

DISSERTAÇÃO DE MESTRADO

UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM
CIÊNCIA DA COMPUTAÇÃO

**“Refinamento de superfícies de quadriláteros
baseado em templates”**

ALUNO: Davi Yoshinori Cangussú Nakano
ORIENTADOR: Prof. Dr. Mario Augusto de Souza Liziér

São Carlos
Maio/2013

CAIXA POSTAL 676
FONE/FAX: (16) 3351-8233
13565-905 - SÃO CARLOS - SP
BRASIL

**UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA
DEPARTAMENTO DE CIÊNCIA DA
COMPUTAÇÃO**

DAVI YOSHINORI CANGUSSÚ NAKANO

**REFINAMENTO DE SUPERFÍCIES DE
QUADRILÁTEROS BASEADO EM
TEMPLATES**

**Dissertação apresentado ao
Programa de PósGraduação
em Ciência da Computação,
para obtenção do título de
mestre em Ciência da
Computação**

Orientação: Prof. Dr. Mário Augusto de Souza Liziér

**SÃO CARLOS - SP
2013**

**Ficha catalográfica elaborada pelo DePT da
Biblioteca Comunitária da UFSCar**

N163rs Nakano, Davi Yoshinori Cangussú.
Refinamento de superfícies de quadriláteros baseado em
templates / Davi Yoshinori Cangussú Nakano. -- São Carlos
: UFSCar, 2015.
68 f.

Dissertação (Mestrado) -- Universidade Federal de São
Carlos, 2013.

1. Computação - matemática. 2. Quadrilátero. 3.
Geometria computacional. 4. Malhas (Matemática). 5.
Superfícies (Matemática). 6. *Templates*. I. Título.

CDD: 004.131 (20ª)

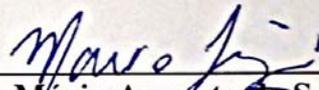
Universidade Federal de São Carlos
Centro de Ciências Exatas e de Tecnologia
Programa de Pós-Graduação em Ciência da Computação

**“Refinamento de Superfícies de Quadriláteros
Baseado em Templates”**

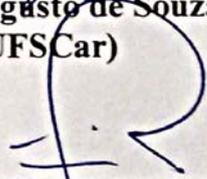
Davi Yoshinori Cangussu Nakano

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação

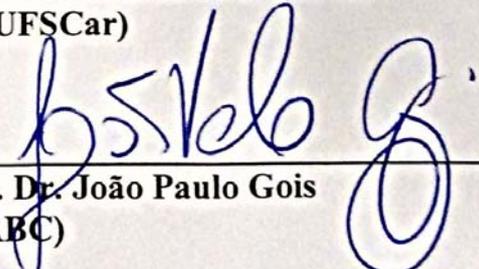
Membros da Banca:



Prof. Dr. Mário Augusto de Souza Liziér
(Orientador - DC/UFSCar)



Prof. Dr. Ricardo José Ferrari
(DC/UFSCar)



Prof. Dr. João Paulo Gois
(UFABC)

São Carlos
Julho/2013

Resumo

A representação computacional de objetos por meio de malhas poligonais é muito utilizada em diversas áreas como modelagem, simulação, engenharia, processo de rendering e diversas outras aplicações CAD (*Computer-Aided Design*) e CAE (*Computer-Aided Engineering*). Essas malhas poligonais são comumente construídas com triângulos e o processamento desses elementos é amplamente estudado, fazendo com que essas técnicas estejam bem consolidadas e robustas. Porém, há um crescente interesse em malhas de quadriláteros, pois proveem características importantes para aplicações como texturização e modelagem com *splines*. Com o crescente aumento do interesse no estudo de superfícies de quadriláteros não estruturadas, há o surgimento de novas técnicas de processamento das mesmas como simplificação e refinamento.

Esta dissertação apresenta um *pipeline* completo para geração indireta de malhas superficiais de quadriláteros, propondo um novo método de refinamento de malhas de quadriláteros. Este novo método consiste em um processo iterativo de subdivisão local, garantindo a consistência da malha a cada iteração e possui como principais características: capacidade de guiar a qualidade dos quadriláteros gerados; controlar o número de elementos gerados; gerar uma família de modelos; controlar a densidade de quadriláteros (adaptatividade) e controlar o alinhamento dos quadriláteros com as linhas de características do modelo. A implementação realizada restringe o refinamento para modelos homeomórficos a uma esfera. Os resultados apresentados ilustram a efetividade da técnica de refinamento e controle da adaptatividade.

Abstract

The computational representation of objects by means of polygonal meshes is widely used in various areas such as modeling, simulation, engineering, rendering process and various other CAD (*Computer-Aided Design*) and CAE (*Computer-Aided Engineering*) applications. These polygonal meshes are commonly built with triangles and processing these elements is widely studied, making that such techniques are well established and robust. However, there is a growing interest in quadrilateral meshes as they provide important characteristics for applications such as texturing and modeling with *splines*. With the increasing interest in the study of unstructured quadrilateral surfaces, there is the rise of new processing techniques related to simplification and refinement.

This dissertation presents a complete *pipeline* for indirect generation of quadrilateral surface meshes, proposing a new refinement method of quadrilateral meshes. This new method is a local iterative subdivision process, ensuring the consistency of the mesh at each iteration and has the following main features: ability to guide the quality of generated quadrilaterals; control the number of generated elements; generate a family of models; control quadrilaterals density (adaptivity) and control the alignment of quadrilaterals lines with features lines of the model. The implementation restricts the refinement to homeomorphic models to a sphere. The results presented illustrate the effectiveness of the refinement technique and adaptivity control.

Sumário

Sumário	iii
Lista de Figuras	v
1 Introdução	1
1.1 Objetivos	4
1.2 Organização	4
2 Malhas de quadriláteros	6
2.1 Geração Indireta	7
2.2 Geração Direta	16
2.3 Remeshing	17
2.4 Simplificação	23
3 Aproximação de superfícies	29
3.1 Superfícies de Bézier	29
3.1.1 Definição	30
3.1.2 Propriedades	32
3.1.3 Bézier Inverso	33
3.2 Parametrização	34
3.3 Geodésicas	36
4 Refinamento de superfícies de quadriláteros	38
4.1 Introdução	39
4.2 Malha base	40
4.3 Refinamento	42
4.3.1 Marcação dos quadriláteros a serem refinados	46
4.3.2 Eliminação dos casos inválidos	48

Sumário

4.3.3	Subdivisão da malha de quadriláteros	53
4.3.4	Processo iterativo	54
5	Resultados	56
6	Conclusões e trabalhos futuros	63
6.1	Trabalhos futuros	64
	Referências Bibliográficas	65

Lista de Figuras

2.1	Subdivisão de triângulos para a formação de 3 quadriláteros [34]	8
2.2	Combinação de triângulos adjacentes para formação de quadriláteros [34] .	8
2.3	Malha de testes da técnica de Burouchaki et al. [7]	10
2.4	Bloco básico e operação de bisseção [48]	11
2.5	Etapas do algoritmo de subdivisão 4-8 [48]	12
2.6	Autofunção Laplaciana, Complexo Morse-Smale, Complexo Otimizado e <i>remeshing</i> semiregular.	12
2.7	Estados possíveis de uma aresta em um front [35]	13
2.8	Etapa de processamento de uma aresta do front [35]	14
2.9	Casos possíveis de operações na árvore gerada a partir do grafo dual da triangulação [37]	15
2.10	Exemplo de decomposição de região poligonal em cinco polígonos convexos [3]	17
2.11	(a) Decomposição da curva em segmentos de linha linear por partes. (b) Construção de um hexágono que contenha todos os pontos da curva. (c) Exclusão de elementos fora ou perto da borda para abrir espaço para geração de quadriláteros com bons ângulos. (d) Aplicação de templates para maximizar a qualidade dos elementos gerados. (e-f) Geração da primeira e segunda camadas a partir do cálculo da bissetriz a cada duas arestas da borda do objeto [29]	18
2.12	Parametrização de uma malha de triângulos, remalhamento no \mathcal{R}^2 e transposição para o \mathcal{R}^3 [20]	20
2.13	Malha de entrada, segmentação em regiões, computação das curvas de energia mínimas de cada região e malha resultante, respectivamente [31]	21

Lista de Figuras

2.14	Malha inicial e features do objeto, decomposição em células para a construção de uma triangulação base, utilização dos templates na triangulação e otimização da localização dos vértices no mapeamento dos mesmos para a superfície original [22]	22
2.15	Resultados do algoritmo proposto por Bommer et al. À esquerda estão evidenciadas as singularidades da malha resultante. À direita está a malha resultante [6]	24
2.16	Exemplo de simplificação de uma malha com níveis de detalhe decrescentes e preservação da topologia do objeto [11]	25
2.17	Exemplo de malhas resultantes da simplificação [46]	27
2.18	Exemplo dos tipos de operações locais [46]	28
3.1	Patch retangular de Bézier [41]	30
3.2	Polinômios de Bernstein no caso cúbico [17]	31
3.3	Exemplo de fecho convexo de curvas de Bézier	32
4.1	Visão geral do pipeline implementado.	39
4.2	Malha base de quadriláteros destacada da superfície de triângulos a ser aproximada.	41
4.3	Regiões da malha de triângulos correspondentes a cada face da malha base.	41
4.4	Modelos de subdivisão que não inserem vértices ruins na malha.	43
4.5	Todas as situações possíveis de quadriláteros com arestas a serem subdivididas. A letra 'D' representa um vizinho que foi subdividido e que compartilha uma aresta que precisa ser subdividida com o quadrilátero central.	45
4.6	Modelos de subdivisão onde há a inserção de um vértice extraordinário de grau 3 no centro do quadrilátero.	46
4.7	Imagem superior esquerda: Nenhuma aresta marcada para subdivisão (NNNN). Imagem inferior esquerda: Dois vizinhos opostos marcam as arestas em comum para subdivisão (DNDN ou NDND). Imagem inferior direita: Dois vizinhos adjacentes marcam as arestas em comum para subdivisão (DDNN, NDDN, NNDD ou DNND). Imagem superior direita: Os quatro vizinhos marcam as arestas em comum para subdivisão (DDDD).	47
4.8	Exemplo de malha antes do processamento dos casos inválidos. As diferentes cores representam as marcações de templates que cada quadrilátero recebeu de acordo com seus vizinhos.	51
4.9	Utilizando a mesma malha da figura 4.8, esta imagem representa a malha pós processamento dos casos inválidos.	52
4.10	Exemplo de um <i>patch</i> que passou pelo processo de parametrização.	53

Lista de Figuras

4.11	Exemplo do processo iterativo desde a malha base até a malha final, com 10 iterações (partindo do modelo inicial, acima e a esquerda).	55
5.1	Exemplo do processo iterativo desde a malha base até a malha final, com 10 iterações.	58
5.2	Resultado com o modelo da figura 5.1 juntamente com um histograma de qualidade.	58
5.3	Exemplo do processo iterativo desde a malha base até a malha final, com 10 iterações.	59
5.4	Resultado com o modelo da figura 5.3 juntamente com um histograma de qualidade.	60
5.5	Exemplo do processo iterativo desde a malha base até a malha final, com 10 iterações.	61
5.6	Resultado com o modelo da figura 5.5 juntamente com um histograma de qualidade.	62
5.7	Zoom do modelo Santa antes e depois da suavização.	62

Introdução

Diversos problemas computacionais envolvem dados geométricos ou que podem ser descritos geometricamente. O estudo e desenvolvimento de modelos de representação, algoritmos e ferramentas para manipular esses dados geométricos tem se tornado uma importante área de pesquisa da ciência da computação, onde as técnicas de geometria computacional fornecem suporte em aplicações que vão desde a área médica até o entretenimento digital. Em geral, estes objetos são representados computacionalmente por funções implícitas, funções paramétricas, nuvem de pontos ou na forma de decomposição de polígonos.

Na representação por funções implícitas, a superfície do objeto é representada por uma constante na imagem da função, $f(x, y, z) = c$, sendo em geral, $c = 0$. O domínio é dividido em regiões de superfície, interior e exterior do objeto. Na região interna do objeto $f(x, y, z) > c$ e na região externa, ou seja, no meio em que o objeto está inserido, $f(x, y, z) < c$. Apesar deste modelo de representação não exigir uma estrutura de dados muito elaborada, representar um objeto com uma superfície complexa é em muitas situações uma tarefa custosa. Para auxiliar nesta modelagem, é muito utilizado um pro-

Introdução

cesso de composição de primitivas simples na formação destas superfícies mais complexas. Tal processo é chamado de CSG (*Constructive Solid Geometry*). Uma grande vantagem da modelagem por funções implícitas é a consulta direta se um determinado ponto está no interior, exterior ou na superfície do objeto. Esta informação é obtida diretamente pela avaliação da função f . Como desvantagem, podemos citar a ausência de relação de vizinhança, onde não conseguimos obter diretamente qualquer informação sobre a região próxima a um ponto qualquer. Operações como percorrer a superfície de um objeto se tornam muito ineficientes computacionalmente.

Já na representação por funções paramétricas, a superfície do objeto é parametrizada e representada pela imagem destas funções. Uma avaliação na função $f(r, s) = (x, y, z)$ retorna um ponto sobre a superfície do objeto. Superfícies *splines*, como os retalhos de *Bézier*, são as funções paramétricas mais utilizadas neste modelo de representação. A vantagem desta abordagem é que temos uma representação da vizinhança muito eficiente, podendo facilmente reconstruir toda a superfície, apenas criando uma amostragem no espaço paramétrico. Por outro lado, operações como determinar se um ponto está fora, sobre ou dentro de um objeto, são bastante custosas.

A modelagem por nuvem de pontos, consiste na representação da superfície do objeto por um conjunto de pontos, sem qualquer informação de proximidade entre eles. Este conjunto de pontos é, em geral, obtido de equipamentos de escaneamento de objetos tridimensionais. A computação de partículas como fogo e água é inerente desse tipo de representação, mas a manipulação deste conjunto de pontos desorganizado exige uma estruturação por meio de uma estrutura de dados espacial. Uma estrutura largamente utilizada é a *Kd-Tree* – uma árvore binária, ou seja, uma estrutura recursiva onde os registros são armazenados como nós de uma árvore.

A modelagem por polígonos aproxima a superfície do objeto por meio de um conjunto de polígonos. Estes polígonos precisam cobrir toda a superfície do objeto e qualquer par de polígonos precisam ter intersecção vazia ou um simplexo de dimensão inferior. Esse conjunto de polígonos é chamado de *malha*.

Introdução

Uma malha superficial pode ser definida como um conjunto de elementos geométricos (normalmente triângulos ou quadriláteros) dispostos em forma para representar a superfície de um objeto. Diversas técnicas e algoritmos para a geração de malhas superficiais de triângulos já foram largamente exploradas na literatura e suas bases são bastante sólidas. Para triângulos, é possível utilizar a classificação feita em [34] para identificar os grupos de algoritmos baseados em seu conceito central, ou seja: *Octrees*, Triangulação de *Delaunay* e *Advancing Front*.

Embora grande parte das pesquisas em malhas presentes na literatura utilizem triângulos, nos últimos anos houve um crescimento no interesse de desenvolvimento de algoritmos para malhas de quadriláteros (i.e. malhas compostas somente de quadriláteros). J. Daniels et al. [11] atribuem essa atração pelas malhas de quadriláteros pela habilidade desses elementos naturalmente descrever campos direcionais de curva principal, curvas que são sempre tangentes a direção principal, e por dividir um domínio comum com soluções de parametrização de superfícies. Essas características possibilitam um número importante de aplicações, incluindo texturização e modelagem com *splines*.

A estruturação das malhas varia de acordo com a regularidade de seus vértices (pontos) que podem estar dispostos de maneira estruturada ou não estruturada. Essa classificação se dá pela conectividade entre os vértices, suas relações de vizinhança e a forma com que os mesmos são armazenados na memória do computador. Assumindo uma matriz que contenha os índices de cada célula, uma malha estruturada possui a característica de ter uma conexão implícita entre os vértices, ou seja, dado um vértice seus vizinhos podem ser obtidos por um incremento ou decremento deste índice. Dessa maneira, esse tipo de estrutura faz com que haja menos acessos a memória e por consequência se torne computacionalmente menos custoso. Porém, sua construção é uma tarefa bastante complexa e em muitos casos a opção não estruturada é procurada. Malhas não estruturadas, apesar da necessidade de armazenamento das relações de vizinhança, é uma alternativa relativamente menos complexa no que tange a sua construção. O processo de construir uma estruturada, ou quase estruturada, a partir de uma malha arbitrária é chamado de

1.1 Objetivos

Este trabalho apresenta um *pipeline* completo para geração de malhas de quadriláteros superficiais a partir de uma malha arbitrária e não estruturada de triângulos. Como “arbitrária” entende-se que não é exigida qualidade mínima para os triângulos da malha superficial de entrada. A implementação realizada introduziu algumas restrições que são apresentadas neste texto, incluindo o tratamento apenas de modelos homeomórficos a uma esfera, em consequência ao não tratamento de bordos e na definição do modelo inicial de quadriláteros. Segundo o *pipeline* proposto, uma nova técnica de refinamento é apresentada.

Os objetivos principais de trabalho são alcançar, com esta sequência de etapas, as seguintes características em um gerador superficial de malhas de quadriláteros: capacidade de guiar a qualidade dos quadriláteros gerados; controlar o número de elementos gerados; gerar uma família de modelos; controlar a densidade de quadriláteros (adaptatividade) e controlar o alinhamento dos quadriláteros com as linhas de características do modelo.

1.2 Organização

Este texto está organizado como segue:

- *Capítulo 2*: Neste capítulo encontra-se a revisão bibliográfica dos métodos e técnicas de geração de malhas de quadriláteros diretas e indiretas, além de um resumo das técnicas de simplificação e remeshing de malhas de quadriláteros.
- *Capítulo 3*: Este capítulo tem foco na aproximação de superfícies, apresentando as definições de curva e superfície de bézier, bem como suas propriedades. Além disso há uma seção dedicada a parametrização e uma dedicada a geodésicas.
- *Capítulo 4*: Neste capítulo será exposto o pipeline desenvolvido e, principalmente, todo o processo iterativo. Essa exposição se dará através da discussão sobre a

Introdução

geração da malha base inicial, do algoritmo de refinamento, da marcação dos quadriláteros e de seus *templates* associados, da eliminação dos templates inválidos e do método de subdivisão.

- *Capítulo 5*: Este capítulo apresenta os resultados do uso do algoritmo implementado através da exibição das malhas resultantes do processo iterativo.
- *Capítulo 6*: O último capítulo mostra a conclusão desta trabalho bem como uma discussão sobre os trabalhos futuros.

Malhas de quadriláteros

Uma malha M é uma aproximação de um domínio por regiões poligonais disjuntas, sem buracos e que podem se intersectar por um vértice ou aresta. Estas regiões poligonais são chamadas de *elementos da malha* ou *células*. Seja a malha M formada pelo conjunto de n_v vértices $V_M = \{v_0, v_1, \dots, v_{n_v-1}\}$ e o conjunto de n_e arestas $E_M = \{e_0, e_1, \dots, e_{n_e-1}\}$, considerando $e_i = (v_j, v_k)$, $v_j \neq v_k$ e $v_j, v_k \in V_M$, cada elemento da malha (região poligonal R) é formado pelo conjunto V_R e E_R , sendo $V_R \subseteq V_M$, $E_R \subseteq E_M$ e para cada $e_i \in E_R$, $e_i = (v_a, v_b)$, $v_a \neq v_b$ e $v_a, v_b \in V_R$.

Uma malha M pode ser classificada quanto a sua estrutura, tipo de elemento e geometria:

- **Estrutura:** a estrutura de M é referente a valência dos seus vértices. A valência de um vértice $v_i \in V_M$ é o número de arestas $e \in E_M$, onde $e = (v_j, v_k)$ com $i = j$ ou $i = k$. Se este número de valência é igual para todos os vértices v de V_M , a malha é considerada *regular*. Do contrário, é considerada *irregular*;
- **Tipo de elemento:** se todos os conjuntos V_R , referentes a cada elemento da malha,

possuírem o mesmo número de vértices, é dito que a malha é *homogênea* e os elementos são do mesmo tipo. Do contrário, é considerada *heterogênea* ou *mista*. Em geral, os tipos de elementos mais utilizadas são os triângulos, onde $|V_R| = |E_R| = 3$, ou quadriláteros, onde $|V_R| = |E_R| = 4$;

- **Geometria:** a geometria da malha é relacionada as suas propriedades métricas. Uma malha M é considerada *uniforme* se M for regular e todas as arestas $e \in E_M$ tiverem o mesmo tamanho. Do contrário, a malha é considerada *não uniforme*.

Vale ressaltar que a técnica proposta neste trabalho possui foco no desenvolvimento de uma nova técnica de geração indireta de malhas superficiais de quadriláteros irregular, ou seja, a valência dos vértices da malha gerada M_o não será a mesma para todo $v \in V_{M_o}$. Nosso objetivo é minimizar o número de vértices com valência diferente de 4.

Na seção seguinte são descritos os métodos indiretos de geração de malhas de quadriláteros. Na seção 2.2, são apresentados métodos diretos de geração de malhas de quadriláteros. Na seção 2.3 há uma revisão das técnicas recentes de *remeshing* visando diferentes abordagens para o problema. Na seção 2.4 são mostradas diferentes propostas de métodos de simplificação de malhas de quadriláteros.

2.1 Geração Indireta

A geração de malhas de quadriláteros de maneira indireta é definida como a construção de uma malha inicial de triângulos, a partir de um conjunto de pontos, e a aplicação de um algoritmo para converter esta malha de triângulos em uma de quadriláteros. Esta abordagem tem sido muito utilizada, pois existem algoritmos de geração de malhas de triângulos robustos, estando bem consolidados e largamente explorados na literatura.

Um dos métodos mais simples de geração indireta de malhas de quadriláteros é a divisão de cada triângulo em três quadriláteros (Catmull-Clark [9]). Esse método garante uma malha composta somente de quadriláteros, porém há um alto número de vértices extraordinários (valência diferente de 4), resultando em uma malha como mostra a figura

Malhas de quadriláteros

2.1. Uma alternativa a essa técnica, é a combinação de triângulos adjacentes para formar cada quadrilátero. Apesar de diminuir o número de vértices extraordinários, a malha resultante não é necessariamente homogênea, pois podem sobrar triângulos isolados, como ilustra a figura 2.2. Estas ideias são utilizadas até hoje como parte de técnicas mais avançadas.

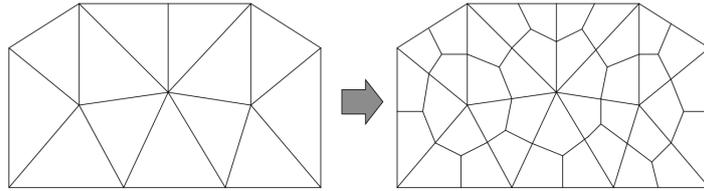


Figura 2.1: Subdivisão de triângulos para a formação de 3 quadriláteros [34]

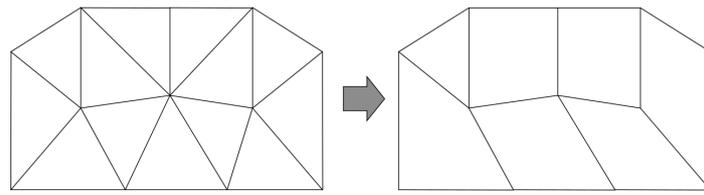


Figura 2.2: Combinação de triângulos adjacentes para formação de quadriláteros [34]

Em geral, as técnicas indiretas de geração de malhas tem a vantagem de serem mais rápidas, pois todas, ou a maioria, de suas operações são locais. Em contrapartida, as técnicas diretas utilizam cálculos globais.

Burouchaki et al. [7] propõem um método baseado na conversão da malha de triângulos para uma malha mista, utilizando a definição de um domínio Ω no \mathbb{R}^2 que contenha a estrutura de Riemann [26] para criar uma malha onde cada aresta possua tamanho unitário. Essa malha resultante é chamada de malha unitária do domínio Ω . A partir dela, é feita uma conversão em uma malha unitária de quadriláteros, mas que ainda podem possuir triângulos isolados – essa malha é denominada *quad-dominante*. Por fim, são utilizados procedimentos de junção de triângulos e conversão de triângulos em quadriláteros através de subdivisão baricêntrica. Este processo de geração da malha é baseado no uso da estrutura de Riemann, onde esta é definida utilizando uma interpolação do campo discreto de

Malhas de quadriláteros

métricas, associadas com os vértices da malha de Ω . O processo de construção da malha de Ω com respeito ao espaço controle (i.e. estrutura resultante, definida pelos vértices da malha de Ω e o campo discreto de métricas) segue duas etapas: a geração da malha unitária da borda de Ω e a geração da malha unitária de Ω utilizando a malha unitária de sua borda como entrada. Depois de construída, essa malha de triângulos é convertida em uma malha de quadriláteros tendo como ideia principal o pareamento de triângulos para a formação dos quadriláteros, visando a qualidade dos mesmos, além de minimizar o número de triângulos que não puderam ser pareados. A métrica de qualidade utilizada foi a observação dos ângulos dos elementos. O ângulo buscado tem valor de $\frac{\pi}{2}$ radianos e quanto maior a distância desse valor, menor é a qualidade do quadrilátero.

Para realizar a conversão da malha de triângulos para a malha quad-dominante, é observado a aresta que um par de triângulos adjacentes dividem para que com a sua remoção seja criado um elemento quadrilateral. É feito então um processo iterativo guiado por uma lista dessas arestas, ordenada em ordem decrescente de acordo com uma qualidade definida. Este método não é capaz de gerar sempre uma malha homogênea. Desta forma, para os triângulos isolados é feito a subdivisão baricêntrica (proposta por Catmull-Clark [9]), que consiste basicamente da adição de três vértices no meio das três arestas que formam o triângulo e um em seu baricentro. Ao criar as arestas que ligam esses vértices, os triângulos serão convertidos para quadriláteros. Para manter a consistência da malha, um processo similar é aplicado a todos os quadriláteros fazendo com que haja um refinamento uniforme em toda a malha, tornando-a puramente formada por elementos quadrilaterais.

Outra abordagem de geração indireta é a utilização de métodos de tesselação. Estes métodos refinam a malha em polígonos idênticos e regulares, produzindo malhas isomórficas as originais, considerando assim que o refinamento deixa uma malha *invariante*. Um exemplo de refinamento deste tipo é a subdivisão 4-8, criada por Velho [48]. Essa subdivisão utiliza somente a informação de conectividade da malha, e não a posição dos vértices, aplicando o operador *face split*. A subdivisão consiste na inserção de um

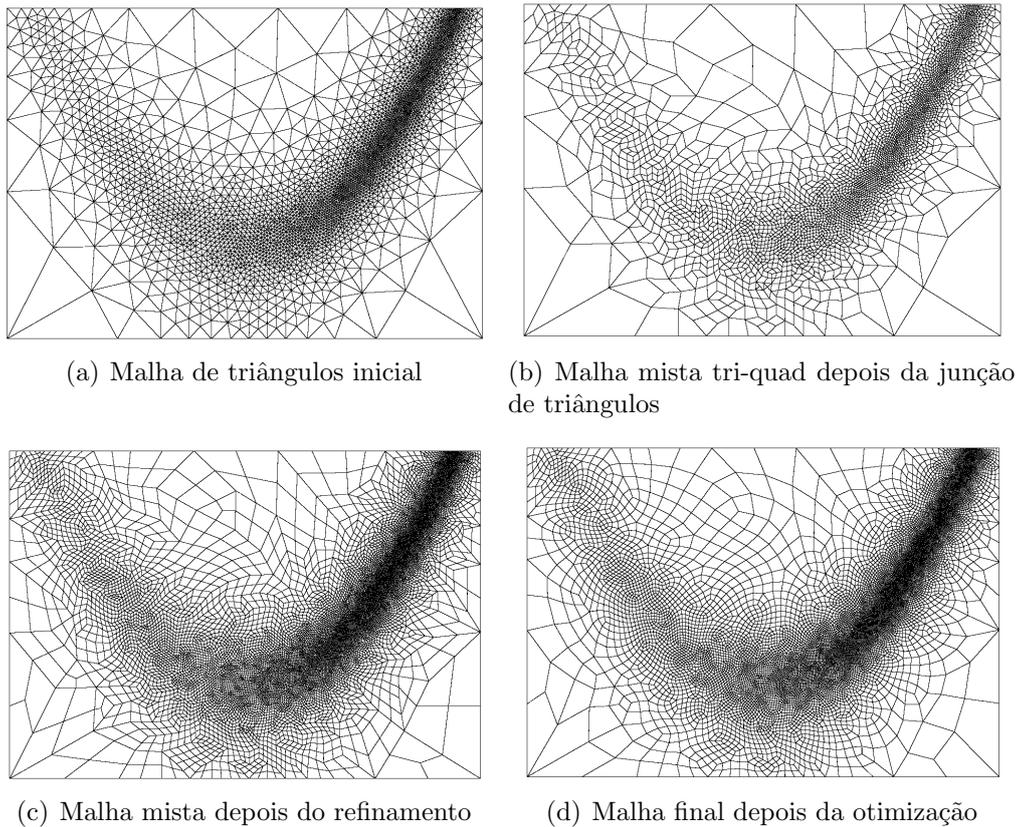


Figura 2.3: Malha de testes da técnica de Burouchaki et al. [7]

vértice em cada face e em cada aresta, subdividindo a face em n quadriláteros onde n é o número de vértices da face. Depois de alguns passos dessa subdivisão, a malha refinada tem a mesma estrutura local de uma malha regular de quadriláteros, com exceção de alguns vértices herdados dos passos anteriores.

O algoritmo proposto utiliza uma estrutura denominada *basic block* ou bloco básico – um par de triângulos que forma um bloco quadrilateral dividido por uma diagonal. A aresta compartilhada entre os dois triângulos de um bloco básico é chamada de *aresta interior* e as demais arestas são chamadas de *arestas exteriores* e essa estrutura forma uma *quadrangulação triangulada*. Levando em consideração que cada triângulo tem uma aresta interior e duas arestas exteriores, é possível utilizar uma operação de refinamento denominada bisseção onde para cada bloco básico é inserido um vértice na aresta interior e cada face de triângulo é subdividida pela ligação desse vértice com o vértice oposto da face. É válido notar que em cada bloco básico há dois vértices que estão na diagonal que

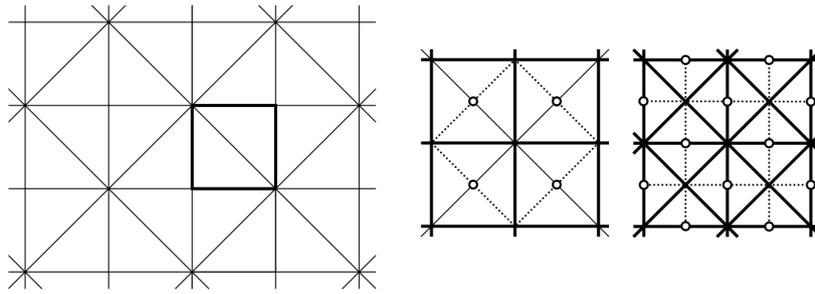


Figura 2.4: Bloco básico e operação de bisseção [48]

a subdivide e dois vértices não estão. Sendo um vértice do tipo 1 participante de pelo menos uma diagonal de um bloco básico adjacente a ele e, caso contrário, um vértice do tipo 2. São enumeradas algumas propriedades:

1. todos os vértices adicionados por um único passo deste refinamento são do tipo 1 com valência 4;
2. um único passo do refinamento converte todos os vértices do tipo 1 para o tipo 2 e sua valência não será maior que o seu dobro;
3. a valência dos vértices do tipo 2 não são alteradas pelo refinamento

A figura 2.4 ilustra a estrutura de bloco básico e a operação de bisseção.

Para a obtenção da malha mista de triângulos e quadriláteros, Velho encontra um conjunto de blocos básicos baseados no tamanho da aresta e remove todas as arestas internas dos mesmos. Como esse casamento de triângulos para a formação de blocos básicos normalmente não é perfeita, é efetuado uma subdivisão baricêntrica dos triângulos restantes, que resulta em três novos triângulos, e dos quadriláteros resultantes, que resulta em 4 triângulos. Com esse pré-processamento, a malha resultante é composta inteiramente de blocos básicos, como ilustra a figura 2.5. A subdivisão é chamada de 4-8 pois os vértices gerados a partir do refinamento proposto possuem valências 4 ou 8. Os demais vértices que já estavam na malha possuem valências que variam entre, sendo n a valência anterior: se $n = 3$, a nova valência será 6; se $4 \leq n \leq 8$, a nova valência será 8; se $n > 8$ e ímpar, a nova valência será $n + 1$; e se $n > 8$ e par, a nova valência será n .

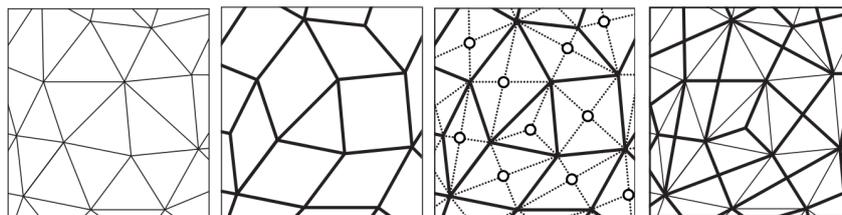


Figura 2.5: Etapas do algoritmo de subdivisão 4-8 [48]

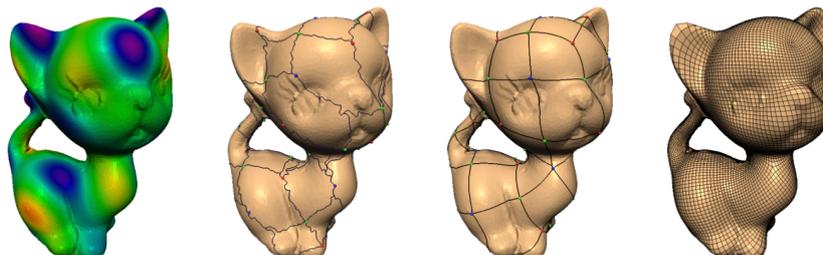


Figura 2.6: Autofunção Laplaciana, Complexo Morse-Smale, Complexo Otimizado e *re-meshing* semiregular.

[14]

Além das técnicas descritas, é possível utilizar uma abordagem espectral para a construção de malhas quadrangulares. Dong et al. [14] baseia sua técnica no teorema de Morse, onde para quase toda função o complexo Morse-Smale, consistindo de *ridge lines* que se estendem desde as suas selas à sua extrema, forma regiões quadrangulares. A função real escolhida para espaçar essas regiões de maneira uniforme sobre a superfície é uma função de forma harmônica de frequência apropriada, calculada como um autovetor da matriz Laplaciana da malha de entrada. O espectro completo da malha define duas famílias de complexos: o primitivo Morse-Smale e seus complexos *quasi-dual* e a qualidade da malha resultante é dependente da escolha desse complexo que, no artigo, foi baseada em distorção paramétrica. As malhas resultantes de cada passo do processo estão representadas na figura 2.6. A grande contribuição de seu método foi demonstrar que harmônicos de forma distribuem seu extrema e então servem como funções ideais para gerar esse domínio quadrangulado.

Owen et al. [35] propôs um método (denominado Quad-Morphing ou Q-Morph) que aborda o problema de geração de malhas de quadriláteros a partir de malhas de triângulos

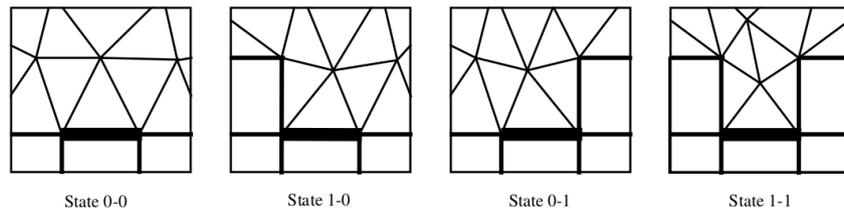


Figura 2.7: Estados possíveis de uma aresta em um front [35]

baseado em *advancing front* (avanço de fronteira). Técnicas baseadas em *advancing front* consistem na escolha de um conjunto de arestas presentes na borda do domínio (*fronts*) e na junção sistemática de triângulos para formar quadriláteros. A cada junção efetuada, o *front* avançou para o interior do domínio e então o conjunto de arestas presentes no novo *front* é recalculado e reordenado de forma a habilitar o algoritmo à realizar a próxima junção e preencher o domínio totalmente com quadriláteros.

No método Q-Morph, dado um domínio triangulado, o *front* inicial é formado por qualquer aresta na triangulação que é adjacente a somente um triângulo. Essas arestas então são classificadas de acordo com seu estado a cada iteração que forma um quadrilátero. O que define os estados das arestas do *front* são os ângulos existentes entre as mesmas. A figura 2.7 ilustra as possíveis classificações de uma aresta de um front.

O processamento dessas arestas do *front* é feita individualmente e cada uma formará um novo quadrilátero. Primeiramente é feita uma verificação para casos especiais como a presença de ângulos pequenos no *front* local. Depois disso são definidas as arestas laterais do quadrilátero que podem ser obtidas através de uma aresta já existente, através de mudança da diagonal de triângulos adjacentes ou através da subdivisão de triângulos para criar arestas novas. A última aresta é criada pela alteração da triangulação local pela mudança da diagonal de triângulos adjacentes para forçar uma aresta entre os vértices das arestas laterais do novo quadrilátero. Os triângulos em seu interior são então fundidos e ocorre uma suavização local para melhorar a qualidade dos elementos. A figura 2.8 ilustra esta etapa de processamento.

O *front* avança pela remoção de arestas que possuem dois quadriláteros adjacentes e

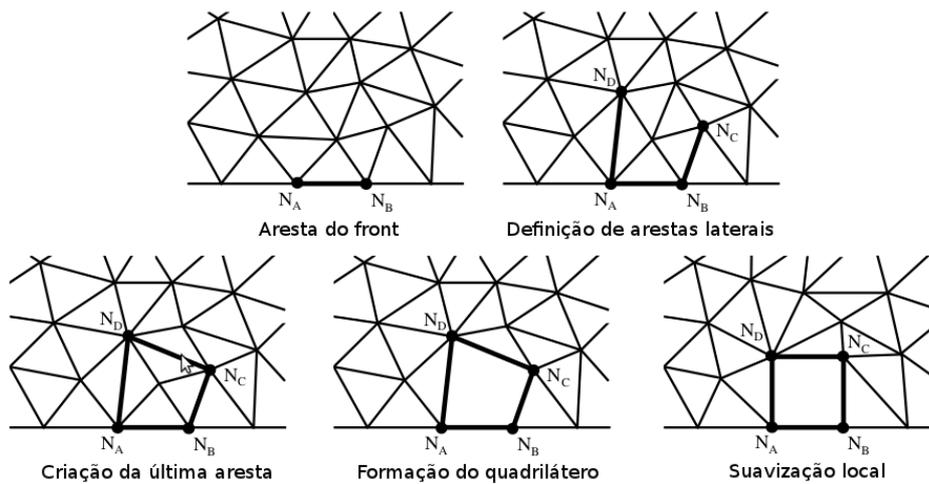


Figura 2.8: Etapa de processamento de uma aresta do front [35]

pela adição de arestas que possuem um triângulo e um quadrilátero adjacente. As arestas do *front* então são novamente reclassificadas pelo seu estado e o processo continua até que a malha seja composta somente por quadriláteros, onde ocorre uma limpeza topológica (transformações locais nos quadriláteros que visam melhorar a valência dos nós bem como a qualidade dos elementos) e uma suavização final.

No intuito de gerar malhas de quadriláteros estritamente convexas (onde cada um dos quatro ângulos de todos os quadriláteros são menores que 180 graus) que sejam adequadas para a análise de elementos finitos, Ramaswami et al. [37] desenvolveram um algoritmo de conversão de malhas triangulares no plano para malhas de quadriláteros estritamente convexas que contém todos os vértices da malha original e alguns vértices extras, chamados de pontos de Steiner.

O algoritmo começa selecionando uma região poligonal dentro da malha e realizando uma triangulação qualquer dessa região. É construído então uma árvore geradora a partir do grafo dual dessa triangulação onde cada nó da árvore representa um triângulo.

Nessa árvore podem ser feitas duas operações, conforme ilustra a figura 2.9. Na esquerda da operação (a), é mostrado uma situação em que há um avô ($r_v = par(par(v))$) que possui um filho ($par(v)$) e este último possui um filho v . Ao centro é ilustrada a representação dos mesmos na triangulação sendo $ele(x)$ o elemento referente ao nó x .

Malhas de quadriláteros

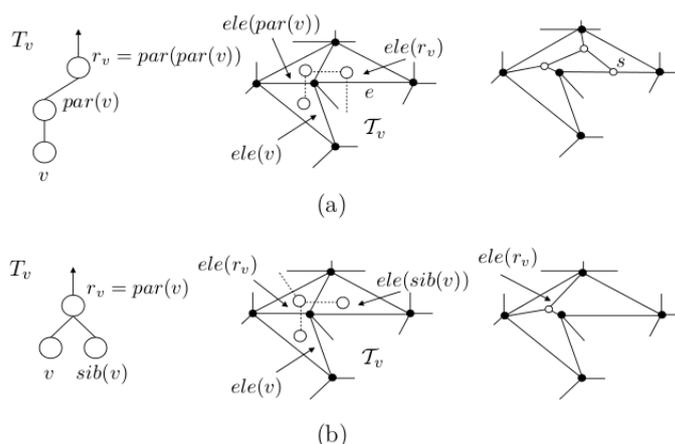


Figura 2.9: Casos possíveis de operações na árvore gerada a partir do grafo dual da triangulação [37]

Na direita, cada vez que uma aresta é atravessada neste processo, é criado um ponto de Steiner. Ao final desta operação, os nós participantes da mesma são excluídos da árvore e esta última é atualizada.

Na esquerda da operação (b), é mostrado uma situação em que há um pai ($r_v = par(v)$) que possui dois filhos (v e $sib(v)$). Ao centro há a representação do fragmento da árvore na malha e na direita o resultado do processamento dessa operação: é adicionado um ponto de Steiner no triângulo referente ao nó pai, neste caso, levando a uma quadrangulação parcial do domínio. Caso esse triângulo seja representado por um nó neto na operação representada na figura 2.9 (a), haverá uma quadrangulação total do domínio.

Os nós são retirados da árvore em uma abordagem *bottom-up*, ou seja, de baixo para cima ou das folhas até a raiz até que não haja mais como efetuar essas operações. É possível verificar com facilidade que esse processo pode ter alguns casos especiais onde existem triângulos excedentes ou pontos de Steiner são inseridos na aresta dos elementos formando quadriláteros degenerados ou pentágonos. Esses casos especiais são tratados com a inserção e manipulação de pontos de Steiner.

A seguir são apresentadas algumas técnicas diretas de geração de malhas de quadriláteros.

2.2 Geração Direta

A geração de malhas de quadriláteros de maneira direta apresenta maneiras de quadrangular um determinado domínio para gerar uma malha de quadriláteros diretamente, sem passar por uma etapa de construção de malha de triângulos. Em geral, as técnicas diretas podem ser subdivididas em dois grupos. No primeiro grupo estão os métodos que se apoiam em alguma forma de decomposição do domínio em regiões mais simples e que podem ser resolvidas por uma série de templates [2] [44] [45] [33]. No segundo grupo estão os métodos que utilizam a ideia de avanço de fronteira para colocação direta de nós e elementos [51] [5] [8]. A seguir são comentadas três técnicas de geração direta de malhas de quadriláteros.

B. Joe [3] apresenta uma técnica de quadrangular um domínio utilizando algoritmos de decomposição poligonal, ou seja, subdivide um polígono em outros polígonos convexos. As principais entradas necessárias para essa técnica são as curvas de borda do polígono a ser quadrangulado e o número desejado de quadriláteros. Essa técnica é baseada em três estágios. No primeiro estágio, o polígono é decomposto em quadriláteros convexos de forma a maximizar os ângulos internos. No segundo estágio é construída uma função de distribuição da malha e ocorre uma nova subdivisão dos polígonos convexos criados no estágio anterior em novas regiões convexas menores baseadas na função criada. No terceiro estágio é utilizada uma versão adaptada para quadriláteros do algoritmo proposto pelo mesmo autor [23] onde foi utilizado um grid quase uniforme de espaçamento pré-definido para a inserção de vértices dentro do polígono.

Liang et al. [29] utiliza uma estratégia baseada em hexágonos. Neste algoritmo, os autores criam uma estrutura de árvore hexagonal balanceada e adaptativa de forma a verificar as curvas do objeto para obter três possíveis casos. No primeiro caso, os elementos podem estar fora da borda do objeto, no segundo caso os elementos possuem parte da borda e no terceiro caso estão totalmente dentro do objeto. A princípio os dois primeiros casos são descartados e os elementos do terceiro caso formam uma malha de

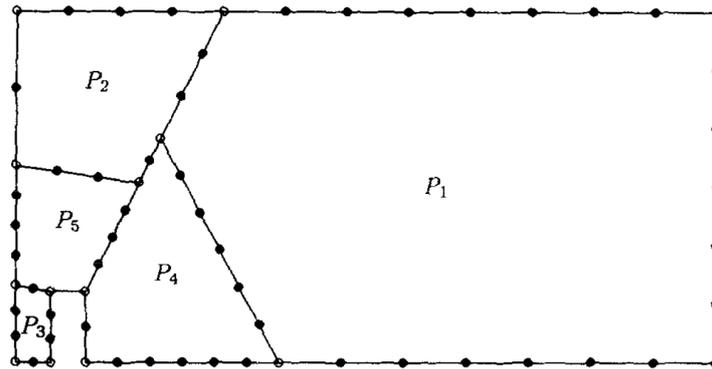


Figura 2.10: Exemplo de decomposição de região poligonal em cinco polígonos convexos [3]

hexágonos inicial que é passível de ser quadrangulada de maneira trivial. Posteriormente as bordas presentes nos elementos do segundo caso são tratados pela criação de templates de quadrangulação de hexágonos. Depois de tratadas, essas duas últimas partes citadas são conectadas de maneira a formar uma malha de quadriláteros.

Sarrate et al. [40] aborda o problema através da decomposição recursiva do domínio. Essa recursão ocorre através de linhas divisórias, conectadas por dois vértices de borda, que dividem o domínio em duas partes a cada iteração. A escolha da melhor linha divisória é baseada numa função objetiva com restrições, que incluem o número total de linhas entre dois vértices não consecutivos.

As técnicas a seguir foram categorizadas como sendo métodos de *Remeshing*, embora estas classificações sejam muito discutíveis.

2.3 Remeshing

Diversas aplicações são beneficiadas pelo processo de *remeshing* (re-malhagem ou regeneração) como a simulação, modelagem, animação, compressão, engenharia reversa e recuperação das características de um objeto. Apesar de muito utilizada, não há uma definição exata e universalmente aceita do termo *remeshing*. Porém, o objetivo geral do *remeshing* é, a partir de uma malha considerada "ruim", gerar uma nova malha que se aproxime do objeto real de maneira aceitável e que respeite critérios de qualidade baseados

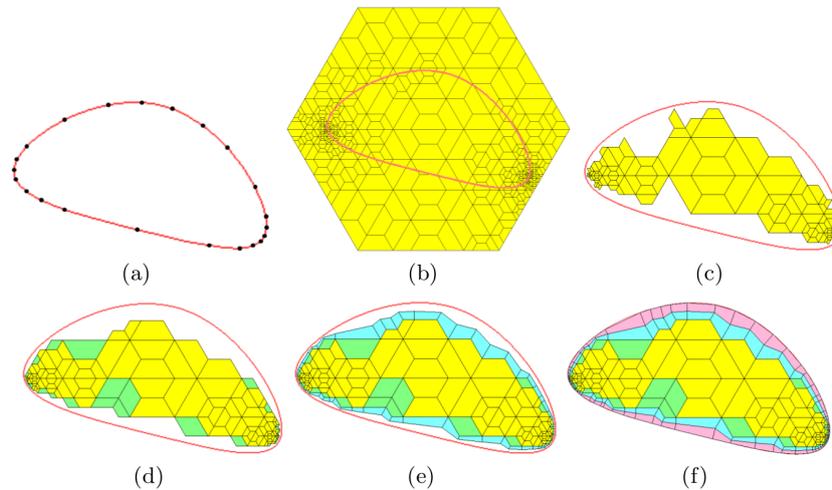


Figura 2.11: (a) Decomposição da curva em segmentos de linha linear por partes. (b) Construção de um hexágono que contenha todos os pontos da curva. (c) Exclusão de elementos fora ou perto da borda para abrir espaço para geração de quadriláteros com bons ângulos. (d) Aplicação de templates para maximizar a qualidade dos elementos gerados. (e-f) Geração da primeira e segunda camadas a partir do cálculo da bissetriz a cada duas arestas da borda do objeto [29]

normalmente em amostragem dos vértices, regularidade e ou qualidade dos elementos.

Realizar um *remeshing* não é uma tarefa trivial. Segundo Alliez et al. [1] gerar uma malha através do *remeshing* traz dificuldades como:

- **Validação** da malha (i.e. ser um *manifold* simples);
- **Qualidade** dos elementos para prover estabilidade numérica (requerida em simulações e no cálculo de equações diferenciais por exemplo)
- **Fidelidade**, deve aproximar a forma do objeto de entrada da melhor maneira possível;
- **Entrada discreta**, a entrada é um conjunto discreto de pontos aproximado da forma do objeto real, dificultando operações como estimações de curvatura e localização de atributos;
- **Número de dados**, para não perder nenhum detalhe, os equipamentos modernos de escaneamento 3D tendem a gerar um grande número de dados, muitos deles

redundantes;

- **Incerteza**, o equipamento de escaneamento pode ter algum ruído que possa ter interferido na correta aquisição da forma do objeto;
- **Correspondência**, encontrar a localização correspondente de um novo vértice na malha de entrada.

Além dessas dificuldades, é esperado que os algoritmos de *remeshing* tenham a possibilidade de gerar malhas de diferentes resoluções, tenham baixa complexidade (existem malhas que possuem centenas de milhares de faces) e que possuam garantias teóricas, quanto a topologia por exemplo.

Hormann et al. [20] contorna o problema de realizar um *remeshing* de uma malha de triângulos para uma malha regular de quadriláteros no \mathbb{R}^3 pelo mapeamento do mesmo no espaço bidimensional \mathbb{R}^2 através de uma função de deflação $f : \mathbb{R}^3 \rightarrow \mathbb{R}^2$, para então efetuar o *remeshing* e aplicar uma função de inflação $F = f^{-1}$ para voltar ao \mathbb{R}^3 . Apesar dos únicos objetos fechados que podem ser inteiramente modelados com uma malha de quadriláteros regulares não degenerados (i.e. quadriláteros não convexos) serem o torus e a garrafa de Klein, qualquer objeto pode ser particionado em malhas com bordas e estas podem ser utilizadas no algoritmo apresentado. Para realizar a parametrização o autor utilizou o método apresentado por Floater [18] que gera uma parametrização sem sobreposição de elementos. Com o problema reduzido ao \mathbb{R}^2 , é necessário encontrar uma malha regular de quadriláteros no plano Q_P cujas bordas coincidem com as bordas da malha de triângulos T_P no plano. Para encontrar Q_P , foram escolhidos quatro vértices de borda de T_P pelo menor ângulo interno tendo em vista o polígono formado pelos vértices de borda, formando assim um retângulo. Os demais vértices de borda foram acrescentados pelos lados do retângulo pela parametrização para curvas chamada de *chord length*. A figura 2.12 mostra a ideia do algoritmo através de um exemplo.

Se este método não parametrizar a borda de T_P corretamente, é possível ainda considerar o retângulo inicial citado anteriormente como uma malha grossa de quadriláteros Q_G

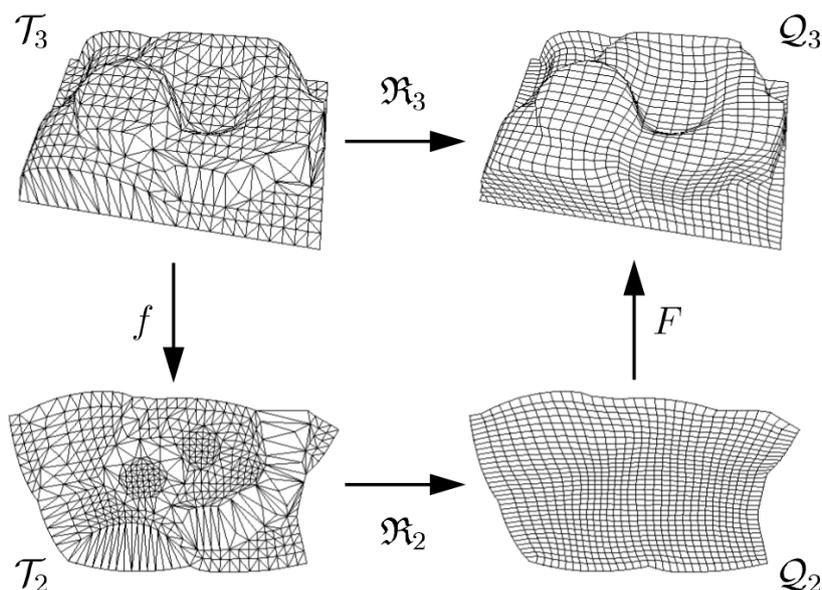


Figura 2.12: Parametrização de uma malha de triângulos, remalhamento no \mathbb{R}^2 e transposição para o \mathbb{R}^3 [20]

e ir refinando a mesma. Cada face de Q_G é dividida em 4 quadriláteros e a posição de seus vértices é tal que os novos vértices de borda são localizados na borda da parametrização espaçados igualmente em relação aos seus vizinhos de borda e os novos vértices internos são colocados no centro da aresta ou da face do seu corresponde em T_P . Posteriormente, é feito um processo de realocização dos vértices utilizando a suavização discreta de Laplace com pesos. Ao final da etapa de *remeshing* no espaço bidimensional, é calculada a distância de Hausdorff [21] que verifica o fecho convexo de um quadrilátero no \mathbb{R}^2 e mede sua distância a face correspondente no \mathbb{R}^3 – distância mínima para quatro triângulos que podem ser construídos com os cantos do quadrilátero. Para reconstruir a superfície, foram interpolados os vértices da malha de quadriláteros resultante do *remeshing* no \mathbb{R}^2 de forma a aproximar os valores originais de TP indiretamente – técnica chamada de interpolação do produto tensor B-Spline.

Marinov et al. [31] faz uma segmentação da malha inicial M em regiões R_i , levando em consideração as características principais do objeto mapeadas na malha (*features*), e efetua o *remeshing* em cada uma dessas regiões. O foco deste método é que a malha

Malhas de quadriláteros

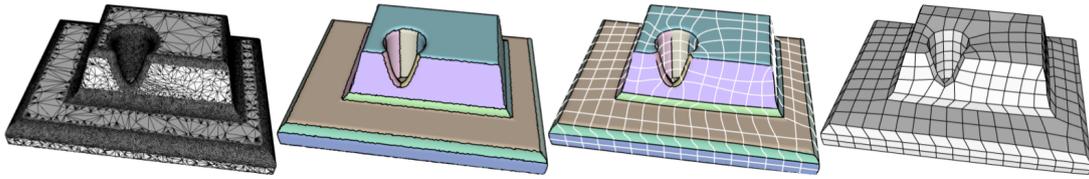


Figura 2.13: Malha de entrada, segmentação em regiões, computação das curvas de energia mínimas de cada região e malha resultante, respectivamente [31]

resultante possua um alinhamento de tal forma que as características do objeto não sejam prejudicadas pelas arestas dos elementos que a compõe, i.e. as arestas dos elementos não podem cruzar com as linhas características do objeto. Para realizar a segmentação, os autores utilizaram uma variação do método VSA (*Variational Shape Approximation*) [10] que inclui um *teleporte* de regiões para tentar amenizar o problema do mesmo travar num mínimo local na tentativa de minimizar uma função. Um resumo do método pode ser visto através da figura 2.13.

Na etapa de pré-processamento para efetuar o *remeshing* de cada região é verificado a topologia da mesma. Essa verificação é feita através de um subconjunto de vértices *âncora* que pertencem a pelo menos três faces da região e pelo cálculo do caminho único de arestas que dividem duas regiões e conectam dois vértices âncoras. Depois disso, dado esse conjunto de arestas com seus vértices, há uma aproximação da forma do objeto efetuada pelo método de mínimos quadrados através de curvas 3D cúbicas de Bézier pela minimização da métrica de integral L^2 . Para efetuar a parametrização de cada região e diminuir o problema para o caso bidimensional, foi utilizada a LSCM [28] que possui propriedades que diminuem a distorção do modelo na troca de espaço dimensional. Com as *features* suavizadas pelas curvas, finalmente ocorre a subdivisão das regiões em quadriláteros.

Daniels J. II et al. [22] aborda as dificuldades do *remeshing* através do uso de *templates*. A primeira etapa do processo se dá pela seleção das *features* na malha de entrada composta por triângulos, que pode ser feita de maneira automática ou manual. Com as features traçadas, foram utilizadas funções harmônicas para segmentar a malha em

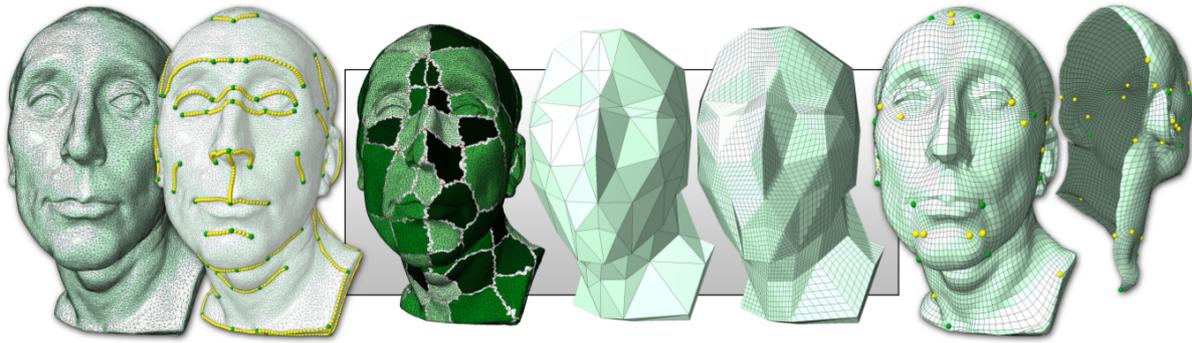


Figura 2.14: Malha inicial e features do objeto, decomposição em células para a construção de uma triangulação base, utilização dos templates na triangulação e otimização da localização dos vértices no mapeamento dos mesmos para a superfície original [22]

regiões de acordo com a localização dos pontos das features. Assim, surgem triângulos que podem ser classificados como dentro de uma região, entre duas regiões ou entre três regiões. Essa classificação gera uma representação *dual* de uma malha onde os triângulos que estão dentro de uma região representam uma face, os que estão entre duas regiões representam uma aresta e os que estão entre três regiões representam um vértice. A malha dual formada pode ter arestas que não correspondem as features estipuladas na primeira etapa. Para resolver isto, foi utilizada uma operação chamada de *vertex split* que consiste em transformar um vértice em dois e movê-los para o mesmo local dos *endpoints* das features. Como as arestas da malha dual não são arestas válidas na malha de entrada (não estão na superfície), são calculados arcos geodésicos para traçar um caminho pela superfície do objeto entre vértices que dividem uma aresta da malha dual.

Como o algoritmo pretende gerar malhas compostas somente por quadriláteros, as regiões trianguladas devem se transformar em quadriláteros. Isso é feito pela junção de dois triângulos que tem como restrição o não cruzamento das features. Essa junção é baseada numa representação de grafos da malha, onde cada triângulo é um vértice e a intersecção entre dois triângulos é uma aresta que liga dois vértices. Para todo vértice, são verificadas as arestas que ele participa e então apenas uma delas é marcada para a fusão. Com os quadriláteros formados, são utilizados templates padrão de triângulos e quadriláteros construídos com quadriláteros para segmentar as regiões na malha dual para

então projetar esses novos pontos internos formados na superfície do objeto. No final do processo, é feita uma suavização da malha de quadriláteros gerada utilizando a suavização Laplaciana. Um exemplo de utilização da técnica em uma malha é mostrado na figura 2.14.

A abordagem proposta por Bommes et al. [6] utiliza o conceito de *mixed-integers*. A partir de uma malha de triângulos, são modeladas equações lineares onde os elementos participantes da equação são em parte números inteiros e em parte números reais. Porém, uma malha compreende um conjunto discreto de elementos, os autores propuseram um *solver* guloso, iterativo, que transforma a parte contínua para um conjunto discreto minimizando o erro localmente de forma a transparecer à qualidade da malha de quadriláteros resultante no final do processo. Supondo uma função energia $E(x_1, x_2, \dots, x_n)$, para cada $x \in E$ sendo $x_0, \dots, x_k \in \zeta$ e $x_{k+1}, \dots, x_n \in \mathfrak{R}$, o algoritmo analisa cada x da parte contínua e arredonda para o número inteiro mais próximo. O erro de arredondamento gerado é então compensado localmente entre um subconjunto do conjunto contínuo. Após a análise desse subconjunto de elementos, a equação linear é então resolvida utilizando Gauss-Seidel.

Essa modelagem do problema em equações lineares de *mixed-integer* é utilizada em duas principais etapas do algoritmo que são: identificação e criação de campos saues direcionais de interpolação e parametrização da malha. Apesar dessa abordagem poder apresentar um grande número de arredondamentos e iterações, os autores comprovaram em seus resultados que o tempo computacional para estes cálculos é baixo se houver a utilização do *solver* proposto. A figura 2.15 ilustra algumas malhas resultantes do processo.

Na seção seguinte é apresentada algumas técnicas que descrevem um processo de simplificação de malhas. Estas etapas são combinadas com uma etapa de refinamento, para a geração de malhas homogêneas de quadriláteros.

2.4 Simplificação

O objetivo dos algoritmos de simplificação é a remoção de elementos de forma a manter ao máximo possível a forma do objeto. Essa operação é muito importante em geometria

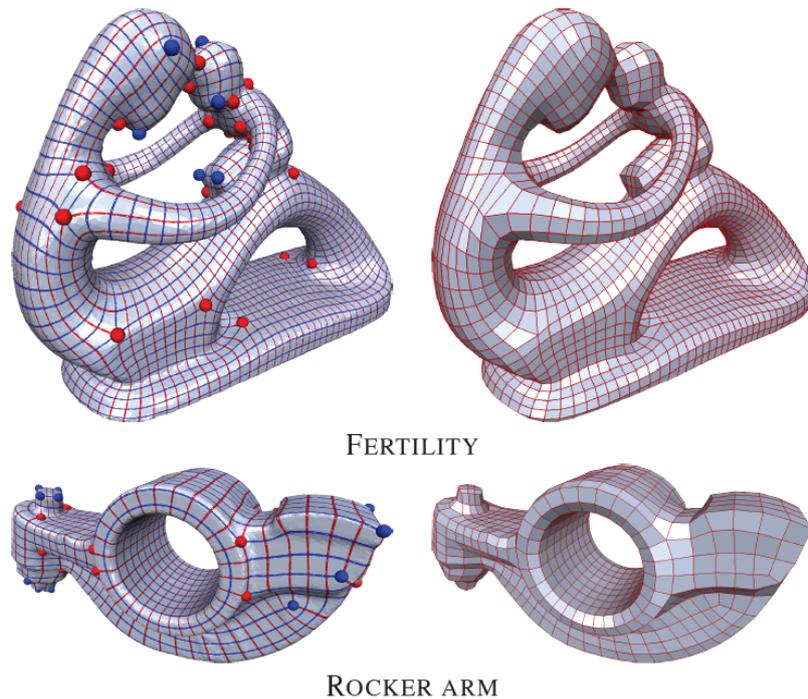


Figura 2.15: Resultados do algoritmo proposto por Bommer et al. À esquerda estão evidenciadas as singularidades da malha resultante. À direita está a malha resultante [6]

computacional pois auxilia algumas etapas de processamento de mais alto nível como a compressão de malhas, renderização, transmissão progressiva, operações de edição, suavização, parametrização e reconstrução de forma. Um desafio grande da simplificação em malhas de quadriláteros é que a estrutura dos elementos força o uso de operações globais, quando não queremos alterar a valência dos vértices. Por exemplo, para remover uma aresta de um triângulo em uma malha de triângulos, só é necessário estar atento aos elementos vizinhos ao mesmo mas a remoção de um único quadrilátero pode acarretar a remoção de um número maior de elementos.

Daniels J. II [11] introduz uma técnica, denominada QMS (*Quadrilateral Mesh Simplification*), baseada na exploração da estrutura dual da malha. Além dos operadores de simplificação, ele utiliza um esquema de priorização para maximizar a qualidade da estrutura da malha em representações menos refinadas e uma extensão da métrica do erro quadrático para a preservação da geometria. Um exemplo de utilização da QMS

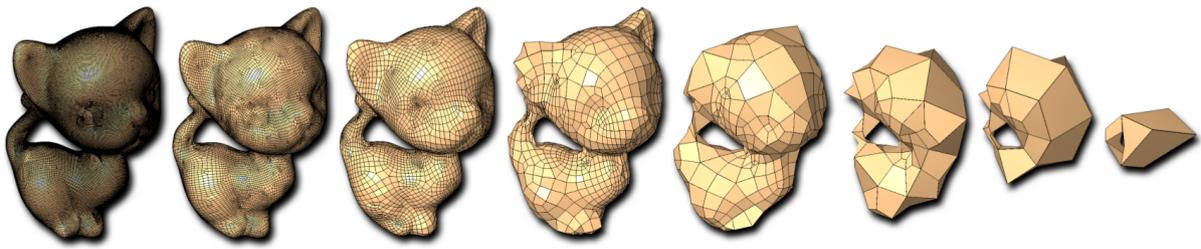


Figura 2.16: Exemplo de simplificação de uma malha com níveis de detalhe decrescentes e preservação da topologia do objeto [11]

em uma malha está apresentada na figura 2.16. A representação dual de uma malha de quadriláteros é definida pelos componentes: o dual de um elemento quadrilátero é sua centroide; o dual de uma aresta é uma corda que conecta os centroides de quadriláteros vizinhos; o dual de um vértice é o polígono formado pela conexão dos centroides dos quadriláteros vizinhos em ordem cíclica. Outro componente importante é a *poli-corda*, uma poli-linha cujos segmentos adjacentes são cordas que se encontram num centroide comum e são dual a arestas opostas no quadrilátero. Numa malha de quadriláteros fechada, as poli-cordas sempre formam *loops* fechados, ou seja, começando em uma aresta e caminhando pelas arestas opostas em quadriláteros adjacentes, ela sempre acabará na aresta inicial.

Há três operações de simplificação: colapso de poli-corda, de quadrilátero e de *doublet* (i.e. quadriláteros vizinhos que dividem duas arestas consecutivas). No colapso de poli-corda, há uma remoção de todos os quadriláteros que formam uma poli-corda pela fusão dos vértices de cada aresta dual da poli-corda selecionada. No colapso de quadrilátero, dois vértices da diagonal de um quadrilátero são fundidos para a remoção do mesmo. No colapso de *doublet* é feita a junção das duas faces divididas pelas arestas do vértice de grau 2. A priorização dessas operações é baseada no impacto que as mesmas terão sobre a malha, levando em consideração um esquema de pesos que verificam a qualidade final da valência do vértice, a perda geométrica e a área de distorção associada a cada operador de colapso. O algoritmo de simplificação QMS faz o balanceamento dessas operações, realizando colapsos de poli-corda até que não seja mais possível efetuar essa

Malhas de quadriláteros

operação para então realizar os colapsos de quadriláteros. Entre a execução das operações de colapso citados, se houver a criação de *doublets*, os mesmos são removidos através da operação de colapso de *doublets*.

Na etapa de pós processamento, é feita uma suavização da malha simplificada através de um procedimento descrito por Zhang et al. [49] que movimenta os vértices no sentido da normal para redução de ruído e na direção da tangente para aumento da relação de aspecto.

Tarini et al. [46] utiliza uma abordagem incremental baseada somente em operações locais. Essa técnica visa simplificar uma malha de forma que a mesma se torne a mais próxima de homeométrica possível – todas as arestas possuindo um tamanho exato l , e todas as diagonais das faces tem exatamente tamanho $l\sqrt{2}$. Ter uma malha homeométrica também significa que todos os ângulos são bons, todas as faces são planas e a distribuição dos vértices é uniforme. O algoritmo conceitual possui três bases: um conjunto de operações locais, uma heurística para selecionar a operação que tende a maximizar a homeometria e uma operação de suavização tangente que reorganiza os vértices na superfície da malha. Para efeito de organização, os passos necessários para realizar esta simplificação seriam (1) suavizar toda a malha inicial M_0 , (2) processar iterativamente a malha M_i para produzir uma malha M_{i+1} (mais grossa) e (3) suavizar globalmente a malha M_n . No segundo passo, o processo iterativo é interrompido quando o critério do usuário (número de quadriláteros) é alcançado. Em cada iteração são executadas operações locais de engrossamento da malha, até que nenhuma mais seja possível, e por fim, a malha é suavizada localmente. Na figura 2.17 há dois exemplos de malhas simplificadas utilizando esta técnica.

O autor utiliza três tipos de operações locais: operações de engrossamento, otimização e limpeza. As operações de engrossamento visam reduzir a complexidade removendo elementos da malha (remoção de um quadrilátero através da junção de dois vértices ou através de uma rotação das arestas do vértice para formar um quadrilátero diagonal possível de ser removido). As operações de otimização visam melhorar a qualidade dos

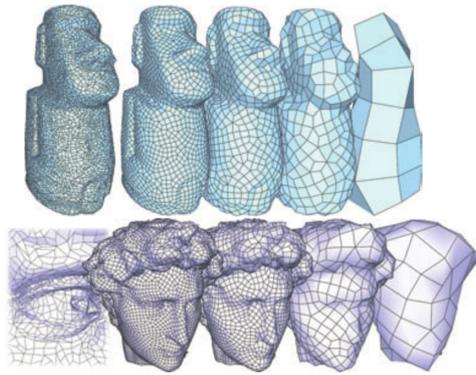


Figura 2.17: Exemplo de malhas resultantes da simplificação [46]

quadriláteros formados por rotações de vértice (rotação das arestas que pertencem ao vértice – estrela do vértice) ou por rotações de aresta (substituição de uma aresta que seja compartilhada por dois quadriláteros e por uma outra no sentido horário ou anti-horário). As operações de limpeza visam remediar quadriláteros degenerados por remoção de *doublets* e *singlets*. Um exemplo dessas operações está na figura 2.18. A escolha dessas operações são baseadas numa heurística de tamanho. Na priorização de operações de colapso, é escolhido o elemento mais curto da malha. Na priorização de operações de otimização, a rotação de vértice é escolhida se a soma do tamanho das arestas se sobrepõe a soma das diagonais e a rotação de aresta é escolhida se houver uma melhora nos ângulos internos dos quadriláteros formados pela nova aresta. Na priorização de operações de limpeza, os *singlets* são removidos assim que detectados e os *doublets* passam por uma verificação de curvatura Gaussiana; o *doublet* só é mantido se essa curvatura for extremamente positiva.

Por fim, é efetuada uma suavização de espaço tangente que consiste em mover os vértices para maximizar a homeometria. Além de aumentar a qualidade da malha, ela ajuda a selecionar a próxima operação a ser executada. Os elementos que não se tornam homeométricos após uma etapa de suavização são ótimos candidatos para a próxima operação de remoção/rotação de elementos. Vértices afetados pelas operações locais são marcados para a suavização. No caso de qualquer vértice ser movido durante a suavização em um valor maior que um dado limite, então seus vizinhos são também marcados para

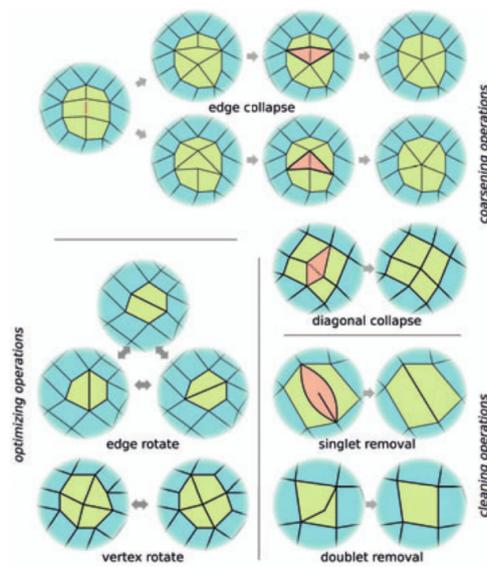


Figura 2.18: Exemplo dos tipos de operações locais [46]

suavização.

No próximo capítulo, são apresentadas definições de curva e superfície de bézier, bem como suas propriedades. Além disso há uma seção dedicada a parametrização e uma dedicada a geodésicas.

Aproximação de superfícies

Para representar computacionalmente objetos contínuos como curvas e superfícies, muitas vezes é necessário discretizar os mesmos. Assim como uma imagem que está no plano \mathbb{R}^2 é representada por uma matriz de *pixels*, uma superfície no \mathbb{R}^3 pode ser representada por projeções discretas do espaço no plano. Essa discretização pode não representar o objeto com a fidelidade desejada, portanto é necessário que haja uma aproximação da superfície do objeto de modo a melhor descrever a sua forma. Uma maneira de realizar essa aproximação é através do uso de superfícies de Bézier.

3.1 Superfícies de Bézier

As curvas de Bézier, criadas por Pierre Bézier inicialmente para descrever curvas suaves no design de carros, são curvas paramétricas muito utilizadas em aplicações de computação gráfica na descrição de vetores, em fontes *TrueType*, na construção do caminho para interpolação de objetos em animação, entre outras. As curvas e superfícies de Bézier são definidas como uma curva de aproximação, pois os únicos pontos que necessariamente estão presentes na curva ou superfície são os seus extremos chamados de *endpoints* ou

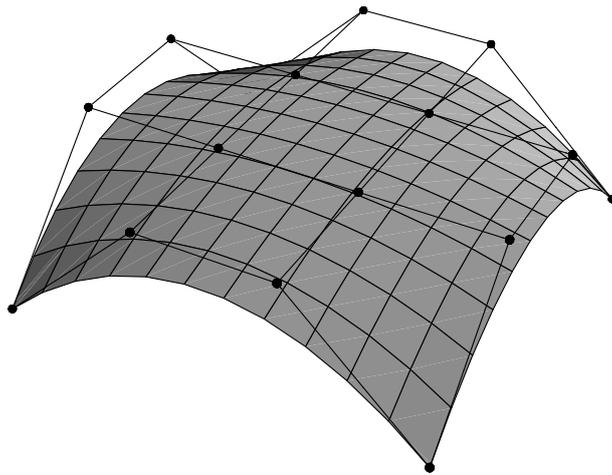


Figura 3.1: Patch retangular de Bézier [41]

pontos fim. No caso de uma curva de Bézier cúbica, são utilizados quatro pontos de controle sendo que a curva só passa sobre dois deles, os *endpoints*. Para representar superfícies, é utilizado o produto cartesiano de duas curvas de Bézier. A figura 3.1 ilustra uma superfície retangular de Bézier bi-cúbica.

3.1.1 Definição

Os pontos que descrevem uma curva de Bézier, inclusive os *endpoints*, são chamados de pontos de controle. O fecho convexo destes pontos é chamado de polígono de controle.

A curva de Bézier é definida matematicamente como segue:

$$P(t) = \sum_{i=0}^n p_i B_{n,i}(t), 0 \leq t \leq 1$$

sendo que p_i representa cada um dos $n + 1$ pontos de controle e sendo $B_{n,i}(t)$ os polinômios de Bernstein (Figura 3.2) que representam a influência de cada ponto sob a curva. Os polinômios de Bernstein são descritos como segue:

$$B_{n,i}(t) = \binom{n}{i} t^i (1-t)^{n-i}$$

sendo $\binom{n}{i} = \frac{n!}{i!(n-i)!}$ se $0 \leq i \leq n$ ou 0 caso contrário.

Um exemplo seria a curva de Bézier cúbica, onde temos quatro pontos de controle $p_0 = (x_0, y_0)$, $p_1 = (x_1, y_1)$, $p_2 = (x_2, y_2)$, $p_3 = (x_3, y_3)$ onde p_0 é o primeiro ponto e,

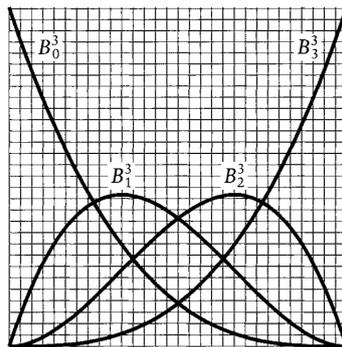


Figura 3.2: Polinômios de Bernstein no caso cúbico [17]

portanto, ponto de início, p_1 é o segundo ponto, p_2 é o terceiro ponto e p_3 é o quarto ponto e, portanto, o ponto fim. A ordem dos pontos é importante pois, além de manter as propriedades de Bézier (seção 3.1.2), ela indica quais são os pontos em que a curva passa, no caso, p_0 e p_3 .

Observando o caso cúbico e utilizando a definição supracitada, a fórmula que descreve uma curva de Bézier cúbica segue:

$$P(t) = (1 - t)^3 \cdot p_0 + 3 \cdot (1 - t)^2 \cdot t \cdot p_1 + 3 \cdot (1 - t) \cdot t^2 \cdot p_2 + t^3 \cdot p_3$$

com $0 \leq t \leq 1$. Quando o valor de t é 0 é possível ver que o resultado da equação é p_0 e quando o valor de t é 1 o resultado da equação é p_3 . Nota-se também que não há valores de t que resultem nas coordenadas dos pontos p_1 e p_2 . Assim, a curva de Bézier é dita ser uma curva de aproximação pois ela passa necessariamente apenas em seus *endpoints*. É possível plotar um gráfico da função $P(t)$ para verificarmos o seu comportamento, bem como o peso que cada ponto de controle exerce sobre a curva, ilustrado na figura 3.2.

Com as curvas de Bézier é possível construir uma superfície de Bézier que pode ser definida como o cartesiano de duas curvas de Bézier. A fórmula geral que descreve essa superfície é dada por:

$$P(r, s) = \sum_{j=0}^n \sum_{k=0}^m p_{j,k} \cdot B_{n,j}(r) \cdot B_{m,k}(s)$$

sendo r e s as coordenadas paramétricas, $p_{j,k}$ os pontos de controle e $B_{n,j}(r)$ e $B_{m,k}(s)$ como sendo polinômios de Bernstein.

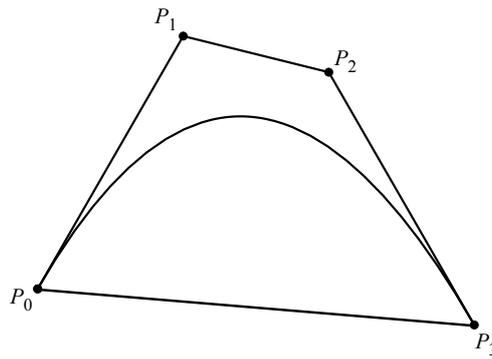


Figura 3.3: Exemplo de fecho convexo de curvas de Bézier

3.1.2 Propriedades

Uma propriedade interessante para a construção de caminhos de Bézier (i.e. ligação de curvas de Bézier) é que as retas tangentes aos *endpoints* são definidas pelos próprios *endpoints* e seus pontos adjacentes. Isso implica que, para fazer uma ligação suave de duas curvas de Bézier, basta que a ligação entre as duas se torne um único ponto e que a reta tangente ao ponto anterior e a ligação seja igual a reta entre o ponto posterior e a ligação.

A reversão da sequência dos pontos de controle não altera a forma e nem o tamanho da curvatura, só a direção da parametrização é alterada. Caso haja uma mudança arbitrária na sequência dos pontos de controle, a curva poderá sofrer transformações. As curvas de Bézier também são invariantes sob uma transformação afim. Na modelagem geométrica há um grande uso de transformações desse gênero sendo duas delas a translação e mudança de escala que, ao ser executada sob a curva, a mesma poderia ser alterada – o que não ocorre com as curvas de Bézier.

Além dessas propriedades existe a de que a curva sempre está dentro do fecho convexo construído pelos seus pontos de controle (Figura 3.3), mesmo que as arestas do polígono se cruzem. Essa propriedade é importante pois com ela é possível detectar a intersecção de duas curvas de Bézier matematicamente.

Na subsecção anterior, a curva de Bézier foi definida matematicamente. Com essa definição, surge a propriedade de partição unitária que diz que independente do valor de

Aproximação de superfícies

u , variando entre 0 e 1, a soma dos pesos será sempre 1 independente do grau da curva. Para exemplificar, se considerarmos novamente a curva cúbica, teremos a equação:

$$(1 - u)^3 + 3(1 - u)^2u + 3(1 - u)u^2 + u^3 = 1$$

Para $u = 0.6$ os valores a equação seria $0.064 + 3 \cdot 0.16 \cdot 0.6 + 3 \cdot 0.4 \cdot 0.36 + 0.216 = 1$. Com isso, há a ideia intuitiva de que os pontos de controle na verdade representam pesos ou o quanto que cada ponto participa na construção da curva assim como ilustrado na figura 3.2.

3.1.3 Bézier Inverso

Usualmente as curvas e superfícies de Bézier são construídas utilizando os pontos de controle fornecidos. Porém, em diversas aplicações, queremos obter uma curva ou uma superfície que represente um objeto já definido (ou aproxime o melhor possível). Nestes casos, não conhecemos os pontos de controle e assim não conseguimos gerar diretamente a curva ou superfície de Bézier correspondente.

Como neste projeto utilizaremos apenas superfícies representadas por pontos extraídos de uma malha de triângulos, a aproximação destas superfícies será orientada na diferença entre a malha de triângulos e a superfície de Bézier calculada. Para calcular os pontos de controle $p_{j,k}$, partindo da equação de Bézier citada na subseção 3.1.1, precisamos conhecer as coordenadas paramétricas (r, s) de cada ponto da amostragem, pois já conhecemos os binômios de Bernstein e a sua localização na superfície (coordenada no espaço tridimensional).

A obtenção das coordenadas paramétricas (r, s) é detalhada na seção seguinte. Com estas informações, conseguimos calcular os pontos de controle pelo método dos mínimos quadrados, conforme é descrito em [16].

Um exemplo de construção do sistema a ser resolvido é exibido a seguir, representando a equação de Bézier em forma de multiplicação de matrizes:

$$\begin{pmatrix} B_{n,0}(r_0)B_{n,0}(s_0) & B_{n,0}(r_0)B_{n,1}(s_0) & \cdots & B_{n,3}(r_0)B_{n,3}(s_0) \\ B_{n,0}(r_1)B_{n,0}(s_1) & B_{n,0}(r_1)B_{n,1}(s_1) & \cdots & B_{n,3}(r_1)B_{n,3}(s_1) \\ \vdots & \vdots & \ddots & \vdots \\ B_{n,0}(r_{99})B_{n,0}(s_{99}) & B_{n,0}(r_{99})B_{n,1}(s_{99}) & \cdots & B_{n,3}(r_{99})B_{n,3}(s_{99}) \end{pmatrix} \cdot \begin{pmatrix} p_{0,0} \\ p_{0,1} \\ \vdots \\ p_{3,3} \end{pmatrix} = \begin{pmatrix} P_0 \\ P_1 \\ \vdots \\ P_{99} \end{pmatrix}$$

sendo, neste exemplo, para uma superfície de Bézier bicúbica e 100 pontos de amostragem.

3.2 Parametrização

Tendo uma superfície de entrada, uma parametrização pode ser definida como o mapeamento dessa superfície em um domínio conveniente de forma que haja a correspondência de pontos tanto na superfície quanto no novo domínio. Além das aplicações intrínsecas dessa definição como o remalhamento de superfícies através da reamostragem do domínio e reparo de modelos CAD, a força motriz do desenvolvimento de técnicas de parametrização foi dada pela possibilidade de mapear texturas (normalmente como parametrização de $\mathbb{R}^2 \rightarrow \mathbb{R}^3$).

Sheffer et al. [43] separam as técnicas de parametrização planar em quatro grupos, baseados na preocupação da distorção causada pela mesma: métodos que ignoram completamente a distorção, métodos que minimizam a distorção angular, métodos que minimizam o esticamento, e métodos que minimizam a distorção em área.

No primeiro grupo são descritos duas formulações. A formulação de grafo incorporado utilizada por Tutte [47] mapeia os vértices de borda da malha para a borda da região convexa em 2D e as posições do resto dos vértices são obtidas pela resolução de um sistema linear que resulta em uma parametrização nem sempre bijetiva e não preserva nenhuma propriedade de forma da malha. A outra formulação é referente ao teorema do empacotamento de círculos [38] onde cada vértice possui uma coleção de círculos tal que dois círculos são tangentes se há uma aresta da malha entre os vértices associados. Assim, pela conexão dos centros dos círculos se obtém uma representação planar da malha.

Embora este último método garanta que o resultado seja bijetivo, ele não preserva as propriedades da forma do objeto.

As técnicas propostas pelo segundo grupo são baseadas na manipulação dos pesos presentes nas equações lineares das técnicas do primeiro grupo. A escolha desses pesos tem influência direta na distorção e na capacidade da parametrização de ser bijetiva. Alguns autores [15] [36] utilizaram pesos harmônicos ou cotangentes que são derivados de uma descrição de elementos finitos de energia harmônica. Um dos problemas principais da parametrização harmônica é que se a malha tiver ângulos obtusos os pesos podem ser negativos e portanto o resultado da parametrização pode não ser bijetivo. Levando em consideração que, tendo uma malha que satisfaz o critério de Delaunay a parametrização obtida com pesos cotangentes sempre será bijetiva, Kharevych et al [24] sugeriu que a entrada do mapeamento harmônico fosse uma triangulação de Delaunay “intrínseca” da superfície para garantir a bijetividade. Outros autores também utilizam mapeamento harmônico discreto [18] [19].

No terceiro grupo os métodos buscam minimizar a distorção linear ao invés de eliminá-la completamente. Formulações do passado [4] [30] [27] eram numericamente complexas e difíceis de minimizar, mas Sander et al. [39] introduziu duas métricas de esticamento, L_2 e L_∞ , que foram comumente utilizadas para comparar distorções lineares entre parametrizações. Os métodos que utilizam essas métricas [50] [42] [24] começam com uma parametrização que preserva a forma [18] e então move um vértice de cada vez para minimizar a métrica L_2 . Outros autores [52] [50] visam preservar as distâncias entre todos os pares de vértices na superfície computando primeiro a distância geodésica e então utilizando *multi-dimensional scaling* (MDS) para trazer a malha para o plano.

As técnicas do quarto grupo se preocupam com a preservação de área. Essas técnicas normalmente introduzem termos adicionais de otimização ou restrições na parametrização. Desbrun et al. [13] obtém uma formulação linear para preservação de área local através do uso de uma derivação similar de seu DCP (*Discrete Conformal Parameterization*). Com uma abordagem não linear, Degener et al. [12] desenvolveu uma formulação de

preservação de área por triângulo.

3.3 Geodésicas

Apesar de não existir uma definição única, do ponto de vista prático, uma geodésica pode ser definida como a menor distância entre dois pontos em uma superfície. Por exemplo, no plano as geodésicas são linhas retas, mas em uma esfera as geodésicas são grandes círculos. O termo “geodésica” vem da ciência geodésia que estuda a forma e as dimensões da Terra buscando determinar também o seu campo externo de gravidade. O primeiro uso deste termo se deu depois de F. W. Bessel estar envolvido com a determinação da forma da Terra como uma elipsoide de rotação e C. G. Jacobi referenciar como “curvas geodésicas” as menores curvas nessa elipsoide de rotação, por volta do ano 1820.

Na tentativa de solucionar esse problema de computar a distância entre dois pontos pela superfície, Martínez et al. [32] propõem um método que consiste em duas etapas principais: primeiramente é computada uma curva Γ_0 ligando os pontos A e B e iterativamente melhorar essa curva (gerando Γ_i com $i = 0, 1, \dots, n$) até chegar em uma boa aproximação da geodésica Γ .

Para obter Γ_i foi utilizado o método FMM (*Fast Marching Method*) – um algoritmo numérico para resolução de equações diferenciais parciais não lineares (*Eikonal equations*). Uma vez obtida a curva Γ_i é necessário efetuar correções de forma a gerar uma curva Γ_{i+1} que seja mais próxima da geodésica Γ . Como Γ tem que coincidir com um segmento de linha dentro de uma face da malha, os vértices das aproximações sucessivas foram restringidos a estarem em uma aresta ou em um vértice da malha. Os autores fazem então as correções para esses dois casos e com a nova Γ_i o processo continua até que esta esteja próxima da geodésica Γ .

O critério de parada utilizado foi a definição de uma métrica de erro baseada na posição e no ângulo dos vértices da curva. Para os vértices que estão nas arestas da malha e vértices que coincidem com os vértices euclidianos da malha, o erro adotado foi a diferença entre os ângulos esquerdo e direito θ_l θ_r . Para vértices que coincidem

Aproximação de superfícies

com vértices esféricos da malha, o erro adotado foi um valor muito grande pois nenhuma geodésica passa por eles. Para vértices que coincidem com vértices hiperbólicos da malha, o erro adotado foi zero se θ_l e θ_r são maiores que π ou, caso contrário, um valor muito grande pois a geodésica só passa no primeiro caso.

No capítulo seguinte é apresentado projeto desenvolvido no mestrado.

Refinamento de superfícies de quadriláteros

Este projeto consiste no desenvolvimento de um novo método para obter malhas superficiais de quadriláteros, a partir de malhas de triângulos. O *pipeline* foi desenvolvido de modo a não exigir triângulos de boa qualidade na malha de entrada, pois as técnicas escolhidas para processar estes elementos são robustas o suficiente para dispensar tal exigência. O método desenvolvido viabiliza três tipos de controle na malha resultante: controle global e local da densidade dos elementos e controle de qualidade dos elementos gerados.

O controle global da densidade dos elementos permite a definição do número e tamanho dos quadriláteros de todo o objeto, viabilizando modelos em diferentes resoluções. Este controle é possível pela exigência mínima de subdivisões no processo iterativo, desconsiderando o fato da região estar bem aproximada. O controle local da densidade dos quadriláteros permite gerar modelos adaptativos, permitindo o uso de diferentes resoluções de acordo com a necessidade de detalhamento em cada região. Este controle é viabilizado nas subdivisões guiadas pelo critério de erro adotado em cada iteração do método.

O controle de qualidade dos elementos é conquistado pelo critério que estabelecido

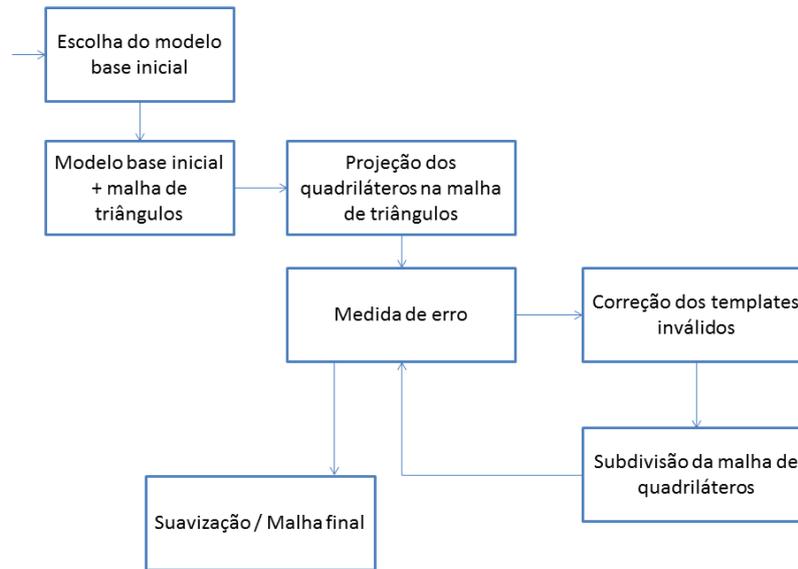


Figura 4.1: Visão geral do pipeline implementado.

no momento da subdivisão de uma superfície. Como cada subdivisão representa novos quadriláteros na malha base, é possível ter controle dos novos quadriláteros gerados. A qualidade destes elementos está diretamente ligada a escolha do ponto no momento da subdivisão.

4.1 Introdução

O algoritmo desenvolvido utiliza uma abordagem iterativa, onde a partir de uma malha base inicial, operadores de subdivisão de quadriláteros são aplicados repetidamente.

O algoritmo é composto por um *pipeline* de três etapas principais: obtenção da malha base de quadriláteros, refinamento da malha para alcançar os objetivos supracitados e por fim, suavização da malha resultante. As etapas deste algoritmo estão ilustradas na imagem 4.1

Na seção seguinte serão explicadas as técnicas utilizadas na construção da malha base inicial bem como as particularidades da escolha dos vértices que a compõe.

4.2 Malha base

Entende-se como malha base, o modelo simplificado que descreve topologicamente o objeto. Neste projeto, a malha base é composta somente por quadriláteros, podendo ser obtida manualmente ou por meio de técnicas automáticas de simplificação de malhas. Vale ressaltar que a malha base não representa com precisão o objeto e serve apenas de guia inicial no processo iterativo. Cada face da malha base de quadriláteros possui uma referência para um conjunto de faces triangulares da malha de triângulos, ou seja, cada quadrilátero é mapeado a uma região da superfície do objeto.

Para fazer este mapeamento, utilizamos como vértices da malha inicial de quadrilátero um subconjunto dos vértices da malha de triângulos, estabelecendo assim uma relação direta entre os vértices das duas malhas. Para cada aresta da malha inicial de quadriláteros, calculamos a geodésica que liga os vértices extremos da aresta, formando assim um caminho sobre a superfície. As regiões formadas entre estes caminhos, são as regiões correspondentes a cada quadrilátero da malha inicial. Na figura 4.2, é ilustrada uma malha inicial de quadriláteros e na figura 4.3 são ilustradas as regiões correspondentes na superfície do objeto.

Como esse caminho traçado pela geodésica busca-se a menor distância entre dois vértices na superfície, este caminho não é restrito às arestas e vértices que já existiam originalmente na malha de triângulos. Levando em consideração que não há garantias que as arestas da malha de triângulos são equivalentes às menores distâncias entre os vértices na superfície, é natural que a geodésica traçada passe por elementos onde a intersecção deste encontro são duas arestas, ou uma aresta e um vértice por exemplo. Para manter esta restrição, a malha é subdividida nos triângulos intersectados pelo caminho da geodésica.

O uso de geodésicas para o cálculo da menor distância entre dois pontos em uma superfície nem sempre mapeia corretamente cada elemento da malha base em uma região correspondente da malha de triângulos. Isso pode ocorrer através da intersecção entre duas geodésicas. Esta possibilidade está presente apenas nas primeiras geodésicas para

Refinamento de superfícies de quadriláteros

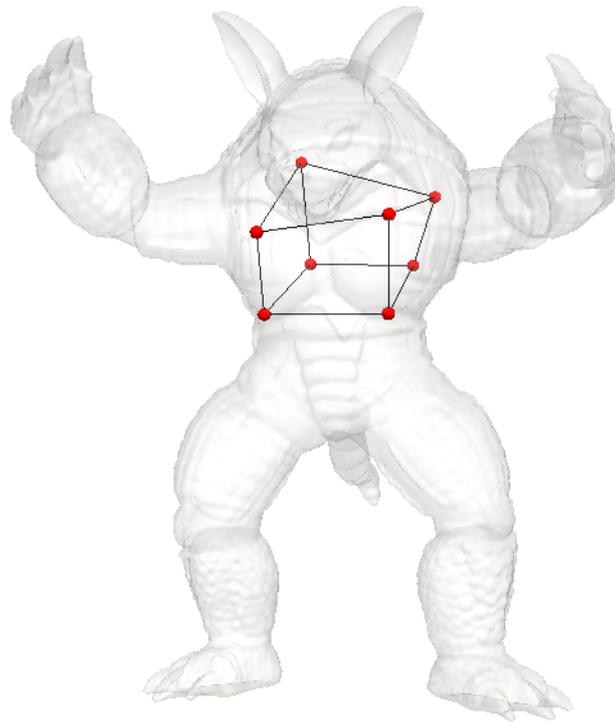


Figura 4.2: Malha base de quadriláteros destacada da superfície de triângulos a ser aproximada.

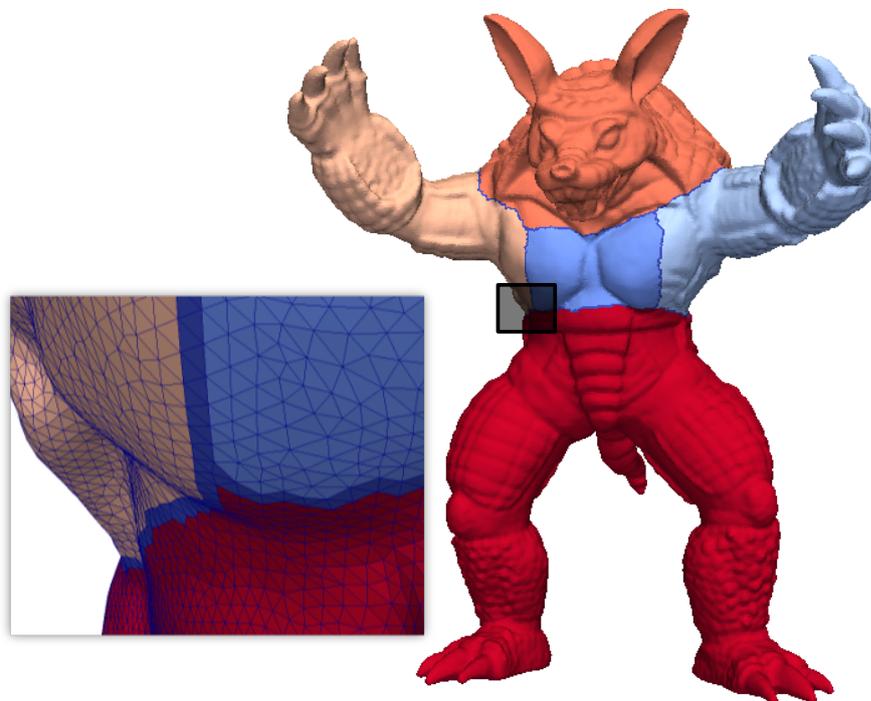


Figura 4.3: Regiões da malha de triângulos correspondentes a cada face da malha base.

o mapeamento inicial da malha base, podendo comprometer o término do mapeamento inicial. Porém, nas demais etapas do processo (refinamento), este problema foi solucionado através da restrição do espaço de busca para o cálculo da geodésica.

A subdivisão da malha de triângulos em regiões correspondentes a malha base inicial, utilizando esta abordagem de geodésicas, ainda é um problema em aberto.

Em seguida, é executado um algoritmo de *flood* (inundação). A partir de uma face aleatória contida em uma determinada região, as faces vizinhas são marcadas como pertencentes a esta mesma região. Este processo de crescimento de região é interrompido quando a aresta em comum com a face vizinha pertencer a alguma geodésica. Após todas as regiões serem marcadas o mapeamento de cada região com um quadrilátero da malha base pode ser determinado pela vizinhança.

Como a malha base inicial é um modelo simplificado, ela não aproxima bem o objeto alvo. Neste ponto, o foco do algoritmo está no refinamento desta malha base até que esta aproxime a superfície do objeto. Este refinamento é realizado quadrilátero por quadrilátero, subdividindo-os por padrões pré-estabelecidos (*templates*). Este processo de subdivisão pode ser executado até que a aproximação do objeto pela malha de quadriláteros seja satisfatória, ou então, até que um grau de refinamento especificado pelo usuário seja alcançado. Na seção seguinte será explicado com mais detalhes este algoritmo de refinamento iterativo.

4.3 Refinamento

Com a primeira etapa concluída, temos uma malha inicial de quadriláteros formada por um subconjunto dos vértices da malha de triângulos, e ainda uma segmentação da malha de triângulos em regiões, cada uma correspondente a um quadrilátero.

O refinamento da malha de quadriláteros só se faz necessário quando a aproximação da superfície do objeto é muito ruim (segundo um critério de erro de aproximação). Esta definição de má aproximação, é determinada em cada quadrilátero individualmente, dividindo os quadriláteros da malha base em dois subconjuntos: quadriláteros que precisam

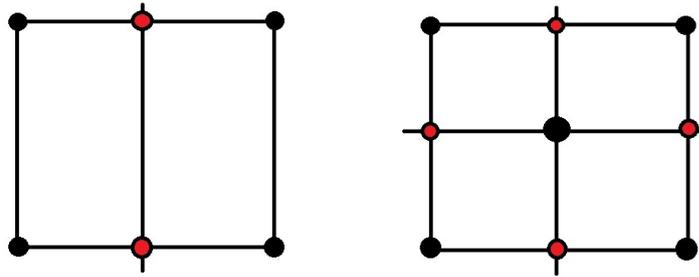


Figura 4.4: Modelos de subdivisão que não inserem vértices ruins na malha.

e os quadriláteros que não precisam ser refinados, ou seja, que aproximam bem ou não suas regiões correspondentes.

Os quadriláteros que precisam ser subdivididos são refinados, gerando assim uma nova malha base. A partir desta nova malha base, o processo de refinamento é repetido até que o subconjunto de quadriláteros que aproximam mal sua região correspondente seja vazio.

No exemplo da figura 4.2, todos os seis quadriláteros que formam a malha base inicial são marcados para serem subdivididos, uma vez que eles não aproximam a malha alvo de maneira satisfatória.

Durante esta repetição do algoritmo de refinamento, como cada quadrilátero representa uma região da malha de triângulos, após algumas iterações possivelmente existirão quadriláteros que já aproximam bem a superfície e não necessitam mais de subdivisões, enquanto que outros ainda deverão ser marcadas para subdivisão.

Existem inúmeros modos de subdividir um quadrilátero. As subdivisões que não introduzem vértices de grau diferente de 4 na malha são as mais desejadas por tentar manter a malha o mais próxima de ser regular. Na figura 4.4 são exibidos dois modelos de subdivisão que não inserem vértices ruins na malha. Vale ressaltar, que os vértices marcados como vermelho terão grau 4 assim que os quadriláteros vizinhos forem subdivididos.

Estes templates (exibidos na figura 4.4) criam uma dependência com os quadriláteros vizinhos, pois nas arestas que foram subdivididas é necessário que o quadrilátero vizinho também a subdivida. Esta dependência pode se propagar em toda malha, sendo este um

Refinamento de superfícies de quadriláteros

problema comum em todas as técnicas de refinamento de quadriláteros. Para solucionar este problema, é necessário inserir vértices com grau diferente de 4.

A figura 4.5 ilustra todas as situações possíveis de um quadrilátero que tem pelo menos um de seus vizinhos subdividindo a aresta em comum. O template exibido à esquerda da figura 4.4, é apropriado para os casos (b) e (c) da figura 4.5. O template exibido à direita na figura 4.4, é apropriado apenas para o caso (d) da figura 4.5. Para os casos (e), (f), (g) e (h) da figura 4.5 não podemos utilizar os templates da figura 4.4, exigindo a adição de um vértice com grau diferente de 4. Uma solução imediata é utilizar um dos templates exibidos na figura 4.6. Os demais casos, não é possível subdividir o quadrilátero de forma a respeitar estritamente as arestas já subdivididas. Desta maneira, desenvolvemos um algoritmo para eliminar estes casos, os quais chamamos de inválidos. Este algoritmo será detalhado na seção 4.3.2.

No algoritmo de refinamento, o conjunto de quadriláteros marcados para serem subdivididos são refinados pelo template representado à direita na figura 4.4. Este template foi escolhido por não adicionar vértice extraordinário e subdividir o quadrilátero em um número maior de elementos, maximizando as chances de que na próxima iteração, os elementos novos não sejam subdivididos.

Alguns quadriláteros do conjunto que já aproximam bem a superfície, podem ser subdivididos nos casos em que possuem uma das arestas subdivididas pelo vizinho correspondente, associando assim um template correspondente. Os casos classificados como inválidos, estes quadriláteros são fornecidos como entrada para o algoritmo de eliminação de casos inválidos.

Neste caso, são necessárias marcações que identifiquem cada caso de subdivisão em relação aos vizinhos. A subseção seguinte explica como as marcações de subdivisão foram feitas para todas as relações possíveis entre um quadrilátero e seus vizinhos.

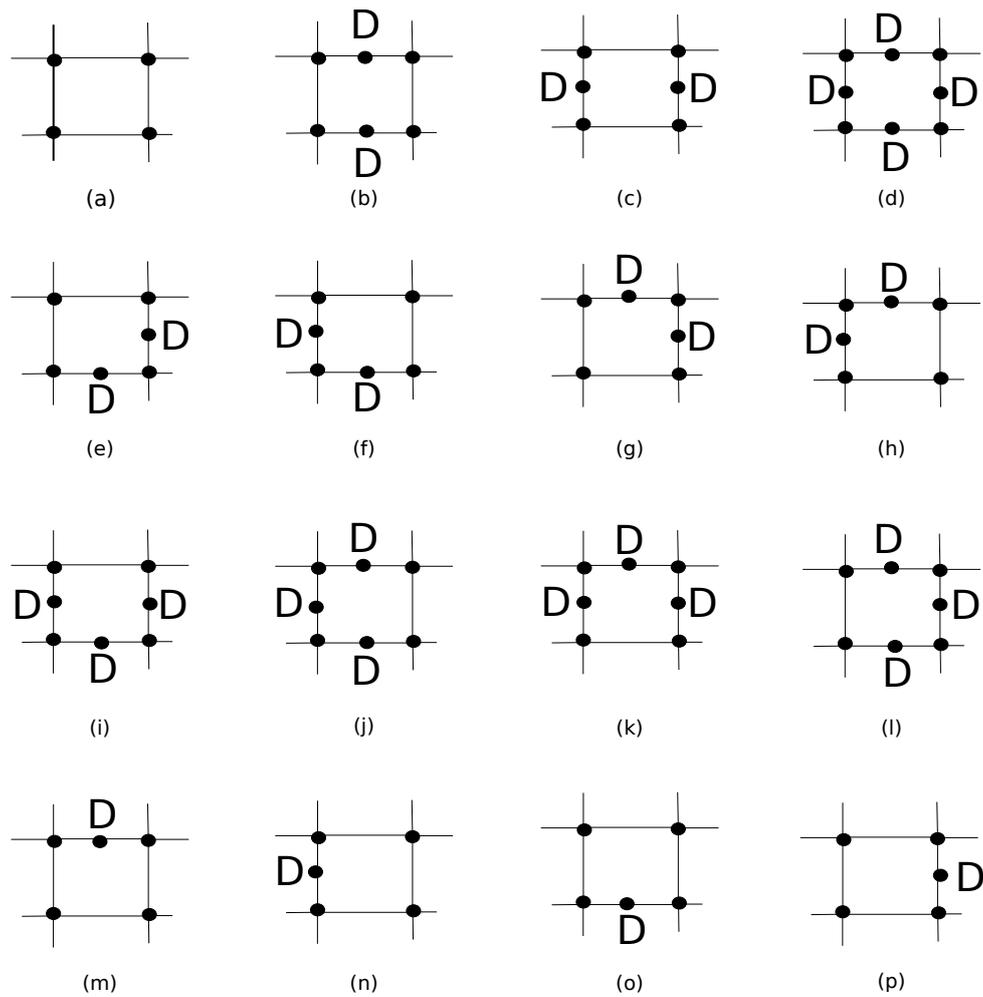


Figura 4.5: Todas as situações possíveis de quadriláteros com arestas a serem subdivididas. A letra 'D' representa um vizinho que foi subdividido e que compartilha uma aresta que precisa ser subdividida com o quadrilátero central.

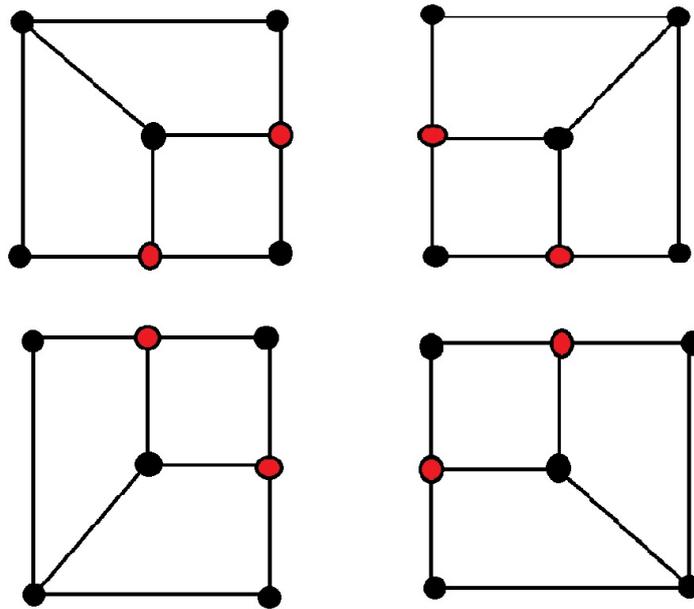


Figura 4.6: Modelos de subdivisão onde há a inserção de um vértice extraordinário de grau 3 no centro do quadrilátero.

4.3.1 Marcação dos quadriláteros a serem refinados

O tipo de marcação que cada quadrilátero recebe tem uma conexão direta com a observação dos quadriláteros vizinhos. Para cada possível caso, existe a necessidade de uma determinada subdivisão. Para cada tipo de subdivisão possível foi atribuído um template específico. Levando-se em consideração o quadrilátero central, a imagem 4.7 ilustra as possibilidades de templates, já considerando os casos simétricos. A marcação N significa que o quadrilátero vizinho não subdivide a aresta em comum e a marcação D significa que a aresta em comum ao quadrilátero vizinho está marcada para subdivisão.

Um quadrilátero marcado com template do tipo NNNN simboliza que nenhum de seus vizinhos marcam as arestas em comum para subdivisão. Seguindo o mesmo raciocínio, templates do tipo DNNN, NDNN, NNDN e NNND representam que apenas 1 vizinho está marcando uma aresta para subdivisão. Consideramos todos estes quatro casos como sendo do mesmo tipo (simétricos), o DNNN.

No caso dos templates simétricos de tipo DNDN ou NDND, representam que existem dois vizinhos opostos que marcam arestas para subdivisão (vizinho de cima e o vizinho

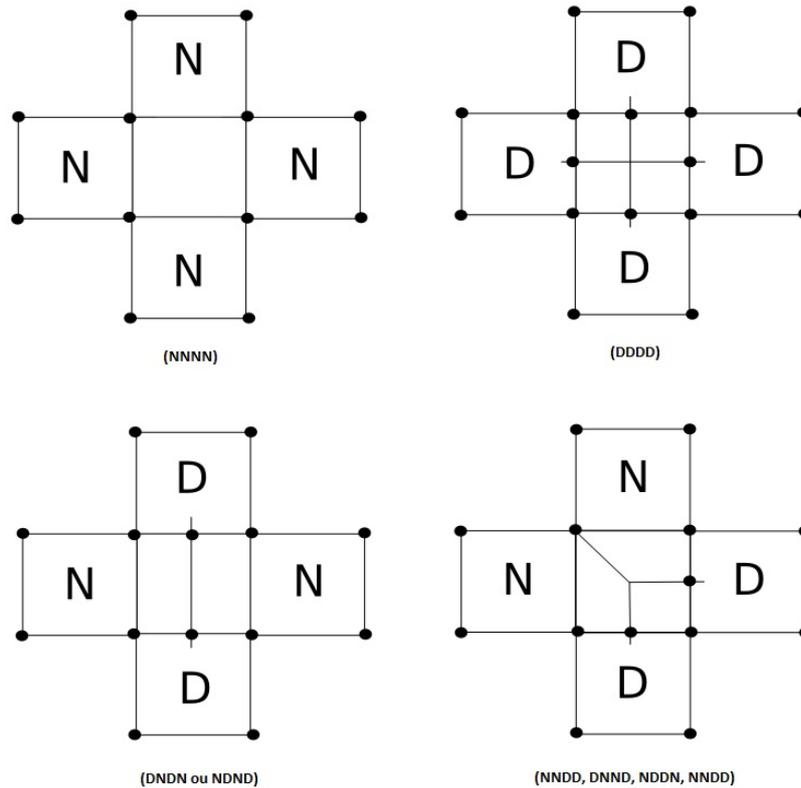


Figura 4.7: Imagem superior esquerda: Nenhuma aresta marcada para subdivisão (NNNN). Imagem inferior esquerda: Dois vizinhos opostos marcam as arestas em comum para subdivisão (DNDN ou NDND). Imagem inferior direita: Dois vizinhos adjacentes marcam as arestas em comum para subdivisão (DDNN, NDDN, NNDD ou DNND). Imagem superior direita: Os quatro vizinhos marcam as arestas em comum para subdivisão (DDDD).

de baixo ou o vizinho da direita e o vizinho da esquerda).

Os templates simétricos do tipo DDNN, NDDN, NNDD ou DNND também representam que dois vizinhos estão marcando as arestas em comum ao quadrilátero para subdivisão, porém esses vizinhos não são opostos. E por fim, os templates simétricos do tipo DDDN, DDND, DNDD ou NDDD representam que existem três vizinhos marcando arestas para subdivisão.

Os templates do tipo DNNN e do tipo DDDN devem ser evitados, assim como os seus simétricos. Nestes casos, não há uma subdivisão que possa ser feita nos quadriláteros marcados com esses tipos que possa manter a malha exclusivamente de quadriláteros, logo, estes tipos precisam ser eliminados.

A seção seguinte detalha o algoritmo de eliminação dos templates inválidos e a garantia de que o resultado de cada iteração gera uma malha consistente.

4.3.2 Eliminação dos casos inválidos

Tendo o conhecimento de todos os quadriláteros que foram marcados como templates inválidos, é feita uma análise individual objetivando a troca destes templates para templates válidos. Essa troca de templates é realizada marcando novas arestas para serem subdivididas (arestas que inicialmente não precisavam ser subdivididas pelo critério de aproximação da superfície), marcando quadriláteros que inicialmente não deveriam ser subdivididos, mas que precisarão ser subdivididos com templates válidos para manter a consistência da malha. Essa atribuição de templates válidos acarreta na subdivisão de novas arestas, podendo tornar válidos ou inválidos os quadriláteros vizinhos que não estavam marcados para subdivisão. Este fato é muito importante, pois pode se propagar para toda a malha.

Esses quadriláteros marcados com templates inválidos são classificados em duas listas. A primeira lista contém os quadriláteros que estão marcados com template DDDN (e seus simétricos). A segunda lista contém os quadriláteros que estão marcados com template DNNN (e seus simétricos).

Refinamento de superfícies de quadriláteros

Analisando os tipos possíveis de template, a única troca possível para um quadrilátero marcado com template DDDN que pode ser realizada é para um template DDDD, pois os demais casos válidos exigiriam a não subdivisão de arestas. Neste contexto, não subdividir uma aresta que inicialmente está marcada para subdivisão contraria a exigência do refinamento, pois se ela foi marcada para subdivisão, o quadrilátero não está bem aproximado (seguindo o critério de erro de aproximação especificado no começo do algoritmo), logo nenhuma substituição de template inválido por válido considera a não subdivisão de arestas que devem ser subdivididas. Assim, todos os quadriláteros da primeira lista terão os seus templates trocados para um template DDDD. Porém, os vizinhos que compartilhavam a aresta em que não era necessária a subdivisão, poderão se tornar inválidos, o que ocasiona a sua inclusão em uma das duas listas. Se estes vizinhos se tornarem válidos, eles precisam ser removidos da lista de inválidos correspondente.

Após todos os templates do tipo DDDN da malha terem sido processados da maneira supracitada, restará uma lista contendo todos os quadriláteros marcados como DNNN (e seus simétricos). Essa lista é processada quadrilátero por quadrilátero de forma a causar a menor propagação possível para o restante da malha. Inevitavelmente, a correção destes templates cria uma região de transição entre as diferentes densidades de quadriláteros (adaptatividade), pois estes templates estão entre uma região marcada para ser subdividida e outra região onde os quadriláteros já possuem tamanho aceitável.

Quadriláteros marcados como DNNN, podem ser substituídos por DDNN, DNND ou DDDD (o caso DDDN é inválido). As escolhas com melhor impacto no número de arestas que serão subdivididas, mesmo não sendo necessária sua divisão (segundo o critério de refinamento), são DDNN e DNND. Neste caso sempre é escolhida uma mesma direção, pois assim a propagação é minimizada quando avaliamos os possíveis vizinhos. Com esta troca de template, temos o mesmo problema descrito anteriormente, vizinho válidos podem se tornar inválidos, assim como vizinhos inválidos podem se tornar válidos.

Apesar de existir a possibilidade da propagação da marcação de subdivisão de arestas ocorrer e todos os quadriláteros da malha serem marcados para subdivisão DDDD, em-

Refinamento de superfícies de quadriláteros

piricamente vimos que isso não acontece na maioria dos casos. Os casos DDDN são os de menor número e sua eliminação provoca a subdivisão de apenas uma aresta. A grande maioria dos casos inválidos são do tipo DNNN, mas escolhendo sempre a subdivisão em um mesmo sentido, por ex. DDNN, temos que a propagação tende-se a concentrar em torno do vértice entre as duas arestas subdivididas.

É possível então organizar em um pseudo-algoritmo a forma com que são tratados estes casos inválidos. É válido ressaltar que essa solução busca fazer a troca dos templates inválidos para templates válidos com o menor impacto possível de propagação para o restante da malha. A lógica está organizada como segue, simplificando para os casos simétricos:

Enquanto listaDDDN e lista DNNN não estão vazias

Se a listaDDDN não está vazias

Remove template da listaDDDN

Troca template de DDDN para DDDD

Senão

Remove template da listaDNNN

Troca template de DNNN para DDNN

Realiza a troca de template nos quadriláteros vizinhos (localmente afetados)

Se os vizinhos tiverem seus templates trocados para DDDN

Inserir vizinhos na listaDDDN

Se os vizinhos tiverem seus templates trocados para DNNN

Inserir vizinhos na listaDNNN

Ao final deste processo, a malha estará com todos os seus quadriláteros marcados com templates válidos e estará pronta para o processo de subdivisão. A figura 4.8 mostra um exemplo de malha marcada antes do processamento dos casos inválidos e a figura 4.9 mostra um exemplo de malha marcada após o processamento desses casos inválidos. Devido a este processo de mudança de templates para que não haviam casos inválidos e inconsistência na malha, é possível notar um avanço nas regiões marcadas para subdivisão. Porém, é possível notar também que esse avanço, na prática, ainda mantém a adaptatividade de refinamento em determinadas regiões.

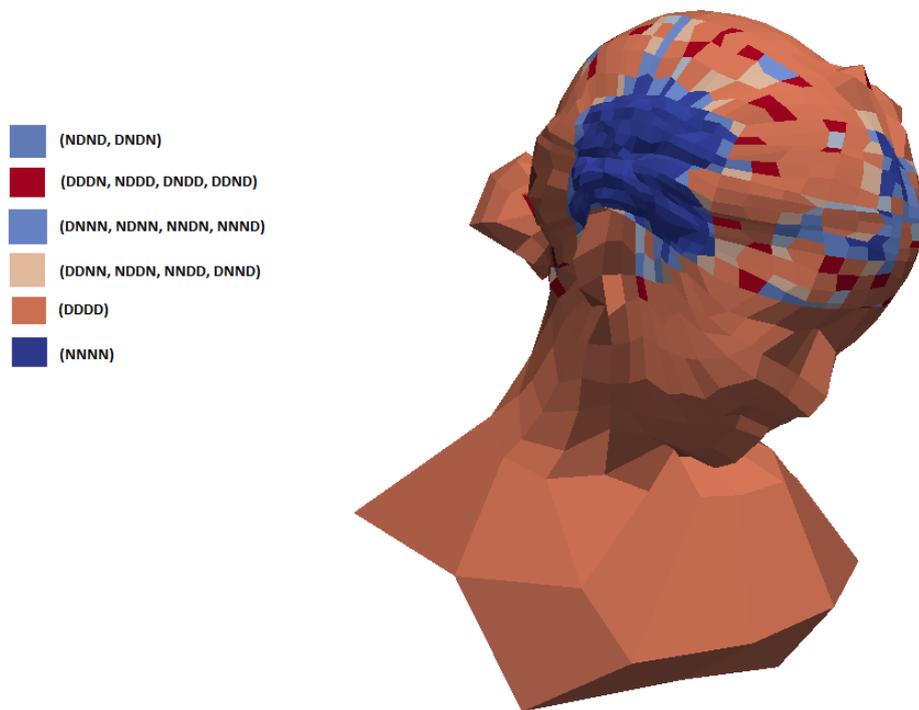


Figura 4.8: Exemplo de malha antes do processamento dos casos inválidos. As diferentes cores representam as marcações de templates que cada quadrilátero recebeu de acordo com seus vizinhos.

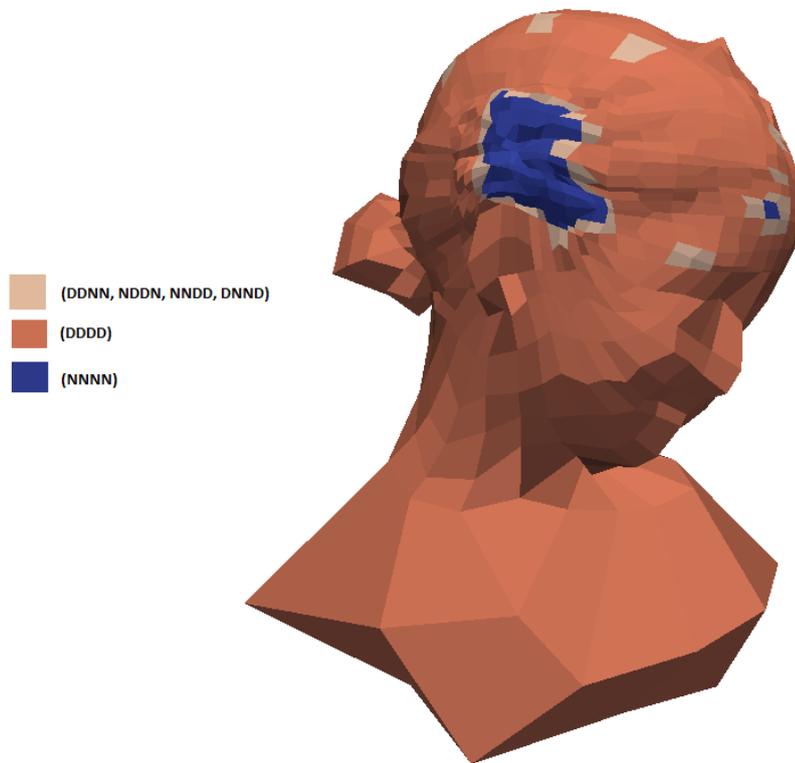


Figura 4.9: Utilizando a mesma malha da figura 4.8, esta imagem representa a malha pós processamento dos casos inválidos.

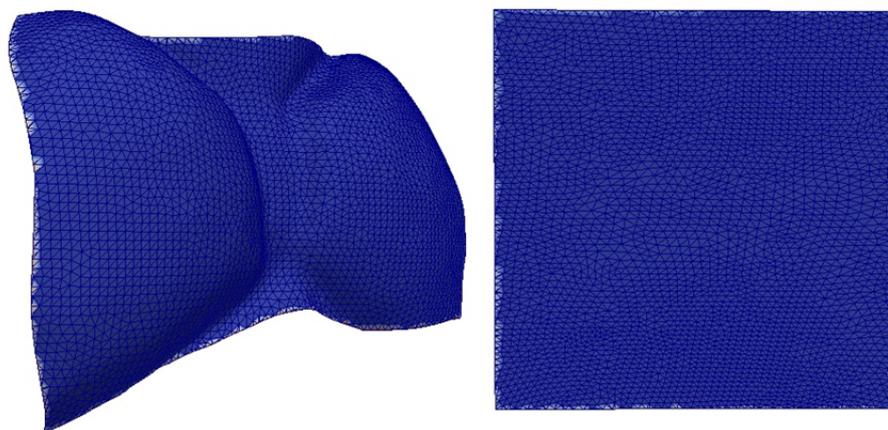


Figura 4.10: Exemplo de um *patch* que passou pelo processo de parametrização.

Na próxima subseção, serão discutidas as técnicas envolvidas no processo de subdivisão da malha de quadriláteros. Essas técnicas incluem a escolha do ponto de subdivisão e do algoritmo de parametrização.

4.3.3 Subdivisão da malha de quadriláteros

Após eliminar os casos inválidos, todos os quadriláteros da malha estarão marcados com templates válidos a serem subdivididos. Assim, os templates válidos (DDDD, DNND, DDNN, NNDD, NDDN, DNDN, NDND, NNNN) podem ser aplicados sem que haja inconsistência da malha, mantendo-a exclusivamente de quadriláteros.

Como cada quadrilátero representa uma região de triângulos da malha de triângulos, e o objetivo é que a malha de quadriláteros se aproxime o máximo possível da malha de triângulos obedecendo ao critério de qualidade especificado no início do algoritmo, é necessário que a subdivisão passe por pontos que estejam na superfície triangulada. Para facilitar a escolha do ponto, transformamos o problema tridimensional para um problema bidimensional através de uma parametrização da superfície triangular em um plano. Este processo auxilia também na subdivisão da superfície para compor as regiões dos novos quadriláteros. A figura 4.10 ilustra um exemplo de parametrização.

Com a parametrização concluída, são calculadas as distâncias geodésicas entre os vértices participantes da subdivisão gerando retas na parametrização que não necessaria-

mente passam somente pelas arestas originais da malha. Se essas retas cruzarem com um vértice ou acompanharem as arestas originais, tanto as arestas quanto os vértices serão mantidos. Porém, se a intersecção desta reta com a aresta original for um ponto, então um novo vértice é criado e, caso seja necessário, são criadas novas arestas e faces que mantenham a malha consistente.

Neste projeto, a escolha do ponto para subdivisão dos templates foi limitado ao ponto central da geodésica traçada no espaço parametrizado. Porém, a escolha deste ponto poderia estar relacionado a qualidade dos quadriláteros gerados, de forma a maximizar a mesma.

Na próxima subseção é apresentado todo o processo iterativo de uma maneira macro.

4.3.4 Processo iterativo

Para efeito de síntese do algoritmo implementado e apresentação de uma visão geral, é possível estruturar o processo iterativo de uma maneira macro conforme segue:

1. Marcar os quadriláteros que aproximam mal sua superfície correspondente (seguindo um critério de erro)
2. Enquanto existirem quadriláteros marcados para subdivisão (e o número máximo de iterações não for alcançado)
 - (a) Marcar para subdividir todas as arestas dos quadriláteros marcados para subdivisão (template DDDD)
 - (b) Classificar cada quadrilátero segundo as arestas marcadas para subdividir
 - (c) Eliminar os casos inválidos
 - (d) Subdividir todos os quadriláteros DDDD, DDNN e DNDN
 - (e) Marcar os quadriláteros que aproximam mal sua superfície correspondente (seguindo um critério de erro)

O algoritmo termina quando o número de iterações especificado pelo usuário for alcançado ou quando a superfície de triângulos já estar sendo bem aproximada pela malha de quadriláteros (seguindo um critério de erro). Ou seja, podem ocorrer casos em que o erro especificado faça com que a superfície que representa um quadrilátero seja tão pequena que a medida de erro já se torna suficiente. Mesmo que o número de iterações



Figura 4.11: Exemplo do processo iterativo desde a malha base até a malha final, com 10 iterações (partindo do modelo inicial, acima e a esquerda).

especificado for muito alto ou que a medida de erro for próxima de zero, o algoritmo chegará a um ponto onde não haverá mais diferença da malha de quadriláteros para a malha de triângulos fazendo com que o mesmo termine.

Ao final deste processo iterativo, a malha resultante ainda possui características a serem melhoradas como por exemplo o alinhamento das arestas. Essas melhorias são alcançadas através de um processo de suavização final. O processo de suavização foi isolado de forma que o algoritmo a ser utilizado pode ser facilmente trocado por outro. Neste trabalho, optamos por utilizar o algoritmo proposto por [49].

A figura 4.11 apresenta cada iteração do processo iterativo descrito.

No próximo capítulo serão exibidos os resultados obtidos com a execução deste projeto. Também serão comentados os parâmetros utilizados, além de um histograma de qualidade da malha final para cada modelo.

Resultados

Em todos os resultados aqui apresentados, a malha base foi gerada manualmente através da escolha de vértices na superfície a ser aproximada. Verificamos que malhas base mais complexas pouco impactam no refinamento, pois em poucas iterações o modelo mais simples se torna mais complexo. Algumas propriedades interessantes podem ser exploradas, como o alinhamento da malha de quadriláteros em linhas de características do modelo. Como todos os templates utilizados não realizam *flip* de arestas, se a malha base repossuir estas linhas, o refinamento não irá removê-las. Todos os modelos que estão apresentados são topologicamente equivalentes ao cubo utilizado como malha base. Esta limitação é devida a implementação realizada, onde não foram tratados os bordos. Essa forma geométrica foi escolhida como malha base em todos os modelos pela sua facilidade de criação manual e por representar a menor malha superficial exclusivamente de quadriláteros possível.

O critério de erro utilizado em todos os testes foi a média da distância ao quadrado dos pontos da malha de triângulos com o respectivo quadrilátero que está aproximando esta região. Quanto menor esta somatória, mais bem aproximado este quadrilátero deverá

Resultados

estar. Esta etapa do *pipeline* pode ser facilmente trocada por outra medida de erro, uma vez que a implementação desse critério de erro está isolada do algoritmo principal deste projeto.

O algoritmo de parametrização utilizado nos resultados apresentados foi o proposto por Floater [18]. Essa parametrização tem como característica principal a preservação da área de seus elementos. Na prática, os diferentes algoritmos de parametrização utilizados durante a implementação deste projeto não apresentaram uma mudança significativa no resultado final dos modelos, mas uma análise mais profunda é necessária, pois diferentes parametrizações tem a capacidade de guiar a subdivisão da superfície correspondente a cada quadrilátero. Vale ressaltar que a escolha deste algoritmo é facilmente alterada na implementação.

A seguir são exibidos os modelos “Bimba”, “Gargoyle” e “Santa”. Estes modelos foram extraídos da base de dados Aim@Shape¹.

A figura 5.1 apresenta o modelo de referência Bimba com erro de 10^{-5} e 10 iterações no máximo. No exemplo apresentado, o critério de erro provocou o término das iterações. A figura 5.2 apresenta o modelo final gerado pelo processo iterativo, junto com um histograma exibindo a qualidade dos quadriláteros gerados. Foi utilizado a medida *ScaledJacobian*[25].

O modelo Gargoyle é apresentado na figura 5.3, utilizando erro de 10^{-4} . Na figura 5.4 é exibido o modelo final Gargoyle gerado pelo processo iterativo com um histograma exibindo a qualidade dos quadriláteros gerados.

A figura 5.5 apresenta o modelo de referência Santa executado com 17 iterações. As imagens intermediárias entre a etapa 10 e a etapa 16 foram suprimidas. Na figura 5.6 temos o modelo final gerado pelo processo iterativo com um histograma de qualidade dos quadriláteros gerados. Podemos observar na figura 5.7 a diferença da malha final gerada a partir do modelo Santa com a malha suavizada. Também é possível visualizar na figura 5.7 a transição entre regiões de diferentes densidades, sendo delimitada por templates

¹<http://www.aimatshape.net/>

Resultados

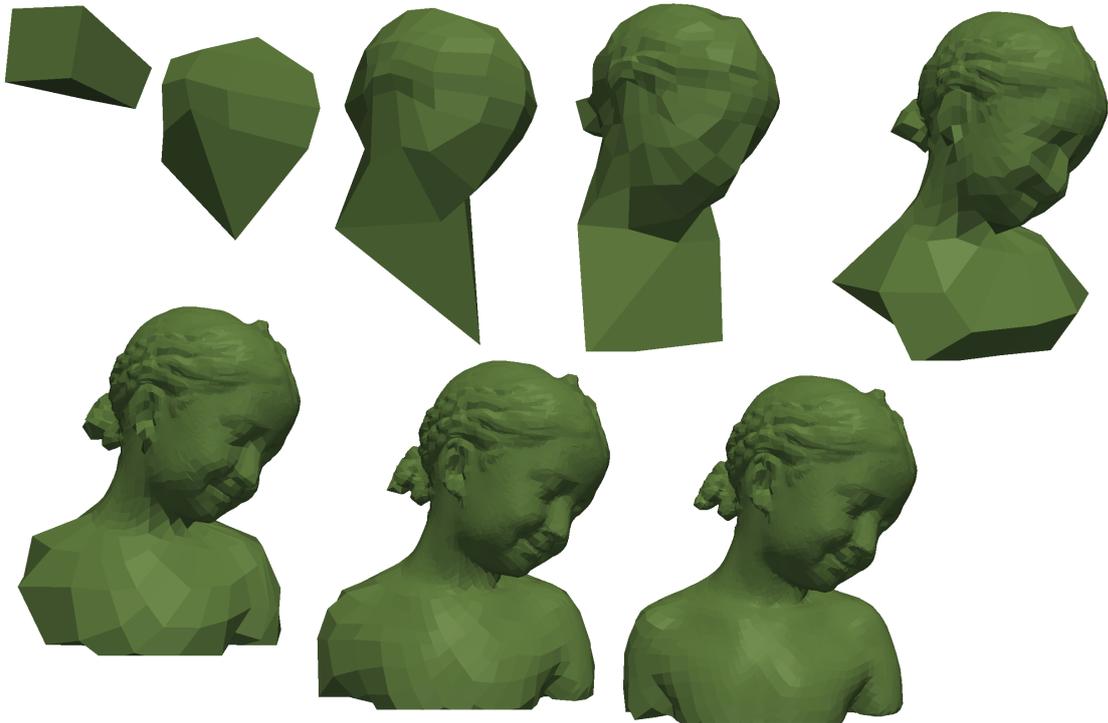


Figura 5.1: Exemplo do processo iterativo desde a malha base até a malha final, com 10 iterações.

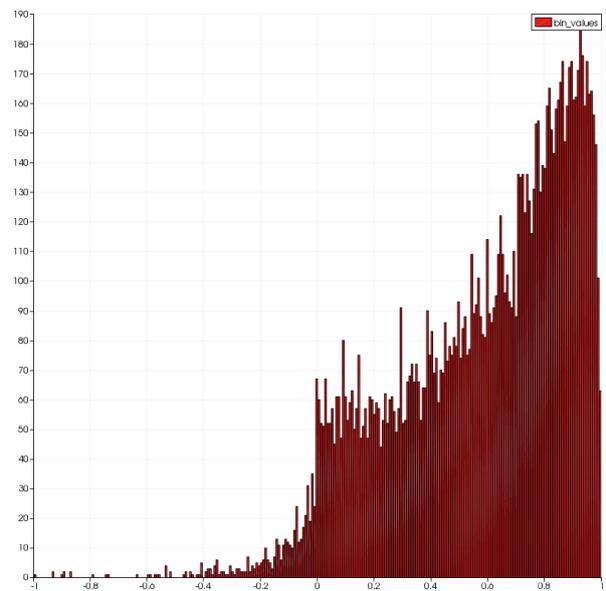


Figura 5.2: Resultado com o modelo da figura 5.1 juntamente com um histograma de qualidade.

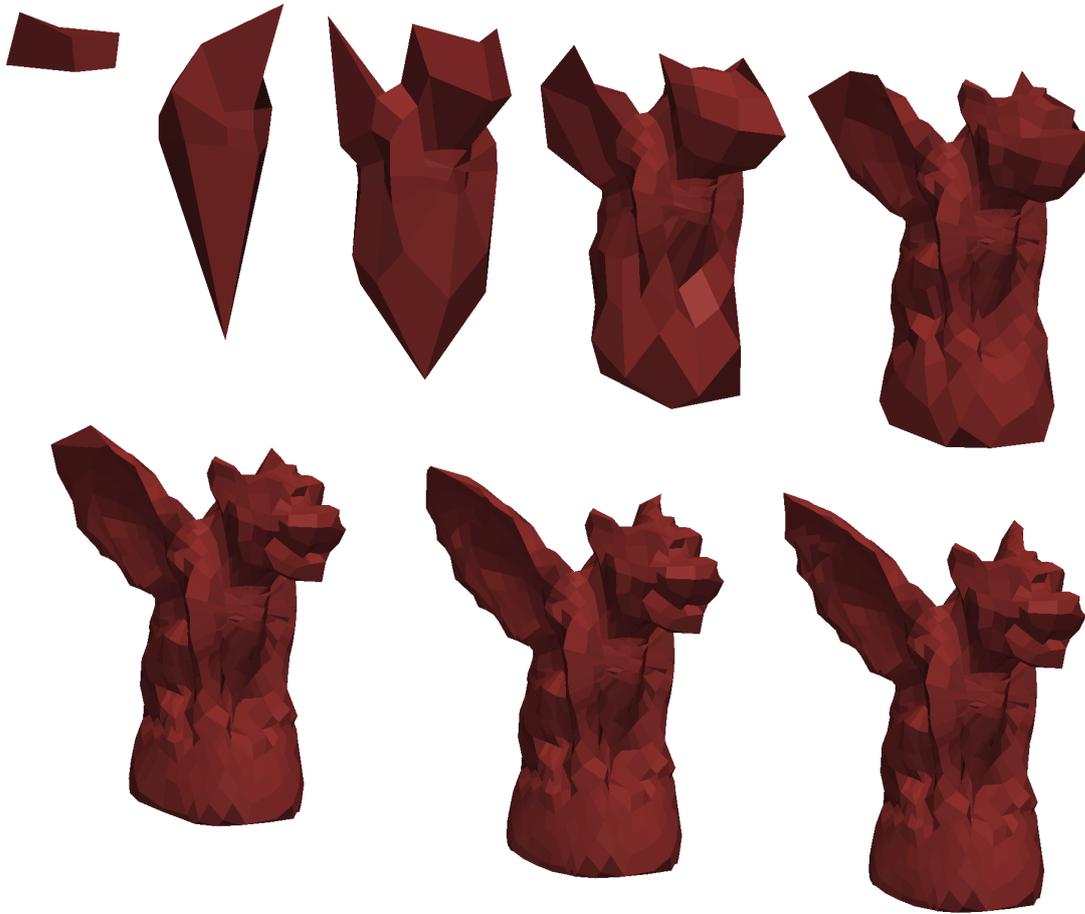


Figura 5.3: Exemplo do processo iterativo desde a malha base até a malha final, com 10 iterações.

Resultados

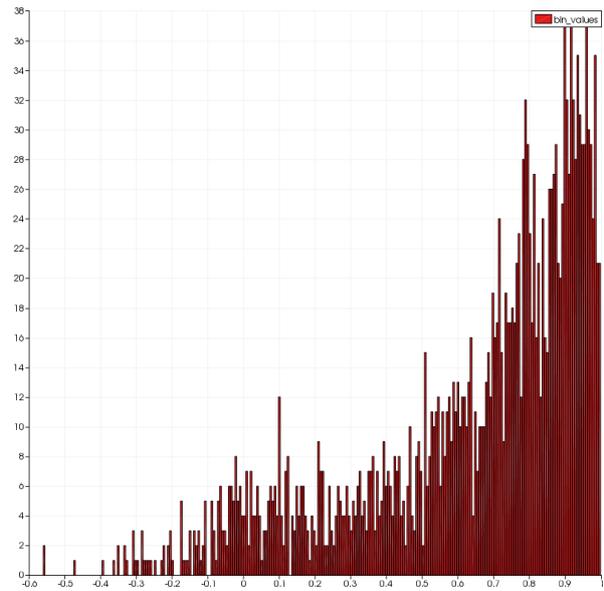


Figura 5.4: Resultado com o modelo da figura 5.3 juntamente com um histograma de qualidade.

DDNN.

No capítulo seguinte serão apresentadas as conclusões e os trabalhos futuros.

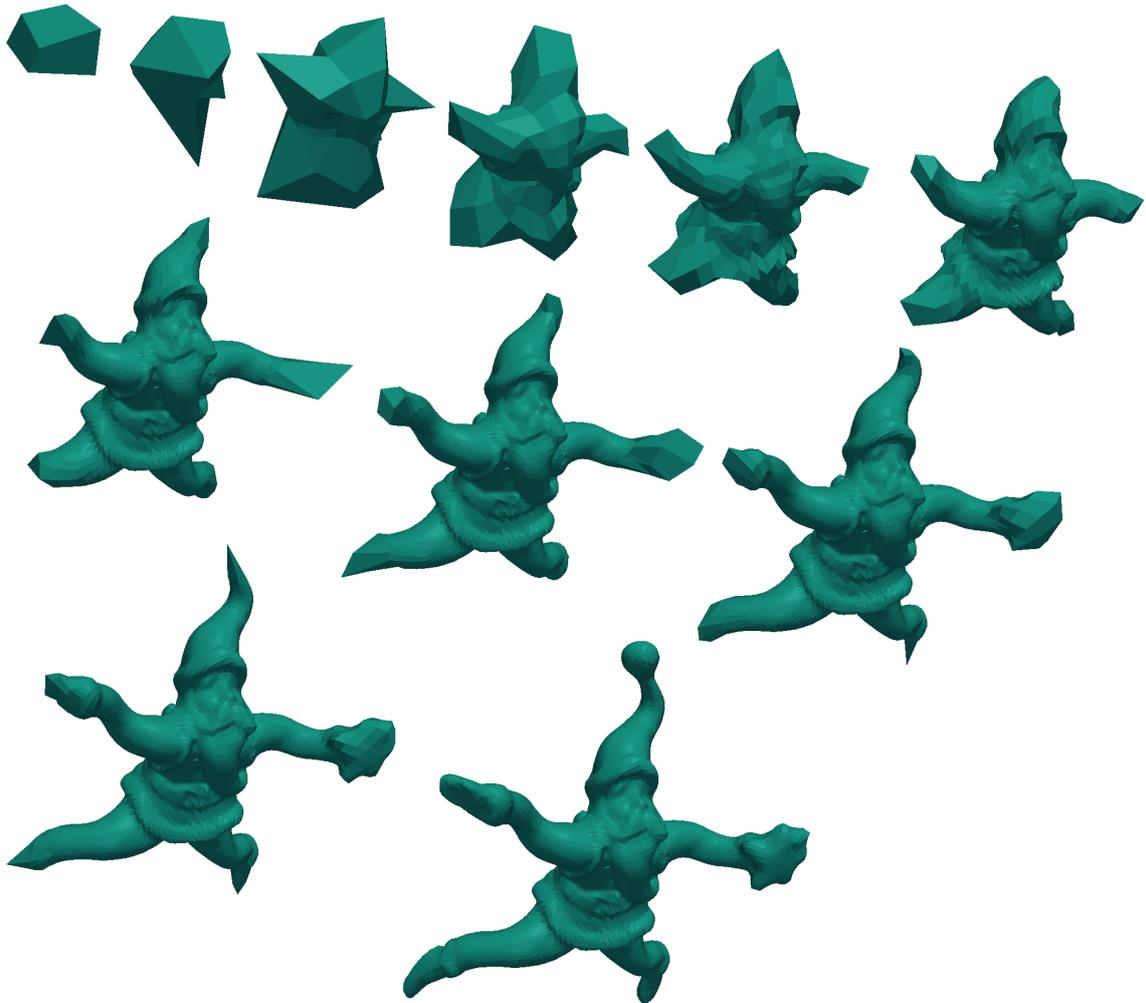


Figura 5.5: Exemplo do processo iterativo desde a malha base até a malha final, com 10 iterações.

Resultados

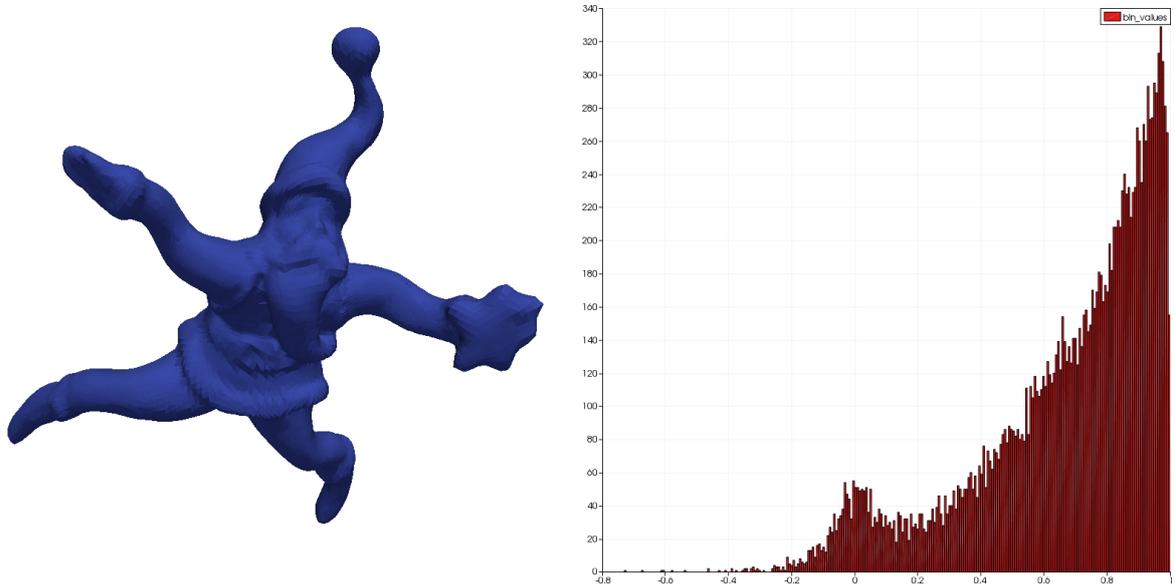


Figura 5.6: Resultado com o modelo da figura 5.5 juntamente com um histograma de qualidade.

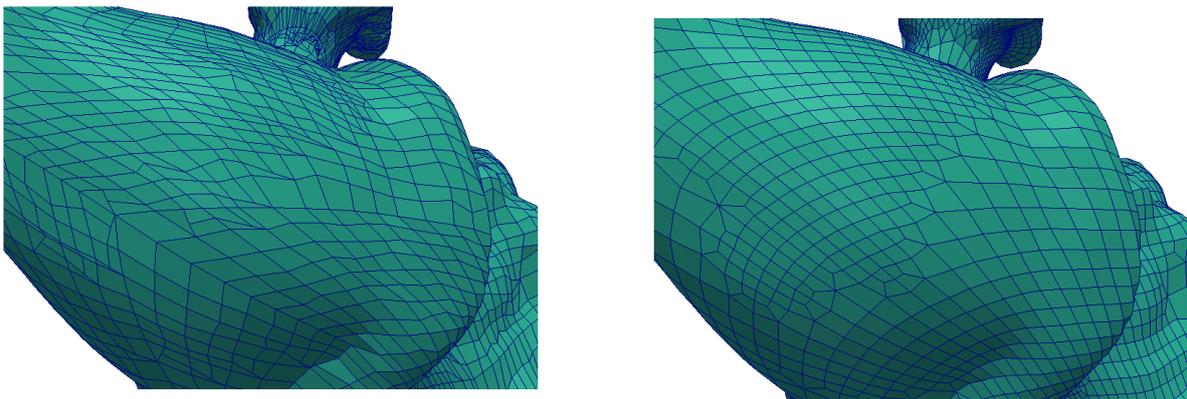


Figura 5.7: Zoom do modelo Santa antes e depois da suavização.

Conclusões e trabalhos futuros

Este trabalho apresentou um *pipeline* para geração de malhas superficiais de quadriláteros a partir de malhas de triângulos. Para este *pipeline* foi desenvolvido um novo método de refinamento de malhas de quadriláteros, com subdivisão baseada em templates.

Os resultados apresentados mostram a execução deste método em vários modelos. A qualidade final dos elementos gerados pode ainda ser melhorada mediante mudanças no algoritmo de parametrização e no corte da malha. Ainda seria possível analisar a construção da malha base inicial, tendo visto que quanto maior a área coberta pelas regiões dos quadriláteros iniciais, mais bem controlados serão os cálculos das geodésicas e a concentração de subdivisões em locais específicos.

O método de eliminação de casos inválidos mostrou-se eficaz para todos os modelos que foram executados. Embora ainda não haja um algoritmo com garantias teóricas, em todos os testes a propagação de subdivisões ficou limitada na borda das regiões a serem refinadas.

A seguir encontram-se os principais trabalhos para o aprimoramento do método apresentado.

6.1 Trabalhos futuros

O trabalho desenvolvido neste mestrado é o início de uma nova abordagem para geração de malhas superficiais de quadriláteros. Por ser um *pipeline* completo, várias etapas podem ser investigadas com mais detalhes e diversas simplificações foram assumidas. A seguir são listadas os principais aspectos que podem ser abordados ou aprimorados:

- Eliminação de casos inválidos: prover garantias sobre a propagação do processo de subdivisão;
- Obtenção automática do modelo inicial: garantindo que o modelo inicial possui a mesma topologia da malha a ser aproximada;
- Medidas de erro de aproximação de superfícies: investigar o impacto de diferentes medidas de erro existente na literatura na aproximação do modelo seguindo o esquema proposto;
- Templates: considerar a qualidade dos quadriláteros gerados ao subdividir uma superfície em cada template;
- Diferentes topologias: suportar objetos de outras topologias, permitir malhas superficiais com borda;
- Parametrizações: explorar as diferenças que podem ser obtidas ao aplicar diferentes parametrizações;
- *Features*: detectar automaticamente (ou prover uma interface semi-automática) linhas de característica dos objetos;
- Suavização: investigação das técnicas de suavização;

Referências Bibliográficas

- [1] Pierre Alliez, Giuliana Ucelli, Craig Gotsman, and Marco Attene. Recent advances in remeshing of surfaces. *Shape Analysis and Structuring*, pages 53–82, 2008.
- [2] Peggy L. Baehmann, Scott L. Wittchen, Mark S. Shephard, Kurt R. Grice, and Mark A. Yerry. Robust, geometrically based, automatic two-dimensional mesh generation. *International Journal for Numerical Methods in Engineering*, 24(6):1043–1078, 1987.
- [3] Barry and Joe. Quadrilateral mesh generation in polygonal regions. *Computer-Aided Design*, 27(3):209 – 222, 1995.
- [4] Chakib Bennis, Jean-Marc Vézien, and Gérard Iglésias. Piecewise surface flattening for non-distorted texture mapping. *SIGGRAPH Comput. Graph.*, 25:237–246, July 1991.
- [5] Ted D. Blacker and Michael B. Stephenson. Paving: A new approach to automated quadrilateral mesh generation. *International Journal for Numerical Methods in Engineering*, 32(4):811–847, 1991.
- [6] David Bommes, Henrik Zimmer, and Leif Kobbelt. Mixed-integer quadrangulation. *ACM Trans. Graph.*, 28(3):77:1–77:10.
- [7] Houman Borouchaki, Houman Borouchaki, Pascal J. Frey, and Pascal J. Frey. Adaptive triangular-quadrilateral mesh generation. *International Journal of Numerical Methods in Engineering*, 41:915–934, 1996.
- [8] R. J. CASS, S. E. BENZLEY, R. J. MEYERS, and T. D. BLACKER. Generalized 3-d paving: An automated quadrilateral surface mesh generation algorithm. *International Journal for Numerical Methods in Engineering*, 39(9):1475–1489, 1996.
- [9] E. Catmull and J. Clark. Recursively generated b-spline surfaces on arbitrary topological meshes. *Computer-Aided Design*, 10(6):350 – 355, 1978.
- [10] David Cohen-Steiner, Pierre Alliez, and Mathieu Desbrun. Variational shape approximation. *ACM Trans. Graph.*, 23:905–914, August 2004.
- [11] Joel Daniels, Cláudio T. Silva, Jason Shepherd, and Elaine Cohen. Quadrilateral mesh simplification. *ACM Trans. Graph.*, 27:148:1–148:9, December 2008.

Referências Bibliográficas

- [12] P. Degener, J. Meseth, and R. Klein. An adaptable surface parameterization method. In *In Proceedings of the 12th International Meshing Roundtable*, pages 201–213, 2003.
- [13] Mathieu Desbrun, Mark Meyer, and Pierre Alliez. Intrinsic parameterizations of surface meshes, 2002.
- [14] Shen Dong, Peer-Timo Bremer, Michael Garland, Valerio Pascucci, and John C. Hart. Spectral surface quadrangulation. *ACM Trans. Graph.*, 25(3):1057–1066, July 2006.
- [15] Matthias Eck, Tony DeRose, Tom Duchamp, Hugues Hoppe, Michael Lounsbery, and Werner Stuetzle. Multiresolution analysis of arbitrary meshes. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques, SIGGRAPH '95*, pages 173–182, New York, NY, USA, 1995. ACM.
- [16] G.E. Farin. *Curves and surfaces for CAGD: a practical guide*. The Morgan Kaufmann series in computer graphics and geometric modeling. Morgan Kaufmann, 2002.
- [17] Gerald Farin. *Curves and surfaces for CAGD: a practical guide*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 5th edition, 2002.
- [18] Michael S. Floater. Parametrization and smooth approximation of surface triangulations. *Computer Aided Geometric Design*, 14:231–250, 1997.
- [19] Igor Guskov. An anisotropic mesh parameterization scheme. In *In Proceedings of the 11th International Meshing Roundtable*, pages 325–332, 2002.
- [20] K. Hormann and G. Greiner. Quadrilateral remeshing. In B. Girod, G. Greiner, H. Niemann, and H.-P. Seidel, editors, *Proceedings of Vision, Modeling, and Visualization 2000*, pages 153–162, Saarbrücken, Germany, November 2000. infix.
- [21] D.P. Huttenlocher, G.A. Klanderman, and W.J. Rucklidge. Comparing images using the hausdorff distance. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 15(9):850–863, 1993.
- [22] J. Daniels II, M. Lizier, M. Siqueira, C.T. Silva, and L.G. Nonato. Template-based quadrilateral meshing. *Computers and Graphics*, 35(3):471 – 482, 2011. [jce:title;Shape Modeling International \(SMI\) Conference 2011;jce:titlej](#).
- [23] Barry Joe. Delaunay triangular meshes in convex polygons. *SIAM Journal on Scientific and Statistical Computing*, 7(2):514–539, 1986.
- [24] Liliya Kharevych, Boris Springborn, and Peter Schröder. Discrete conformal mappings via circle patterns. *ACM Trans. Graph.*, 25:412–438, April 2006.
- [25] Patrick M. Knupp. Achieving finite element mesh quality via optimization of the jacobian matrix norm and associated quantities. part ii: A framework for volume mesh optimization and the condition number of the jacobian matrix. *International Journal for Numerical Methods in Engineering*, 48:1165–1185, julho 2000.

Referências Bibliográficas

- [26] Huiling Le and David G. Kendall. The riemannian structure of euclidean shape spaces: A novel environment for statistics. *The Annals of Statistics*, 21(3):pp. 1225–1271, 1993.
- [27] Bruno Lévy and Jean-Laurent Mallet. Non-distorted texture mapping for sheared triangulated meshes. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '98, pages 343–352, New York, NY, USA, 1998. ACM.
- [28] Bruno Lévy, Sylvain Petitjean, Nicolas Ray, and Jérôme Maillot. Least squares conformal maps for automatic texture atlas generation. *ACM Trans. Graph.*, 21:362–371, July 2002.
- [29] Xinghua Liang and Yongjie Zhang. Hexagon-based all-quadrilateral mesh generation with guaranteed angle bounds. *Computer Methods in Applied Mechanics and Engineering*, 200(23–24):2005 – 2020, 2011.
- [30] Jérôme Maillot, Hussein Yahia, and Anne Verroust. Interactive texture mapping. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '93, pages 27–34, New York, NY, USA, 1993. ACM.
- [31] Martin Marinov and Leif Kobbelt. A robust two-step procedure for quad-dominant remeshing. *Computer Graphics Forum*, 25:537–546, 2006.
- [32] Dimas Martínez, Luiz Velho, and Paulo C. Carvalho. Computing geodesics on triangular meshes. *Computers and Graphics*, 29(5):667 – 675, 2005.
- [33] Dietrich Nowottny. Quadrilateral mesh generation via geometrically optimized domain decomposition. In *Proceedings, 6th International Meshing Roundtable, Sandia National Laboratories*, pages 309–320, 1997.
- [34] Steven J. Owen. A survey of unstructured mesh generation technology. In *INTERNATIONAL MESHING ROUNDTABLE*, pages 239–267, 1998.
- [35] Steven J. Owen, Matthew L. Staten, Scott A. Canann, and Sunil Saigal. Advancing front quadrilateral meshing using triangle transformations, 1998.
- [36] Ulrich Pinkall, Strasse Des Juni, and Konrad Polthier. Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics*, 2:15–36, 1993.
- [37] Suneeta Ramaswami, Marcelo Siqueira, Tessa Sundaram, Jean Gallier, and James Gee. Constrained quadrilateral meshes of bounded size, 2005.
- [38] Burt Rodin and Dennis Sullivan. The convergence of circle packings to the riemann mapping. *Journal of Differential Geometry*, 26:349–360, 1987.
- [39] Pedro V. Sander, John Snyder, Steven J. Gortler, and Hugues Hoppe. Texture mapping progressive meshes. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '01, pages 409–416, New York, NY, USA, 2001. ACM.

Referências Bibliográficas

- [40] Josep Sarrate and Antonio Huerta. Efficient unstructured quadrilateral mesh generation. *International Journal for Numerical Methods in Engineering*, 49(10):1327–1350, 2000.
- [41] Philip J. Schneider and David H. Eberly. *Geometric Tools for Computer Graphics*. Morgan Kaufmann, 2003.
- [42] Alla Sheffer, Bruno Lévy, Maxim Mogilnitsky, and Alexander Bogomyakov. Abf++: fast and robust angle based flattening. *ACM Trans. Graph.*, 24:311–330, April 2005.
- [43] Alla Sheffer, Emil Praun, and Kenneth Rose. Mesh parameterization methods and their applications. In *Foundations and Trends in Computer Graphics and Vision*, page 2006. Now Publishers, 2006.
- [44] Jeffrey A. Talbert and Alan R. Parkinson. Development of an automatic, two-dimensional finite element mesh generator using quadrilateral elements and bezier curve boundary definition. *International Journal for Numerical Methods in Engineering*, 29(7):1551–1567, 1990.
- [45] T.K.H. Tam and C.G. Armstrong. 2d finite element mesh generation by medial axis subdivision. *Advances in Engineering Software and Workstations*, 13(5–6):313 – 324, 1991.
- [46] Marco Tarini, Nico Pietroni, Paolo Cignoni, Daniele Panozzo, and Enrico Puppo. Practical quad mesh simplification. *Computer Graphics Forum*, 29(2):407–418, 2010.
- [47] W. T. Tutte. How to draw a graph. In *Proceedings London Mathematical Society*, pages 743–768, 1963.
- [48] Luiz Velho and Denis Zorin. 4-8 subdivision. *Computer Aided Geometric Design*, pages 397–427, 2001.
- [49] Yongjie Zhang, Rajit Bajaj, and Guoliang Xu. Surface smoothing and quality improvement of quadrilateral/hexahedral meshes with geometric flow. In *In Proceedings, 14th International Meshing Roundtable*, pages 449–468. John Wiley and Sons, 2005.
- [50] Kun Zhou, John Synder, Baining Guo, and Heung-Yeung Shum. Iso-charts: stretch-driven mesh parameterization using spectral analysis. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, SGP '04, pages 45–54, New York, NY, USA, 2004. ACM.
- [51] J. Z. Zhu, O. C. Zienkiewicz, E. Hinton, and J. Wu. A new approach to the development of automatic quadrilateral mesh generation. *International Journal for Numerical Methods in Engineering*, 32(4):849–866, 1991.
- [52] G. Zigelman, R. Kimmel, and N. Kiryati. Texture mapping using surface flattening via multidimensional scaling. *IEEE Transactions on Visualization and Computer Graphics*, 8:198–207, April 2002.