

UNIVERSIDADE FEDERAL DE ALAGOAS

INSTITUTO DE MATEMÁTICA

PROGRAMA DE PÓS GRADUAÇÃO EM MATEMÁTICA

Diego Chicuta Macedo

SIMPLIFICAÇÃO DE MALHAS TRIANGULARES BASEADA EM TEMPLATES

Maceió

2015

DIEGO CHICUTA MACEDO

SIMPLIFICAÇÃO DE MALHAS TRIANGULARES BASEADA EM TEMPLATES

Dissertação de Mestrado submetida em Outubro de 2015 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

Catálogo na fonte
Universidade Federal de Alagoas
Biblioteca Central
Divisão de Tratamento Técnico
Bibliotecário Responsável: Valter dos Santos Andrade

M141s Macedo, Diego Chicuta.
Simplificação de malhas triangulares baseada em templates / Diego Chicuta
Macedo. - 2015.
61 f. : il.

Orientador: Dimas Martínez Morera.
Coorientador: Thales Miranda de Almeida Vieira.
Dissertação (Mestrado em Matemática) – Universidade Federal de Alagoas.
Instituto de Matemática. Maceió, 2015.

Bibliografia: f. 59-60.

1. Geometria. 2. Simplificação de malhas. 3. Topologia. 4. Distância geodésica. 5. Principal Components Analysis (PCA). 6. Fast Marching Method (FMM). I. Título.

CDU: 514.772.2:004.92

DIEGO CHICUTA MACEDO

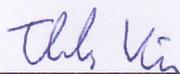
SIMPLIFICAÇÃO DE MALHAS TRIANGULARES BASEADA EM TEMPLATES

Dissertação de Mestrado submetida em Outubro de 2015 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.

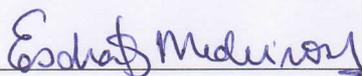
Banca Examinadora:



Prof. Dr. Dimas Martínez Morera
Orientador
UFAL



Prof. Dr. Thales Miranda de Almeida Vieira
Co-Orientador
UFAL



Prof. Dr. Esdras Soares de Medeiros Filho
UFC

Aos meus pais, irmãs, noiva e avós.

Dedico este trabalho aos meus pais que nunca mediram esforços para que seus filhos evoluíssem em conhecimento e em espírito, minhas irmãs, minha noiva e meus avós que sempre me deram forças para continuar firme em minha caminhada.

AGRADECIMENTOS

Primeiramente e sobretudo à DEUS pela força, equilíbrio e determinação de todos os dias durante o desenvolvimento deste trabalho, que só assim consegui ter inspiração e motivação para continuar.

As minhas primeiras palavras de apreço tem que ir, sem qualquer dúvida, para os meus pais, irmãs, avós e noiva. Eles são a razão da minha existência, o meu porto seguro, a minha luz guia, fonte de inspiração e força. Sem eles não seria quem sou, nem estaria aqui hoje no final desta etapa. A eles devo tudo o que tenho e que venha a ter. Um agradecimento especial a toda a minha família pelo apoio.

Quero prestar agradecimento ao meu orientador professor Dimas Martínez pela supervisão prestada durante todas as fases deste trabalho, pelas reuniões de aconselhamento e verificação do trabalho e como guia na execução do trabalho de forma mais correta, bem como seu imenso incentivo e credibilidade diante deste trabalho e principalmente por sua grande paciência e compreensão que foram de fundamentais importância para que esse trabalho tivesse o objetivo alcançado, meus sinceros agradecimentos. Agradeço também ao professor Thales por suas sugestões, verificações, conselhos e discussões que engrandeceram bastante este trabalho.

Uma palavra de agradecimento à todos os docentes do programa, em especial ao professor Adelailson por seu grande companheirismo e amizade em todos os meus momentos aqui. Agradeço também à todos que fazem a coordenação do programa como Ewerton e Ana que me ajudaram bastante com esclarecimentos e todo o apoio e acompanhamento de minha vida acadêmica junto ao departamento. Agradeço aos colegas de minha turma, do IM, do CPMAT que me ajudaram na execução do trabalho, me deram apoio, idéias e ajuda técnica, pelos bons e por muitas vezes descontraídos momentos do café, que foram muito importantes em todos os momentos.

Não posso deixar de agradecer aos meus amigos. Aos do laboratório e da turma em especial, Tiago, Alexandre, Ícaro, Michel, Fabrício, Ailton, Nayane, Rogério e Robson, que representam a minha “família da UFAL”. Os amigos que me acompanharam ao longo de todo esse tempo, tanto os que passaram quanto os que permanecem aqui, de trabalho árduo, de muitas horas de estudo, de muitos risos e bons momentos e partilha de todas as experiências vividas no mestrado ao longo destes anos. E aos demais amigos que passaram pela minha vida acadêmica e pessoal.

Minha gratidão a todos aqueles que estiveram ao meu lado. A ajuda dessas pessoas sempre foi de fundamental importância.

A todos, muito obrigado.

RESUMO

Algoritmos para simplificação de malhas podem ser aplicados de forma a eliminar vértices da malha resultante, sem prejudicar a topologia da mesma. Neste trabalho desenvolvemos um técnica de simplificação de malhas 3D baseada em *templates*, que reduz a sua quantidade de vértices. Apresentaremos um método que busca otimizar a complexidade de uma malha a partir da redução no seu número de vértices e faces, de forma que a geometria e topologia da superfície sejam preservadas. Isso é garantido com o uso de uma malha base, ou *template*, que tem gênero zero, topologia e conectividade desejada. Nosso objetivo é definir a geometria do *template*, de modo que aproxime nossa superfície original. Usaremos *Análise de Componentes Principais* (PCA) para o alinhamento inicial das duas superfícies. Este alinhamento busca encaixar as duas malhas de forma que tenhamos um alinhamento de suas direções principais, o que permite a devida geração de correspondência entre seus vértices. Com o intuito de melhorar a correspondência entre os vértices de ambas as malhas, adicionamos a informação da *distância geodésica* de cada vértice à dos pontos fixos da malha densa, usando o *Fast Marching* (FMM). Isso nos permite discriminar vértices distantes na malha, mas próximos na distância euclidiana. Para a geração da nova malha que é menos densa que a inicial usaremos uma técnica baseada em projeções que visa trocar a posição de vértices de uma malha densa por suas respectivas correspondências no *template* que possui quantidade de vértices menor. Como resultado destas projeções teremos uma malha resultante muito menos densa que a inicial, que preserva as características geométricas e topológicas.

Palavras-Chave: Simplificações. Geometria. Topologia. Template. Distância Geodésica.

ABSTRACT

Algorithms for mesh simplification can be applied to remove vertices avoiding changes to the topology of the object. In this work we have developed a technique for simplification of 3D meshes based on templates. We present a method that tries to optimize the complexity of a mesh decreasing the number of vertices and faces, so that the geometry and topology of the surface are preserved. Our goal is to define the geometry of the template so that it approximates our original surface. We will use the *Principal Components Analysis* (PCA) to achieve an initial alignment of the two surfaces. This alignment searches a fit between the two meshes so that we have an alignment of its main directions, which allows proper generation of correspondence between its vertices. With the aim of improving the correspondence between the vertices of both meshes, we added the geodesic distance information of each vertex to the of fixed points of dense mesh, using *Fast Marching* (FMM). This allows us to discriminate vertices distant in the mesh, but next in the distance euclidean. For the generation of the new mesh that is less dense than the initial we will use a technique based on projections that aims to change the position of the vertices of a dense mesh by their respective correspondences in the template that has quantity of vertices less. As a result of these projections we will have a mesh resulting much less dense than the initial, which preserves the geometrical and topological properties.

Keywords: Simplifications. Geometry. Topology. Template. Geodesic Distance.

LISTA DE FIGURAS

1.1	Simplificação	p. 13
1.2	Classificação dos Vértices	p. 14
1.3	Mesh Decimation	p. 15
1.4	Movimentos Legais	p. 17
1.5	Transformação de Collapse	p. 18
1.6	Elipsóides de Erro	p. 18
1.7	Colapso de Aresta	p. 19
1.8	Junção de Vértices	p. 19
2.1	Superfície	p. 21
2.2	Malha de uma Superfície	p. 22
2.3	Bitoro de Gênero 2	p. 23
2.4	Vizinhança Estrelada	p. 24
2.5	Terceira Vizinhança Estrelada	p. 24
3.1	Aplicação do PCA ao Bunny e a Mão	p. 30
3.2	Alinhamento	p. 31
4.1	Ilustração FMM	p. 32
4.2	Discretização do FMM	p. 33
4.3	Processo de Iteração	p. 34
4.4	Processo de Propagação	p. 35
4.5	FMM na Esfera e no Bunny	p. 36
4.6	FMM no Cavalo e no Armadillo	p. 36
5.1	Primeira Projeção para o Bunny	p. 39
5.2	Relação das Correspondências entre o Bunny e a Esfera	p. 41

5.3	Terceira Vizinhança para cada Vértice do Template	p. 42
5.4	Correção dos Vértices Distantes	p. 44
5.5	Processo de Subdivisão	p. 46
6.1	Simplificação do Bunny	p. 49
6.2	Simplificação de Vênus	p. 50
6.3	Simplificação de Buste	p. 51
6.4	Simplificação de Max	p. 52
6.5	RMSD Bunny	p. 55
6.6	RMSD Vênus	p. 55
6.7	RMSD Buste	p. 56
6.8	RMSD Max	p. 56
6.9	Colorização de Max	p. 58

SUMÁRIO

	Página
INTRODUÇÃO	p. 13
1.1 Objetivo	p. 13
1.2 Trabalhos Relacionados	p. 14
1.2.1 Mesh Decimation	p. 14
1.2.2 Re-Tiling Polygonal Surfaces	p. 16
1.2.3 Otimização de Energia	p. 16
1.2.4 Métrica de Erros por Quádricas	p. 18
1.3 Contribuições	p. 19
1.4 Visão Geral do Trabalho	p. 19
2 FUNDAMENTOS	p. 21
2.1 Superfícies	p. 21
2.2 Malhas	p. 22
2.3 Vizinhança Estrelada	p. 23
2.4 Transformações Tridimensionais	p. 24
2.4.1 Translação	p. 25
2.4.2 Rotação	p. 25
2.4.3 Escala	p. 26
3 ANÁLISE DE COMPONENTES PRINCIPAIS	p. 27
3.1 Componentes Principais	p. 27
3.1.1 Matriz de Covariância	p. 28

3.2	Análise de Componentes Principais - PCA	p. 29
3.2.1	Alinhamento	p. 30
4	FAST MARCHING	p. 32
4.1	Fast Marching	p. 32
4.2	Fast Marching na Malha	p. 36
5	SIMPLIFICAÇÃO DE MALHAS	p. 38
5.1	Projeção do Template	p. 38
6	RESULTADOS	p. 48
6.1	Análise dos Resultados	p. 53
7	CONCLUSÕES	p. 59
	REFERÊNCIAS	p. 60

INTRODUÇÃO

A modelagem de objetos 3D em computação gráfica é usualmente aplicada para o desenvolvimento de ferramentas cada vez mais eficazes e com baixo custo computacional, que visam ter uma boa performance de execução. Malhas triangulares são tipicamente a escolha adotada, dado a eficiência de algoritmos e seu tratamento especial em computadores. Entretanto, quando estas malhas são muito grandes, algoritmos para reduzir a complexidade e gerar malhas menos densas são necessários.

Neste trabalho desenvolvemos uma técnica de simplificação de malhas densas baseada em *templates* que preservam regiões perceptualmente interessantes da malha, do ponto de vista geométrico e topológico. Para isto, propomos uma técnica baseada em projeções de vértices entre uma malha densa, com muitos vértices, e uma malha grosseira ou menos densa, que chamamos de *template*, com menos vértices que a anterior. Tais projeções são realizadas a partir da criação de correspondências entre estas duas malhas, as quais são obtidas a partir da distância euclidiana entre cada ponto da malha densa ao seu mais próximo no *template*, depois de ter alinhado as duas malhas.

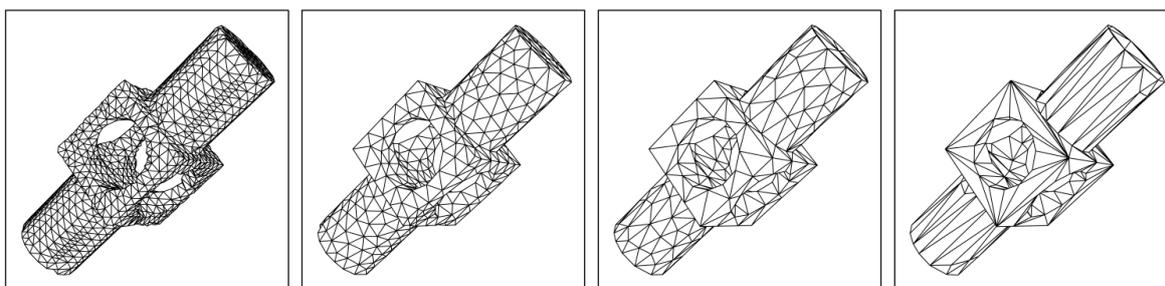


Figura 1.1: Simplificação

Fonte: Disponível em [\[Hoppe\]](#)

1.1 Objetivo

Neste trabalho apresentaremos um método capaz de simplificar uma malha a partir de projeções. Esse método deve manter a topologia e preservar ao máximo a geometria inicial da superfície. Para isto usaremos um *template*, que tem a mesma topologia da malha original, mas conectividade e tamanho desejados. Dessa forma nosso principal objetivo é definir a geometria do *template*, de modo que aproxime nossa superfície original.

1.2 Trabalhos Relacionados

A simplificação de malhas tridimensionais possibilita a obtenção de novas representações de uma superfície através de malhas que se caracterizam por uma quantidade de informação menor do que a malha inicialmente dada. De um modo geral, os trabalhos relacionados à simplificação de malhas triangulares que podemos encontrar em nossa literatura, buscam representar uma malha por uma equivalente que possui um número de vértices e faces menor.

Em nossa literatura a simplificação de uma malha é geralmente utilizada em compressão, velocidade de processamento bem como para melhorar sua qualidade. Este último caso por vezes é uma característica subjetiva, uma vez que depende única e exclusivamente de sua aplicação. A malha pode ser modificada em termos da sua quantidade de elementos bem como obter uma resultante que apresente poucos vértices, arestas e faces, mantendo a topologia e preservando a geometria original. A seguir apresentaremos algumas abordagens do problema de simplificação de malhas sob seus diferentes aspectos para a obtenção de uma malha resultante.

1.2.1 Mesh Decimation

A *decimação* de uma malha triangular consiste em eliminar de forma iterativa vértices, arestas e faces, que não satisfazem um determinado critério local e efetuar uma triangulação na região removida.

Incremental Decimation

Segundo [Schroeder] a decimação de uma malha segue os seguintes passos:

Passo 1: Os vértices da malha são classificados como *simplex*, *singular*, *de bordo*, *de aresta*, *interior* ou *de canto*.



Figura 1.2: Classificação dos Vértices

Fonte: Disponível em [Schroeder]

Passo 2: Nesta etapa são avaliados se os vértices candidatos satisfazem determinados critérios para a remoção. Assim para cada vértice simplex é calculado um plano médio π com normal N_π e com $x \in \pi$ tal que:

$$N_\pi = \frac{\sum N_i A_i}{\sum A_i}$$

e

$$x = \frac{\sum c_i A_i}{\sum A_i}$$

Onde N_i , A_i e c_i são respectivamente as normais, as áreas e os centróides de cada triângulo da vizinhança estrelada dos vértices simples.

Sendo assim, um vértice v será removido se $d(v, \pi) > \epsilon$.

Passo 3: Nesta etapa é realizada a triangulação nas regiões onde os vértices foram removidos.

Vertex Clustering

No trabalho de [Gotsman] é desenvolvida uma representação compacta de um modelo 3D que inclui:

- Métodos para reduzir a complexidade de uma malha por simplificação, reduzindo assim o número de vértices e faces na malha.
- Métodos para reamostrar a geometria na ordem de distribuição dos vértices.

A técnica tratada no referido trabalho descreve uma classe de algoritmos para transformar uma malha inicial M_i em uma malha M_f com uma quantidade menor de vértices, arestas e faces. O processo é usualmente controlado por critérios definidos pelo usuário que preservem as propriedades específicas iniciais da malha.

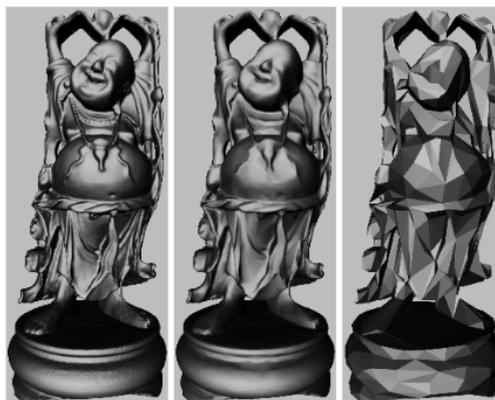


Figura 1.3: Mesh Decimation

Fonte: Disponível em [Gotsman]

A ideia básica é, para uma determinada tolerância ϵ é particionada uma região da malha em células que possuem diâmetro menor que ϵ . Com base na proximidade geométrica, para cada célula é determinado um vértice representante que é atribuído à todos os vértices que estão nesta célula. Nesta célula temos um *cluster* o qual é composto por vários vértices que tem uma relação de adjacência com o representante, daí o processo busca substituir os vértices deste *cluster* por um único vértice. No geral este é um algoritmo eficaz e robusto cuja complexidade é linear no número de vértice da malha, no entanto a qualidade da malha resultante não é satisfatória.

1.2.2 Re-Tiling Polygonal Surfaces

Neste método de simplificação descrito por [Turk] a entrada do algoritmo é a 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.

Feito isto, os vértices adicionados estarão fixados na malha. Daí é gerada uma nova malha através da adição destes vértices, e em seguida os vértices da malha original são removidos e uma nova triangulação é refeita obtendo assim a malha simplificada.

1.2.3 Otimização de Energia

Este método de simplificação de malhas visa realizar operações de remoção ou contrações locais com o objetivo simplificar a malha através de uma função de energia. Neste sentido a abordagem de otimização da malha define uma função de energia que mede a qualidade da aproximação da malha inicial à resultante.

Mesh Optimization

Um trabalho que traz uma outra abordagem para o problema de simplificação é baseado na *minimização da função energia da malha* [Hoppe]. Dada uma malha inicial M_0 e um conjunto de vértices V_0 , queremos encontrar uma malha M que aproxime V_0 com a menor quantidade de vértices e seja topologicamente equivalente à M_0 , através da minimização da função energia:

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

Onde:

- E_{dist} : Soma das distâncias dos pontos originais à malha, que controla a posição dos vértices.
- E_{rep} : Fator proporcional ao número de vértices à malha, que controla a quantidade de vértices.
- E_{spring} : Soma dos comprimentos das arestas, para garantir a convergência.

A otimização é iterativa obtida realizando movimentos *legais* que são movimentos que não alteram a topologia da malha, como *edge collapse*, *edge swap* e *edge split*. Estes movimentos legais são conduzidos por um processo de otimização da função de energia. Em cada etapa, o vértice cuja eliminação causa o menor aumento da função de energia é suprimido.

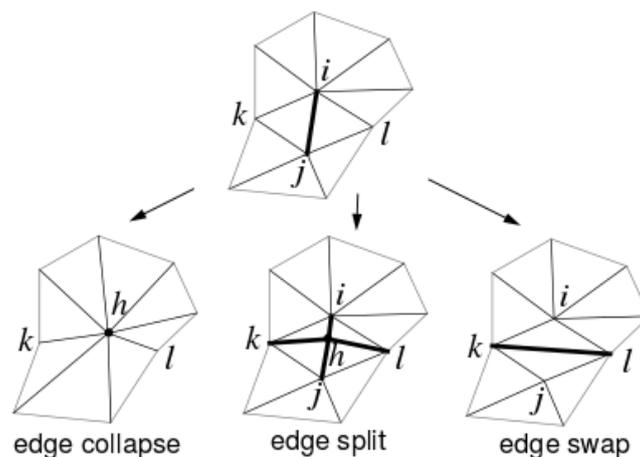


Figura 1.4: Movimentos Legais

Fonte: Disponível em [\[Hoppe\]](#)

O processo de minimização da energia ocorre de forma iterativa através da otimização da topologia e geometria da malha. Em cada etapa é gerado um complexo simplicial K através de uma operação legal. Para K fixo, a posição dos vértices que minimiza o erro ϵ é determinada, definindo assim uma malha M . Assim, se $E(M)$ é menor que o erro da malha anterior, M passa a ser a nova malha. A simplificação é obtida quando algum critério de convergência seja satisfeito.

Progressive Meshes

Este método [\[Hugges\]](#) também segue a linha do trabalho descrito anteriormente. Dada uma malha inicial M_0 , a malha simplificada M_r é obtida através de um conjunto de operações de *colapso* de arestas denominadas por *ecol*.

Aqui apenas a operação de *edge collapse*, e sua inversa, são consideradas para navegar sobre diferentes níveis de representação de detalhes da malha.

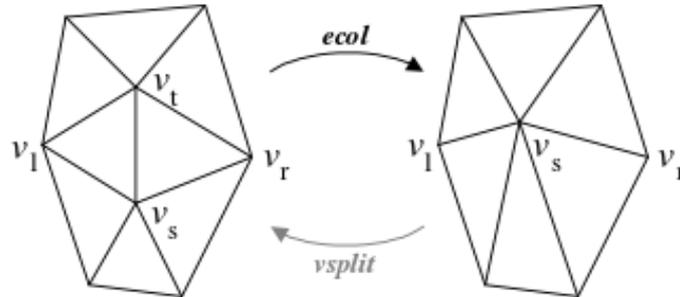


Figura 1.5: Transformação de Collapse

Fonte: Disponível em [Hugges]

1.2.4 Métrica de Erros por Quádricas

A métrica de erros por quádricas descrito por [Garland] 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. Dado 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 à T_i . O funcional $E(x) = \sum (n_i^T x - p_i)^2$ define o erro no vértice e seus isocontornos são elipsóides. Na figura abaixo, é ilustrado o erro associado a cada vértice:



Figura 1.6: Elipsóides de Erro

Fonte: Disponível em [Garland]

Dessa forma, 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, como mostra a figura 1.7.

Caso se deseje uma maior simplificação, deve-se considerar vértices que não são adjacentes para junção, figura 1.8. Uma possível escolha é tomar vértices e planos em uma determinada região do espaço. Isso permite alterar a topologia da malha simplificada.

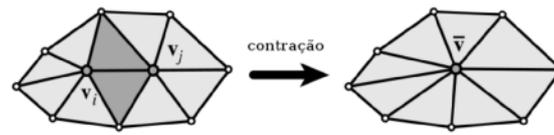


Figura 1.7: Colapso de Aresta

Fonte: Disponível em [Garland]

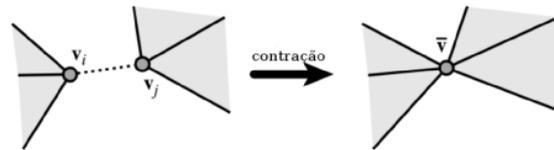


Figura 1.8: Junção de Vértices

Fonte: Disponível em [Garland]

1.3 Contribuições

O trabalho aqui apresentado, mostra uma nova forma de simplificar malhas a partir de *templates*, diferenciando-se um pouco dos trabalhos relacionados que buscam aplicar operações e movimentos legais aos componentes da malha, tais como usando *swap*, *collapse* e *split* em uma malha e minimizando os termos de energia, resultando assim em uma malha simplificada.

A principal contribuição deste trabalho radica na obtenção de uma malha simplificada com propriedades topológicas definidas a priori em uma malha base que tem a função de um *template*.

1.4 Visão Geral do Trabalho

Em nosso trabalho, a solução do problema é encontrada de modo iterativo. Inicialmente temos duas malhas M_d e M_0 de forma que a primeira é densa, e fixa ao longo de todo o processo, e a segunda a do *template*. O algoritmo inicia com estas duas malhas, primeiramente elas são alinhadas, de forma que suas direções principais coincidam. Após o alinhamento, criamos uma correspondência entre cada vértice da malha densa com o mais próximo dele no *template*, segundo a distância euclidiana. Dessa forma são gerados *clusters* de vértices da malha densa, cada um correspondente a um vértice do *template*.

No final deste processo, percebemos que há vértices cujos *clusters* possuem duas componentes conexas, e para estes vértices realizamos um procedimento de identificação e correção. Após corrigirmos tais vértices, aplicamos o processo de subdivisão das arestas do *template* cujo tamanho ficou muito grande após a projeção. Estas arestas grandes aparecem em áreas de

concauidade da malha.

Com isso temos o fim do primeiro passo da iteração, que gera uma malha M_r que é geométrica e topologicamente mais próxima de M_d . Daí o processo de iteração é repetido sempre entre a nova malha gerada M_r , com a malha fixa M_d . E ao fim de toda a iteração temos como resultado a simplificação de M_d . O processo é finalizado quando encontramos uma malha resultante da simplificação que é o mais próximo possível da malha densa inicial, do ponto de vista topológico e geométrico.

No segundo capítulo descreveremos as principais características de uma malha do ponto de vista topológico e geométrico, bem como trataremos das transformações geométricas e topológicas em uma malha. No terceiro capítulo abordaremos a *Análise de Componentes Principais* e sua aplicação para o alinhamento das malhas. No quarto capítulo tratamos do *Fast Marching Method*. No quinto capítulo descreveremos o algoritmo de simplificação para reduzir o número de elementos na malha através das projeções, e no sexto capítulo apresentaremos alguns resultados bem como algumas análises.

2 FUNDAMENTOS

Neste capítulo apresentamos as definições e conceitos matemáticos que estão por trás das técnicas desenvolvidas neste trabalho.

2.1 Superfícies

Definição 1. Um subconjunto $S \subset \mathbb{R}^3$ é uma superfície regular se, para cada $p \in S$, existe uma vizinhança V de p em \mathbb{R}^3 e uma aplicação $\mathbf{x} : U \rightarrow V \cap S$ de um aberto $U \subset \mathbb{R}^2$ sobre $V \cap S \subset \mathbb{R}^3$ tal que:

1. \mathbf{x} é diferenciável. Isto significa que se escrevemos

$$\mathbf{x}(\mathbf{u}, \mathbf{v}) = (x(u, v), y(u, v), z(u, v))$$

com $(u, v) \in U$, as funções $x(u, v)$, $y(u, v)$ e $z(u, v)$ têm derivadas parciais contínuas de todas as ordens em U .

2. \mathbf{x} é um homeomorfismo. Como \mathbf{x} é contínua pela condição 1, isto significa que \mathbf{x} tem inversa $\mathbf{x}^{-1} : V \cap S \rightarrow U$ que é contínua.
3. Para todo $q \in U$, a diferencial $d\mathbf{x}_q : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ é injetiva

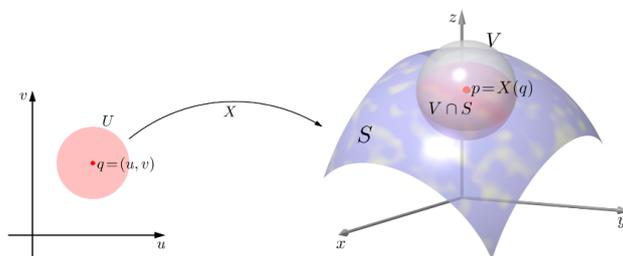


Figura 2.1: Superfície

Fonte: Disponível em http://www.professores.uff.br/katia_frensel/aulasgeodif/gdif.pdf

2.2 Malhas

Uma malha triangular de uma superfície, é formada por três conjuntos: *vértices*, *arestas* e *faces*. A informação que descreve os elementos da malha consiste na *topologia* e *geometria* da malha. A topologia da malha fornece as relações de incidência entre seus elementos. Estas relações de incidência especificam para cada face os vértices e arestas que à contém, para cada aresta os vértices que estas incidem bem como as faces a qual a aresta é incidente, e para cada vértice as arestas e face incidentes.

Uma malha M caracteriza a geometria da superfície pela posição de seus vértices

$$V = \{v_1, v_2, \dots, v_n\}$$

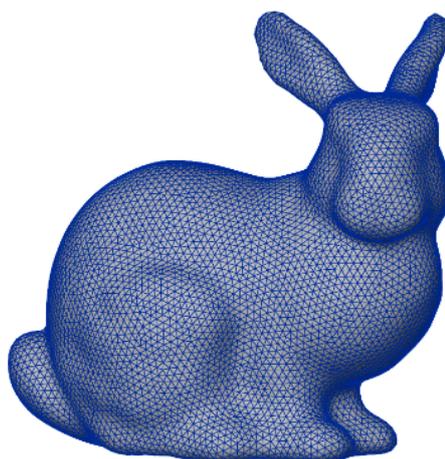


Figura 2.2: Malha de uma Superfície

Fonte: Autor, 2015

Definição 2. *Uma malha M de uma superfície S é um conjunto de faces em que a intersecção de duas faces quaisquer é uma aresta comum, um vértice comum a estas duas faces ou vazia.*

A orientação de uma face é uma ordem cíclica dos vértices incidentes. Há duas possibilidades de orientação para cada face, no sentido horário ou anti-horário. A orientação de duas faces adjacentes é *compatível* se, e somente se, os dois vértices da aresta comum incidente estão em ordem inversa. Uma malha é denominada *orientável*, se e somente se, existe uma escolha de orientações da face que faz com que todos os pares de faces adjacentes sejam compatíveis.

A *Característica de Euler* é um invariante topológico que descreve uma relação entre o número de vértices V , arestas A e faces F , e o tipo topológico de uma variedade orientável:

$$|V| - |A| + |F| = 2(1 - g)$$

onde g é o *genus* (gênero) da superfície, isto é, o número de furos que a superfície apresenta.

Uma malha tem gênero g se, e somente se, podemos cortar a malha ao longo de dois caminhos fechados sem desconectar a malha. A esfera, por exemplo, tem gênero zero e o toro tem gênero um. Algumas malhas de gênero g podem ser deformadas em uma esfera com g alças.



Figura 2.3: Bitoro de Gênero 2

Fonte: Disponível em https://en.wikipedia.org/wiki/Genus-2_surface

No caso especial de uma variedade fechada cada aresta tem exatamente dois triângulos incidentes e cada triângulo três arestas incidentes, daí teremos:

- O número de faces é duas vezes o número de vértices, $|F| \approx 2|V|$.
- O número de arestas é três vezes o número de vértices, $|A| \approx 3|V|$.

2.3 Vizinhança Estrelada

Definição 3. *Seja v_i um vértice da malha M . A vizinhança estrelada de ordem k de v_i é o conjunto de vértices $\{v_1, v_2, \dots, v_n\}$ que está separado de v_i por no máximo k arestas.*

A primeira vizinhança estrelada de v_i é composta pelos vértices adjacentes a este vértice inicial, a segunda vizinhança de v_i é o conjunto de vértices adjacentes a cada vértice da primeira vizinhança de v_i junto com esta primeira.

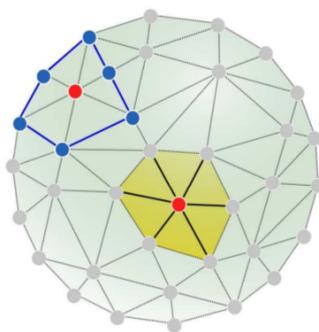


Figura 2.4: Vizinhança Estrelada

Fonte: Disponível em http://www.maxwell.vrac.puc-rio.br/8176/8176_3.PDF

A k -ésima vizinhança estrelada é o conjunto de vértices adjacentes a cada vértice da $(k - 1)$ -ésima vizinhança, junto com esta vizinhança.

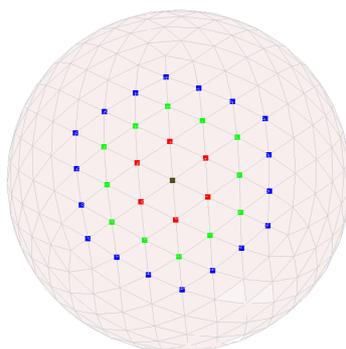


Figura 2.5: Terceira Vizinhança Estrelada

Fonte: Autor, 2015

2.4 Transformações Tridimensionais

Em diversas aplicações na área de computação gráfica, há a necessidade de alterar e manipular a posição dos elementos que compõem uma dada superfície ou malha. Mudanças na orientação, tamanho e formato do objeto estão ligadas às transformações geométricas. Estas são aplicadas à malha para alterar sua geometria sem fazer alterações topológicas. Uma transformação geométrica é uma *aplicação bijetiva* entre duas malhas de forma que a partir de uma malha inicial M_i dada se forma outra topologicamente equivalente, M_f .

As principais transformações geométricas, como *translação*, *rotação* e *escala*, serão discutidas aqui, as quais são de fundamental importância para o alinhamento de duas malhas usando PCA.

2.4.1 Translação

Seja M uma malha a ser transladada. A translação de M é uma mudança na posição dos vértices v_i em relação ao seu sistema de coordenadas. Transladar M é adicionar uma constante para cada v_i , gerando assim novas posições para cada vértice de M .

Seja $v = (x, y, z)$ um vértice de M , se transladamos v por (t_x, t_y, t_z) à nova posição de v será um vértice $v' = (x', y', z')$ tal que:

$$x' = x + t_x$$

$$y' = y + t_y$$

$$z' = z + t_z$$

Então transladar uma malha M_i é somar um vetor de translação a cada vértice da malha, e como resultado temos uma malha M_f que preserva a geometria e topologia de M_i .

2.4.2 Rotação

Seja M uma malha a ser rotacionada. A rotação de M é uma mudança na posição dos vértices v_i em relação ao seu sistema de coordenadas. Em nosso caso particular, rotacionar M é, para cada $v_i \in M$ aplicarmos uma matriz de mudança de base gerando assim novas posições para cada vértice de M .

Seja $T_M = \{m_1, m_2, m_3\}$ a base do espaço definida por M e $T_N = \{n_1, n_2, n_3\}$ a base do espaço definida por N , então a rotação de M é a mudança de base $T : M \rightarrow N$.

Para rotacionar M mudamos a base de T_M para T_N . Sendo assim, existe uma lista de escalares λ_{ij} tal que:

$$n_1 = \lambda_{11}m_1 + \lambda_{21}m_2 + \lambda_{31}m_3$$

$$n_2 = \lambda_{12}m_1 + \lambda_{22}m_2 + \lambda_{32}m_3$$

$$n_3 = \lambda_{13}m_1 + \lambda_{23}m_2 + \lambda_{33}m_3$$

E se quisermos:

$$n_j = \sum_{i=1}^3 \lambda_{ij}m_i$$

A matriz de mudança da base M para a base N é:

$$M_{N \leftarrow M} = \begin{bmatrix} \lambda_{11} & \lambda_{21} & \lambda_{31} \\ \lambda_{12} & \lambda_{22} & \lambda_{32} \\ \lambda_{13} & \lambda_{23} & \lambda_{33} \end{bmatrix}$$

Esta é a matriz de mudança de base que será aplicada a cada um dos vértices de M , após sua aplicação teremos a nova posição de cada vértice que implica na rotação de M .

2.4.3 Escala

Seja M uma malha a ser escalada. A escala de M é uma mudança na posição dos vértices v_i em relação ao seu sistema de coordenadas. Escalar M é multiplicar todas as coordenadas do vértice v_i por um fator não nulo, gerando assim novas posições para cada vértice de M .

Seja $v = (x, y, z)$ um vértice de M , se escalarmos v por (s_x, s_y, s_z) a nova posição de v será um vértice $v' = (x', y', z')$ tal que:

$$x' = x \cdot s_x$$

$$y' = y \cdot s_y$$

$$z' = z \cdot s_z$$

Em termos matriciais temos:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Tomando $S = (s_x, s_y, s_z)$, podemos ter:

- Se $S > 1$ a escala provoca uma ampliação em M na direção do(s) eixo(s) afetado(s) pelo fator;
- Se $0 < S < 1$ a escala provoca uma redução de M ;

3 ANÁLISE DE COMPONENTES PRINCIPAIS

A *Análise de Componentes Principais* ou *Principal Components Analysis* PCA é um método que tem por finalidade básica tanto a análise de dados visando reduzir sua dimensão quanto a eliminação de sobreposições a partir de combinações lineares das variáveis originais.

A PCA por muitas vezes é chamado também de *Transformada Discreta de Karhunen-Loève* (KLT) ou ainda de *Transformada Hotelling* [Lindsay], em homenagem a *Kari Karhunen*, *Michael Loève* e *Harold Hotelling*. Esta transforma variáveis discretas em coeficientes descorrelacionados. Foi derivada por *Hotelling* e por ele denominada como *Método dos Componentes Principais*.

3.1 Componentes Principais

A PCA é uma maneira de identificar a relação entre características extraídas de dados. É de grande utilidade quando os vetores de características tem muitas dimensões, quando uma representação gráfica não é possível, mas também pode ser útil em dimensões menores. Em nosso caso particular, temos uma malha M que possui v vértices. Nosso objetivo ao aplicar PCA, é gerar em M um *triedro* que fornecerá as direções principais desta malha, de forma que estas sejam representadas por vetores no \mathbb{R}^3 , onde a maior direção indica onde os vértices estão mais espalhados e a menor direção indica onde estão menos espalhados os vértices ao longo da malha.

A componente principal é a direção que melhor representa a distribuição dos vértices ao longo da malha, de forma que é nesta componente onde há menor concentração de vértices ao longo da malha, onde há maior concentração denominamos por *componente secundária*, e entre estas há uma outra componente de distribuição à qual chamamos de *componente média da distribuição*. Com estas três componentes formamos o triedro associado à malha, que fornece à informação da distribuição dos vértices ao longo da malha em suas direções, principal, secundária e média.

Para a determinação da PCA em uma malha M , tomemos V como o conjunto de todos os vértices v_i , com $i = 1, \dots, n$ de M , tal que $V = \{v_1, v_2, \dots, v_n\}$. Para determinar os componentes principais seguimos os passos:

- 1 - Tomamos todos os vértices v de M , os quais escrevemos como uma lista de dimensão n .
- 2 - Determinamos a média v_m de todos os vértices de M .
- 3 - Determinamos a diferença de cada v à v_m .
- 4 - Calcular a matriz de covariância usando todas as diferenças.
- 5 - Calcular os autovalores e autovetores da matriz de covariância.
- 6 - Arranjar a *Matriz de Hotelling*, de tal forma que as colunas são formadas a partir dos autovetores da matriz de covariância arranjados de modo que a primeira coluna, seja o autovetor correspondente ao maior autovalor, e a última coluna corresponda ao menor autovalor.

O autovetor com maior autovalor associado, corresponde à componente principal do conjunto de vértices da malha, conseqüentemente o autovetor com menor autovalor associado, corresponde à componente secundária.

3.1.1 Matriz de Covariância

Dado um conjunto de dados, sabemos que é possível determinar suas principais características no que diz respeito à estatística. Sobre tais dados podemos fazer uma análise a partir da *Média Aritmética*, *Desvio Padrão* e *Variância*.

Definição 4. A covariância é um conjunto de primeira ordem das variáveis aleatórias x e y , centrados nas respectivas médias, ou seja, é a média do grau de interdependência entre estas duas variáveis.

A covariância é dada por:

$$\text{cov}(x, y) = \frac{1}{n} \sum_{i=1}^n [(x_i - m_x) \cdot (y_i - m_y)]$$

Onde, x e y são as componentes bidimensionais da lista de dados, m_x e m_y são respectivamente, as médias da componente x e y da lista. E x_i e y_i , são os elementos da listas nas duas direções em sua i -ésima posição. E finalmente n é o número de elementos da lista.

Propriedade: Sejam x e y duas variáveis aleatórias de valor real, então:

(a) $\text{cov}(x, x) = \text{var}(x)$

(b) $\text{cov}(x, y) = \text{cov}(y, x)$

Como estamos trabalhando com superfícies, então sendo M a malha, nossa lista de dados guarda todos os vértices de M , com isso para cada $i = 0, \dots, n$ teremos os vértices $v_i = (x_i, y_i, z_i)$. Então como esta lista de vértices está em coordenadas tridimensionais, calculamos a covariância entre cada par de dimensões, isto é, determinamos $\text{cov}(x, y)$, $\text{cov}(x, z)$ e $\text{cov}(y, z)$.

O vetor médio para os vértices de M em cada coordenada é:

$$m_x = \frac{1}{n} \sum_{i=1}^n x_i$$

$$m_y = \frac{1}{n} \sum_{i=1}^n y_i$$

$$m_z = \frac{1}{n} \sum_{i=1}^n z_i$$

Sendo assim teremos a matriz de covariância:

$$C = \begin{bmatrix} \text{cov}(x, x) & \text{cov}(x, y) & \text{cov}(x, z) \\ \text{cov}(y, x) & \text{cov}(y, y) & \text{cov}(y, z) \\ \text{cov}(z, x) & \text{cov}(z, y) & \text{cov}(z, z) \end{bmatrix}$$

A diagonal principal da matriz C contém as variâncias e as demais posições a correlação entre as direções. Essa matriz é simétrica e real, de modo que é sempre possível encontrar um conjunto de autovetores ortonormais.

3.2 *Análise de Componentes Principais - PCA*

Dada uma malha M , aplicar PCA à M consiste em encontrar os eixos principais nesta malha.

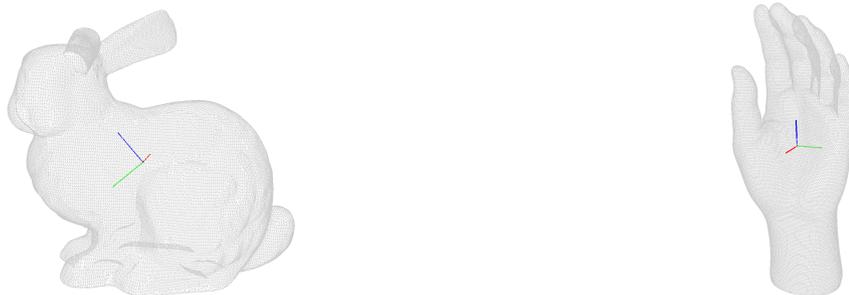


Figura 3.1: Aplicação do PCA ao Bunny e a Mão

Fonte: Autor, 2015

Vemos que após aplicarmos PCA na malha original, temos um *triedro* na nova malha que indica as direções principais a partir do centro de massa de M . Este triedro é composto pelos autovetores ordenados segundo seus autovalores, os quais formam as colunas em ordem crescente da matriz de covariância. O triedro mostrado na figura 3.1 foi colorido segundo o espalhamento dos vértices naquela direção:

- **Azul:** Aponta na direção com maior espalhamento de vértices;
- **Vermelho:** Aponta na direção com menor espalhamento de vértices;
- **Verde:** Aponta na direção com espalhamento de vértices intermediárias as anteriores;

A matriz de transformação para o cálculo da PCA consiste na matriz cujas colunas são os autovetores da matriz de covariância dos vértices da malha. A matriz de covariância é uma matriz simétrica e definida positiva, que guarda a informação sobre as variâncias em todos os eixos onde os dados estão distribuídos.

3.2.1 Alinhamento

O processo de alinhamento de duas malhas usando PCA, nos fornece um encaixe entre estas duas malhas de forma à facilitar o processo de projeções que veremos no capítulo posterior. Sejam M_s e M_b as malhas que queremos alinhar. Dessa forma, o alinhamento ocorre da seguinte maneira:

1. Transladamos o baricentro de M_s para o baricentro de M_b .
2. Após a translação, aplicamos a rotação em M_s , que nada mais é que mudar a base do triedro de M_s dada pelo PCA para a base de M_b , o mesmo para seus vértices.
3. E finalmente, aplicamos uma escala M_s que busca deixar a Esfera com o "tamanho" aproximadamente igual ao do Bunny.

O procedimento de aplicarmos as transformações geométricas de translação, escala e rotação a uma das malhas, é fundamentado primeiramente pelo uso do PCA, pois é nesta que temos o ponto de referência inicial para ter o suporte necessário à aplicação das transformações.

Todos os procedimentos para o alinhamento inicial das malhas são baseados no PCA, pois é aí que temos a determinação dos baricentros e triedros das malhas, os quais nos dão o suporte necessário para à aplicação das transformações rígidas.

No processo de alinhamento inicial das malhas, é necessário a determinação do vetor adicional para a translação que denominamos aqui por (t_x, t_y, t_z) . Sendo assim, dadas as malhas M e N , inicialmente queremos transladar M para N , para isto determinamos primeiro o baricentro para ambas as malhas, $B_M = (b_{mx}, b_{my}, b_{mz})$ e $B_N = (b_{nx}, b_{ny}, b_{nz})$. A translação simplesmente é fazer coincidir os baricentros de ambas as malhas, dessa forma o vetor de translação será a diferença dos baricentros B_N e B_M :

$$(t_x, t_y, t_z) = (b_{nx} - b_{mx}, b_{ny} - b_{my}, b_{nz} - b_{mz})$$

Quando aplicamos a PCA, o triedro é ordenado segundo o seu autovalor associado. Sejam $(\lambda_{M_x}, \lambda_{M_y}, \lambda_{M_z})$ e $(\lambda_{N_x}, \lambda_{N_y}, \lambda_{N_z})$ os autovalores ordenados em valores ascendentes para os triedros de M e N respectivamente. Então o fator de escala (s_x, s_y, s_z) aplicado à M é:

$$(s_x, s_y, s_z) = \left(\sqrt{\frac{\lambda_{N_x}}{\lambda_{M_x}}}, \sqrt{\frac{\lambda_{N_y}}{\lambda_{M_y}}}, \sqrt{\frac{\lambda_{N_z}}{\lambda_{M_z}}} \right)$$

Desta maneira, com a determinação da translação, rotação e escala, e com a aplicação da PCA já temos as ferramentas necessárias para realizar as transformações geométricas entre duas malhas afim de determinarmos o alinhamento inicial entre elas, para daí dar início à construção de suas resultantes simplificadas a partir do processo de projeções. Então, aplicados todos os procedimentos descritos acima para as superfícies da Esfera e do Bunny temos o alinhamento:



Figura 3.2: Alinhamento

Fonte: Autor, 2015

4 FAST MARCHING

Quando se atira uma pedra em um lago é gerada uma série de ondas consecutivas e concêntricas, as quais se expandem sempre ao mesmo tempo. Este fenômeno natural se deve à geração da onda em um meio homogêneo que possibilita a velocidade de expansão constante no meio. O ponto de origem da expansão é o ponto onde a pedra cai, que irá gerar as demais ondas circulares onde teremos um certo tempo para alcançar cada uma delas.

Dessa forma, durante a expansão da onda, temos o tempo para se alcançar os pontos do espaço, sendo o tempo onde a onda se origina em $T = 0$.

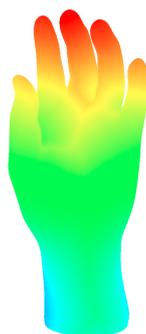


Figura 4.1: Ilustração FMM

Fonte: Autor, 2015

O *Fast Marching Method* (FMM), proposto inicialmente por J. Sethian em 1996 [Sethian], é um método numérico que tem por objetivo aproximar a solução viscosa em uma malha M em $O(n \log n)$ passos, onde n é o número de vértices de M , da equação **Eikonal**:

$$|\nabla T| = F(x, y) \quad (4.1)$$

4.1 Fast Marching

A equação (4.1) descreve como se expande uma onda, onde x é a posição, $F(x)$ é a velocidade de propagação da onda na posição dada, e $T(x)$ é o tempo para alcançar a posição x . A idéia central é construir uma aproximação para o gradiente de T .

Assim, utilizamos $F \equiv 1$, o que leva a função distância a um ponto ser monótona crescente. Na figura 4.5, mostramos a propagação de uma onda.

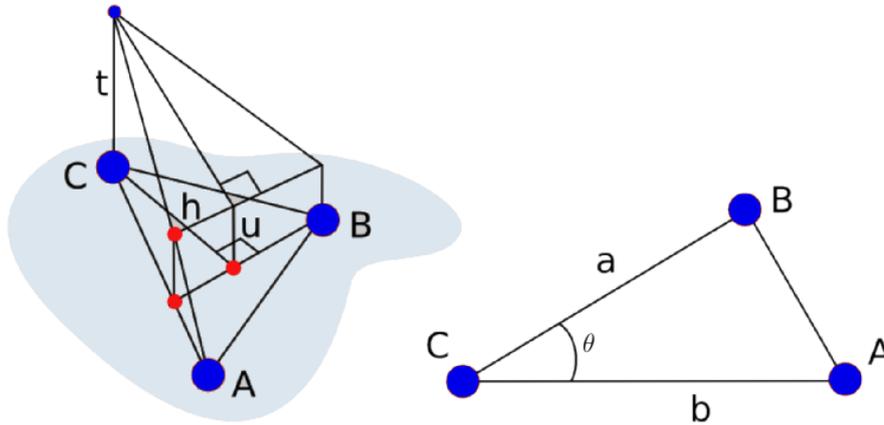


Figura 4.2: Discretização do FMM

Fonte: Autor, 2015

Para determinar a distância do ponto u ao ponto v , calcula-se inicialmente a distância à primeira vizinhança estrelada de u utilizando a distância euclidiana. A partir daí o FMM 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. Dessa forma, utilizando a notação da figura 4.2, deve-se determinar o valor de $T(C)$ a partir dos valores conhecidos de $T(A)$ e $T(B)$.

Em uma construção para triângulos agudos, a discretização é obtida da seguinte maneira:

- $u = T(B) - T(A)$
- Resolver a equação em t :

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

- Se $u < t$ e $\cos \theta < \frac{b(t-u)}{t} < \frac{a}{\cos \theta}$:

$$T(C) = \min\{T(C), t + T(A)\}$$

- Caso contrário:

$$T(C) = \min\{T(C), bF + T(A), cF + T(B)\}$$

A equação 4.1 é um caso particular da *Equação de Hamilton - Jacobi* [Peixoto e Velho]. A solução numérica desta equação é baseada nas *Leis da Conservação Hiperbólica*. Sendo assim, a equação Eikonal pode ser resolvida pela equação:

$$\left[\max(D_{ij}^{-x}T, -D_{ij}^{+x}T, 0)^2 + \max(D_{ij}^{-y}T, D_{ij}^{+y}T, 0)^2 \right]^{\frac{1}{2}} = F_{ij}$$

Onde:

$$D_{ij}^{-x} = \frac{T_{ij} - T_{i-1,j}}{\Delta x}$$

e

$$D_{ij}^{+x} = \frac{T_{i+1,j} - T_{ij}}{\Delta x}$$

$$D_{ij}^{-y} = \frac{T_{ij} - T_{i,j-1}}{\Delta y}$$

e

$$D_{ij}^{+y} = \frac{T_{i,j+1} - T_{ij}}{\Delta y}$$

A idéia central por trás do FMM para resolver estas equações é o processo de iteração nos pontos da malha, e cada iteração os novos pontos vão sendo calculados:

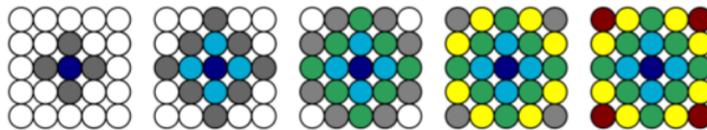


Figura 4.3: Processo de Iteração

Fonte: Disponível em [Verónica Gonzalez]

Ao aplicarmos o FMM, durante o cálculo dos valores de T a informação vai sempre se propagando na direção dos pontos que possuem os menores valores de T .

Pela figura abaixo, vamos verificar como o processo de propagação ocorre à cada iteração:

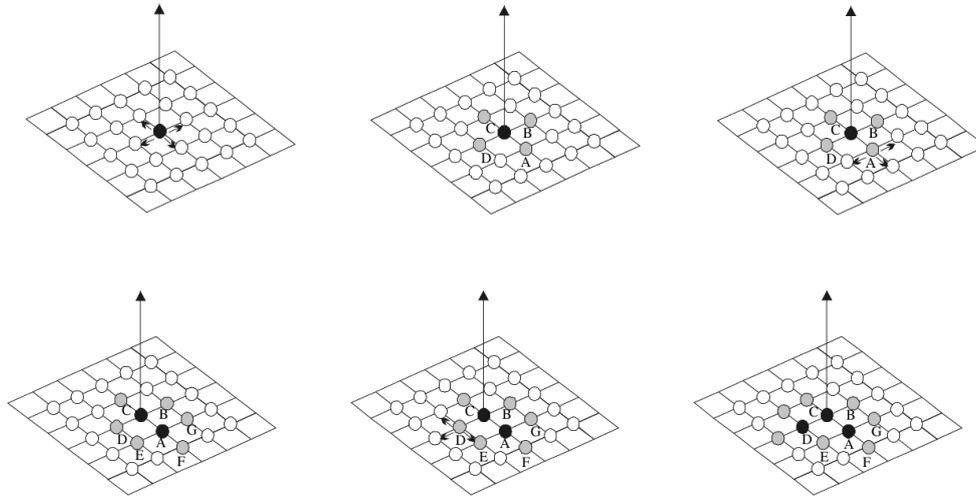


Figura 4.4: Processo de Propagação

Fonte: Disponível em [\[Peixoto e Velho\]](#)

Os pontos pretos representam as posições onde a função T é conhecida e os pontos brancos, posições onde T é desconhecida. Partindo de uma interface inicial, representada por um ponto preto, onde $T = 0$ são calculados os valores de seus quatro vizinhos (representados pelos pontos cinzas A, B, C e D, na segunda imagem), através da equação Eikonal discreta. Dentre estes quatro pontos a propagação deve seguir a partir daquele que tiver o menor valor de T , ou seja, o algoritmo para propagação requer que haja uma ordenação dos pontos cinzas. Supondo que o ponto A contém o menor T , a propagação deve prosseguir a partir dele (terceira imagem), ou seja, o ponto A é alterado para preto, indicando que a propagação já é conhecida em A, e seus vizinhos são calculados (representados pelos pontos cinzas E, F e G, na quarta imagem).

É importante observar que o ponto preto inicial, apesar de ser vizinho de A, não entrou neste processo, pois a propagação tem sentido único, não é retroativa. A propagação deve continuar a partir do ponto cinza (B, C, D, E, F ou G) que tiver o menor valor de T . Mais uma vez será necessária a rotina de ordenação para selecionar o ponto cinza de menor T . Supondo que o ponto D contém o menor valor, a propagação deve seguir a partir dele (quinta imagem). O valor de T é calculado nos vizinhos de D, que são setados para cinza. É importante observar que um dos vizinhos de D (o ponto E) já era cinza, pois também é vizinho de A, mas seu valor deve ser recalculado (apenas os vizinhos pretos são poupados). O processo se repete até que a função T seja determinada.

Usando a equação Eikonal, o cálculo de T nos vizinhos nunca fornece valores menores do que os pontos já conhecidos e portanto, a “marcha” segue sempre em um único sentido, se afastando da origem.

4.2 Fast Marching na Malha

Agora vamos aplicar o FMM a uma malha M começando de um vértice inicial v_0 , tomado arbitrariamente, de forma que a "marcha" percorre para fora em única direção sem retorno ao vértice inicial, alcançando outros vértices da malha. O objetivo principal é explorar uma técnica baseada em uma fila de prioridades rápida para localizar sistematicamente o vértice da malha adequado para atualizar, de modo que nunca precisa recuar para os vértices previamente avaliados. A técnica resultante varre através da malha de n vértices em $O(n \log n)$ passos para obter a distância ao ponto inicial.

Vejam a aplicação do FMM em algumas malhas:



Figura 4.5: FMM na Esfera e no Bunny

Fonte: Autor, 2015



Figura 4.6: FMM no Cavalo e no Armadillo

Fonte: Autor, 2015

O vértice tomado inicialmente, está na região representada na cor azul nas figuras acima. Sendo assim os pontos mais próximos ao vértice inicial são estes que aparecem na região azul, os da região verde são os pontos que estão à uma distância média do vértice inicial, já os pontos da região em vermelho ou que se aproximam de vermelho, em tons, são os pontos que estão mais afastados do vértice tomado inicialmente.

Ao aplicarmos o FMM a um vértice da malha M , estamos criando uma relação de distâncias deste vértice inicial aos demais vértices da malha, isto significa que além de determinarmos o vértice que está mais afastado do inicial, temos as distâncias entre este e os demais.

5 SIMPLIFICAÇÃO DE MALHAS

Diferente de como é tratado em alguns trabalhos relacionados a simplificações de malhas que usualmente vemos nas referências atuais, a proposta deste trabalho é realizar uma simplificação baseada em *templates*, de tal forma que inicialmente temos duas malhas M_d e M_0 , com M_d densa, e a malha simplificada resultante vem da técnica de projeções dos vértices de M_0 em M_d baseado na geração de correspondências entre vértices por uma distância euclidiana mínima.

Neste capítulo trataremos do problema principal de nosso trabalho, de forma que dadas duas malhas M_d e M_0 , com M_d mais densa que M_0 , vamos adaptar a geometria de M_0 , de modo que fique o mais próximo possível de M_d . Note que o *template* M_0 já vem com a topologia predefinida.

5.1 Projeção do Template

Para iniciarmos o processo de projeção dos vértices do *template* na malha densa, e conseqüentemente começar a gerar a malha simplificada resultante, é necessário estabelecermos uma relação entre cada vértice de ambas as malhas. Essa relação será dada usando as menores distâncias entre cada vértice da malha densa aos vértices do *template*.

Sejam, M_0 e M_d o *template* e a malha densa, respectivamente. A geração de correspondência busca para cada vértice $v_d \in M_d$ quem é o vértice mais próximo $v_g \in M_0$. Após calcular para cada vértice da malha densa seu mais próximo no *template*, naturalmente cada vértice do *template* está associado a um *cluster* de vértices da malha densa, que eventualmente pode estar vazio ou ter apenas um elemento.

Então, para cada uma destas situações descritas acima, teremos um tratamento diferenciado:

Cluster com um ou mais Elementos

Para cada *cluster*, iremos agora determinar a média dos vértices que o compõem. Assim sendo, todos os *clusters* C_i terão associados a ele uma média M_i de seus n elementos.

A princípio a projeção é dada da seguinte maneira:

1. Determinamos as correspondências entre a malha densa e o *template*.
2. Determinadas as correspondências, a projeção consiste em colocar cada vértice do *template* na média de seu *cluster*.

Cluster Vazio

Eventualmente existem vértices de M_0 que não são mais próximos de nenhum vértice de M_d . Neste caso fazemos um procedimento semelhante ao de determinar os mais próximos, só que neste no sentido contrário. De tal forma que, para cada $v_i \in M_0$ que não é mais próximo de ninguém, determinamos agora quem é o vértice de M_d que é mais próximo deste. Então esses vértices que inicialmente não eram correspondentes de ninguém serão trocados pelos vértices de M_d que são mais próximo a eles. A projeção para estes vértices é determinada pelo seu mais próximo da malha densa.

Para uma primeira verificação do resultado da simplificação, tomemos a malha densa como sendo a malha do *Bunny* e o *template* à *Esfera*. Daí temos o seguinte resultado:

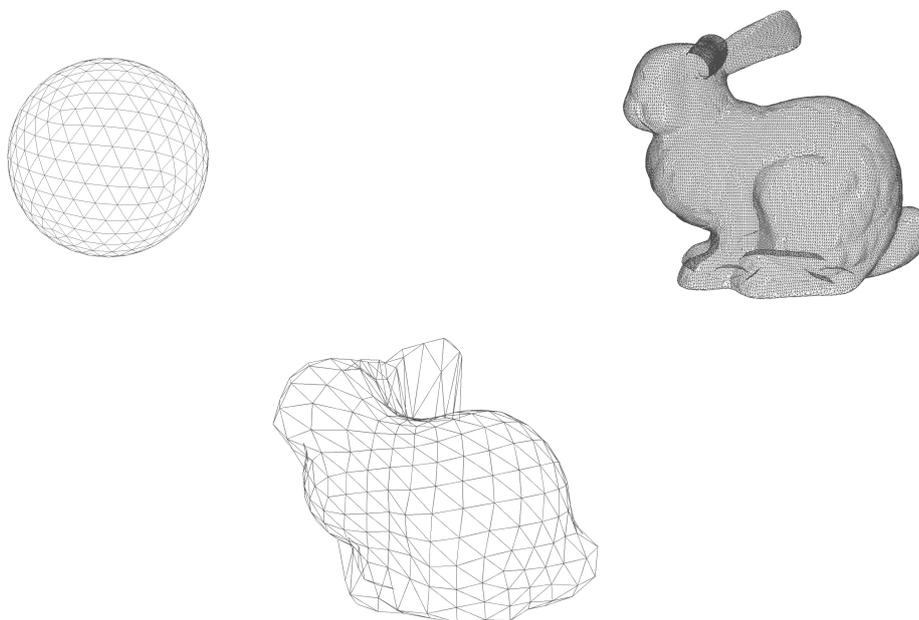


Figura 5.1: Primeira Projeção para o Bunny

Fonte: Autor, 2015

Algoritmo Projeções

Vejam agora os passos a serem seguidos para a realização da projeção dos vértices do *template* M_0 na malha densa M_d .

```

entrada: Malha densa  $M_d$  e o Template  $M_0$ 
saída : Um vetor de listas de correspondências  $L_{dg}$  e Malha Projetada
1 Seja  $L_{dg}$  uma lista de listas vazias com o tamanho de  $M_0$ .
2 para cada vértice  $v_d$  em  $M_d$  faça
3   | Calcule o vértice  $v_g \in M_0$  mais próximo.
4   | Insira  $v_d$  em  $L_{dg}$  na posição de seu mais próximo  $v_g$ 
5 fim
6 para cada entrada de  $L_{dg}$  faça
7   | se a quantidade de vértices é maior que zero, ou se a lista não for vazia então
8     | Determine os clusters
9     | Calcule a média de cada cluster
10    | Na posição  $v_g$  coloque a média do cluster correspondente
11    fim
12  senão
13    | Coloque na posição de  $v_g$  seu mais próximo na malha densa
14  fim
15 fim

```

Algoritmo 5.1: Projeção

No caso especial do Bunny que inicialmente possui 35.165 vértices, e após o processo de projeção passa a ter 482 vértices, percebemos que na região com a concavidade na malha do Bunny, entre a orelha e seu lombo, não fica muito bem representada geometricamente. Isto do ponto de vista topológico não é um problema, e muito menos em seu aspecto de simplificação, pois a malha resultante é muito menos densa que a malha original.

O que ocorre é o seguinte, nesta região da concavidade há correspondentes da esfera que apesar de estarem no mesmo *cluster* pertencem à regiões não convexas do Bunny:

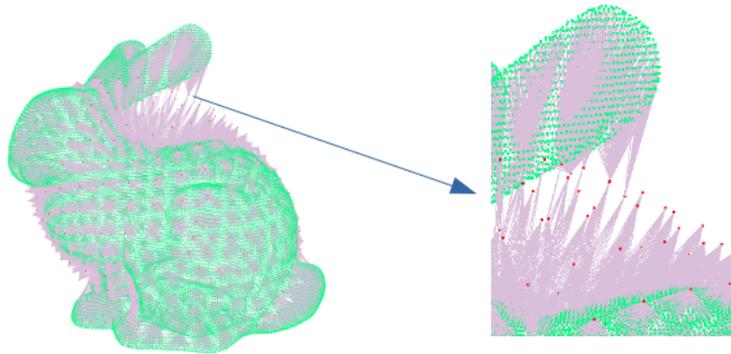


Figura 5.2: Relação das Correspondências entre o Bunny e a Esfera

Fonte: Autor, 2015

Dessa forma, no momento em que a malha resultante é redesenhada estes vértices no "meio do caminho" geram faces entre as regiões não convexas do Bunny, ou seja, as faces ligam a orelha ao lombo da malha. Para resolver esta situação nossa proposta inicialmente é detectar esses vértices em cada *cluster* do Bunny. Em resumo queremos tomar os vértices dos *clusters* que possuem pontos em duas ou mais regiões da malha, e estas regiões estão distantes se andamos sobre esta malha, e tratá-los de alguma maneira que venha a resolver ou minimizar esta situação. A estes vértices cujos *clusters* têm duas componentes conexas denominaremos como *Vértices Distantes*.

Vértices Distantes

Aqui denominamos como *vértices distantes*, os vértices de M_d que pertencem aos mesmos *clusters*, mas estão em regiões diferentes da malha. Para resolvermos esse problema, iremos determinar uma relação entre a distância da média dos vértices do *cluster* ao qual cada vértice distante pertence à uma outra média, que é a *média dos clusters* de M_d . Esse procedimento serve para detectar os vértices de M_d que estão mais afastados da média dos *clusters*, um fato natural do ponto de vista estatístico, que é encontrar e de alguma maneira tratar o vértice que se afasta e/ou está longe da média, que possivelmente são estes que causam o problema da concavidade. Sendo assim vamos construir essa ferramenta para a criação de uma métrica que estabelece quem é ou não distante.

Dado um vértice v_i qualquer da malha densa M_d , tomado aleatoriamente, ao aplicarmos o FMM em v_i estamos criando uma lista de distâncias entre v_i e os demais vértices de M_d . Seja v_d o vértice de M_d que é o mais distante de v_i pelo FMM. Agora aplicamos novamente o FMM ao vértice v_d . Daí temos todas as distâncias de cada vértice de M_d a v_i e v_d . Então para

qualquer vértice de M_d sabemos sua distância para cada um destes dois vértices ao aplicarmos o FMM.

Qualquer vértice de M_d , digamos v , é da forma $v = (x, y, z)$. Desta feita, nosso objetivo aqui é criar uma relação de distância entre cada vértice da malha e os vértices v_i e v_d , de forma que o vértice v agora além de guardar as informações das coordenadas x , y e z , passsam a guardar as duas informações de sua distância a cada um dos vértices que fora aplicada o FMM.

Em resumo seja v um vértice da malha, como vimos acima, sabemos a distância $d_i = d(v, v_i)$ bem como a distância $d_f = d(v, v_d)$, daí cada vértice da malha agora passa a guardar mais estas duas informações das distâncias, então v passa a ser da forma $v = (x, y, z, d_i, d_f)$. O que estamos fazendo aqui é colocando mais duas coordenadas a cada vértice de M_d , nas quais as duas últimas são uma aproximação da geodésica discreta entre dois pontos. Isso permite que nós possamos usar estas novas coordenadas como termos ponderadores no cálculo das médias que fazemos para a projeção das malhas.

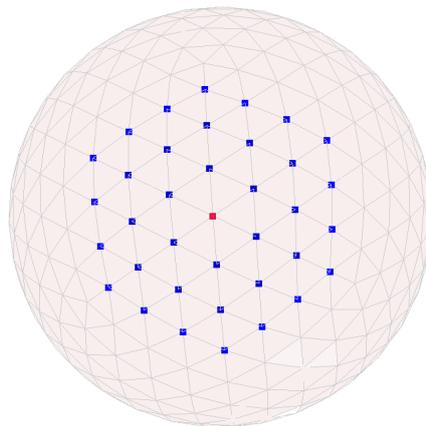


Figura 5.3: Terceira Vizinhança para cada Vértice do Template

Fonte: Autor, 2015

Para cada vértice v de M_0 , representado em vermelho na figura 5.3, tomamos sua terceira vizinhança estrelada. Fazemos o seguinte:

1. Determinamos a quantidade de vértices n_i de seu *cluster* correspondente.
2. Determinamos a média m_i do seu *cluster* correspondente.

Como sabemos, os vértices do *template* são correspondentes de algum *cluster* de vértices da malha densa. Sendo assim cada vértice v_t do *template* possui duas informações que dizem respeito à média m_i de seu cluster correspondente, bem como a quantidade n_i de vértices deste cluster. Então cada vértice v_i da terceira vizinhança, em azul na figura 5.3, guarda estas duas informações, com $v_i = (m_i, n_i)$. Com isso podemos então calcular a média das correspondências na terceira vizinhança para cada vértice do *template*.

Assim, para cada vértice de M_0 a média dos *clusters* é calculada por:

$$M_{\text{clusters}} = \frac{\sum_{i=1}^n m_i \cdot n_i}{\sum_{i=1}^n n_i}$$

Determinada a média das correspondências dos *clusters* para cada vértice de M_0 , podemos então estabelecer um critério para marcar quem é ou não um vértice distante em M_d . Para isto usamos o critério de que o vértice $v_i \in M_d$ será marcado como *distante* se:

$$\|m_i - M_{\text{clusters}}\| > \delta$$

Onde m_i é a média do *cluster* correspondente ao v_i e M_{clusters} a média dos *clusters* correspondentes aos vértices da terceira vizinhança de v_i . Isto nos diz basicamente que são distantes aqueles vértices que estão muito afastados da média dos *clusters*. Em nosso trabalho o valor de δ é dado como parâmetro, o obtemos simplesmente testando alguns valores que identificassem os vértices que realmente causavam problemas naquela região da concavidade. Assim feito o valor adotado na maioria dos resultados foi de $\delta = 4.2$. Por conta da dificuldade de relacionar as características presentes em tudo que é calculado e determinado em cada malha com essa tolerância, fica difícil estabelecer um valor fixo para δ , um critério determinístico para isso, daí nossos resultados buscaram o melhor valor por tentativas que nos fornecessem os resultados esperados.

Solução para os Vértices Distantes

Após determinar os vértices distantes em M_d bem como suas duas novas coordenadas dadas pelo FMM, temos que resolver o que fazer com eles. A princípio fazemos o seguinte, para cada vértice $v_d \in M_d$ em cada *cluster*:

- Se v_d não é distante, a média do *cluster* é mantida na posição inicial da correspondência.
- Se v_d é distante, verificamos a vizinhança de seu vértice correspondente v_{dc} na malha

simplificada.

Agora para estes novos vértices de v_{dc} que ficaram na posição dos vértices distantes, nós fazemos a seguinte análise:

1. Tomamos a vizinhança estrelada $S_{v_{dc}}$.
2. Determinamos a quantidade de vértices n_v de $S_{v_{dc}}$ que são provenientes de médias.
3. Se $n_v \geq 3$, calculamos a média destes vértices m_v , e na posição de v_{dc} colocamos m_v .
4. Se $n_v < 3$, colocamos a média do *cluster* correspondente à v_{dc} em sua posição.

Após aplicarmos o procedimento acima para determinação e substituição dos vértices distantes obtemos o seguinte resultado:

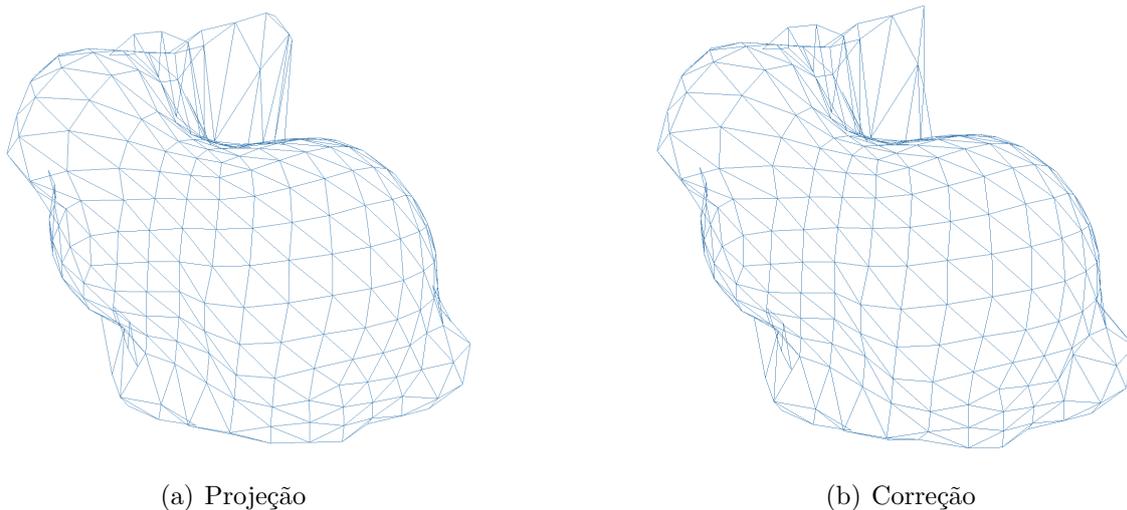


Figura 5.4: Correção dos Vértices Distantes

Fonte: Autor, 2015

Pelo visto acima, após realizarmos a correção dos vértices distantes, há ainda arestas que ligam as regiões da orelha esquerda do Bunny com seu lombo. Apesar de feita esta correção e termos agora uma malha resultante topologica e geometricamente correta, há ainda essa situação das arestas grandes que tornam a malha não muito agradável do ponto de vista perceptível em comparação com a malha inicial.

Sendo assim, para resolvermos este problema, vamos denominar como *arestas grandes*, as arestas que possuem um comprimento maior do que as demais arestas da malha, por exemplo, estas arestas que ligam o lombo e a orelha do Bunny. Para consertar isso usaremos o processo de *subdivisão* destas arestas grandes, a fim de que à cada iteração tenhamos uma redução no tamanho destas arestas e por fim estas deixem de aparecer como no caso acima.

Algoritmo para Detecção e Correção dos Vértices Distantes

```

entrada: Cluster C
saída : Lista de Vértices Distantes  $L_d$  e Malha Corrigida
1 para cada vértice  $v_c$  em  $C$  faça
2   | Calcule a distância  $d$  da média do cluster de  $v_c$  à média dos clusters de  $S_3$  de  $v_c$ .
3   se  $d >$  tolerância então
4     | Marque  $v_c$  como distante
5     | Na posição de  $v_c$  coloque o seu correspondente da malha grosseira
6   fim
7   senão
8     | Na posição coloque a média de seu cluster.
9   fim
10 fim
11 para cada vértice  $v_c$  marcado como distante faça
12   | Verifique a quantidade  $n_v$  de vizinhos que não são distantes
13   se  $n_v >$  3 então
14     | Calcule a média  $m_n$  destes vizinhos.
15     | Na posição de  $v_c$  coloque  $m_n$ .
16   fim
17   senão
18     | Na posição de  $v_c$  coloque a média de seu cluster
19   fim
20 fim
```

Algoritmo 5.2: Detecção e Correção dos Vértices Distantes

Subdivisão das Arestas Grandes

Como descrito acima, temos por objetivo agora resolver o caso particular das arestas que possuem tamanho superior as demais arestas que compõem a malha original. Para isto usamos o processo de *subdivisão* que busca reduzir tais arestas acrescentando um novo vértice à malha bem como duas novas faces que serão associadas a cada aresta grande encontrada.

Como visto anteriormente, mesmo após encontrar e resolver os vértices distantes de M_d ocorre a situação acima apresentada que são essas *arestas grandes* na região côncava do Bunny entre a orelha e o lombo. Primeiramente temos que detectar quais são estas arestas na malha resultante da projeção.

O processo de subdivisão ocorre da seguinte maneira:

1. Encontramos a maior aresta l_e do *template*.
2. Para cada aresta l_n da malha resultante da projeção, construímos a desigualdade $l_n > k \cdot l_e$
3. Se a condição da desigualdade é verificada, a aresta l_n é subdividida.
4. Se não, a aresta l_n é mantida.

Temos os seguinte resultado para a subdivisão:

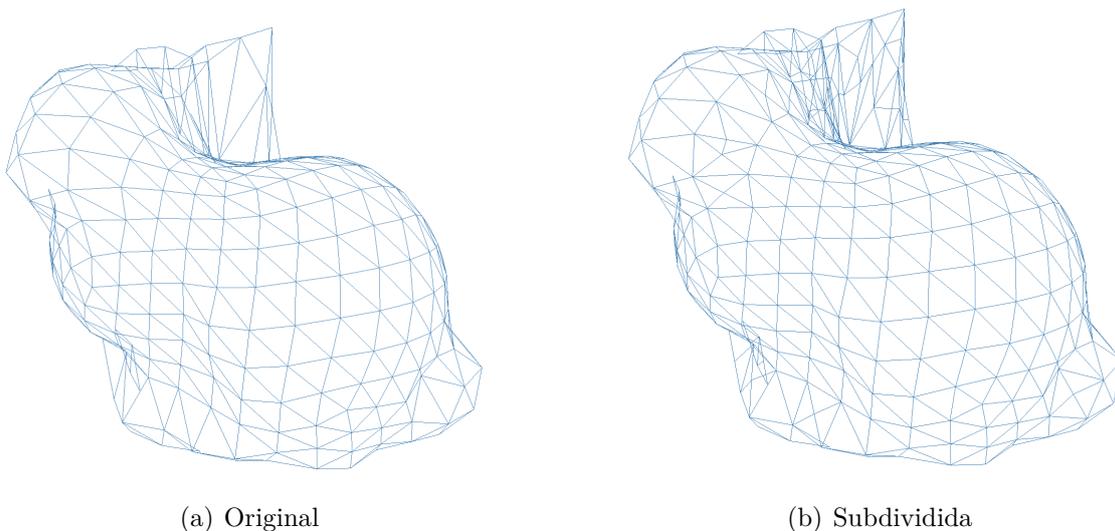


Figura 5.5: Processo de Subdivisão

Fonte: Autor, 2015

O processo para verificar quais arestas serão subdivididas é descrito pelo algoritmo:

Algoritmo para Subdividir

```
entrada: Malha densa  $M_d$  e Template  $M_g$ 
saída  : Arestas Subdivididas
1 Calcule a maior aresta  $l_e$  de  $M_g$ 
2 para cada aresta  $l_d$  em  $M_d$  faça
3   se  $l_d > 1.2 \cdot l_e$  então
4     | Subdivida  $l_d$ 
5   fim
6   senão
7     | Mantenha  $l_d$ 
8   fim
9 fim
```

Algoritmo 5.3: Arestas Grandes

Concluída a subdivisão, dadas as malhas M_0 do *template* e M_d densa, obtemos uma malha M_r tal que a projeção agora é um processo completo dado da seguinte maneira:

1. Aplicamos o PCA a M_0 para alinhá-la à malha M_d .
2. Criamos as correspondências e realizamos a primeira projeção.
3. Detectamos e resolvemos os vértices problemáticos de M_d .
4. Detectamos as arestas grandes e aplicamos a subdivisão a estas arestas.
5. Realizamos todo o processo de projeção novamente, agora levando em conta os novos vértices.

6 RESULTADOS

Para gerar os resultados implementamos as aplicações em C++ usando `gsl`, `OpenGL` e a estrutura de dados *Directed edge*. O fator para determinar os vértices distantes de M_d foi $\delta = 4.2$. Para a aplicação da subdivisão nas arestas grandes usamos o fator multiplicativo da desigualdade das arestas de $k = 1.2$. Nos resultados mostrados a seguir, a nova malha simplificada gerada apresentará a quantidade de vértices pelo menos igual a do *template*.

Nas primeiras imagens mostradas abaixo, temos que:

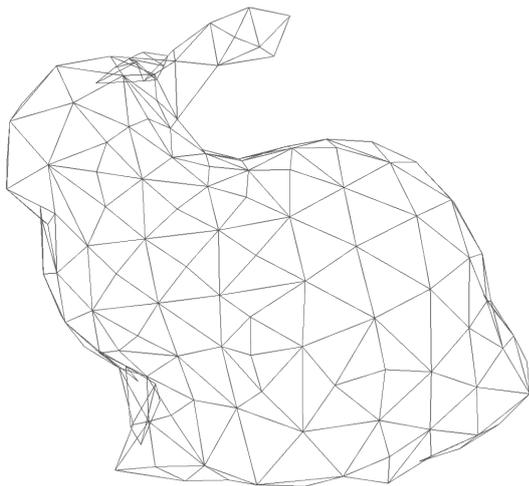
- (a) **Bunny Original**: Refere-se à malha original do Bunny que possui 35.165 vértices;
- (b) **Vênus Original**: Refere-se à malha original de Vênus que possui 100.759 vértices;
- (c) **Buste Original**: Refere-se à malha original de Buste que possui 28.955 vértices;
- (d) **Max Original**: Refere-se à malha original de Max que possui 24.491 vértices;

Em todos os resultados gerados usamos como *template* a esfera, tanto a esfera menor com 162 vértices, até uma esfera maior com 5.122 vértices. No caso do Bunny o processo de iteração foi repetido quatro vezes, para as demais malhas um única vez.

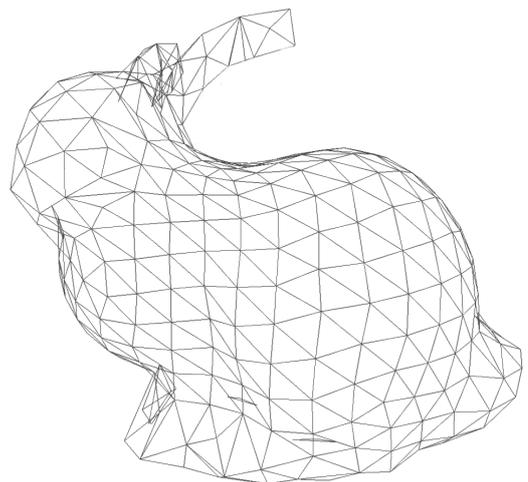
Esfera e Bunny



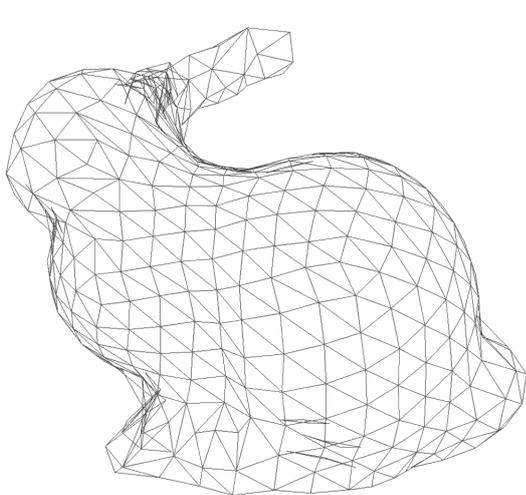
(a) Bunny Original



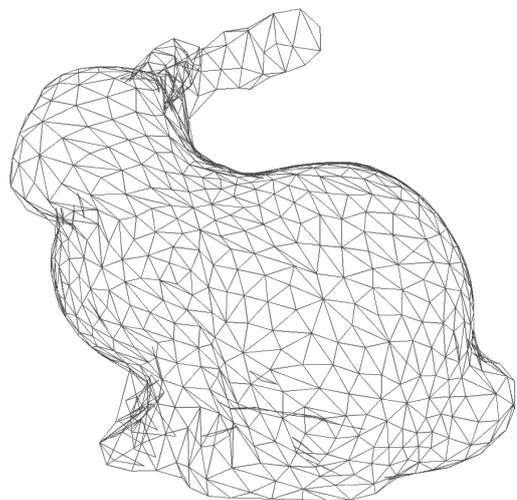
(b) Bunny com 285 Vértices



(c) Bunny com 564 Vértices



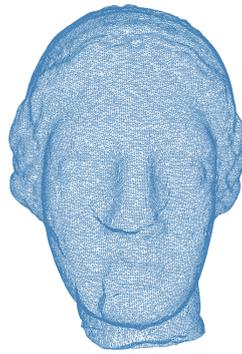
(d) Bunny com 775 Vértices



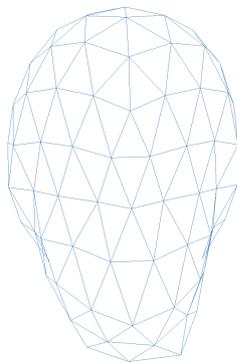
(e) Bunny com 1443 Vértices

Figura 6.1: Simplificação do Bunny

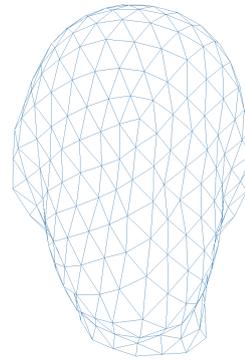
Esfera e Vênus



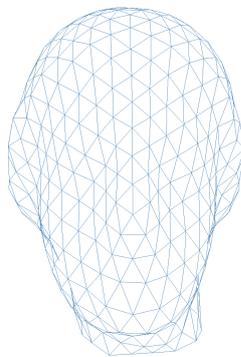
(a) Vênus Original



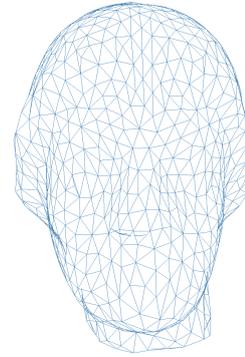
(b) 162 Vértices



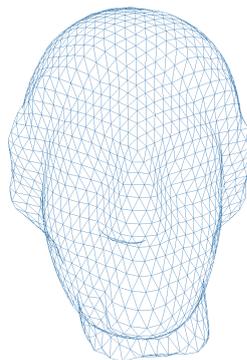
(c) 482 Vértices



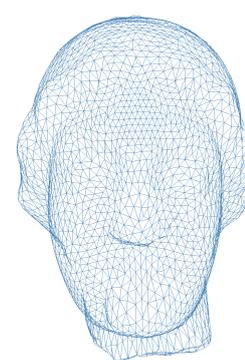
(d) 642 Vértices



(e) 1282 Vértices



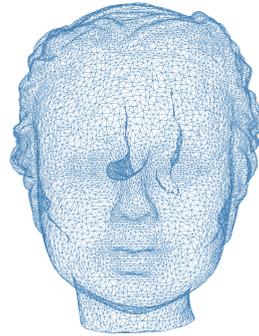
(f) 2562 Vértices



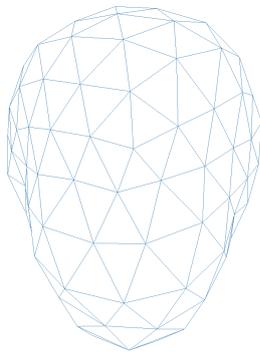
(g) 5122 Vértices

Figura 6.2: Simplificação de Vênus

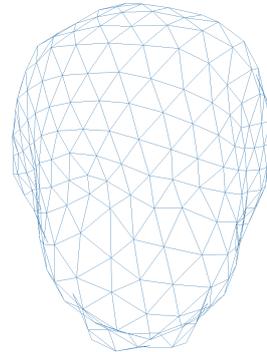
Esfera e Buste



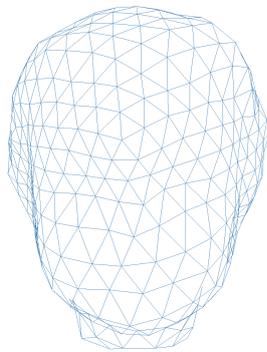
(a) Buste Original



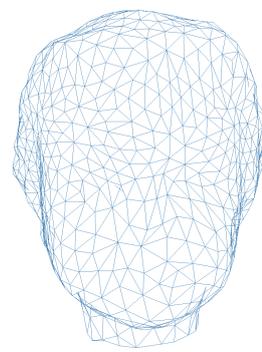
(b) 162 Vértices



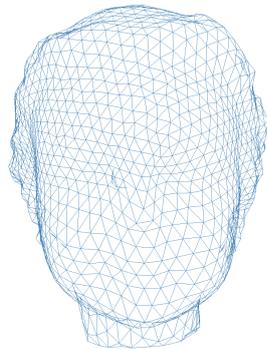
(c) 482 Vértices



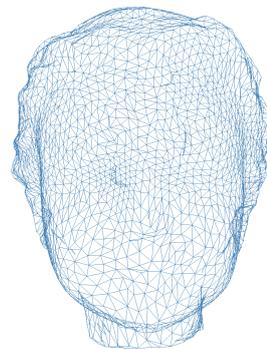
(d) 642 Vértices



(e) 1282 Vértices



(f) 2562 Vértices

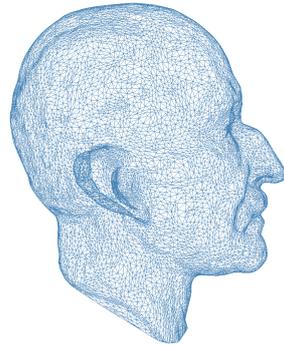


(g) 5122 Vértices

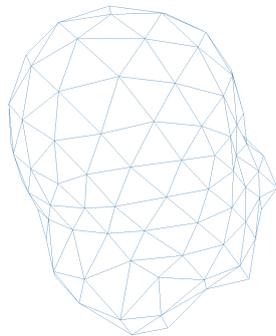
Figura 6.3: Simplificação de Buste

Fonte: Autor, 2015

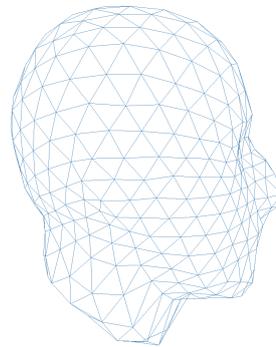
Esfera e Max



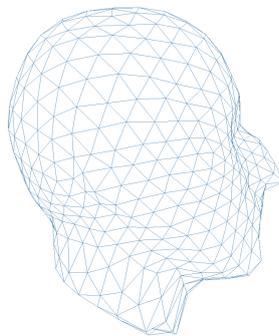
(a) Max Original



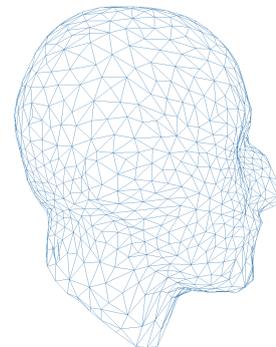
(b) 162 Vértices



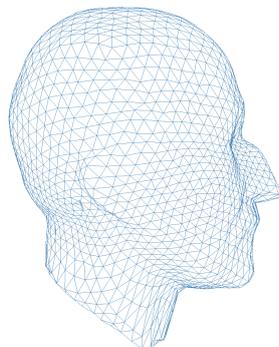
(c) 482 Vértices



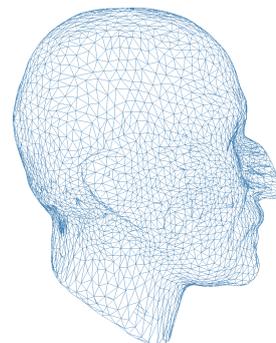
(d) 642 Vértices



(e) 1282 Vértices



(f) 2562 Vértices



(g) 5122 Vértices

Figura 6.4: Simplificação de Max

Fonte: Autor, 2015

6.1 Análise dos Resultados

Tabelas

Para as malhas densas que usamos em nosso processo de simplificação, temos as quantidades de vértices e arestas:

INFORMAÇÕES INICIAIS		
Malha	Vértices	Faces
Bunny	35165	70326
Vênus	100759	201514
Buster	28955	57906
Max	24941	49878

Após realizarmos o processo de simplificação pelas projeções, as malhas apresentadas nas figuras acima apresentam as seguintes características:

INFORMAÇÕES SOBRE OS RESULTADOS			
Figura	Vértices	Faces	RMSD
Bunny 6.1(b)	272	540	0.030918
Bunny 6.1(c)	531	1058	0.030502
Bunny 6.1(d)	725	1446	0.029839
Bunny 6.1(e)	1443	2882	0.029312
Vênus 6.2(b)	162	320	0.046982
Vênus 6.2(c)	482	960	0.046670
Vênus 6.2(d)	642	1280	0.043839
Vênus 6.2(e)	1282	2560	0.042576
Vênus 6.2(f)	2562	5120	0.042144
Vênus 6.2(g)	5122	10240	0.041623
Buster 6.3(b)	162	320	0.048447
Buster 6.3(c)	482	960	0.048390
Buster 6.3(d)	642	1280	0.048373
Buster 6.3(e)	1282	2560	0.048349
Buster 6.3(f)	2562	5120	0.048332
Buster 6.3(g)	5122	10240	0.048319
Max 6.4(b)	162	320	0.026986
Max 6.4(c)	482	960	0.026091
Max 6.4(d)	642	1280	0.025967
Max 6.4(e)	1282	2560	0.025602
Max 6.4(f)	2562	5120	0.025329
Max 6.4(g)	5122	10240	0.025143

Tabela 6.1: Malhas Simplificadas

Nas tabelas apresentadas acima, vemos que após o processo de simplificação pela projeção do *template* em cada uma das malhas dadas obtemos um resultado bastante satisfatório, uma vez que além de reduzir o número de vértices da malha inicial, conseguimos construir a geometria do *template* de forma que este preservou a topologia inicial da malha.

Obtidas as malhas simplificadas, agora iremos analisar o quão fiel é a representação geométrica desta malha com relação a malha densa. Para isto, iremos verificar a proximidade geométrica entre os vértices destas duas malhas no que diz respeito ao erro que mede sua aproximação.

O erro de proximidade ao qual nos referimos é o *desvio quadrático médio* (RMSD) ou a *raiz do quadrado médio de erro* (RMSE), que é uma medida das diferenças entre os valores previstos por um modelo ou um estimador e os valores efetivamente observados.

Gráficos

Dadas as duas malhas, M_d densa e M_r a simplificada resultante, iremos agora verificar o quanto estas duas estão próximas do ponto de vista geométrico. Para isto usaremos uma métrica de distância que busca quantificar a diferença entre a malha inicial e a malha simplificada, que é dada pela distância de *Hausdorff* [Huttenlocher] que mede o quão distantes entre si estão dois conjuntos de pontos. Esta distância atribui para dois conjuntos de vértices uma medida de distância entre eles através de seus vértices.

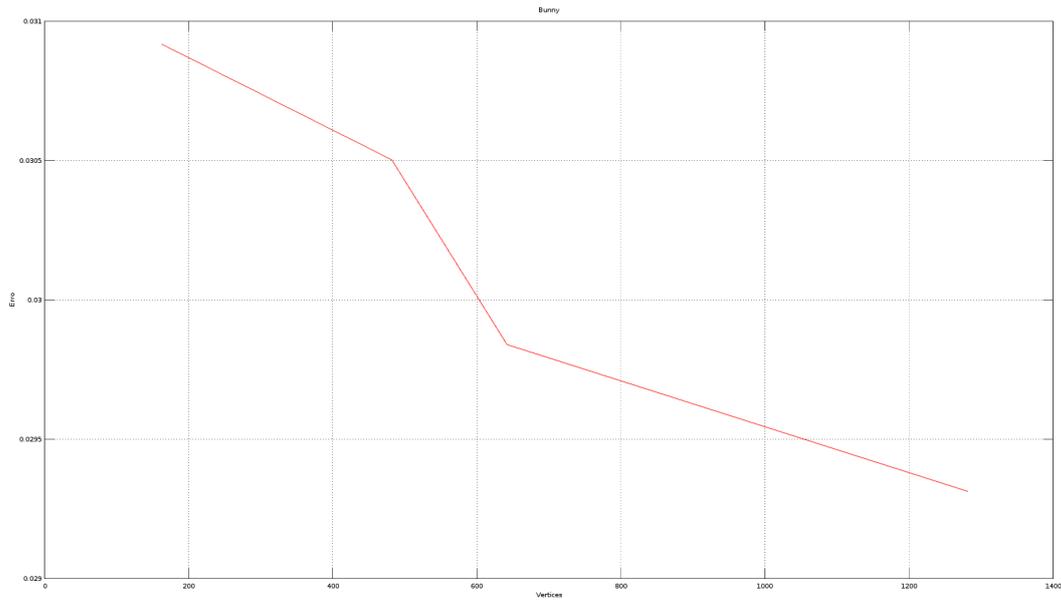
Definição 5. (*Distância de Hausdorff*): Sejam A e B dois subconjuntos não vazios em um espaço métrico (M, d) . A função $H : A \times B \rightarrow \mathbb{R}$, com $H(A, B) = \sup_{a \in A} \{ \inf_{b \in B} d(a, b) \}$ mede a proximidade de A e B .

Assim, a distância de Hausdorff d_H é definida como:

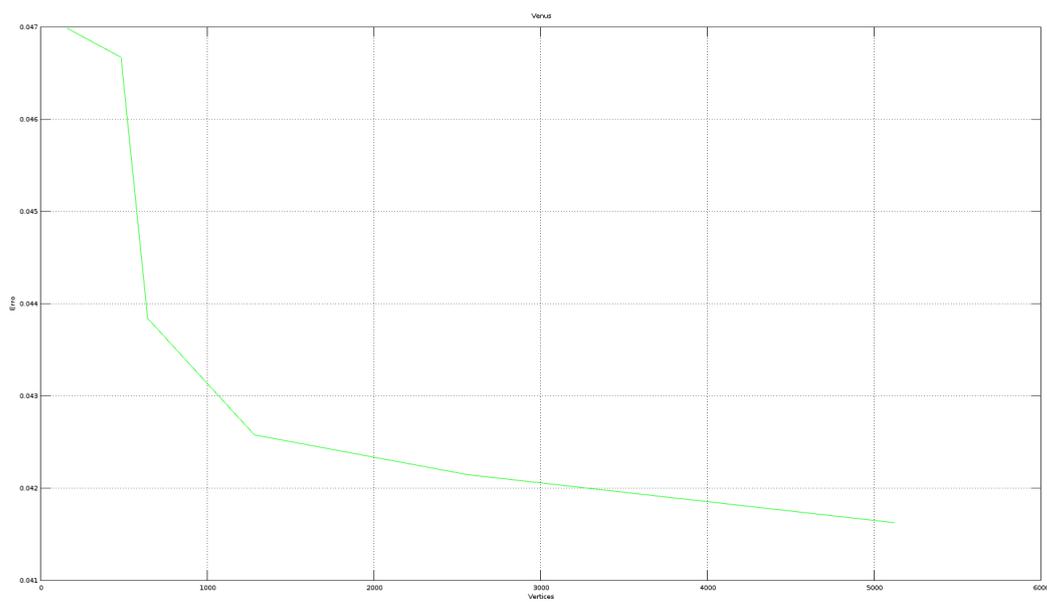
$$d_H(A, B) = \max\{H(A, B), H(B, A)\}$$

A distância de *Hausdorff* representa uma distância entre duas superfícies. Vejamos a aplicação desta distância em cada uma das malhas resultantes na simplificação apresentada em nosso trabalho. Para cada par de malhas, densa e sua respectiva simplificação, determinamos a similaridade entre estas duas malhas por um medida de erro dada pela distância de *Hausdorff*, que a cada malha simplificada que apresenta maior quantidade de vértices o erro de aproximação diminui indicando assim que as duas malhas ficam cada vez mais próximas.

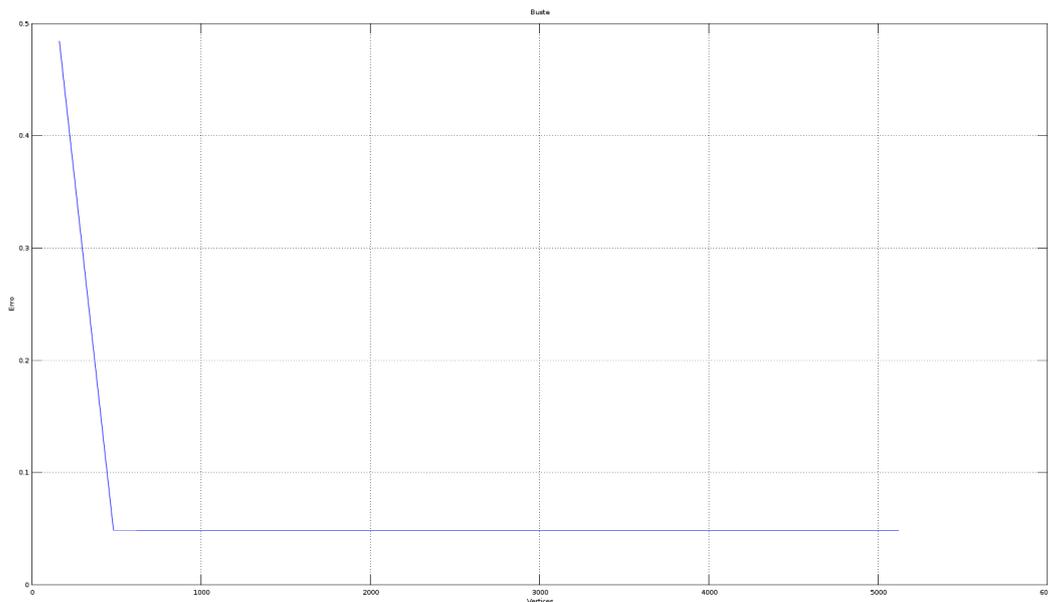
Vejamos nos gráficos abaixo como ficaram os resultados após à aplicação da distância de *Hausdorff* e a determinação do erro de quantificação da diferença entre ambas as malhas.

**Figura 6.5: RMSD Bunny**

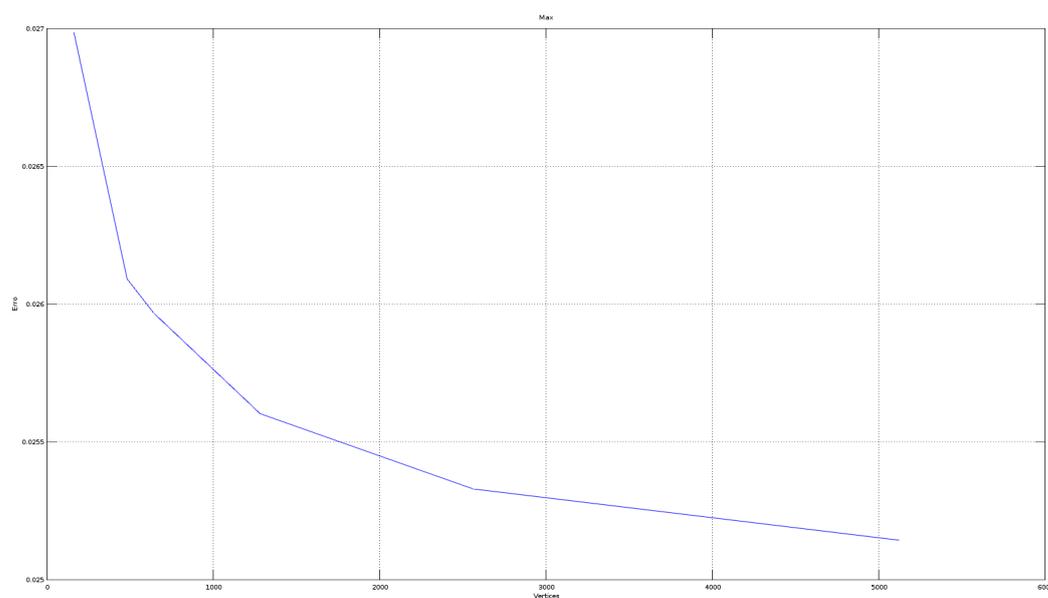
Fonte: Autor, 2015

**Figura 6.6: RMSD Vênus**

Fonte: Autor, 2015

**Figura 6.7: RMSD Buste**

Fonte: Autor, 2015

**Figura 6.8: RMSD Max**

Fonte: Autor, 2015

Pelos gráficos mostrados acima para cada uma das malhas densas, percebemos que ao passo que aumentamos a quantidade de vértices do *template*, a medida do erro de sua malha resultante à inicial vai diminuindo. Isto significa que, no processo de projeção do *template* na malha densa, quanto mais vértices tem o *template* mais fidelizada é sua representação geométrica e topológica com respeito a malha inicial, de forma que termos duas superfícies geometricamente mais próximas quando o erro é menor.

Colorização

Para verificar a medida do erro que foi determinada para cada malha simplificada usando a distância de *Hausdorff*, iremos agora mostrar como fica representado tal erro na malha densa original. Para isto iremos plotar o erro nesta malha usando um processo de colorização dos vértices baseado em sua proximidade à malha inicial.

Na tabela abaixo mostramos os valores de *mínimo* e *máximo* obtidos pela distância de *Hausdorff* para cada uma das malhas simplificadas de *Max*.

MAX		
Malha	Erro Mínimo	Erro Máximo
Figura 6.4 (b)	0.000001	0.213896
Figura 6.4 (c)	0.000001	0.168313
Figura 6.4 (d)	0.000003	0.133386
Figura 6.4 (e)	0.000002	0.123945
Figura 6.4 (f)	0.000001	0.110488
Figura 6.4 (g)	0.000001	0.109941

Tabela 6.2: Erros Max

Nos resultados mostrados abaixo, as regiões em vermelho correspondem ao erro mínimo, e as em azul ao erro máximo. O processo de colorização mostrado a seguir foi obtido usando o *MeshLab*.

Como vemos na tabela acima, para cada malha simplificada temos um valor para o erro mínimo e outro para o erro máximo. Então, para realizar o processo de colorização mostrado na figura 6.9, determinamos a escala do histograma, de forma que esta foi a mesma para todas as malhas simplificadas. Para a determinação desta escala, dentre todos os valores mínimos dos erros tomamos o menor v_{\min} , e dentre todos os valores máximos dos erros tomamos o maior v_{\max} . Sendo assim nossa escala para o histograma usado na colorização foi fixa (v_{\min}, v_{\max}) .

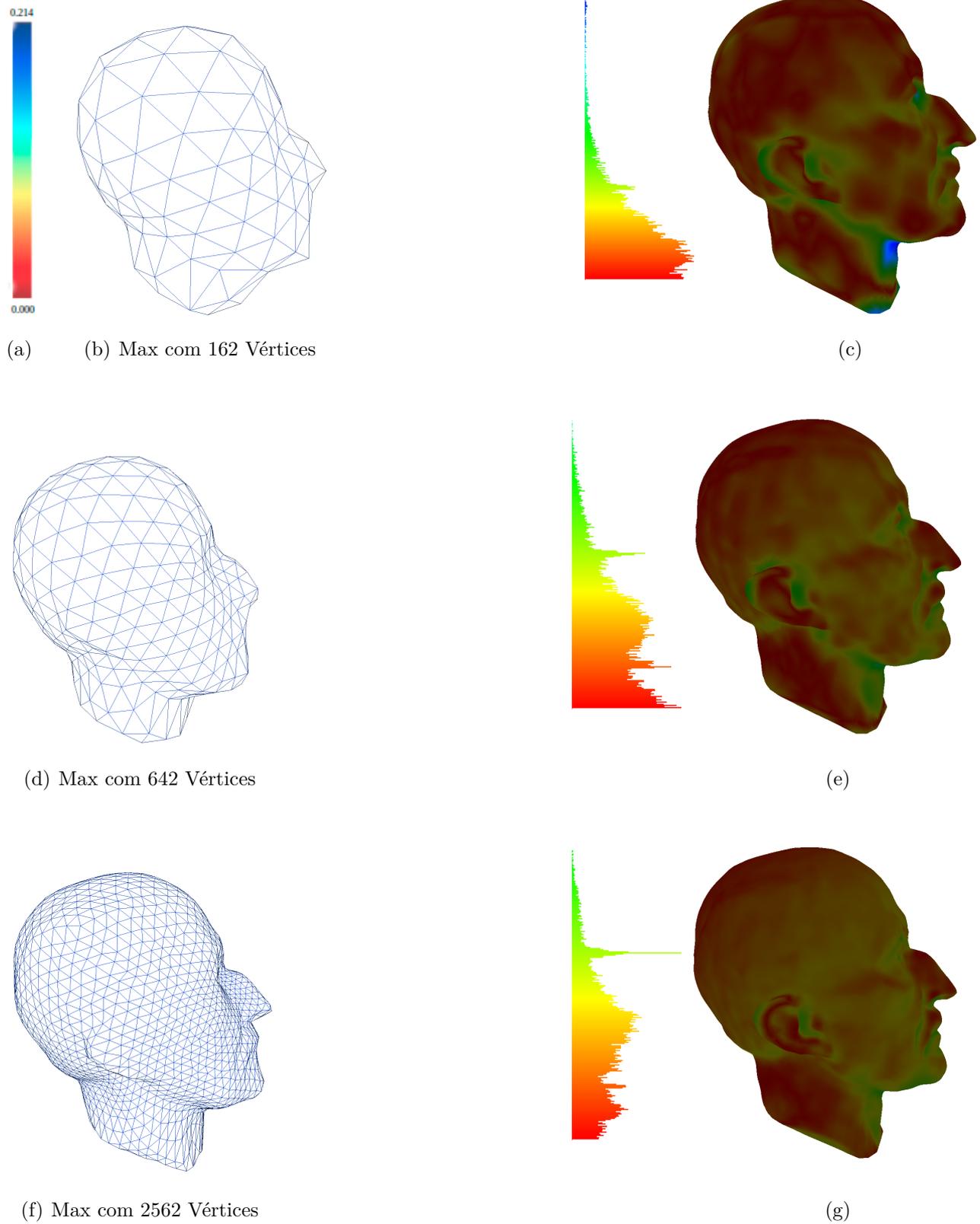


Figura 6.9: Colorização através da distância de *Hausdorff* em *Max*.

7 CONCLUSÕES

Este trabalho apresentou um procedimento para simplificar malhas de superfícies densas a partir da projeção de vértices de um *template*. Para isto utilizamos alguns conceitos bastante conhecidos, como a *Análise de Componentes Principais* e *Fast Marching*.

Nosso trabalho foi dividido em quatro etapas. A primeira foi dada duas malhas M_d densa e M_g do *template*, calculamos o alinhamento entre estas duas a partir da PCA que fornece o melhor "encaixe" entre elas. A segunda etapa foi a realização das projeções a partir da geração de correspondências pela menor distância euclidiana, onde os vértices de M_g foram projetados em M_d segundo sua relação com a quantidade de elementos de cada *cluster*. Realizadas todas as projeções, na terceira etapa identificamos e tratamos todos os vértices de M_d cujo *cluster* possuía duas componentes conexas. Na quarta etapa tratamos os casos das arestas grandes da malha resultante das projeções na região côncava desta malha.

De um modo geral, nosso trabalho apresentou bons resultados principalmente para aquelas malhas que possuem uma geometria muito parecida com a da esfera, como por exemplo a de Vênus que na primeira iteração do processo já gerou uma boa aproximação em sua malha resultante. Além disso, a proposta deste trabalho nos fornece a possibilidade de gerar uma malha simplificada de uma inicialmente densa a partir do processo de projeção dos vértices do *template*, o que viabiliza a reconstrução de malhas simplificadas entre superfícies que apresentam a mesma topologia. A contribuição dada por este trabalho é de construir um algoritmo simples que projeta vértices de um *template* em uma malha densa, tendo a topologia e geometria definidas pela primeira.

Para um trabalho futuro, poderemos ter:

1. Otimizar o tempo de execução do nosso algoritmo.
2. Generalizar este processo de simplificação para malhas com gêneros diferentes de zero.
3. Encontrar uma relação entre os parâmetros para a determinação dos vértices cujos *clusters* possuem duas componentes conexas, bem como para aplicar o processo de subdivisão das arestas grandes.
4. Determinar uma solução mais eficaz para tratar os vértices cujos *clusters* possuem duas componentes conexas.

REFERÊNCIAS

- [Hoppe] Hoppe Hugues, Tony DeRose, Tom Duchamp, Jonh McDonald e Werner Stuetzle. **Mesh Optimization**. Univertsity of Washington.
- [Gotsman] Craig Gotsan, Stefan Gumhold and leif Kobbelt. **Simplification and Compression of 3D Meshes**. Computer Graphics Group, RWTH-Aachen, Germany.
- [Garland] Michael Garland and Paul S. Heckbert. **Surface Simplification Using Quadric Error Metrics**. Carnegie Mellon University, 1997.
- [do Carmo] Manfredo Perdigao do Carmo. **Geometria diferencial de curvas e superficies**. SBM, 2010.
- [Lindsay] Lindsay I Smith. **A tutorial on Principal Components Analysis**. February, 2002.
- [Sethian] Ron Kimmel e James A Sethian. **Computing Geodesic paths on Manifolds**. **Proceedings of the National Academy of Sciences**. Departament of Mathematics and Lawrence Berkley National Laboratory, University of California, Berkeley. May, 1998.
- [Verónica Gonzalez] Verónica González Pérez, Concepción A. Monje e Carlos Balaguer. **Planificación de Misiones de Vehículos Áereos no Tripulados con Fast Marching en un Entorno 3D**. Dpto. Ingeniería de Sistemas y Automática, Universidad Carlos III de Madrid.
- [Peixoto e Velho] Adelailson Peixoto e Luiz Velho. **Transformadas de Distâncias**. PUC-Rio, Setembro, 2000.
- [Morera] Dimas Martínez Morera. **Geodesic-based Modeling on Manifold Triangulations**. IMPA, 2006. Oriented by Paulo Cezar Carvalho and Luiz Velho.
- [Lorusso] A. Lorusso, D. W. Eggert and R. B. Fisher. **A comparasion of Four Algorithms for Estimating 3D Rigid Transformations**. Robotics and Automation Departament TecnoPolis CSATA Novus Ortus. Valenzano, Bari, Italy.
- [Shaoyi] Shaoyi Du, Nanning Zheng, Shihui Ying, Qubo You, Yang Wu. **An Extension Of The ICP Algorithm Considering Scale Factor**. Institute of Artificial Intelligence and Robotics. Xi'an Jiaotong University, Xi'an Shaanxi Province, China.
- [Horn] Berthold K.P Horn. **Closed Form Solution of Absolute Orietation Using Unit Quaternios**. Departament of Electrical Engineering. University of Hawaii. November, 1986.
- [Vidal] Vincent Vidal, Christian Wolf. **Robust Feature Line Extraction on CAD Triangular Meshes**. Université de Lyon. Villeurbanne, France.

-
- [Cullière] Jean Christophe Cullière, Vincent Francois, Jean Marc Drouert. **Automatic Mesh Generation and Transformation for Topology Optimization Methods**. Departement de Génie Mécanique, Université du Québec. Canada.
- [Sethian] J. A. Sethian. **A Fast marching Level Set Method for Monotonically Advancing Fronts**. Department of Mathematics, University of California, Berkeley. November, 1995.
- [Schroeder] William J. Schroeder, Jonathan A. Zarge, William E. Lorensen. **Decimation of Triangle Meshes**. General Electric Company. Schenectady, NY.
- [Turk] Greg Turk. **Re-Tiling Polygonal Surfaces**. Department of Computer Science, University of North Carolina at Chapel Hill .
- [Hugges] Hugues Hoppe. **Progressive Meshes**. Microsoft Research.
- [Huttenlocher] Daniel P. Huttenlocher, Gregory A. Klandermam and William J. Rucklidge. **Comparing Images Using the Hausdorff Distance**.