SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito: 11/03/2005

Assinatura:

**Reconstrução a partir de nuvem de pontos
com a utilização de funções de Morse
discretas**

Helton Hideraldo Biscaro

Orientador: *Prof. Dr. Antônio Castelo Filho*
Co-orientador: *Prof. Dr. Luis Gustavo Nonato*

Tese apresentada ao Instituto de Ciências Matemáticas e de
Computação - ICMC-USP, como parte dos requisitos para
obtenção do título de Doutor em Ciências – Ciências de
Computação e Matemática Computacional.

USP – São Carlos
Março de 2005

Aluno: Helton Hideraldo Bísaro

A Comissão Julgadora:

Prof. Dr. Antonio Castelo Filho

Prof. Dr. Luis Gustavo Nonato

Prof. Dr. Luiz Carlos Pacheco Rodrigues Velho

Prof. Dr. Renato Cardoso Mesquita

Prof. Dr. Manuel Menezes de Oliveira Neto

Handwritten signatures of the five judges in red ink, positioned over horizontal lines. The signatures are: Antonio Castelo Filho, Luis Gustavo Nonato, Luiz Carlos Pacheco Rodrigues Velho, Renato Cardoso Mesquita, and Manuel Menezes de Oliveira Neto.

*Dedico esta tese à minha família:
minha mãe Elide, minha irmã Aline e
meu pai Sidney (in memoriam). Que me
ensinaram muitas das coisas realmente
importantes na vida.*

Agradecimentos

Gostaria de prestar meus sinceros agradecimentos as seguintes pessoas e instituições:

Em primeiro lugar a Deus por ter me proporcionado saúde, energia, e disposição para enfrentar todos os desafios surgidos no decorrer do trabalho;

Especialmente a minha namorada Edsandra, por ter me dado auxílio e inspiração tanto nos momentos fáceis, quanto nos difíceis;

Ao meu orientador Prof Dr. Antonio Castelo Filho e ao meu co-orientador, o Prof Dr Luis Gustavo Nonato, por terem me dado oportunidade de realizar este trabalho e serem exemplos de dedicação e perseverança;

Ao meu amigo e ex-orientador Prof Dr. Marcelo José Saia por ter sempre me incentivado e aconselhado nos momentos necessários;

Aos meus companheiros de república Marcio e Edivaldo cuja convivência nos tornaram grandes amigos;

Aos meus companheiros de laboratório: Igor, João Paulo, Alex, Cassio, Fernando, Silvino, Fabrício, Gerson, Gil, Marcela, Fernanda, Ana Paula, Patrícia e Ana Alice, que estiveram sempre prontos a me esclarecer dúvidas e dar boas sugestões no decorrer do desenvolvimento deste trabalho, certamente não teria realizado este trabalho desta forma sem a ajuda deles;

Aos os amigos do futebol e da escalada, que conviveram comigo durante esses anos, me deram forças nos momentos difíceis e comemoraram comigo nos momentos alegres;

Aos amigos do departamento de Matemática, Maurício, Marcos, João, José Paulo, Rodrigo, Fabio, Elíris e Helenice;

A todas as pessoas, citadas aqui ou não, que conviveram comigo nesse período e que contribuíram de alguma forma para o desenvolvimento deste projeto;

Finalmente, a Fapesp por ter financiado o projeto.

*“Considere-se que não há nada mais
difícil, nem de sucesso mais duvidoso,
nem mais arriscado, do que iniciar uma
nova ordem das coisas.”*

Nicolau Maquiavel, 1469-1527

Sumário

Capítulo 1	Introdução	1
1.1	Considerações Iniciais	1
1.2	Motivação	1
1.3	Contribuições	3
1.4	Organização da Tese	4
Capítulo 2	Conceitos Fundamentais	7
2.1	Considerações Iniciais	7
2.2	Conceitos de Topologia	7
2.3	Conceitos de Geometria Computacional	9
2.3.1	Triangulação de Delaunay e Diagrama de Voronoi	11
2.4	Conceitos de Álgebra	15
2.5	Conceitos de Homologia Simplicial	18
2.5.1	Operador Bordo	19
2.5.2	Grupos de Homologia	19
2.6	Considerações Finais	20
Capítulo 3	Levantamento Bibliográfico	21
3.1	Considerações Iniciais	21
3.2	Trabalhos Correlatos Bidimensionais	22
3.3	Trabalhos correlatos tridimensionais	28
3.3.1	Métodos baseados em modelos deformáveis	28
3.3.2	Métodos baseados em zeros de funções	30
3.3.3	Métodos incrementais	32
3.3.4	Métodos de esculpimento	35
3.3.5	Crust Tridimensional	42
3.3.6	Power Crust	44
3.3.7	Cocone	46
3.3.8	Tight Cocone	48
3.3.9	Comparações	49
3.4	Considerações Finais	50
Capítulo 4	Teoria de Morse Discreta	51
4.1	Considerações Iniciais	51
4.2	Preliminares	51
4.2.1	Funções de Morse Combinatórias	52
4.3	Teoremas de Morse	53
4.4	Exemplos de Funções de Morse Discretas	53
4.5	O Campo de Vetores Gradiente Discreto e o Fluxo Associado	54

4.6	Caminhos Gradientes	60
4.7	Considerações Finais	61
Capítulo 5	Reconstrução de Curvas	63
5.1	Considerações Iniciais	63
5.2	Algoritmo Bidimensional	63
5.3	Garantias Teóricas	68
5.4	Resultados	71
5.5	Considerações Finais	74
Capítulo 6	Reconstrução de Superfícies	75
6.1	Considerações Iniciais	75
6.2	Preliminares	75
6.3	Algoritmo	76
6.4	Resultados Computacionais	81
6.5	Garantias Teóricas	87
6.6	Considerações Finais	95
Capítulo 7	Conclusões e Trabalhos Futuros	97
7.1	Considerações Iniciais	97
7.2	Principais Resultados	97
7.3	Principais Conclusões	98
7.4	Trabalhos Futuros	99
7.5	Considerações Finais	100

Lista de Figuras

1.1	Malhas a) estruturada b) não estruturada	2
1.2	(Gopi e Krishnan, 2002) Processos de amostragem de uma superfície.	4
2.1	a) exemplo de uma 2-variedade b) exemplo de uma não variedade	9
2.2	Variedade suave de dimensão 2 imersa em $\mathbb{R}^3$	9
2.3	Estrela: a) de um vértice b) de uma aresta	11
2.4	Link: a) de um vértice b) de uma aresta	11
2.5	a) diagrama de Voronoi b) triangulação de Delaunay	12
2.6	eixo medial representado pela linha pontilhada	15
2.7	exemplo bidimensional de local feature size	15
2.8	Regiões próximas ao eixo medial: a) alta curvatura b) estrangulamento	15
2.9	Exemplo do Cálculo da Homologia	20
3.1	(Dey e Kumar, 1999) Reconstrução de curvas no espaço	24
3.2	(Cheng et al., 2003) Estimativa das normais de um conjunto de amostras	25
3.3	a) Nuvem de pontos. b) Triangulação de Delaunay. c) Diagrama de Voronoi. d) Crust.	27
3.4	Regiões proibidas: da esquerda para a direita $\beta = 1, \frac{3}{2}, 2$	27
3.5	(Bardinet et al., 1998) Reconstrução que utiliza superelipsóides	29
3.6	a) Ball Pivoting bidimensional com diferentes curvaturas e taxas de amostragem b) Ball Pivoting tridimensional	33
3.7	Processo de estimativa das curvaturas da superfície e da malha proposta por Huang e Menq (2002)	35
3.8	Dragão com 10000 pontos a) $\alpha = 35$ b) $\alpha = 44$	36
3.9	Coelho com 1000 pontos a) $\alpha = 30$ b) $\alpha = 50$ c) $\alpha = 70$	36
3.10	Exemplos de λ -bolas para uma aresta definida pelos vértices a e b	38
3.11	(Adamy et al., 2000) (a). guarda-chuva, (b) e (c) são contra-exemplos.	39
3.12	(Giesen e John, 2003) reconstrução do Buda via complexo de fluxo	40
3.13	(Amenta et al., 1998b) Célula de Voronoi tridimensional de uma amostra	44
3.14	Intersecção entre duas bolas polares	45
3.15	(Gois, 2004) Exemplos da aproximação de objetos por bolas polares	46
3.16	Ilustração de um Cocone de uma amostra p e sua célula de Voronoi	47
3.17	a) e b) Exemplos de superfícies water tight, c) e d) são contra-exemplos	48
4.1	Exemplo de Complexo Simplicial	56
5.1	Construção da função de Morse em um complexo simplicial formado por dois triângulos, cinco arestas e quatro vértices	66
5.2	Reconstrução de um pato. a) arestas na $Im(V)$; b) arestas críticas; c) Curva final.	67
5.3	Arestas incidentes a uma amostra	68
5.4	Aresta na $Im(V)$	70

5.5	Proposição 2	70
5.6	Exemplo de reconstrução a) conjunto de pontos b) curva reconstruída	71
5.7	Reconstrução de um pato a) Tor b) Crust; c) Power-Crust; d) Beta-skeleton e) NN-Crust	72
5.8	Reconstrução do mapa do Brasil: a) Conjunto de amostras; b) Tor; c) Crust; d) Power-Crust; e) Beta-skeleton f) NN-Crust	73
5.9	Instabilidade numérica do Power-Crust	74
6.1	Tetraedro Sliver	80
6.2	Falha decorrente de tetraedros slivers.	80
6.3	Reconstrução de um cavalo. (a) apenas as faces pertencentes ao campo de vetores gradiente; (b) pontos críticos da função de Morse; (c) o modelo completo.	81
6.4	Modelo de um dragão reconstruído com o algoritmo Tor. (a) Textura; (b) Textura e malha; (c) Zoon.	82
6.5	Reconstrução do modelo de uma mão com aplicação de textura.	83
6.6	Reconstrução do Coelho a) Nosso algoritmo; b) Crust; c) Power Crust; d) Cocone; e) Tight Cocone	84
6.7	Modelo da Enterprise reconstruído por: (a) Crust; (b) Cocone; (c) Tight Cocone; (d) Tor.	85
6.8	Modelos com grandes quantidades de pontos. a) leão 636.646 pontos; b) Lucy 921.085 pontos	88
6.9	Vizinhança de uma amostra.	90
6.10	Demonstração do lema 17	94

Lista de Tabelas

3.1	(Gois, 2004) Informações Teóricas	49
3.2	(Gois, 2004) Custo Computacional	50
3.3	(Gois, 2004) Vantagens e Desvantagens	50
5.1	Tempo de execução dos algoritmos bidimensionais.	74
6.1	Tempo de processamento, em segundos, para a obtenção dos modelos mostrados.	83
6.2	Tempo de processamento, em segundos, para a obtenção dos modelos mostrados.	84
6.3	Pontos com adição artificial de ruídos.	85
6.4	Reconstrução dos modelos com ruídos retornada pelo algoritmo Tor (Topological Object Reconstructor).	86
6.5	Reconstrução dos modelos com ruídos retornada pelo algoritmo Crust.	86
6.6	Reconstrução dos modelos com ruídos retornada pelo algoritmo Power Crust.	86
6.7	Reconstrução dos modelos com ruídos retornada pelo algoritmo Cocone.	87
6.8	Reconstrução dos modelos com ruídos retornada pelo algoritmo Tight Cocone.	87

Resumo.

Reconstrução a partir de pontos não organizados é um problema relevante e comum a vários tipos de aplicações, como tratamento de imagens médicas e computação gráfica. Este tipo de abordagem é relativamente recente e busca gerar malhas não estruturadas a partir de conjuntos de pontos geralmente fornecidos por scanners tridimensionais, sondas sísmicas, radares e amostragens de superfícies implícitas. O principal desafio a ser superado neste tipo de reconstrução é a falta de informação, tanto geométrica quanto topológica, a respeito do objeto a ser reconstruído. Essa falta de informação tem impossibilitado a elaboração de algoritmos robustos e eficientes.

Diversas técnicas para a resolução deste problema são descritas na literatura, tais como técnicas baseadas em zeros de funções, em “esculpimento”, em modelos deformáveis e em métodos incrementais. Cada uma delas possui vantagens e desvantagens. No entanto, a maioria utiliza operações geométricas caras e pouco estáveis.

Este projeto de doutorado propõe uma nova abordagem para o problema de reconstrução a partir de pontos não organizados. A abordagem é baseada em “esculpimento”, e sua principal característica é diminuir a quantidade de operações geométricas, substituindo-as por operações topológicas, mais robustas e mais eficientes. Para isso, a teoria de Morse em complexos simpliciais é utilizada como ferramenta de decisão da inclusão ou não de simplexes na malha do objeto reconstruído. Além disso, apresentam-se garantias teóricas de que, sob uma taxa de amostragem adequada, a reconstrução é homeomorfa e próxima do objeto original.

Adicionalmente, esse projeto vem atender as necessidades do grupo de mecânica de fluidos computacional do ICMC-USP, fornecendo um gerador de malhas não estruturadas a partir de pontos não organizados, o qual deverá ser empregado onde técnicas mais convencionais, tais como técnicas de triangulação de polítopos e de reconstrução por seções planares, não produzem resultados satisfatórios.

Abstract.

Performing extensive geometrical computations is a major source of problems in reconstruction from sample points. Such operations are computationally intensive, and introduce numerical instabilities that may be difficult to handle. We introduce an innovative approach for generating a piecewise linear approximations from sample points that reduces geometrical calculations drastically by converting geometrical information into topological attributes. Geometrical tests are employed only at an initial step, to compute a Delaunay triangulation of the points. Discrete Morse theory is used to derive a topological framework towards reconstruction. The result is a robust reconstruction algorithm that handles multiple components and has reduced computational costs. In this work we introduce an innovative approach for generating 3D piecewise linear approximations from sample points that is strongly based on topological information, thus reducing the computational cost and numerical instabilities typically associated with geometric computations. Discrete Morse theory provides the basis of a topological framework for reconstruction that results in a robust reconstruction algorithm capable of handling multiple components and with low computational cost. We describe the proposed approach and introduce the reconstruction algorithm, called Tor - Topological Object Reconstructor. Reconstruction results are presented and the performance of Tor is compared with that of other reconstruction approaches for some standard point sets. Theoretical guarantees of reconstruction are also provided.

Capítulo 1

Introdução

1.1 Considerações Iniciais

Ao longo das últimas décadas, a computação gráfica tem se expandido, tornando-se uma poderosa ferramenta de apoio à solução de inúmeros problemas. Muitos desses problemas são resolvidos por meio de modelagem geométrica, que é a área da computação gráfica responsável pela criação e representação de modelos que descrevem a forma e outras características de um objeto.

Dentre as ferramentas utilizadas em modelagem geométrica, a topologia assume posição de destaque. Embora informações geométricas sejam relevantes, algoritmos fundamentados em conceitos topológicos são em geral mais robustos e eficientes, o que os torna mais atrativos. Desta forma, o desenvolvimento de novas técnicas de modelagem geométrica embasadas em conceitos topológicos torna-se muito relevante, sendo esse um dos aspectos importantes do trabalho aqui proposto.

1.2 Motivação

Diversos fenômenos em física e em engenharia, tais como condução de calor, escoamento de fluidos, propagação de ondas eletromagnéticas, podem ser modelados por equações diferenciais parciais. A maioria dessas equações não possui solução analítica e, por isso, métodos de aproximação numérica se fazem necessários.

Métodos numéricos para a solução de tais equações incluem diferenças finitas, elementos finitos e volumes finitos. Eles aproximam a solução das equações diferenciais por meio de um sistema que contém um número finito de equações algébricas. Esse processo requer uma discretização no domínio do problema, ou seja, decomposição do domínio em um número finito de elementos

(geralmente triângulos ou quadriláteros em dimensão dois e tetraedros ou hexaedros em dimensão três). Essa decomposição é chamada de **malha**.

Malhas podem ser divididas em duas categorias, **estruturadas** e **não estruturadas**. Informalmente, malhas estruturadas são aquelas onde os elementos podem ser indexados de forma matricial, obtendo-se as relações de vizinhança a partir de elementos vizinhos na matriz. Já em malhas não estruturadas essas relações não são óbvias e necessitam de espaço extra de memória para o seu armazenamento. A figura 1.1 mostra a diferença entre os dois tipos de malhas.

Figura 1.1: Malhas a) estruturada b) não estruturada

Este projeto de doutorado está relacionado com a geração de malhas não estruturadas para domínios com geometria complexa. A elaboração de um gerador de malha baseado em conceitos topológicos representa uma inovação e um desafio bastante interessante, além de atender as necessidades do grupo de mecânica de fluidos computacional existente no ICMC-USP, o qual pretende utilizar técnicas de volumes finitos e de elementos finitos em malhas não estruturadas para simulação numérica de escoamento de fluidos. Um bom levantamento bibliográfico sobre geração de malhas não estruturadas pode ser encontrado no trabalho de Teng e Wong (2000).

A representação do domínio onde se dá a decomposição espacial influencia, de forma direta, a escolha da estratégia de geração de malha.

Existem basicamente três formas de representação de domínio:

1. A fronteira da região de interesse é dada por um politopo;
2. Seções planares da região de interesse;
3. Pontos amostrados na região de interesse;

As estratégias de geração de malhas mais conhecidas atuam no primeiro tipo de representação descrito acima. Entre essas estratégias estão “advancing-front” (Jin e Tanner, 1993), inserção

de pontos e reconexão local (Marcum e Weatherill, 1995), “bubble meshing” (Dibben, 2000) e poligonalização de sólidos definidos implicitamente (Moreira, 1998).

As técnicas que atuam no segundo tipo de representação visam reconstruir objetos amostrados a partir de tomógrafos computadorizados, microscópios a laser e ressonância magnética. Reconstrução volumétrica via triangulação de Delaunay (Boissonnat, 1988; Nonato et al., 2001; Bajaj et al., 1999), geração de malha através de “prismatoids” (Bajaj et al., 1995), e técnicas de reconstrução da superfície limitante, (Jones e Chen, 1994), juntamente com triangulação de politopos (Ruppert e Seidel, 1989) são exemplos de estratégias voltadas para esse tipo de aplicação.

Métodos de geração de malhas a partir de pontos não organizados, também chamadas de técnicas de reconstrução a partir de nuvem de pontos, são mais recentes e buscam gerar uma malha não estruturada a partir de um conjunto de pontos, geralmente fornecido por scanners 3D, radares, sondas sísmicas, ou pontos amostrados em superfícies implícitas (ver trabalhos de (Amenta et al., 1998b; Dey e Goswami, 2002a; Giesen e John, 2003; Edelsbrunner, 2002)).

O principal objetivo deste projeto de doutorado é o desenvolvimento de uma técnica de geração de malhas não estruturadas a partir de conjuntos de pontos não organizados. Essa técnica visa gerar malhas que representem uma superfície onde a representação foi obtida a partir de pontos amostrados de maneira não organizada. Regiões rochosas amostradas por meio de sensores sísmicos e o trabalho desenvolvido pelo “Digital Michelangelo Project” (Levoy et al., 2000) são exemplos típicos para esse tipo de aplicação.

Basicamente, existem dois processos envolvidos em reconstrução a partir de pontos não organizados. O primeiro deles é o processo de amostragem dos pontos e o segundo é o processo de reconstrução propriamente dito.

O processo de amostragem envolve coletar pontos (amostras) sobre a curva ou superfície de interesse, enquanto que os processos de reconstrução objetivam restaurar o objeto original a partir das amostras. A figura 1.2 ilustra o processo de obtenção de amostras por meio de um scanner tridimensional.

1.3 Contribuições

Destacam-se aqui as principais contribuições desta tese de doutorado.

Uma grande fonte de problemas na área de reconstrução a partir de nuvem de pontos é a grande quantidade de cálculos geométricos envolvidos no processo, os quais introduzem instabil-

Figura 1.2: (Gopi e Krishnan, 2002) Processos de amostragem de uma superfície.

idades numéricas difíceis de controlar, tornando os algoritmos excessivamente lentos. A maioria das técnicas encontradas na literatura depende fortemente de complexas informações geométricas que devem ser calculadas a partir das amostras. Pouco tem sido feito no sentido de superar essas dificuldades geométricas que, em grandes conjuntos de dados, podem tornar o algoritmo proibitivo.

Apresenta-se neste trabalho, uma nova abordagem para reconstrução a partir de nuvem de pontos que reduz drasticamente a quantidade de operações geométricas envolvidas no processo. De fato, testes geométricos pesados são realizados apenas no passo inicial, para calcular a triangulação de Delaunay do conjunto de amostras. Teoria de Morse discreta (Forman, 1998a,b) é a base teórica para esta abordagem. O algoritmo aqui desenvolvido decide quais simplexes adicionar à reconstrução por meio de análise de um campo gradiente discreto derivado de uma função de Morse definida sobre a triangulação de Delaunay do conjunto de amostras. Como consequência direta desta abordagem topológica, consegue-se um algoritmo robusto capaz de manipular múltiplas componentes conexas e com um custo computacional bastante reduzido.

Além de um algoritmo de reconstrução baseado em topologia, apresentam-se também, garantias teóricas de que, sob uma adequada taxa de amostragem, a reconstrução fornecida pelo nosso algoritmo é homeomorfa e próxima ao objeto original.

1.4 Organização da Tese

Esta tese está organizada da seguinte forma:

O capítulo 2 apresenta os conceitos matemáticos fundamentais utilizados no restante do trabalho, tais como Topologia, Geometria computacional, Álgebra e Homologia simplicial. O capítulo 3 descreve os principais trabalhos sobre reconstrução a partir de pontos não organizados encontrados na literatura. Além disso, procurou-se destacar as vantagens e desvantagens de cada algoritmo

de modo a colocar este trabalho dentro de um contexto mais específico. O capítulo 4 apresenta os principais conceitos sobre teoria de Morse discreta, que é a chave para o desenvolvimento dos algoritmos. O capítulo 5 apresenta a versão bidimensional do algoritmo desenvolvido nesta tese, descrevendo ainda os resultados computacionais e as garantias teóricas de que, sob uma taxa de amostragem adequada, o algoritmo retorna uma reconstrução correta do objeto. O capítulo 6 descreve detalhadamente a versão tridimensional do algoritmo, apresentando resultados computacionais, comparações com algoritmos clássicos e garantias teóricas de reconstrução correta. Finalmente, o capítulo 7 sumariza os resultados alcançados, comenta as principais conclusões e sugere alguns trabalhos futuros.

Capítulo 2

Conceitos Fundamentais

2.1 Considerações Iniciais

Neste capítulo serão apresentadas definições sobre topologia, geometria computacional e álgebra, assim como alguns resultados clássicos dessas áreas que servirão como base para o restante do trabalho.

O capítulo começa apresentando conceitos sobre espaços topológicos e espaços topologicamente equivalentes. Descreve, em seguida, alguns conceitos sobre geometria computacional e noções gerais sobre complexos simpliciais, com ênfase na triangulação de Delaunay e nas suas principais propriedades, além do seu dual, o diagrama de Voronoi. Apresenta noções gerais sobre estruturas algébricas que incluem grupos, classes laterais, grupos quocientes, homomorfismos de grupos, anéis e R-módulos. Finalmente, descreve os principais conceitos sobre homologia, que serão utilizados no restante do trabalho.

2.2 Conceitos de Topologia

Uma maneira natural de investigar o espaço no qual se está trabalhando é obter informações sobre as relações de vizinhanças dos elementos deste espaço. Esta é a arte da Topologia, que procura deduzir a partir de observações locais (as relações de vizinhança), propriedades globais. Uma referência clássica sobre topologia pode ser encontrada no trabalho de Lima (1976)

Definição 1 *Um **espaço topológico** é um conjunto X de pontos com uma coleção C de conjuntos chamados abertos com as seguintes propriedades*

1. *A união arbitrária de abertos é um aberto;*

2. A intersecção de um número finito de abertos é um aberto;

3. O conjunto vazio e X são abertos;

São os abertos que definem as relações de vizinhanças. Uma *vizinhança* de um ponto é qualquer aberto que o contém. Um exemplo de espaços topológicos são os espaços métricos (qualquer espaço onde se possam medir distâncias) e, neste caso, os abertos são as *bolas abertas*, isto é, $B_{c,r} = \{x \in X : d(x, c) < r\}$, suas uniões e intersecções, onde d é a distância definida nestes espaços.

Seja X um espaço topológico e $B \subset X$ um conjunto qualquer, $x \in B$ é dito *ponto interior* se existe um aberto U tal que $x \in U \subset B$. Um conjunto é chamado de *conjunto aberto* se todos os seus pontos são pontos interiores. O *interior* de um conjunto B é o conjunto de todos os pontos de B que são interiores, sendo denotado por $\overset{\circ}{B}$. A *fronteira* de um conjunto B é o conjunto de pontos $x \in B$ tal que para todo aberto U contendo x , U possui pontos de B e do complementar de B . O *fecho* de um conjunto B é definido como o interior de B mais a fronteira de B , e é denotado como $\bar{B}$.

Definição 2 Sejam X e Y dois espaços topológicos e seja $f : X \rightarrow Y$ uma aplicação de X em Y . Diz-se que f é um **homeomorfismo** se f é contínua, bijetora e sua inversa f^{-1} também é contínua.

Dois espaços topológicos são topologicamente equivalentes, isto é, **homeomorfos**, se existe um homeomorfismo entre eles.

Definição 3 Diz-se que $f : X \rightarrow Y$ é um difeomorfismo de classe C^k se f e f^{-1} são diferenciáveis de classe C^k . Neste caso X e Y são ditos difeomorfos.

Definição 4 Uma variedade (manifold) de dimensão k contida em $\mathbb{R}^n$ ($n \geq k$) é um conjunto M de pontos (ver figura 2.1), onde, para qualquer ponto $p \in M$, existe uma vizinhança de p em M que é equivalente topologicamente a um aberto de $\mathbb{R}^{k-1} \times \mathbb{R}^+$. Se $k = 2$, tem-se uma variedade bidimensional, também chamada de superfície.

Se existir pelo menos um ponto do conjunto acima que não satisfaz essa condição o conjunto é dito ser uma não-variedade (non manifold), e o ponto que não satisfaz a condição é chamado de ponto singular.

Figura 2.1: a) exemplo de uma 2-variedade b) exemplo de uma não variedade

Definição 5 O conjunto $M \subset \mathbb{R}^n$ é uma variedade *suave* de dimensão k e classe C^r (ver figura 2.2) se para todo $x \in M$ existem uma vizinhança aberta $U \ni x$, V um aberto de $\mathbb{R}^k$ e um difeomorfismo de classe C^r $\varphi : U \cap M \rightarrow V$ (ver figura 2.2).

Figura 2.2: Variedade suave de dimensão 2 imersa em $\mathbb{R}^3$

2.3 Conceitos de Geometria Computacional

Um **simplexo p-dimensional** ou **p-simplexo** em $\mathbb{R}^n$ é o fecho convexo de $p + 1$ pontos geometricamente independentes, e um conjunto de pontos $\{a_0, a_1, \dots, a_p\} \subset \mathbb{R}^n$ é dito geometricamente independente se o conjunto de vetores $\{(a_1 - a_0), \dots, (a_p - a_0)\}$ é linearmente independente. O fecho convexo de um conjunto de pontos é o menor conjunto convexo que contém tais pontos. Um 0-simplexo é chamado de **vértice**; um 1-simplexo é chamado de **aresta**; um 2-simplexo de **triângulo**; e um 3-simplexo, de **tetraedro**. Considerando-se $\sigma = [a_0, a_1, \dots, a_p]$ um p -simplexo, qualquer k -simplexo $0 \leq k \leq p$ formado por um subconjunto de $\{a_0, a_1, \dots, a_p\}$ é chamado de **face** de σ . As faces diferentes de σ são chamadas de **faces próprias**, e a união das faces próprias de σ

é chamada de **bordo** de σ e denotada por $\partial\sigma$. O interior de um simplexo é definido como sendo $\sigma - \partial\sigma$ e denotado por $\overset{\circ}{\sigma}$.

Definição 6 Um *complexo simplicial* $\mathcal{K}$ em $\mathbb{R}^n$ é uma coleção de p -simplexos tal que:

1. Toda face de um simplexo de $\mathcal{K}$ está em $\mathcal{K}$;
2. A intersecção de quaisquer dois simplexos de $\mathcal{K}$ é uma face comum a cada um deles ou é vazia;
3. Qualquer conjunto compacto¹ de $\mathbb{R}^n$ intersecta um número finito de simplexos;

A dimensão de um complexo simplicial $\mathcal{K}$, denotada por $\dim \mathcal{K}$, é o máximo das dimensões dos simplexos de $\mathcal{K}$. Se L é uma subcoleção de $\mathcal{K}$ tal que L é um complexo simplicial, então L é chamado **subcomplexo** de $\mathcal{K}$. A coleção de todos os simplexos de $\mathcal{K}$ com dimensão menor ou igual a n (n menor que dimensão de $\mathcal{K}$) é um subcomplexo chamando **n-esqueleto** de $\mathcal{K}$ e será denotado por $\mathcal{K}^{(n)}$. A união de todos os simplexos de $\mathcal{K}$ é chamada de **politopo** de $\mathcal{K}$ (ou espaço gerado por $\mathcal{K}$) e denotada por $|\mathcal{K}|$. $X \subset \mathbb{R}^n$ é dito ser um **poliedro** se $X = |\mathcal{K}|$ para algum complexo simplicial finito $\mathcal{K}$. O grau de um vértice $v \in \mathcal{K}$ é definido como o número de arestas que tem v como vértice.

Considere-se $\mathcal{K}$ é um complexo simplicial, V o conjunto dos vértices de $\mathcal{K}$. Seja $\mathbf{k}$ a coleção de todos os subconjuntos $\{a_0, \dots, a_n\}$ de V , tal que os vértices $a_0, \dots, a_n$ geram um simplexo de $\mathcal{K}$. Tal coleção é chamada de **esquema de vértices** de $\mathcal{K}$.

Seja σ um p -simplexo. Duas orientações de seus vértices são ditas equivalentes se elas diferem por uma permutação par. Uma permutação é denominada **par** se necessita de um número par de trocas para transformá-la na permutação identidade e é **ímpar** se necessita de um número ímpar de trocas para transformá-la na identidade. Se $p > 0$, então existem duas classes de equivalência; caso contrário, apenas uma. Cada uma dessas classes é chamada de uma orientação de σ . Um simplexo orientado é um simplexo σ juntamente com uma orientação.

Considerando-se o simplexo formado pelos vértices v_0, v_1, v_2 e v_3 , a orientação $[v_0, v_1, v_2, v_3]$ será escolhida como sendo a orientação positiva, e, ainda, $-\sigma$ fica definido como sendo σ com sua orientação oposta.

Para um complexo simplicial $\mathcal{K}$ de dimensão n , um $(n-1)$ -simplexo é um **simplexo do interior** quando compartilhado por dois n -simplexos; caso contrário, é um **simplexo de bordo**. Os

¹Limitado e Fechado

simplexos de dimensão menor que $n - 1$ contidos em simplexos de bordo são também chamados de **simplexos de bordo**. O **bordo** de $\mathcal{K}$ é um subcomplexo $S \subset \mathcal{K}$ formado por todos os p -simplexos de bordo, $p = 0, 1, \dots, n - 1$ de $\mathcal{K}$, e cada componente conexa de $|S|$ é chamada de **componente de bordo** de $\mathcal{K}$.

A **estrela** de um simplexo(figura 2.3) $\sigma \in \mathcal{K}$ é definida como a união de todos os simplexos de $\mathcal{K}$ que tem σ como face, e é denotada por $st(\sigma)$. O **link** ou **enlace** de σ , denotado por $lk(\sigma)$, (figura 2.4) é definido como a união de todos os simplexos em $st(\sigma)$ que são disjuntos de σ .

Figura 2.3: Estrela: a) de um vértice b) de uma aresta

Figura 2.4: Link: a) de um vértice b) de uma aresta

2.3.1 Triangulação de Delaunay e Diagrama de Voronoi

Uma outra ferramenta importante para este trabalho é a triangulação de Delaunay e seu dual, o diagrama de Voronoi. Algumas das suas principais características serão mostradas a seguir.

Um conjunto finito, não vazio $S = \{x_1, \dots, x_n\}$ em $\mathbb{R}^m$ está em posição geral se nenhum sub-espaço afim de $\mathbb{R}^m$ contém S , e se não há uma esfera S^{m-1} passando através de um subconjunto de S com $m + k, k > 1$ pontos. No texto a seguir, apenas conjuntos de pontos em posição geral serão considerados.

Definição 7 O *diagrama de Voronoi* V de S é uma decomposição de $\mathbb{R}^m$ em células m -dimensionais

$\{V_1, \dots, V_n\}$ com as seguintes propriedades: (figura 2.5 a):

1. Cada V_i contém apenas um ponto $x_i \in S$;
2. Dado $x \in \mathbb{R}^m$, $x \in V_i$ se, e somente se, $d(x, x_i) \leq d(x, x_j)$ para todo $j \neq i$, onde d é a distância euclidiana;

Cada V_i será chamado **poliedro de Voronoi** associado a x_i . Note-se que $\mathbb{R}^m = \cup V_i$ e que cada poliedro de Voronoi é definido pela intersecção de $n-1$ semi-espacos fechados. Como semi-espacos são convexos e a intersecção de conjuntos convexos é convexa, tem-se que cada poliedro de Voronoi é convexo.

Dessas propriedades conclui-se que a intersecção de k poliedros do diagrama de Voronoi, onde $2 \leq k \leq m+1$, ou é vazia ou é uma célula $(m-k+1)$ -dimensional contida no diagrama.

Definição 8 A *triangulação de Delaunay* DT é o dual geométrico do diagrama de Voronoi (VD), isto é, para cada simplexo p -dimensional do diagrama pode-se associar, de forma natural, um $(m-p)$ -simplexo da triangulação de Delaunay (Figura 2.5 b).

Por exemplo, no caso bidimensional:

- Um vértice do diagrama de Voronoi associa-se a um triângulo (2-simplexo) de DT ;
- Uma aresta do diagrama de Voronoi associa-se a uma aresta (1-simplexo) de DT ;
- Um simplexo de dimensão 2 do diagrama de Voronoi associa-se a um vértice (0-simplexo) de DT ;

Figura 2.5: a) diagrama de Voronoi b) triangulação de Delaunay

Uma maneira alternativa de definir a triangulação de Delaunay é a seguinte. Dado um conjunto de pontos $S = \{x_1, \dots, x_n\} \subset \mathbb{R}^m$ em posição geral, uma triangulação é de Delaunay se a circunsfera

de todo m -simplexo é livre de pontos, ou seja, se não existe nenhum $x_i \in S$ no interior de tal esfera. Esta triangulação possui algumas propriedades importantes. Algumas delas são enumeradas abaixo.

Propriedade 1 *Para um conjunto de pontos em posição geral, a triangulação de Delaunay é única.*

Propriedade 2 *A triangulação de Delaunay gera o fecho convexo de S .*

Propriedade 3 (circulo vazio) *Todo d -simplexo $0 \leq d \leq m$ possui uma circunsfera vazia de pontos.*

Uma generalização da triangulação de Delaunay é a triangulação de Delaunay com pesos, também chamada de triangulação regular, que fica caracterizada pelas seguintes definições:

Definição 9 (Ponto com peso) *Diz-se que $\hat{p} = (p, w^2) \in \mathbb{R}^m \times \mathbb{R}_+$ é um ponto com peso, onde w^2 é o peso associado ao ponto $\hat{p}$ e p é a sua posição em $\mathbb{R}^m$.*

Definição 10 (distância com peso) *A distância com peso entre dois pontos com peso $\hat{p} = (p, w_1^2)$ e $\hat{q} = (q, w_2^2)$ é dada por $\|\hat{p} - \hat{q}\|_w = \|p - q\| - w_1^2 - w_2^2$.*

Diz-se que dois pontos com peso $\hat{p}$ e $\hat{q}$ são ortogonais se $\|\hat{p} - \hat{q}\|_w = 0$. Além disso, Um ponto com peso $\hat{p} = (p, w^2)$ pode ser interpretado geometricamente como uma esfera de centro em p e raio w . A definição de distância com peso pode ser vista em termos de uma esfera ao invés de um ponto com peso (Edelsbrunner, 2001).

Definição 11 (Power Diagrama) *O power diagrama de um conjunto de pontos com peso é definido como o diagrama de Voronoi que utiliza a distância com peso, em vez da distância euclidiana.*

Definição 12 (Triangulação de Delaunay com pesos) *Define-se por triangulação regular Triangulation como o dual do Power Diagrama.*

Em seguida, alguns tipos de grafos, utilizados por algoritmos clássicos da literatura, serão definidos.

Definição 13 (Grafo) *Uma grafo $G(V, E)$ é um conjunto finito de vértices $V \neq \emptyset$ juntamente com um conjunto finito de arestas E que conectam os vértices de V .*

Definição 14 (Grafo do vizinho mais próximo) O grafo do vizinho mais próximo (*Nearest Neighbour Graph, NNG*) de um conjunto de vértices V é um grafo (V, E) , de tal forma que $E = \{e_k = (v_i, v_j), k = 1, \dots, n : v_j \text{ é o ponto mais próximo de } v_i\}$.

Definição 15 (Árvore geradora mínima euclidiana) Uma árvore geradora mínima euclidiana (*Euclidean Minimum Spanning Tree, EMST*) é uma árvore (V, E) , de tal forma que $E = \{e_k = (v_i, v_j), k = 1, \dots, n : \sum \|v_i - v_j\| \text{ é mínima}\}$.

Definição 16 (Grafo de Gabriel) O grafo de Gabriel de V é um grafo (V, E) de tal forma que $E = \{e_k = (v_i, v_j), k = 1, \dots, n : \text{a menor esfera tocando } v_i \text{ e } v_j \text{ não contém pontos de } V\}$.

Definição 17 (Hipergrafo de Gabriel estendido) O hipergrafo de Gabriel estendido (*Extended Gabriel Hypergraph, EGH*) é um conjunto de vértices, arestas e triângulos (V, E_{EGH}, T) onde o conjunto inicial de arestas é o mesmo do grafo de Gabriel, e o conjunto de triângulos é definido da seguinte forma:

- Para quaisquer $e_1, e_2 \in E_{EGH}, e_1 = (v_1, v_2), e_2 = (v_2, v_3)$. Se v_1, v_2 e v_3 forem não colineares e se a menor esfera tocando os três vértices for vazia de pontos de V , então $e_3 = (v_1, v_3) \in E_{EGH}$;
- Qualquer ciclo de três arestas em E_{EGH} é um triângulo em T ;

Não é difícil ver que o hipergrafo de Gabriel estendido é um subconjunto da triangulação de Delaunay em dimensão três.

As definições seguintes são úteis para determinar uma condição de amostragem sob a qual pode-se garantir a correta reconstrução de uma variedade (curva ou superfície). Tal conceito será empregado nas garantias teóricas do algoritmo descrito neste texto.

Definição 18 Seja M uma variedade suave. O **eixo medial** (ver figura 2.6) de M é conjunto de centros de esferas vazias tocando M em pelo menos dois pontos.

Para qualquer ponto $p \in M$ o *Local Feature Size* em p é definido como a distância de p até o eixo medial e é denotado por $LFS(p)$ (ver figura 2.7). Um conjunto P de amostras é dito ser uma **r-amostragem** se a distância de qualquer ponto $a \in M$ até a amostra mais próxima $p \in P$ é, no máximo, uma constante r vezes o $LFS(a)$. Neste caso, diz-se que a variedade M está r -amostrada.

Figura 2.6: eixo medial representado pela linha pontilhada

Para valores apropriados de r esta definição garante que regiões mais próximas ao eixo medial sejam amostradas de forma mais densa, por exemplo, regiões de alta curvatura ou regiões com estrangulamento, como pode ser visto na figura 2.7.

Figura 2.7: exemplo bidimensional de local feature size

Figura 2.8: Regiões próximas ao eixo medial: a) alta curvatura b) estrangulamento

2.4 Conceitos de Álgebra

Os conceitos algébricos aqui apresentados podem ser encontrados no trabalho de Hungerford (1996).

Definição 19 Dado um conjunto $S \neq \emptyset$, qualquer função $\varphi : S \times S \rightarrow S$ é chamada de operação

binária

Exemplo 1

$$\begin{aligned}\varphi &: \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R} \\ (a, b) &\mapsto a + b\end{aligned}$$

Definição 20 Uma operação $*$ definida em um conjunto S é associativa se para quaisquer $a, b, c \in S$ vale $(a * b) * c = a * (b * c)$.

Definição 21 Um **semigrupo** é um par $(S, *)$ onde $S \neq \emptyset$ e $*$ é uma operação binária associativa.

Um semigrupo $(S, *)$ tem um elemento identidade se existe $e \in S$ tal que para qualquer $a \in S$ vale $a * e = e * a = a$.

Definição 22 Seja $(S, *)$ um sistema matemático com identidade e . Um elemento $a \in S$ tem **inverso** se existe $b \in S$, tal que $a * b = b * a = e$.

Neste caso, b é chamado de inverso de a e é denotado por a^{-1} .

Definição 23 Um par $(G, *)$ é um **grupo** se, e somente se, $(G, *)$ é um semigrupo com identidade tal que cada elemento de G tem inverso.

1. Além disso, $(G, *)$ é um **grupo abeliano**, quando vale a lei de comutatividade para sua operação, isto é, quando para quaisquer $a, b \in G$ vale $(a * b) = (b * a)$. Para $H \subseteq G$ e $H \neq \emptyset$ diz-se que $(H, *)$ é **subgrupo** de $(G, *)$ se ele próprio é grupo.

Definição 24 Sejam $(G, *)$ e $(G^*, \circ)$ dois grupos e $f : G \rightarrow G^*$. Se para quaisquer $a, b \in G$ vale $f(a * b) = f(a) \circ f(b)$, f é chamado de **homomorfismo**.

Um homomorfismo injetor é chamado de **monomorfismo**, um homomorfismo sobrejetor é chamado de **epimorfismo**; e ainda um homomorfismo bijetor é chamado de **isomorfismo**. Dois conjuntos G e G^* são ditos isomorfos se existe um isomorfismo entre eles, que se denota por $G \cong G^*$. Do ponto de vista algébrico, é importante identificar grupos isomorfos, pois eles tem as mesmas propriedades.

O **kernel** de um homomorfismo $f : G \rightarrow G^*$ é definido como o conjunto $\{g \in G \text{ tal que } f(g) = 0\}$ e denotado por $\ker(f)$. A imagem de f é definida como o conjunto $\{b \in G^* \text{ tal que existe } g \in G \text{ com } f(g) = b\}$ e denotada por $\text{Im}(f)$.

Definição 25 Seja $(H, *)$ um subgrupo de $(G, *)$ e seja $a \in G$. O conjunto $\{a * h \text{ com } h \in H\}$ é chamado de **classe lateral à esquerda** de H e denotado por $a * H$ e o conjunto $\{h * a \text{ com } h \in H\}$ é chamado de **classe lateral à direita** de H sendo denotado por $H * a$.

Definição 26 $(H, *)$ é um **subgrupo normal** de $(G, *)$ se, para qualquer $a \in G$, vale $a * H = H * a$.

Definição 27 Seja H um subgrupo normal de G . defini-se o **grupo quociente** $\frac{G}{H} = \{a * H, a \in G\}$ onde a seguinte operação é definida:

$$\begin{aligned} & + \\ \otimes & : \frac{G}{H} \times \frac{G}{H} \rightarrow \frac{G}{H} \\ (a * H, b * H) & \rightarrow (a * H) \otimes (b * H) = (a * b) * H \end{aligned}$$

Teorema 1 (Hungerford, 1996) Seja $f : G \rightarrow G^*$ um epimorfismo, então $\frac{G}{\ker(f)} \cong G^*$.

Este teorema é conhecido como teorema fundamental de grupos e fornece uma indicação de como encontrar isomorfismos entre grupos.

Definição 28 Um **anel** é um sistema matemático $(A, *, \circ)$ constituído por um conjunto não vazio A e duas operações definidas em A satisfazendo

1. $(A, *)$ é um grupo abeliano;
2. $(A, \circ)$ é um semigrupo;
3. para quaisquer $a, b, c \in A$ valem

$$(a) \quad a \circ (b * c) = (a \circ b) * (a \circ c)$$

$$(b) \quad (b * c) \circ a = (b \circ a) * (c \circ a)$$

Exemplo 2 O conjunto dos números inteiros com as operações de soma e de multiplicação $(\mathbb{Z}, +, *)$ é um anel.

Se na definição 28 for acrescentada a condição de que para quaisquer $a, b \in A$ vale $a \circ b = b \circ a$, A é chamado de anel comutativo, e, ainda, se existir $1 \in A$ tal que $a \circ 1 = 1 \circ a = a$ para qualquer $a \in A$, então A é chamado de anel com identidade. $(\mathbb{Z}, +, *)$ também é um exemplo de anel comutativo com identidade.

Definição 29 *Seja R um anel comutativo com identidade. Um R -Módulo é um grupo abeliano X junto com uma função $\mu : R \times X \rightarrow X$ tal que para quaisquer $\alpha, \beta \in R$ e $x, y \in X$, vale*

$$1. \mu(\alpha + \beta, x) = \mu(\alpha, x) + \mu(\beta, x);$$

$$2. \mu(\alpha, x + y) = \mu(\alpha, x) + \mu(\alpha, y);$$

$$3. \mu(\alpha, \mu(\beta, x)) = \mu(\alpha\beta, x);$$

$$4. \mu(1, x) = x;$$

μ neste contexto é chamada de multiplicação por escalar.

Exemplo 3 *Qualquer grupo abeliano X é um $\mathbb{Z}$ -módulo com a operação*

$$\begin{aligned} \mu &: \mathbb{Z} \times X \rightarrow X \\ \mu(m, x) &= mx \end{aligned}$$

2.5 Conceitos de Homologia Simplicial

A teoria de homologia é uma maneira eficiente de descrever, unindo conceitos de álgebra e de topologia, algumas propriedades ligadas à conectividade de complexos simpliciais, como o número de componentes conexas, de buracos e de cavidades. Seus principais invariantes são os **números de Betti**, que estão diretamente ligados a essas propriedades.

Considere-se um complexo simplicial finito $\mathcal{K}$ e o conjunto de todos os simplexes de dimensão p , denotado por $\mathcal{K}_p$.

O $\mathbb{Z}$ -módulo livre das p -cadeias do complexo $\mathcal{K}$ é o $\mathbb{Z}$ -módulo livre gerado por $\mathcal{K}_p$, denotado por $C_p(\mathcal{K})$. Desde que $\mathcal{K}$ é finito, a quantidade de elementos de $\mathcal{K}_p$ também é, diga-se $b(p)$.

Observe-se que $C_p(\mathcal{K})$ é isomorfo ao $\mathbb{Z}$ -módulo livre gerado pela soma direta de $b(p)$ cópias de $\mathbb{Z}$, isto é,

$$C_p(\mathcal{K}) \simeq \mathbb{Z}^{b(p)} \simeq \mathbb{Z} \oplus \mathbb{Z} \oplus \dots \oplus \mathbb{Z}$$

Para exibir este isomorfismo, é suficiente exibir uma bijeção entre $\mathcal{K}_p$, base de $C_p(\mathcal{K})$, e a base canônica de $\mathbb{Z}^{b(p)}$. Para isso, basta ordenar os elementos de $\mathcal{K}_p$ e da base canônica de $\mathbb{Z}^{b(p)}$, por exemplo, usando-se a ordem lexicográfica e identificar o primeiro elemento de $\mathcal{K}_p$ com o primeiro elemento da base canônica de $\mathbb{Z}^{b(p)}$ e, assim, sucessivamente. E, ainda, se $\mathcal{K}$ não tiver simplexes de dimensão p , define-se $C_p(\mathcal{K}) = 0$.

2.5.1 Operador Bordo

Cada p -simplexo é formado por $p + 1$ simplexes de dimensão $p - 1$ e estas faces são obtidas ao se omitir um vértice no p -simplexo considerado. Assim para cada número j entre 1 e $p + 1$ tem-se um modo de associar a cada p -simplexo um $(p - 1)$ -simplexo simplesmente omitindo o j -ésimo vértice.

Esse fato permite construir um $\mathbb{Z}$ -homomorfismo de $C_p(\mathcal{K})$ em $C_{p-1}(\mathcal{K})$, chamado operador da j -ésima face de um p -simplexo, que é denotado por ∂_p^j , e seu valor em um elemento da base é dado por :

$$\partial_p^j[v_0, v_1, \dots, v_p] = [v_0, v_1, \dots, v_{j-1}, v_{j+1}, \dots, v_p] = [v_0, v_1, \dots, \hat{v}_j, \dots, v_p]$$

Onde $\hat{v}_j$ significa a omissão do vértice v_j .

Definição 30 O operador bordo é o $\mathbb{Z}$ -homomorfismo ∂_p de $C_p(\mathcal{K})$ em $C_{p-1}(\mathcal{K})$ dado pela soma alternada de seus $p + 1$ operadores faces, isto é:

$$\partial_p[v_0, v_1, \dots, v_p] = \sum_{j=0}^p (-1)^j [v_0, v_1, \dots, \hat{v}_j, \dots, v_p]$$

onde $-[v_0, v_1, \dots, v_p]$ representa o simplexo $[v_0, v_1, \dots, v_p]$ com orientação invertida.

Propriedade 4 (Massey, 1991) Para $p > 1$, $\partial_{p-1} \circ \partial_p = 0$.

2.5.2 Grupos de Homologia

O grupo dos **p-ciclos** de $\mathcal{K}$ é definido como $Z_p(\mathcal{K}) = \text{Ker}(\partial_p) \subset C_p(\mathcal{K})$, e o grupo dos **p-bordos** de $\mathcal{K}$ como $B_{p-1}(\mathcal{K}) = \text{Im}(\partial_p) \subset C_{p-1}(\mathcal{K})$. Segue-se da propriedade 4 que $B_p(\mathcal{K}) \subset Z_p(\mathcal{K})$.

Finalmente, o p -ésimo grupo de homologia de $\mathcal{K}$ é definido como $H_p(\mathcal{K}) = \frac{Z_p(\mathcal{K})}{B_p(\mathcal{K})}$.

O **rank** ou **posto** de $H_p(\mathcal{K})$ (número de elementos da base), denotado por $\beta_p(\mathcal{K})$, é chamado de p -ésimo **número de Betti**. Os números de Betti estão associados ao número de componentes conexas, número de buracos e número de cavidades em $\mathcal{K}$ para $p = 0, 1$ e 2 respectivamente. Se n_v, n_e, n_f e n_t representam o número de vértices, arestas, triângulos e tetraedros de $\mathcal{K}$ respectivamente, sua característica de Euler é definida como $\chi(\mathcal{K}) = n_v - n_e + n_f - n_t$. Um resultado encontrado no trabalho de Lima (1981) garante que a característica de Euler de $\mathcal{K}$ pode também ser calculada como $\chi(\mathcal{K}) = \beta_0(\mathcal{K}) - \beta_1(\mathcal{K}) + \beta_2(\mathcal{K})$.

Exemplo 4 Considere-se $\mathcal{K}$ o complexo simplicial representado pela figura 2.9

Figura 2.9: Exemplo do Cálculo da Homologia

Deve-se calcular $H_1(\mathcal{K})$; seja $C = m_1e_1 + m_2e_2 + m_3e_3 + m_4e_4$ uma 1-cadeia genérica. Exige-se que $\partial C = 0$ pois se deseja o $\text{Ker}(\partial_1)$. Observe-se que $\partial C = 0 \iff m_1 = m_2 = m_3 = m_4$ pois $\partial e_1 = (v_1 - v_4)$, $\partial e_2 = (v_2 - v_1)$, $\partial e_3 = (v_3 - v_2)$ e $\partial e_4 = (v_4 - v_3)$. Isso quer dizer que os 1-ciclos são elementos da forma $m(e_1 + e_2 + e_3 + e_4)$ com $m \in \mathbb{Z}$; portanto $Z_1(\mathcal{K}) = \mathbb{Z}$ e $B_1(\mathcal{K}) = 0$, uma vez que as arestas de $\mathcal{K}$ não pertencem ao bordo de ninguém. Pode-se concluir, então, que $H_1(\mathcal{K}) = \frac{Z_1(\mathcal{K})}{B_1(\mathcal{K})} = \mathbb{Z}$ tendo como gerador o elemento $(e_1 + e_2 + e_3 + e_4)$, o que é equivalente a dizer que $\mathcal{K}$ possui um buraco.

2.6 Considerações Finais

Neste capítulo, os principais conceitos necessários para o entendimento do restante deste trabalho foram apresentados. Definições e resultados ainda serão introduzidos quando for necessário fazê-lo.

O próximo capítulo apresentará um extenso levantamento bibliográfico, que tem por objetivo colocar este trabalho dentro do contexto adequado.

Capítulo 3

Levantamento Bibliográfico

3.1 Considerações Iniciais

Segundo Teng e Wong (2000), a área de geração de malha é um grande exemplo de pesquisa interdisciplinar, uma vez que ela é construída sobre avanços em geometria computacional e combinatória, estruturas de dados, análise numérica e aplicações científicas. Este sucesso pode ser atribuído não apenas a elegantes demonstrações matemáticas a respeito da qualidade da malha resultante, mas também ao bom desempenho computacional das aplicações práticas, incluindo bons algoritmos e heurísticas eficientes.

Este capítulo apresenta uma descrição geral de trabalhos envolvendo reconstrução a partir de nuvem de pontos encontrados na literatura. Um completo levantamento bibliográfico pode ser encontrado no trabalho de Gois (2004), o qual divide as técnicas em quatro grandes classes.

1. Técnicas baseadas em modelos deformáveis.
2. Técnicas baseadas em zeros de funções.
3. Técnicas incrementais.
4. Técnicas baseadas em esculpimento ou em decomposição espacial.

Métodos baseados em modelos deformáveis começam com uma superfície inicial e seguem aplicando deformações até obter uma boa aproximação para a nuvem de pontos. Estes métodos são rápidos e apresentam resultados satisfatórios se a superfície inicial não estiver muito distante da forma original.

Os métodos baseados em zeros de função consistem em utilizar o conjunto de amostras para definir uma função distância com sinal f , de tal forma que a superfície seja formada pelo conjunto de zeros da função f , que é denotado por $Z(f)$.

Os métodos incrementais começam com uma aresta inicial (ou um conjunto de arestas iniciais), que é constituída pela conexão de pontos da nuvem e é denominada fronteira de avanço, e em seguida, eles constroem a superfície de maneira iterativa. O principal desafio dessa abordagem é definir as arestas iniciais do processo de reconstrução. Os métodos incrementais também são conhecidos como métodos de avanço de fronteira.

O princípio básico da abordagem de esculpimento consiste em obter, no primeiro estágio, a triangulação de Delaunay do conjunto de pontos e, em seguida, por meio de heurísticas denominadas heurísticas de esculpimento, remover um conjunto de simplexes para obter uma forma que melhor represente o objeto original. O conjunto de faces que representa a superfície será um subconjunto da triangulação de Delaunay.

Além dessa divisão, apresentam-se as técnicas bidimensionais e tridimensionais separadamente, uma vez que desenvolveu-se dois algoritmos, um bidimensional e um tridimensional, com diferenças marcantes, principalmente no que se refere às garantias teóricas de reconstrução.

Ainda neste capítulo, uma atenção especial será dada aos métodos da família Crust, não apenas pelo fato de serem os primeiros a apresentarem resultados práticos satisfatórios e garantias teóricas a respeito da reconstrução, mas também pelo fato de que vários de seus resultados teóricos são utilizados neste trabalho.

3.2 Trabalhos Correlatos Bidimensionais

Em poucas palavras, algoritmos de reconstrução bidimensionais se propõem a encontrar, a partir de um conjunto P de amostras, um polígono (ou um conjunto de polígonos) que aproxima a curva original. Neste caso, a reconstrução correta deve conectar as amostras na mesma ordem em que elas estão posicionadas ao longo da curva. Em duas dimensões, existe uma grande quantidade de algoritmos, dos quais serão comentados alguns dos mais conhecidos.

Vale a pena salientar que alguns dos algoritmos bidimensionais aqui descritos possuem sua versão tridimensional, que também será descrita posteriormente.

Uma das primeiras tentativas de reconstruir curvas (e superfícies) a partir de uma nuvem de pontos foi o algoritmo α -shape de Edelsbrunner e Mücke (1994). Começando com a triangulação

de Delaunay do conjunto de amostras, o α -shape remove todos os simplexes que não estão contidos em uma esfera vazia de amostras de raio α . Embora o α -shape seja simples de implementar, ele só funciona bem para conjuntos uniformemente amostrados, uma vez que o mesmo valor de α é aplicado em todo o conjunto de pontos.

Bernardini e Bajaj propõem métodos práticos, para 1-variedades em $\mathbb{R}^2$, que selecionam automaticamente valores de α de forma que a reconstrução seja homeomorfa à curva original (Bernardini e Bajaj, 1997). No entanto, este trabalho não se estendeu para o caso tridimensional.

Da mesma forma que o α -shape, o algoritmo β -skeleton (Kirkpatrick e Radke, 1985) extrai os polígonos que se ajustam ao conjunto de pontos usando esferas vazias e a triangulação de Delaunay. Nesta abordagem, regiões proibidas definidas através de esferas vazias fornecem a aproximação poligonal da curva. O β -skeleton, que é definido apenas no caso bidimensional, é muito simples de implementar, mas apresenta as mesmas desvantagens do α -shape.

Fazendo uso de um tipo especial de árvore geradora mínima, Figueiredo e Miranda (1995) apresentaram um algoritmo com garantias teóricas de reconstrução para algumas classes de curvas. Este foi um dos primeiros trabalhos com fundamentação teórica a respeito da reconstrução, entretanto, o algoritmo só funciona para alguns tipos particulares de curvas e exige uma densa taxa de amostragem.

Inspirado no Crust, que será discutido mais adiante, Dey e Kumar (1999) apresentaram um algoritmo bastante simples, chamado de NN-Crust, descrito no algoritmo 1. Tal algoritmo é capaz de reconstruir curvas não apenas no plano, mas também em dimensões mais altas (ver figura 3.1). O NN-Crust também apresenta garantias teóricas a respeito da reconstrução e, segundo os autores, apresenta uma excelente performance computacional.

Algoritmo 1 NN-CRUST

(Para um conjunto P de amostras)

1. Calcule-se a triangulação de Delaunay de P .
 2. Calcule-se o conjunto de arestas N que conecta os vizinhos mais próximos.
 3. Para cada amostra p contida em apenas uma aresta e de N . Calcule-se a menor aresta incidente em p , dentre todas as arestas que formam um ângulo maior que $\frac{\pi}{2}$ com e . Seja D o conjunto de todas essas arestas.
 4. Retorne-se $N \cup D$
-

O NN-Crust é bastante semelhante ao nosso algoritmo bidimensional, porém, além de não ser desenvolvido via teoria de Morse discreta, ele executa um cálculo de ângulos, o qual foi evitado nesta abordagem e, além disso, não apresenta nenhuma extensão para a reconstrução de superfícies.

Figura 3.1: (Dey e Kumar, 1999) Reconstrução de curvas no espaço

Também baseado no Crust, Dey et al. (1999b) apresentam um algoritmo chamado de Conservative Crust, que reconstrói tanto curvas fechadas quanto arcos. Embora este algoritmo exija uma taxa de amostragem menos rígida, ele não assegura a reconstrução correta do objeto original. Outra desvantagem desse método é a grande quantidade de testes geométricos realizados, além da triangulação de Delaunay. O algoritmo faz um refinamento em um grafo de Gabriel, o qual utiliza muitas operações geométricas.

O primeiro trabalho a garantir a reconstrução correta para curvas não suaves é apresentado em Giesen (1999), que assegura a correta reconstrução utilizando o algoritmo do caixeiro viajante, o que torna essa abordagem computacionalmente muito cara. Althaus e Mehlhorn (2000) melhoraram o algoritmo de Giesen mostrando que, sob uma taxa de amostragem adequada, o problema do caixeiro viajante pode ser resolvido em tempo polinomial.

Cheng et al. (2003) propõem um algoritmo que reconstrói curvas suaves a partir de um conjunto de amostras com ruído. O modelo de ruído é obtido por meio de uma perturbação aleatória no vetor normal definido nas amostras. A novidade desta técnica é agrupar os pontos de forma que cada agrupamento represente uma porção da curva. A figura 3.2 ilustra este processo. No lado esquerdo, um conjunto de pontos com ruídos pode ser visto; ao centro tem-se um agrupamento ou cluster e à direita os pontos mais representativos juntamente com seus vetores normais. Os autores ainda argumentam que esta abordagem também pode ser útil para curvas menos restritivas, por exemplo, curvas não suaves. Este trabalho apresenta garantias teóricas a respeito da reconstrução,

entretanto nenhuma generalização para o caso tridimensional foi apresentada ainda.

Figura 3.2: (Cheng et al., 2003) Estimativa das normais de um conjunto de amostras

O Power Crust (Amenta et al., 2001b) é um melhoramento do algoritmo Crust. Esta abordagem emprega um tipo de diagrama de Voronoi com pesos, chamado Power diagrama, para calcular uma aproximação linear por partes de uma curva suave.

O caso tridimensional do Power Crust será descrito com mais detalhes porque apenas este apresenta garantias teóricas de gerar uma reconstrução correta. Entretanto, um problema com o Power Crust é a instabilidade numérica dos cálculos geométricos realizados no Power diagrama. Na prática, quando as amostras são perturbadas, diferentes resultados são obtidos.

Os algoritmos Crust (versão bidimensional) e β -skeleton serão discutidos com um pouco mais de detalhes, não apenas porque foram realizadas comparações entre esses métodos e o deste trabalho, mas também porque foram utilizados alguns dos seus resultados teóricos nas demonstrações aqui apresentadas.

Crust bidimensional e β -skeleton

O primeiro algoritmo a apresentar uma reconstrução satisfatória e com garantias teóricas foi o Crust de Amenta et al. (1998a). Sob uma taxa de amostragem adequada, os autores asseguram que é possível construir uma aproximação poligonal correta para uma curva fechada usando, além das amostras, os vértices de Voronoi calculados a partir destes pontos. O Crust funciona para conjuntos de pontos que não são uniformemente amostrados, reduzindo a intervenção do usuário durante o processo de reconstrução. A instabilidade dos cálculos geométricos necessários para obter os vértices de Voronoi é a principal desvantagem desta técnica. Além disso, o algoritmo deve construir duas triangulações de Delaunay; uma para obter os vértices de Voronoi e outra para gerar o Crust, aumentando o custo computacional. Outra contribuição apresentada neste trabalho é um

melhoramento no algoritmo β -skeleton, estabelecendo uma fundamentação teórica que assegura, para um valor específico de β , uma reconstrução correta.

Considere-se F uma curva fechada diferenciável no plano, podendo ter mais de uma componente conexa, e $P \subset F$ um conjunto finito de amostras (pontos) de F .

Os resultados teóricos apresentados neste artigo, são baseados nas seguintes definições:

Definição 31 *Uma reconstrução poligonal de F a partir de P é um grafo que conecta cada par de amostras adjacentes ao longo de F e nenhuma outra.*

Ambos os grafos definidos estão relacionados com o diagrama de Voronoi e com a triangulação de Delaunay do conjunto de amostras e utilizam a conhecida propriedade do círculo vazio (ver propriedade 3, capítulo 2).

Definição 32 *Seja P um conjunto finito de pontos no plano, considere-se a triangulação de Delaunay de P e seja V o conjunto dos vértices do diagrama de Voronoi obtido. Seja $P^1 = P \cup V$, e considere-se ainda, a triangulação de Delaunay de P^1 . Diz-se que uma aresta da triangulação de Delaunay de P^1 pertence ao **Crust** de P , se seus dois vértices pertencem a P .*

Esta definição fornece um algoritmo para se obter o Crust de um conjunto de amostras P dado (figura 3.3), bastando para isso seguir os seguintes passos.

Algoritmo 2 Crust

(Para um conjunto P de amostras.)

- 1 Construa-se a triangulação de Delaunay de P .
 - 2 Considere-se o conjunto V dos vértices do diagrama de Voronoi de P .
 - 3 Construa-se a triangulação de Delaunay de $P^1 = P \cup V$.
 - 4 Adicionem-se ao Crust as arestas da triangulação de P^1 cujos dois vértices pertencem a P .
-

A definição do Crust usa a idéia intuitiva de que os vértices do diagrama de Voronoi de P são uma boa aproximação para o eixo medial. Além disso, os autores mostram que o Crust de P é um subconjunto da triangulação de Delaunay de P .

O β -skeleton é definido a partir do conceito, que será apresentado a seguir, de região proibida de dois pontos no plano.

Definição 33 *Sejam p_1 e p_2 um par de pontos no plano e seja $\beta \in \mathbb{R}$ uma constante. A **região proibida** de p_1, p_2 é a união de dois discos de raio $\frac{\beta}{2}d(p_1, p_2)$ tocando p_1 e p_2 , onde $d(p_1, p_2)$ é a distância entre p_1 e p_2 (ver figura 3.4).*

Figura 3.3: a) Nuvem de pontos. b) Triangulação de Delaunay. c) Diagrama de Voronoi. d) Crust.

Figura 3.4: Regiões proibidas: da esquerda para a direita $\beta = 1, \frac{3}{2}, 2$.

Definição 34 Seja P um conjunto finito de pontos no plano e sejam $p_1, p_2 \in P$. A aresta que conecta p_1 à p_2 pertence ao β -skeleton, se a região proibida de p_1, p_2 é vazia de pontos de P .

Observe-se que, com um bom programa de triangulação de Delaunay, tanto o Crust quanto o β -skeleton são bem fáceis de calcular. Para obter o β -skeleton, basta calcular a triangulação de Delaunay de P e seleccionar as arestas cujos raios dos circuncírculos dos triângulos adjacentes sejam maiores ou iguais a $\frac{\beta}{2}$ vezes o comprimento da aresta.

Os seguintes teoremas garantem que, para uma determinada taxa de amostragem, tanto Crust quanto o β -skeleton são uma reconstrução poligonal para uma dada curva F .

Teorema 2 (Amenta et al., 1998a) Seja F uma curva suave r -amostrada no plano, $r < 1$. A triangulação de Delaunay do conjunto P de amostras contém uma aresta entre cada par de pontos adjacentes.

Teorema 3 (Amenta et al., 1998a) O Crust de uma curva suave r -amostrada F , para $r < 0.4$

contém uma aresta entre cada par de pontos adjacentes.

Teorema 4 (Amenta et al., 1998a) *O Crust de uma curva suave r -amostrada F , para $r < 0.252$ não contém nenhuma aresta entre pares de pontos não adjacentes.*

Teorema 5 (Amenta et al., 1998a) *Seja P uma r -amostragem de uma curva suave F . Se $r < 0.297$, então O β -skeleton é uma reconstrução poligonal de F para $\beta = 1.7$.*

3.3 Trabalhos correlatos tridimensionais

Sejam uma superfície $S \subset \mathbb{R}^3$ e um conjunto de pontos $P \subset S$, que é definido como nuvem de pontos de S . Uma reconstrução da superfície S é uma aproximação linear por partes N de S .

Os métodos de reconstrução buscam satisfazer os seguintes itens:

- N deve manter a mesma topologia de S ;
- Dado um ponto de N , sua distância à superfície S deve ser pequena;

Questões críticas em reconstrução de superfícies são manipular superfícies de topologia arbitrária, utilizar conjunto de pontos não-uniformes e produzir superfícies que garantam uma boa representação da forma original do objeto. Além disso, os algoritmos de reconstrução também buscam satisfazer requisitos como eficiência de tempo e espaço (estrutura de dados espaciais e topológicas eficientes).

3.3.1 Métodos baseados em modelos deformáveis

Nestes métodos de reconstrução, a partir de uma superfície inicial, aplicam-se deformações gerando uma aproximação para a superfície que originou a nuvem de pontos. Estes métodos têm uma boa performance computacional, entretanto não apresentam resultados satisfatórios quando a superfície inicial não está próxima da forma original.

Existem basicamente dois subconjuntos de métodos baseados em modelos deformáveis: o primeiro é baseado em Equações Diferenciais Parciais (E.D.P.'s) para deformação dos modelos iniciais (Zhao et al., 2000). No entanto, estes trabalhos necessitam de discretizações das E.D.P.'s e resolução de grandes sistemas lineares. Já o segundo subconjunto de métodos é baseado em funções paramétricas (Bardinet et al., 1998). Dada uma função paramétrica inicial, aplicam-se alterações nos parâmetros

até atingir um valor satisfatório que melhor enquadre a função à nuvem de pontos (ver figura 3.5). O problema com funções paramétricas é que elas são limitadas quando existe muita variação de topologia e de detalhes da superfície.

Bardinet et al. (1998) apresentam um trabalho baseado em superquádricas, cuja principal vantagem é o número reduzido de parâmetros necessários para representar o objeto. No entanto, mesmo fornecendo boas aproximações para objetos pouco complexos, esta abordagem não apresenta bons resultados no caso de superfícies mais complexas. Neste caso, uma segunda fase de processamento, denominada deformação de forma livre FFD (*free-form deformation*), é aplicada com o objetivo de fazer ajustes mais finos no modelo reconstruído.

A utilização de superquádricas requer algum conhecimento sobre a forma global do objeto. Os autores argumentam que o trabalho está voltado para nuvens de pontos adquiridas a partir de imagens médicas, em particular, do coração. Como exemplo, apresentam reconstruções do ventrículo esquerdo, utilizando superelipsóides como forma inicial. O superelipsóide pode ser representado pela função implícita:

Figura 3.5: (Bardinet et al., 1998) Reconstrução que utiliza superelipsóides

$$\left(\left(\left(\frac{x}{r_1} \right)^{\frac{2}{\lambda_2}} + \left(\frac{y}{r_2} \right)^{\frac{2}{\lambda_2}} \right)^{\frac{\lambda_2}{\lambda_1}} + \left(\frac{z}{r_3} \right)^{\frac{2}{\lambda_1}} \right)^{\frac{\lambda_1}{2}} = 1 \quad (3.1)$$

Como consequência imediata, a equação do superelipsóide permite definir uma função:

$$F : \mathbb{R}^3 \rightarrow \mathbb{R}$$

$$F(x, y, z) = \left(\left(\left(\frac{x}{r_1} \right)^{\frac{2}{\lambda_2}} + \left(\frac{y}{r_2} \right)^{\frac{2}{\lambda_2}} \right)^{\frac{\lambda_2}{\lambda_1}} + \left(\frac{z}{r_3} \right)^{\frac{2}{\lambda_1}} \right)^{\frac{\lambda_1}{2}} \quad (3.2)$$

de maneira que dado um ponto p :

- Se $F(p) > 1$, p está fora da superfície;
- Se $F(p) = 1$, p está sobre a superfície;
- Se $F(p) < 1$, p está dentro da superfície;

Isto significa que é possível avaliar se os pontos da nuvem estão próximos à superfície inicial. No FFD o objeto a ser deformado é mergulhado em uma grade volumétrica ligando-a ao objeto (por um polinômio que define uma função deformação). A grade volumétrica então é deformada por meio de deslocamentos, e a deformação é transmitida ao objeto pela função deformação. Muitas variações de FFD podem ser encontradas na literatura. No entanto, os autores utilizaram a definição clássica, por assegurar bons resultados e bom desempenho computacional. Na FFD clássica, utilizam-se polinômios de Bernstein (Isaacson e Keller, 1966) como funções de deformação e uma grade cartesiana volumétrica para definir a grade de controle.

3.3.2 Métodos baseados em zeros de funções

Estes métodos consistem em utilizar a amostragem para definir uma função distância com sinal f e calcular o conjunto de zeros de f , que será denotado por $Z(f)$. A superfície é estimada como a superfície de nível de f .

O problema de reconstrução a partir de nuvem de pontos se tornou popular após o artigo de Hoppe et al. (1992), que descreve um algoritmo executado em dois estágios:

O primeiro consiste em definir uma função $f : D \subset \mathbb{R}^3 \rightarrow \mathbb{R}$, de tal forma que o conjunto $Z(f)$ seja uma estimativa para a superfície S procurada. A segunda fase consiste em aproximar $Z(f)$ por um complexo simplicial.

A função distância definida pelos autores é dada por $f(p) = \text{dist}_i(p) = (p - o_i) \cdot \hat{n}_i$, onde o_i é o centro (centróide) de uma região plana e $\hat{n}_i$ é a normal unitária desta região. O centro o_i é definido

pelo conjunto dos k pontos mais próximos de p (p pertence à nuvem de pontos), denotado por $V(p)$. O cálculo do vetor normal é feito pela matriz de covariância simétrica e semidefinida positiva dada por:

$$Cov = \sum_{x \in V(p)} (x - o_i)^t (x - o_i)$$

Para obter uma aproximação do conjunto de zeros, os autores utilizam uma variação do algoritmo Marching Cubes na qual cada cubóide é decomposto em tetraedros. Infelizmente, esse algoritmo não é capaz de capturar detalhes finos de um objeto e, além disso, envolve grande quantidade de cálculos geométricos.

O algoritmo desenvolvido por Curless e Levoy (1996) é mais efetivo; ele utiliza uma combinação acumulativa de pesos para construir sua função distância. Entretanto, esse algoritmo utiliza informações do scanner, além das amostras. Os autores argumentam que, sob certas condições, a superfície é ótima com relação à aproximação.

Bajaj et al. (1995) propõem um método que opera em três fases:

A primeira fase, denominada pré-processamento, consiste em criar uma função distância com sinal, em que o sinal da função vem da classificação dos tetraedros da triangulação de Delaunay (como internos e externos). A classificação dos tetraedros é realizada com auxílio de um α -shape calculado sobre a amostragem (o ponto está fora se estiver contido em um tetraedro externo ou o ponto está dentro se estiver localizado no interior de um tetraedro interno). A distância da função é obtida com auxílio do diagrama de Voronoi.

Na aproximação, como é designada a segunda fase, o espaço é subdividido em tetraedros que contêm porções do domínio. Os tetraedros são refinados até satisfazer condições de erro estabelecidas pelo usuário. Nesta fase, garante-se uma aproximação de classe C^0 .

A fase final, denominada suavização, consiste em suavizar a superfície obtida nas fases anteriores transformando-a numa superfície de classe C^1 . Observe que, também nesse algoritmo, uma grande quantidade de operações geométricas é realizada antes de obter a aproximação final, o que torna o algoritmo bastante vulnerável a erros de ordem numérica.

Boissonnat e Cazals (2002) utilizam vizinhanças naturais e superfícies r -amostradas para estabelecer garantias teóricas a respeito de sua reconstrução.

Dado um conjunto de pontos $P \subset \mathbb{R}^3$, os vizinhos naturais $N_{x,P}$ de um ponto $x \in \mathbb{R}^3$ são dados

pelo seguinte conjunto:

$$N_{x,P} = \{p \in P \mid V(x) \cap V_p \neq \emptyset\}$$

onde $V(x)$ é a célula de Voronoi de x em $V_{P \cup \{x\}}$ (diagrama de Voronoi de $P \cup \{x\}$), e V_p é a célula de Voronoi de p em V_P .

Em seguida, uma função distância global h é construída com base neste conjunto de vizinhos naturais. Esta função é construída de modo a ser continuamente diferenciável e, assim, evitar pontos não regulares.

Finalmente, aplica-se h a todos os vértices das arestas de Voronoi. O conjunto de triângulos duais das arestas de Voronoi que possuem sinais diferentes em seus dois vértices define a aproximação linear por partes da superfície.

Recentemente, novas técnicas de reconstrução baseadas em zeros de funções têm sido propostas. O trabalho de Carr et al. (2001), por exemplo, emprega uma base de funções radiais para aproximar a função distância. O algoritmo proposto, embora computacionalmente custoso, é capaz de lidar com falhas na amostragem e capturar detalhes finos do modelo.

Ohtake et al. (2003) utilizam a partição da unidade para derivar um sofisticado algoritmo capaz de tratar amostragens da ordem de milhões de pontos, usando relativamente pouca memória RAM (aproximadamente 800 Mb para cada 4 milhões de pontos) e com uma considerável velocidade. Embora a capacidade de trabalhar com grandes conjuntos de dados seja uma vantagem das abordagens implícitas, elas em geral produzem superfícies que não necessariamente interpolam as amostras, o que pode ser uma solução inadequada para algumas aplicações. Além disso, a maioria delas não apresenta garantias teóricas.

3.3.3 Métodos incrementais

Os métodos incrementais partem de uma aresta, ou conjunto de arestas iniciais, que é denominada fronteira de avanço, e obtêm os demais simplexes da reconstrução de maneira iterativa. Tais métodos também são conhecidos como métodos de avanço de fronteira.

Uma etapa de pré-processamento necessária nos métodos incrementais é o armazenamento dos pontos em alguma estrutura de dados espacial (kd-tree, quad-tree ou grade volumétrica (Samet, 1990)), pois dada uma aresta da fronteira de avanço e_f , o algoritmo deve ser capaz de encontrar de maneira eficiente o conjunto de pontos candidatos a definir um novo triângulo com e_f . Testes de interseção também são freqüentes nestes métodos.

Bernardini et al. (1999) apresentam um dos mais conhecidos algoritmos incrementais. Ele é denominado *Ball-Pivoting Algorithm* (BPA) e é conceitualmente simples. Considere-se uma 2-variedade em $\mathbb{R}^3$ e uma amostragem suficientemente densa desta variedade. Seja B , uma bola de raio ρ que não atravessa a variedade sem tocar em pontos da amostragem.

O processo se inicia “colocando” a bola B em contato com três pontos da amostragem (triângulo inicial). Mantendo a bola em contato com dois destes pontos, ela é pivoteada até tocar um outro ponto. O pivoteamento se dá ao redor de cada aresta da fronteira atual, e cada triplas de pontos tocados pela bola pivoteada define um novo triângulo (ver figura 3.6 b)).

A saída deste algoritmo é um subconjunto do α -shape. Os autores argumentam que algumas das propriedades do α -shape também são garantidas para o BPA.

Uma das principais vantagens deste algoritmo é que sua saída é um subconjunto da triangulação de Delaunay obtida sem a necessidade do cálculo da triangulação, uma vez que este é um processo custoso. Infelizmente, o BPA é sensível a variações de curvatura, (ver figura 3.6 a)), exigindo assim, uma amostragem com densidade uniforme.

Figura 3.6: a) Ball Pivoting bidimensional com diferentes curvaturas e taxas de amostragem b) Ball Pivoting tridimensional

Gopi et al. (2000) apresentam um algoritmo denominado Triangulação de Delaunay Localizada.

O algoritmo tem esse nome devido ao fato de executar cálculos locais de triangulações de Delaunay bidimensionais. São apresentadas garantias teóricas baseadas nas curvaturas da superfície e na distribuição dos pontos. O algoritmo é capaz de reconstruir variedades com ou sem bordo.

Quatro passos principais definem este algoritmo:

1. Cálculo da normal: O primeiro passo neste algoritmo é calcular a normal de todas as amostras. Este passo também orienta, de maneira consistente, as normais das amostras para obter-se uma variedade orientada. O cálculo da normal é semelhante ao proposto por Hoppe et al. (1992).
2. Seleção dos pontos candidatos: Neste passo, escolhem-se os pontos que podem ser os possíveis vizinhos de um vértice na triangulação final. Então, encontram-se os pontos do conjunto P_p vizinhos para cada ponto p .
3. Cálculo da triangulação de Delaunay em vizinhanças: Todos os pontos em P_p são projetados para o plano tangente de p e este conjunto de pontos projetados é denotado por P_p^T .
4. Finalmente, o cálculo da triangulação de Delaunay bidimensional local dos pontos de P_p^T é realizado. Este cálculo é repetido para todos os pontos do conjunto de amostras.

Além disso, os autores apresentam alternativas para evitar possíveis inconsistências na representação da superfície.

Huang e Menq (2002) propõem um algoritmo que não busca uma malha que seja um subconjunto da triangulação de Delaunay. Um dos principais diferenciais deste trabalho está na etapa de pós-processamento em que os autores relacionam conceitos de Geometria Diferencial para definir e estimar tanto a curvatura de arestas quanto a curvatura da nuvem de pontos, que é uma estimativa da curvatura da superfície original.

A figura 3.7 ilustra a estimativa da curvatura da superfície. A estimativa é obtida a partir da definição de uma estrutura de vizinhança. Em 3.7-a), tem-se a configuração original da malha. Em 3.7-b), observam-se a estimativa da curvatura da aresta c , representada pela curva c e a estimativa da curvatura da superfície, representada pela curva d . Em 3.7-c) Nota-se que a malha foi reestruturada localmente para que as curvaturas fiquem próximas.

Mederos et al. (2003) apresentam um trabalho no qual é proposta uma redução das nuvens de pontos para, em seguida, realizar a reconstrução da superfície. Os principais passos deste algoritmo são:

Figura 3.7: Processo de estimativa das curvaturas da superfície e da malha proposta por Huang e Menq (2002)

1. Clusterização: Primeiro, divide-se a nuvem de pontos em conjunto de clusters¹. A característica principal desta etapa é que a variação da curvatura da superfície em cada cluster deve ser pequena.
2. Redução: Encontra-se o ponto representativo para cada cluster. O conjunto de pontos representativos será utilizado no algoritmo de reconstrução.
3. Triangulação: Aplica-se um método incremental para a triangulação dos pontos representativos com base no grafo do vizinho mais próximo (definição 14, capítulo 2).

Os autores também propõem um refinamento que melhor adapte a geometria do modelo. Entretanto, nenhuma garantia teórica é apresentada.

3.3.4 Métodos de esculpimento

Rodríguez et al. (2000) propõem uma heurística de esculpimento baseada no tamanho das faces dos tetraedros medida por meio do perímetro. Existem outras medidas, tais como a área e o circunraio. Segundo os autores, a escolha do perímetro tem vantagens, não só pelo fato de ser

¹Agrupamentos de pontos

computacionalmente mais barata que o cálculo de outras medidas, mas também porque, dependendo da posição dos pontos, a triangulação de Delaunay gera elementos muito irregulares; com isso o circunraio¹ e a área podem não ser boas escolhas. De fato, existem boas chances de que triângulos com perímetro pequeno estejam presentes na superfície reconstruída, entretanto será mostrado mais adiante neste trabalho que o circunraio é uma medida mais adequada.

O algoritmo α -shape de Edelsbrunner e Mücke (1994) possui sua versão tridimensional, porém, como na versão bidimensional, é extremamente dependente do parâmetro α , o que o torna útil apenas para amostragens com densidade uniforme.

Uma visão intuitiva de α -shape para complexo simplicial consiste em imaginar uma esfera com raio α cujo objetivo é “remover” todo o espaço que consegue atravessar, sem tocar nenhum dos pontos da triangulação. O objeto resultante é denominado α -shape. Dado um complexo simplicial, existe um número finito de α -shapes. Dois extremos são o 0-shape, que se reduz à coleção de pontos original, e o ∞ -shape, que coincide com o fecho convexo do conjunto de pontos. As figuras 3.8 e 3.9 mostram modelos reconstruídos com diferentes valores de α .

Figura 3.8: Dragão com 10000 pontos a) $\alpha = 35$ b) $\alpha = 44$

Figura 3.9: Coelho com 1000 pontos a) $\alpha = 30$ b) $\alpha = 50$ c) $\alpha = 70$

Teichmann e Capps (1998) apresentaram uma extensão do α -shape que varia automaticamente o valor de α de acordo com a densidade da amostragem. Entretanto, esta abordagem necessita

dos vetores normais nas amostras e, além disso, nenhuma garantia teórica sobre a reconstrução é apresentada.

Partindo da triangulação de Delaunay de um conjunto de amostras, Attene e Spagnuolo (2000) propõem uma heurística baseada na árvore geradora mínima euclidiana - EMST (definição 15) e no hipergrafo de Gabriel estendido - EGH (definição 17, capítulo 2). O EGH e o EMST são utilizados para localizar, na triangulação de Delaunay, triângulos que têm maior probabilidade de pertencer à superfície original. As seguintes condições (que são chamadas de condições de singularidade) são utilizadas em conjunto com o EGH e o EMST no critério de esculpimento proposto por Attene e Spagnuolo.

1. A remoção de um tetraedro com três triângulos de fronteira desconectará um vértice da fronteira atual. Portanto, remover este tipo de tetraedro não é permitido;
2. Um tetraedro com apenas uma face na fronteira pode ser removido apenas se o vértice oposto à face não está na fronteira (caso contrário gera um ponto singular);
3. Um tetraedro com duas faces na fronteira que compartilham uma aresta em comum pode ser removido apenas se sua aresta oposta não está na fronteira (caso contrário, gera uma aresta singular).

O cálculo de tais grafos não é tarefa trivial e representa a principal desvantagem desta técnica. O algoritmo 3.3.4 ilustra esta técnica de esculpimento.

Algoritmo 3 Heurística baseada nos EMST e EGH

```

1 Calcule-se a triangulação de Delaunay de  $P$ .
2 Construa-se uma pilha ordenando os tetraedros de maneira que a maior aresta que pertence à
  fronteira deve ser removida primeiro (Note-se que a remoção de um tetraedro deve obedecer às
  condições de singularidade).
Se o tetraedro  $t$  tem apenas uma face na fronteira, que não pertence ao EGH, Então
  3. Remova-se  $t$ .
Fim do Se
Se o tetraedro  $t$  tem duas faces na fronteira que não pertencem ao EGH, e a aresta comum a
  elas não pertence ao EMST, Então
  4 Remova-se  $t$ .
Fim do Se
```

Adamy et al. (2000) apresentam um algoritmo de esculpimento baseado no filtro guarda-chuva. Neste trabalho, garantias de reconstrução homeomorfa e próxima à superfície original são apresentadas para o caso bidimensional. Já no caso tridimensional, para obter reconstruções mais

satisfatórias, o autor propõe um pós-processamento baseado em técnicas de programação linear, o que acaba encarecendo o algoritmo.

As seguintes definições são empregadas no algoritmo proposto por Adamy:

Definição 35 (λ -bola) *Sejam $\lambda \in (0, 1]$ e σ um $(n-1)$ -simplexo. Uma λ -bola, denotada por $b_\lambda(\sigma)$, é uma bola aberta de dimensão n com diâmetro $\frac{\text{diam}(\sigma)}{\lambda}$ contendo os vértices de σ na sua fronteira.*

Definição 36 (λ -intervalo) *O λ -intervalo associado a um $(n-1)$ -simplexo σ constituído por pontos da amostragem é definido como:*

$$I(\sigma) = \{\lambda \in \mathbb{R} / b_\lambda(\sigma) \cap P = \emptyset\}$$

onde P é o conjunto de amostras.

A figura 3.10 ilustra o conceito de λ -bolas para diversos valores de λ .

Figura 3.10: Exemplos de λ -bolas para uma aresta definida pelos vértices a e b .

Definição 37 (guarda-chuva) - *Guarda-chuva U_p de uma amostra p é uma vizinhança de p homeomorfa a um disco fechado de dimensão 2, de tal forma que p não faça parte de seu bordo (ver figura 3.11).*

O critério de esculpimento para determinar o λ -complexo é definido da seguinte maneira: para cada ponto, escolhe-se um guarda-chuva ótimo a partir do grafo de Gabriel utilizando-se o seguinte critério de otimização: escolhe-se o guarda-chuva que minimize os limites inferiores dos λ -intervalos de todos os seus $(n-1)$ -simplexos. Isto é, a idéia do critério de otimização é escolher os triângulos

Figura 3.11: (Adamy et al., 2000) (a). guarda-chuva, (b) e (c) são contra-exemplos.

incidentes a cada vértice com limite superior $= 1$ e limites inferiores menores possíveis até definirem um guarda-chuva. Descartam-se os triângulos que não fazem parte do guarda-chuva.

Os autores mostraram que os λ -intervalos podem ser eficientemente calculados a partir da triangulação de Delaunay e ainda argumentam que esse trabalho tenta manter as boas características do α -shape e evitar características indesejáveis.

Giesen e John (2002b, 2003) apresentam um algoritmo de esculpimento baseado em sistemas dinâmicos. Este algoritmo utiliza o conjunto de amostras para definir uma função distância que tem o grafo de Gabriel como conjunto de pontos críticos².

Primeiramente, define-se a função distância para o conjunto de amostras da seguinte forma:

$$h : \mathbb{R}^3 \rightarrow \mathbb{R}$$

$$h(x) = \min_{p \in P} \|x - p\| \quad (3.3)$$

Giesen e John argumentam que ela não pode ser utilizada como uma função para os métodos baseados em zeros de funções. além disso, os autores demonstram que os pontos críticos de h são as intersecções entre simplexos da triangulação de Delaunay com seus respectivos duais do diagrama de Voronoi.

A partir desta função, então é definido um sistema dinâmico:

Definição 38 (Sistema Dinâmico) Seja $f : \mathbb{R}^3 \rightarrow \mathbb{R}$ uma função suave e ∇f o campo gradiente de f . A partir desta definição, tem-se a seguinte equação diferencial ordinária:

$$\frac{d}{dt} \phi(t, x) = \nabla f(\phi(t, x))$$

²Pontos onde a derivada da função se anula.

onde a aplicação $\phi : \mathbb{R} \times \mathbb{R}^3 \rightarrow \mathbb{R}^3$ deve satisfazer as seguintes propriedades:

1. $\phi(0, x) = x$
2. $\phi(t, \phi(s, x)) = \phi(t + s, x)$

Pode-se interpretar ϕ como a posição do ponto x no tempo t , que é denominado sistema dinâmico ou fluxo em $\mathbb{R}^3$.

Os autores definem um sistema dinâmico por meio da função h , definida em 3.3, impondo condições para h se tornar suave (da forma como h é definida, ela é suave apenas no interior das células de Voronoi).

Denominam-se por pontos fixos os pontos x , tal que $\phi(t, x) = x, \forall t \in \mathbb{R}$. Neste trabalho, os pontos fixos de ϕ são os pontos críticos de h .

Em seguida, os autores definem a *órbita* ϕ_x de cada ponto x , a qual descreve o movimento de x ao longo do tempo. Variedades estáveis são definidas como sendo, para cada ponto crítico x de h , o conjunto de todos os pontos cujas órbitas terminam em x .

Na seqüência, um subconjunto de *variedades estáveis* é utilizado para definir o *complexo de fluxo*, que é a reconstrução da superfície. A figura 3.12 mostra um exemplo de reconstrução onde o complexo de fluxo pode ser visto à esquerda e a superfície reconstruída à direita.

Figura 3.12: (Giesen e John, 2003) reconstrução do Buda via complexo de fluxo

Uma etapa de pós-processamento, denominada complexo de fluxo reduzido também é proposta para melhorar a reconstrução pela eliminação de alguns simplexos.

Edelsbrunner (2002) apresenta um algoritmo em que utiliza teoria de Morse contínua (ver Milnor (1963)) que funciona para dimensão arbitrária. Entretanto, apenas o caso tridimensional é discutido neste artigo.

Em linhas gerais, Edelsbrunner propõe um algoritmo que constrói uma relação de ordem acíclica sobre os simplexes da triangulação de Delaunay. Essa relação de ordem é motivada por um campo de fluxo contínuo gerado por uma função de Morse específica. O campo de fluxo é construído de tal forma que cada amostra é atraída pelo vértice de Voronoi mais próximo. Na sequência, o algoritmo proposto por Edelsbrunner “esculpe”, isto é, remove os simplexes, de acordo com a relação de ordem, até obter uma reconstrução da superfície.

Na prática, o autor, ao contrário da proposta deste trabalho, substitui as propriedades topológicas que definem a relação de ordem, por testes geométricos que dependem do tipo de intersecção entre os simplexes e seus duais, o que acaba encarecendo o algoritmo. Adicionalmente, nenhuma garantia teórica a respeito da reconstrução é oferecida.

Também é importante mencionar que este algoritmo foi implementado com fins comerciais em 1996 pela empresa Raindrop Geomagic³ e patentado nos Estados Unidos.

Kolluri et al. (2004) introduzem um algoritmo denominado Eight-Crust, capaz de eliminar ruídos de uma maneira bastante eficaz.

O Eight-Crust inicia com o cálculo da triangulação de Delaunay e depois utiliza uma variação de um grafo espectral particionado para decidir quais tetraedros estão dentro e quais estão fora do objeto original. O objeto reconstruído é formado pelo conjunto de triângulos compartilhados, ao mesmo tempo, por tetraedros externos e internos.

Os autores argumentam que, devido ao fato de o grafo espectral tomar decisões locais com base em informações globais do modelo, os pontos de ruído podem ser eliminados. Além disso, os autores garantem a produção de uma variedade linear por partes como saída do algoritmo. A grande desvantagem desta técnica é que ela se baseia no cálculo dos auto-valores da matriz do grafo espectral, que pode ser extremamente grande, o que pode encarecer bastante o processo.

Assim como foi feito na descrição dos algoritmos bidimensionais, será dada uma atenção especial a um determinado grupo de algoritmos, no caso os algoritmos derivados do Crust (Crust, Power Crust, Cocone e Tight Cocone), devido ao fato de vários de seus resultados serem utilizados e de se compararem os resultados do algoritmo aqui proposto com os dos algoritmos deste grupo.

Família Crust tridimensional (Crust, Power Crust, Cocone e Tight Cocone)

Vale para o caso bidimensional que os vértices do diagrama de Voronoi de uma curva “bem

³<http://www.geomagic.com/>

amostrada” é uma aproximação do eixo medial da curva. Infelizmente, esse fato não é verdade no caso tridimensional, pois os vértices de Voronoi podem aparecer arbitrariamente próximos da superfície, o que os torna uma péssima aproximação para o eixo medial (ver figura 3.13). Além disso, em regiões de curvatura muito alta, o eixo medial praticamente toca a superfície, o que exigiria uma quantidade extremamente grande de amostras para garantir uma boa reconstrução.

Por outro lado, muitos dos vértices do diagrama de Voronoi se aproximam do eixo medial. Logo, a principal idéia dos métodos do tipo Crust é identificar tais vértices e usá-los como aproximação do eixo medial.

3.3.5 Crust Tridimensional

Como extensão do algoritmo Crust bidimensional, foram apresentados dois trabalhos que se complementam. O primeiro mostra resultados teóricos a respeito da reconstrução (Amenta e Bern, 1999) e o segundo discute a complexidade do algoritmo, sua eficiência e aspectos de sua implementação (Amenta et al., 1998b). Este algoritmo é denominado Raw Crust.

Amenta e Bern (1999) apresentam garantias de reconstrução da superfície original para o algoritmo Raw Crust. No entanto, este exige uma taxa de amostragem muito densa. Para contornar o problema, etapas de pós-processamento são propostas. O Raw Crust, quando acrescido das etapas de pós-processamento, passa a ser denominado Crust (em sua versão tridimensional).

É importante observar que a definição formal de Crust não difere da definição utilizada por Amenta e Bern no algoritmo Raw Crust. Entretanto, quando se fala em *algoritmo Crust* está-se nos referindo ao Raw Crust juntamente com as etapas de pós-processamento.

O problema de selecionar os vértices de Voronoi adequados para a reconstrução é resolvido com a introdução do conceito de pólo.

Definição 39 (Amenta e Bern, 1999) (**Pólos**) Seja F uma superfície suave, p uma amostra e V_p sua célula de Voronoi. Os dois vértices de V_p mais distantes de p , porém um do lado de dentro e um do lado de fora de F , são ditos **pólos** de p .

Definição 40 (Amenta e Bern, 1999) (**Crust**) Seja $P \subset \mathbb{R}^3$ um conjunto de pontos e V_P o conjunto dos vértices de Voronoi de P que são pólos. O Crust de P é definido como o conjunto de triângulos da triangulação de Delaunay de $P \cup V_P$ os quais possuem seus três vértices no conjunto P .

Das duas definições acima, um algoritmo de reconstrução tridimensional pode ser derivado.

Algoritmo 4 Raw Crust

(Para um conjunto P de amostras.)

1 Compute-se o diagrama de Voronoi do conjunto P .

Para cada amostra p . Faça

2 Computem-se seus pólos.

Fim do Para

3 Seja V_P o conjunto de todos os pólos. Compute-se a triangulação de Delaunay de $P \cup V_P$.

4 Mantenham-se apenas os triângulos cujos vértices pertencem a P .

O teorema a seguir fornece a taxa de amostragem suficiente para garantir a reconstrução.

Teorema 6 (Amenta e Bern, 1999) *Seja P um conjunto de amostras de uma superfície r -amostrada e suave S . Se $r < 0.06$, então:*

1. *O Crust de P contém um conjunto de triângulos topologicamente equivalente a S .*
2. *Cada ponto do Crust tem distância $5rLFS(x)$ a algum ponto x de S , onde $LFS(x)$ é a distância de x ao eixo medial.*

A duas etapas de pós-processamento propostas pelos autores para tornar a reconstrução mais eficiente são:

1. *Filtering by Normal:* Remove os triângulos cujo ângulo entre sua normal e o vetor definido pelo pólo mais distante de um vértice do triângulo é grande (os autores iniciam com um ângulo próximo de $\frac{\pi}{2}$ e vão diminuindo-o até o surgimento de buracos na superfície).
2. *Trimming:* Orienta os triângulos e pólos consistentemente, isto é, todos os triângulos possuirão orientação no sentido anti-horário, e os pólos serão marcados como internos ou externos à superfície. Nesta etapa, extrai-se uma variedade linear por partes sem bordo.

Ainda neste trabalho, os autores propõem um método de detecção de pólos (o que não é uma tarefa trivial) e demonstram que para uma superfície r -amostrada o método é eficiente.

Demonstraram ainda que, neste contexto, as células de Voronoi são finas e compridas e, por isso, os pólos podem ser utilizados como aproximação para os vetores normais da superfície. Pode ser observado, na figura 3.13, (onde os pontos vermelhos representam os vértices de Voronoi) que existe um vértice de Voronoi extremamente próximo da superfície, provavelmente vindo de um tetraedro sliver.

Figura 3.13: (Amenta et al., 1998b) Célula de Voronoi tridimensional de uma amostra

3.3.6 Power Crust

São apresentados, em primeiro lugar, as definições necessárias para a caracterização do Power Crust.

Definição 41 (Amenta et al., 2001b) (*Bolas Polares*) *As esferas com centro nos pólos e que tocam as amostras mais próximas são denominados bolas polares.*

Definição 42 (Amenta et al., 2001b) (*Power Crust*) *O power diagrama⁴ divide o espaço em um conjunto de células que se podem classificar como internas ou externas à superfície. O conjunto de faces poligonais que separam as células internas das externas é chamado de Power Crust.*

Dois artigos descrevem o algoritmo Power Crust: o primeiro descreve fundamentos teóricos, resultados de garantia da reconstrução e um algoritmo teórico para a rotulação das bolas polares (Amenta et al., 2001b); o segundo, detalhes do algoritmo e características da implementação (Amenta et al., 2001a). Neste último, um algoritmo prático para a rotulação das bolas polares é apresentado, pois, na prática, rotular corretamente as bolas polares como internas e externas pode ser uma tarefa complicada. Também são apresentadas extensões do algoritmo para tratamento de amostragens ruidosas, erros de aproximação de ponto flutuante e representações de variedades com bordo.

O seguinte teorema, demonstrado por Amenta et al. (2001b), dá uma ideia intuitiva de que o power diagrama pode ser utilizado para reconstrução:

Teorema 7 (Amenta et al., 2001b) *Dado o diagrama de Voronoi V de uma nuvem de pontos em $\mathbb{R}^n$, o Power Diagrama de todos os vértices de V , sendo os pesos de tais vértices a distância ao*

⁴Calculado considerando as bolas polares como pontos com peso.

quadrado do vértice de Voronoi à amostra mais próxima, é a triangulação de Delaunay da nuvem de pontos.

Observação 1 (Condição de amostragem) Os autores assumem que as amostragens utilizadas nas demonstrações das garantias são sempre r -amostragens, com $r < 0.1$ e que as variedades são sem bordo e sem auto-intersecções.

Os seguintes teoremas, também demonstrados por Amenta et al. (2001b), garantem que o Power Crust, para uma r -amostragem com r suficientemente pequeno é um objeto homeomorfo e próximo a superfície original.

Teorema 8 (Amenta et al., 2001b) (Intersecção rasa) Seja S uma superfície suave r -amostrada, com r suficientemente pequeno. O ângulo entre uma bola polar interna e externa é no máximo $r_1 = \frac{r}{1-r} = O(r)$ (ver figura 3.14).

Figura 3.14: Intersecção entre duas bolas polares

Uma outra idéia por trás deste teorema é que as bolas polares internas estão quase inteiramente contidas no interior do objeto. Amenta e Kolluri (2000) apresentam um trabalho no qual relatam que a união das bolas polares internas é uma aproximação para o objeto. A figura 3.15 apresenta exemplos desta situação.

Teorema 9 (Amenta et al., 2001b) Cada ponto da amostragem pertence ao Power Crust.

Teorema 10 (Amenta et al., 2001b) (Proximidade) Qualquer ponto u de uma face do Power Crust está a uma distância de, no máximo, $O(r)LFS(x)$ para algum x da superfície.

Figura 3.15: (Gois, 2004) Exemplos da aproximação de objetos por bolas polares

Teorema 11 (Amenta et al., 2001b) (*Homeomorfismo*) *Existe uma deformação contínua que transforma o Power Crust na superfície.*

Observação 2 *A saída do algoritmo Power Crust é sempre uma variedade sem bordo.*

Em linhas gerais, o algoritmo 5 caracteriza o Power Crust. Observe-se que, como no caso bidimensional, é necessário calcular duas triangulações de Delaunay. Além disso, o processo de nomeação dos pólos é caro e sujeito a instabilidade numérica.

Algoritmo 5 Power Crust

(Para um conjunto P de amostras)

1. Compute-se o diagrama de Voronoi do conjunto P .
 - Para cada amostra p . Faça
 2. computem-se seus pólos.
 - Fim do Para
 3. Compute-se o Power Diagrama dos pólos.
 4. Marque-se cada pólo como interno ou externo a superfície F .
 5. Retornem-se as faces separando as células internas e externas como **Power Crust**.
-

3.3.7 Cocone

Apesar de as garantias teóricas de reconstrução e de os modelos gerados através do Crust e do Power Crust serem satisfatórios, o esforço computacional necessário para realizar duas triangulações de Delaunay é a principal desvantagem destes algoritmos.

Amenta et al. (2002a) propõem uma elegante simplificação do Crust que utiliza apenas uma triangulação de Delaunay no processo de reconstrução. Este algoritmo é denominado Cocone. As garantias teóricas apresentadas neste trabalho também são simplificadas.

Um Cocone C_p de um ponto p de uma amostragem é definido como o duplo cone, sendo p seu ápice. O ângulo de abertura do cone é definido a partir da normal no ponto p como sendo $\frac{3\pi}{8}$ (ver figura 3.16).

Como o pólo de p^+ de p aproxima o vetor normal n em p , o cocone aproxima uma vizinhança do plano tangente em p . Para cada ponto p , o algoritmo determina todas as arestas de Voronoi de V_p que são intersectadas pelo cocone C_p . Os triângulos duais à essas arestas constituem o conjunto de triângulos que definem a reconstrução.

Figura 3.16: Ilustração de um Cocone de uma amostra p e sua célula de Voronoi

Amenta et al. (2002a) garantem homeomorfismo e proximidade na reconstrução gerada pelo cocone. Os autores demonstraram que os triângulos candidatos à representação da superfície tendem a ser pequenos e a apresentar uma forma de um triângulo equilátero

Teorema 12 (Amenta et al., 2002a) *Seja $r_c(t)$ o circunraio de qualquer triângulo t de uma r -amostragem. Então, para cada vértice p de t , tem-se: $r_c(t) \leq \frac{1.3r}{1-r} LFS(p)$, para $r \leq 0.09$.*

Mostram ainda que tais triângulos com circunraio pequeno tendem a serem bem alinhados com a superfície original. E finalmente, que, para valores de r adequados, a superfície gerada pelo Cocone é homeomorfa à superfície original.

Teorema 13 (Amenta et al., 2002a) (**Homeomorfismo**) *Seja uma superfície S r -amostrada com $r \leq 0.06$. Então, a reconstrução pelo algoritmo Cocone é homeomorfa à S .*

Além da utilização de apenas uma triangulação de Delaunay, este algoritmo também elimina

a etapa de pós-processamento filtering by normal, proposta para o algoritmo Crust, pois ela já é alcançada diretamente pelo Cocone. Apenas a segunda etapa de pós-processamento é aplicada.

3.3.8 Tight Cocone

O Tight Cocone, proposto por Dey e Goswami (2003), é uma etapa de pós-processamento do Cocone. A saída deste algoritmo é uma superfície denominada water tight:

Definição 43 (Dey e Goswami, 2002a) (**Water Tight**) Uma superfície é denominada water tight se é um 2-complexo simplicial em $\mathbb{R}^3$ cujo espaço gerado é o mesmo que a fronteira de uma 3-variedade em $\mathbb{R}^3$.

As figuras 3.17 (a) e (b) mostram exemplos de superfícies water tight, enquanto que as figuras 3.17 (c) e (d) mostram superfícies que não são water-tight.

Figura 3.17: a) e b) Exemplos de superfícies water tight, c) e d) são contra-exemplos

A idéia geral deste algoritmo é nomear os tetraedros de Delaunay como internos e externos, com base na superfície obtida pelo Cocone. Os tetraedros nomeados como externos são removidos, e a fronteira dos tetraedros nomeados como internos definem a fronteira de uma superfície water tight.

Esse fato é possível devido ao princípio da localidade, que assegura que regiões mal amostradas são locais.

Isso significa que o Cocone calcula a maioria da superfície esperada, com buracos que são localmente reparáveis. Claro que, se este princípio não for satisfeito, a saída do Tight Cocone pode não ser uma boa representação da superfície original.

Entretanto, os estudos realizados por Gois (2004) mostraram que o Tight Cocone não é capaz de reconstruir superfícies com mais de uma componente conexa.

3.3.9 Comparações

Sumarizando as informações sobre os algoritmos da família Crust, são apresentadas as tabelas de 3.1 à 3.3, extraídas do trabalho de Gois (2004).

Na tabela 3.1, são apresentadas informações teóricas sobre os algoritmos. A segunda coluna se refere ao tipo de malha superficial gerada por cada método. A terceira coluna apresenta resultados teóricos referentes à garantia de homeomorfismo. A última coluna diz se os algoritmos, em geral, fornecem resultados geometricamente próximos à superfície original.

Na tabela 3.2, são discutidos os custos computacionais de cada algoritmo. A segunda coluna apresenta o número de triangulações de Delaunay, em que n é número de pontos considerados durante o processo de triangulação. Na terceira coluna, o número de cálculos de duais e o número de pontos envolvidos para cada cálculo é especificado. Na última coluna, são apontados os cálculos geométricos mais custosos para cada algoritmo.

As principais vantagens e desvantagens mais significativas de cada um dos algoritmos são mostradas na tabela 3.3.

Algoritmo	Tipo de Malha	Garantias de Homeomorfismo	Proximidade Geométrica
raw crust	Triangular	Sim	Sim
cocone	Triangular	Sim	Sim
tight cocone	Triangular	Sim	Sim
power crust	Híbrida	Sim	Sim

Tabela 3.1: (Gois, 2004) Informações Teóricas

Algoritmo	Delaunay / Número de Pontos	Voronoi / Número de Pontos	Cálculo Geométrico mais Custoso
raw crust	1 $3n$	1 $2n$	Duais
cocone	1 n	1 n	Duais e ângulos entre vetores
tight cocone	1 n	1 n	Duais e ângulos entre vetores
power crust	2 n e $2n$	2 n e $2n$	Duais, ângulos entre vetores, ângulos entre esferas e funções trigonométricas

Tabela 3.2: (Gois, 2004) Custo Computacional

Algoritmo	Vantagens	Desvantagens
raw crust	Proximidade Geométrica	número de pontos na triangulação, cálculo de duais e triângulos indesejáveis na reconstrução
cocone	Melhoramento do Crust e cálculo de uma única triangulação	Aparecimento de buracos na superfície
tight cocone	Reconstrução homeomorfa e cálculo de uma única triangulação	Pontos isolados na reconstrução e não identificação de componentes internas
power crust	Reconstrução homeomorfa	Cálculo de duais, Pontos Steiner na reconstrução e instabilidade numérica

Tabela 3.3: (Gois, 2004) Vantagens e Desvantagens

3.4 Considerações Finais

Foram apresentados, neste capítulo, as principais técnicas de reconstrução a partir de nuvem de pontos, tanto bidimensionais, quanto tridimensionais. Comparações mais apuradas entre as técnicas e implementações de algumas delas podem ser encontradas no trabalho de Gois (2004). Além dessas implementações, serão usadas, para comparações, as implementações dos algoritmos Cocone e Tight Cocone, as quais nos foram gentilmente cedidas pelo pesquisador Tamal K. Dey.

A seguir, serão apresentados conceitos sobre teoria de Morse discreta, que é a principal ferramenta utilizada em nos algoritmos de reconstrução desenvolvidos nesta tese.

Capítulo 4

Teoria de Morse Discreta

4.1 Considerações Iniciais

A teoria de Morse (Milnor, 1963) é uma importante ferramenta utilizada na investigação da topologia de variedades diferenciáveis. Por meio dessa teoria, estabeleceu-se que a topologia de uma variedade está relacionada com os pontos críticos de uma função real nela definida. Os trabalhos de Forman (1998a,b) estenderam diversos aspectos dessa teoria para estruturas discretas, o que proporciona um aspecto combinatório à teoria, permitindo a implementação computacional praticamente independente de uma realização geométrica.

Neste capítulo, os principais conceitos sobre teoria de Morse discreta serão apresentados. Tal teoria é parte importante do arcabouço teórico deste projeto, além de desempenhar um importante papel nos critérios de reconstrução. O capítulo começa com as informações preliminares, em seguida descreve os principais teoremas, apresenta alguns exemplos de funções de Morse, define o campo de vetores gradiente e seu fluxo associado e termina descrevendo os caminhos gradientes. As demonstrações que não aparecem nesse capítulo podem ser encontradas no artigo “Morse Theory for Cell Complexes” (Forman, 1998a).

4.2 Preliminares

Embora a teoria de Morse discreta seja válida para decomposições celulares em geral, este trabalho utiliza apenas complexos simpliciais regularizados. Isso significa que qualquer simplexo de dimensão menor que n está contido em um n -simplexo, para $n = 2, 3$.

Seja $\mathcal{K}$ um complexo simplicial finito e K o conjunto de células de $\mathcal{K}$. Seja K_p o conjunto de células de dimensão p , e uma célula σ (aberta) de dimensão p , que é denotada por $\sigma^{(p)}$. Duas

células são relacionadas usando-se o símbolo $\tau \succ \sigma$ (ou $\sigma \prec \tau$) significando que $\sigma \neq \tau$ e $\sigma \subset \bar{\tau}$ onde $\bar{\tau}$ é o fecho de τ (σ é face de $\bar{\tau}$). Observe-se que $(\prec)$ não é uma relação de ordem.

4.2.1 Funções de Morse Combinatórias

Os principais conceitos e definições sobre funções de Morse combinatórias serão apresentados nessa seção. As mesmas definições e notações da seção anterior serão utilizadas, e $\mathcal{K}$ será considerado um complexo simplicial regularizado.

Definição 44 *Uma função de Morse discreta em $\mathcal{K}$ é uma função $f : K \rightarrow \mathbb{R}$ satisfazendo para todo $\sigma \in K_p$*

$$\#\{\beta \prec \sigma \mid f(\beta) \geq f(\sigma)\} \leq 1$$

$$\#\{\gamma \succ \sigma \mid f(\gamma) \leq f(\sigma)\} \leq 1$$

onde $\#$ denota o número de elementos no conjunto, sendo β e γ simplexes de dimensão $(p-1)$ e $(p+1)$, respectivamente

Definição 45 *Seja f uma função de Morse discreta em $\mathcal{K}$. É dito que $\sigma \in K_p$ é um ponto crítico de índice p se:*

$$\#\{\beta \prec \sigma \mid f(\beta) \geq f(\sigma)\} = 0$$

$$\#\{\gamma \succ \sigma \mid f(\gamma) \leq f(\sigma)\} = 0$$

Observação 3 *Uma vez que σ é face de dimensão p e, portanto, não pode ser ponto crítico de qualquer outro índice se não p , será chamado apenas de ponto crítico.*

Da definição 45, segue que um p -simplexo não é ponto crítico se, e somente se, uma das duas condições valem:

1. $\exists \tau^{\{p+1\}} \succ \sigma$ tal que $f(\tau) \leq f(\sigma)$;
2. $\exists v^{\{p-1\}} \prec \sigma$ tal que $f(v) \geq f(\sigma)$;

Lema 1 (Forman, 1998b) *As duas condições acima não podem ser ao mesmo tempo verdadeiras.*

4.3 Teoremas de Morse

Nesta seção apresentam-se os principais resultados em teoria de Morse discreta.

Definição 46 Para $c \in \mathbb{R}$, define-se

$$M(c) = \bigcup_{\substack{\sigma \in K \\ f(\sigma) \leq c}} \sigma \bigcup_{\tau \prec \sigma} \tau$$

Isto é, $M(c)$ denota todos os simplexos para as quais $f \leq c$, assim como todas as suas faces. Em particular, $M(c)$ é um subcomplexo de K .

Lema 2 (Forman, 1998b) Seja σ um p -simplexo de K e suponha-se $\tau \succ \sigma$. Então existe um $(p+1)$ -simplexo $\tilde{\tau}$ com $\tau \preceq \tilde{\tau} \succ \sigma$ e $f(\tilde{\tau}) \leq f(\tau)$.

Seja $m_p(f)$ (ou m_p) o número de pontos críticos de f de índice p . Os m'_p s são chamados números de Morse de f .

Seja F um corpo e $b_i = \dim(H_i(K, F))$ o i -ésimo número de Betti com coeficientes em F .

Corolário 1 (Forman, 1998b) (As desigualdades fortes de Morse) Para qualquer $N \geq 0$

$$m_N - m_{N-1} + \dots \pm m_0 \geq b_N - b_{N-1} + \dots \pm b_0$$

Corolário 2 (Forman, 1998b) (As desigualdades fracas de Morse)

1. Para todo $N, m_N \geq b_N$
2. $\chi(K) = b_0 - b_1 + \dots \pm b_{\dim(K)} = m_0 - m_1 + \dots \pm m_{\dim(K)}$

4.4 Exemplos de Funções de Morse Discretas

Cada complexo simplicial K tem uma função de Morse, basta tomar para todo $\sigma \in K$, $f(\sigma) = \dim(\sigma)$. Note-se que para esta função em particular, todas as células são críticas.

Lema 3 (Forman, 1998b) Seja K um complexo simplicial e $N \subseteq K$ um subcomplexo. Então qualquer função de Morse discreta em K pode ser restringida para uma função de Morse discreta em N . E, se $\sigma \subseteq N$ é um ponto crítico na função original, então também é ponto crítico na restrição.

Lema 4 *Seja $\mathcal{K}$ um complexo simplicial e $N \subseteq \mathcal{K}$ um subcomplexo. Então, qualquer função de Morse discreta em N pode ser estendida para $\mathcal{K}$, isto é, se f é uma função de Morse em N , então existe uma função de Morse g em $\mathcal{K}$ tal que*

$$g(\sigma) = f(\sigma) \text{ sempre que } \sigma \subseteq N.$$

Demonstração. Seja $c := \max_{\sigma \subseteq N} f(\sigma)$, g é definida como:

$$g(\sigma) = \begin{cases} f(\sigma) & \text{se } \sigma \subseteq N \\ c + \dim \sigma & \text{se } \sigma \not\subseteq N \end{cases}$$

Note-se que mais uma vez esta definição é muito ineficiente, pois todas as células de $\mathcal{K} - N$ são críticas. ■

4.5 O Campo de Vetores Gradiente Discreto e o Fluxo Associado

Em teoria de Morse contínua, o gradiente da função de Morse e o fluxo associado são ferramentas essenciais para a investigação da relação entre os pontos críticos e a topologia de uma determinada variedade. Nesta seção serão definidos o campo de vetores gradientes e o fluxo correspondente em um contexto discreto.

Seja $\mathcal{K}$ um complexo simplicial. O primeiro objetivo é definir um objeto V_f que representará um objeto análogo ao campo de vetores gradiente $-\nabla_f$ no caso contínuo; em seguida será definido o fluxo associado ϕ_f . Deve-se assumir que a função de Morse está definida, $\mathcal{V}$ denota V_f , e ϕ denota ϕ_f . Começando com os vértices de $\mathcal{K}$, seja $v \in K_0$, se v é um ponto crítico, qualquer definição razoável de gradiente deve se anular em v . Logo:

$$\mathcal{V}(v) = 0$$

Se v não é ponto crítico, então existe uma única aresta $e \succ v$ com $f(e) \leq f(v)$. A aresta e especifica uma única direção para a qual f não está crescendo, então $-\nabla_f$ deve ser e . Entretanto, é necessário ser mais rigoroso. $\mathcal{V}(v)$ deve ser pensado como um vetor tangente discreto partindo de v , isto é, $\mathcal{V}(v)$ deve ser e com uma orientação escolhida.

Serão introduzidos, aqui, os complexos das p -cadeias. Seja $C_p(\mathcal{K})$ o $\mathbb{Z}$ -módulo livre gerado sobre $\mathbb{Z}$ pelas células orientadas de $\mathcal{K}$ de dimensão p (definido no capítulo 2). Ainda σ é identificada como sendo σ em sua orientação oposta, e seja ∂ o operador bordo usual.

$$\begin{aligned}\partial & : C_p(\mathcal{K}) \rightarrow C_{p-1}(\mathcal{K}) \\ \partial\sigma & = \sum_{v^{(p-1)} \prec \sigma} \varepsilon(\sigma, v) v\end{aligned}$$

Onde $\varepsilon(\sigma, v)$ são inteiros chamados de números de incidência.

É conveniente introduzir um produto interno $\langle, \rangle$ em C_p , uma vez que as células de $\mathcal{K}$ formam uma base ortonormal. Utilizando-se o produto interno, o operador bordo pode ser escrito como:

$$\partial\sigma = \sum_{v^{(p-1)} \prec \sigma} \langle \partial\sigma, v \rangle v$$

Agora a definição de $\mathcal{V}$ pode ser finalizada. Se $v \in K_0$ não é ponto crítico e a aresta e satisfaz $e \succ v$ e $f(e) \leq f(v)$, $\mathcal{V}$ é definido como segue:

$$\mathcal{V}(v) = \pm e$$

onde o sinal é escolhido de tal forma que:

$$\langle \partial(\mathcal{V}(v)), v \rangle = -1$$

Fica claro agora como definir na versão discreta o fluxo ϕ nos vértices de $\mathcal{K}$: se v é crítico, isto é, se $\mathcal{V}(v) = 0$ então v é constante pelo fluxo gradiente. Logo

$$\phi(v) = v$$

Se v não é crítico, isto é, $\mathcal{V}(v) = \pm e$, então v deve “fluir” para o outro vértice de e , isto é,

$$\phi(v) = v + \partial(\mathcal{V}(v))$$

Observe-se que esta definição vale agora para qualquer vértice, seja ele crítico ou não.

$\mathcal{V}$ e ϕ podem ser estendidos linearmente para aplicações sobre as p -cadeias.

$$\mathcal{V} : C_0(\mathcal{K}) \rightarrow C_1(\mathcal{K})$$

$$\phi : C_0(\mathcal{K}) \rightarrow C_0(\mathcal{K})$$

Estas definições ainda precisam ser estendidas para células de dimensão maior que zero.

Definição 47 *Seja σ um p -simplexo de $\mathcal{K}$ (fixada uma orientação). Se existe uma $\tau^{(p+1)} \succ \sigma$ com $f(\tau) \leq f(\sigma)$, $\mathcal{V}$ é definido como segue:*

$$\mathcal{V}(\sigma) = -\langle \partial\tau, \sigma \rangle \tau$$

Note-se que, para assegurar a validade desta definição, σ deve ser face regular de τ , isto é, $\langle \partial\tau, \sigma \rangle = \pm 1$; caso contrário:

$$\mathcal{V}(\sigma) = 0$$

E, finalmente, para cada p , $\mathcal{V}$ é estendido linearmente para as p -cadeias

$$\mathcal{V} : C_p(\mathcal{K}) \rightarrow C_{p+1}(\mathcal{K})$$

Observe-se que o fato de $\mathcal{V}(\sigma) = 0$ não significa que σ é crítico. No exemplo mostrado pela figura 4.1 $\mathcal{V}(e_1)$ se anula. Entretanto, e_1 não deveria permanecer fixo em ϕ , pois f decresce de e_1 para os bordos.

Figura 4.1: Exemplo de Complexo Simplicial

Definição 48 *Para cada face orientada σ de $\mathcal{K}$, o fluxo gradiente ϕ é definido como :*

$$\phi(\sigma) = \sigma + \partial\mathcal{V}(\sigma) + \mathcal{V}(\partial\sigma)$$

ou sucintamente:

$$\phi = 1 + \partial\mathcal{V} + \mathcal{V}\partial$$

As principais propriedades de $\mathcal{V}$ e ϕ estão nos teoremas abaixo.

Teorema 14 *Seja $\mathcal{V} : C_p(\mathcal{K}) \rightarrow C_{p+1}(\mathcal{K})$:*

1. $\mathcal{V} \circ \mathcal{V} = 0$;
2. Se σ é um p -simplexo orientado então: $\# \{v^{(p-1)} / \mathcal{V}(v) = \pm\sigma\} \leq 1$;
3. σ é um simplexo crítico $\iff [\sigma \notin \text{Im}(\mathcal{V}) \text{ e } \mathcal{V}(\sigma) = 0]$;

Demonstração.

1. Se $\mathcal{V}(v^{(p-1)}) = \pm\sigma^{(p)}$ então $v \prec \sigma$ e $f(\sigma) \leq f(v)$. Pelo lema 1 não existe $\tau^{(p+1)} \succ \sigma$ com $f(\tau) \leq f(\sigma)$ então $\mathcal{V}(\sigma) = 0$
2. Se $\mathcal{V}(v^{(p-1)}) = \pm\sigma^{(p)}$ então $v \prec \sigma$ e $f(\sigma) \leq f(v)$. Pela condição (2) da definição 44 v é único.
3. A partir da definição 45 σ é crítico se e somente se

- (a) $\# \{\tau^{(p+1)} \succ \sigma / f(\tau) \leq f(\sigma)\} = 0$
- (b) $\# \{v^{(p-1)} \prec \sigma / f(v) \geq f(\sigma)\} = 0$

Mas estas afirmações são equivalentes a dizer que

1. (a) Não existe $\tau^{(p+1)}$ tal que $\mathcal{V}(\sigma) = \pm\tau$
- (b) Não existe $v^{(p-1)}$ tal que $\mathcal{V}(v) = \pm\sigma$

O que é equivalente a dizer que

1. (a) $\mathcal{V}(\sigma) = 0$
- (b) $\sigma \notin i\mathcal{K}(\mathcal{V})$

Teorema 15 *Seja $\sigma_1, \dots, \sigma_r$ a coleção das p -células de $\mathcal{K}$ (com uma orientação escolhida) escreva-se $\phi(\sigma_i) = \sum_j a_{ij} \sigma_j$*

1. $\partial\phi = \phi\partial$;
2. para todo i , $a_{ii} = 0$ ou $a_{ii} = 1$ e $a_{ii} = 1$ se e somente se σ_i é crítico;
3. se $i \neq j$, então $a_{ij} \in \mathbb{Z}$ e $a_{ij} \neq 0$ então $f(\sigma_j) < f(\sigma_i)$;

Demonstração.

$$1. \phi\partial = (1 + \partial\mathcal{V} + \mathcal{V}\partial)\partial = \partial + \partial\mathcal{V}\partial + \mathcal{V}\partial^2 = \partial + \partial\mathcal{V}\partial = \partial + \partial\mathcal{V}\partial + \partial^2\mathcal{V} = \partial(1 + \partial\mathcal{V} + \mathcal{V}\partial) = \partial\phi$$

(2) e (3) serão provadas simultaneamente. Note-se que ∂ e $\mathcal{V}$ levam cadeias sobre $\mathbb{Z}$ em cadeias sobre $\mathbb{Z}$ cada $a_{ij} \in \mathbb{Z}$. Pelo teorema 14, vale que $\sigma^{(p)}$ satisfaz uma das seguintes propriedades:

1. σ é crítico;
2. $\pm\sigma \in \text{im}(\mathcal{V})$;
3. $\mathcal{V}(\sigma) \neq 0$;

Cada caso será tratado separadamente. (1) se σ é crítico, então $\mathcal{V}(\sigma) = 0$. Logo

$$\phi(\sigma) = \sigma + \mathcal{V}(\partial\sigma) = \sigma + \sum_{v^{(p-1)} \prec \sigma} \langle \partial\sigma, v \rangle \mathcal{V}(v).$$

Como σ é crítico, para cada $v^{(p-1)} \prec \sigma$ então $f(v) < f(\sigma)$ e para cada v ou $\mathcal{V}(v) = 0$ ou $\mathcal{V}(v) = \tilde{\sigma}^{(p)}$ com $f(\tilde{\sigma}) \leq f(v) < f(\sigma)$ então

$$\begin{aligned} \phi(\sigma) &= \sigma + \sum a_{\tilde{\sigma}} \tilde{\sigma} \\ \text{onde } f(\tilde{\sigma}) &< f(\sigma) \end{aligned}$$

(2) Suponha-se que $\pm\sigma \in \text{im}(\mathcal{V}) \subseteq \text{Ker}(\mathcal{V})$. Então,

$$\phi(\sigma) = \sigma + \mathcal{V}(\partial\sigma) = \sigma + \sum_{v^{(p-1)} \prec \sigma} \langle \partial\sigma, v \rangle \mathcal{V}(v).$$

Pelo teorema 14 existe um único simplexo $\tilde{v}^{(p-1)} \prec \sigma$ tal que $\mathcal{V}(\tilde{v}) = \pm\sigma$ e $\langle \partial\sigma, \tilde{v} \rangle \mathcal{V}(\tilde{v}) = -\sigma$. Além disso, se $v \neq \tilde{v}$ então ou $\mathcal{V}(v) = 0$ ou $\mathcal{V}(v) = \pm\tilde{\sigma}^{(p)}$ com $f(\tilde{\sigma}) \leq f(v) < f(\sigma)$. Logo:

$$\begin{aligned} \phi(\sigma) &= \sum a_{\tilde{\sigma}} \tilde{\sigma} \\ \text{Com } f(\tilde{\sigma}) &< f(\sigma) \end{aligned}$$

(3) Suponha-se que $\mathcal{V}(\sigma) = -\langle \partial\tau, \sigma \rangle \tau \neq 0$. Então

$$\phi(\sigma) = \sigma + \mathcal{V}(\partial\sigma) + \partial(\mathcal{V}(\sigma))$$

E, desde que $\mathcal{V}(\sigma) \neq 0$, $\sigma \notin \text{im}(\mathcal{V})$, logo para cada $v^{(p-1)} \prec \sigma$ então ou $\mathcal{V}(v) = 0$ ou $\mathcal{V}(v) = \pm\tilde{\sigma}^{(p)}$ com $f(\tilde{\sigma}) \leq f(v) < f(\sigma)$

Olhe agora para $\partial(\mathcal{V}(\sigma))$. $\partial(\mathcal{V}(\sigma)) = -\langle \partial\tau, \sigma \rangle \partial\tau = -\langle \partial\tau, \sigma \rangle^2 \sigma + \sum b_{\tilde{\sigma}} \tilde{\sigma} = -\sigma + \sum b_{\tilde{\sigma}} \tilde{\sigma}$ onde $f(\tilde{\sigma}) < f(\tau) \leq f(\sigma)$ ■

Intuitivamente, este teorema mostra que ϕ decresce f e que $\sigma \subseteq \phi(\sigma)$ se, e somente se, σ é crítico. Observe-se novamente o exemplo mostrado pela figura 4.1 e suponha que e_1 esteja orientada no sentido de v_2 para v_3 , e_2 no sentido de v_3 para v_1 , e finalmente e_3 no sentido de v_1 para v_2 . Serão calculados $\phi(e_1)$, $\phi(e_2)$ e $\phi(e_3)$.

$$\begin{aligned} \phi(e_1) &= e_1 + \mathcal{V}(\partial e_1) = e_1 + \mathcal{V}(v_3 - v_2) = \\ &= e_1 + e_2 + e_3 \end{aligned}$$

$$\begin{aligned} \phi(e_2) &= e_2 + \mathcal{V}(\partial e_2) = e_2 + \mathcal{V}(v_1 - v_3) = \\ &= e_2 + 0 - e_2 = 0 \end{aligned}$$

$$\begin{aligned}
 \phi(e_3) &= e_3 + \mathcal{V}(\partial e_3) = e_3 + \mathcal{V}(v_2 - v_1) = \\
 &= e_3 - e_3 + 0 = 0
 \end{aligned}$$

4.6 Caminhos Gradientes

Definição 49 Um caminho gradiente de dimensão p é uma seqüência γ de p -células de $\mathcal{K}$, $\gamma = \sigma_0, \sigma_1, \dots, \sigma_r$ tal que para todo $i = 0, \dots, r$, as seguintes propriedades são verdadeiras:

1. Se $\mathcal{V}(\sigma_i) = 0$, então $\sigma_{i+1} = \sigma_i$;
2. Se $\mathcal{V}(\sigma_i) \neq 0$, então $\sigma_{i+1} \prec \mathcal{V}(\sigma_i)$ e $\sigma_{i+1} \neq \sigma_i$;

γ é dito um caminho gradiente de σ_0 até σ_r . O comprimento de γ , denotado por $|\gamma|$, é r .

O lema a seguir mostra duas propriedades facilmente verificáveis de caminhos gradientes.

Lema 5 1. se $\gamma = \sigma_0, \sigma_1, \dots, \sigma_r$ é um caminho gradiente, então para cada $i = 0, \dots, r-1$ ou $\sigma_{i+1} = \sigma_i$ ou $f(\sigma_i) > f(\sigma_{i+1})$;

2. se $\gamma_1 = \sigma_0, \sigma_1, \dots, \sigma_r$ e $\gamma_2 = \sigma_r, \sigma_{r+1}, \dots, \sigma_{r+s}$ são duas seqüências de p -células, então

$$\sigma_0, \sigma_1, \dots, \sigma_r, \dots, \sigma_{r+s}$$

é um caminho gradiente se, e somente se, γ_1 e γ_2 são caminhos gradientes;

Suponha $\tilde{\sigma} \neq \sigma$ duas p -células de $\mathcal{K}$ e $\tau^{(p+1)}$ um $(p+1)$ -simplexo tal que $\sigma \prec \tau$ e $\tilde{\sigma} \prec \tau$. Então, uma orientação em σ induz uma orientação em $\tilde{\sigma}$ da seguinte maneira: uma orientação em σ induz uma orientação sobre τ (de modo que $\langle \partial\tau, \sigma \rangle = -1$). Então, dada uma orientação em τ , uma orientação em $\tilde{\sigma}$ é escolhida de tal maneira que $\langle \partial\tau, \tilde{\sigma} \rangle = 1$. Equivalentemente, fixada uma orientação em τ e em σ , uma orientação para $\tilde{\sigma}$ é escolhida de tal forma que

$$\langle \partial\tau, \sigma \rangle \langle \partial\tau, \tilde{\sigma} \rangle = -1$$

Intuitivamente, a orientação em $\tilde{\sigma}$ é escolhida “deslizando” σ através de τ na direção de $\tilde{\sigma}$. Uma orientação é fornecida por σ , e as orientações em τ e $\tilde{\sigma}$ são induzidas.

Por outro lado, se $\tilde{\sigma} = \sigma$, então uma orientação em σ induz a mesma orientação em $\tilde{\sigma}$. Logo, se $\gamma = \sigma_0, \sigma_1, \dots, \sigma_r$ é um caminho gradiente, uma orientação em σ_0 induz uma orientação em cada σ_i , em particular em σ_r . Lembrando-se que foi fixada, a priori, uma orientação para cada simplexo de $\mathcal{K}$. É dito que $m(\gamma) = 1$ se a orientação fixada em σ_0 induz a orientação fixada em σ_r e $m(\gamma) = -1$ caso contrário, ou equivalentemente

$$m(\gamma) = \prod_{\substack{i=0 \\ \mathcal{V}(\sigma_i) \neq 0}}^{r-1} \langle \partial \mathcal{V}(\sigma_i), \sigma_i \rangle \langle \partial \mathcal{V}(\sigma_i), \sigma_{i+1} \rangle \quad (6.3)$$

Com essa formula, fica definida a multiplicidade de um caminho gradiente qualquer.

Definição 50 *Seja*

$$\gamma = \sigma_0, \sigma_1, \dots, \sigma_r$$

um caminho gradiente de dimensão p . A multiplicidade de γ é definida pela formula 6.3 e denotada por $m(\gamma)$.

Observe-se que, se $\gamma_1 = \sigma_0, \sigma_1, \dots, \sigma_r$ e $\gamma_2 = \sigma_r, \sigma_{r+1}, \dots, \sigma_{r+s}$ são caminhos gradientes, então

$$m(\gamma_1) m(\gamma_2) = m(\gamma_1 \circ \gamma_2)$$

onde $m(\gamma_1 \circ \gamma_2) = \sigma_0, \sigma_1, \dots, \sigma_r, \dots, \sigma_{r+s}$.

Observação 4 *O lema 5 indica que ao “andar” por um caminho gradiente chegar-se-á a um ponto de mínimo.*

4.7 Considerações Finais

Recentemente, várias aplicações para a teoria discreta de Morse discreta em Computação Gráfica tem sido apontadas pela literatura. Algumas delas estão sugeridas nos trabalhos de Lewiner (2002) e Lewiner et al. (2004). Entre elas podem-se citar compressão de malhas volumétricas, resolução de singularidades que aparecem nas construções de modelos, e *morphing* topologicamente consistente.

No próximo capítulo, será detalhado o algoritmo bidimensional para reconstrução de curvas em $\mathbb{R}^2$ proposto neste trabalho. Serão apresentados resultados computacionais, comparações com os

algoritmos mais clássicos e, ainda, serão dadas garantias teóricas a respeito da reconstrução correta do objeto original.

Capítulo 5

Reconstrução de Curvas

5.1 Considerações Iniciais

Este capítulo apresenta um algoritmo para reconstrução de curvas no plano a partir de um conjunto de amostras. Na sequência, mostra alguns resultados computacionais e comparações com alguns algoritmos da literatura, a saber, Crust, β -skeleton, Power-Crust, e NN-Crust.

Como foi mencionado anteriormente, o algoritmo de reconstrução aqui proposto é baseado em teoria de Morse discreta. A idéia do algoritmo é obter a aproximação poligonal da curva por meio da análise das arestas críticas e regulares de uma determinada função de Morse definida sobre a triangulação de Delaunay das amostras.

Nosso algoritmo será denominado Tor (Topological Object Reconstructor), cuja versão bidimensional será apresentada neste capítulo.

5.2 Algoritmo Bidimensional

Antes de se apresentar o algoritmo propriamente dito, a função de Morse discreta empregada nesta técnica será definida formalmente.

Sejam P um conjunto de amostras de uma curva fechada e suave $C \subset \mathbb{R}^2$, DT a triangulação de Delaunay de P , e $l(e)$ o comprimento de uma aresta $e \in DT$. Sejam também V , E e T o conjunto de vértices, arestas e triângulos de DT , respectivamente. A função de Morse discreta será construída de forma recursiva juntamente com o campo gradiente nos vértices e nas arestas de maneira a evitar ambigüidades.

Considere-se a seguinte função $f : DT \rightarrow \mathbb{R}$

1. para cada triângulo $t \in T$, $f(t) = \max\{l(e) \mid e \subset t\}$;
2. para cada aresta $e \in E$, $f(e) = l(e)$;
3. para cada vértice $v \in V$, $f(v) = \min\{l(e) \mid v \in e\}$.

A função definida acima pode não ser uma função de Morse discreta devido a ambigüidades. Por exemplo, uma aresta e pode ser a menor aresta incidente em seus dois vértices. Com objetivo de eliminar estas ambigüidades e transformar f em uma função de Morse discreta, introduziu-se o seguinte esquema de perturbação:

Suponha-se que f já esteja definida em todas as arestas como $f(e) = l(e)$. Para cada vértice $v \in V$ seja $\alpha = \min\{f(e) \mid v \in e; e \in E\}$ e $E_v = \{e \supset v \mid f(e) = \alpha\}$ o conjunto das arestas de menor comprimento incidentes em v . Existem duas situações possíveis: ou E_v possui um único elemento e , ou possui mais de um. No primeiro caso, se e ainda não é vetor gradiente de seu outro vértice, $f(v) = f(e)$ e, ainda, e é marcada como vetor gradiente em v ; caso contrário $f(v) = 0$ (e v é um vértice singular). No segundo caso, em que E_v possui mais de um elemento, uma aresta e é escolhida arbitrariamente. Como no primeiro caso, se e já é vetor gradiente de seu outro vértice, $f(v) = 0$; caso contrário, $f(e)$ é redefinida para $f(e) = l(e) - \varepsilon$ e $f(v) = f(e)$ onde $\varepsilon > 0$ é escolhido de tal forma a garantir que $f(e)$ continue maior do que o valor de f no outro vértice de e . Novamente $f(v) = f(e)$ e, em seguida, e é marcada como vetor gradiente em v .

Já para cada triângulo $t \in T$, sejam $\beta = \max\{f(e) \mid e \subset t\}$ e $E_t = \{e \subset t \mid f(e) = \beta\}$ o conjunto das maiores arestas de t . Se E_t contiver um único elemento e , este é selecionado; caso contrário, uma aresta e é escolhida arbitrariamente e o valor da função em e é redefinido para $f(e) = f(e) + \varepsilon$, ($\varepsilon > 0$). Uma vez selecionada a aresta e , existem duas situações possíveis: ou e é aresta com maior valor de f apenas para t , ou e é a aresta de maior valor para t e t' , onde t' é o outro triângulo que compartilha e (no caso bidimensional existem no máximo dois triângulos que compartilham a mesma aresta). No primeiro caso, $f(t)$ é definida como $f(t) = f(e)$ e t é marcado como vetor gradiente de e . No segundo caso, um dos triângulos, por exemplo t , é escolhido arbitrariamente, $f(t)$ é definida como $f(t) = f(e)$ e t é marcado como vetor gradiente de e , e finalmente, $f(t')$ é definida como $f(t) + 1$ tornando-se um triângulo singular.

Pode-se observar, na figura 5.1, um exemplo da construção da função de Morse discreta em um complexo simplicial formado pelos vértices a, b, c e d . O primeiro passo é definir o valor da função nas arestas (que poderá ser perturbado durante o processo); em seguida deve-se definir o valor da função em cada vértice. No vértice a , o conjunto E_v contém apenas a aresta $\overline{ab}$. Neste caso, defini-se $f(a) = f(\overline{ab})$ e $\overline{ab}$ é rotulada como vetor gradiente de a . Já para o vértice b , o conjunto E_v também contém apenas a aresta $\overline{ab}$, mas como esta aresta já é vetor gradiente do vértice a , defini-se $f(b) = 0$ e conseqüentemente b se torna um ponto crítico de f . A situação no vértice c é similar à do vértice a , defini-se $f(c) = f(\overline{cb})$, e a aresta $\overline{cb}$ é rotulada como vetor gradiente de c . Já para o vértice d , o conjunto E_v possui duas arestas ($\overline{dc}$ e $\overline{da}$). Neste caso, a aresta $\overline{dc}$ foi escolhida, o valor de f nela foi redefinido para $f(\overline{dc}) - \varepsilon$ e ela foi rotulada como vetor gradiente de d . O último passo é definir o valor da função nos triângulos. Para o triângulo $t_1 = bcd$ a maior aresta é $\overline{db}$. Neste caso, define-se $f(t_1) = f(\overline{db})$ e t_1 é rotulado como vetor gradiente de $\overline{db}$. Finalmente, para o triângulo $t_2 = abd$, a maior aresta também é $\overline{db}$. Como ela já tem vetor gradiente, defini-se $f(t_2) = f(\overline{db}) + 1$ e, por conseguinte, t_2 também é um ponto crítico de f .

Teorema 16 *A função f juntamente com o esquema de perturbação introduzido acima caracteriza uma função de Morse discreta.*

Demonstração. Para cada vértice v , a única forma de f não satisfazer a definição 44 é haver mais de uma aresta e , incidente em v , com $f(e) \leq f(v)$. Como pode ser visto na definição acima, o valor na função nos vértices ou é zero ou é o valor da menor aresta, que é único devido ao esquema de perturbação. Para cada aresta $e = \overline{v_1 v_2}$, existem duas formas de f não satisfazer 44: ou tem-se $f(v_1) \geq f(e)$ e $f(v_2) \geq f(e)$ ou $f(e)$ é maior ou igual ao valor da função nos dois triângulos que a compartilham. O primeiro caso é evitado pelo esquema de perturbação, que zera a função em um dos vértices, tornando-o singular. O segundo caso é evitado pela escolha do valor da função no triângulo como sendo o valor da aresta de maior comprimento ou, no caso da perturbação, um valor maior ainda. Para cada triângulo t , uma ambigüidade ocorreria se $f(t) \leq f(e)$ para mais de uma aresta $e \subset t$. Novamente, o esquema de perturbação impede que isso ocorra. ■

Um fato interessante derivado da definição acima é que os vértices de P ou são críticos (ver a definição 45) ou seu vetor gradiente é constituído de tal forma que o conjunto imagem de $\mathcal{V}$,

Figura 5.1: Construção da função de Morse em um complexo simplicial formado por dois triângulos, cinco arestas e quatro vértices

denotada por $Im(\mathcal{V})$, é o conjunto das “menores arestas” em DT . Embora essenciais, estas arestas não são suficientes para garantir a reconstrução correta do objeto original, de forma que, além da $Im(\mathcal{V})$, serão usadas algumas arestas críticas, como mostrado no seguinte algoritmo:

Algoritmo 6 Tor (Topological Object Reconstructor) versão bidimensional

(Para um conjunto P de amostras)

1. Calcule-se a triangulação de Delaunay de P .
2. Defina-se a função de Morse discreta f como foi descrita anteriormente.
3. Calcule-se $Im(\mathcal{V})$.

Para cada amostra p contida em apenas uma aresta de $Im(\mathcal{V})$ Faça

4. Calcule-se a menor aresta crítica incidente em p . Seja E_c o conjunto de todas estas arestas.

Fim do Para

5. Retorne-se $G = E_c \cup Im(\mathcal{V})$
-

O conjunto de arestas G retornado pelo algoritmo 6 é a reconstrução poligonal da curva suave C . Assumiu-se que a curva suave original C é uma curva fechada simples (sem auto-intersecções). A aproximação poligonal G também deve ser simples e fechada.

A figura 5.2 mostra o resultado do algoritmo. A figura 5.2a) apresenta o conjunto de arestas regulares, isto é, as arestas em $Im(\mathcal{V})$. As arestas críticas que completam a reconstrução são mostradas nas figuras 5.2b) e a reconstrução completa, na figura 5.2c).

Figura 5.2: Reconstrução de um pato. a) arestas na $Im(\mathcal{V})$; b) arestas críticas; c) Curva final.

Observe-se que este procedimento não é equivalente a tomar as duas menores arestas incidentes em cada amostra. Na figura 5.3, onde as flechas representam o campo vetorial em cada aresta, pode ser visto que a segunda menor aresta incidente à amostra p não faz parte da reconstrução e não é capturada pelo algoritmo porque não é uma aresta crítica.

A complexidade computacional do algoritmo 6 é dominada pela triangulação de Delaunay, uma vez que a função de Morse discreta e os elementos críticos e regulares podem ser computados em $\Theta(n)$, onde n é o número de amostras.

Figura 5.3: Arestas incidentes a uma amostra

A seguir será discutida a demonstração de que, sob uma taxa de amostragem adequada, G é, na verdade, uma correta reconstrução da curva original C .

5.3 Garantias Teóricas

Com o objetivo de garantir a correta reconstrução poligonal, assume-se que o conjunto de amostras P é uma r -amostragem de uma curva fechada simples C .

É iniciada a seção estabelecendo três resultados demonstrados em Amenta et al. (1998a).

Lema 6 (Amenta et al., 1998a) *Seja C uma curva fechada suave em $\mathbb{R}^2$. Qualquer disco B que intersecta C em pelo menos duas componentes conexas contém pelo menos um ponto do eixo medial de C .*

Lema 7 (Amenta et al., 1998a) *Seja C uma curva suave e r -amostrada em $\mathbb{R}^2$ com $r < 1$. O ângulo entre três amostras adjacentes é pelo menos $\pi - 4 \arcsin(r/2)$.*

Teorema 17 (Amenta et al., 1998a) *A distância de um ponto p em uma curva suave r -amostrada C até a reconstrução poligonal de C é no máximo $(\frac{r^2}{2})lfs(p)$.*

O lema seguinte apresenta uma importante propriedade que relaciona a distância entre amostras e seus respectivos lfs .

Lema 8 $lfs(p) \leq lfs(q) + l(\overline{pq})$ para quaisquer dois pontos p e q em P .

Demonstração. Considerando-se $\tilde{q}$ o ponto do eixo medial tal que $lfs(q) = l(q\tilde{q})$, segue-se que $lfs(p) \leq l(\overline{p\tilde{q}}) \leq lfs(q) + l(\overline{pq})$. ■

Lema 9 *Seja $e = \overline{ab}$ uma aresta conectando duas amostras adjacentes a e b . Se $r < 1$, então $l(e) \leq \frac{2r}{1-r} lfs(a)$.*

Demonstração. O bissetor perpendicular de e intersecta C em pelo menos dois pontos. Seja p o ponto de intersecção contido no arco de C definido por $a, p, e b$ e que não contém nenhum outro ponto de P além de a e de b .

Seja B o disco centrado em p com raio $l(\overline{pa}) = l(\overline{pb})$. B intersecta C ou em apenas uma componente conexa ou em mais de uma componente. Se ocorrer a intersecção em mais de uma componente, pelo lema 6, o raio de B deve ser maior do que o $lfs(p)$, o que implica $l(\overline{pa}) \geq lfs(p)$. Isto contraria a condição de r -amostragem. Assim, $B \cap C$ gera apenas um arco de C . Pelo lema 8, tem-se $lfs(p) \leq lfs(a) + l(\overline{pa}) \leq lfs(a) + r \times lfs(p)$, então $lfs(p) \leq \frac{lfs(a)}{1-r}$. O lema segue desde que $l(e) \leq 2r \times lfs(p)$. ■

Proposição 1 *Seja $e = \overline{ab}$ uma aresta de DT tal que b é a amostra mais próxima adjacente a a . Se $r < \frac{1}{3}$, então $e \in Im(\mathcal{V})$.*

Demonstração. Seja p o ponto como na demonstração do lema 9. Sem perda de generalidade, pode-se assumir $lfs(p) = 1$. O teorema 17 assegura que a distância de p até e é, no máximo, $\frac{r^2}{2}$. Dessa forma, $l(e) \leq 2r\sqrt{1 - \frac{r^2}{4}}$. A bola B' centrada em a com raio $l(e)$ não contém pontos de P em seu interior além de a e b . De fato, se existir uma amostra c no interior de B' , ou c é adjacente a b , ou c pertence a um arco de C intersectado por B' . No primeiro caso, o ângulo definido por abc é menor que $\pi/2$, contrariando o lema 7. Se c pertence a um arco de C , pelo lema 6, B' contém pelo menos um ponto do eixo medial de C em seu interior. Note-se que a distância de p até a é menor do que $r \times lfs(p) = r$ e que $r + l(e) = r + 2r\sqrt{1 - \frac{r^2}{4}} < 1$. Dessa forma, B' está contida na bola unitária centrada em p (figure 5.4), que não contém pontos do eixo medial em seu interior. Então, e é a menor aresta incidente em a , assegurando que $e \in Im(\mathcal{V})$. ■

A proposição acima é extremamente importante, uma vez que ela assegura que cada amostra se conecta, através de seu campo vetorial, com sua amostra adjacente mais próxima. Além disso, a proposição garante também que toda a amostra está contida em pelo menos uma aresta na $Im(\mathcal{V})$.

A próxima proposição é o ingrediente final para assegurar a reconstrução correta fornecida pelo algoritmo proposto neste capítulo.

Demonstração. Suponha-se que $e \notin Im(V)$ não é uma aresta crítica, isto é, e é a maior aresta de algum triângulo $t \in DT$. Seja c um vértice de t que não é incidente em e , e seja B o circuncírculo de t (figure 5.5). Pelo lema 7, sabe-se que c não é adjacente nem a a nem a b . Então, B intersecta C em mais de uma componente conexa. Logo, pelo lema 6, B contém um ponto x do eixo medial de C em seu interior. Como $r < 1$, x deve estar na região de B entre e e c , dessa forma, $l(e) > lfs(a)$. Como $l(e) \leq \frac{2r}{1-r}lfs(a)$ (lema 9), tem-se $lfs(a) < l(e) \leq \frac{2r}{1-r}lfs(a)$ contrariando a hipótese de $r < \frac{1}{3}$.

Figura 5.5: Proposição 2

O teorema seguinte segue diretamente das proposições 1 e 2.

Teorema 18 *Se $r < \frac{1}{3}$, então as arestas de G conectam exatamente cada par de amostras adjacentes de P .*

É importante notar que a amostragem exigida pelo teorema 18 é melhor do que a de alguns algoritmos descritos na literatura.

5.4 Resultados

Nesta seção, são apresentados os resultados do nosso algoritmo quando aplicado em problemas práticos. Também são mostradas comparações com alguns resultados de outros métodos de reconstrução bidimensionais, a saber, Crust, β -skeleton, Power-Crust, e NN-Crust. Os algoritmos foram implementados em C++, e conjuntos de amostras de um mapa do Brasil e de um pato foram utilizados como entrada de dados durante as comparações. Os testes foram realizados em um processador Intel Pentium III de 1.0 Ghz de velocidade. Além disso, os tempos mostrados na tabela 5.1 desconsideram a primeira triangulação de Delaunay, já que ela é o passo inicial de todos os algoritmos comparados.

A figura 5.6 mostra um exemplo de um conjunto de amostras e sua reconstrução. Pode-se observar, neste exemplo, que o algoritmo Tor é capaz de capturar diferentes componentes conexas num mesmo conjunto de pontos.

Figura 5.6: Exemplo de reconstrução a) conjunto de pontos b) curva reconstruída

A figura 5.7 mostra a reconstrução de um pato obtida pelos quatro algoritmos. Pode ser observado que os algoritmos Tor, Power Crust e NN-Crust obtiveram reconstruções satisfatórias;

O Crust e o β -skeleton apresentaram falhas na reconstrução.

Figura 5.7: Reconstrução de um pato a) Tor b) Crust; c) Power-Crust; d) Beta-skeleton e) NN-Crust

Novamente, a figura 5.8 mostra os resultados obtidos com cada algoritmo. Os melhores resultados foram obtidos pelo nosso algoritmo e pelo Power-Crust, como pode ser visto nas figuras 5.8 b) e d) , respectivamente. A figura 5.8 c), mostra que o algoritmo Crust perde algumas arestas, gerando um conjunto de curvas abertas. O β -skeleton retorna uma reconstrução não satisfatória (figura 5.8e)) e, finalmente, o algoritmo NN-Crust produz algumas arestas que não fazem parte da reconstrução (ver figura 5.8 f)).

Figura 5.8: Reconstrução do mapa do Brasil: a) Conjunto de amostras; b) Tor; c) Crust; d) Power-Crust; e) Beta-skeleton f) NN-Crust

A tabela 5.1 apresenta os tempos de processamento gastos por cada algoritmo. Pode-se ver que o algoritmo aqui desenvolvido é por volta de quinze vezes mais rápido que o Power Crust, o único, além dele, a produzir uma reconstrução satisfatória do mapa do Brasil. Embora o β -skeleton e o NN-Crust tenham apresentado uma performance similar à do algoritmo Tor, ambos produziram arestas que não pertencem à reconstrução.

Outro fato importante que merece ser mencionado é a instabilidade numérica apresentada pelo Power-Crust. Perturbando alguns pontos próximos da ilha de Marajó (observe a figura 5.9a)), o

Tabela 5.1: Tempo de execução dos algoritmos bidimensionais.

Algoritmo	Tempo (s)
Tor	0.032
Power-Crust	0.484
Crust	0.359
β -Skeleton	0.032
NN-Crust	0.032

Power-Crust conectou o continente à ilha, como mostrado na figura 5.9b). Com a mesma perturbação, nosso algoritmo ainda produziu uma reconstrução correta, como na figura 5.8b).

Figura 5.9: Instabilidade numérica do Power-Crust

5.5 Considerações Finais

Os resultados, tanto teóricos quanto práticos, permitem concluir que a reconstrução utilizando funções de Morse discretas representa um avanço para a reconstrução de curvas a partir de nuvem de pontos, uma vez que preserva as boas propriedades dos melhores algoritmos, a saber: garantias teóricas, robustez e performance computacional.

Além disso, os resultados bidimensionais se mostraram fundamentais para fornecer a intuição necessária para produzir um algoritmo de reconstrução tridimensional com o mesmo nível de eficiência e robustez do algoritmo bidimensional.

O capítulo seguinte apresentará em detalhes o algoritmo Tor na sua versão tridimensional, discutindo seus principais aspectos e apresentando resultados computacionais, garantias teóricas e comparações com os algoritmos mais clássicos da literatura.

Capítulo 6

Reconstrução de Superfícies

6.1 Considerações Iniciais

Os resultados obtidos no caso bidimensional, além de motivarem o desenvolvimento da extensão para o caso tridimensional do algoritmo Tor, ofereceram a intuição necessária a tal extensão.

É apresentado, neste capítulo, o algoritmo Tor em sua versão tridimensional. Assim como no caso bidimensional, ele se baseia na análise do campo gradiente discreto de uma determinada função de Morse definida sobre a triangulação de Delaunay das amostras.

A maior complexidade do caso tridimensional colocou este pesquisador diante de problemas bastante atuais da área de reconstrução a partir nuvem de pontos, o que lhe permitiu entrar em contato direto com os pesquisadores de destaque deste campo do conhecimento.

6.2 Preliminares

Vários problemas foram encontrados durante a extensão do algoritmo Tor para o caso tridimensional. O principal deles foi definir, de forma adequada, a função de Morse discreta para o caso de complexos simpliciais tridimensionais. Algumas tentativas de definir a função a partir dos triângulos foram testadas utilizando-se como primitivas o perímetro, área e perímetro/razão de aspecto. Na versão final, que será apresentada a seguir, a função de Morse é definida a partir do circunraio dos triângulos.

Um ponto essencial apontado em vários algoritmos da literatura (Amenta et al., 2001b; Amenta e Choi, 1999; Amenta et al., 1998a; Dey e Goswami, 2003) é a utilização de uma triangulação de Delaunay robusta e confiável. Optou-se por utilizar a triangulação de Delaunay implementada na biblioteca CGAL (2004). A biblioteca CGAL é um conjunto de classes escritas em C++ para

algoritmos geométricos. Seu objetivo é fornecer estruturas de dados e algoritmos geométricos robustos, confiáveis e fáceis de usar. Uma de suas ferramentas mais utilizadas é a classe que inclui a triangulação de Delaunay. Esta implementação se beneficia de várias características presentes nas triangulações encontradas nas classes da biblioteca CGAL, entre elas estão:

- Funções versáteis de acesso às faces da triangulação na forma de *iterators* e *circulators*;
- Mecanismos flexíveis para adicionar atributos aos vértices e aos tetraedros da triangulação;
- Escolha de representações numéricas mais confiáveis do que a representação de ponto flutuante.

Uma característica que não se encontrou nas classes da biblioteca CGAL e que foi fundamental para o funcionamento do algoritmo Tor é a capacidade de adicionar atributos às faces e às arestas de uma triangulação. Para resolver este problema, implementou-se uma adaptação da solução encontrada no trabalho de Giesen e John (2002a).

Outro problema que merece ser mencionado é o de saber quais arestas considerar para a reconstrução. Observe-se que, no caso bidimensional, o algoritmo captura as arestas a partir dos vértices (amostras). Uma extensão natural para o caso tridimensional seria capturar as faces a partir das arestas. Entretanto, a questão principal é como identificar as arestas que fazem parte da reconstrução, levando-se em conta que agora tem-se um complexo simplicial de três dimensões, com vértices, arestas, triângulos e tetraedros. Giesen e John (2002b), em seu trabalho, utilizam as arestas do grafo de Gabriel como aproximação inicial, mas isso representa um cálculo geométrico que poderia aumentar o custo do algoritmo aqui proposto. A solução foi partir de um conjunto inicial de arestas, as quais sabe-se pertencer a reconstrução da superfície e, em seguida, ir adicionando arestas a este conjunto à medida que são selecionadas as faces pertencentes à reconstrução.

6.3 Algoritmo

Descreve-se, a seguir, o algoritmo desenvolvido para o processo de reconstrução de superfícies em $\mathbb{R}^3$. Exatamente como no algoritmo bidimensional, a idéia deste algoritmo é construir uma aproximação linear por partes de uma determinada superfície S , a partir da análise do campo gradiente discreto de uma específica função de Morse definida sobre a triangulação de Delaunay das amostras.

Antes de apresentar o algoritmo propriamente dito, definir-se-á formalmente a função de Morse utilizada.

Sejam P um conjunto de amostras de uma superfície fechada e suave $S \subset \mathbb{R}^3$, DT o complexo simplicial formado pela triangulação de Delaunay de P , e $r_c(t)$ o circunraio de um triângulo $t \in DT$. Sejam também V, E, T e C o conjunto de vértices, arestas, triângulos e células (tetraedros) de DT , respectivamente. Seja ainda $\mathcal{V} : C_2(DT) \rightarrow C_3(DT)$ a função que a cada triângulo associa o seu vetor gradiente.

Assim como no caso bidimensional, a função de Morse discreta $f : DT \rightarrow \mathbb{R}$ será construída de forma recursiva juntamente com o campo gradiente nas arestas e nas faces, de maneira a evitar ambigüidades. Considere-se $f : DT \rightarrow \mathbb{R}$ a seguinte função:

1. para cada célula $c \in C$, $f(c) = \max \{f(t) \mid t \subset c\}$;
2. para cada triângulo $t \in T$, $f(t) = r_c(t)$;
3. para cada aresta $e \in E$, $f(e) = \min \{f(t) \mid e \subset t\}$;
4. para cada vértice $v \in V$, $f(v) = -1$.

De maneira semelhante à versão bidimensional, a função pode não satisfazer a definição de função de Morse discreta devido a possíveis ambigüidades. Para poder evitá-las, também foi introduzido o seguinte esquema de perturbações:

Suponha-se que a função f já esteja definida nos triângulos $t \in T$ como $f(t) = r_c(t)$ e nos vértices $v \in V$ como $f(v) = -1$. Seja $e \in E$ uma aresta de DT , $\alpha = \min \{r_c(t) \mid e \subset t; t \in T\}$; e seja $T_e = \{t \mid e \subset t \text{ e } r_c(t) = \alpha\}$ o conjunto de triângulos de menor circunraio contendo e . Duas situações são possíveis: ou T_e possui um único elemento t , ou possui mais de um elemento. No primeiro caso, se t ainda não é vetor gradiente de nenhuma de suas arestas, $f(e) = f(t)$ e t é marcado como vetor gradiente de e . Caso contrário, $f(e) = 0$ (e neste caso e é ponto crítico de f). No segundo caso, em que T_e possui mais de um elemento, um triângulo t é arbitrariamente escolhido a partir de T_e . Como no primeiro caso, se t já é vetor gradiente de alguma de suas arestas, então $f(e) = 0$. Caso contrário, $f(t)$ é redefinida para $f(t) = r_c(t) - \varepsilon$ e $f(e) = f(t)$, onde $\varepsilon > 0$ é escolhido de maneira a assegurar que o valor de f em t continue maior ou igual ao valor de f nas arestas de t .

Existe um tipo similar de ambigüidade nos tetraedros de DT que pode ser resolvido com um esquema de perturbação equivalente.

Observa-se que a função f é definida nos vértices apenas por uma questão de rigor matemático, uma vez que os valores da função nos vértices não são usados durante o processo de reconstrução.

Teorema 19 *A função construída acima, juntamente com o esquema de perturbação aplicado é uma função de Morse discreta.*

Demonstração. É necessário mostrar que f satisfaz a definição 44 para os vértices, arestas, triângulos e tetraedros de DT . É trivial verificar que, segundo a definição de f , todos os vértices são singulares. Suponha-se que $f(e)$ não satisfaz a definição 44 para alguma aresta e que não pode ser vetor gradiente de nenhum de seus vértices. A única alternativa é que existam dois triângulos nos quais o valor de f é menor ou igual ao valor de f em e , mas o esquema de perturbação assegura que, no caso de e não ser ponto crítico de f , uma única face é escolhida como seu vetor gradiente. Se algum triângulo t não satisfaz a definição 44, três situações são possíveis: ou t é vetor gradiente para mais que uma de suas arestas, ou existem dois tetraedros como vetores gradiente de t , ou, ainda, t é vetor gradiente de uma de suas arestas e existe um tetraedro como seu vetor gradiente. O esquema de perturbação assegura que o primeiro caso não pode acontecer. O fato de t ter dois vetores gradientes nele definidos implica que t deve ser a face de maior circunraio para os dois tetraedros que o compartilham. Como este esquema de perturbação escolhe apenas um tetraedro para ser vetor gradiente de t , esse caso também não pode ocorrer. Finalmente, seja e e c , respectivamente, a aresta da qual t é vetor gradiente e o tetraedro que é vetor gradiente de t . Isso implica que c tem duas faces de mesmo circunraio] contendo e . Neste caso, o esquema de perturbação teria diminuído o valor de f em uma delas. Portanto, esse caso também não pode acontecer. O argumento para mostrar que f respeita a definição 44 nos tetraedros é semelhante ao utilizado nas arestas. ■

Uma vez definida a função de Morse discreta, pode-se derivar a versão tridimensional do algoritmo Tor (algoritmo 7). A seguir este algoritmo será descrito e detalhes de sua implementação serão comentados.

Algoritmo 7 Tor (Topological Object Reconstructor) versão tridimensional(Para um conjunto P de amostras.)

1. Calcule-se a triangulação de Delaunay de P .
2. $N := \emptyset$; $E_p := \emptyset$
3. Defina-se a função de Morse discreta f como foi descrita acima.
- Para cada amostra $p \in P$. Faça
 4. Calcule-se a menor aresta e incidente em p e adicione-se e ao conjunto E_p .
- Fim do Para
- Enquanto houverem arestas livres não visitadas em E_p Faça
 - Para cada aresta livre e não visitada em E_p Faça
 5. Marque a aresta e como visitada.
 - Se existir t , vetor gradiente de e , Então
 6.
$$\left(\begin{array}{l} N := N \cup \{t\} \\ \text{Adicionem-se à } E_p \text{ as arestas livres de } t \text{ que não estavam em } E_p. \end{array} \right)$$
 - Caso Contrário
 7.
$$\left(\begin{array}{l} \text{Seja } t' \text{ a face de menor circunraio incidente em } e, \text{ tal que } t' \notin N \text{ e } \mathcal{V}(t') = 0. \\ N := N \cup \{t'\}. \\ \text{Adicionem-se à } E_p \text{ as arestas livres de } t' \text{ que não estavam em } E_p. \end{array} \right)$$
 - Fim do Se
- Fim do Para
- Fim do Enquanto
8. Retorne-se N .

No algoritmo acima, N é o conjunto de faces que perfaz a aproximação linear por partes da superfície S . E_p armazena o conjunto de arestas ativas que guia o processo de reconstrução. Estão sendo chamadas de arestas “livres” aquelas incidentes em não mais que uma face do conjunto N . No passo 4, o algoritmo captura as menores arestas incidentes em cada amostra (observe-se que, neste momento, todas elas são livres). Dar-se-á uma demonstração de que essas arestas realmente pertencem à reconstrução do modelo. No passo 6, apenas as faces regulares são incluídas no modelo e, no passo 7, ambos os tipos de faces, críticas e regulares, podem ser incluídas.

É fácil observar que as faces incluídas na aproximação poliedral são as menores com relação ao circunraio. Será demonstrado que, para uma r -amostragem adequada, as faces com circunraio pequeno são aquelas que formam a reconstrução final do modelo. Vale a pena mencionar que a função de Morse faz o trabalho de converter informação geométrica em atributos topológicos.

Dada uma triangulação de Delaunay eficiente, a versão tridimensional do algoritmo Tor é de simples implementação. Ele não exige cálculo de normais, rotulação de células ou cálculo de pólos.

Outro problema que foi enfrentado na versão tridimensional do algoritmo Tor e que é comum em praticamente todos os algoritmos de reconstrução tridimensional baseados em esculpimento, é o aparecimento de tetraedros *sliver*. Um tetraedro *sliver* é um tetraedro com uma razão de aspecto

ruim e ainda com um circunraio razoavelmente pequeno (ver figura 6.1). Um tetraedro desse tipo é formado por quatro vértices quase igualmente espaçados na área do equador de sua circunsfera.

Figura 6.1: Tetraedro Sliver

O principal problema deste tipo de tetraedro é que ele possui as faces quase do mesmo tamanho e, por isso, é fácil tomar uma face errada como parte da reconstrução. Dey e Goswami (2002b), em seu trabalho, argumentam que vários algoritmos de reconstrução, dentre eles o Crust (Amenta et al., 1998a) e o Cocone (Amenta e Choi, 1999), selecionam um conjunto de faces que são as “candidatas” à reconstrução e, em seguida, executam uma etapa de pós-processamento para “extrair” uma variedade. Este pós-processamento, que também é detalhadamente discutido no artigo de Dey et al. (2001), trata exatamente do problema dos tetraedros sliver. A figura 6.2 ilustra o aparecimento de triângulos mal posicionados devido à presença de tetraedros slivers. Observe-se que, neste caso, aparecem arestas compartilhadas por mais de dois triângulos, fato que não pode ocorrer numa dois variedade.

Figura 6.2: Falha decorrente de tetraedros slivers.

A solução adotada neste trabalho foi evitar que ele aconteça. Ao se adicionar uma face no conjunto de saída, um teste verifica se acontece alguma situação semelhante a da figura 6.2. Caso

isso aconteça, a face não é adicionada.

Assim como no algoritmo bidimensional, o algoritmo tridimensional captura dois tipos de faces. O primeiro são as faces que pertencem ao campo de vetores gradientes nas arestas da triangulação de Delaunay, isto é, pontos regulares da função de Morse; o segundo tipo, por sua vez, é composto por faces que são pontos críticos da função de Morse. A figura 6.3 ilustra ambos os tipos de faces capturadas pelo algoritmo Tor.

Figura 6.3: Reconstrução de um cavalo. (a) apenas as faces pertencentes ao campo de vetores gradiente; (b) pontos críticos da função de Morse; (c) o modelo completo.

6.4 Resultados Computacionais.

São apresentados aqui alguns exemplos práticos do algoritmo Tor em funcionamento. As figuras 6.4 e 6.5 apresentam dois modelos reconstruídos pelo algoritmo Tor e iluminados com um software de Ray-tracing, e as figuras 6.4 (b) e (c) dão uma idéia da qualidade da malha gerada pela reconstrução dos modelos.

Foram realizadas comparações com alguns algoritmos encontrados na literatura, a saber, o Crust (Amenta et al., 1998a), o Power Crust (Amenta et al., 2001b), o Cocone (Amenta e Choi, 1999) e o Tight Cocone (Dey e Goswami, 2003). Todos os algoritmos foram implementados em C++ e

Figura 6.4: Modelo de um dragão reconstruído com o algoritmo Tor. (a) Textura; (b) Textura e malha; (c) Zoon.

utilizam a triangulação de Delaunay da biblioteca CGAL. Os algoritmos Crust e Power Crust foram implementados pelo aluno João Paulo Gois, como parte de seu projeto de mestrado, e os algoritmos Cocone e Tight Cocone foram gentilmente cedidos pelo pesquisador Tamal Dey. O primeiro teste foi feito empregando-se um conjunto de pontos que constitui uma amostragem pobre de um objeto, no caso um coelho com 1000 pontos. Este teste mostrou que, para este exemplo, o algoritmo aqui desenvolvido teve um desempenho semelhante ao do Cust e Power Crust, e ligeiramente superior ao Cocone e ao Tight Cocone, uma vez que estes últimos desconectaram partes do modelo, como pode ser visto na figura 6.6.

A segunda comparação foi a velocidade de processamento. Escolheram-se alguns modelos com maior quantidade de pontos e mediu-se o tempo de processamento de cada algoritmo. Descontou-

Figura 6.5: Reconstrução do modelo de uma mão com aplicação de textura.

se o tempo da triangulação de Delaunay, já que ela é passo inicial de todos os algoritmos citados. Os testes de velocidade foram executados em um processador dual Athlon 2200+ com 1Gb de memória RAM. As comparações de velocidade são mostradas nas tabelas 6.1 e 6.2, cujas figuras foram geradas pelo algoritmo Tor.

	9.697 pts	13.321 pts	35.947 pts	54.707 pts
Crust	23,47	40,75	85.83	160,88
Power Crust	21,59	31,74	82.73	129,86
Cocone	7,27	10,35	33,18	49,91
Tight Cocone	5,11	7,86	20,29	35,09
Tor	1,33	1,93	5,59	10,01

Tabela 6.1: Tempo de processamento, em segundos, para a obtenção dos modelos mostrados.

Observamos que o algoritmo Tor se mostrou mais rápido (aproximadamente três vezes mais rápido que o Tight Cocone, que é o mais veloz deles) e apresentou reconstruções satisfatórias dos objetos testados.

Adicionalmente, a figura 6.7 mostra as reconstruções de um modelo da nave estelar Enterprise obtidas pelos algoritmos Crust, Cocone, Tight Cocone e Tor. Pode-se observar que faces impróprias

Figura 6.6: Reconstrução do Coelho a) Nosso algoritmo; b) Crust; c) Power Crust; d) Cocone; e) Tight Cocone

	54.366 pts	109.108 pts	172.974 pts	921.085 pts
Crust	145,75	534,41	536,21	Sem resultado
Power Crust	125,89	268,55	458,46	Sem resultado
Cocone	46,16	85,14	246,38	1286.01
Tight Cocone	33,68	66,75	112,49	1354,42
Tor	10,54	17,1	29,39	446,21

Tabela 6.2: Tempo de processamento, em segundos, para a obtenção dos modelos mostrados.

foram adicionadas aos modelos obtidos pelos algoritmos Crust e Cocone, muito provavelmente em consequência de instabilidades numéricas geradas pelo cálculo do diagrama de Voronoi. Problemas similares afetaram a rotulação de células do Tight Cocone, resultando numa reconstrução insatisfatória. O algoritmo TOR produziu uma melhor aproximação para o modelo, embora problemas localizados nas regiões próximas de onde a superfície não é suave podem ser observados. Entretanto, este é um bom exemplo da robustez proporcionada por uma abordagem topológica.

Executaram-se ainda, testes adicionando-se, de forma aleatória, ruídos aos conjunto de pontos.

Figura 6.7: Modelo da Enterprise reconstruído por: (a) Crust; (b) Cocone; (c) Tight Cocone; (d) Tor.

A tabela 6.3 mostra um determinado conjunto de pontos com três níveis de ruídos e em ordem crescente de perturbação. Na sequência, as tabelas 6.4, 6.5, 6.6, 6.7 e 6.8 mostram os modelos perturbados reconstruídos pelos algoritmos Tor, Crust, Power Crust, Cocone e Tight Cocone, respectivamente.

Tabela 6.3: Pontos com adição artificial de ruídos.

Tabela 6.4: Reconstrução dos modelos com ruídos retornada pelo algoritmo Tor (Topological Object Reconstructor).

Tabela 6.5: Reconstrução dos modelos com ruídos retornada pelo algoritmo Crust.

Tabela 6.6: Reconstrução dos modelos com ruídos retornada pelo algoritmo Power Crust.

Pode-se observar, nesta sequência de testes com adição de ruídos, que todos os algoritmos testados apresentaram resultados semelhantes, com a ressalva de que a implementação do Power Crust produzida no trabalho de Gois (2004) levou uma pequena desvantagem, apresentando uma superfície rugosa já no primeiro nível de perturbação. Além disso, o Tight Cocone não produziu nenhuma saída para o último nível de perturbação. Vale lembrar que nenhum desses algoritmos foi projetado especificamente para o tratamento de ruídos.

Tabela 6.7: Reconstrução dos modelos com ruídos retornada pelo algoritmo Cocone.

Tabela 6.8: Reconstrução dos modelos com ruídos retornada pelo algoritmo Tight Cocone.

Acredita-se que funções de Morse discretas possam ser bastante úteis no tratamento de conjuntos com ruídos e, por isso, talvez o próximo passo na evolução do algoritmo Tor seja adaptá-lo para identificar e eliminar ruídos.

O algoritmo Tor também demonstrou robustez no tratamento de conjuntos com grandes quantidades de pontos, chegando próximo da casa de um milhão de pontos sem apresentar problemas. A figura 6.8 mostra dois exemplos da reconstrução destes modelos.

6.5 Garantias Teóricas

Serão apresentadas, nesta seção, as demonstrações matemáticas de que o algoritmo Tor retorna, sob uma taxa de amostragem adequada, uma reconstrução homeomorfa e próxima à da superfície original.

Assumiu-se que $P \subset \mathbb{R}^3$ é uma r -amostragem de uma superfície S e que P está em posição geral. Será mostrado que o espaço gerado pelas faces capturadas por este algoritmo é homeomorfo e próximo à superfície S .

Em duas dimensões, fica claro que a “resposta certa” para o problema de reconstrução é

Figura 6.8: Modelos com grandes quantidades de pontos. a) leão 636.646 pontos: b) Lucy 921.085 pontos

uma curva linear por partes conectando pontos adjacentes ao longo da curva onde as amostras foram tomadas. A generalização dessa ideia para o caso tridimensional não é imediatamente óbvia. Será usada a definição de conjunto “correto” de triângulos encontrada em Chew (1993) e em Edelsbrunner e Shah (1994). A ideia é considerar os triângulos da triangulação de Delaunay, que são duais a arestas do diagrama de Voronoi que intersectam a superfície S . Os triângulos deste conjunto são chamados de *triângulos de Delaunay de superfície* ou simplesmente de triângulos *Delaunay restritos*.

Formalmente, sejam D_P e V_P a triangulação de Delaunay e o diagrama de Voronoi de P , respectivamente. Considere-se a intersecção de V_P com a superfície S . Esta intersecção define uma partição de S em regiões. Essa decomposição é chamada de *diagrama de Voronoi restrito* de P em S e é denotada por $V_{P,S}$. O dual do diagrama de Voronoi restrito define a *triangulação de Delaunay restrita* e é denotado por $D_{P,S}$. Seja $p \in P$. A intersecção da célula de Voronoi de p com a superfície S é denotada por $V_{p,S}$. Mais especificamente, uma aresta pq está em $D_{P,S}$ se, e somente se, $V_{p,S} \cap V_{q,S} \neq \emptyset$; um triângulo pqr está em $D_{P,S}$ se, e somente se, $V_{p,S} \cap V_{q,S} \cap V_{r,S} \neq \emptyset$. Assumindo-se a hipótese de posição geral, não existem tetraedros em $D_{P,S}$.

Edelsbrunner e Shah mostraram que a superfície gerada por $D_{P,S}$ é homeomorfa a S se vale a propriedade das “bolas fechadas”. Se cada $V_{p,S}$ é uma bola bidimensional, cada $V_{p,s} \cap V_{q,s}$ é

uma bola unidimensional e cada $V_{p,S} \cap V_{q,S} \cap V_{r,S}$ é um único ponto, isto é, uma bola fechada de dimensão zero. Posteriormente, Amenta e Bern (1998) usaram este resultado para mostrar que se P é uma r -amostragem de S com $r < 0.1$, $V_{P,S}$ satisfaz esta propriedade e, portanto, D_P contém todos os triângulos necessários para uma reconstrução correta da superfície S .

Os triângulos Delaunay restritos possuem certas propriedades específicas, como o circunraio pequeno. Todo triângulo Delaunay restrito t possui circunraio medindo $O(r) lfs(p)$, onde p é um vértice de t . Além disso, são triângulos “chatos”, isto é, a normal de cada triângulo t forma um ângulo pequeno, $O(r)$, com a normal da superfície no vértice p , em que p é o vértice de t com maior ângulo interno. Amenta e Bern mostraram que triângulos “pequenos” são sempre “chatos”.

Considere-se S uma superfície suave sem auto-intersecções, P uma r -amostragem de S e $r < 0.1$.

Lema 10 (Amenta e Choi, 1999) Para quaisquer dois pontos p e q em S com $d(p, q) \leq \rho \min\{lfs(p), lfs(q)\}$ para $\rho \leq \frac{1}{3}$, o ângulo entre as normais a S nos pontos p e q é, no máximo, $\frac{\rho}{1-3\rho}$.

Lema 11 (Amenta e Choi, 1999) Seja y um ponto qualquer em V_p tal que $d(p, y) \geq \delta lfs(p)$ para $\delta > \frac{r}{1-r}$. O ângulo entre $\vec{y}$ e n_p , onde $\vec{y}$ é o vetor $\overrightarrow{py}$ e n_p é o vetor normal à S em p , é menor que $\arcseno\left(\frac{r}{\delta(1-r)}\right) + \arcsen\left(\frac{r}{1-r}\right)$.

Corolário 3 (Amenta e Choi, 1999) Para qualquer y em V_p tal que o ângulo entre $\vec{y}$ e n_p é maior que $\arcseno\left(\frac{r}{\delta(1-r)}\right) + \arcsen\left(\frac{r}{1-r}\right)$ a $d(y, p) \leq \delta lfs(p)$.

Estes dois últimos resultados dão a idéia de que, se a superfície estiver bem amostrada, a célula de Voronoi tridimensional de uma amostra s deve ser “fina” e “comprida”.

Teorema 20 (Amenta e Choi, 1999) Seja $r(t)$ o circunraio de um triângulo Delaunay restrito t . Para qualquer vértice p de t , $r(t) \leq \frac{r}{1-r} lfs(p)$.

Um resultado imprescindível nesta demonstração é de que as faces capturadas pelo algoritmo Tor tenham circunraio pequeno. Observe-se que ele captura dois tipos de faces, faces pertencentes ao campo gradiente nas arestas e faces com vetor gradiente nulo. Cada tipo será analisado separadamente.

Dado um vértice p em um complexo simplicial, uma vizinhança de p é um conjunto de simplexos incidentes a p . Observe-se que, quando $r < 0.1$, existe um guarda-chuva U_p (ver definição 37) que aproxima localmente a superfície S em p .

Lema 12 *Seja p uma amostra e q sua amostra mais próxima. Para $r < 0.1$ a aresta conectando p e q é Delaunay restrita.*

Demonstração. Suponha-se, por absurdo, que $e = \overline{pq}$ não seja Delaunay restrita. Como $r < 0.1$, existe uma superfície N linear por partes composta apenas por triângulos Delaunay restritos e homeomorfa à superfície S . Existe, ainda, um guarda-chuva U_p , composto por triângulos Delaunay restritos, que é homeomorfo a uma vizinhança de p em S (ver figura 6.9). Como e não é Delaunay restrita, ela não pertence a nenhum triângulo de U_p ; no entanto, todas as arestas que contêm p e pertencem aos triângulos de U_p têm comprimento de, no máximo, $\frac{2r}{1-r}lfs(p)$, pois este é limitado pelo circunraio de cada triângulo. Como e é a menor aresta que contém p , seu comprimento também é menor que $\frac{2r}{1-r}lfs(p)$ e, além disso, ela não conecta p com nenhum de suas amostras adjacentes sobre S . Considere-se B a bola que tem e como seu diâmetro. B toca a superfície S em pelo menos duas componentes conexas, portanto contém pontos do eixo medial, e além disso, o raio r_B de B é, no máximo, $\frac{r}{1-r}lfs(p)$. Então, tem-se $lfs(p) \leq 2r_B \leq \frac{2r}{1-r}lfs(p)$, portanto $lfs(p)(1 - \frac{2r}{1-r}) \leq 0$, o que é uma contradição, desde que $r \leq 0.1$. ■

Figura 6.9: Vizinhança de uma amostra.

Lema 13 *Seja e uma aresta Delaunay restrita e $\mathcal{V}(e)$, se existir, o seu vetor gradiente. Então, $\mathcal{V}(e)$ é um triângulo com circunraio menor ou igual que $\frac{r}{1-r}lfs(p)$ onde p é um vértice qualquer de e .*

Demonstração. Note-se que, para $r < 0.1$, e está contida em dois triângulos Delaunay restritos, e portanto com circunraio menor ou igual que $\frac{r}{1-r}lfs(p)$. Se $\mathcal{V}(e)$ não for nenhum deles, terá circunraio menor do que o de ambos. ■

Lema 14 *Seja e uma aresta Delaunay restrita incidente em uma única face $t \in N$, de tal forma que t é campo vetorial de uma de suas arestas. Então, a face t' , capturada no passo 7 do algoritmo Tor, tem circunraio menor ou igual que $\frac{r}{1-r}lfs(p)$ onde p é um vértice de e .*

Demonstração. Suponha-se, por absurdo, que $r_c(t') > \frac{r}{1-r}lfs(p)$. Sem perda de generalidade, pode-se supor que t' é a primeira face escolhida pelo algoritmo com essa propriedade. Sejam p e q os vértices de e , s o vértice de t' oposto a e e U_p , como na demonstração do lema 12. Como t' não está em U_p (teorema 20) ou q ou s (ou ambos) não estão em U_p , o vértice q tem que estar em U_p ; caso contrário, a condição de r -amostragem seria violada. Logo $e \in U_p$. Pelo teorema 20, existe uma face t_1 com $r_c(t_1) \leq \frac{r}{1-r}lfs(p)$. $\mathcal{V}(t_1) = c_1$ deve ser não nulo; caso contrário, ela seria capturada pelo algoritmo. Neste caso, haveria uma face t_2 , também compartilhando e , com $r_c(t_2) < r_c(t_1)$. Neste caso, $\mathcal{V}(t_2) = c_2$ também é não nulo. Seguindo esse raciocínio seria encontrada uma face t_i com $\mathcal{V}(t_i) = 0$ e, ainda, $r_c(t_i) < r_c(t')$ o que leva a uma contradição. ■

Os dois resultados acima mostram que os dois tipos de faces capturadas pelo algoritmo Tor têm circunraio pequeno. Isto é um fator essencial no restante das demonstrações.

O próximo resultado, de Amenta e Bern, demonstra que triângulos “pequenos” são “chatos”.

Teorema 21 (Amenta et al., 2002b) *Seja p o vértice incidente no maior ângulo de um triângulo t e n_p a normal a S em p . Se t possui circunraio menor que $\frac{r}{1-r}lfs(p)$, então seu vetor normal forma um ângulo menor que $\alpha + \arcsen\left(\frac{2}{\sqrt{3}}\sin(2\alpha)\right)$ com n_p , em que $\alpha \leq \arcsen\left(\frac{r}{1-r}\right)$.*

Mostrar-se-ão um homeomorfismo entre S e uma variedade linear por partes N formada pelos triângulos “candidatos” capturados pelo algoritmo Tor. O homeomorfismo é definido explicitamente por meio de uma função. Primeiro é necessário definir uma aplicação $\mu : \mathbb{R}^3 \rightarrow S$, em seguida considera-se sua restrição à variedade N .

Seja $\mu : \mathbb{R}^3 \rightarrow S$ uma aplicação que leva cada ponto $p \in \mathbb{R}^3$ no ponto mais próximo de S . A restrição de μ à variedade N será denotada por μ_N .

Lema 15 *A restrição de μ à superfície N é uma função contínua e bem definida*

Demonstração. As descontinuidades de μ , como aplicação de $\mathbb{R}^3$ em S , estão exatamente nos pontos do eixo medial. Considere-se $q \in N$. Isso implica que q pertence a algum triângulo t de N . Considerando-se p um vértice de t , a distância de q até o ponto $x = \mu_N(q)$, denotada por

$d(q, x)$, é menor ou igual a $d(q, p)$; aliás, é menor ou igual à distância de q até qualquer vértice de t , uma vez que os vértices de t estão em S . Como o circunraio de t é menor ou igual a $\frac{r}{1-r}lfs(p)$, $d(q, p) \leq \frac{2r}{1-r}lfs(p)$ mas existe um vértice de t , diga-se s , cuja distância até q é menor ou igual a $\frac{r}{1-r}lfs(p)$; portanto $d(q, x) \leq \frac{r}{1-r}lfs(p)$, onde p é um vértice qualquer de t , para $r \leq 0.1$. Logo N não contém pontos do eixo medial. Como N evita pontos do eixo medial, μ_N é contínua em N .

■

O próximo passo é mostrar que μ_N é “bem comportada” sobre as amostras e que esse “bom comportamento” se estende para o interior de cada triângulo. Para isso será usado um resultado demonstrado em Amenta et al. (2002b)

Corolário 4 (Amenta et al., 2002b) *A função μ_N é injetiva sobre o conjunto P .*

Em seguida mostrar-se-á que, na verdade, μ_N é injetora em todo N . Antes disso, necessita-se de mais um resultado técnico.

Lema 16 *Seja q um ponto sobre um triângulo $t \in N$. O ângulo entre n_x , normal à superfície S em $x = \mu_N(q)$, e a normal n_t de t é no máximo 43.643° para $r \leq 0.063$.*

Demonstração. Seja p o vértice de t com maior ângulo interno. No pior caso, $d(x, p) \leq d(x, a) + d(p, a)$, em que a é o vértice de t mais próximo de x . Logo, $d(x, p) \leq r lfs(x) + \frac{2r}{1-r}lfs(a)$. Da propriedade Lipschitz de $lfs(\cdot)$, $d(x, p) \leq r lfs(x) + \frac{2r}{1-r}(lfs(x) + d(x, a))$, então temos $d(x, p) \leq r lfs(x) + \frac{2r}{1-r}(lfs(x) + r lfs(x)) \leq \left(r + \frac{2r+2r^2}{1-r}\right)lfs(x)$. Para $r \leq 0.063$, obtém-se $d(x, p) \leq 0.205 lfs(x)$; e $d(x, p) \leq 0.201 lfs(p)$; uma vez que $d(x, p) \leq \frac{3r}{1-r}lfs(p)$. Logo, $d(x, p) \leq \rho \min\{lfs(x), lfs(p)\}$, $\rho < 1/3$, o ângulo entre n_x e n_p é no máximo 30.876° (lema 10).

Como o ângulo interno em p é maior ou igual a $\pi/3$, e o circunraio de t é menor ou igual a $\frac{r}{1-r}(lfs(p))$, o teorema 21 assegura que o ângulo entre n_t e n_p é, no máximo, 12.768° (para $r < 0.063$). Portanto, o ângulo n_x e n_t é, no máximo, 43.643° . ■

Da mesma forma que no artigo de (Amenta et al., 2002b), vai-se assumir como hipótese que dois triângulos adjacentes se encontram em um vértice (ou aresta) em um ângulo maior que $\frac{\pi}{2}$. Define-se dois triângulos como adjacentes se eles compartilham pelo menos um vértice. Esta hipótese evita variedades com cantos excessivamente agudos.

A demonstração se segue em três passos. Vai-se mostrar, em primeiro lugar, que μ_N induz um homeomorfismo em cada triângulo; na sequência, em cada par de triângulos adjacentes e, finalmente, em N como um todo.

Lema 17 *Seja U uma região contida em um triângulo $t \in N$ ou em triângulos adjacentes de N . A função μ_N define um homeomorfismo entre U e $\mu_N(U) \subset S$.*

Demonstração. Sabe-se que μ_N é contínua e está bem definida em U ; portanto, basta mostrar que μ_N é injetora. Vai-se mostrar, em primeiro lugar, que μ_N é injetora em um triângulo t . Suponha-se que existam dois pontos, q e $y \in t$ com $\mu_N(y) = \mu_N(q)$. Os pontos q , $\mu_N(q)$ e y são colineares e estão na reta suporte do vetor normal a S em $\mu_N(q)$; caso contrário, haveria uma violação na condição de r -amostragem. O fato de t conter o segmento de reta entre q e y contradiz o lema 16.

Considere-se, agora, o caso em que U não está completamente contido em um único triângulo. Suponha-se, por absurdo, que existam dois pontos q e y em U tais que $\mu_N(y) = \mu_N(q) = x$ e seja v um vértice comum aos triângulos que contêm U . O fato de μ_N ser injetora em um único triângulo significa que q e y estão em triângulos distintos, diga-se t_q e t_y . Seja n_x o vetor normal em x . Considere-se l a reta suporte de n_x . Esta reta deve intersectar a região U pelo menos duas vezes. Se q e y não forem adjacentes ao longo de l , basta redefinir q para que isso seja verdade. Considere-se, agora, a orientação da região U de acordo com o pólo (vértice de Voronoi mais distante) em v . Ou l atravessa N de dentro para fora e volta para dentro, ou de fora para dentro e volta para fora, o que significa que os dois triângulos possuem orientações opostas.

Os ângulos entre as normais dos triângulos t_q , t_y e n_x são de, no máximo, 43.643° (lema 16), isto é, os triângulos são perfurados por n_x quase perpendicularmente. Uma vez que os dois triângulos possuem orientações opostas, o ângulo entre suas duas normais é de, pelo menos, $\pi - 87.286$, significando que t_q e t_y se encontram em um ângulo obtuso. Isso contraria a hipótese assumida anteriormente. ■

Termina-se esta demonstração utilizando-se um teorema da topologia.

Teorema 22 *A função μ_N define um homeomorfismo da triangulação N para a superfície S .*

Demonstração. Considere-se $\mu_N(N) = S' \subset S$. Vai-se, em primeiro lugar, mostrar que (N, μ_N) é um espaço de recobrimento para S' . De maneira informal, pode-se pensar que (N, μ_N)

Figura 6.10: Demonstração do lema 17

é um espaço de recobrimento de S' se μ_N leva N em S' de maneira suave sem nenhuma dobra ou outro tipo de singularidade. Detalhes sobre esses tipos de singularidades podem ser encontrados em (Massey, 1967), capítulo 5. Mostrar que (N, μ_N) é um espaço de recobrimento sobre S' é um resultado mais fraco do que mostrar que μ_N é um homeomorfismo. Um espaço de recobrimento admite, por exemplo, que várias componentes conexas de N sejam levadas na mesma componente conexa de S' . A reta real, por exemplo, é um espaço de recobrimento da esfera s^1 .

Formalmente, (N, μ_N) é um espaço de recobrimento de S' se, para todo $x \in S'$, existe uma vizinhança elementar V_x , conexa por caminhos e contendo x de tal forma que cada componente conexa por caminhos de $\mu_N^{-1}(V_x)$ é levada homeomorficamente em V_x por μ_N .

Construa-se uma vizinhança elementar de x . Note-se que o conjunto de pontos $\{\mu_N^{-1}(x)\}$ é não vazio e finito, desde que existe somente um número finito de triângulos em N .

Para cada ponto q em $\mu_N^{-1}(x)$, considere-se uma vizinhança aberta U_q contendo q e suficientemente pequeno para que U_q esteja contido apenas nos triângulos que contém q . Afirma-se que μ_N leva U_q homeomorficamente em $\mu_N(U_q)$, uma vez que ela é contínua e injetora (lema 17). Seja $U'(x) = \bigcap_{q \in \mu_N^{-1}(x)} \mu_N(U_q)$ a intersecção da imagem de cada um dos U_q . $U'(x)$ é a intersecção de um número finito de conjuntos abertos, portanto ele é aberto e ainda contém x . Pode-se extrair de $U'(x)$ um disco V_x contendo x e conexo por caminhos de forma que $\mu_N^{-1}(V_x)$ é um subconjunto de algum U_q . em triângulos adjacentes e, pelo lema 17, é injetora. Logo, (N, μ_N) é um espaço de recobrimento de S' .

Mostrar-se-á, agora, que μ_N define um homeomorfismo entre N e S' . Para isto, basta mostrar

que μ_N é injetora. Considere-se uma componente conexa G de S' . Um teorema da Topologia Algébrica (ver Massey (1967), Capítulo 5 Lema 3.4) garante que, quando (N, μ_N) é um espaço de recobrimento de S' , os conjuntos $\mu_N^{-1}(x)$ para todo $x \in G$ tem a mesma cardinalidade. Usando-se o fato de que μ_N é injetora em toda a amostra (corolário 4), como cada componente conexa de S' contém várias amostras, conclui-se que μ_N é injetora em todo N .

Finalmente, é necessário mostrar que $S = S'$. Tem-se que S' é um conjunto compacto (limitado e fechado) pois N o é. Sendo assim, S' não pode incluir parte de uma componente conexa de S e, então, S' deve consistir de um subconjunto de componente conexas de S . Desde que cada componente conexa de S contém uma amostra s (de fato contém várias), e $\mu_N(s) = s$, todas as componentes conexas de S estão em S' . Portanto, $S' = S$ e conseqüentemente, N e S são homeomorfos. ■

6.6 Considerações Finais.

Concluiu-se, neste capítulo, a apresentação de um algoritmo para a reconstrução de superfícies a partir de pontos não organizados. O algoritmo apresentado neste trabalho se mostrou altamente eficiente comparado aos algoritmos mais conhecidos da literatura. Acredita-se que tal eficiência é consequência direta da abordagem topológica empregada para derivar o algoritmo. Além disso, apresentou-se garantias teóricas de que, sob uma taxa de amostragem adequada, a reconstrução retornada pelo algoritmo Tor é homeomorfa e próxima à superfície original.

Serão resumidos, no próximo e último capítulo, os principais resultados e conclusões deste trabalho, além de algumas considerações sobre as possibilidades de trabalhos futuros.

Capítulo 7

Conclusões e Trabalhos Futuros

“Batalhas são ganhas por meio de carnificina e manobras. Quanto melhor a estratégia, mais ela contribui em manobras e menos exige em carnificina” - Winston Churchill.

7.1 Considerações Iniciais

Enfatiza-se que o uso de conceitos de topologia em aplicações computacionais tem sido objeto de recente pesquisa e tem apresentado resultados bastante promissores, especialmente quando a ênfase dos problemas se encontra na conectividade e continuidade, tanto de espaços quanto de funções (ver trabalhos de Dey et al. (1999a), Lewiner (2002) e Bern e Eppstein (1999)). Este trabalho apresenta uma elegante aplicação de topologia num problema geométrico, fato que permitiu o desenvolvimento de um algoritmo robusto e eficiente.

Este capítulo finaliza o trabalho de doutorado resumando os resultados alcançados no decorrer do período de desenvolvimento, apresentando suas principais conclusões. Também são feitas algumas considerações sobre possibilidades de trabalhos futuros e sobre questões em aberto deixadas por este trabalho.

7.2 Principais Resultados

Este trabalho se iniciou com o estudo da teoria de Robin Forman (Forman, 1998a,b) sobre teoria de Morse discreta. Posteriormente, um algoritmo de reconstrução de curvas a partir de nuvem de pontos, chamado de *Tor* (topological object reconstructor) foi desenvolvido. Foram feitas comparações com alguns algoritmos bidimensionais bastante conhecidos na literatura especializada, o β -skeleton, o *Crust*, o *NN-Crust* e o *Power Crust*.

O Crust e o Power Crust apresentaram resultados semelhantes do algoritmo Tor, entretanto os tempos de processamento foram bastante superiores devido à grande quantidade de operações geométricas executadas por esses algoritmos. O NN-Crust se mostrou quase equivalente ao nosso algoritmo bidimensional, apresentando apenas alguma instabilidade numérica devido ao cálculo de ângulos que ele realiza. Finalmente, o β -skeleton apresentou um tempo de processamento semelhante do algoritmo deste trabalho, porém com resultados muito inferiores.

Apresentaram-se, ainda, garantias teóricas de que, sob uma taxa de amostragem adequada, o algoritmo Tor, na sua versão bidimensional, fornece uma reconstrução correta do objeto original.

Na sequência, estende-se o algoritmo Tor para o caso tridimensional. Vários problemas e desafios foram encontrados durante esta extensão. Entre os principais podem-se citar o problema de uma triangulação de Delaunay tridimensional robusta e eficiente, o problema de encontrar uma função de Morse Discreta adequada para a aplicação em reconstrução e, ainda, o problema da manipulação de tetraedros slivers, que são relativamente inevitáveis no caso tridimensional.

Realizaram-se, como no caso bidimensional, comparações com alguns dos algoritmos mais conhecidos na literatura, como o Crust, Power Crust, Cocone e Tight Cocone. Enfatizou-se que o algoritmo Tor saiu-se extremamente bem nessas comparações. Foram apresentados tempos de processamento consideravelmente mais baixos do que os dos outros métodos baseados em esculpimento e, ainda, com resultados semelhantes em termos de reconstrução. O algoritmo Tor também se mostrou útil em modelos com grandes quantidades de pontos (próximas a um milhão de pontos), o que demonstra a robustez do código.

Finalmente, apresentou-se garantias teóricas de que, sob determinada taxa de amostragem, o algoritmo Tor retorna uma reconstrução homeomorfa e próxima à superfície original.

7.3 Principais Conclusões

A teoria de Morse discreta permite transformar informações de natureza geométrica em propriedades topológicas. Essa transformação possibilitou o desenvolvimento de um algoritmo que tem como principais vantagens a robustez, a eficiência e uma elevada velocidade de processamento, além de capturar componentes internas da superfície. As principais desvantagens do algoritmo Tor são a exigência de uma superfície suave e fechada, isto é, sem bordo (o algoritmo Cocone reconstrói superfícies com bordo) e a falha na identificação e eliminação de ruídos.

Além disso, foram elaboradas demonstrações teóricas, tanto para o caso bidimensional quanto

para o caso tridimensional, de que, sob uma taxa de amostragem adequada, o algoritmo aqui proposto produz uma correta reconstrução do objeto amostrado. Vale a pena salientar que a taxa de amostragem obtida no caso bidimensional é melhor do que as dos algoritmos utilizados nas comparações.

Finalizando, a teoria de Morse discreta se mostrou uma excelente ferramenta, quando aplicada em um problema prático, pois permitiu o desenvolvimento de um algoritmo de reconstrução a partir de nuvem de pontos que preserva todas as boas propriedades dos algoritmos mais conhecidos na literatura, com um ganho considerável de desempenho computacional.

7.4 Trabalhos Futuros

São comentados, aqui, algumas possibilidades de trabalhos futuros deixadas em aberto por esta tese.

O próximo passo natural é o uso de funções de Morse discretas para a identificação e eliminação de ruídos. Uma possibilidade, por exemplo, é tentar encontrar uma função de Morse na qual todos pontos que não pertencem à superfície fossem pontos críticos da função. Outra possibilidade a ser explorada é a utilização dos caminhos gradientes (ver definição 49, capítulo 4) para a identificação dos ruídos. Uma extensão para que o algoritmo funcione em superfícies com bordo também pode ser pensada.

Outra questão interessante que pode colaborar com os testes teóricos dos algoritmos é a determinação da condição de r -amostragem para conjunto de amostras. Em todos os conjuntos testados, não havia como saber se o conjunto estava r -amostrado e, se estivesse, não havia como saber qual o valor de r .

Uma questão que também merece ser explorada mais a fundo é a relação existente entre os pontos críticos da função de Morse e a homologia, isto é, número de cavidades, buracos e componentes conexas de um determinado conjunto de amostras. Talvez seja possível identificar e tratar estes objetos por meio da análise dos pontos críticos de uma determinada função de Morse discreta.

Uma grande limitação presente em todas as técnicas de esculpimento é a necessidade da construção de uma triangulação de Delaunay tridimensional. Além do custo computacional elevado, esta exigência impede a aplicação destas técnicas em conjuntos de dados muito grandes. Um grande desafio, portanto, seria a tentativa de aplicar os conceitos topológicos evitando a construção da triangulação de Delaunay tridimensional como passo inicial.

7.5 Considerações Finais

Os resultados apresentados neste trabalho permitem concluir que o uso de funções de Morse discretas representa um avanço em reconstrução a partir de nuvem de pontos, tanto bidimensional quanto tridimensional, pois diminui consideravelmente a quantidade de operações geométricas envolvidas no processo e preserva as boas propriedades dos melhores algoritmos, tais como garantias teóricas, robustez e boa performance computacional.

Acredita-se ainda que o uso de conceitos de topologia em problemas de geometria computacional é uma tendência que vem se desenvolvendo entre os pesquisadores tanto de uma área quanto de outra, mostrando um bom exemplo de integração entre estes dois campos do conhecimento. Essa integração é bastante positiva e pode trazer avanços para os pesquisadores de ambas as áreas.

Ao longo do desenvolvimento desta tese, este pesquisador teve dois artigos publicados: o primeiro, intitulado “Uma Abordagem Topológica Para a Reconstrução de Curvas e Superfícies a Partir de Pontos não Organizados”, publicado nos proceedings do Cilamce 2004; o segundo, com o título de “Topological Tetrahedron Characterization with Application in Volumetric Reconstruction” foi aceito para publicação na revista “International Journal of Shape Modeling”. Foram produzidos, ainda, mais dois artigos que estão submetidos, um com o título de “A Topological Approach to Curve Reconstruction from Scattered Points” e o outro com o de “A Topological Approach for Surface Reconstruction from Sample Points”.

Referências Bibliográficas

- Adamy, U., Giesen, J., e John, M. (2000). New techniques for topologically correct surface reconstruction. *Proceedings of the 11th IEEE Visualization 2000 Conference (VIS 2000)*., páginas: 373-380.
- Althaus, E. e Mehlhorn, K. (2000). Tsp-based curve reconstruction in polynomial time. *Symposium on Discrete Algorithms*, 2 páginas: 686-695.
- Amenta, N. e Bern, M. (1999). Surface reconstruction by voronoi filtering. *Discrete and Computational Geometry*, 22 páginas: 481-504.
- Amenta, N., Bern, M., e Eppstein, D. (1998a). The crust and the -skeleton: Combinatorial curve reconstruction. *Graphical models and image processing: GMIP*, 60(2) páginas: 125-135.
- Amenta, N., Bern, M., e Kamvysselis, M. (1998b). A new Voronoi-based surface reconstruction algorithm. *Computer Graphics*, 32(Annual Conference Series) páginas: 415-421.
- Amenta, N. e Bern, M. W. (1998). Surface reconstruction by voronoi filtering. In *Symposium on Computational Geometry*. páginas: 39-48.
- Amenta, N. e Choi, S. (1999). One-pass delaunay filtering for homeomorphic 3d surface reconstruction. Technical Report TR99-08, UT Technical Report.
- Amenta, N., Choi, S., Dey, T. K., e Leekha, N. (2002a). A simple algorithm for homeomorphic surface reconstruction. *International Journal of Computational Geometry and Applications*, 12(1-2) páginas: 125-141.
- Amenta, N., Choi, S., Dey, T. K., e Leekha, N. (2002b). A simple algorithm for homeomorphic surface reconstruction. *International Journal of Computational Geometry and Applications*, 12(1-2) páginas: 125-141.

- Amenta, N., Choi, S., e Kolluri, R. (2001a). The power crust. In *6th ACM Symposium on Solid Modeling*, 249-260.
- Amenta, N., Choi, S., e Kolluri, R. K. (2001b). The power crust, unions of balls, and the medial axis transform. *Computational Geometry*, 19(2-3) páginas: 127-153.
- Amenta, N. e Kolluri, R. K. (2000). Accurate and efficient unions of balls. In *Symposium on Computational Geometry*, páginas: 119-128.
- Attene, M. e Spagnuolo, M. (2000). Automatic surface reconstruction from point sets in space. *Computer Graphics Forum*, 19(3) páginas: 0-0(1).
- Bajaj, C. L., Bernardini, F., e Xu, G. (1995). Automatic reconstruction of surfaces and scalar fields from 3d scans. *Computer Graphics*, 29(Annual Conference Series) páginas: 109-118.
- Bajaj, C. L., Coyle, E. J., e Lin, K.-N. (1999). Surface and 3d triangular meshes from planar cross sections. *Computer Methods Applied Mechanics Engineer*, (179) páginas: 266-286.
- Bardinet, E., Cohen, L. D., e Auache, N. (1998). A parametric deformable model to fit unstructured 3d data. *Computer Vision and Image Understanding*, 71(1) páginas: 39-54.
- Bern, M. e Eppstein, D. (1999). Emerging challenges in computational topology. Report from the NSF-funded Workshop on Computational Topology, Miami Beach, FL, June 11-12, 1999.
- Bernardini, F. e Bajaj, C. L. (1997). Sampling and reconstructing manifolds using alpha-shapes. In *Proc. 9th Canadian Conf. Computational Geometry*, páginas: 193-198.
- Bernardini, F., Mittleman, J., Rushmeier, H., Silva, C., e Taubin, G. (1999). The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 5(4) páginas: 349-359.
- Boissonnat, J. D. (1988). Shape reconstruction from planar cross-sections. *Computational Vision Image*, 44 páginas: 1-29.
- Boissonnat, J. D. e Cazals, F. (2002). Smooth surface reconstruction via natural neighbour interpolations of distance functions. *Computational Geometry Theory and Applications*, 22 páginas: 185-203.

- Carr, J. C., Beatson, R. K., Cherrie, J. B., Mitchell, T. J., Fright, W. R., McCallum, B. C., e Evans, T. R. (2001). Reconstruction and representation of 3d objects with radial basis functions. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, páginas: 67–76. ACM Press.
- CGAL (2004). <http://www.cgal.org> (acessado em 05/02/2004) cgal - computational geometry algorithms library.
- Cheng, S.-W., Funke, S., Golin, M., Kumar, P., Poon, S.-H., e Ramos, E. (2003). Curve reconstruction from noisy samples. In *Proceedings of the nineteenth annual symposium on Computational geometry*, páginas: 302–311. ACM Press.
- Chew, L. P. (1993). Guaranteed-quality mesh generation for curved surfaces. In *ACM Symposium on Computational Geometry*, páginas: 274–280.
- Curless, B. e Levoy, M. (1996). A volumetric method for building complex models from range images. In *In Proceedings ACM SIGGRAPH96.*, páginas: 303–312.
- Dey, T., Edelsbrunner, H., e Guha, S. (1999a). Computational topology. In *Advances in Discrete and Computational Geometry (Contemporary mathematics 223)*, páginas: 109–143.
- Dey, T. K., Giesen, J., e Zhao, W. (2001). Robustness issues in surface reconstruction. *Lecture Notes in Computer Science*, 2073 páginas: 658–663.
- Dey, T. K. e Goswami, S. (2002a). Tight cocone : A water-tight surface reconstruction. Technical Report OSU-CISRC-12/02-TR31, The Ohio State University.
- Dey, T. K. e Goswami, S. (2002b). Tight cocone and cgal. In *CGAL workshop*.
- Dey, T. K. e Goswami, S. (2003). Tight cocone: a water-tight surface reconstructor. In *Proceedings of the eighth ACM symposium on Solid modeling and applications*, páginas: 127–134. ACM Press.
- Dey, T. K. e Kumar, P. (1999). A simple provable algorithm for curve reconstruction. *Proc. 10th. ACM-SIAM Symposium on Discrete Algorithms (SODA '99)*, páginas: 893–894.
- Dey, T. K., Mehlhorn, K., e Ramos, E. A. (1999b). Curve reconstruction: Connecting dots with good reason. In *Symposium on Computational Geometry*, páginas: 197–206.

- Dibben, D. C. (2000). 3d mesh generation using bubble meshing for microwave applicators. *IEEE T MAGN* 36:, 4 páginas: 1514–1518.
- Edelsbrunner, H. (2001). *Geometry and Topology for Mesh Generation*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge.
- Edelsbrunner, H. (2002). Surface reconstrution by wrapping finite point set in space. Rick Pollack e Eli Goodman Festschrift, ed. A. Aronov, S. Basu, J. Pach e M. Sharir. Springer-Verlag, páginas: 379–404.
- Edelsbrunner, H. e Mücke, E. P. (1994). Three-dimensional alpha shapes. *ACM Transactions on Graphics*, 13(1) páginas: 43–72.
- Edelsbrunner, H. e Shah, N. (1994). Triangulating topological spaces. In *Proc. 10th ACM Symp. Computational Geometry*, páginas: 285–292.
- Figueiredo, L. H. e Miranda, J. G. (1995). Computational morphology of curves. *Visual Computer*, 11 páginas: 105–112.
- Forman, R. (1998a). Morse theory for cell complexes. *Advances in Mathematics*, 134(1) páginas: 90–145.
- Forman, R. (1998b). Witten-morse theory for cell complexes. *Topology*, 37(5) páginas: 945–979.
- Giesen, J. (1999). Curve reconstruction in arbitrary dimension and the traveling salesman problem. *Lecture Notes in Computer Science*, 1568 páginas: 164–176.
- Giesen, J. e John, M. (2002a). How to add facet attributes to cgal’s 3d geometric triangulations. CGAL User Workshop (2002) 4 pages.
- Giesen, J. e John, M. (2002b). Surface reconstruction based on a dynamical system. In Drettakins, G. e Siedel, H., editors, *Eurographics*, volume 21-3.
- Giesen, J. e John, M. (2003). The flow complex: a data structure for geometric modeling. In *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, páginas: 285–294. Society for Industrial and Applied Mathematics.
- Gois, J. P. (2004). Reconstrução de superfícies a partir de nuvens de pontos. Dissertação de Mestrado, Universidade de São Paulo - Instituto de Ciências Matemáticas e Computação.

- Gopi, M. e Krishnan, S. (2002). A fast and efficient projection based approach for surface reconstruction. In *proceedings of the 15th Brazilian Symposium on Computer Graphics and Image Processing*, páginas: 179-186.
- Gopi, M., Krishnan, S., e Silva, C. T. (2000). Surface reconstruction based on lower dimensional localized delaunay triangulation. In Gross, M. e Hopgood, F. R. A., editors, *Computer Graphics Forum (Eurographics 2000)*, volume 19(3), páginas: C467-C478.
- Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., e Stuetzle, W. (1992). Surface reconstruction from unorganized points. *Computer Graphics*, 26(2) páginas: 71-78.
- Huang, J. e Menq, C. H. (2002). Combinatorial manifold reconstruction and optimization from unorganized point cloud with arbitrary topology. *Computer-Aided Design*, 1(34) páginas: 149-165.
- Hungerford, T. W. (1996). *Álgebra /Thomas W. Hungerford*. New York : Springer.
- Isaacson, E. e Keller, H. B. (1966). *Analysis of Numerical Methods*. John Wiley & Sons.
- Jin, H. e Tanner, R. I. (1993). Generation of unstructured tetrahedral meshes by advancing front technique. *International Journal of Numerical Methods in Engineering*, (36) páginas: 1805-1823.
- Jones, M. W. e Chen, M. (1994). A new approach to the construction of surfaces from contour data. In *Eurographics*, volume 13, páginas: 75-84.
- Kirkpatrick, D. G. e Radke, J. D. (1985). Framework for computational morphology - computational geometry. *Comput Methods Applied Mechanical Engineering*, páginas: 217-245.
- Kolluri, R., Shewchuk, J. R., e O'Brien, J. F. (2004). Spectral surface reconstruction from noisy point clouds. In *Siggraph*, páginas: 11-22.
- Levoy, M., Pulli, K., Curless, B., Rusinkiewicz, S., Koller, D., Pereira, L., Ginzton, M., Anderson, S., Davis, J., Ginsberg, J., Shade, J., e Fulk, D. (2000). The digital michelangelo project: 3D scanning of large statues. In Akeley, K., editor, *Siggraph 2000, Computer Graphics Proceedings*, páginas: 131-144. ACM Press / ACM SIGGRAPH / Addison Wesley Longman.

- Lewiner, T. (2002). Constructing discrete Morse functions. Dissertação de Mestrado, Department of Mathematics, PUC-Rio.
- Lewiner, T., Lopes, H., e Tavares, G. (2004). Applications of formans discrete morse theory to topology visualization and mesh compression. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 10(5) páginas: 499–508.
- Lima, E. L. (1976). *Elementos de topologia geral*. Rio de Janeiro : Ao Livro Tecnico, elementos de matemtica (impa) edition.
- Lima, E. L. (1981). *Curso de Anlise Vol. 2*. Instituto de Matemtica Pura e Aplicada IMPA.
- Marcum, D. L. e Weatherill, N. P. (1995). Unstructured grid generation using iterative point insertion and local reconnection. *AIAA Journal*, 33(9) páginas: 1619–1625.
- Massey, W. S. (1967). *Algebraic Topology: An Introduction*. Spring-Verlag, Graduate texts in Mathematics 56.
- Massey, W. S. (1991). *A Basic Course in Algebraic Topology*. Spring-Verlag, Graduate texts in Mathematics 127.
- Mederos, B., Velho, L., e de Figueiredo, L. H. (2003). Moving least squares multiresolution surface approximation. In *SIBGRAPI 2003*, páginas: 19–26.
- Milnor, J. (1963). *Morse Theory*. Princeton University Press, Princeton, NY.
- Moreira, J. G. V. (1998). Desenvolvimento de um modelador para superfícies implícitas em $\mathbb{R}^3$. Dissertação de Mestrado, Universidade de São Paulo - Instituto de Ciências Matemáticas e Computação.
- Nonato, L., Minghim, R., Oliveira, M., e Tavares, G. (2001). A novel approach for delaunay 3d reconstruction with a comparative analysis in the light of applications. *Computer Graphics Forum*, 20(2) páginas: 161–171(11).
- Ohtake, Y., Belyaev, A., Alexa, M., Turk, G., e Seidel, H.-P. (2003). Multi-level partition of unity implicits. *ACM Trans. Graph.*, 22(3) páginas: 463–470.

- Rodríguez, A., Espadero, J. M., López, D., e Pastor, L. (2000). Delaunay surface reconstruction from scattered points. In *DGCI '00: Proceedings of the 9th International Conference on Discrete Geometry for Computer Imagery*, páginas: 272–283. Springer-Verlag.
- Ruppert, J. e Seidel, R. (1989). On the difficulty of tetrahedralizing 3-dimensional non-convex polyhedra. páginas: 380–392. 5th Annual ACM Symposium Comput Geom Proceedings.
- Samet, H. (1990). *The design and analysis of spatial data structures*. Addison-Wesley Longman Publishing Co., Inc.
- Teichmann, M. e Capps, M. (1998). Surface reconstruction with anisotropic density-scaled alpha shapes. In *Proceedings of the conference on Visualization '98*, páginas: 67–72. IEEE Computer Society Press.
- Teng, S. H. e Wong, C. W. (2000). Unstructured mesh generation: Theory, practice, and perspectives. *International Journal of Computer Geometry*, 10(3) páginas: 227–266.
- Zhao, H., Merriman, B., Osher, S., e Kang, M. (2000). Implicit nonparametric shape reconstruction from unorganized points using a variational level set method. *Computer Vision and Image Understanding*, 60 páginas: 295 – 313.