cada vértice simples, através das normais  $N_i$ , centróides  $c_i$  e áreas  $A_i$  de cada triângulo de sua 1-vizinhança estrelada, é calculado um plano médio  $P$  com normal  $N_P$  e com  $x \in P$  de forma que

$$N_P = \frac{\sum N_i A_i}{\sum A_i}, x = \frac{\sum c_i A_i}{\sum A_i}.$$

Assim, o vértice  $v$  será removido se  $d(v, P) > \epsilon$ .

A terceira e última etapa consiste em efetuar triangulações locais nas regiões que houve remoção de vértice.

É um método rápido, de fácil implementação e tolerante à topologia. Como elementos geométricos que não preservam localmente a topologia de um disco não são classificados para remoção, acaba por limitar a simplificação.

### Simplificação por envelopes [Cohen et al., 1996]

Difere dos principais algoritmos de decimação pois ao invés de utilizar critérios locais, utiliza um critério global. O algoritmo inicia criando duas novas malhas através de um *offset* na malha original, a cada uma é dado o nome de *envelope*. Ambas são construídas através do deslocamento de seus vértices por uma distância  $\epsilon$  ao longo da direção do seu vetor normal. Na imagem 2.6(a), os envelopes são construídos tomando  $\epsilon > 0$  e  $\epsilon < 0$ .

Os vértices da malha original são ordenados em uma fila. Para cada vértice retirado da fila sua remoção é aceita se a triangulação da região sem o vértice ainda for interior aos envelopes interno e externo, como pode ser visto na figura 2.6(b).

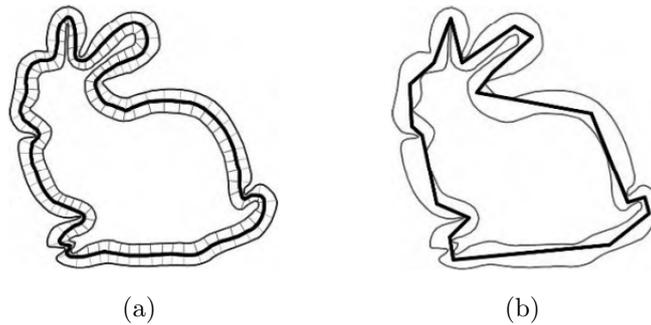


Figura 2.6: Exemplo do processo de simplificação por envelopes.

### Métrica de erro por quádricas [Garland and Heckbert, 1997]

Uma forma mais elaborada consiste em definir para cada vértice em cada iteração uma métrica de erro de acordo com o desvio do vértice às faces as quais ele pertencia na iteração anterior. Para tal, observe que para um triângulo  $T_i$ , a sua distância de um ponto  $x$  pode ser escrita como  $(n_i^T x - p_i)^2$ , com  $n_i$  normal a  $T_i$ . O funcional  $E(x) = \sum_i (n_i^T x - p_i)^2$  define o erro no vértice e seus isocontornos são elipsóides. Na figura 2.7, é ilustrado o erro associado a cada vértice.



Figura 2.7: Elipsóides de erro  $E(x)$ .

Assim, os pares de vértices adjacentes que possuem menos erro são marcados para colapso da aresta, que pode ser para qualquer ponto da aresta formada por esses vértices. Ilustrado na figura 2.8.

Como pode ser visto na figura 2.9, caso se deseje uma maior simplificação, deve-se considerar vértices que não são adjacentes para junção. Uma possível escolha é tomar vértices e planos em uma determinada dada região do espaço. Note que isso permite alterar a topologia da malha simplificada.

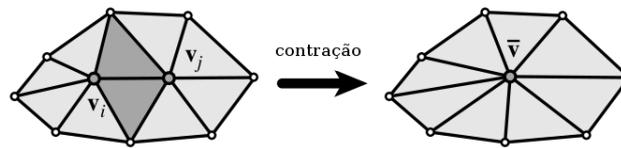


Figura 2.8: Colapso de aresta  $\{v_i, v_j\}$ .

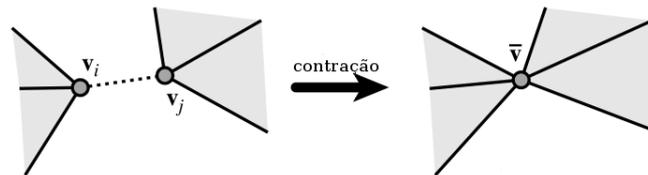


Figura 2.9: Junção de vértices não adjacentes.

### *Re-tiling* [Turk, 1992]

O algoritmo recebe como entrada uma malha densa e a quantidade de vértices da malha simplificada. Esses pontos são distribuídos aleatoriamente sobre as faces e até mesmo sobre os vértices da malha original. Em seguida ocorre um processo de relaxamento da posição desses pontos sobre a malha, de forma que para um ponto  $p$ , os pontos próximos a ele são projetados no plano tangente à superfície em  $p$  e calcula-se a força de repulsão que cada um exerce sobre ele, movendo-o para uma nova posição na malha.

Após essa etapa, os pontos adicionados estarão fixados na malha. Gera-se uma nova malha triangulada através da adição deles e em seguida, os vértices da malha original são removidos e uma triangulação local é refeita. Observe que isso aproxima a geometria e topologia originais da malha, apesar de não obter garantias e de possuir limitações inerentes a um remalhamento.

### Otimização de energia

Nessa classe de algoritmos são efetuadas operações de remoção ou contração locais com o objetivo de controlar a simplificação através de um funcional de energia.

### Mesh optimization [Hoppe et al., 1993]

A partir de uma malha inicial  $M_0$  e um conjunto de pontos  $X$ , encontrar uma malha  $M$  que aproxime  $X$  com uma menor quantidade de vértices e seja topologicamente equivalente

a  $M_0$  através da minimização do funcional

$$E(M) = E_{dist}(M) + E_{rep}(M) + E_{spring}(M)$$

onde o primeiro e segundo termos controlam a posição e a quantidade de vértices, de forma indireta, na malha resultante e o último utilizado para garantir a convergência.

Três são as operações possíveis na malha: *edge collapse*, *edge split* e *edge swap*. Quando uma operação não altera a topologia geral da malha ela é dita *legal*. Na figura 2.10, as três operações são mostrados.

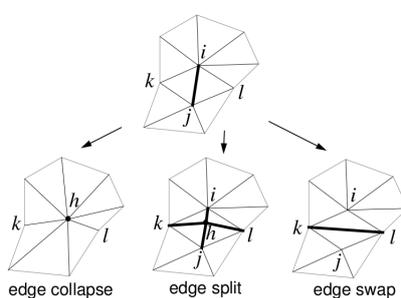


Figura 2.10: Possíveis operações sobre o complexo simplicial que compõe a malha.

O processo ocorre de forma iterativa através de duas otimizações, uma sobre sua topologia e a outra sobre sua geometria. Para minimizar sobre ambas, em cada etapa gera-se um complexo simplicial  $K'$  através de uma *operação legal*. Para  $K'$  fixo, a posição dos vértices que minimiza o erro  $V'$  é determinada, definindo uma malha  $M'$ . Assim, se  $E(M')$  é menor que o erro da malha anterior, passa a ser a nova malha. A simplificação é obtida quando algum critério de convergência seja satisfeito.

Observe que isso corresponde a um algoritmo guloso, o que não garante que a minimização encontrada seja global.

### ***Progressive meshes*** [Hoppe, 1996]

Este trabalho é uma extensão do anterior [Hoppe et al., 1993]. Parte de uma malha inicial  $M^n$  para uma malha simplificada  $M_0$  através de um conjunto de operações de colapso de arestas (*ecol*), ou seja,  $M^n \rightarrow ecol_{n-1} \rightarrow \dots \rightarrow ecol_0$ , de forma a minimizar

$$E(M) = E_{dist}(M) + E_{spring}(M) + E_{scalar}(M) + E_{disc}(M)$$

De forma análoga, os dois primeiros termos controlam a posição e a quantidade de

vértices, de forma indireta. O terceiro controla as distâncias no espaço de atributos (cor, normal). O último termo controla as arestas características.

Ao invés de considerar um conjunto de operações possíveis [Hoppe et al., 1993], utiliza apenas uma, o colapso de arestas (e sua inversa, a separação de vértices) para navegar sobre diferentes níveis de representação de detalhes. Com as duas operações juntas, o processo de transmissão progressiva fica facilitado [Cheng, 2008]. Na figura 2.11, ambas operações são exibidas.

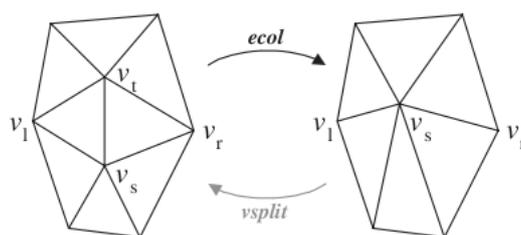


Figura 2.11: Operação de colapso de arestas e separação de vértices.

## Agrupamento de vértices

Neste conjunto de algoritmos, através de um certo limite de erro  $\epsilon$ , vértices da malha são agrupados particionando a malha inicial ou o espaço no qual ela está contida em conjuntos de raio menor que  $\epsilon$ . Para cada agrupamento, um vértice é calculado (podendo ser um subconjunto da malha inicial) e definido como representante do agrupamento. Os algoritmos associados diferem na forma que calculam o agrupamento e seus vértices representativos.

### Agrupamento espacial [Rossignac and Borrel, 1992]

Trata-se de uma forma bem genérica de simplificação, onde o volume que envolve o objeto é subdividido uniformemente em uma grade regular de forma que para cada unidade da grade (*voxel*) apenas um vértice é mantido. A escolha se dá de forma que a cada vértice da malha original é associado um peso. Para uma determinada célula, o vértice com maior peso é dito seu representante.

Duas abordagens diferentes são utilizadas no cálculo do peso. A primeira prioriza vértices próximos da silhueta em uma direção arbitrária, ou seja,  $w_1(p) = \frac{1}{\max\{\theta_i(p)\}}$ , onde  $\theta_i(p)$  é o  $i$ -ésimo ângulo entre pares de arestas incidentes a  $p$ . Já a segunda, prioriza vértices que estão em grandes faces através de  $w_2(p) = \|\max\{e_i(p)\}\|$ , com  $e_i$  a  $i$ -ésima aresta incidente no vértice  $p$  e  $\|\cdot\|$  representa o comprimento da aresta.

Após efetuada a seleção dos vértices representativos, é efetuada sua triangulação seguindo a topologia inicial da malha.

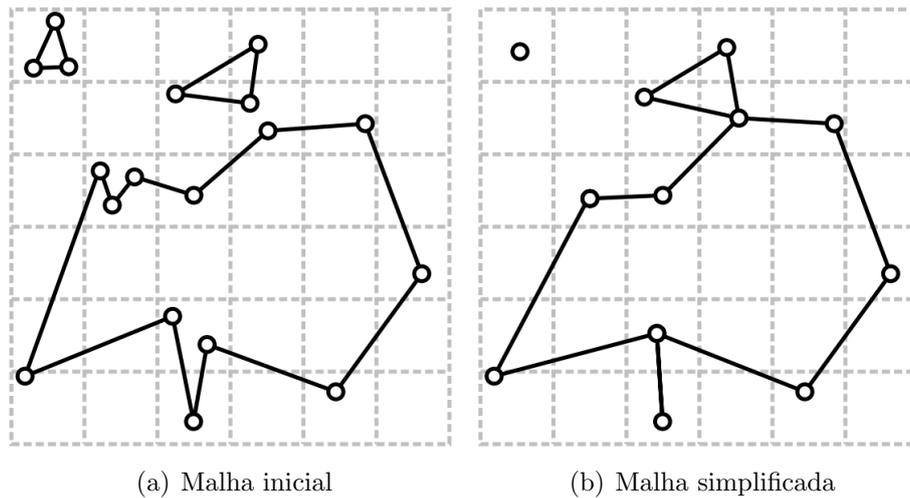


Figura 2.12: Exemplo de malha 2D simplificada através de agrupamento espacial.

O algoritmo consegue obter grandes simplificações de modelos, dado que é possível alterar a topologia, porém, com o custo de não preservar detalhes da malha e de degenerar triângulos, como pode ser visto na figura 2.12.

Uma extensão natural de agrupamento espacial uniforme [Rossignac and Borrel, 1992] é através da adaptação da simplificação à geometria da malha, considerando o refinamento do espaço em grades não regulares (*quadtrees*, *octrees*) [Luebke, 1996].

# 3. UM MÉTODO DE SIMPLIFICAÇÃO POR AGRUPAMENTO

O processo de simplificação descrito e implementado neste trabalho baseia-se principalmente nos trabalhos de Peixoto [2002] e Boubekur and Alexa [2009]. Pode ser classificado como uma abordagem determinística por agrupamento de vértices que, em contraste com agrupamentos espaciais, é intrínseco à malha. Ele não altera buracos na superfície nem suas componentes conexas, ou seja, preserva topologia. Além disso, os vértices da malha simplificada são obtidos como um subconjunto de vértices da malha original, o processo é estático e independente do observador.

A simplificação é dividida em três etapas: a primeira etapa consiste na seleção de pontos da malha original que serão vértices da malha simplificada a partir da criação de uma cobertura de discos na malha original; em seguida é criado um diagrama de Voronoi na malha; e o passo final é gerar a malha simplificada através do dual do diagrama de Voronoi.

## 3.1 Cobertura de discos

Para o processo de criação de discos é necessário que a malha utilizada represente uma variedade discreta regular, de forma que a vizinhança considerada defina um disco na malha controlado por um determinado raio.

Antes de descrever a etapa de criação de discos algumas definições são úteis.

**Definição 3.1** (Disco na malha). *Seja  $V$  o conjunto de vértices  $\{v\}$  da malha. Um disco na malha  $D_\delta(v)$  possui centro em  $v$  e raio máximo  $\delta$  e é definido como o conjunto  $\{w_i\} \subset V$  tais que*

$$\|w_i - v\|_{\hat{M}} < \delta,$$

onde  $\|\cdot\|_{\hat{M}}$  é uma métrica  $\hat{M}$  definida na malha.

**Definição 3.2** (Cobertura da malha por discos). *Seja  $V$  o conjunto de vértices de uma malha  $M$  e  $CD$  uma coleção de discos na malha definida como*

$$CD = \{D_\delta(v) : \text{com } v \in V \text{ e } \delta \in \mathbb{R}\}.$$

*Dizemos que  $CD$  é uma cobertura de  $M$  por discos se satisfaz a seguinte condição:*

$$V \subset \bigcup_{D_\delta(v) \in CD} D_\delta(v). \quad (3.1)$$

A etapa inicial consiste em determinar um subconjunto de vértices  $V_0 \subset V$  da malha original. Para isso, é criada uma cobertura de discos  $CD$  sobre a malha. Como visto na expressão 3.1, todo vértice da malha original pertence a pelo menos um disco. Em cada disco o seu centro  $c_i$  é utilizado como vértice representante de forma que pertencerá apenas àquele disco e  $V_0 = \{c_0, c_1, \dots, c_n\}$ . Assim a escolha dos vértices da malha simplificada corresponde a determinar um critério de geração da cobertura.

Na figura 3.1, é mostrado um exemplo de uma cobertura de discos  $CD$ .

Vale observar também que esse processo interfere diretamente tanto na geometria quanto na topologia da malha simplificada resultante, uma vez que, a densidade da amostragem é determinada pelo raio de cada disco na cobertura.

### 3.1.1 Seleção de pontos

A seguir, serão examinados alguns critérios que foram utilizados ao longo deste trabalho para geração da cobertura de discos. Em todas as abordagens, a cobertura é gerada a partir de um valor fixo para o raio, que aqui utilizamos  $\delta$  para ilustrar.

## Incremental

Uma primeira abordagem consiste em obter  $V_0$  de forma *incremental*. Para tal é construída uma fila de vértices. O primeiro vértice  $v_0$  é utilizado como centro do primeiro disco



Figura 3.1: Exemplo de uma cobertura de discos sobre o modelo *Bimba Con Nastrino*.

$D_\delta(v_0)$  e adicionado à cobertura. A partir daí, são retirados vértices da fila até que algum  $v_i$  não esteja presente na cobertura, sendo então utilizado como centro do segundo disco  $D_\delta(v_i)$  da cobertura. O algoritmo procede até que todos da fila tenham sido processados, fazendo com que todos vértices da malha original pertençam a, pelo menos, um disco. Esse critério é mais simples de implementar, mas acaba sendo muito simplificado pois não leva em consideração características inerentes à malha original. Essa é a abordagem utilizada na tese de Peixoto [2002].

### Ponto mais distante

Outra forma de proceder é escolhendo um vértice inicial qualquer  $v_0$  como o centro do primeiro disco  $D_\delta(v_0)$  e a partir daí, o algoritmo procede escolhendo o centro do próximo disco como sendo o *ponto mais distante* a toda a cobertura atual, ou seja, para cada disco que fora criado e adicionado à cobertura  $CD$  toma-se um vértice  $u$  de forma que este seja o mais distante, na malha, dos centros dos discos da cobertura  $CD$

$$u = \arg \max_w \left\{ d \left( w, \bigcup_{D \in CD} D \right), w \in V \setminus CD \right\}$$

Note que a função de distância  $d$  aqui utilizada pode representar diferentes métricas (ver 3.1.2) e que o conceito de ponto mais distante vem da ideia de excentricidade de um vértice no grafo. A ideia de utilizar essa abordagem é tentar reduzir a quantidade de pontos necessários para definir a cobertura de discos.

## Curvatura gaussiana

Ambos os métodos descritos não levam em consideração a geometria da malha como critério para a criação da cobertura. Uma extensão natural consiste em utilizar atributos intrínsecos à superfície e sua representação para obter uma melhor aproximação da malha original na malha simplificada.

Dada uma representação da superfície por uma malha de triângulos, que pode ser vista como sua aproximação linear por partes, as estimativas de operadores diferenciais precisam ser adaptadas para uma versão discreta.

Utilizando o teorema de Gauss-Bonnet e considerando uma malha de triângulos, temos que a curvatura total é escrita como:

$$\iint_{A_M} K \, dA = 2\pi - \sum_{i=1}^{\#f} \theta_i$$

com  $K$  a curvatura gaussiana,  $\theta_i$  o ângulo em  $v$  do  $i$ -ésimo triângulo da  $N_1(v)$ , ilustrado na figura 3.2 à esquerda, e  $A_M$  um retalho adequado da malha.

Como pode ser visto em Meyer et al. [2002], para garantir uma boa discretização da curvatura gaussiana utiliza-se médias locais em torno de cada vértice. Assim, a curvatura gaussiana pode ser definida como:

$$\hat{K} = \frac{1}{A_{mixed}} \iint_{A_{mixed}} K \, dA$$

onde  $A_{mixed}$  é uma generalização da área de Voronoi para triângulos obtusos, ilustrada na figura 3.2, que constitui um bom controle numérico de erro. Ela é calculada de acordo com o algoritmo 1.

A seleção de pontos ocorre de forma semelhante à abordagem incremental, porém os vértices são inseridos em uma fila de prioridade pelo valor absoluto da curvatura gaussiana.

```

início
   $A_{mixed}(v) \leftarrow 0$ 
  // Triângulo  $t$  na 1-vizinhança estrelada de  $v$ 
  para todo  $t \in F(N_1(v))$  faça
    se  $t$  não é obtuso então
       $A_{mixed}(v) += \frac{1}{8} \sum_{v_i \in N_1(v)} (\cot \alpha_i + \cot \beta_j) \|v - v_i\|^2$  // Área de Voronoi
    fim
  senão
    se  $t$  é obtuso em  $v$  então
       $A_{mixed}(v) += \text{Area}(t)/2$ 
    fim
    senão
       $A_{mixed}(v) += \text{Area}(t)/4$ 
    fim
  fim
fim

```

**Algoritmo 1:** Área de Voronoi generalizada  $A_{mixed}(v)$ .

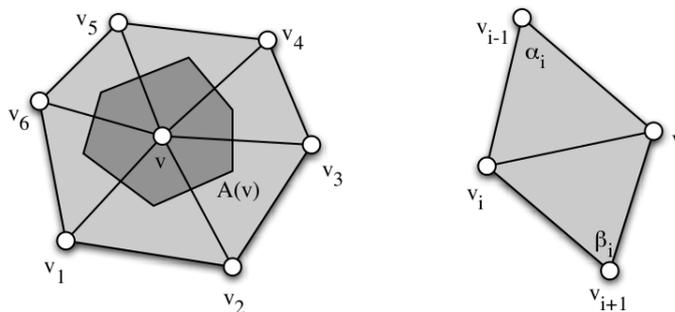


Figura 3.2: À esquerda,  $N_1(v)$  em cinza claro e em cinza escuro a região de Voronoi associada à  $v$ . À direita, os ângulos utilizados na discretização pela cotangente.

### 3.1.2 Métricas

O conceito de métrica aqui utilizado refere-se a funções utilizadas para medir distância em uma malha de triângulos.

Até então não foi feita nenhuma restrição quanto à métrica utilizada, o que modifica o conjunto de discos sobre a malha. Utilizaremos o conceito de *distância geodésica* para medir a distância entre vértices da malha.

**Definição 3.3** (Distância geodésica em malhas). *A distância geodésica do vértice  $u$  ao*

vértice  $v$ , numa malha, é definida como o mínimo dos comprimentos dos caminhos sobre a malha ligando  $u$  e  $v$ .

Note que o conceito fica vago se  $u$  e  $v$  estão em diferentes componentes conexas. Sendo assim, a distância entre eles passa a ser  $d(u, v) = \infty$ .

Ao longo do trabalho foram utilizadas duas diferentes métricas. Ambas são aproximações para a distância geodésica sobre a malha e serão expostas a seguir.

### Dijkstra

Inicialmente, é utilizada uma técnica bastante difundida em teoria de grafos para cálculo do caminho de custo mínimo que é o algoritmo de Dijkstra [Dijkstra, 1959; Boaventura, 2006].

Em linhas gerais, para calcular a distância entre um ponto  $v$  e um  $w$ , o algoritmo consiste em partir de  $v$  calcular a distância euclidiana para todos os seus vizinhos e definir como o próximo vértice a ser avaliado o que possui menor distância a  $v$ , ou seja, a aresta incidente em  $v$  com menor comprimento. A cada iteração, novas vizinhanças são atualizadas até que o valor seja calculado em  $w$ . Observe que isso acaba fazendo com que a distância seja definida como a soma dos comprimentos das arestas do caminho mínimo entre eles.

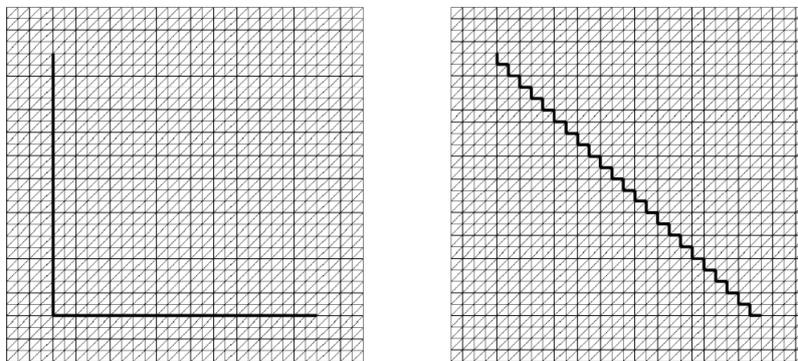


Figura 3.3: Diferentes aproximações da geodésica entre dois pontos. À esquerda, um possível resultado por Dijkstra e à direita utilizando FMM [Morera, 2006].

### Método *Fast marching* - FMM

Uma abordagem possível é considerar a distância não apenas calculada pelas arestas. Para isso, utilizamos uma adaptação para malhas triangulares do método *Fast marching* que utiliza a ideia de avanço de frentes sobre a malha [Kimmel and Sethian, 1998]. Seu

algoritmo define uma aproximação numérica da distância através da solução da equação Eikonal:

$$\|\nabla T\| = F \text{ com } T(v) = 0$$

onde  $F$  controla a velocidade do avanço e  $v$  é o ponto inicial e com  $\|\nabla T\| \equiv \sqrt{T_x^2 + T_y^2}$ .

A idéia do algoritmo é a de uma onda propagando-se na direção do seu gradiente ao longo de uma região com uma determinada velocidade constante, onde define-se a distância de um ponto qualquer da região como o tempo  $T$  em que a onda leva para atingí-lo. Assim, utilizamos  $F \equiv 1$ , o que leva a função distância a um ponto ser monótona crescente. Na figura 3.5, mostramos momentos diferentes da propagação de uma onda.

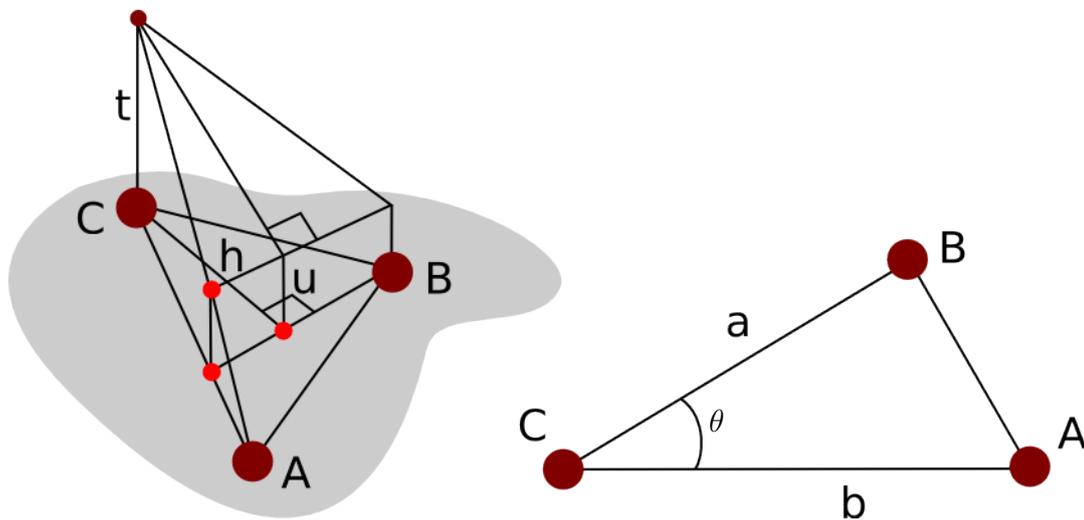


Figura 3.4: Detalhes da discretização do FMM.

Assim como Dijkstra, para determinar a distância do ponto  $u$  ao ponto  $v$ , calcula-se inicialmente a distância à 1-vizinhança estrelada de  $u$  utilizando a distância euclidiana. A partir disso, o algoritmo procede com o cálculo da propagação da onda sobre a malha pela discretização adequada do gradiente definindo uma solução  $T$  em cada vértice  $w$  na vizinhança de  $u$ , utilizando os valores de distância de vizinhos já definidos. Assim, utilizando a notação da figura 3.4, deve-se determinar o valor de  $T(C)$  a partir dos valores conhecidos de  $T(A)$  e  $T(B)$ . Para triângulos agudos, a discretização pode ser obtida da seguinte forma:

- $u = T(B) - T(A)$

- Resolver a equação quadrática

$$(a^2 + b^2 - 2ab \cos \theta)t^2 + 2bu(a \cos \theta - b)t + b^2(u^2 - F^2 a^2 \sin^2 \theta) = 0$$

- Se  $u < t$  e  $a \cos \theta < \frac{b(t-u)}{t} < \frac{a}{\cos \theta}$ , então  $T(C) = \min\{T(C), t+T(A)\}$ . Caso contrário,  $T(C) = \min\{T(C), bF + T(A), aF + T(B)\}$

Para triangulações mais gerais, o algoritmo pode ser encontrado em [Spira and Kimmel \[2003\]](#).

Apesar das semelhanças, difere de Dijkstra pois a propagação considera o interior das faces, e não apenas as arestas [[Kimmel and Sethian, 1998](#); [Sethian, 1999](#)]. Adicionalmente, na figura 3.5, é ilustrado o processo diretamente em uma malha.

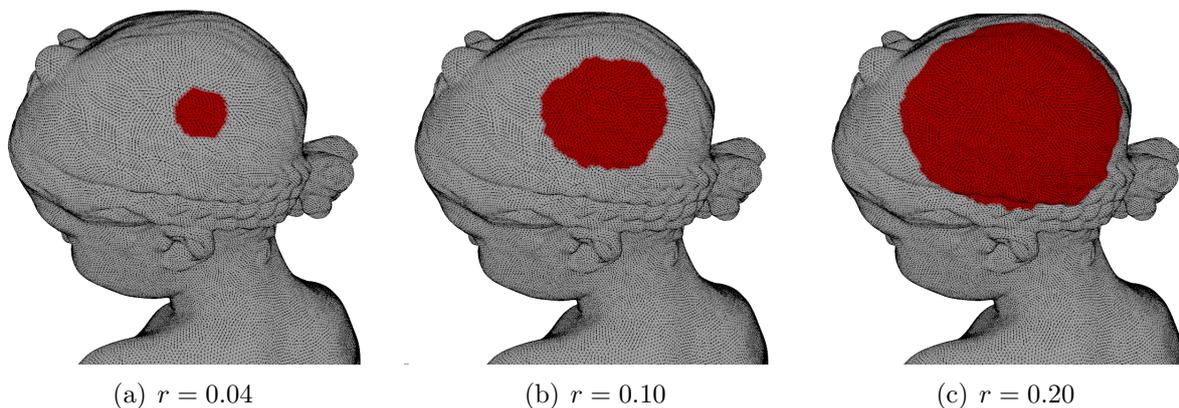


Figura 3.5: Disco gerado sobre o modelo *Bimba con Nastrino* com a distância através do método *Fast-Marching*.

### 3.1.3 Geração da Cobertura

Como visto anteriormente, na etapa de geração dos discos, cada vértice definido como o centro de um disco pertence apenas a esse disco. Ao passo que os demais, podem pertencer a vários discos. O processo toma um vértice inicial  $v_0$ , que depende da abordagem utilizada para a escolha dos pontos, e cria um disco com centro  $v_0$  e raio  $r$ . Baseado na métrica escolhida, novos vértices são adicionados a uma fila de prioridade para determinação de sua distância. A geração do disco continua até que um vértice  $w$ , tal que  $d(v_0, w) > r$  seja encontrado.

```

Entrada: Lista de vértices  $V$  e raio  $r$ 
Saída: Cobertura de discos sobre a malha
início
   $\hat{V} \leftarrow V$ 
  enquanto  $\hat{V} \neq \emptyset$  faça
     $v \leftarrow \text{PróximoVértice}()$  // Próximo vértice da lista  $\hat{V}$ 
    se  $v$  não é proibido então
      Crie o disco  $D(v, r, \text{metrica})$ 
      Marca os vértices de  $V$  adicionados ao disco  $D$  como proibidos ;
      // Garante unicidade do centro
    fim
  fim
fim

```

**Algoritmo 2:** Pseudo código da geração da cobertura de discos.

Essa é a ideia principal na geração da cobertura, com pequenas modificações para a geração da cobertura por pontos mais distantes. Observe que o método *PróximoVértice()* encapsula a estratégia que é utilizada para a seleção de pontos e, a cada iteração, remove o próximo vértice da lista de vértices.

## 3.2 Diagrama de Voronoi intrínseco

O diagrama de Voronoi de um conjunto de pontos  $P$  no  $\mathbb{R}^2$  é a divisão do plano em regiões onde para cada ponto  $p \in P$ , chamado de sítio, cada região contém uma parte do plano que é a mais próxima desse sítio do que qualquer outro. A partir disso, define-se uma extensão do diagrama para malhas de triângulos.

**Definição 3.4** (Diagrama de Voronoi em uma malha). *Seja  $M$  uma malha de triângulos e  $P = \{p_1, p_2, \dots, p_n\} \subset V(M)$  vértices de  $M$ . O diagrama de Voronoi em  $M$ , divide-a em  $n$  células de Voronoi  $V_i$  tais que*

$$V_i := \{z \in V(M) : \|z - p_i\| \leq \|z - p_j\|, j = 1, \dots, n, j \neq i\} \quad (3.2)$$

*quando a norma é do espaço  $\mathbb{R}^3$ , o diagrama de Voronoi na malha se diz restrito. Quando a norma define uma distância geodésica na malha, o diagrama se diz intrínseco.*

Utilizar o diagrama de Voronoi restrito poderia criar células com mais de uma componente conexa, o que não é desejável para o nosso método, como será visto na seção

3.2.1. Dessa forma, o diagrama de Voronoi intrínseco será considerado no processo de simplificação.

Por semelhança com a definição de cobertura de discos, chamaremos de *cobertura de Voronoi (CV)* ao diagrama intrínseco de Voronoi definido sobre a malha. Note que da forma como é definida, a malha original constitui uma triangulação de cada célula de Voronoi.

### 3.2.1 Boa formação do diagrama de Voronoi

A posição e a quantidade de células de Voronoi criadas podem constituir problemas para a geração da malha simplificada [Peixoto, 2002; Boubekur and Alexa, 2009]. Na figura 3.6, mostramos um exemplo de um cobertura de Voronoi não adequada, o que faz com que a malha dual tenha arestas e vértices singulares.

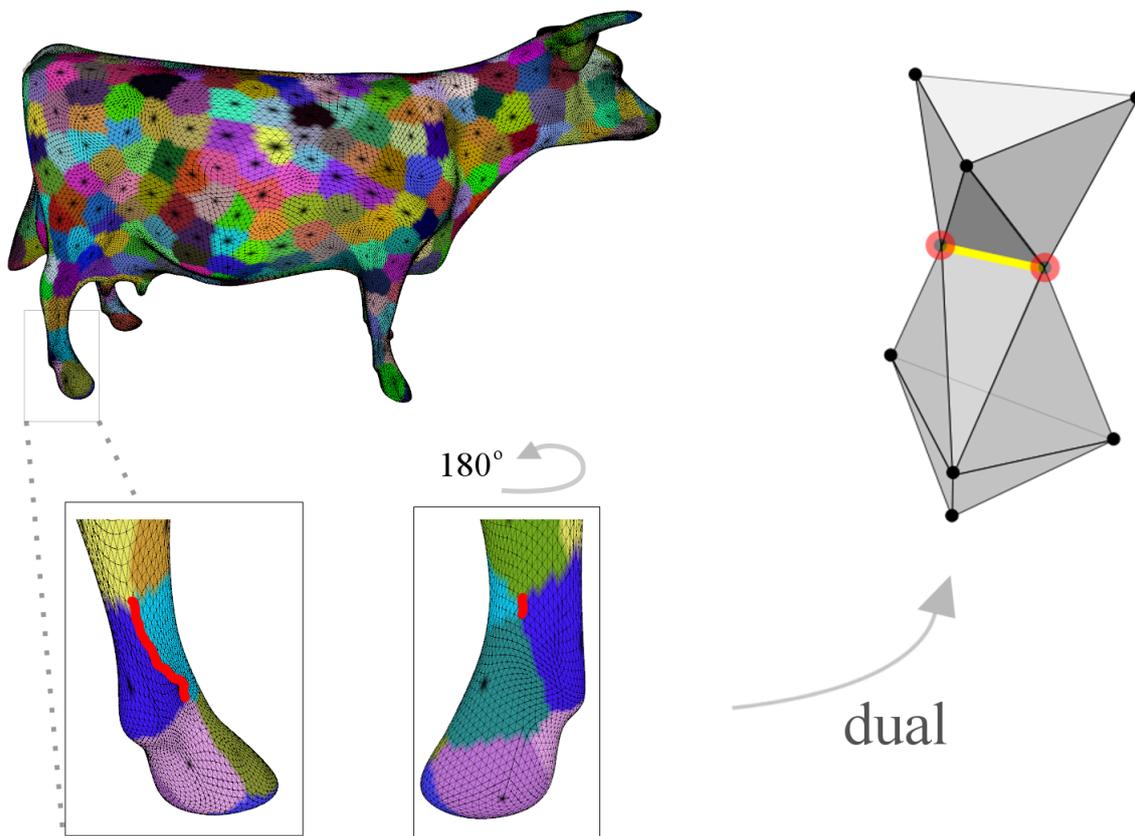


Figura 3.6: Triangulação dual degenerada.

Para que o dual da cobertura de Voronoi constitua uma variedade discreta regular e seja topologicamente equivalente à malha original, é necessário que o diagrama de Voronoi

não viole a *propriedade da bola fechada* [Edelsbrunner and Shah, 1994] enunciada a seguir:

**Proposição 3.1.** *Seja uma cobertura finita de células de Voronoi  $(C_i)_{i=1..k}$ . Essa cobertura satisfaz a propriedade da bola fechada se valem as seguintes condições:*

1.  $C_i$  é um 2-disco, ou seja, é equivalente a um disco topológico,
2. A intersecção não vazia entre  $C_i \cap C_j$ ,  $i \neq j$  é uma única curva simples,
3. A intersecção não vazia entre  $C_i \cap C_j \cap C_k$ ,  $i \neq j \neq k$  é um ponto.

*Demonstração.* A demonstração da proposição pode ser encontrada em Edelsbrunner and Shah [1994]. □

**Definição 3.5** (Boa formação do diagrama de Voronoi). *Um diagrama de Voronoi intrínseco (ou restrito) é dito bem formado se não viola a propriedade da bola fechada*

Antes de descrever como cada condição é avaliada, é importante lembrar a característica de Euler de uma triangulação. Trata-se de um invariante topológico, ou seja, depende apenas da topologia e não da quantidade de elementos da malha.

**Definição 3.6** (Característica de Euler). *A característica de Euler  $\chi$  de uma triangulação  $S$  é dada pela expressão*

$$\chi(S) := v - a + f, \tag{3.3}$$

onde  $v$  é o número vértices,  $a$  de arestas e  $f$  de faces.

### Primeira condição da bola fechada

Uma forma direta de avaliar a primeira condição é utilizando a característica de Euler de uma célula de Voronoi  $C$ , que será homeomorfa a um disco (2-bola) se for uma variedade discreta, conexa e  $\chi(C) = 1$ . Aqui é utilizado o fato da malha original constituir uma triangulação de cada célula para facilitar os cálculos de  $\chi(C)$ .

Na figura 3.7, é exibido um disco de raio  $R$  que viola a primeira condição

### Segunda condição da bola fechada

A intersecção não vazia entre duas células de Voronoi é uma curva poligonal  $\gamma$ , não necessariamente conexa. Esta curva é bem aproximada pelo grafo dual do grafo formado pelas arestas, que têm um vértice em cada uma dessas duas células. Na figura 3.8 à esquerda,  $\gamma$  são as arestas em vermelho. À direita, a aproximação pelo grafo dual.

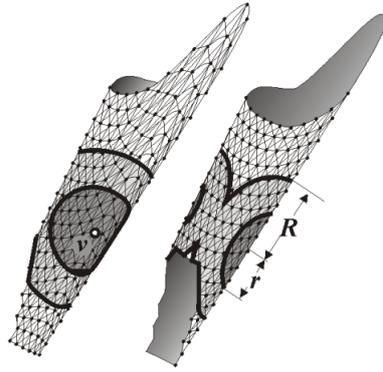


Figura 3.7: Violação da primeira condição.

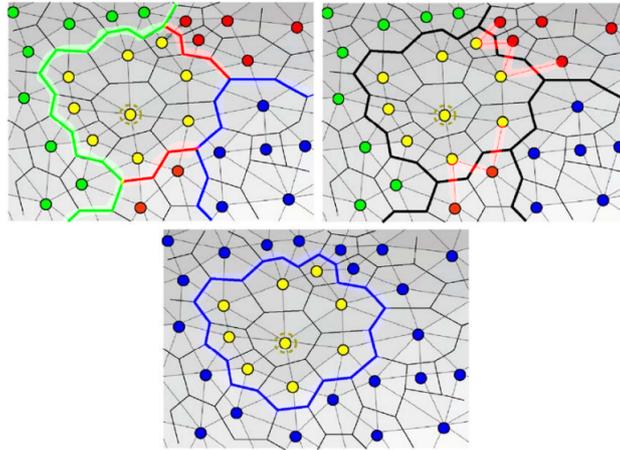


Figura 3.8: Violações da segunda condição da propriedade da bola fechada.

Note que a característica de Euler de  $\chi(\gamma)$  corresponde ao seu número de componentes conexas  $n_c$ . Assim, uma célula *viola a segunda condição se  $n_c \neq 1$* .

Nos casos ilustrados na figura 3.8, temos  $\chi(\gamma) = 2$ , na parte superior, e  $\chi(\gamma) = 0$  na parte inferior.

### Terceira condição da bola fechada

A terceira condição equivale a restringir para três o número mínimo de células vizinhas a uma célula [Dyer et al., 2008; Peixoto, 2002]. Note que isso restringe a malha original a possuir mais que três vértices. Além disso, observe que a maior simplificação de uma malha sem bordo é um tetraedro.

No algoritmo 3, mostramos como pode ser feita a checagem das violações topológicas.

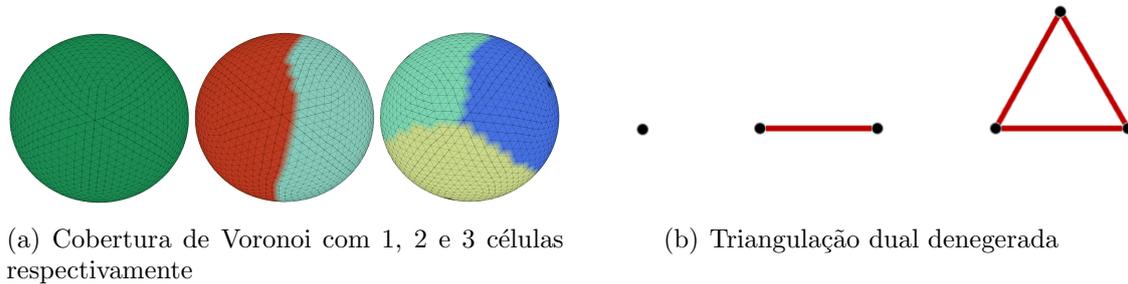


Figura 3.9: Violação da terceira condição.

```

Entrada: Célula  $c$  da cobertura de Voronoi na malha
Saída: Se há ou não violação topológica naquela célula
início
   $viol \leftarrow$  falso
  se  $((\chi(c) \neq 1) \text{ ou } (\#(N_1(c)) < 3))$  então
     $viol \leftarrow$  verdadeiro
  fim
  para todo  $c_i \in N_1V(c)$  faça // 1-vizinhança de Voronoi de  $c$ 
     $\gamma \leftarrow$  CurvaPoligonalInterseccao( $c, c_i$ )
    se  $\chi(\gamma) \neq 1$  então
       $viol \leftarrow$  verdadeiro
    fim
  fim
  retorna  $viol$ 
fim

```

Algoritmo 3: Pseudo código da HaViolacaoCelula(c).

### 3.2.2 Geração do diagrama de Voronoi

Uma vez criada a cobertura de discos, vem a etapa de criação da cobertura de Voronoi. Inicialmente, são criadas  $n$  células vazias, onde  $n$  é o número de discos da etapa anterior. A partir daí, para cada vértice  $v$  da malha original deve ser decidido a qual célula ele pertence. Para isso, cada um dos discos que contém  $v$  tem armazenado a distância  $d_v$ , sobre a malha, do seu centro até  $v$ . Logo,  $v$  será adicionado à célula na qual o disco associado possui menor distância  $d_v$ . O processo é ilustrado na figura 3.10 e no algoritmo 4. Após gerada uma cobertura inicial, o algoritmo procede com a etapa de checagem da boa formação e refinamento.

Caso alguma célula  $c_k$  viole as condições da propriedade da bola fechada, o refinamento é efetuado da seguinte maneira. O disco associado à célula  $c_k$  tem seu raio reduzido por

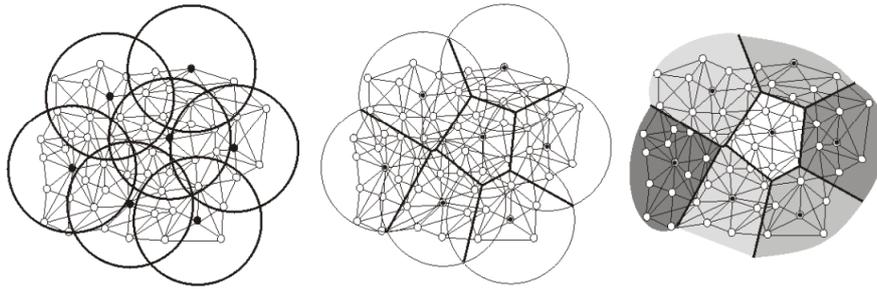


Figura 3.10: Geração das células de Voronoi.

algum fator  $fator \in [0, 1]$ . Foi utilizado  $fator = 0.5$ . Para a lista de vértices que foram excluídos desse disco, caso eles não pertençam a nenhum outro, executa-se a geração da cobertura de discos e em seguida a de Voronoi.

Observe que no algoritmo 5, o método *AtualizaVoronoi()*, deve levar em consideração os novos discos criados, gerando novas células adequadamente. Ao final do processo, obtém-se uma cobertura de Voronoi bem formada, como na figura 3.11.

```

Entrada: Uma cobertura de discos  $CD$ 
Saída: Uma cobertura de Voronoi  $CV$ 
início
  InicializaCélulas()
  para todo  $v \in V$  faça
     $id_{disc} \leftarrow \text{IdDiscoProximo}(v)$  // Menor distância aos centros
    AdicionaNaCélula( $id_{disc}, v$ )
  fim
fim

```

**Algoritmo 4:** Pseudo código da geração da cobertura de Voronoi.

### 3.3 Grafo dual e malha simplificada

No caso do plano, dado um conjunto de pontos  $X$ , a *triangulação de Delaunay* é a melhor triangulação possível desses, pois ela maximiza os menores ângulos de cada triângulo [de Berg et al., 2008]. Uma forma de obtê-la é através do grafo dual ao diagrama de Voronoi de  $X$  [de Berg et al., 2008; O'Rourke, 1998].

No entanto, sua extensão para dimensões maiores tem ganhado atenção através de vários trabalhos, como Dyer [2010] que estuda a relação entre diferentes aproximações de Delaunay. Ele mostra que, para diagramas de Voronoi bem formados, o grafo dual representa

```

início
  para todo  $c \in CV$  faça // Cada célula na cobertura de Voronoi
    enquanto HaViolacaoCelula(c) faça
       $d \leftarrow cell_{id}$ 
      RefinaDisco(d)
      AtualizaVoronoi()
    fim
  fim
fim

```

**Algoritmo 5:** Pseudo código do refinamento da cobertura de discos e de Voronoi.



Figura 3.11: Exemplo de uma cobertura de Voronoi sobre o modelo *Bimba con Nastrino*.

corretamente uma malha de triângulos, que não necessariamente é uma extensão de Delaunay para o espaço.

### 3.3.1 Geração da malha

O processo anterior, de geração da cobertura de Voronoi, induz uma relação de equivalência  $\sim$  entre os vértices da malha original. Dois vértices são equivalentes se eles pertencem a uma mesma célula de Voronoi. Observe que isso define  $n$  classes de equivalência

na malha, com  $n$  o número de células de Voronoi. Essa relação induz três classes de equivalência nos triângulos da malha original, de forma que sejam  $v_1, v_2$  e  $v_3$  três vértices de um triângulo  $t$  qualquer de  $M$ :

1. Se  $v_1 \sim v_2 \sim v_3$ , então  $t$  está totalmente contido em uma célula de Voronoi e será descartado
2. Se  $v_1 \sim v_2 \not\sim v_3$  (ou permutações entre os índices), então  $t$  está na intersecção de duas células e degenera para um segmento de aresta da triangulação resultante
3. Se  $v_1 \not\sim v_2 \not\sim v_3$ , então células correspondentes aos vértices determinam um triângulo na malha simplificada

A última classe de equivalência dos triângulos é bem útil na geração da malha simplificada. Os triângulos que pertencem a ela são ditos *triângulos característicos*.

Para gerar a malha simplificada, percorre-se a cobertura de Voronoi e para cada célula  $C$  e sua 1-vizinhança de Voronoi  $N_1V(C)$ , faz-se uma identificação da célula com seu dual, ou seja, a célula de Voronoi representada pelo seu sítio passa a ser um vértice de um triângulo na malha simplificada e um vértice de Voronoi e dois sítios de células adjacentes definem um triângulo da malha simplificada. Observe que os vértices de Voronoi são interiores a triângulos característicos. O algoritmo 6 descreve este processo.

```

Entrada: A cobertura de Voronoi CV sobre a malha original
Saída: Malha simplificada
início
  para todo  $c \in CV$  faça
     $c_{it} \leftarrow \text{PrimeiroVizinho}(c)$ 
    enquanto  $c \neq c_{prox}$  faça
      // Utiliza os triângulos característicos
       $c_{prox} \leftarrow \text{ProxVizinho}(c, c_{it})$ 
      se  $(c, c_{it}, c_{prox}$  não é triângulo da malha simplificada) então
        | CriaTriangulo( $c, c_{it}, c_{prox}$ )
      fim
    fim
  fim
fim

```

**Algoritmo 6:** Pseudo código da geração da malha simplificada.

Construir de forma coerente a orientação da malha simplificada corresponde a determinar uma ordenação adequada na geração do dual de células de Voronoi adjacentes. Por

construção, os triângulos característicos possuem uma relação biunívoca com os triângulos da malha simplificada. Assim, para definir a orientação da malha simplificada, basta utilizar a orientação definida pelos triângulos característicos, ou seja a orientação da malha simplificada é induzida pela malha original.

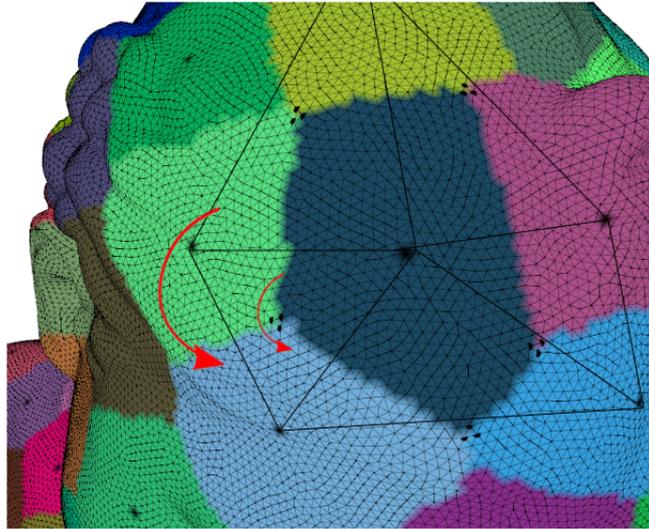


Figura 3.12: Geração do dual ao diagrama de Voronoi. Orientação através dos triângulos característicos.

## 4. RESULTADOS

Para obtenção dos resultados foi implementada uma aplicação na linguagem C/C++, utilizando as bibliotecas OpenMesh 2.0 RC5, FLTK 1.3 e OpenGL 3.0. Para auxiliar na visualização e no cálculo da distância de Hausdorff foram utilizados as ferramentas *Metro* e MeshLab.

A figura 4.1 ilustra os estágios do algoritmo. Na parte superior à esquerda, a malha original; abaixo, a seleção de vértices da malha simplificada; à direita inferior, o diagrama de Voronoi intrínseco e à direita superior, a malha simplificada.

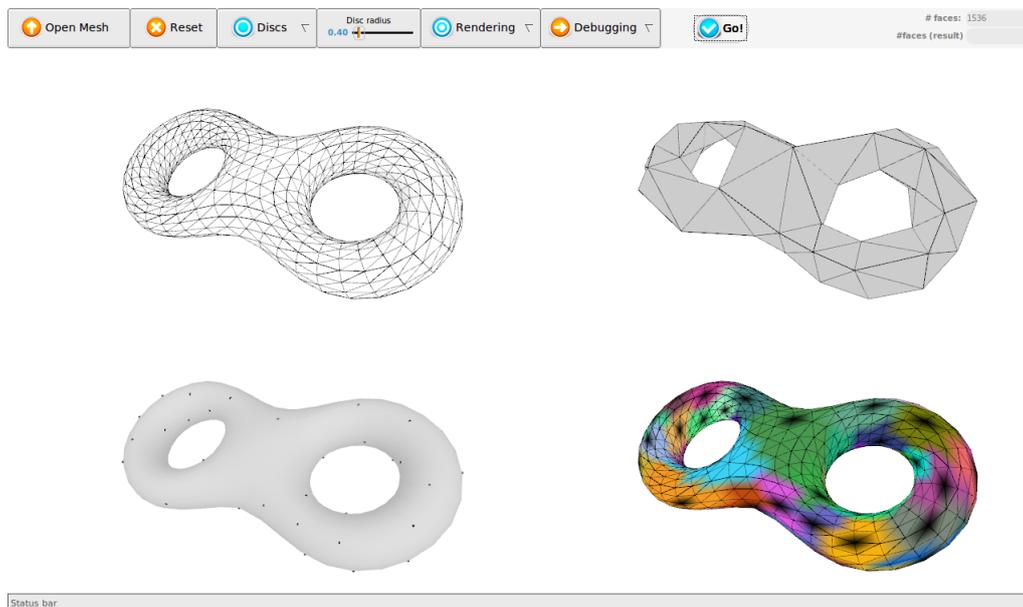


Figura 4.1: Interface da aplicação desenvolvida

É feita uma análise da simplificação através da distância de Hausdorff entre a malha original e a malha simplificada. Para isso, geramos algumas imagens com uma colorização dentre o menor e o maior valor de distância para malhas com métricas, estratégia de distribuição de discos e raio variáveis. Um problema dessa análise é que essa distância

é extrínseca à malha e pode apresentar resultados inadequados para malhas onde duas regiões estejam próximas no espaço, mas distantes por métricas sobre a malha.

Uma outra análise possível é através da razão de aspecto dos triângulos ao longo da malha, que pode ser definida como o quociente entre os comprimentos da maior e menor aresta do triângulo.

Como pode ser visto nas figuras 4.2 e 4.4, para um raio fixo, a utilização do *Fast Marching* para calcular distâncias implica no aumento do número de triângulos na malha final, conseqüentemente na redução do erro de simplificação.

Para simplificar na identificação de cada resultado, utilizamos a notação da tabela 4.2:

<b>Sigla</b>	<b>Métrica</b>	<b>Estratégia</b>
DI	Dijkstra	Incremental
DC	Dijkstra	Baseado em curvatura
DD	<i>Dijkstra</i>	Ponto mais distante
FI	<i>Fast Marching</i>	Incremental
FC	<i>Fast Marching</i>	Baseado em curvatura

Tabela 4.1: Notação utilizada nos resultados

Durante o trabalho, malhas com bordo não foram tratadas. Por isso, não serão exibidos resultados com elas. Ainda assim, o método funciona perfeitamente com esse tipo de malha, basta apenas uma adaptação nos métodos que geram os triângulos a partir do diagrama de Voronoi.

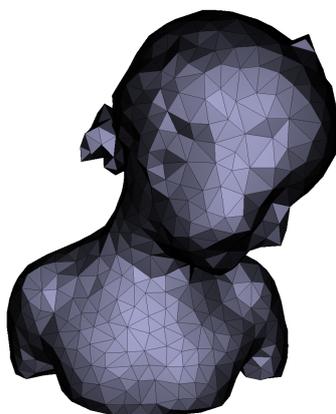
O processo de simplificação, em geral, é rápido. Acontece que quando o diagrama obtido não é bem formado e é necessário um refinamento, o tempo para geração do diagrama de Voronoi aumenta consideravelmente, como no resultado 4.7(f). Uma vez que esta parte do refinamento ainda não está otimizada e não foi feita uma análise de complexidade detalhada, ainda não é possível afirmar que esse tempo alto é inerente ao método.

<b>Resultado</b>	<b>Cob. Discos</b> (em s)	<b>Cob. Voronoi</b> (em s)	<b>Geração dual</b> (em s)	<b>Tempo total</b> (em s)
4.2(b)	4,347	0,438	0,091	<b>4,876</b>
4.2(c)	5,438	624,7	0,006	<b>630,1</b>
4.2(d)	4,846	0,486	0,011	<b>5,343</b>
4.2(e)	5,450	0,489	0,012	<b>5,951</b>
4.2(f)	4,149	731,1	0,030	<b>735,3</b>
4.2(g)	4,143	319,3	0,019	<b>323,5</b>
4.4(b)	1,416	90,35	0,049	<b>91,82</b>
4.4(c)	0,903	353,9	0,224	<b>355,0</b>
4.4(d)	1,416	440,4	0,024	<b>441,8</b>
4.4(e)	0,970	113,4	0,026	<b>114,4</b>
4.4(f)	0,569	289,0	0,180	<b>289,7</b>
4.4(g)	0,903	353,9	0,224	<b>355,1</b>
4.6(b)	0,010	0,031	0,028	<b>0,069</b>
4.6(c)	0,013	0,014	0,007	<b>0,034</b>
4.6(e)	5,637	1076	0,463	<b>1082</b>
4.6(f)	5,029	46,32	0,065	<b>51,41</b>
4.6(h)	0,251	6,780	0,004	<b>7,035</b>
4.6(i)	0,251	4,913	0,003	<b>5,167</b>
4.7(b)	4,058	1,405	7,065	<b>12,53</b>
4.7(c)	4,503	1,956	7,177	<b>13,64</b>
4.7(e)	9,676	2,681	3,986	<b>16,34</b>
4.7(f)	12,25	6899	0,002	<b>6911</b>
4.7(h)	0,397	7,751	0,005	<b>8,153</b>
4.7(i)	0,231	5,787	0,004	<b>6,022</b>

Tabela 4.2: Tempo de processamento dos resultados



(a) Malha original com 149.524 triângulos



(b) Malha simplificada com 2056 triângulos, DI,  $r=0.04$



(c) Malha simplificada com 182 triângulos, FC,  $r=0.20$



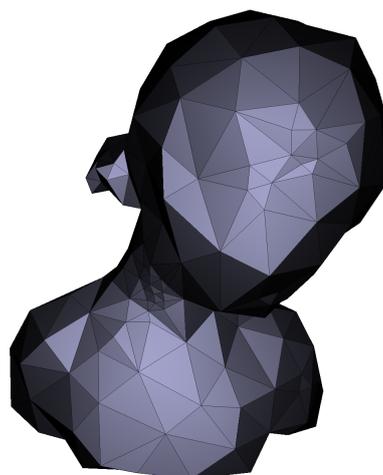
(d) 318 triângulos, DI,  $r=0.10$



(e) 332 triângulos, DC,  $r=0.10$



(f) 804 triângulos, FI,  $r=0.10$



(g) 530 triângulos, FC,  $r=0.10$

Figura 4.2: Diferentes simplificações do modelo *Bimba con Nastrino*.

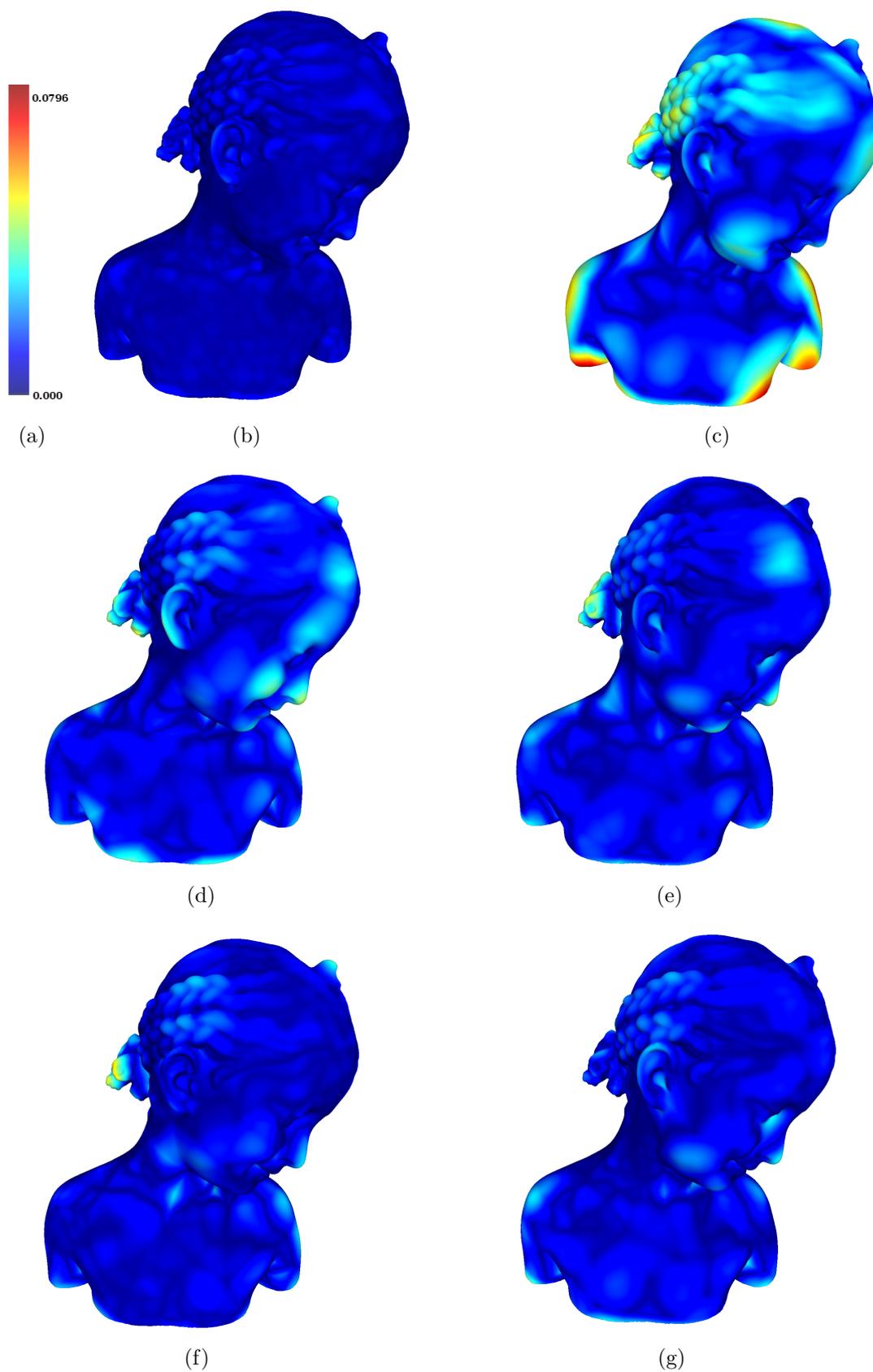
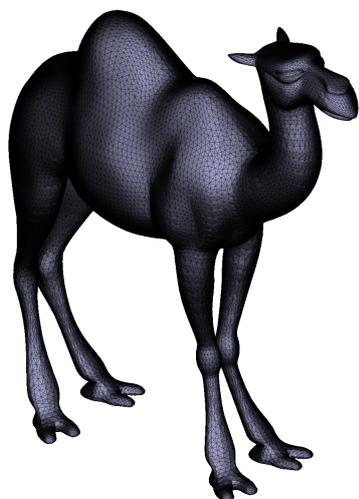


Figura 4.3: Colorizações através da distância de Hausdorff da malha *Bimba con Nastrino*, com as malhas simplificadas exibidas na figura 4.2.



(a) Malha original com 48.614 triângulos



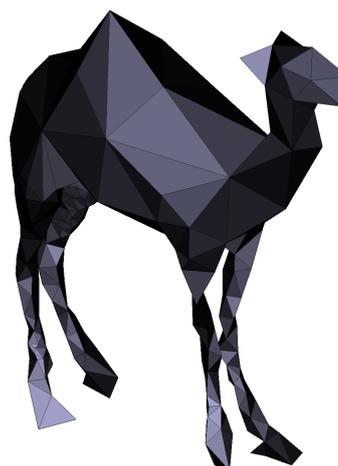
(b) Malha simplificada com 1128 triângulos, DI,  $r=0.05$



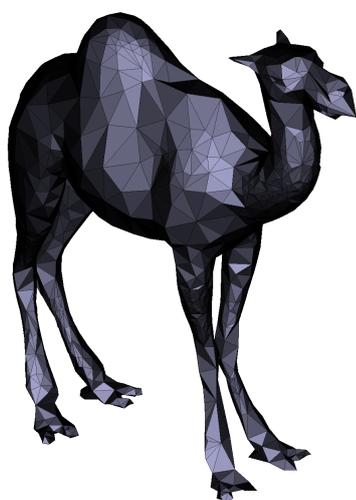
(c) Malha simplificada com 578 triângulos, DD,  $r=0.15$



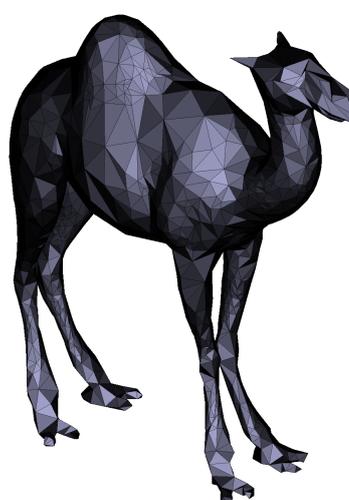
(d) 662 triângulos, DI,  $r=0.15$



(e) 718 triângulos, DC,  $r=0.15$



(f) 3482 triângulos, FI,  $r=0.15$



(g) 4016 triângulos, FC,  $r=0.15$

Figura 4.4: Diferentes simplificações do modelo *Camel*.

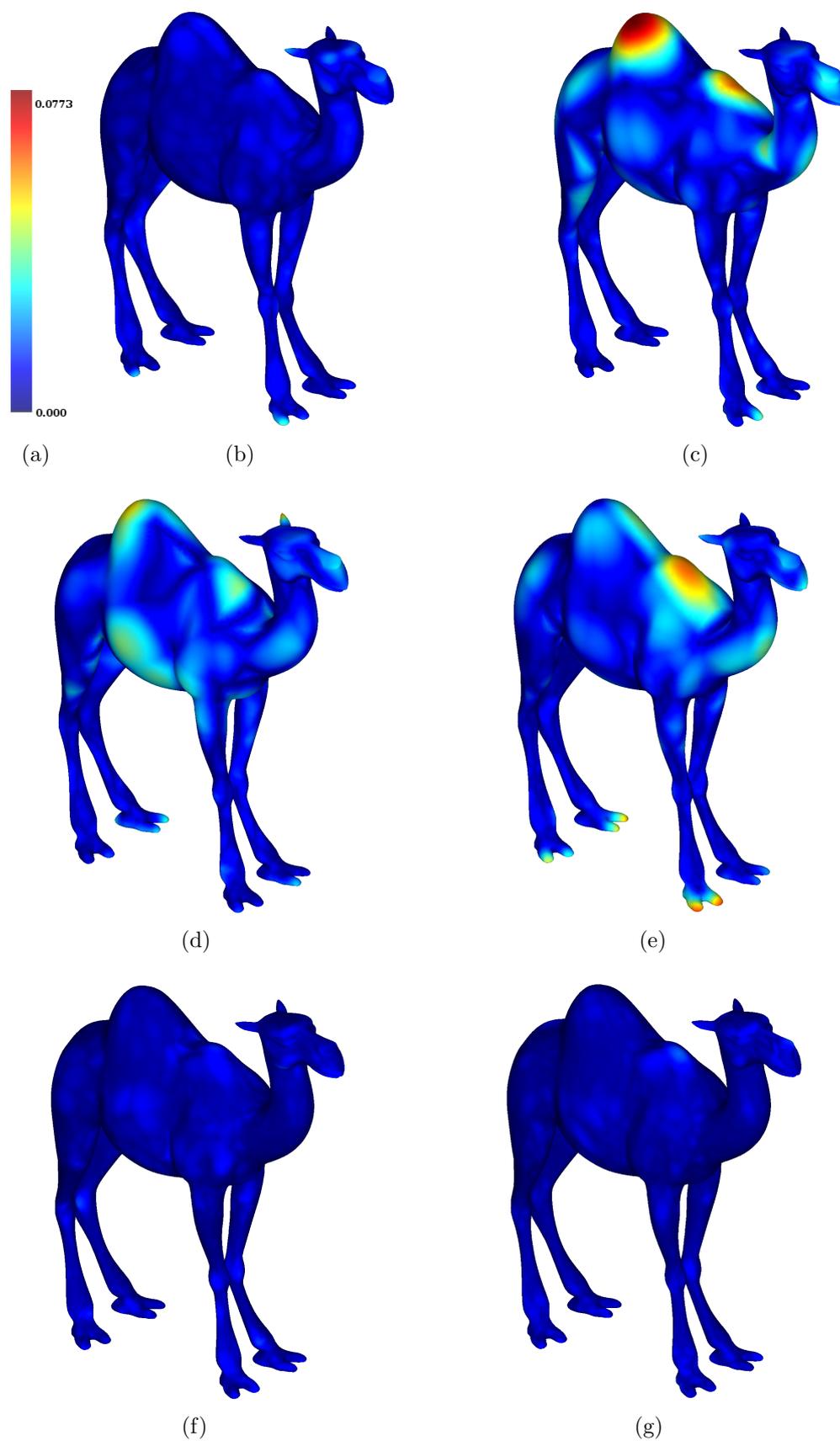


Figura 4.5: Colorizações através da distância de Hausdorff da malha *Camel*, com as malhas simplificadas exibidas na figura 4.4.

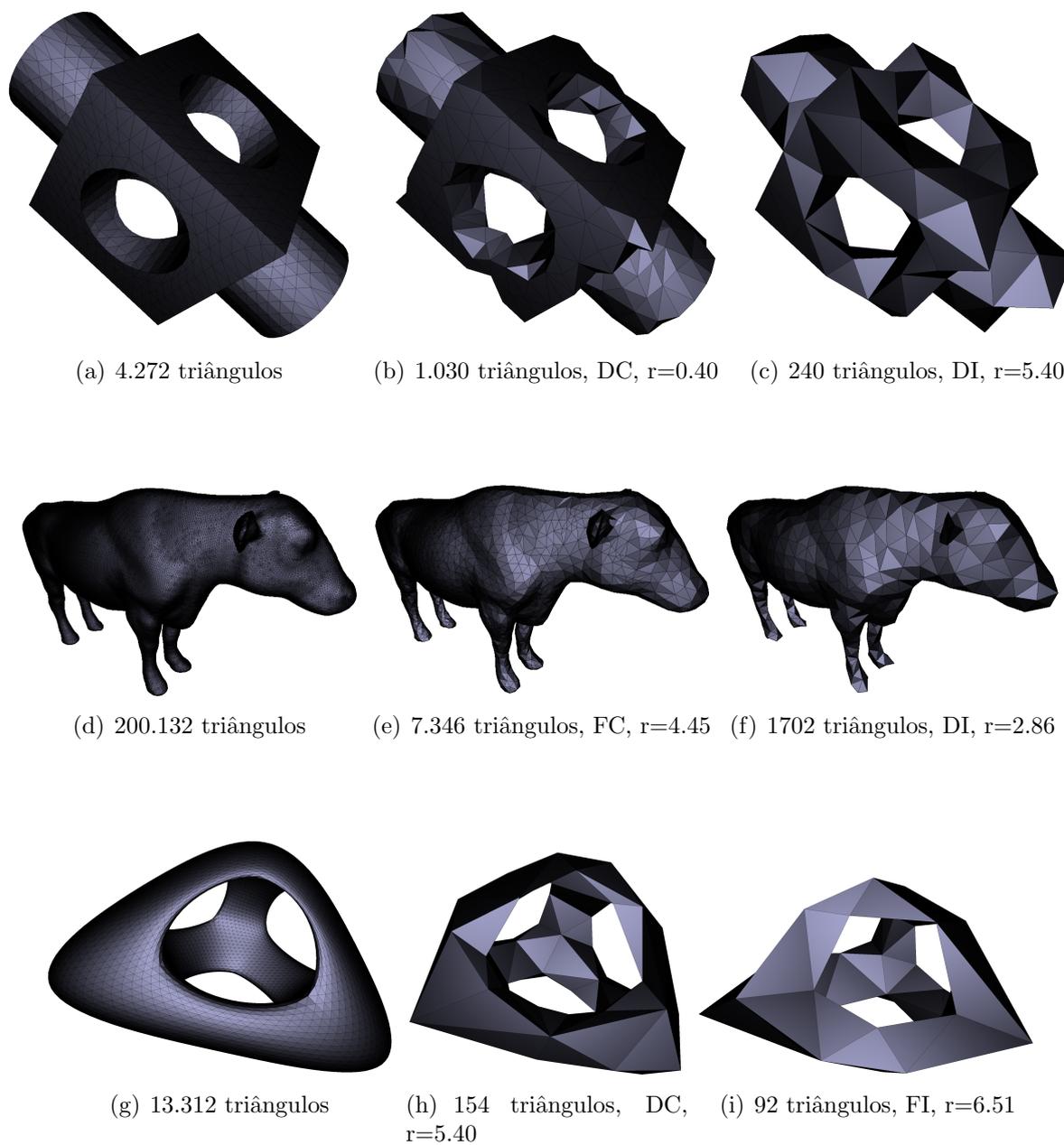


Figura 4.6: Alguns resultados mostrando as malhas originais e diferentes simplificações.

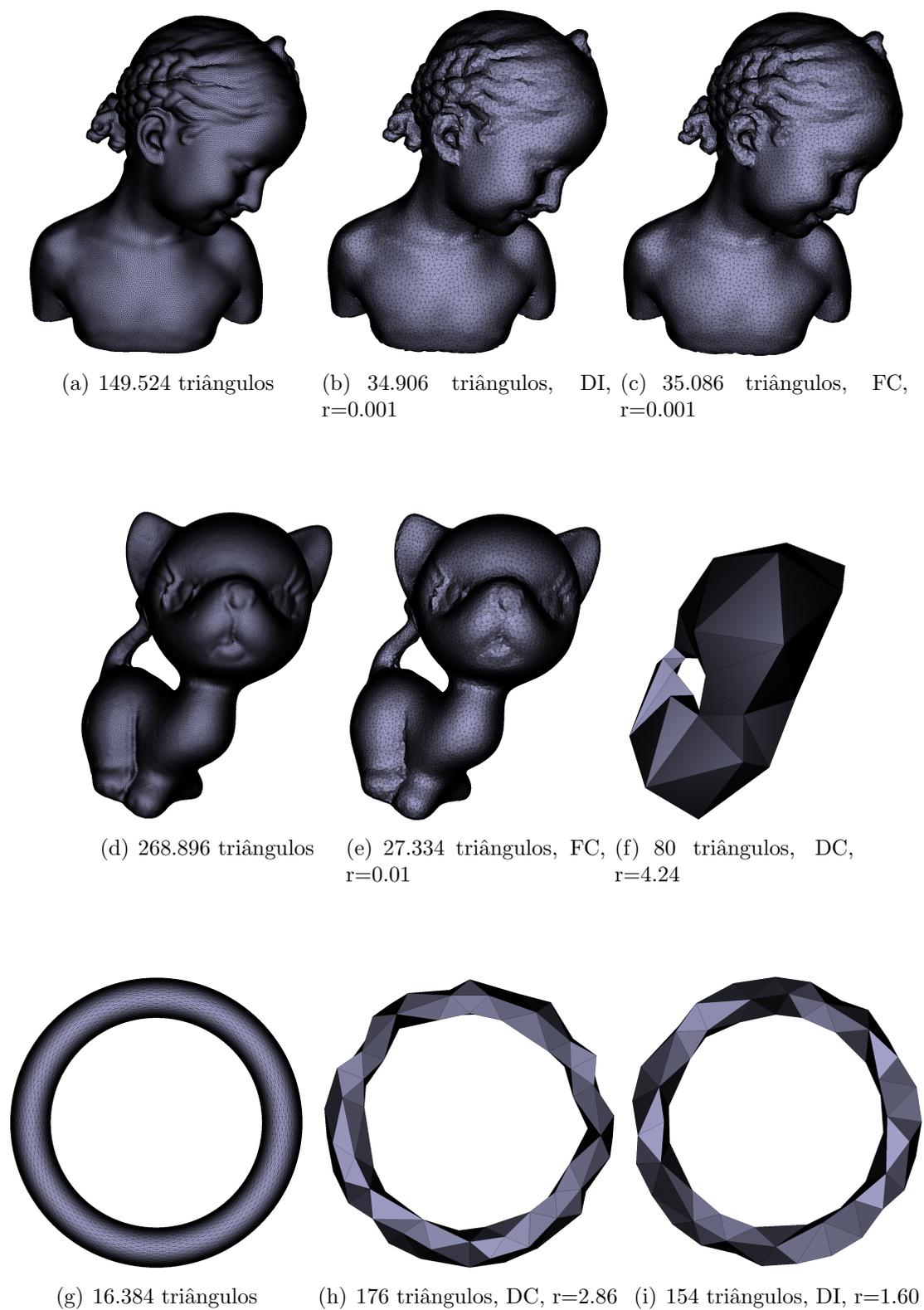
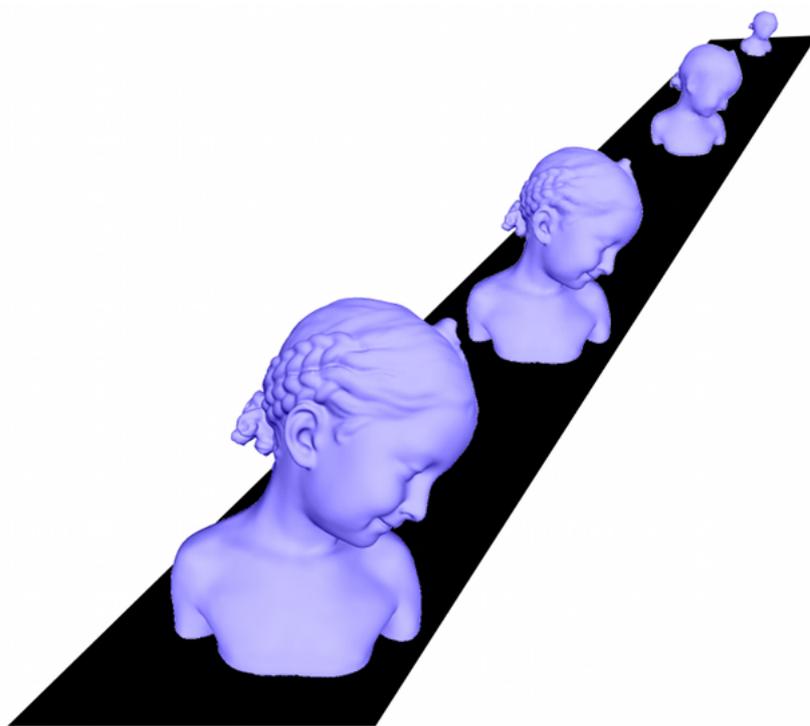


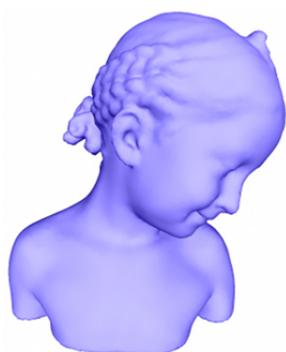
Figura 4.7: Mais alguns resultados mostrando as malhas originais e diferentes simplificações.



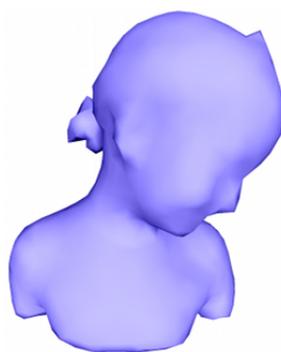
(a)



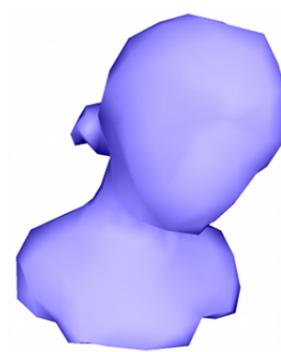
(b) 149.524 triângulos



(c) 35.086 triângulos



(d) 2056 triângulos



(e) 530 triângulos

Figura 4.8: Um exemplo de utilização de diferentes níveis de detalhes.

## 5. CONSIDERAÇÕES FINAIS

Neste trabalho foi discutido o processo de simplificação de malhas através de uma classificação das principais características que podem definir cada processo. Foram também apresentadas algumas abordagens clássicas da literatura.

Adicionalmente, foi apresentada uma adaptação para malhas triangulares densas do método proposto por Peixoto [2002]. O processo ocorre definindo uma cobertura sobre a malha para selecionar os vértices da malha simplificada. Na etapa de geração dos discos foram utilizadas duas métricas, Dijkstra e *Fast Marching*. Além disso, foram utilizadas três estratégias de distribuição dos discos, incremental, baseada em curvatura e ponto mais distante. A partir disso, foi definido um Diagrama de Voronoi intrínseco, de maneira que, as células sejam bem formadas [Dyer et al., 2008]. Isso é uma condição necessária para que a malha simplificada seja uma triangulação topologicamente equivalente à malha original [Edelsbrunner and Shah, 1994; Boubekur and Alexa, 2009]. A malha simplificada é então gerada como dual à cobertura de Voronoi e orientada através da malha original.

Como foi observado nos resultados 4.7(c) e 4.7(f), é possível obter tanto simplificações pequenas quanto expressivas, mas ainda assim preservando a topologia da superfície. Ao longo do trabalho, a adaptação à geometria ocorreu apenas no critério de seleção dos pontos, ao passo que, o controle da topologia constitui uma parte fundamental do processo. Naturalmente, uma extensão é adaptar a cobertura de discos em regiões de interesse, como altas curvaturas.

Uma vez que a mudança da métrica alterou na quantidade de discos, e a mudança de estratégia de seleção de pontos mudou o erro entre a malha original e simplificada, a escolha entre quais critérios utilizar para uma simplificação específica varia muito com a malha inicial e com o objetivo. Como pode ser visto nos resultados 4.4(e) e 4.4(g), utilizar *Fast Marching* e seleção por curvatura preservou bem regiões de alta curvatura, porém não obteve necessariamente resultados perceptuais melhores, o que pode ser observado nas figuras 4.4(d) e 4.4(f).

Um aspecto a ser investigado está na razão de aspecto dos triângulos. Nos resultados que utilizaram Dijkstra como métrica, observam-se triângulos com melhores razões de aspecto se comparado ao uso do *Fast Marching*.

Como foi visto no resultado 4.4(c), caso o objetivo seja obter a menor quantidade de triângulos para um raio fixo, a estratégia mais adequada é a de ponto mais distante para gerar a cobertura.

A escolha manual do raio a ser utilizado pode ser um problema, uma vez que é um critério indireto e que, para um mesmo raio e diferentes métricas, há uma boa diferença nas simplificações, como pode ser visto nos resultados 4.4(c) e 4.4(f).

Como trabalhos futuros, pretendemos:

- Investigar a definição automática do raio através de critérios quantitativos, por exemplo, a curvatura, com o objetivo de adaptar a geometria da malha simplificada em determinadas regiões,
- Utilizar a curvatura média para selecionar os pontos com o objetivo de preservar vértices em regiões onde a curvatura gaussiana se anula,
- Controlar a razão de aspecto dos triângulos da malha simplificada através da restrições ao raio de células adjacentes,
- Otimizar a implementação e liberá-la sobre alguma licença de software livre para comparação com outros métodos da literatura.

# REFERÊNCIAS

- Akenine-Moller, T., Moller, T. and Haines, E. [2002], *Real-Time Rendering*, 2nd edn, A. K. Peters Ltd.
- Alhichri, H. S. and Kamel, M. [2002], ‘Multi-resolution image registration using multi-class hausdorff fraction’, *Pattern Recognition Letters* **23**, 279–286.  
**URL:** <http://portal.acm.org/citation.cfm?id=633720.633747>
- Alliez, P., Ucelli, G., Gotsman, C. and Attene, M. [2008], *Recent Advances in Remeshing of Surfaces*, Mathematics and Visualization, Springer Berlin Heidelberg.
- Boaventura, P. O. [2006], *Grafos: Teoria, Modelos, Algoritmos*, Edgard Blücher.
- Botsch, M., Pauly, M., Kobbelt, L., Alliez, P., Lévy, B., Bischoff, S. and Rössl, C. [2007], Geometric modeling based on polygonal meshes, in ‘SIGGRAPH 2007 courses’.  
**URL:** <http://doi.acm.org/10.1145/1281500.1281640>
- Boubekur, T. and Alexa, M. [2009], ‘Mesh simplification by stochastic sampling and topological clustering’, *Computers and Graphics* **33**(3), 241 – 249. IEEE International Conference on Shape Modelling and Applications.  
**URL:** <http://www.sciencedirect.com/science/article/B6TYG-4VVXSVW-1/2/4ac6aee627a6347bb8379c44260ef12d>
- Cheng, W. [2008], Streaming of 3d progressive meshes, in ‘International Conference on Multimedia’, pp. 1047–1050.  
**URL:** <http://doi.acm.org/10.1145/1459359.1459570>
- Cignoni, P., Montani, C. and Scopigno, R. [1997], ‘A comparison of mesh simplification algorithms’, *Computers and Graphics* **22**, 37–54.
- Cignoni, P., Rocchini, C. and Scopigno, R. [1998], ‘Metro: Measuring error on simplified surfaces’, *Computer Graphics Forum* **17**(2), 167–174.

- Clark, J. H. [1976], ‘Hierarchical geometric models for visible surface algorithms’, *Communication of the ACM* **19**(10), 547–554.
- Cohen, J., Varshney, A., Manocha, D., Turk, G., Weber, H., Agarwal, P., Brooks, F. and Wright, W. [1996], Simplification envelopes, in ‘SIGGRAPH’, ACM, pp. 119–128.  
**URL:** <http://doi.acm.org/10.1145/237170.237220>
- Date, H., Kanai, S., Kishinami, T. and Nishigaki, I. [2005], Mesh simplification and adaptive LOD for finite element mesh generation, in ‘International Conference on Computer Aided Design and Computer Graphics’, IEEE, pp. 339–344.  
**URL:** <http://dx.doi.org/10.1109/CAD-CG.2005.56>
- de Berg, M., Cheong, O., van Kreveld, M. and Overmars, M. [2008], *Computational Geometry: Algorithms and Applications*, 3rd edn, Springer.  
**URL:** <http://www.worldcat.org/isbn/3540779736>
- Dijkstra, E. W. [1959], ‘A note on two problems in connexion with graphs’, *Numerische Mathematik* **1**(1), 269–271.  
**URL:** <http://dx.doi.org/10.1007/BF01386390>
- Dyer, R. [2010], Self-Delaunay meshes for surfaces, PhD thesis, Simon Fraser University, Burnaby, Canada.
- Dyer, R., Zhang, H. and Möller, T. [2008], ‘Surface sampling and the intrinsic Voronoi diagram’, *Computer Graphics Forum* **27**(5), 1393–1402.  
**URL:** <http://dx.doi.org/10.1111/j.1467-8659.2008.01279.x>
- Eck, M., DeRose, T., Duchamp, T., Hoppe, H., Lounsbery, M. and Stuetzle, W. [1995], Multiresolution analysis of arbitrary meshes, in ‘SIGGRAPH’, ACM, pp. 173–182.  
**URL:** <http://doi.acm.org/10.1145/218380.218440>
- Edelsbrunner, H. and Shah, N. R. [1994], Triangulating topological spaces, in ‘Symposium on Computational geometry’, ACM, pp. 285–292.  
**URL:** <http://doi.acm.org/10.1145/177424.178010>
- Garland, M. and Heckbert, P. S. [1997], Surface simplification using quadric error metrics, in ‘SIGGRAPH’, ACM, pp. 209–216.  
**URL:** <http://dx.doi.org/10.1145/258734.258849>

Google [2010], ‘Google body’.

**URL:** <http://bodybrowser.googlelabs.com/> Último acesso em: 10/03/2011

Gotsman, C., Gumhold, S. and Kobbelt, L. [1998], Simplification and compression of 3d meshes, in ‘European Summer School on Principles of Multiresolution in Geometric Modelling (PRIMUS)’, Springer, pp. 319–361.

Gross, M., Staadt, O. and Gatti, R. [1996], ‘Efficient triangular surface approximation using wavelets and quadtree data structures’, *IEEE Transactions on Visualization and Computer Graphics* **2**(2), 130–143.

Heckbert, P. S. and Garland, M. [1997], ‘Survey of polygonal surface simplification algorithms’, Multiresolution Surface Modeling Course - SIGGRAPH.

Hoppe, H. [1996], Progressive meshes, in ‘SIGGRAPH’, ACM, pp. 99–108.

**URL:** <http://doi.acm.org/10.1145/237170.237216>

Hoppe, H. [1997], View-dependent refinement of progressive meshes, in ‘SIGGRAPH’, ACM, pp. 189–198.

**URL:** <http://dx.doi.org/10.1145/258734.258843>

Hoppe, H., DeRose, T., Duchamp, T., McDonald, J. and Stuetzle, W. [1993], Mesh optimization, in ‘SIGGRAPH’, ACM, pp. 19–26.

**URL:** <http://doi.acm.org/10.1145/166117.166119>

Hormann, K., Labsik, U. and Greiner, G. [2001], ‘Remeshing triangulated surfaces with optimal parameterizations’, *Computer-Aided Design* **33**(11), 779 – 788.

**URL:** <http://www.sciencedirect.com/science/article/B6TYR-452V6SP-4/2/2e5057daf126282316b2d63266bea161>

Huttenlocher, D. and Rucklidge, W. [1993], A multi-resolution technique for comparing images using the hausdorff distance, in ‘Computer Vision and Pattern Recognition’, pp. 705 –706.

Kalvin, A. D. and Taylor, R. H. [1996], ‘Superfaces: Polygonal mesh simplification with bounded error’, *IEEE Computer Graphics and Applications* **16**, 64–77.

Kimmel, R. and Sethian, J. [1998], ‘Computing geodesic paths on manifolds’, *Proceedings of the National Academy of Sciences of the United States of America* **95**(15), 8431–8435.

**URL:** <http://www.pnas.org/content/95/15/8431.abstract>

- Kobbelt, L. and Botsch, M. [2000], ‘An interactive approach to point cloud triangulation’, *Computer Graphics Forum* **19**(3), 479–487.  
**URL:** <http://dx.doi.org/10.1111/1467-8659.00440>
- Lewiner, T., Lopes, H., Vieira, A. W. and Tavares, G. [2003], ‘Efficient implementation of Marching Cubes cases with topological guarantees’, *Journal of Graphics Tools* **8**(2), 1–15.  
**URL:** [http://thomas.lewiner.org/pdfs/marching\\_cubes\\_jgt.pdf](http://thomas.lewiner.org/pdfs/marching_cubes_jgt.pdf)
- Lorensen, W. E. and Cline, H. E. [1987], ‘Marching cubes: A high resolution 3d surface construction algorithm’, *SIGGRAPH Comput. Graph.* **21**(4), 163–169.
- Luebke, D. [1996], Hierarchical structures for dynamic polygonal simplification, Technical report.
- Luebke, D., Reddy, M., Cohen, J. D., Varshney, A., Watson, B. and Huebner, R. [2003], *Level of Detail for 3D Graphics*, Morgan Kaufman.
- Meyer, M., Desbrun, M., Schröder, P. and Barr, A. H. [2002], ‘Discrete Differential-Geometry Operators for Triangulated 2-Manifolds’.  
**URL:** <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.24.3427>
- Morera, D. M. [2006], Geodesic-based modeling on manifold triangulations, PhD thesis, IMPA.
- Munkres, J. [1988], *Topology: A First course*, Prantice Hall.
- O’Rourke, J. [1998], *Computational Geometry in C*, 2nd edn, Cambridge University Press.  
**URL:** <http://cs.smith.edu/orourke/books/compgeom.html>
- Peixoto, A. [2002], Extração de malhas adaptativas em Multi-resolução a partir de volumes usando simplificação e refinamento, PhD thesis, Pontifícia Universidade Católica - Rio de Janeiro.
- Rossignac, J. and Borrel, P. [1992], Multi-resolution 3d approximations for rendering complex scenes, Technical report, IBM.
- Schroeder, W., Zarge, J. and Lorensen, W. [1992], ‘Decimation of triangle meshes’, *SIGGRAPH* **26**, 65–70.  
**URL:** <http://doi.acm.org/10.1145/142920.134010>

- 
- Sethian, J. [1999], *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, Cambridge University Press.
- Spira, A. and Kimmel, R. [2003], An efficient solution to the Eikonal equation on parametric manifolds, *in* ‘INTERPHASE 2003’, Vol. 3, pp. 315–327.  
**URL:** <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.3.5112>
- Turk, G. [1992], ‘Re-tiling polygonal surfaces’, *SIGGRAPH* **26**, 55–64.  
**URL:** <http://doi.acm.org/10.1145/142920.134008>
- Vieira, A. W., Lewiner, T., Velho, L., Lopes, H. and Tavares, G. [2004], ‘Stellar mesh simplification using probabilistic optimization’, *Computer Graphics Forum* **23**(4), 825–838.  
**URL:** [http://thomas.lewiner.org/pdfs/fast\\_stellar\\_cgf.pdf](http://thomas.lewiner.org/pdfs/fast_stellar_cgf.pdf)