Visualização Automática de Complexos Celulares Arbitrários

Rober Marcone Rosi

Visualização Automática de Complexos Celulares Arbitrários

Este exemplar corresponde à redação final da tese devidamente corrigida e defendida pelo Sr. Rober Marcone Rosi e aprovada pela Comissão Julgadora.

Campinas, 20 de novembro de 1995.

Jorge Stolfi
Orientador

Dissertação apresentada ao Instituto de Matemática, Estatística e Ciência da Computação, UNICAMP, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

Visualização Automática de Complexos Celulares Arbitrários

Rober Marcone Rosi

Dissertação apresentada ao
Instituto de Matemática, Estatística e Ciência da Computação da UNICAMP,
como requisito parcial para a obtenção do título de
Mestre em Ciência da Computação.

Banca Examinadora:

Ronaldo César Marinho Persiano
COPPE – Universidade Federal do Rio de Janeiro

Cândido Ferreira Xavier de Mendonça Neto

João Carlos Setubal (suplente)

Jorge Stolfi (orientador)

O autor é Bacharel em Matemática Aplicada e Computacional
pela Universidade Federal do Espirito Santo.

Departamento de Ciência da Computação
IMECC — UNICAMP

Dedicatória

Dedico este trabalho aos meus pais pelo apoio e incentivo recebidos ao longo de toda minha vida. A minha irmã, por toda alegria que sempre demonstra ter.

Agradecimentos

De forma especial, agradeço a minha mãe, por sua constante preocupação com meu bem estar, estando sempre pronta a resolver meus problemas durante todo tempo que estive ausente, por estar sempre pronta a me ouvir e por sempre ter me incentivado, mesmo nos momentos mais difíceis por que passei.

Ao professor e amigo Jorge Stolfi, pela lição de profissionalismo, por sua constante dedicação ao trabalho, por sua compreensão quanto as minhas limitações, por suas palavras de incentivo, e principalmente, por ter sempre me tratado não apenas como um "orientado" mas como um amigo.

Aos meus amigos de república, pelas horas de agradável convívio que tivemos juntos.

A todos os amigos com quem tive a oportunidade de conviver durante os anos de mestrado.

Aos amigos do volei, pelos agradáveis momentos esportivos que compartilhamos.

Aos meus professores de graduação da Universidade Federal do Espirito Santo (UFES), por terem sidos os primeiros a incentivarem minha vinda para o mestrado.

À CAPES/PICD por ter financiado meu trabalho, a FAEP/UNICAMP por ter me proporcionado um complemento a minha bolsa de mestrado, e ao PROTEM/CNPQ por financiar minha participação na IX Escola de Computação (Recife).

Resumo

Um *complexo celular bidimensional* é uma subdivisão de uma superfície num número finito de elementos — faces (discos abertos), arestas (curvas abertas) e vértices (pontos). Descreve-se aqui um programa que, dada apenas a estrutura topológica de um complexo celular (ou seja, as relações de incidência e adjacência entre seus elementos), determina uma representação geométrica do mesmo (uma superfície subdividida), que é "bonita" e permite visualizar facilmente a topologia do complexo.

Abstract

A *two-dimensional cell complex* is a partition of a surface into a finite number of elements — faces (open discs), edges (open curves) and vertices (points). Here, is described a program which given only the topological structure of a cell complex (that is, the incidence and adjacency relationships between its elements), constructs a geometric representation of it — a subdivided surface — which is "nice looking" and allows one to clearly visualize the topology of the complex.

Conteúdo

Dedicatória iv

Agradecimentos v

Resumo vi

Abstract vii

1 Introdução 1

1.1 O problema: visualização da topologia 1

1.2 Solução: uma ferramenta de visualização 4

1.3 O que é uma representação bonita? 4

1.4 Trabalhos relacionados 5

2 Complexos celulares 7

2.1 Conceitos fundamentais 7

2.1.1 Variedade 7

2.1.2 Definição de complexo celular 8

2.1.3 Modelo de colagem 8

2.1.4 Funções de percurso 9

2.1.5 Subdivisão baricêntrica 12

2.1.6 Complexo celular abstrato 13

2.1.7 Complexos duais 13

2.1.8 Duais de arestas dirigidas e orientadas 14

2.1.9 A estrutura *quad-edge* 15

3 Modelagem geométrica do complexo 19

3.1 Ladrilhamento do complexo 19

3.1.1 Modelagem por retalhos polinomiais 20

3.1.2 Modelagem por triângulos 21

3.2 O algoritmo de ladrilhamento 22

3.2.1 Construção dos ladrilhos 22
3.2.2 Colagem dos ladrilhos 23
3.2.3 Enumeração dos pares a colar 24
3.2.4 Descrição detalhada do algoritmo de colagem 24
3.2.5 Eliminação de triângulos degenerados 26
3.3 Visualização gráfica da superfície 31

4 A função energia **33**
4.1 Configurações 33
4.1.1 Vértices fixos e variáveis 33
4.2 Funções energia utilizadas 34
4.2.1 Configurações degeneradas 34
4.3 Energia de dobramento 35
4.3.1 Controle Local 36
4.3.2 Simplificações 36
4.4 Energia de excentricidade 37
4.5 Energia de ângulos 37
4.6 Energia de proximidade 38
4.7 Energia de espalhamento 44
4.8 Energia de área de ladrilhos 44
4.9 Outras energias 45
4.10 Análise das energias em função da triangulação 46
4.10.1 Energia de dobramento 47
4.10.2 Energia excêntrica 48
4.10.3 Energia de áreas de ladrilhos 48
4.10.4 Energia de espalhamento 49
4.10.5 Energia de proximidade 49
4.10.6 Energia de ângulos 49
4.11 Energias mistas 50
4.11.1 Peso crítico: $\mathcal{E}^{prox} \times \mathcal{E}^{dobr}$ 50

5 Otimização **53**
5.1 Introdução 53
5.1.1 Mínimo global vs. local 54
5.2 Métodos genéricos e exatos de otimização 54
5.2.1 Minimização coordenada a coordenada 55
5.2.2 Descida pelo gradiente 56
5.2.3 O método do simplexo de Nelder-Mead 56
5.2.4 O método de Powell e Brent 59

5.3 Comparação entre os métodos de otimização 60
5.3.1 Comparação entre `Simplex`, `Praxis` e `Coord` 60
5.3.2 Comparação entre `Grad` e `Coord` 63
5.4 Métodos Heurísticos 66
5.4.1 Heurística de desdobramento 66
5.4.2 Heurística de equalização de ângulos 68
5.4.3 Normalização 69

6 Resultados **71**
6.1 Toro 71
6.2 Garrafa de Klein 73
6.3 Salsicha 74
6.4 Salame 76
6.5 Laranja 77
6.6 Tritoro 78
6.7 Estrelas e não-estrelas 79
6.8 Influência das várias energias 82
6.8.1 Combinação $\mathcal{E}^{dobr} + \mathcal{E}^{excen} + \mathcal{E}^{prox}$ 82
6.8.2 Combinação $\mathcal{E}^{dobr} + \mathcal{E}^{area} + \mathcal{E}^{ang}$ 83
6.8.3 Combinação $\mathcal{E}^{dobr} + \mathcal{E}^{area} + \mathcal{E}^{prox}$ 84
6.8.4 Combinações com $\mathcal{E}^{dobr}$ e $\mathcal{E}^{excen}$ 85
6.8.5 Combinação $\mathcal{E}^{area} + \mathcal{E}^{excen} + \mathcal{E}^{prox}$ 87

7 Conclusões **89**
7.1 Os problemas tratados 89
7.2 Os resultados 90
7.3 Sugestões e trabalhos futuros 90

A Conceitos elementares de topologia **91**
A.1 Espaço topológico 91
A.2 Topologia da reta real 91
A.3 Espaço produto 92
A.4 Sub-espaços 92
A.5 Espaço quociente 92
A.6 Separabilidade 93
A.7 Vizinhança, fecho e fronteira 93
A.8 Ponto de acumulação e limite 93
A.9 Continuidade e homeomorfismo 94

Bibliografia **95**

Lista de Figuras

1.1 Dois objetos cujas representações de fronteira tem a mesma topologia. . . . 1
1.2 Um modelo de colagem descrevendo a topologia dos objetos da figura 1.1(a) e (b) 2
1.3 Uma garrafa de Klein. 3
1.4 Um toro. 3
1.5 Um complexo celular (a), e duas representações (b) e (c). 5

2.1 Orientações longitudinal e transversal de uma aresta [12]. 9
2.2 Resultado de um *Flip* aplicada sobre a aresta e da figura 2.1 [12]. 10
2.3 As funções de percurso. 11
2.4 A subdivisão baricêntrica de um complexo. 12
2.5 Um complexo C (linha cheia) e um dual estrito C^* do mesmo (linha tracejada). 14
2.6 Alguns Complexos celulares representáveis pela *quad-edge* 16
2.7 Um complexo celular e sua respectiva representação na estrutura *quad-edge* 17

3.1 Um complexo celular bidimensional, mostrando arestas primais (linhas cheias), arestas duais (linhas tracejadas), a divisão em ladrilhos (linhas pontilhadas) e um dos ladrilhos (em cinza). 20
3.2 O ladrilho de uma aresta e e a malha 5×5 correspondente. 21
3.3 A triangulação de um ladrilho, e a função *Corner*. 22
3.4 Representação da colagem de dois ladrilhos adjacentes 23
3.5 Efeito da operação *Splice* sobre duas arestas a e b. 25
3.6 Pares de triângulos degenerados. 26
3.7 Possíveis tipos de vértices antes da colagem (C = Canto, L = Lado, e I = Interior do ladrilho). 27
3.8 Possíveis tipos de triângulos para ladrilhos antes da colagem. 28
3.9 Possíveis combinações dos triângulos R e S antes da colagem. 29
3.10 Pares de triângulos que se tornarão degenerados criados pela colagem. . . . 30
3.11 Colagem de um ladrilho a si mesmo, e eliminação de triângulos degenerados 31
3.12 Um ladrilho de dimensão 3×3 representando os quatro quadrantes coloridos. 32

4.1 Gráfico de $2(1 - \cos\theta)$ em função de θ. 36
4.2 Realizações com trançamentos 38
4.3 Energia de proximidade (a) usando cargas pontuais e (b) usando cargas nebulosas. 40
4.4 Gráficos de $\mathcal{E}^{prox}$ para dois ladrilhos paralelos, em função de seu deslocamento vertical (*Offset*), para diferentes valores do parâmetro α 41
4.5 Gráficos de $\mathcal{E}^{prox}$ para dois ladrilhos cruzados, em função do ângulo de cruzamento, para diferentes valores de α 42
4.6 Gráficos de $\mathcal{E}^{prox}$ para dois ladrilhos cruzados perpendicularmente, em função da posição do cruzamento, para diferentes valores de α. 43
4.7 Gráfico da energia de área de um quartilho $\mathcal{E}_i^{area}$, em função de sua área a_i. 45
4.8 Um ladrilho triangulado de ordem 1×1 (a), e sua duplicação (b). 46
4.9 O "efeito abacaxi". 50
4.10 A triangulação usada na análise do "efeito abacaxi". 51
4.11 Representação das energias de proximidade (a), de dobramento (b), e total (c), em função do parâmetro h. 51

5.1 As três operações do método `Simplex`: reflexão (x^r), expansão (x^e), e contração (x^c). 57
5.2 O complexo `tetra` utilizado nos testes. 60
5.3 Comparação dos métodos `Simplex`, `Praxis`, e `Coord` partindo de uma configuração aleatória (energia $\times$ segundos de CPU). 61
5.4 Comparação dos métodos `Simplex`, `Praxis`, e `Coord` partindo de uma configuração "amaciada" (energia $\times$ segundos de CPU). 61
5.5 Comparação dos métodos `Simplex`, `Praxis`, e `Coord` partindo de uma configuração aleatória (energia $\times$ chamadas da função). 62
5.6 Comparação dos métodos `Simplex`, `Praxis`, e `Coord` partindo de uma configuração "amaciada" (energia $\times$ chamadas da função). 63
5.7 Comparação dos métodos `Grad` e `Coord` partindo de uma configuração aleatória (energia $\times$ segundos de CPU). 64
5.8 Comparação dos métodos `Grad` e `Coord` partindo de uma configuração "amaciada" (energia $\times$ segundos de CPU). 64
5.9 Comparação dos métodos `Grad` e `Coord` partindo de uma configuração aleatória (energia $\times$ chamadas da função). 65
5.10 Comparação dos métodos `Grad` e `Coord` partindo de uma configuração "amaciada" (energia $\times$ chamadas da função). 65
5.11 (a) Variáveis relevantes para o amaciamento. (b) A função $f(\theta)$. 67
5.12 A estrela de um vértice u, o plano Φ e os eixos x e y definidos pelo vértice v_k. 69

6.1 O modelo de colagem para o complexo `torus` 71

6.2 Realização heurística da triangulação `torus-5`. 72
6.3 Otimização da energia da triangulação `torus-5`. 72
6.4 O modelo de colagem do complexo `klein` 73
6.5 Evolução da triangulação `klein-5`. 73
6.6 Estrutura do estágio final da figura 6.5. 74
6.7 O modelo de colagem do complexo `sausage` 74
6.8 Evolução do modelo `sausage-3`. 75
6.9 O modelo `sausage-3` otimizado. 75
6.10 O modelo de colagem do complexo `salami`. 76
6.11 Evolução do modelo `salami-5`. 76
6.12 O modelo de colagem `orange` 77
6.13 Configuração final do modelo `orange-3`. 77
6.14 O modelo de colagem `tritorus` 78
6.15 Configuração final do modelo `tritorus-3`. 78
6.16 O modelo de colagem do complexo `star` 79
6.17 Primeira tentativa de construir o modelo `star-3`. 80
6.18 Segunda tentativa de construir o modelo `star-3`. 80
6.19 Configuração final obtida para modelo `star-3`. Agora o algoritmo de construção parece estar correto... 81

Capítulo 1

Introdução

1.1 O problema: visualização da topologia

Em computação gráfica, objetos tridimensionais são freqüentemente modelados pela técnica de *representação de fronteira*, que consiste em descrever a superfície de cada objeto como a união de *faces* (retalhos de superfícies, planas ou curvas). As faces são limitadas por *arestas* (segmentos de retas ou curvas), cujos extremos são os *vértices* do objeto.

Na implementação de algoritmos que manipulam tais objetos, a experiência aconselha separar os aspectos *topológicos* (que dizem respeito aos contatos entre os vértices, arestas, e faces) e os aspectos *geométricos* (que incluem as coordenadas dos vértices, a forma das arestas e das faces, etc.). Veja a figura 1.1. Esta separação facilita o desenvolvimento e depuração de algoritmos, e a modelagem dos objetos [23].

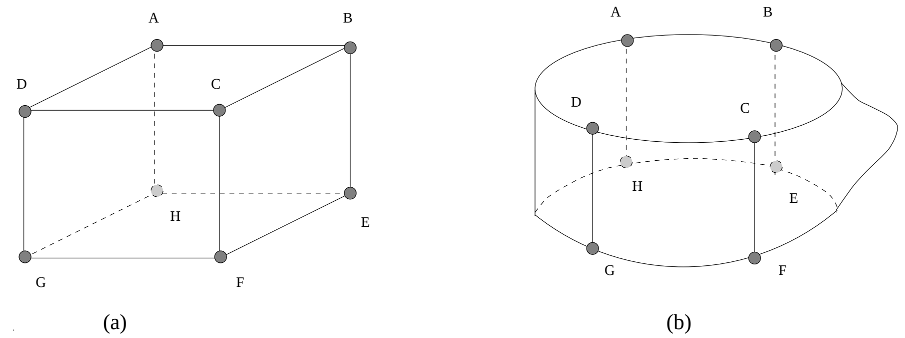


Figura 1.1: Dois objetos cujas representações de fronteira tem a mesma topologia.

Informalmente, as relações topológicas entre as faces, arestas, e vértices de um objeto podem ser descritas por meio de um *modelo de colagem*. Nesse modelo, cada face do objeto é representada por um polígono simples no plano, cada lado do qual está decorado com uma seta longitudinal e um rótulo. Cada rótulo aparece em exatamente dois lados, no mesmo polígono ou em polígonos diferentes. Entende-se que esses dois lados, unidos como indicado pelas setas, constituem uma única aresta do objeto. Entende-se também que a forma e o tamanho dos polígonos não tem relação nenhuma com a forma e tamanho das faces correspondentes do objeto.

A figura 1.2 é um modelo de colagem que descreve a topologia das arestas, vértices, e faces de um cubo, como o da figura 1.1(a). Note que esse mesmo modelo de colagem descreve também a topologia da representação de fronteira do sólido da figura 1.1(b).

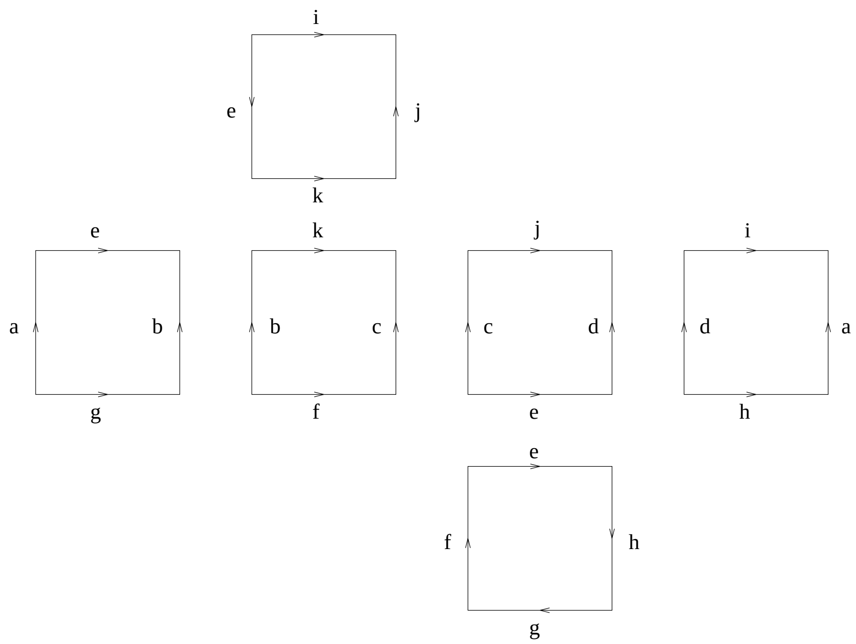


Figura 1.2: Um modelo de colagem descrevendo a topologia dos objetos da figura 1.1(a) e (b)

A topologia definida por um modelo de colagem pode ser bastante difícil de visualizar. Por exemplo, o modelo de colagem da figura 1.3(a), com apenas uma face e duas arestas, descreve a topologia das faces, arestas e vértices do objeto ilustrado na figura 1.3(b); a superfície em questão é a *garrafa de Klein*, que tem a peculiaridade de possuir um único lado.

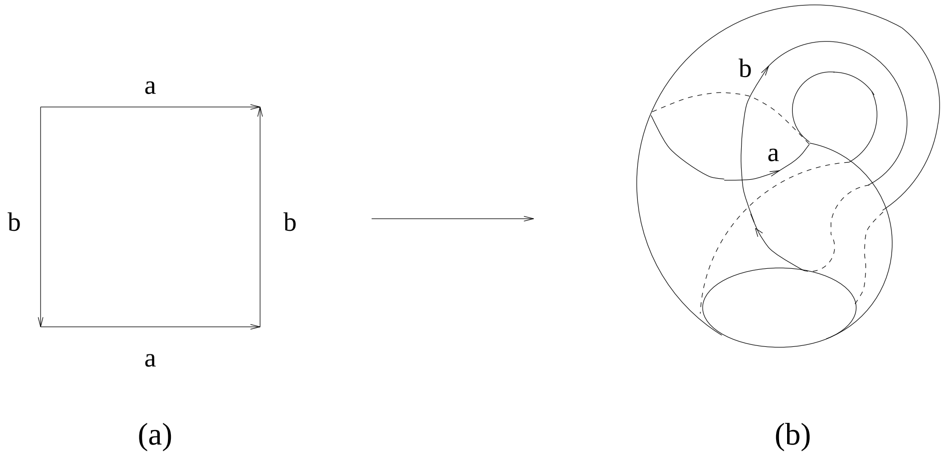


Figura 1.3: Uma garrafa de Klein.

Compare este exemplo com o modelo de colagem da figura 1.4(a), que descreve o objeto da figura 1.4(b); a superfície neste caso é um toro, que tem dois lados.

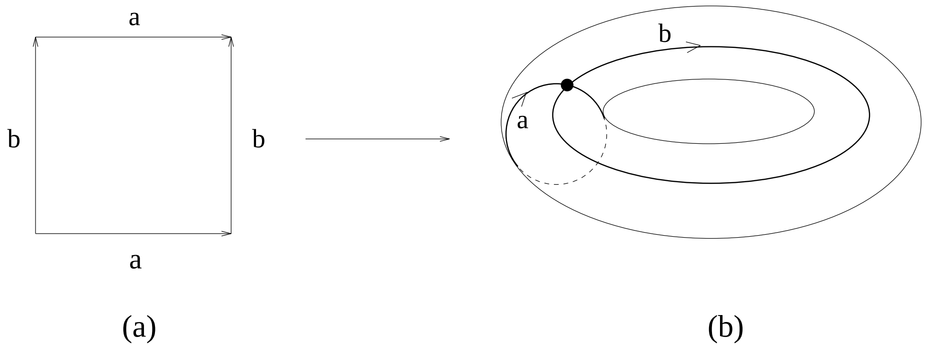


Figura 1.4: Um toro.

As figuras 1.3 e 1.4 demonstram que a topologia de um objeto complexo, com um número grande de faces e arestas, é praticamente impossível de visualizar sem algum auxilio computacional.

1.2 Solução: uma ferramenta de visualização

Esta dificuldade nos motivou a desenvolver um programa para a visualização automática da topologia de complexos celulares arbitrários. A entrada deste programa é uma estrutura de dados que representa a topologia de um complexo, como nas figuras 1.3(a) e 1.4(a). A saída é uma realização geométrica "bonita" da mesma, como nas figuras 1.3(b) e 1.4(b). Esta representação consiste de uma superfície de forma específica, mergulhada no espaço $\mathbb{R}^3$; e de uma divisão da mesma em faces, vértices, e arestas, correspondentes aos elementos do complexo, com as mesmas relações de adjacência. Esta superfície pode ser então desenhada por ferramentas padrões de computação gráfica.

1.3 O que é uma representação bonita?

Para resolver este problema automaticamente, é preciso primeiro definir o conceito de "representação bonita." Informalmente, o que se quer é uma superfície que, em primeiro lugar, permita a visualização fácil da topologia do complexo celular; e, em segundo lugar, seja esteticamente agradável.

De modo geral, para satisfazer estes objetivos, a superfície precisa ser suave e bem distribuída no espaço; na medida do possível, ela não deve cruzar a si mesma, e as faces e arestas devem ter formas simples e tamanhos similares.

Note a diferença entre a topologia do *complexo* (as relações de adjacência entre faces, vértices e arestas) e a topologia da *superfície* (que se resume na sua orientabilidade e conectividade). Para uma boa visualização da primeira, não basta que a topologia da superfície seja fácil de entender; é preciso também que as arestas e faces, pintadas sobre a superfície, sejam simples e bem separadas, e que suas adjacências sejam claramente visíveis.

Em particular, mesmo que um complexo celular tenha superfície homeomorfa a uma esfera, sua melhor representação não é necessariamente uma esfera geométrica. Por exemplo, a estrutura do complexo da figura 1.5(a) fica mais fácil de visualizar numa superfície alongada, como em (c), do que na esfera (b).

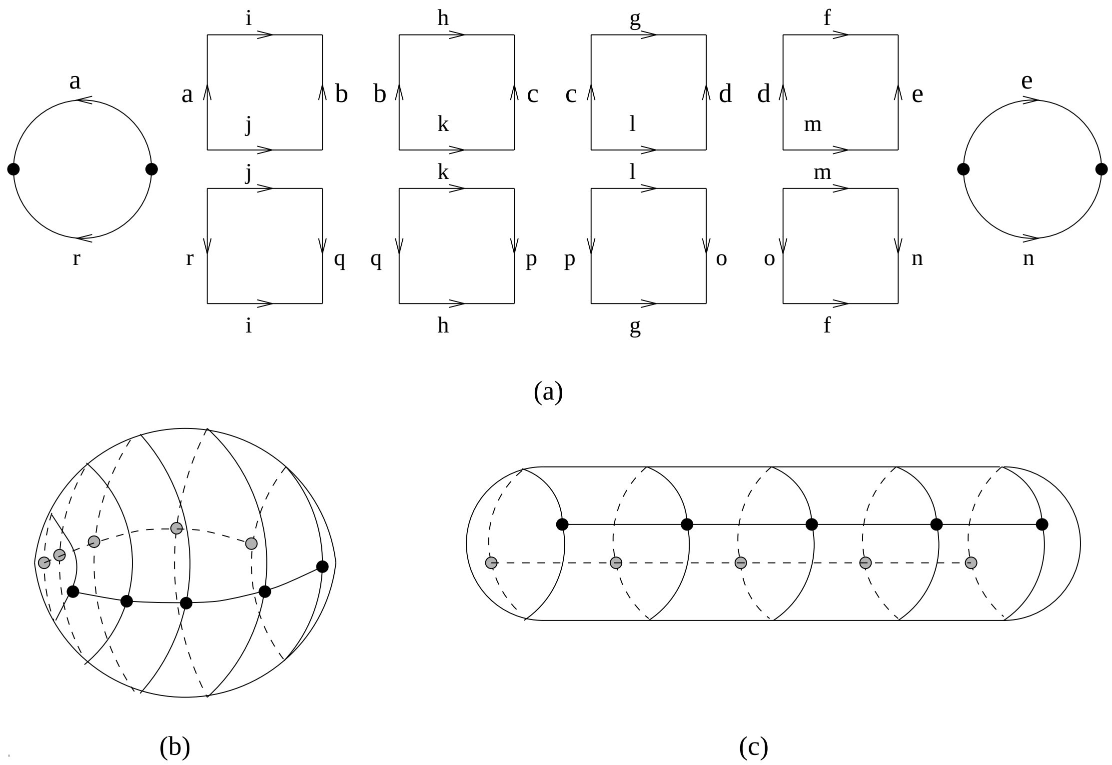


Figura 1.5: Um complexo celular (a), e duas representações (b) e (c).

A fim de quantificar estes critérios, definimos uma função numérica da geometria da superfície, chamada de *energia*, e que mede as características indesejáveis da mesma: curvatura, auto-intersecções, sobreposições, variância do tamanho dos elementos, etc. Uma vez definida esta função, o problema passa a ser determinar, dentre todas as representações geométricas do complexo dado, aquela que tem energia mínima.

1.4 Trabalhos relacionados

O problema de desenho "bonito" de grafos no plano [4] tem recebido bastante atenção devido ao grande número de aplicações: projeto de VLSI, bancos de dados, algoritmos de animação, linguagens visuais, gerência de redes, ferramentas para desenvolvimento de software, etc. [22]. Apesar do nosso problema não ser o mesmo que o de desenho de grafos em 2 ou 3 dimensões, a solução apresentada usa técnicas semelhantes às já usadas com sucesso nesta área; e, por outro lado, espera-se que algumas das idéias, aqui abordadas, possam vir a ser úteis também para a visualização de tais grafos.

Este trabalho tem grandes semelhanças com certas técnicas de relaxação usadas em projeto industrial, para o amaciamento de juntas e cantos ("fairing" e "filleting") em modelos de sólidos [20, 30]. Em Moreton [30, 29], dada uma forma inicial para a superfície, o ajuste entre os retalhos polinomiais é feito usando-se técnicas padrões de otimização não-linear baseadas no método do gradiente. Entretanto, nessas aplicações o objetivo é obter uma superfície suave interpolando certas curvas com forma e posição dadas. Portanto, esses trabalhos usam uma função energia mais simples que a usada aqui, não têm que lidar com topologias muito complicadas, e partem de uma solução inicial bastante próxima à desejada.

Outro programa semelhante ao nosso trabalho é o *Surface Evolver* desenvolvido por Brakke [7]. Trata-se de um programa para o estudo da evolução de superfícies sujeitas a forças tais como: tensão superficial, forças elásticas, pressão, etc. Contudo, as energias usadas nesse trabalho foram escolhidas por sua significância matemática e física, não por seus efeitos visuais. Além disso, as energias implementadas pelo *Surface Evolver* não são suficientes para a solução dos problemas que nós estamos apresentando, já que elas dependem apenas da forma da superfície, e não do grafo desenhado sobre ela.

Nosso trabalho também está relacionado com o problema de calcular a configuração espacial de uma molécula de proteína, a partir da seqüência de seus aminoácidos [27, 33]. A relação vem do fato que, a energia "estética" aqui abordada é qualitativamente semelhante à energia físico-química que determina as posições relativas dos átomos na molécula. Ambas tem centenas de variáveis de natureza geométrica, e inúmeros mínimos locais separados por barreiras elevadas. Portanto, parece razoável supor que quaisquer técnicas de minimização que são boas para um destes problemas também são úteis para o outro.

A idéia geral de usar "funções energia" para quantificar a "feiúra" de um desenho aparentemente tornou-se popular após o artigo de Kamada [21, 11]. De fato, algumas das funções energias que nós usamos são similares às "molas" do modelo adotado por Kamada. Outras funções energia são similares a energia de dobramento, geralmente tomada como a integral do quadrado da curvatura da superfície, que é freqüentemente a única energia considerada nos estudos sobre superfícies minimais [7, 17, 18, 20, 30, 31].

Capítulo 2

Complexos celulares

2.1 Conceitos fundamentais

Antes de mais nada, precisamos definir precisamente o conceito de complexo celular. Vamos supor conhecidos os seguintes conceitos elementares de topologia: *espaço topológico*, *conjunto aberto e fechado*, *subespaço*, *vizinhança*, *ponto de acumulação*, *limite*, e *homeomorfismo* (ou *eqüivalência topológica*). As definições podem ser encontradas no apêndice A, ou em textos introdutórios de topologia de conjuntos [2, 34, 24, 36].

2.1.1 Variedade

Um espaço topológico S é *compacto* se toda seqüência infinita de pontos de S tem um ponto de acumulação em S.

Uma *variedade topológica bidimensional* é um espaço topológico S no qual todo ponto tem uma vizinhança que é homeomorfa a $\mathbb{R}^2$.

Informalmente, uma variedade topológica bidimensional compacta é um espaço topológico equivalente a um subconjunto do $\mathbb{R}^n$ (para algum n) topologicamente fechado e de diâmetro finito, e que é bidimensional ao redor de qualquer de seus pontos. A esfera $\mathbb{S}_2$, a superfície de um toro, e o plano projetivo são exemplos típicos.

Por outro lado, a superfície de um cilindro de comprimento infinito não é compacta (ela contém seqüências infinitas de pontos cujo limite não está na superfície). Pelo mesmo motivo, um hemisfério (sem os pontos do equador) ou uma fita de Möbius (sem os pontos da borda) também não são compactos. Se incluirmos os pontos da borda, estes conjuntos tornam-se compactos, mas deixam de ser variedades (pois um ponto na borda não tem nenhuma vizinhança que seja um disco).

No que segue, usaremos o termo *variedade* para significar *variedade topológica bidimensional compacta*.

2.1.2 Definição de complexo celular

Informalmente, um complexo celular é um grafo desenhado sobre uma variedade, de tal forma que toda face seja um pedaço "simples" da mesma, sem furos ou alças. Mais formalmente:

Definição 2.1 *Um* segmento aberto *de um espaço topológico S é um subespaço de S homeomorfo à reta real $\mathbb{R}$.*

Definição 2.2 *Um* disco aberto *de S é um subespaço de S homeomorfo ao plano $\mathbb{R}^2$.*

Definição 2.3 *Um* complexo celular *é uma partição de uma variedade numa coleção finita de* elementos*, cada um dos quais é um ponto* (vértice)*, um segmento aberto* (aresta)*, ou um disco aberto* (face) *da variedade.*

Se C é um complexo celular, denotaremos o conjunto de seus vértices por $\mathbb{V}C$, o das arestas por $\mathbb{E}C$, e o das faces por $\mathbb{F}C$.

Dizemos que C é *conexo* se a variedade subjacente é conexa. As *componentes conexas* de C correspondem de maneira óbvia às componentes da variedade subjacente. Neste trabalho consideramos apenas complexos celulares conexos; para visualizar um complexo geral com as nossas ferramentas, é necessário aplicá-las a cada componente conexa separadamente.

2.1.3 Modelo de colagem

Vamos definir agora o *modelo de colagem* de um complexo celular que será adotado por nós neste trabalho.

Um modelo de colagem consiste de:

- um conjunto $P = \{p_1, p_2, \ldots, p_k\}$ de polígonos, cada um dos quais é um disco fechado, cuja fronteira é dividida num número finito de segmentos (lados) e pontos (cantos);

- uma função bijetora contínua ϕ, cujo o domínio e imagem são a união de todos os lados desses polígonos. A função ϕ deve ser sua própria inversa $\phi^{-1} = \phi$. Para todo lado e, deve-se ter $\phi(e) \neq e = \phi(\phi(e))$, e a restrição de ϕ ao lado e deve ser um homeomorfismo entre e e $\phi(e)$.

Dizemos que um modelo de colagem, como o acima, `representa` um complexo celular C sobre uma variedade S se existir uma função contínua $\sigma : \cup_i p_i \to S$ que mapeia cada polígono p_i no fecho de uma face distinta f_i de C, cada lado de p_i numa aresta de C incidente a essa face, e cada canto de p_i num vértice incidente a essa face, tal que σ é um homeomorfismo quando restrita ao interior de p_i, e tal que $\sigma(c_1) = \sigma(c_2)$ se e somente se $\phi(c_1) = (c_2)$ para dois pontos quaisquer c_1, c_2 nos lados de P.

Em topologia prova-se que todo complexo celular pode ser representado por um modelo de colagem, e vice-versa.

2.1.4 Funções de percurso

Para facilitar a descrição de algoritmos que trabalham com complexos celulares é conveniente definir funções que permitem "caminhar" no mapa, de um elemento para os elementos vizinhos. Neste trabalho usaremos as funções definidas em [16], com pequenas modificações.

Orientação longitudinal

Seja e uma aresta de um complexo celular. Existem exatamente duas maneiras de definir uma *orientação longitudinal* sobre e, que são as duas maneiras de ordenar os pontos de e de forma compatível com sua topologia.

Podemos indicar uma destas direções por uma seta sobre qualquer ponto da aresta. Outra maneira de visualizar a orientação de uma aresta é imaginar um observador minúsculo andando sobre a superfície, ao longo da aresta: a direção longitudinal diz em que sentido ele está caminhando. Veja a figura 2.1.

Orientação transversal

Da mesma forma, para cada aresta e de um complexo celular existem exatamente duas maneira de definirmos uma *orientação transversal* sobre a, ou seja, duas maneiras possíveis de definir o "lado esquerdo" e o "lado direito" da aresta, numa vizinhança suficientemente pequena da mesma.

Novamente, imagine um observador minúsculo caminhando em uma aresta sobre uma superfície. Portanto, a orientação diz em que lado da aresta está seu pé esquerdo (ou seja, sobre que lado da superfície ele está andando). Veja a figura 2.1.

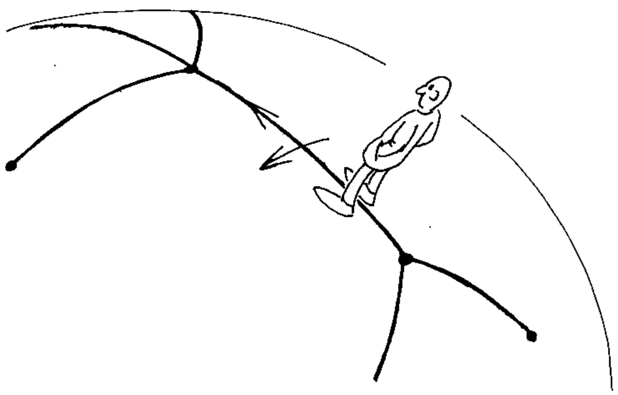


Figura 2.1: Orientações longitudinal e transversal de uma aresta [12].

Arco

Portanto, cada aresta do complexo tem exatamente quatro combinações de orientação longitudinal e transversal possíveis. Note que isto é verdade mesmo que os dois extremos da aresta incidam no mesmo vértice, ou que os dois lados da aresta pertençam à mesma face. A partir de agora, chamaremos de *arco* do complexo a uma aresta com uma determinada orientação longitudinal e transversal.

Definindo funções de percurso

Para um arco e, podemos definir sem ambigüidade (graças à sua orientação longitudinal) o *vértice de origem* $Org(e)$ e o *vértice de destino* $Dest(e)$. Analogamente, usando a orientação transversal, podemos definir a *face esquerda* $Left(e)$ e a *face direita* $Right(e)$ da aresta. Note que podemos ter $Org(e) = Dest(e)$, ou $Left(e) = Right(e)$.

Se e é uma aresta orientada, denotaremos por $Sym(e)$ a *aresta com orientação simétrica*, que consiste da mesma aresta e, mas com as duas orientações (longitudinal e transversal) invertidas. Note que apesar de termos invertido as orientação transversal, continuamos do mesmo lado da superfície.

Denotaremos também por $Flip(e)$ o resultado de inverter apenas a orientação transversal da aresta orientada e, mantendo a orientação longitudinal. Neste caso, nosso observador minúsculo passa para o outro lado da superfície, mas continua andando no mesmo sentido ao longo da aresta. Veja a figura 2.2.

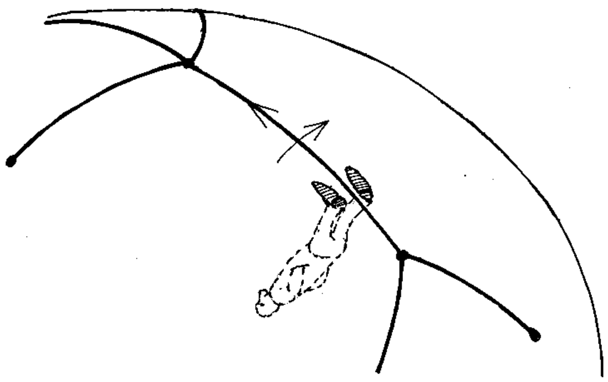


Figura 2.2: Resultado de um *Flip* aplicada sobre a aresta e da figura 2.1 [12].

Seja v a origem de um arco e. A orientação transversal de e, considerada em pontos arbitrariamente próximos a v, define um sentido de rotação em torno de v. Considere todos os arcos com origem v que definem o mesmo sentido de rotação. Podemos distinguir o *próximo arco com mesma origem que* e, $ONext(e)$, como sendo o arco seguinte a e no sentido anti-horário da orientação transversal de e.

As funções Sym e $Flip$ são suas próprias inversas; e a inversa da função $ONext$ é $Flip\,ONext\,Flip$. Esta última, chamada $OPrev(e)$, devolve o *arco anterior com mesma origem* que o argumento, no sentido horário da orientação transversal de e.

A partir das funções acima, podemos definir o *próxima arco com mesma face esquerda* e o *arco anterior com mesma face esquerda* respectivamente por: $LNext(e) = OPrev(Sym(e))$ e $LPrev(e) = Sym(ONext(e))$. Informalmente, $LNext(e)$ e $LPrev(e)$ são os arcos que seguem e precedem e, respectivamente, quando percorremos a fronteira da face $Left(e)$ no sentido que concorda com a orientação longitudinal de e. A figura 2.3 ilustra estas e outras funções que serão definidas nas próximas seções.

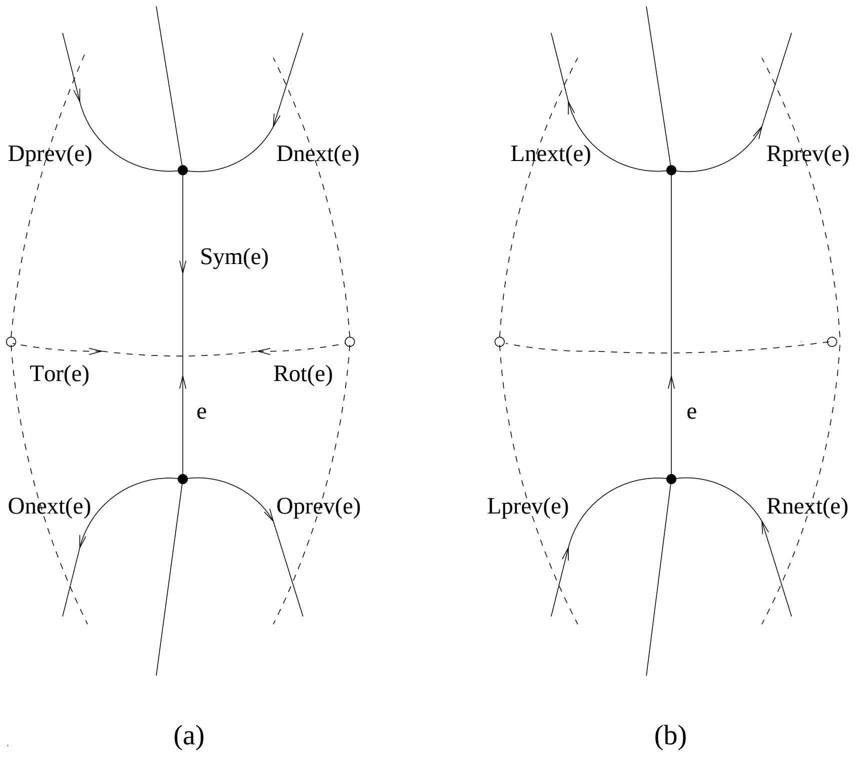


Figura 2.3: As funções de percurso.

2.1.5 Subdivisão baricêntrica

Informalmente, a *subdivisão baricêntrica* de um complexo celular C é obtida inserindo-se um novo vértice, v_e, no meio de cada aresta de C; um novo vértice, v_f, no meio de cada face f; e ligando-se este último a todos os vértices (originais e novos) na fronteira dessa face f, por novas arestas dentro da face f. Veja a figura 2.4.

Mais formalmente, uma subdivisão baricêntrica de C é outro complexo celular R, mais refinado, tal que

- Toda aresta de R incide em dois vértices distintos;
- Toda face de R incide em três vértices distintos e três arestas distintas;
- Todo vértice de C é um vértice de R;
- Toda aresta e de C é a união de um vértice v_e e duas arestas e', e'' de R incidentes em v_e;
- Toda face f de C é a união de um vértice v_f de R, e de um número par de arestas e outras tantas faces de R, todas distintas e incidentes em v_f.

Note que os vértices de R podem ser divididos em três tipos: os que são vértices de C (vértices primais, tipo `V`), os que pertencem a arestas de C (vértices mediais, tipo `E`), e os que pertencem a faces de C (vértices duais, tipo `F`).

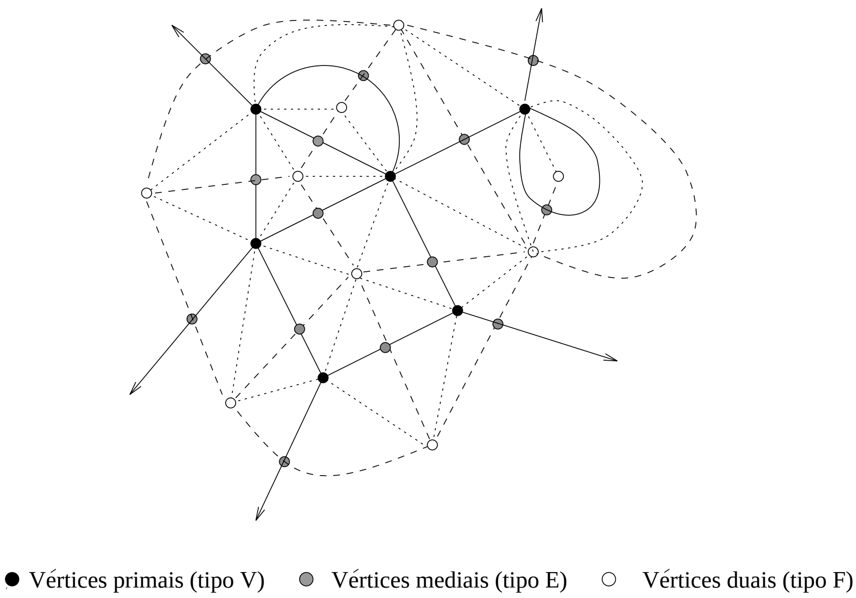


Figura 2.4: A subdivisão baricêntrica de um complexo.

Note também que cada aresta de R liga dois vértices de tipos diferentes; portanto, temos três tipos de arestas, `VE`, `VF`, e `EF`. As arestas de tipo `VE` são partes de arestas de C; as demais estão contidas nas faces de C. Finalmente, note que cada face de R é incidente a exatamente um vértice de cada tipo.

Com isso, podemos afirmar que uma face de R é homeomorfa a um triângulo planar, onde cada canto representa um vértice e cada lado representa uma aresta de R.

2.1.6 Complexo celular abstrato

Informalmente, um isomorfismo ϕ entre dois complexos celulares é uma bijeção entre os elementos desses complexos que preserva todas as relações de incidência e adjacência.

Mais precisamente, sejam A e B dois complexos celulares, e $ONext_A$ e $ONext_B$ as respectivas funções de percurso. Dizemos que A é isomorfo a B se existe uma bijeção ϕ dos arcos de A para os arcos de B, tal que $ONext_A(\phi(e)) = \phi(ONext_B(e))$ para todo arco e. Note que esta condição estabelece também uma bijeção entre os vértices e faces dos complexos, que podem ser identificados com as órbitas de $\{ONext, Flip\}$ e de $\{LNext, Sym \circ Flip\}$, respectivamente.

Um *complexo celular abstrato* é definido como sendo uma classe de equivalência de complexos celulares, isto é, o conjunto de todos os complexos celulares que são isomorfos a um complexo dado.

2.1.7 Complexos duais

Informalmente, o *dual* de um complexo celular C é um complexo C^* que tem as mesmas relações de incidência e adjacência que C, exceto que os papéis dos vértices e faces estão trocados.

Mais formalmente, dois complexos celulares C e C^*, numa mesma superfície S, são *duais estritos* se toda face de C contém um único vértice de C^*, e vice-versa, e os dois complexos admitem uma subdivisão baricêntrica comum R.

Neste caso, forçosamente, os vértices de C^* são os vértices de tipo `F` de R; e cada aresta de C^* consiste de um vértice de tipo `E` de R, mais as duas arestas de R de tipo `EF` incidentes a esse vértice.

A figura 2.5 mostra um complexo C (linha cheia) e seu dual (linha tracejada). Note que cada aresta não dirigida de C cruza somente sua respectiva aresta dual, e que cada vértice primal está na sua correspondente face dual; e vice-versa.

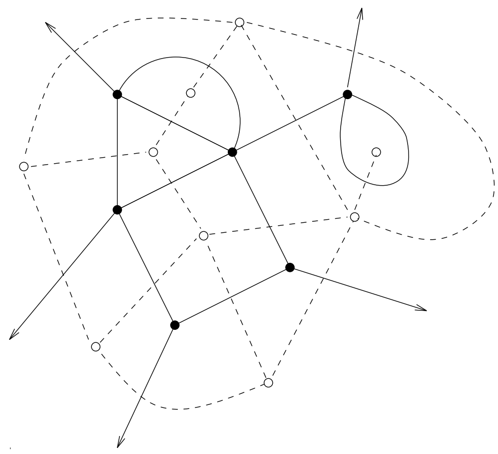


Figura 2.5: Um complexo C (linha cheia) e um dual estrito C^* do mesmo (linha tracejada).

Podemos notar que para todo complexo existem infinitos complexos que são seus duais estritos. Entretanto, prova-se que todos esses duais são isomorfos entre si; e, mais ainda, se o complexo A é isomorfo a B, todo dual estrito de A é isomorfo a todo dual estrito de B. Portanto, todo complexo abstrato C tem um único complexo abstrato dual C^*.

2.1.8 Duais de arestas dirigidas e orientadas

Seja C um complexo celular, e C^* um dual estrito de C. A cada aresta orientada e dirigida e de C existe uma única aresta orientada e dirigida de C^* que cruza e da direita para a esquerda e é cruzada por e da esquerda para a direita. Vamos denotar essa aresta orientada e dirigida por $Rot(e)$.

Informalmente, $Rot(e)$ é obtida rodando-se e 90° no sentido anti-horário, ao redor do ponto de cruzamento da aresta e com sua aresta dual (do ponto de vista do observador que anda sobre a aresta). Veja a figura 2.5.

Como a relação entre C^* e C é simétrica, esta definição também vale para as arestas de C^*. Ou seja, se e é um arco de C^*, $Rot(e)$ é um arco de C que cruza e da direita para a esquerda.

Segue-se daí que $Rot(Rot(e)) = Sym(e)$, para todo arco de C ou C^*; e, portanto, $Rot^4(e) = e$ para toda aresta e. A inversa de Rot, pode ser definida como $Rot^{-1}(e) = Rot(Rot(Rot(e)))$, e será chamada por nós de $Tor(e)$.

A partir das funções acima, podemos definir, para uma dada aresta e, as seguintes funções (veja a figura 2.3):

$RNext(e)$ é o primeiro arco encontrada ao avançarmos para a aresta seguinte a e na fronteira da face $Right(e)$, no sentido anti-horário: $RNext(e) = Tor(ONext(Rot(e)))$.

$RPrev(e)$ é o primeiro arco encontrada ao avançarmos para a aresta anterior a e na fronteira da face $Right(e)$, no sentido horário: $RPrev(e) = Tor(OPrev(Rot(e)))$.

$DNext(e)$ é o primeiro arco encontrada ao avançarmos para a aresta seguinte a e, ao redor de $Dest(e)$, no sentido anti-horário: $DNext(e) = Sym(ONext(Sym(e)))$.

$DPrev(e)$ é o primeiro arco encontrada ao avançarmos para a aresta anterior a e, ao redor de $Dest(e)$, no sentido horário: $DPrev(e) = Sym(OPrev(Sym(e)))$.

É fácil ver que, a partir de qualquer arco inicial e, podemos atingir todas os arcos orientados na componente conexa do complexo que contém e, usando combinações apropriadas de *Flip*, *Rot*, e *ONext*.

Outra observação a ser feita é que, todas as funções já definidas até aqui (exceto *Flip*), podem ser obtidas por combinações de *Rot* e *ONext*, pois

$$\begin{aligned}
OPrev(e) &= Rot(ONext(Rot(e))) \\
LNext(e) &= Rot(ONext(Rot^3(e))) \\
LPrev(e) &= Rot^2(ONext(e)) \\
RNext(e) &= Rot^3(ONext(Rot(e))) \\
RPrev(e) &= ONext(Rot^2(e)) \\
DNext(e) &= Rot^2(ONext(Rot^2(e))) \\
DPrev(e) &= Rot^3(ONext(Rot^3(e)))
\end{aligned}$$

2.1.9 A estrutura *quad-edge*

Encontram-se na literatura muitas estruturas de dados destinadas a representar a topologia de complexos celulares [15, 10, 23, 35]. De modo geral, todas estas estruturas usam um registro para cada elemento do complexo (vértice, aresta, ou face), com apontadores para os elementos adjacentes. Assim, as funções de percurso da seção 2.1.4 se reduzem a seguir apontadores; e modificações locais na topologia do complexo correspondem a modificações locais nos nós e apontadores da estrutura.

Neste trabalho, decidimos adotar a estrutura de dados *quad-edge* [16, 25]. Essa estrutura é similar às estruturas *winged edge* e *half-edge* [5], amplamente usadas em CAD e computação gráfica; mas tem sobre elas a vantagem de permitir a representação de complexos celulares não-orientáveis, como a garrafa de Klein ou o plano projetivo.

Além disso, a estrutura quad-edge permite também representar complexos celulares com vértices de grau 1 (figura 2.6(a)), múltiplas arestas unindo o mesmo par de vértices (fig. 2.6(b)), arestas com origem e destino no mesmo vértice (fig. 2.6(c)), e faces incidentes múltiplas vezes ao mesmo vértice (fig. 2.6(d)). Na verdade, esta generalidade é também uma desvantagem, pois exige cuidados especiais na modelagem da superfície, como será visto mais adiante

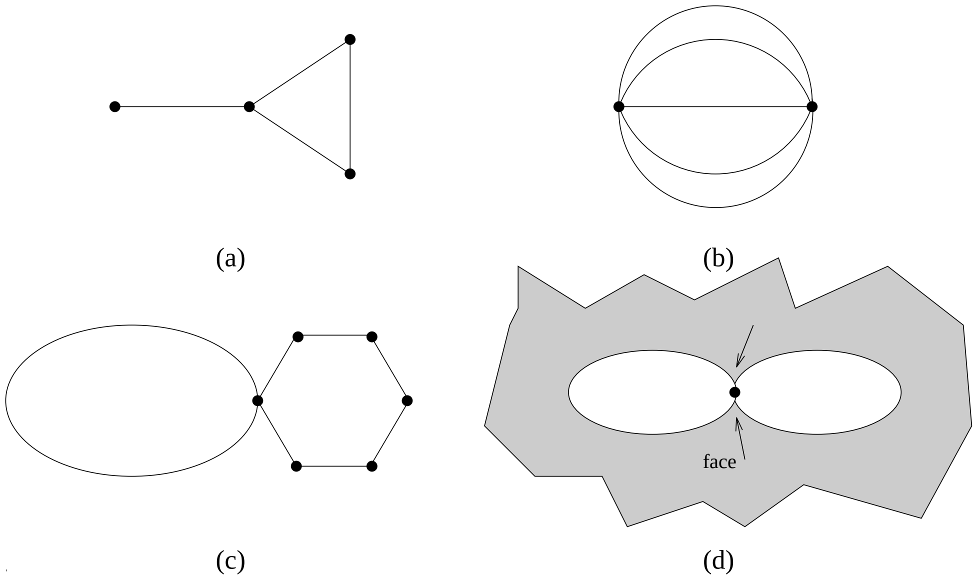


Figura 2.6: Alguns Complexos celulares representáveis pela *quad-edge*

Por outro lado, a estrutura quad-edge exige que toda face seja topologicamente equivalente a um disco, sem buracos. Esta restrição é às vezes inconveniente, por necessitar a introdução de arestas sem significado geométrico; mas simplifica bastante a representação e o manuseio da estrutura.

Na estrutura *quad-edge*, cada aresta e do complexo celular é representada por um grupo de oito *arcos*, que correspondem às quatro orientações da aresta e, e às quatro orientações da aresta dual.

Para construir a estrutura de dados, nós selecionamos arbitrariamente uma arco canônico em cada grupo. Desta forma, qualquer arco e pode ser escrita como $Flip^f(Rot^r(e_{00}))$, onde $r \in \{0, 1, 2, 3\}$, $f \in \{0, 1\}$, e e_{00} é a arco canônico do grupo a que e pertence.

Assim, as oito possíveis orientações de e podem ser representadas na estrutura de dados por um *registro* a, dividido em quatro partes, a[0], ..., a[3] (a[r] corresponde a aresta $Rot^r(e_{00})$ e guarda uma referência para o $ONext(Rot^r(e_{00}))$. Uma aresta genérica $e = Flip^f(Rot^r(e_{00}))$ é então representada pela tripla (a, r, f).

Para qualquer arco $e_{rf} = Flip^f(Rot^r(e_{00}))$ pertencente ao grupo e, é possível calcular $ONext(e_{rf})$ a partir dos campos a[i], através das identidades:

$$ONext(Flip(a)) = Flip(OPrev(a))$$

e

$$OPrev(a) = Rot(ONext(Rot(a)))$$

Na figura 2.7, a parte (a) mostra a representação de um registro para uma aresta, (b) mostra um complexo celular e (c) a estrutura de dados para o complexo (b).

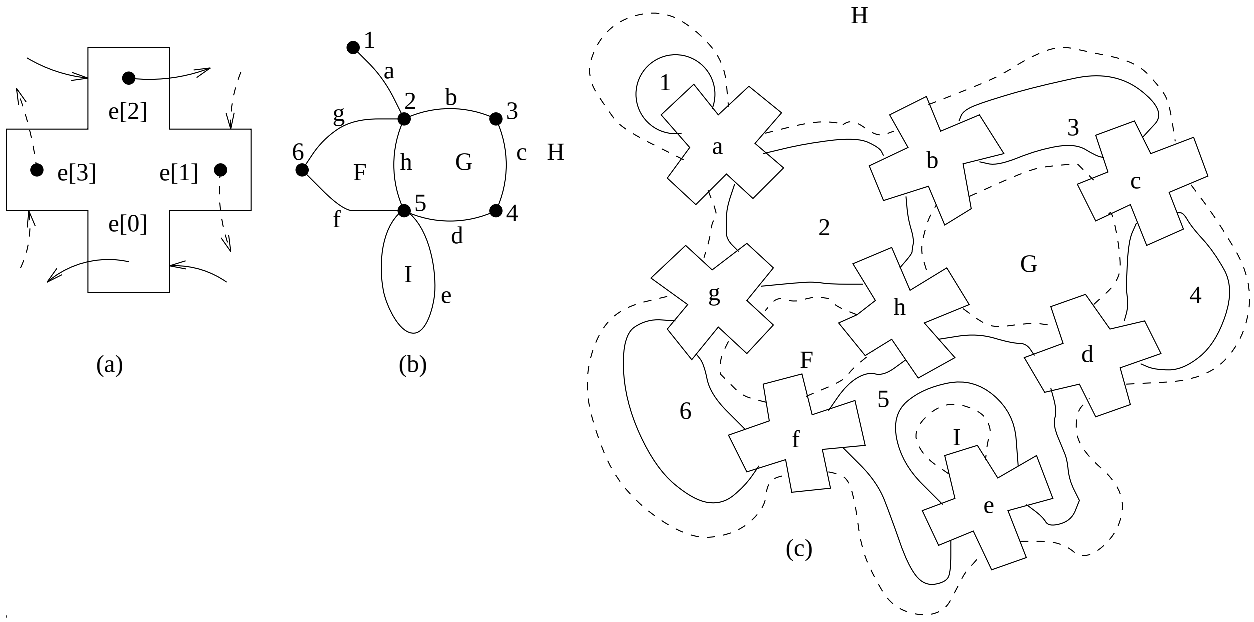


Figura 2.7: Um complexo celular e sua respectiva representação na estrutura *quad-edge*

Capítulo 3

Modelagem geométrica do complexo

Uma maneira óbvia de modelar a geometria da superfície seria modelar cada face do complexo por um retalho polinomial, implícito ou paramétrico, com restrições de continuidade entre retalhos adjacentes [3, 26].

Entretanto, as faces do complexo podem ter número arbitrário de lados, colados entre si de maneira arbitrária. Em geral, não é possível representar a topologia destas colagens com um único retalho polinomial de grau limitado.

3.1 Ladrilhamento do complexo

Para contornar essa dificuldade, introduzimos o conceito de *ladrilhamento* de um complexo celular C, que é uma subdivisão da superfície de C em retalhos mais simples, os *ladrilhos*. Dada uma subdivisão baricêntrica de C, R, podemos definir o ladrilho de uma aresta e de C como sendo a união de quatro faces triangulares de R que têm uma das duas arestas `VE` de R, que são partes da aresta e, como um de seus lados.

Topologicamente, o ladrilhamento substitui o complexo celular original por um novo complexo celular $\square C$ sobre a mesma superfície, cujas faces tem todas quatro arestas.

Assim, cada ladrilho pode ser visto como um quadrilátero, colado por seus lados a outros quatro ladrilhos (possivelmente a si próprio). Os vértices de um ladrilho são alternadamente vértices de C e de C^* (o dual estrito de C). Existe exatamente um ladrilho $ladr(e)$ para cada aresta e do complexo. A aresta e e sua dual e^* são as duas diagonais do ladrilho. Veja a figura 3.1.

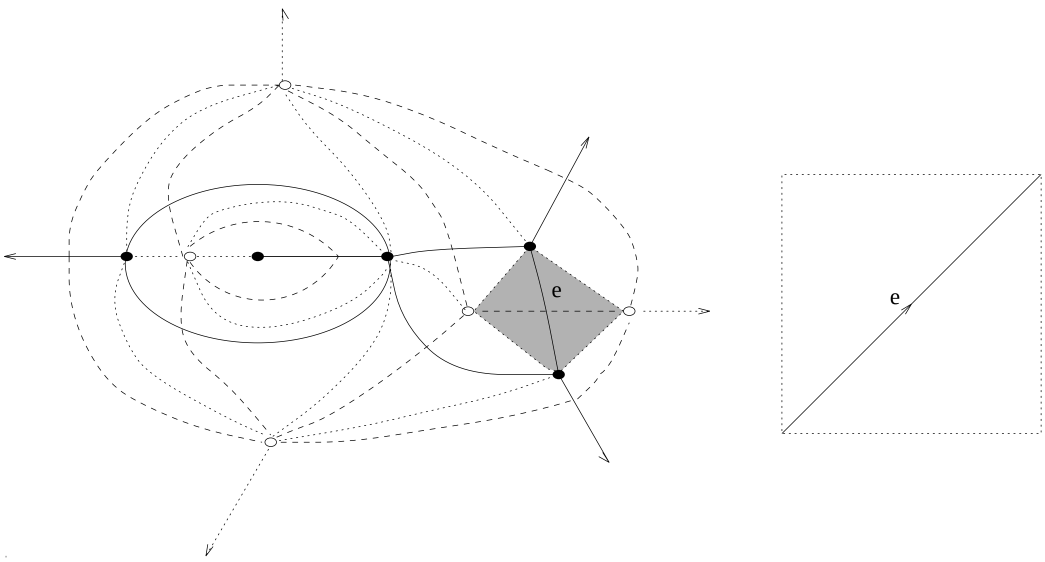


Figura 3.1: Um complexo celular bidimensional, mostrando arestas primais (linhas cheias), arestas duais (linhas tracejadas), a divisão em ladrilhos (linhas pontilhadas) e um dos ladrilhos (em cinza).

Portanto, em vez de considerar a superfície como uma união de faces, ela é considerada como uma união de ladrilhos. Uma vez que cada ladrilho tem apenas quatro lados, e portanto quatro ladrilhos vizinhos, ele pode ser modelado em princípio por um objeto geométrico de complexidade finita. O problema reduz-se então à atribuir geometria a cada ladrilho, garantido continuidade (suavidade), não interferência, e outras qualidades visuais, como veremos no cap. 4.

3.1.1 Modelagem por retalhos polinomiais

Em computação gráfica, superfícies curvas são quase que universalmente representadas por superfícies polinomiais paramétricas. Portanto, é natural pensarmos em representar cada ladrilho por um retalho quadrangular de superfície deste tipo; ou seja, pelo conjunto

$$\{ (x(u,v), y(u,v), z(u,v)) \ : \ u, v \in [0,1] \}$$

onde x, y, e z são polinômios nos parâmetros u e v. Em particular, quando os polinômios são de grau 3, temos os *retalhos de Bézier* [14, 13].

Infelizmente, este modelo geométrico introduz algumas dificuldades bastante sérias. Em primeiro lugar, observe que, para que a superfície toda seja contínua, precisamos escolher os polinômios de tal forma que os retalhos de ladrilhos adjacentes estejam colados no espaço; ou seja, que os valores das funções x, y, e z de um retalho sejam iguais aos valores das funções do outro retalho ao longo de sua fronteira comum.

Além disso, se quisermos que a superfície seja suave, sem pontas ou quinas, precisamos garantir a igualdade também do plano tangente ao longo dessa fronteira. [14, 26]. Para garantir que estas restrições tenham solução, qualquer que seja a topologia do complexo, precisamos usar polinômios de grau bastante elevado—pelo menos 5 em cada um dos parâmetros u e v.

Estas restrições de continuidade e suavidade se traduzem em restrições algébricas complicadas sobre os coeficientes dos polinômios em questão. Por exemplo, a continuidade do plano tangente, entre dois retalhos vizinhos de grau 5, equivale a 15 equações, envolvendo uma centena de determinantes 3x3 distintos [30, 29].

Finalmente, precisamos também excluir singularidades geométricas (cúspides) no interior de cada retalho. Esta condição corresponde a exigir que três polinômios de grau 9 em u e v não se anulem ao mesmo tempo.

Como se pode ver, é bastante difícil manter todas estas restrições satisfeitas durante o processo de otimização da energia. Sem mencionar o fato que as funções de energias mais importantes, como a integral da curvatura (veja seção 4.3) são bastante difíceis de calcular a partir dos coeficientes dos polinômios.

3.1.2 Modelagem por triângulos

Em vista destas dificuldades, decidiu-se modelar cada ladrilho por uma superfície poliédrica; especificamente, por uma malha de $k \times k$ quadriláteros, cada um dos quais dividido em quatro triângulos planos. Veja a figura 3.2.

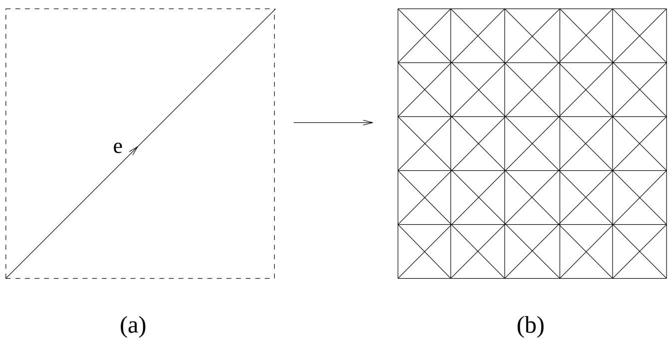


Figura 3.2: O ladrilho de uma aresta e e a malha 5×5 correspondente.

Desta forma, substituímos o complexo celular original C, com m arestas, por um complexo mais refinado, uma *triangulação* $\Delta_k(C)$ da superfície, com $6mk^2$ arestas e $4mk^2$ faces triangulares. Cada uma destas faces será modelada por um triângulo geométrico — plano, com lados retilíneos.

Ao adotarmos este modelo, temos que desistir da condição de continuidade do plano tangente; em vez disso, nos contentamos em minimizar os ângulos entre triângulos vizinhos.

O valor apropriado de k depende da estrutura do complexo. Para os complexos que usamos nos nossos testes, conseguimos resultados satisfatórios com $k \leq 5$.

3.2 O algoritmo de ladrilhamento

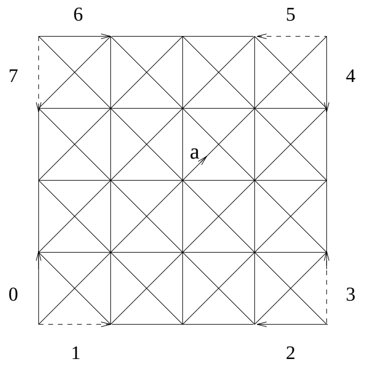
O algoritmo de ladrilhamento divide-se em três etapas: a *construção* dos ladrilhos, a *colagem* dos mesmos, e a *eliminação* de triângulos degenerados. A seguir, vamos discutir detalhadamente cada etapa deste algoritmo.

3.2.1 Construção dos ladrilhos

Na etapa de construção, montamos para cada aresta e do complexo C dado, um ladrilho triangulado, como ilustrado na figura 3.2. Este ladrilho tambem é modelado pela estrutura quad-edge. Note que o exterior do ladrilho é modelado por uma face adicional completando uma esfera. Essa face é eliminada automáticamente pelo processo de colagem (veja seção 3.2.2).

Uma diagonal do ladrilho, consistindo de $2k$ arestas da triangulação, representa a aresta e do complexo original; e a outra diagonal representa a aresta dual de e. Os quatro cantos do ladrilho representam, alternadamente, vértices de C e centros de faces de C.

Nesta etapa, construímos uma tabela que associa, a cada arco a do complexo original C, um arco $Corner(a)$ do ladrilho correspondente, como ilustrado na figura 3.3.



0 $Corner(a)$ 4 $Corner(Sym(a))$

1 $Corner(Flip(a))$ 5 $Corner(Flip(Sym(a)))$

2 $Corner(Rot(a))$ 6 $Corner(Tor(a))$

3 $Corner(Flip(Rot(a)))$ 7 $Corner(Flip(Tor(a)))$

Figura 3.3: A triangulação de um ladrilho, e a função *Corner*.

Note que as arestas representadas com linha tracejada estão no lado de baixo do ladrilho. Especificamente, se $\delta_1, \delta_2, \ldots, \delta_{2k}$ é a seqüência de arcos na diagonal do ladrilho que representam o arco a, então $Corner(a) = ONext(\delta_1)$. Definimos também a função $CCorner(a) = OPrev(\delta_1) = Flip(Corner(Flip(a)))$.

3.2.2 Colagem dos ladrilhos

Na etapa de colagem, unimos o ladrilho de cada arco a do mapa original com o ladrilho do arco $b = ONext(a)$, juntando pares de vértices correspondentes ao longo dos lados apropriados, e eliminando arestas paralelas. Veja a figura 3.4.

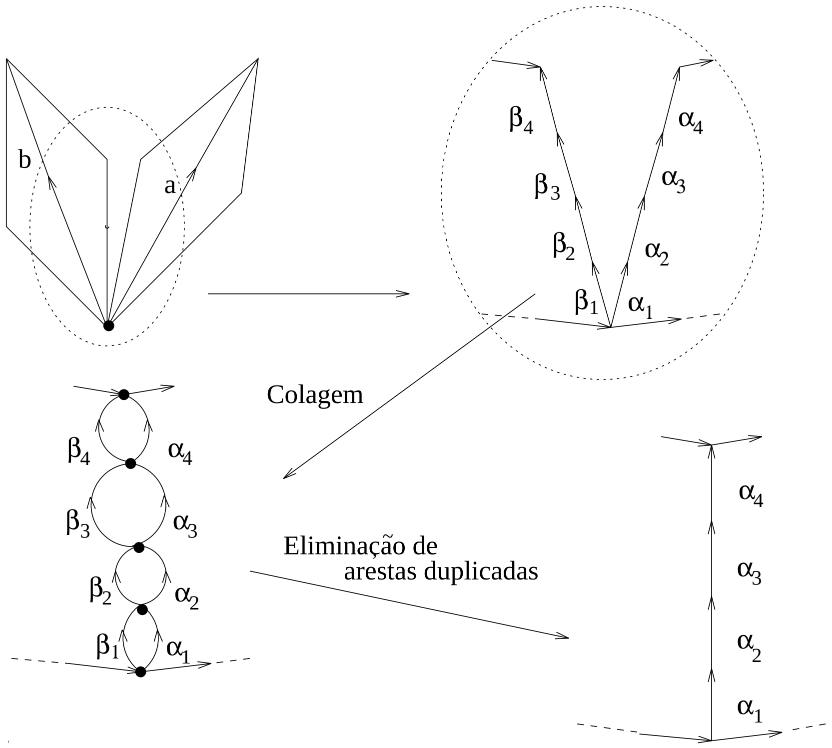


Figura 3.4: Representação da colagem de dois ladrilhos adjacentes

Mais precisamente, as arestas a serem identificadas são $\alpha_1, \alpha_2, \ldots, \alpha_k$ no ladrilho de a, e $\beta_1, \beta_2, \ldots, \beta_k$ no ladrilho de b, onde $\alpha_1 = Corner(a)$, $\beta_1 = CCorner(b)$, e e

$$\begin{aligned} \alpha_i &= OPrev(Sym(\alpha_{i-1})) \\ \beta_i &= ONext(Sym(\beta_{i-1})) \end{aligned}$$

para $i = 2, 3, \ldots, k$.

Note-se que para colar as arestas α_i e β_i temos que retirar da triangulação uma dessas duas arestas. Em nosso programa, convencionamos retirar a aresta β_i.

3.2.3 Enumeração dos pares a colar

Em princípio a colagem deve ser aplicada a todos os pares de arcos $(a, ONext(a))$ do complexo, primais e duais. Entretanto, note que, ao colarmos a com b, estamos na verdade colando os quatro pares

$$\begin{array}{rcl} a & \leftrightarrow & b \\ Flip(b) & \leftrightarrow & Flip(a) \\ Rot(b) & \leftrightarrow & Rot^{-1}(a) \\ Flip(Rot^{-1}(a)) & \leftrightarrow & Flip(Rot(b)) \end{array}$$

Portanto, se aplicamos a colagem a um destes pares, devemos evitar de aplicá-la aos outros três.

No caso de complexos orientáveis, esta condição pode ser trivialmente satisfeita: basta limitar a enumeração às arestas dirigidas e orientadas primais de um lado do complexo — isto é, todas os arcos que podem ser atingidos de um arco inicial e por aplicações múltiplas de, *ONext* e *Sym*.

Entretanto, no caso de complexos não orientáveis, esta abordagem não funciona, pois é possível atingir $Flip(a)$ por combinações de *ONext* e *Sym*. Para tratar esses casos, é necessário acrescentar um teste no início do algoritmo de colagem, que retorna imediatamente se os lados a colar já estão colados.

3.2.4 Descrição detalhada do algoritmo de colagem

Segue abaixo a descrição detalhada do processo de colagem de dois ladrilhos.

Nessa descrição usamos a operação fundamental de edição da estrutura *quad-edge*, denotada por *Splice*. Não cabe aqui descrever o efeito geral desta operação; para nossos fins, basta saber que $Splice(a, b)$ modifica os apontadores *ONext* de forma a juntar os vértices de origem das arestas dirigidas e orientadas a e b, se eles forem distintos; ou separá-los, se forem o mesmo vértice.

Mais precisamente, se antes da operação temos $ONext(a) = a'$ e $ONext(b) = b'$, depois dela teremos $ONext(a) = b'$ e $ONext(b) = a'$. Ao mesmo tempo, os apontadores *ONext* da subdivisão dual são atualizados, de forma a separar as faces $\alpha = Left(a)$ e $\beta = Left(b)$ se elas forem a mesma face; ou juntá-las, se elas forem distintas. Veja a figura 3.5. Para mais detalhes sobre o operador *Splice*, veja o artigo de Guibas e Stolfi [16].

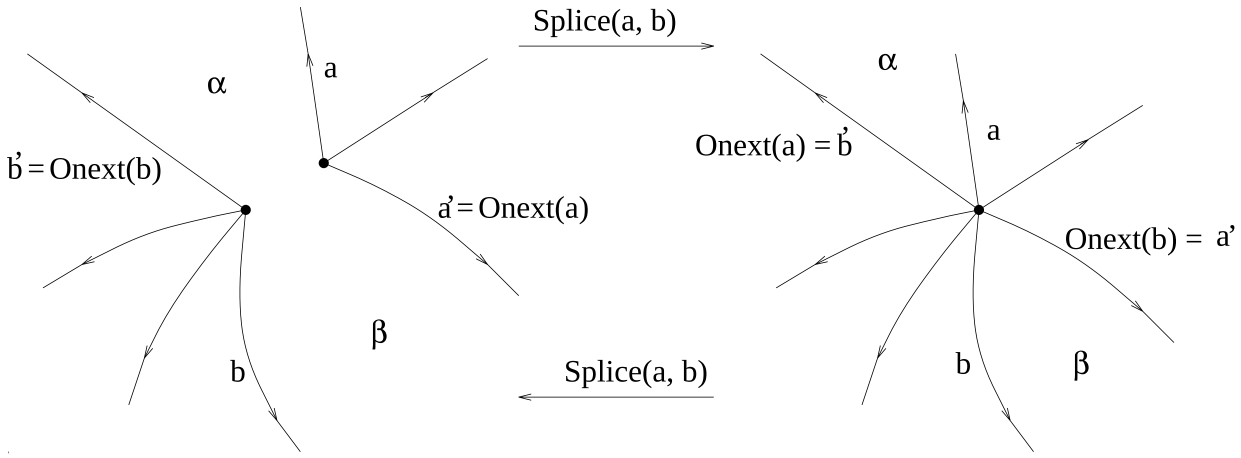


Figura 3.5: Efeito da operação *Splice* sobre duas arestas a e b.

O processo de colagem de dois ladrilhos pode ser descrito então pelo procedimento *GluePatch* abaixo. Ele usa um procedimento auxiliar *identify*, definido mais adiante.

Algoritmo 3.1 $GluePatch(a, b, n)$: cola dois lados dos ladrilhos correspondentes aos arcos de a e b do complexo C, identificando em $\Delta_k(C)$ os arcos $\alpha_1, \ldots, \alpha_n$ com os arcos $\beta_1, \ldots, \beta_n$, onde $\alpha_1 = Corner(a)$, $\beta_1 = CCorner(b)$ e $\alpha_i = OPrev(Sym(\alpha_{i-1}))$, $\beta_i = ONext(Sym(\beta_{i-1}))$; atualiza $Corner(a)$, $CCorner(b)$; elimina os triângulos degenerados.

1. Se $Corner(a) \neq CCorner(b)$, então
   1.1. Faça $\alpha_1 = Corner(a)$, $\beta_1 = CCorner(b)$.
   1.2. Para i de 2 a n faça
      1.2.1. $\alpha_i \leftarrow OPrev(Sym(\alpha_{i-1}))$.
      1.2.2. $\beta_i \leftarrow ONext(Sym(\beta_{i-1}))$.
   1.3. Para i de 1 a n, faça
      1.3.1. $Identify(\alpha_i, \beta_i)$.
   1.4. Atualize $CCorner(b)$, $Corner(Rot(b))$.
   1.5. Elimine triângulos degenerados.

Algoritmo 3.2 *Identify*(α, β): identifica os arcos α e β, removendo β.

1. Se $ONext(\alpha) \neq \beta$, então

   1.1. $Splice(\alpha, OPrev(\beta))$.

2. Se $ONext(Sym(\beta)) \neq Sym(\alpha)$, então

   2.1. $Splice(OPrev(Sym(\alpha)), Sym(\beta))$.

3. $Splice(\beta, OPrev(\beta))$.

4. $Splice(Sym(\beta),\ OPrev(Sym(\beta)))$.

3.2.5 Eliminação de triângulos degenerados

Na etapa de colagem, é possível que um ladrilho tenha que ser colado consigo mesmo. Isto ocorre em casos onde temos: $ONext(a) = a$, $ONext(a) = Sym(a)$, ou $ONext(a) = Flip(Sym(a))$. Nesse caso, o complexo refinado vai conter *pares de triângulos degenerados*: triângulos distintos incidentes nos mesmos três vértices (veja a figura 3.6).

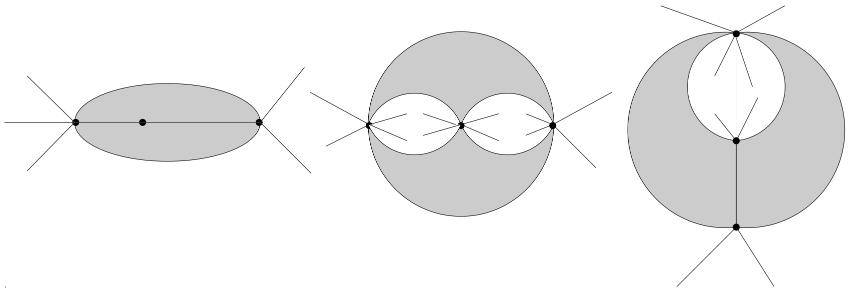


Figura 3.6: Pares de triângulos degenerados.

Pares de triângulos degenerados são indesejáveis, pois eles são sempre coincidentes, quaisquer que sejam as coordenadas atribuídas aos vértices, e resultam em descontinuidades e "abas" pendentes na superfície.

Portanto, após a colagem de todos os ladrilhos, precisamos localizar e eliminar quaisquer pares de triângulos que tenham se tornado degenerados. Este processo pode ser feito de maneira eficiente, graças ao seguinte resultado:

Teorema 3.1 *Seja $\Delta_k(C)$ uma triangulação qualquer obtida por ladrilhamento de um complexo celular C com ladrilhos de ordem k. Se $k \geq 2$, só teremos pares de triângulos degenerados se algum ladrilho tiver sido colado a si mesmo; e nesse caso os dois triângulos degenerados pertencem ao mesmo ladrilho.*

Prova: Sejam R e S dois triângulos de $\Delta_k(C)$ que depois de colados possuem três vértices comuns abc. A colagem só pode ter identificado dois desses vértices, pois pelo menos um vértice de cada triângulo é interior ao ladrilho. Portanto, R e S tinham, pelo menos, um vértice interior em comum antes da colagem. Assim, eles pertencem ao mesmo ladrilho.

Isto ainda não prova que o ladrilho foi colado consigo mesmo, pois a identificação dos vértices de R e S poderia ter acontecido por via indireta; o ladrilho L_1 que contém R e S é colado com L_2, L_2 é colado com $L_3, \ldots$, que é colado com o ladrilho $L_n = L_1$. Precisamos portanto provar que a colagem foi direta (L_1 com L_1).

Cada um desses vértices pode estar no canto do ladrilho (vértice de tipo `C`), ou num dos lados do ladrilho (vértice de tipo `L`) ou no interior do ladrilho (vértice de tipo `I`). Veja a figura 3.7.

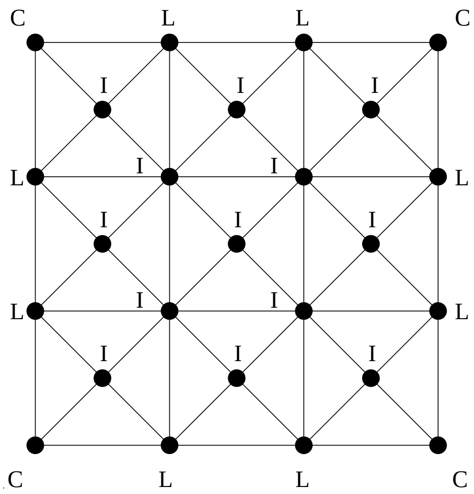


Figura 3.7: Possíveis tipos de vértices antes da colagem (C = Canto, L = Lado, e I = Interior do ladrilho).

Na verdade, só as cinco combinações de tipos ilustradas na fig. 3.8 são possíveis:

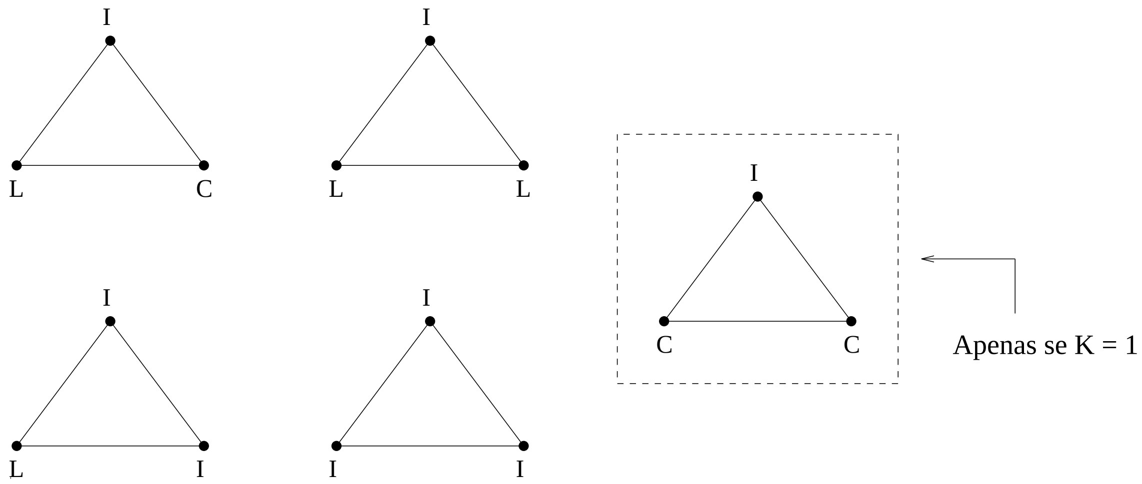


Figura 3.8: Possíveis tipos de triângulos para ladrilhos antes da colagem.

Antes da colagem, R e S tinham um ou dois vértices em comum. A colagem só identifica vértices de tipo `C` com vértices de tipo `C`, e vértices de tipo `L` com vértices de tipo `L`; portanto, a colagem deve ter identificado um ou dois pares de vértices, cada um de tipo `L-L` ou tipo `C-C`.

Na figura 3.9, temos todas as possíveis posições relativas dos triângulos R e S antes da colagem e que satisfazem estas condições. Note que, devido a topologia de nossos ladrilhos, podemos concluir que os casos (1) a (5) na figura 3.9 não ocorrem e que os casos (8) e (9) só ocorrem para $k = 1$. Assim, todas as possibilidades que restam incluem uma identificação de um vértice `L` do ladrilho com outro vértice `L` do mesmo ladrilho. Mas vértices de tipo L só são identificados aos pares, portanto isto só ocorre por colagem direta de um ladrilho consigo mesmo.

Fim da prova.

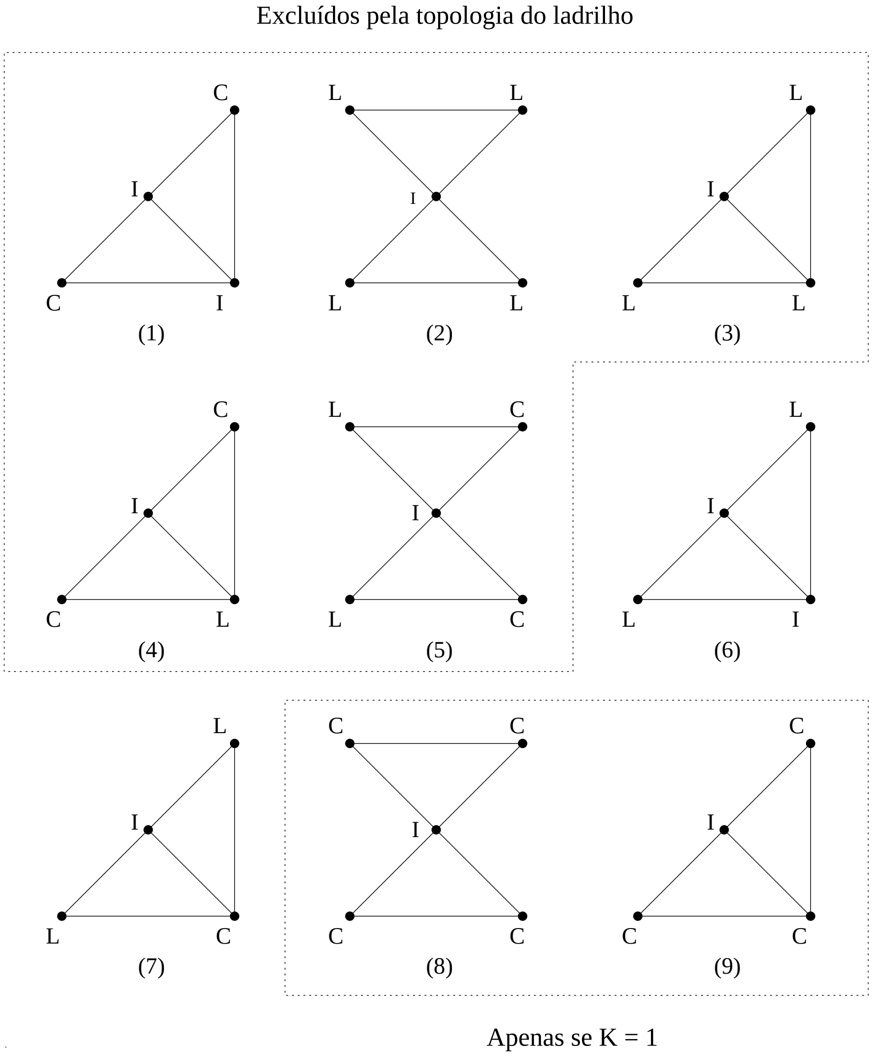


Figura 3.9: Possíveis combinações dos triângulos R e S antes da colagem.

A partir deste teorema podemos identificar facilmente os pares de triangulos degenerados criados pela colagem, para ladrilhos com ordem $k \geq 2$. Especificamente, todo par (R, S) que vai se tornar degenerado é um dos dois tipos: (i) $R = Right(\alpha_1)$ e $S = Left(\beta_1)$, ou (ii) $R = Right(DNext(\alpha_1))$ e $S = Left(DPrev(\beta_1))$; onde $\alpha_1 = Corner(e)$ e $\beta_1 = OPrev(OPrev(Corner(e)))$, para algum arco e (primal ou dual) do complexo de partida tal que $ONext(e) = e$. Veja figura 3.10.

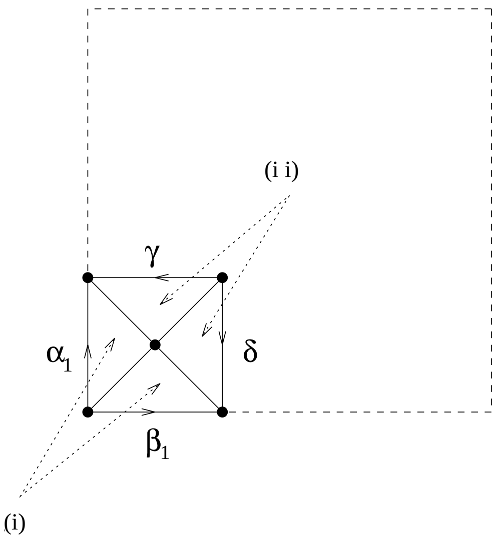


Figura 3.10: Pares de triângulos que se tornarão degenerados criados pela colagem.

Podemos concluir também o seguinte: uma vez que cada ladrilho tem 4 lados, ele pode ser colado consigo mesmo no máximo duas vezes. Portanto podemos ter, no máximo, 4 pares de triângulos degenerados por ladrilho, formando 2 grupos de 4 triângulos; e nenhuma aresta da triangulação $\Delta_k(C)$ é comum aos 2 grupos.

A operação de limpeza consiste na eliminação dos 4 triângulos indicados na figura 3.10 e a identificação das arestas $\gamma = DNext^2(\alpha_1)$ e $\delta = DPrev^2(\beta_1)$. Esta limpeza deve ser repetida para todo canto de ladrilho adjacente a dois lados que foram colados entre si.

Note que se um ladrilho contem dois grupos de triângulos degenerados, as duas arestas γ e δ de um grupo não ocorrem no outro grupo, portanto o princípio da colagem de arestas aos pares é respeitado. Note também que a limpeza não identifica novos pares de vértices que já não estejam identificados; portanto ela não introduz novos pares de triângulos degenerados.

Assim, a eliminação simultânea de todos os pares de triângulos degenerados produz uma triangulação sem nenhum par de triângulos degenerados. Portanto, a eliminação dos triângulos degenerados pode ser feita em paralelo com a colagem: sempre que colamos dois lados adjacentes $\alpha_1, \ldots, \alpha_k$ e $\beta_1, \ldots, \beta_k$ de um mesmo ladrilho, eliminamos os 4 triângulos degenerados e identificamos as arestas γ e δ.

A figura 3.11 ilustra este processo, com a triangulação (parcial) que resulta da colagem de um ladrilho de dimensão 2×2 (figura 3.11(a)) a si próprio—especificamente, da borda inferior com a borda esquerda (figura 3.11(b)).

Note que, em conseqüência da eliminação das arestas duplicadas que surgem depois da colagem, obtemos dois pares de triângulos degenerados adjacentes ao vértice u (figura 3.11(c)). A figura 3.11(d) mostra o resultado da eliminação destes 4 triângulos.

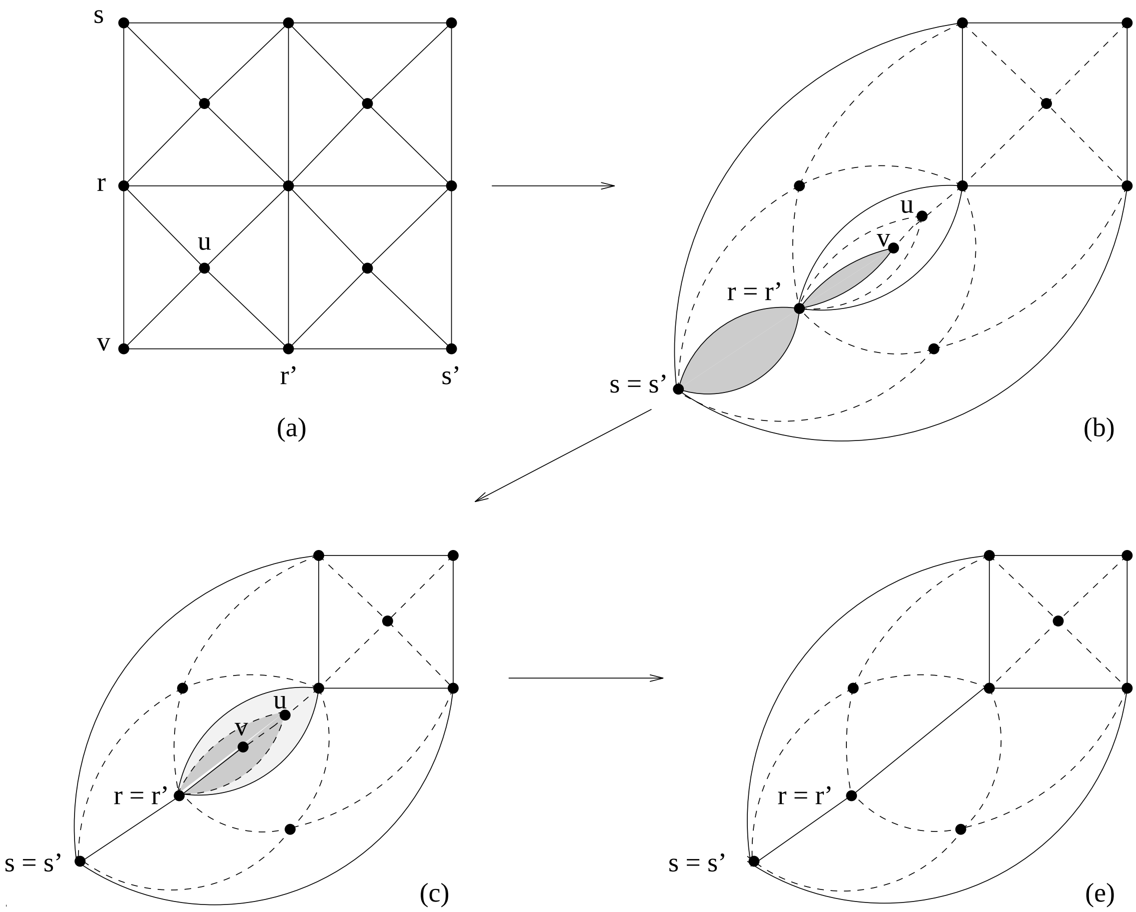


Figura 3.11: Colagem de um ladrilho a si mesmo, e eliminação de triângulos degenerados

3.3 Visualização gráfica da superfície

Para produzir a imagem final da triangulação $\Delta_k(C)$ de um complexo celular C, nós usamos técnicas tradicionais de computação gráfica. Cada face da $\Delta_k(C)$ é desenhada como um triângulo plano. As arestas da triangulação que correspondem a pedaços de arestas do mapa original são representadas por cilindros; e os vértices entre essas arestas são representados por esferas de mesmo raio. As esferas internas, que representam os vértices internos ao ladrilho, têm a finalidade de proporcionar suavidade às articulações dos cilindros que representam as arestas. Além disso, os vértices extremos da diagonal de cada ladrilho, que representam vértices do complexo celular dado, são representados por uma esfera de raio maior que as demais.

Para diferenciarmos quais arestas da triangulação pertencem as arestas primais e duais do complexo original, dividimos cada ladrilho em 4 quadrantes (*quartilhos*) e rotulamos os triângulos com números 0–3, como indicado na figura 3.12. A representação da aresta primal, por definição, é o conjunto de arestas que separam 0–1 e 2–3; da mesma forma, a aresta dual pode ser representada pelo conjunto de arestas que separam 0–3 e 2–1.

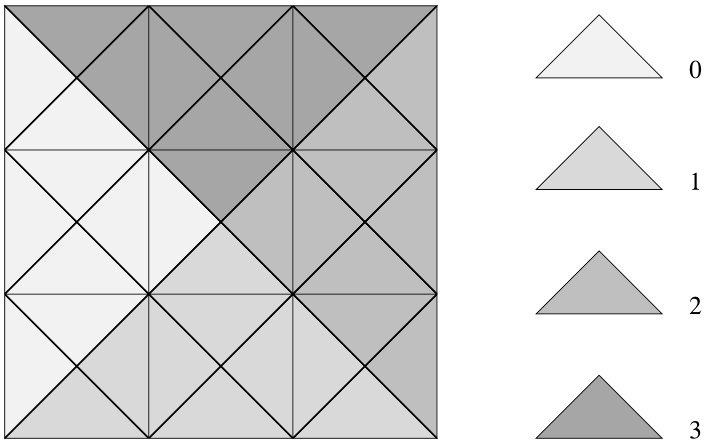


Figura 3.12: Um ladrilho de dimensão 3×3 representando os quatro quadrantes coloridos.

Esta abordagem funciona, porque os triângulos que sobreviveram ao processo de eliminação de triângulos degenerados (seção 3.2.5 são suficientes para visualizar as arestas e faces do complexo original. Especificamente, segue do teorema 3.1 a seguinte proposição:

Proposição 3.1 *Para qualquer ladrilhamento de ordem $k \geq 3$, depois do processo de limpeza, ainda temos para cada ladrilho:*

- *Pelo menos quatro triângulos, um em cada quartilho;*
- *Pelo menos uma aresta de $\Delta_k(C)$, na fronteira entre cada dois quartilhos adjacentes.*

Capítulo 4

A função energia

Como já vimos, nosso objetivo é resolver o seguinte problema : dada uma triangulação topológica T, determinar uma configuração geométrica para a mesma — isto é, as coordenadas de seus vértices $\mathbb{V}T$ — de maneira que a superfície obtida esteja próxima do que definimos como "representação bonita".

Nossa intenção neste capítulo é definir funções numéricas da geometria da superfície, *energias*, que medem as características indesejáveis da mesma, ou seja, o quanto a superfície está distante de uma representação que definimos como "bonita".

4.1 Configurações

Uma vez que cada ladrilho é modelado por uma rede de triângulos *planos*, a forma da superfície, e portanto sua energia, depende apenas das coordenadas dos n vértices dos triângulos.

Definimos, portanto, uma *configuração* da superfície como sendo uma tupla de n pontos $x_1, x_2, \ldots, x_n$ do $\mathbb{R}^3$. Uma *função energia* é então uma função real de $3n$ argumentos reais.

4.1.1 Vértices fixos e variáveis

Para maior versatilidade, nossa implementação permite especificar individualmente se cada vértice é *fixo* ou *variável*. Um vértice variável possui a propriedade de poder alterar suas coordenadas de uma configuração geométrica para a outra. A capacidade de manter certos vértices fixos permite usar nossa ferramenta para projeto de juntas suaves entre superfícies dadas (filetes, chanfros,etc.).

4.2 Funções energia utilizadas

Dentre as funções energia que estudamos, as que produziram melhores resultados (isoladamente ou em combinação) foram as seguintes:

- *Energia de dobramento* $\mathcal{E}^{dobr}$: Mede os ângulos diedrais entre faces adjacentes. Minimizar $\mathcal{E}^{dobr}$ tende a tornar a superfície mais plana, e distribuir uniformemente a curvatura entre todas as arestas.

- *Energia de excentricidade* $\mathcal{E}^{excen}$: Mede a assimetria dos comprimentos e direções das arestas incidentes em cada vértice. Minimizar $\mathcal{E}^{excen}$ tende a tornar a superfície mais plana e os triângulos vizinhos a cada vértice mais uniformes.

- *Energia de ângulos* $\mathcal{E}^{ang}$: mede irregularidades nas direções das arestas incidentes em cada vértice. Minimizar $\mathcal{E}^{ang}$ tende a evitar pontos de trançamento (definidos mais adiante), desdobrar a superfície, e igualar os ângulos entre as arestas adjacentes.

- *Energia de proximidade* $\mathcal{E}^{prox}$: mede a proximidade entre pares de triângulos. Minimizar $\mathcal{E}^{prox}$ tende a evitar a superposição de partes não-adjacentes da superfície, e fazer com que as intersecções da superfície com si mesma sejam transversais em vez de rasantes.

- *Energia de espalhamento* $\mathcal{E}^{espal}$: mede a distância entre os vértices e a origem do $\mathbb{R}^3$. Minimizar $\mathcal{E}^{espal}$ tende a limitar a extensão da superfície, desfavorecendo soluções com vértices excessivamente espalhados.

- *Energia de áreas de ladrilhos* $\mathcal{E}^{area}$: mede a diferença entre a área de um quartilho na superfície e sua área nominal (uma constante). Minimizar esta energia tende a deixar as arestas do complexo original C bem espaçadas, e manter a área total da superfície constante.

4.2.1 Configurações degeneradas

De modo geral, cada uma das energias acima atinge seu mínimo em configurações degeneradas pouco interessantes. Por exemplo, uma configuração que minimiza a energia de proximidade $\mathcal{E}^{prox}$ tem todos os vértice infinitamente distantes uns dos outros; a que minimiza a energia de espalhamento $\mathcal{E}^{espal}$ tem todos os vértices na origem. Este problema pode ser enfrentado através de duas técnicas tradicionais de otimização.

Uma maneira é restringir a minimização a um subconjunto adequado de configurações. Por exemplo, poderíamos considerar apenas configurações *normalizadas* $x_1, \ldots, x_n$ tais que

$$\sum_{i=1}^{n} |x_i|^2 = 1$$

Soluções degeneradas também podem se evitadas fixando-se um número suficiente de vértices da triangulação.

Funções de penalidade

Outra maneira clássica de evitar soluções degeneradas é usar uma *função de penalidade*, que assume valores elevados nas configurações degeneradas. No nosso caso, podemos usar as próprias *energias* como função de penalidade. Cada uma das energias só penaliza um tipo de defeito da superfície, mas algumas destas energias têm efeitos opostos entre si: não existe nenhuma configuração que seja ótima para todas as energias ao mesmo tempo.

Portanto, é possível obter configurações interessantes minimizando uma *energia mista*, que consiste numa combinação linear de duas ou mais energias. Assim, por exemplo, uma configuração que minimiza $\alpha\mathcal{E}^{prox} + \beta\mathcal{E}^{espal}$, com $\alpha, \beta > 0$, tem seus vértices a distâncias finitas e relativamente uniformes. Este assunto será examinado mais a fundo na seção 4.11.

No restante deste capítulo, examinaremos mais detalhadamente as principais funções de energia que testamos e seus efeitos sobre a superfície final. Também abordaremos, sem muitos detalhes, outros tipos de energias que estudamos mas não chegamos a usar.

4.3 Energia de dobramento

Definimos a energia de dobramento de uma aresta e como sendo o quadrado do ângulo diedral externo θ_e entre as faces adjacentes a essa aresta, multiplicado pelo comprimento $|e|$ da aresta. A energia de dobramento total $\mathcal{E}^{dobr}$ da configuração é a soma destas energias parciais, para todas as arestas da triangulação:

$$\mathcal{E}^{dobr} = \sum_{e \in \mathbb{E}T} |e|\, \theta_e^2 \tag{4.1}$$

Intuitivamente, podemos supor que cada aresta e é uma mola bidimensional cuja posição de repouso acontece quando as faces adjacentes a e formam um único plano. O gradiente desta energia é a resultante das forças exercidas por essas molas sobre as faces adjacentes a cada uma das arestas da triangulação.

O ângulo diedral θ_e da aresta pode ser calculado pela fórmula

$$\cos\theta_e = \frac{(\vec{a}\cdot\vec{b})(\vec{a}\cdot\vec{c}) - (\vec{a}\cdot\vec{a})(\vec{b}\cdot\vec{c})}{\sqrt{(\vec{a}\cdot\vec{a})(\vec{b}\cdot\vec{b}) - (\vec{a}\cdot\vec{b})^2}\,\sqrt{(\vec{a}\cdot\vec{a})(\vec{c}\cdot\vec{c}) - (\vec{a}\cdot\vec{c})^2}}$$

onde $\vec{a}$ é o vetor definido pela aresta e (isto é, $Dest(e) - Org(e)$), e $\vec{b}$ e $\vec{c}$ são os vetores definidos pelas arestas adjacentes $ONext(e)$ e $OPrev(e)$ [8].

4.3.1 Controle Local

Nossa implementação permite especificar separadamente para cada aresta de T, se a mesma se comporta como uma mola (como descrito em 4.3) ou se ela se comporta como uma dobradiça sem mola.

Assim, podemos usar nossa ferramenta para problemas em que a borda é articulada, para o caso em que as arestas possuem atributo de mola, ou engastada, caso contrário.

4.3.2 Simplificações

O custo elevado da função arccos, necessária para obter o ângulo θ_e a partir do valor de $\cos\theta$, nos levou a substituir a mesma pela aproximação

$$\theta^2 = (2\frac{\theta}{2})^2 \approx (2\sin\frac{\theta}{2})^2 = 2(1-\cos\theta)$$

Esta aproximação dá resultados satisfatórios mesmo para ângulos grandes, pois $1-\cos\theta$ é uma função crescente de θ^2, para todo θ entre $-\pi$ e $+\pi$. Veja o gráfico na figura 4.1.

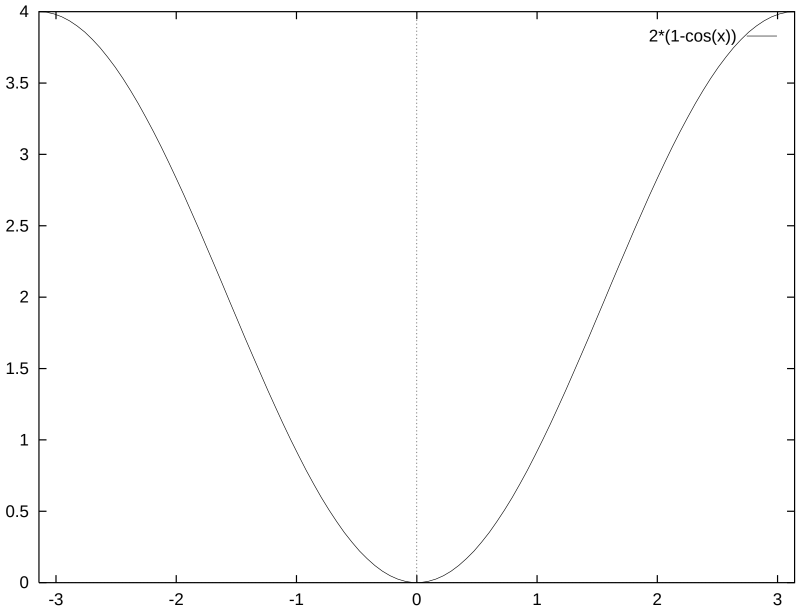


Figura 4.1: Gráfico de $2(1-\cos\theta)$ em função de θ.

Observe que a energia de dobramento, apesar de definida em termos puramente locais, tem um efeito global: a curvatura total da superfície acaba sendo distribuída sobre todas as arestas, de maneira mais ou menos uniforme. Na minimização numérica, observa-se um processo de difusão, em que a curvatura "flui" de regiões de relevo acidentado — picos, depressões, e dobras — para regiões mais planas. Este comportamento é uma característica comum a várias das funções energia que veremos a seguir.

4.4 Energia de excentricidade

A *energia de excentricidade* $\mathcal{E}^{excen}$ mede o quanto cada vértice v esta longe do baricentro do conjunto N_v de seus vizinhos (considerados como tendo a mesma massa). Especificamente,

$$\mathcal{E}^{excen} = \sum_{v \in \mathbb{V}T} |v - b_v|^2 \tag{4.2}$$

onde

$$b_v = \frac{1}{|N_v|} \sum_{u \in N_v} u$$

O gradiente desta energia é uma força que atrai cada vértice para o baricentro de seus vizinhos, com magnitude proporcional à distância entre esses dois pontos.

Intuitivamente, minimizar esta energia tende a aplainar a superfície, e uniformizar os comprimentos das arestas que saem de cada vértice. A superfície se comporta como uma rede cujas arestas são fios elásticos (com comprimento de repouso nulo).

Esta energia não pode ser usada sozinha em triangulações sem vértices fixos, pois seu único mínimo é uma configuração degenerada que tem todos os vértices no mesmo ponto.

4.5 Energia de ângulos

A energia de ângulos $\mathcal{E}^{ang}$ é calculada comparando-se os ângulos entre as k arestas que incidem num determinado vértice com o ângulo ideal $2\pi/k$.

Mais precisamente, sejam $v_0, \ldots, v_{k-1}$ os vértices vizinhos a um vértice u dado. A *direção normal* à superfície no vértice u é, por definição, a do vetor

$$\sum_{i=0}^{k-1} (v_i - u) \times (v_{i+1} - u)$$

que é equivalente a

$$\sum_{i=0}^{k-1} v_i \times v_{i+1}$$

Obtida a normal n, calculamos as projeções d_i dos vetores $v_i - u$ sobre um plano perpendicular a n, e definimos θ_i como sendo o ângulo entre os vetores d_i e d_{i+1}. A energia de ângulos do vértice u é então

$$\mathcal{E}^{ang}(u) = \sum_{i=0}^{k-1}(\theta_i - \frac{2\pi}{k})^2 \tag{4.3}$$

A energia de ângulos da configuração é a soma de $\mathcal{E}^{ang}(u)$ sobre todos os vértices u.

Intuitivamente, podemos supor que a normal de cada vértice u é o eixo de uma dobradiça com k "asas" planas separadas por molas que tendem a igualar os ângulos entre as asas. Cada vizinho de u está restrito à asa correspondente, mas pode se deslocar livremente sobre a mesma. O gradiente da energia é a resultante das forças exercidas por essas molas sobre os vértices da triangulação.

A motivação para projetar as arestas sobre o plano tangente, em vez de medir os ângulos sobre a própria superfície, é penalizar *vértices de trançamento*, que são vértices v cujos vizinhos, projetados num plano tangente, perfazem mais de uma volta em torno de v, ou nenhuma volta. Tais vértices ocorrem quando a superfície contem auto-intersecções; por exemplo, as representações da garrafa de Klein e do toro ilustradas na figura 4.2.

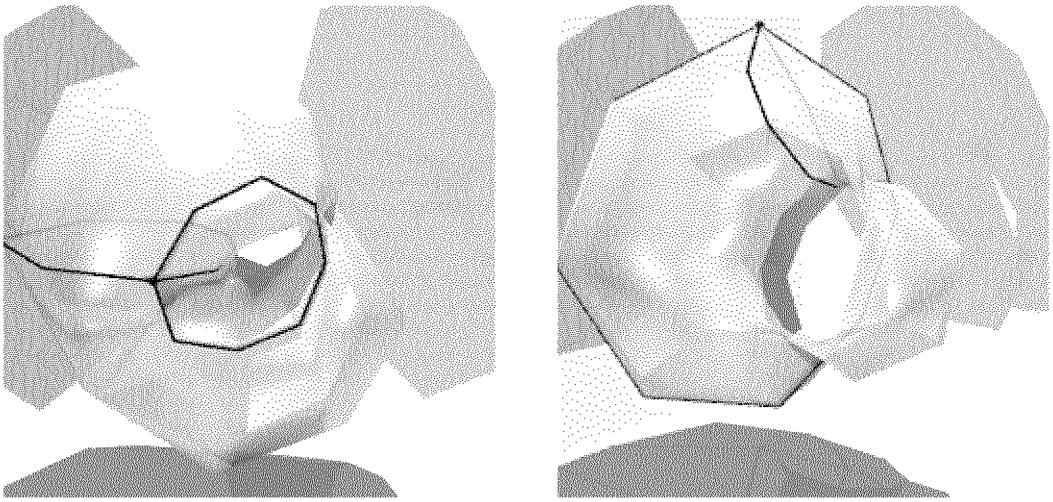


Figura 4.2: Realizações com trançamentos

4.6 Energia de proximidade

A energia de proximidade $\mathcal{E}^{prox}$ é calculada como se a superfície estivesse coberta de cargas elétricas que se repelem entre si. O valor de $\mathcal{E}^{prox}$ é a energia potencial desse sistema de cargas elétricas.

O gradiente desta energia é a resultante das forças de repulsão entre todas essas cargas. Minimizar $\mathcal{E}^{prox}$ significa afastar os triângulos uns dos outros, o que tende a evitar a superposição de partes distintas da superfície, e fazer com que as intersecções da superfície com si mesma sejam transversais em vez de rasantes.

O ideal seria calcular $\mathcal{E}^{prox}$ supondo que a carga está uniformemente espalhada sobre a superfície. Infelizmente, isso tornaria o cálculo da energia muito complicado e dispendioso. Em vista disso, decidimos aproximar a distribuição uniforme por uma coleção de cargas discretas, posicionadas no baricentro de cada triângulo. Assim, a energia elétrica pode ser calculada pela fórmula

$$\mathcal{E}^{prox} = \sum_{1 \leq i < j \leq n} \mathcal{E}_{ij}^{prox} \tag{4.4}$$

onde $\mathcal{E}_{ij}^{prox}$ é a energia potencial entre as duas cargas associadas aos triângulos i e j. A aproximação mais simples para este termo corresponderia a supor duas cargas pontuais unitárias situadas nos baricentros desses triângulos; ou seja

$$\mathcal{E}_{ij}^{prox} = \frac{1}{|b_i - b_j|}$$

onde b_k é o baricentro do triângulo k. Entretanto, esta definição introduz um grande número de mínimos locais, em que a superfície está arbitrariamente trançada, mas onde as cargas estão intercaladas — isto é, cada superfície passa pelos buracos entre as cargas da outra (veja fig. 4.3(a)). Para destrançar essas configurações seria necessário aproximar temporariamente algumas cargas, aumentando a energia elétrica.

A fim de contornar este problema, sem aumentar muito o custo, decidimos calcular esta energia supondo que a carga elétrica de cada triângulo está espalhada no espaço, de tal forma que a energia do par é dada pela fórmula

$$\mathcal{E}_{ij}^{prox} = \frac{1}{\sqrt{|b_i - b_j|^2 + \mu_i^2 + \mu_j^2}}$$

onde o parâmetro μ_k mede o raio aproximado da nuvem de carga do triângulo k.

Os parâmetros μ_k devem ser grandes o bastante para que a união das nuvens de carga elétrica cubra toda a superfície sem deixar buracos. Nosso programa define μ_k a partir dos vértices u, v, w do triângulo k, segundo a fórmula

$$\begin{aligned}
\mu_k^2 &= \frac{\alpha^2}{3}(|u - b_k|^2 + |v - b_k|^2 + |w - b_k|^2) \\
&= \frac{\alpha^2}{9}(|u - v|^2 + |v - w|^2 + |w - u|^2)
\end{aligned}$$

onde α é um parâmetro ajustável pelo usuário. (Nos nossos testes, adotamos $\alpha \approx 0.5$.)

A motivação para cargas "nebulosas" pode ser compreendida considerando-se o exemplo da figura 4.3. Na configuração esquematizada na figura 4.3(a) com duas superfícies trançadas, a energia de proximidade é localmente mínima. Neste caso, usando cargas pontuais há poucas esperanças que um método de otimização consiga destrançar às superfícies; para isso seria necessário transpor uma barreira de alta energia. A figura 4.3(b) ilustra duas cargas nebulosas nos vértices u e v. Note que dentro destas áreas, a repulsão entre as duas superfícies não cresce consideravelmente a medida que os outros vértices se aproximam de u ou v, isto é, mesmo que um vértice se sobreponha à outro, a energia de proximidade não será infinita. Isto permite que uma superfície atravesse a outra.

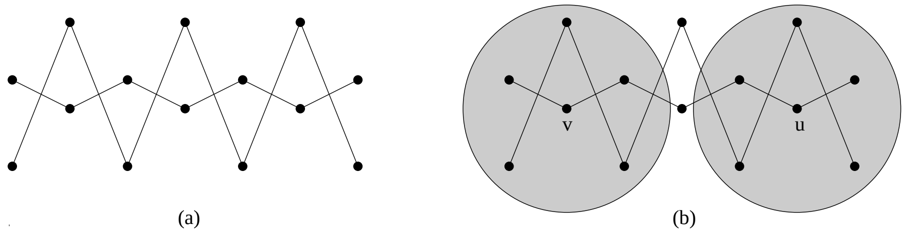
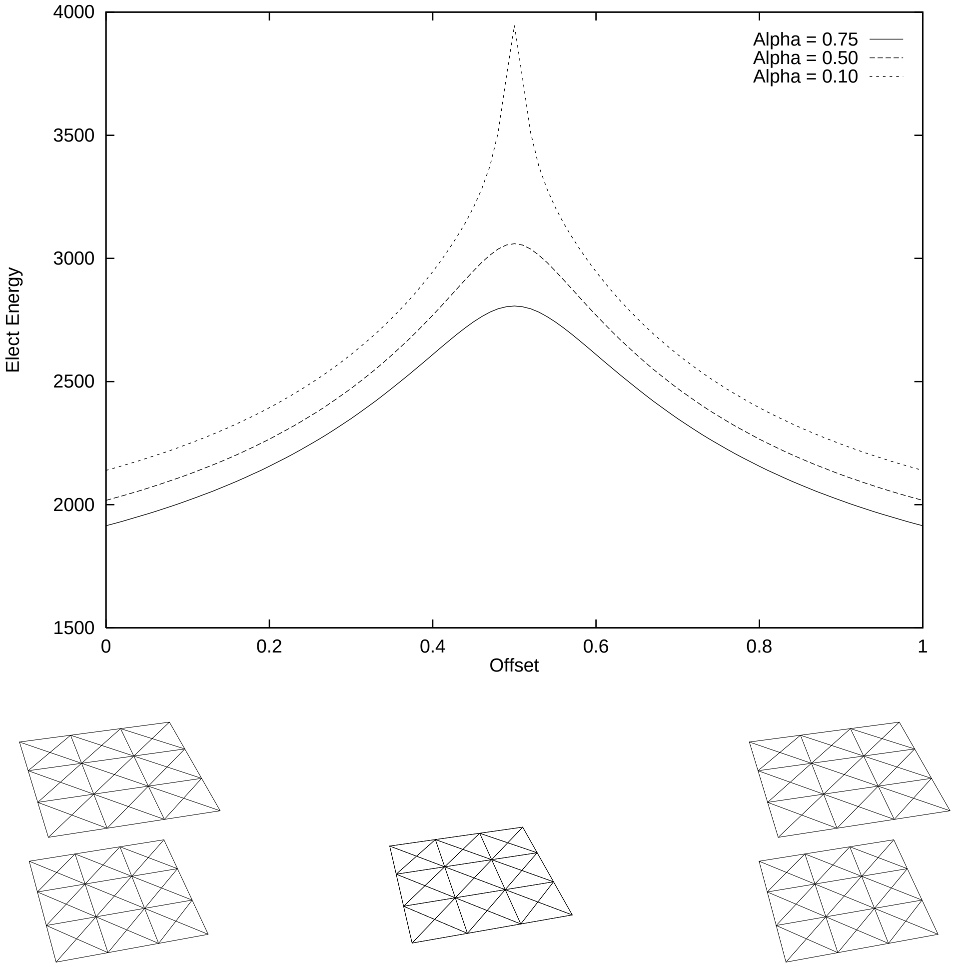


Figura 4.3: Energia de proximidade (a) usando cargas pontuais e (b) usando cargas nebulosas.

A desvantagem deste modelo de cargas "nebulosas" tridimensionais é que a repulsão entre dois triângulos próximos entre si é mais fraca do que a esperada para cargas distribuídas na superfície dos mesmos, e varia mais lentamente em função da distância entre eles. Em conseqüência, a minimização da energia elétrica fica menos eficiente na eliminação de intersecções e sobreposições da superfície.

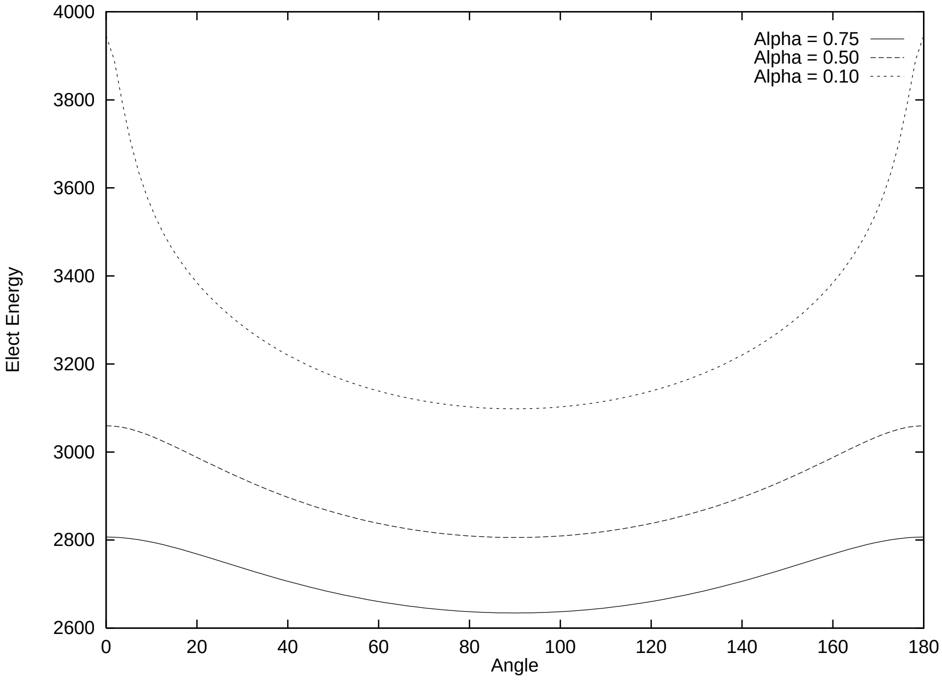
As figuras 4.4, 4.5, e 4.6 representam o comportamento da energia elétrica de dois ladrilhos, respectivamente: paralelos em função da distância, cruzados em função do ângulo, cruzados em função da posição do cruzamento. Em cada uma das figuras, o gráfico mostra a variação de $\mathcal{E}^{prox}$ para 3 diferentes valores de α: 0.10, 0.50 e 0.75. Neste último, note que para cargas mais concentradas ($\alpha = 0.10$), a função energia tem vários mínimos locais, separados por barreiras altas de energia.

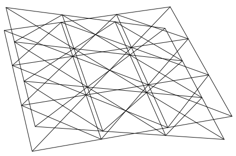
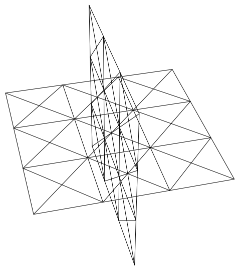
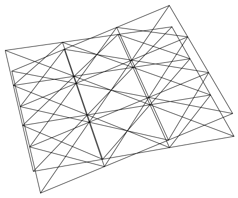
Uma alternativa para este modelo seria considerar cargas discretas (nebulosas) centradas nos vértices, em vez dos baricentros dos triângulos. Entretanto, a equação de Euler implica que o número n de triângulos é aproximadamente o dobro do número m de vértices. Portanto, ao usar os triângulos estamos distribuindo a carga mais uniformemente sobre a superfície.



Deslocamento 0.0 Deslocamento 0.5 Deslocamento 1.0

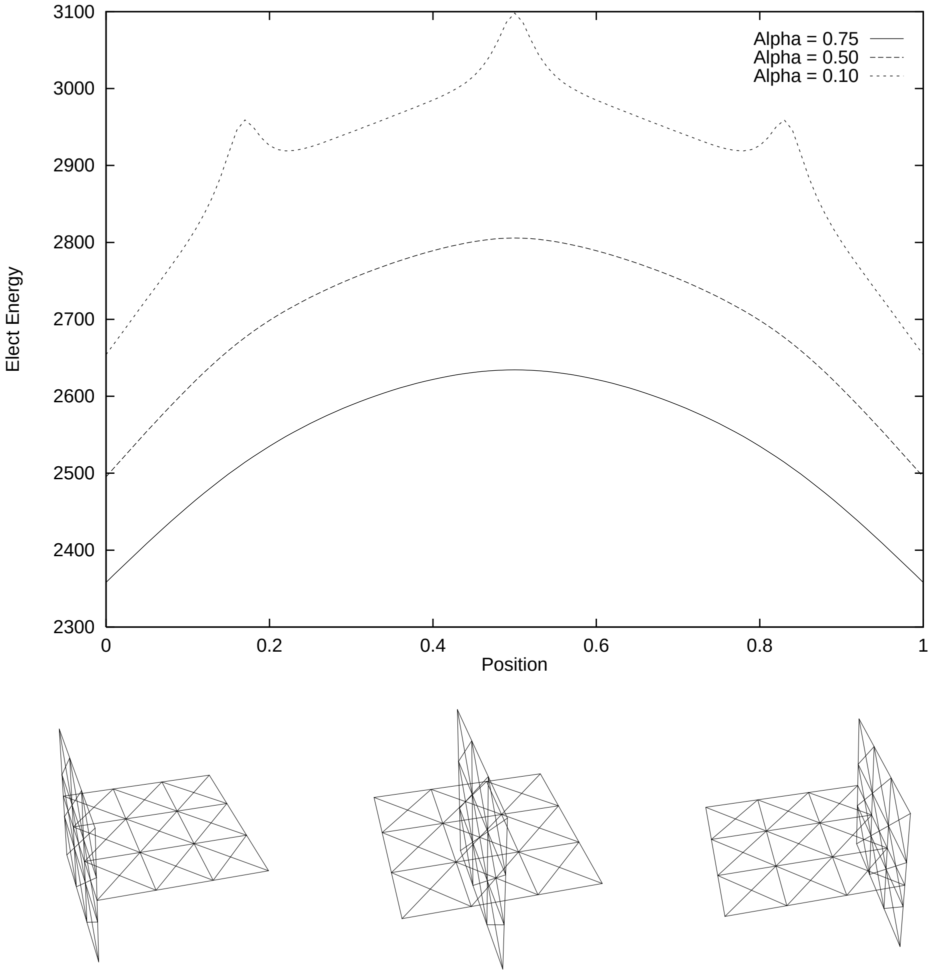
Figura 4.4: Gráficos de $\mathcal{E}^{prox}$ para dois ladrilhos paralelos, em função de seu deslocamento vertical (*Offset*), para diferentes valores do parâmetro α



Ângulo 10^o Ângulo 90^o Ângulo 170^o

Figura 4.5: Gráficos de $\mathcal{E}^{prox}$ para dois ladrilhos cruzados, em função do ângulo de cruzamento, para diferentes valores de α



Posição 0.0 Posição 0.5 Posição 1.0

Figura 4.6: Gráficos de $\mathcal{E}^{prox}$ para dois ladrilhos cruzados perpendicularmente, em função da posição do cruzamento, para diferentes valores de α.

4.7 Energia de espalhamento

A energia de espalhamento $\mathcal{E}^{espal}$ é por definição a soma dos quadrados das distâncias dos vértices à origem:

$$\mathcal{E}^{espal} = \sum_{i=1}^{m} |v_i|^2 \tag{4.5}$$

O gradiente desta energia é uma força que atrai cada vértice para a origem, proporcionalmente à distância.

A energia de espalhamento não pode ser usada sozinha, pois seu único mínimo é a configuração degenerada onde todo vértice está na origem. Como já vimos, ela é útil em combinação com outras energias, especialmente $\mathcal{E}^{prox}$. Nessas combinações, ela age como uma função de penalidade, evitando que os vértices se afastem infinitamente uns dos outros.

4.8 Energia de área de ladrilhos

Para definir a energia de área de ladrilhos $\mathcal{E}^{area}$, dividimos cada ladrilho do mapa original em quatro quartos de ladrilho — *quartilhos* — pelas arestas primal e dual correspondentes, como mostrado na figura 3.12. Cada quartilho corresponde a uma parte bem definida da superfície, que consiste de todos os triângulos provenientes daquela parte do ladrilho que sobreviveram à operação de limpeza (veja seção 3.2).

A energia $\mathcal{E}^{area}$ é então definida comparando-se a área a_i de cada quartilho na configuração corrente com sua área nominal A (uma constante), segundo a fórmula

$$\mathcal{E}_i^{area} = \frac{a_i}{A} + \frac{A}{a_i} - 2 \tag{4.6}$$

e somando-se esse termo para todos os quartilhos.

A figura 4.7 mostra o comportamento desta fórmula em função da área a_i. Note que $\mathcal{E}^{area}$ torna-se infinita sempre que qualquer das áreas a_i tende a zero. Portanto, ao incluir $\mathcal{E}^{area}$ na fórmula da energia, com qualquer peso positivo, estamos garantindo que todo quartilho tem área não-nula.

O gradiente desta energia é uma espécie de “pressão bidimensional” que tende a manter a área de cada quartilho igual a A. O comportamento intuitivo de cada quartilho não tem semelhança com materiais físicos comuns: ele resiste a mudança de área, como um tecido, mas é totalmente indiferente a mudanças de forma.

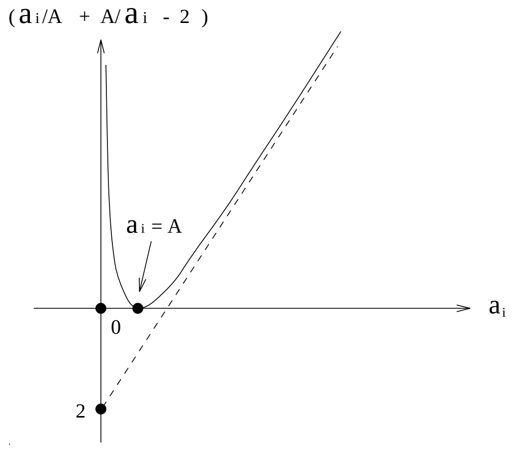


Figura 4.7: Gráfico da energia de área de um quartilho $\mathcal{E}_i^{area}$, em função de sua área a_i.

4.9 Outras energias

Outras energias que consideramos, mas que não deram resultados satisfatórios, são:

- *Energia de compressão de triângulos* $\mathcal{E}^{compr}$, que mede a discrepância entre a área corrente a_i de cada triângulo de T e sua *área nominal* A_i (uma constante):

$$\mathcal{E}^{compr} = \sum_{i=1}^{n} \left(\log \frac{a_i}{A_i} \right)^2$$

  Minimizar esta energia significa forçar que as áreas dos triângulos se aproximem o máximo das respectivas áreas nominais.

- *Energia de desigualdade de áreas de triângulos* $\mathcal{E}^{relar}$, que mede a discrepância entre as áreas de dois triângulos ($Left(e)$, $Right(e)$), adjacentes a cada aresta e:

$$\mathcal{E}^{relar} = \sum_{e \in \mathbb{E}T} \left(\log \frac{a_{Left(e)}}{a_{Right(e)}} \right)^2$$

  Minimizar $\mathcal{E}^{relar}$ significa tornar as áreas de todos os triângulos próximas entre si.

- *Energia de deformação de triângulos* $\mathcal{E}^{deform}$, mede a discrepância entre os ângulos dos triângulos e seus ângulos nominais. Esta energia é semelhante à energia de ângulos $\mathcal{E}^{ang}$, exceto que os ângulos são medidos entre as arestas, em vez de entre suas projeções no plano tangente.

- *Energia elástica de arestas* $\mathcal{E}^{elast}$, calculada supondo-se que cada aresta é um fio elástico, que puxa ou empurra os vértices extremos com tensão que depende da discrepância entre o comprimento corrente da aresta e seu *comprimento nominal* L_e:

$$\mathcal{E}^{elast} = \sum_{e \in \mathbb{E}T} \left(\log \frac{|Org(e) - Dest(e)|}{L_e} \right)^2$$

Estas energias tem um problema comum: elas obrigam os ladrilhos a se comportarem como retalhos de pano, que podem ser dobrados mas resistem a mudança de forma. Isto faz com que as juntas entre os ladrilhos virem quinas e saliências da superfície.

Além disso, as energias $\mathcal{E}^{compr}$ e $\mathcal{E}^{relar}$ permitem triângulos finos e compridos, desde que tenham a área especificada. Controlar apenas as áreas dos triângulos não garante que as faces do complexo tenham formas homogêneas. Para que isso seja verdade, estas energias devem ser usadas em conjunto com uma ou mais energias que penalizem a deformação dos triângulos. A energia de excentricidade seria uma boa opção de restrição para esse caso.

4.10 Análise das energias em função da triangulação

Para usar estas funções energia eficazmente, é desejável que elas não dependam do número de triângulos usados em cada ladrilho, mas apenas da topologia e geometria da superfície.

Para formalizar esta idéia vamos introduzir o conceito de *duplicação* de uma triangulação T. Trata-se de outra triangulação T' que é obtida de T dividindo-se cada aresta e em duas metades, ligadas por um novo *vértice médio* $v_m(e)$; e dividindo-se cada triângulo em 4 novos triângulos por meio de novas arestas que ligam os vértices médios consecutivos na sua fronteira. Veja a figura 4.8.

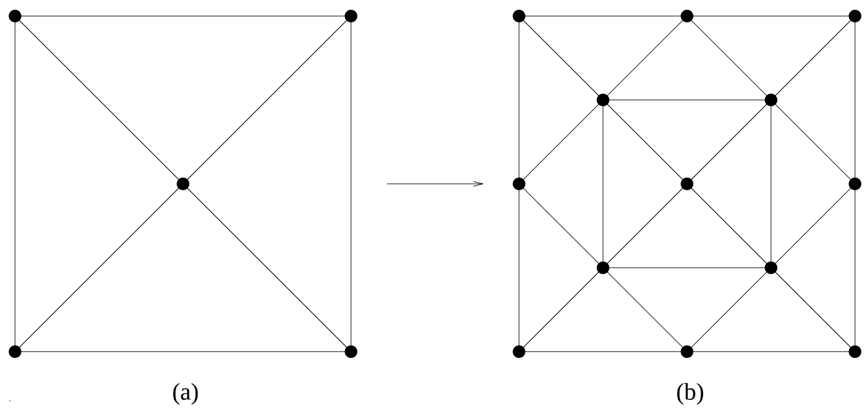


Figura 4.8: Um ladrilho triangulado de ordem 1×1 (a), e sua duplicação (b).

Note que se T foi obtida por ladrilhamento de um complexo celular com ladrilhos de ordem $k \times k$, como descrito no capitulo 2, a duplicação de T praticamente equivale a ladrilhar o mesmo complexo com ladrilhos de ordem $2k \times 2k$. Note porém que a disposição dos triângulos no ladrilho é ligeiramente diferente.

Suponha que $x = (x_1, \ldots, x_n)$ é uma configuração de T que corresponde a um mínimo local de uma certa combinação de energias. Considere a seguinte operação de *refinamento geométrico*: (1) construa a duplicação T' de T, (2) atribua a cada vértice médio $v_m(e)$ de T' a media aritmética das posições dos extremos de e, e (3) partindo desta configuração, encontre uma configuração $x' = (x'_1, \ldots, x'_m)$ de T' que seja mínimo local da mesma função energia.

Para que os resultados da minimização de energia sejam previsíveis e controláveis, é necessário que a forma das superfícies (T, x) e (T', x') sejam semelhantes; e mais ainda, é desejável que as configurações obtidas de (T, X) por refinamentos geométricos repetidos convirjam rapidamente para uma configuração limite bem definida e próxima a (T, x), com uma energia finita e não nula.

Não é nossa intenção determinar rigorosamente as condições matemáticas que garantam esta convergência. Nos limitaremos a ajustar as constantes que aparecem na definição de cada função energia de tal maneira que seus valores não variem substancialmente quando efetuamos um refinamento geométrico de uma configuração otimizada.

Para estimar os valores "típicos" de cada energia, vamos supor uma *configuração de referência* hipotética (T, x) cujos vértices estão próximos a uma esfera de raio r, e cujos triângulos são aproximadamente eqüiláteros e de tamanho uniforme; de modo que todos os ângulos, comprimentos, áreas, etc. serão aproximadamente iguais. Vamos denotar por N_t, N_e, N_v, e N_q o número de triângulos, arestas, vértices, e quartilhos de T, respectivamente.

Obviamente, este modelo é uma aproximação muito grosseira das configurações ótimas reais, mesmo para complexos com poucas arestas e topologia esférica. Portanto, esta abordagem vai fornecer apenas o comportamento assintótico das energias em função do raio aproximado r e dos números N_t, N_e, N_v, e N_q; que é tudo o que queremos. As constantes de proporcionalidade vão depender da topologia do complexo.

Nas análises a seguir, usaremos o fato que numa triangulação de de uma esfera (ou de qualquer superfície com topologia fixa) temos $N_e = O(N_t)$ e $N_v = O(N_t)$.

4.10.1 Energia de dobramento

A energia de dobramento é a soma de N_e termos, cada um dos quais é o comprimento de uma aresta vezes o quadrado do ângulo diedral externo correspondente.

Supondo-se que que os triângulos sejam aproximadamente iguais, a área média $\bar{t}$ de cada triângulo será aproximadamente $4\pi r^2/N_t = O(r^2 N_t^{-1})$. O comprimento médio das arestas será então $\bar{e} = O(\sqrt{\bar{t}}) = O(r N_t^{-1/2})$.

A distância entre os centros de dois triângulos será proporcional ao comprimento médio das arestas, ou seja $\bar{d} = O(rN_t^{-1/2})$. Vista do centro da esfera, esta distância corresponde a um ângulo de $O(\bar{d}/r)$ radianos. Supondo-se que os triângulos sejam aproximadamente tangentes à esfera, concluímos que este é também o ângulo diedral médio $\bar{theta}$ entre faces adjacentes; ou seja, $\bar{theta} = O(\bar{d}/r) = O(rN_t^{-1/2}/r) = O(N_t^{-1/2})$.

Portanto esperamos que

$$\bar{\mathcal{E}}^{dobr} = O(N_e\bar{e}\bar{\theta}^2) = O(N_t)O(rN_t^{-1/2})O((N_t^{-1/2})^2) = O(rN_t^{-1/2})$$

Assim, para tornar a energia de dobramento da configuração ótima independente de N_t, basta multiplicar a fórmula de $\mathcal{E}^{dobr}$ (4.1) pelo fator $K^{dobr} = N_t^{1/2}$.

4.10.2 Energia excêntrica

Numa triangulação ótima, espera-se que a energia excêntrica $\mathcal{E}^{excen}$ seja devida exclusivamente à curvatura da superfície; isto é, cada vértice v está no ponto da superfície mais próximo do baricentro b_v de seus vizinhos.

Na nossa configuração de referência, já vimos que a distância média entre o vértice v e seus vizinhos é $\bar{e} = O(rN_t^{-1/2})$. Essa distância, vista do centro da esfera, corresponde a um ângulo típico $\bar{\alpha} = o(\bar{e}/r) = O(N_t^{-1/2})$. Portanto a distância típica de b_v à superfície da esfera será

$$\bar{u} \approx r(1 - cos\bar{\alpha}) = O(r\bar{\alpha}^2) = O(rN_t^{-1})$$

e

$$\mathcal{E}^{excen} = O(N_v\bar{u}^2) = O(N_t r^2 N_t^{-2}) = O(r^2 N_t^{-1})$$

Concluímos que, para manter $\mathcal{E}^{excen}$ independente de N_t, basta multiplicar a fórmula (4.2) pelo fator $K^{excen} = N_t$.

4.10.3 Energia de áreas de ladrilhos

A fórmula energia $\mathcal{E}^{area}$ já é praticamente invariante sob refinamento geométrico, pois ela leva em conta as áreas dos quartilhos, que são pouco afetadas por essa operação.

Entretanto, é conveniente acrescentar à formula de $\mathcal{E}^{area}$ (4.6) um fator $1/N_p$, onde N_p é o número de quartilhos. Esta precaução torna $\mathcal{E}^{area}$ independente do número de arestas do complexo original; seu valor passará então a medir consistentemente a variação média de área por quartilho.

Além disso, é aconselhável definir a área ideal A de um quartilho como sendo A_{tot}/N_p, onde A_{tot} é uma constante. Esta escolha de A faz com que a energia $\mathcal{E}^{area}$ seja nula quando a superfície tiver área total A_{tot}, com retalhos de mesmo tamanho.

4.10.4 Energia de espalhamento

Supondo-se que a distância quadrática média de cada vértice à origem seja $O(r^2)$, concluímos imediatamente que

$$\mathcal{E}^{espal} = O(N_v r^2) = O(N_t r^2)$$

Portanto, é conveniente acrescentar à formula desta energia (4.5) o fator de normalização $K^{espal} = N_t^{-1}$.

4.10.5 Energia de proximidade

Na seção 4.6, justificamos a fórmula da energia de proximidade $\mathcal{E}^{prox}$ como sendo uma aproximação da energia potencial de uma distribuição contínua de cargas elétricas sobre a superfície da configuração. Portanto, o refinamento da triangulação deveria apenas tornar essa aproximação mais exata; ou seja, o valor de $\mathcal{E}^{prox}$ deveria ser assintoticamente independente de N_t.

Entretanto, a fórmula (4.4) que usamos para $\mathcal{E}^{prox}$ na seção 4.6 supõe que cada triângulo tem carga total unitária, qualquer que seja seu tamanho. Isso significa que a carga total espalhada sobre a superfície é proporcional a N_t, e portanto a energia potencial cresce assintoticamente como $O(N_t^2)$.

Concluímos que para tornar $\mathcal{E}^{prox}$ invariante sobe refinamento é preciso acrescentar à fórmula (4.4) um fator $K^{prox} = N_t^{-2}$.

4.10.6 Energia de ângulos

A análise teórica do comportamento assintótico desta energia parece ser bastante difícil. Entretanto, testes empíricos nos levaram a concluir que, quando uma configuração ótima é refinada e re-otimizada, o valor desta energia cresce proporcionalmente ao número de triângulos (ou vértices). Isto é, o valor médio de cada parcela (o desvio médio de cada ângulo, relativo ao ângulo ideal) não é alterado pelo refinamento.

Portanto, para manter esta energia invariante sob refinamento, é necessário acrescentar à fórmula (4.3) o fator $K^{ang} = N_t^{-1}$.

4.11 Energias mistas

Embora, como vimos, seja necessário o uso de uma energia mista, não existe uma única combinação fixa de energias que produz bons resultados para qualquer complexo celular.

A escolha da combinação mais adequada parece ser um problema muito difícil. No estado atual dos nossos estudos, os pesos de cada energia só podem ser determinados por tentativa e erro.

Uma solução parcial para este problema seria controlar interativamente os pesos. Pense em um painel de controle, onde você possa controlar a dose com que cada energia irá influenciar no resultado final do processo de otimização. Outra solução mais avançada seria utilizar técnicas de aprendizagem automática para deduzir os pesos ótimos, a partir de uma coleção de exemplos de realizações "bonitas" fornecidas pelo usuário [28]. Deixaremos esses melhoramentos para trabalhos futuros.

4.11.1 Peso crítico: $\mathcal{E}^{prox} \times \mathcal{E}^{dobr}$

Para ilustrar a dificuldade da escolha adequada dos pesos, vamos considerar o caso de uma mistura simples das energias de dobramento $\mathcal{E}^{dobr}$ e proximidade $\mathcal{E}^{prox}$.

Sozinha, a energia de proximidade não leva em conta a suavidade da superfície. Portanto, a energia de proximidade deve ser usada sempre em combinação com alguma outra energia que favoreça superfícies suaves, como a energia de curvatura ou a excêntrica. Mesmo assim, dependendo dos pesos com que as energias são combinadas, pode ocorrer o que denominamos de "efeito abacaxi" — vértices deslocados alternadamente para dentro e para fora da superfície "média" suave (veja a figura 4.9).

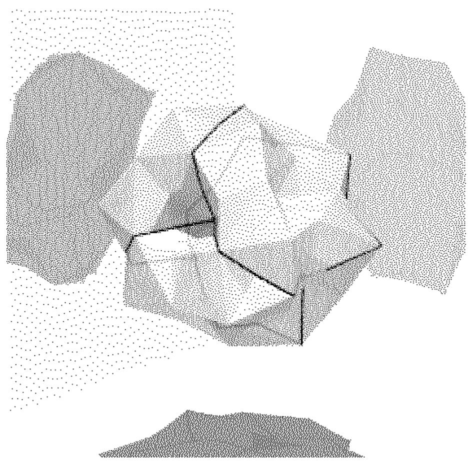


Figura 4.9: O "efeito abacaxi".

Para entender a origem deste problema, considere uma triangulação plana, com apenas um vértice variável (veja figura 4.10).

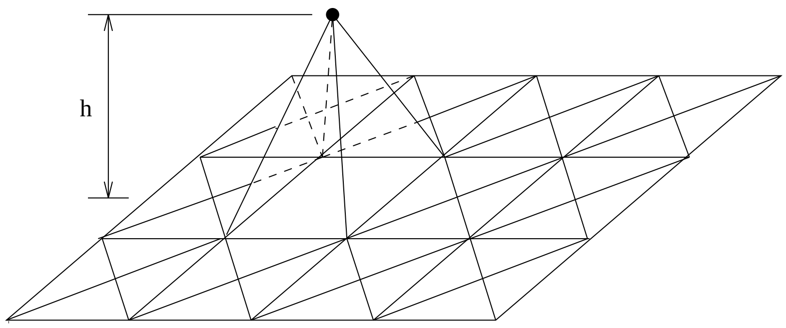


Figura 4.10: A triangulação usada na análise do "efeito abacaxi".

Os gráficos na figura 4.11 mostram respectivamente as energias $\mathcal{E}^{prox}$, $\mathcal{E}^{dobr}$, bem como sua combinação $\mathcal{E}^{total}$, em função do parâmetro h da figura 4.10.

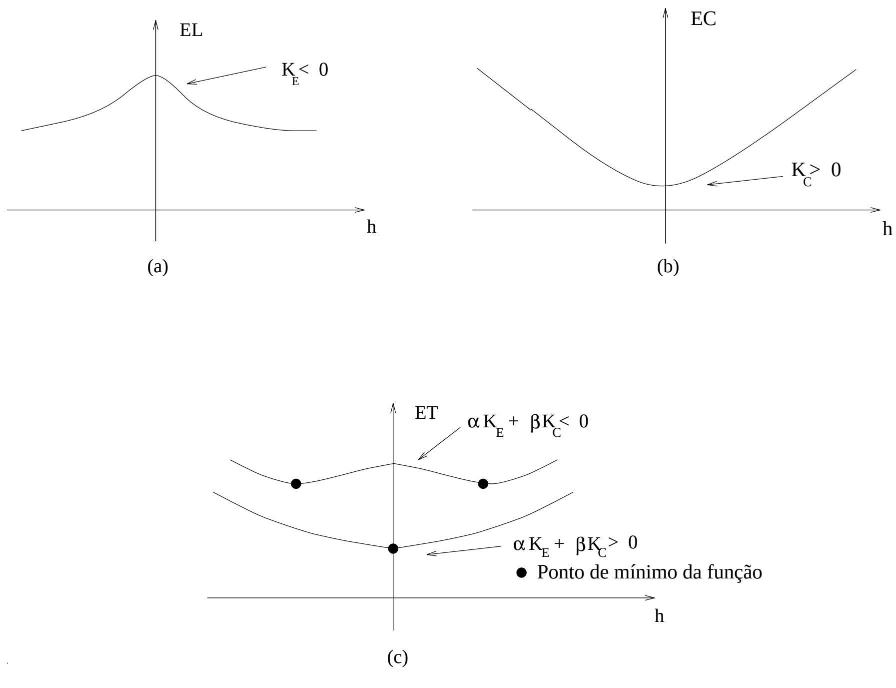


Figura 4.11: Representação das energias de proximidade (a), de dobramento (b), e total (c), em função do parâmetro h.

Note que em (a) a segunda derivada $K_E < 0$ e em (b) a segunda derivada $K_C > 0$.

Portanto, para uma energia $\mathcal{E}^{total} = \alpha\mathcal{E}^{prox} + \beta\mathcal{E}^{dobr}$ tem-se que: se $\alpha K_E + \beta K_C < 0$, há dois mínimos com $h \neq 0$; se $\alpha K_E + \beta K_C > 0$, há um único mínimo em $h = 0$.

Se o mínimo ocorrer no ponto $h \neq 0$, a minimização de $\mathcal{E}^{total}$ sobre a superfície toda resultará no que pode ser chamado de fenômeno "abacaxi". Caso contrário, se o mínimo ocorre em $h = 0$ tem-se uma superfície suave, sem ondulações.

Capítulo 5

Otimização

5.1 Introdução

Neste capítulo descreveremos as várias técnicas de minimização de energia que experimentamos no decorrer deste trabalho.

Vamos classificar os algoritmos de otimização usados por nós em dois grupos: (1) os *genéricos e exatos* e (2) os *específicos e heurísticos*.

Os algoritmos do primeiro grupo são *genéricos* porque encaram a minimização da função energia como um problema genérico de otimização não-linear: dada uma função f de $\mathbb{R}^n$ em $\mathbb{R}$, encontrar um vetor $x = (x_1, ..x_n)$ que minimiza o valor de $f(x)$. Isto é, se aplicam a qualquer função, não dependendo da natureza da mesma. Além disso, são *exatos* pois, se fossem rodados por tempo suficiente e em um ambiente "ideal", onde não houvessem problemas de precisão, convergiriam para um mínimo local.

Os algoritmos do segundo grupo são *específicos* porque são especializados para as nossas funções energias; e são *heurísticos* pois nem sempre convergem e, se convergem, não convergem necessariamente para um mínimo local. Estes algoritmos aplicam repetidamente certas operações de ajuste local que afetam apenas um vértice e seus vizinhos. Cada um destes ajustes procura minimizar apenas um tipo de energia sem se preocupar com os outros tipos; e, além disso, é geralmente baseado num modelo qualitativo da energia. Portanto, o ajuste pode aumentar a energia em vez de diminuí-la.

A experiência mostra que os métodos heurísticos funcionam muito bem no início da otimização, quando a configuração ainda está muito distante do mínimo desejado. Por isso, a melhor estratégia é usar algumas iterações das heurísticas para gerar a forma inicial aproximada e, a partir desta forma inicial, continuar otimizando usando algoritmos exatos até atingir um ótimo local. Na verdade, a configuração obtida aplicando apenas os métodos heurísticos em geral já é visualmente muito boa.

5.1.1 Mínimo global vs. local

Um *mínimo local* de uma função f é um ponto x^* de $\mathbb{R}^n$ tal que $f(x) \geq f(x^*)$ para todo x em uma vizinhança de x^*. O ponto x^* é um *mínimo global* se $f(x) \geq f(x^*)$ para todo x em $\mathbb{R}^n$.

Identificar um mínimo local é um problema clássico de análise numérica, para o qual existem muitos algoritmos confiáveis e uma vasta teoria matemática. Infelizmente, as funções energia que queremos otimizar têm uma estrutura bastante complexa, com um grande número de mínimos locais, separados por descontinuidades e barreiras de alta energia. Os métodos de otimização genéricos e exatos geralmente param no primeiro mínimo local encontrado. Portanto, encontrar um mínimo local não é suficiente; precisamos encontrar um ponto com energia baixa — se não o mínimo global, pelo menos um mínimo local com energia comparável.

Este é um problema muito mais difícil, de caráter principalmente combinatório. Métodos gerais eficientes para este problema — recozimento simulado, algoritmos genéticos, redes neurais — são bastante recentes, e portanto bastante imprevisíveis: não existe ainda nenhuma teoria que permita prever seu desempenho em problemas complexos como o nosso. Na verdade, é fácil provar que o problema geral de otimização combinatória, que esses métodos pretendem resolver, é NP-completo ou pior. Portanto, tudo o que se pode esperar desses métodos é que eles sejam eficientes para as funções "típicas" de uma certa aplicação.

Entretanto, estes métodos gerais de otimização combinatória parecem ser pouco aplicáveis ao nosso problema. Eles pressupõem uma codificação discreta efetiva dos estados do problema combinatório, isto é, os mínimos locais, e que é impraticável devido à complexidade das nossas funções energia. Além disso, a experiência com estes métodos indica que provavelmente eles seriam demasiados lentos para nossas necessidades.

Em vista dessas dificuldades e incertezas, decidimos não investir tempo nestes métodos. Em seu lugar, usamos um técnica trivial de randomização, que consiste em aplicar as técnicas de otimização local a k configurações iniciais geradas aleatoriamente, e, dentre as configurações resultantes, devolver a que tem energia mínima.

5.2 Métodos genéricos e exatos de otimização

Os métodos de otimização local que serão abordados neste trabalho são de um tipo bem conhecido em minimização irrestrita; são os *métodos de descida* ou *métodos de minimização direcional*.

Todos estes métodos têm o mesmo passo central: a partir de uma configuração corrente $x^{(k)}$, eles escolhem uma direção $d^{(k)}$ na qual a função f provavelmente diminui, e procuram encontrar um escalar $\alpha^{(k)}$ tal que $f(x^{(k)} + \alpha^{(k)} d^{(k)})$ seja menor que o mínimo encontrado até o momento. [1, 19]. Os vários métodos de otimização diferem entre si pela maneira de escolher $x^{(k)}$, $d^{(k)}$ e $\alpha^{(k)}$.

Em nosso trabalho fizemos uso de quatro desses métodos: minimização coordenada a coordenada (que denotaremos por `Coord`), descida pelo gradiente (`Grad`), o método do simplexo de Nelder-Mead (`Simplex`), e o método das direções principais de Powell e Brent (`Praxis`). Nas próximas seções, vamos descrevê-los em linhas gerais, sem descer a detalhes de implementação.

5.2.1 Minimização coordenada a coordenada

Neste método de otimização, o ponto $x^{(k)}$ é a configuração de menor energia encontrada até o momento, e as direções $d^{(k)}$ são os próprios eixos de coordenadas $e_1, e_2, \ldots, e_n$ $\mathbb{R}^n$, nessa ordem, repetidas circularmente. Para cada direção $d^{(k)} = e_i$, o comprimento do passo $\alpha^{(k)}$ é determinado aplicando-se um método de minimização unidimensional à função energia, como por exemplo o método de Brent [9].

Ou seja, a cada iteração o método `Coord` ajusta apenas uma coordenada de um vértice da triangulação. Ele modifica a primeira coordenada x do primeiro vértice até encontrar um mínimo da função. Em seguida ele varia a segunda coordenada y procurando um novo mínimo; e o mesmo para a coordenada z. Estes ajustes são aplicados a todos os vértices, um de cada vez, circularmente.

Eventualmente, à medida que a configuração se torna mais macia, estas otimizações unidimensionais ficam cada vez menos eficientes; pois passa a ser necessário mover vários vértices simultaneamente para obter diminuições significativas da energia. Em geral, para funções do $\mathbb{R}^n$, se suas segundas derivadas são muito maiores em algumas direções do que em outras, e essas direções não coincidem com os eixos das coordenadas, os passos acima serão muito pequenos em comparação com a distância entre $x^{(k)}$ e o mínimo procurado.

Para atenuar este problema, depois de cada $n+1$ iterações do método acima, fazemos uma iteração extra, usando a direção "diagonal" $d^{(k+1)} = x^{(k)} - x^{(k-n)}$. A experiência indica que este passo avança significativamente na direção do mínimo, pois de certa forma ele resume e extrapola a informação sobre f que foi coletada pelos $n+1$ passos anteriores.

Apesar de sua simplicidade, o método `Coord` mostrou-se bastante eficaz para nosso problema, como veremos na seção 5.3 A razão é que para muitas energias (dobramento, espalhamento, excentricidade, etc.), a energia de uma configuração aleatória é dominada por termos que dependem da posição de cada vértice em relação a seus vizinhos. Nesse caso, a otimização da energia para um único vértice causa uma grande diminuição na energia total. Assim, as otimizações unidimensionais do método `Coord`, aplicadas no início, proporcionam grandes diminuições da energia a um custo relativamente pequeno.

5.2.2 Descida pelo gradiente

O gradiente de uma função f de $\mathbb{R}^n$ para $\mathbb{R}$ é o vetor

$$\nabla f(x_1, \ldots, x_n) = \left(\frac{\partial f}{\partial x_1}(x_1, \ldots, x_n), \ldots, \frac{\partial f}{\partial x_n}(x_1, \ldots, x_n) \right)$$

O método de *descida pelo gradiente* (`Grad`) tenta seguir a trajetória $x(t)$ definida pela equação diferencial

$$\frac{dx}{dt} = -\nabla f(x(t)) \tag{5.1}$$

partindo da configuração inicial $x(0) = x^{(0)}$, até atingir um ponto onde $\nabla f = 0$ (que, com probabilidade 1, será um mínimo local de f). Note que esta equação obriga a configuração a evoluir sempre na direção de maior decrescimento da energia.

Mais precisamente, o método `Grad` segue a trajetória (5.1) acima usando um simples integrador de Euler para equações diferenciais ordinárias [32]. Este método consiste em calcular cada nova configuração $x^{(k+1)}$ a partir da configuração anterior $x^{(k)}$, pela fórmula

$$x^{k+1} = x^{(k)} - \alpha^{(k+1)} \nabla f(x^{(k)})$$

O passo $\alpha^{(k+1)}$ é escolhido automaticamente pelo integrador, de modo a tentar manter o erro de integração limitado.

De modo geral, algoritmos de otimização que usam o gradiente convergem mais rapidamente do que algoritmos que usam apenas o valor da função. Suas principais desvantagens são a sensitividade a estrutura "miscroscópica" da função (como por exemplo os falsos mínimos locais criados por erros de arredondamento), e a necessidade de se calcular o gradiente da função f.

À primeira vista, o cálculo do gradiente de funções complexas, como as que usamos, pode parecer excessivamente dispendioso. Entretanto, Bauer e Strassen [6] mostraram que, aplicando-se sistematicamente a regra da cadeia a cada operação, é possível calcular o gradiente de qualquer função matemática a um custo bem modesto, apenas três ou quatro vezes o custo de calcular a própria função. Com a técnica de Bauer e Strassen, o custo de se calcular o gradiente é quase sempre compensado pela aceleração da convergência que pode ser obtida com o mesmo.

5.2.3 O método do simplexo de Nelder-Mead

O método do simplexo de Nelder-Mead [32], que denotaremos pela sigla `Simplex`), é um algoritmo clássico de otimização não-linear que não exige o cálculo do gradiente. A idéia deste método é calcular o valor da função energia nos vértices de um simplexo n-dimensional (isto é, em $n+1$ pontos do $\mathbb{R}^n$), e "caminhar" com esse simplexo na direção aproximada do mínimo, movendo um vértice de cada vez..

No algoritmo de Nelder-Mead, o progresso do simplexo é feito através de três operações básicas, *reflexão*, *expansão*, e *contração*. A figura 5.1 mostra estas três operações para o caso bi-dimensional.

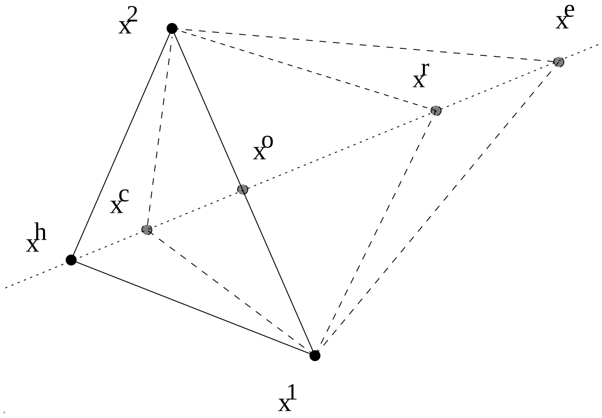


Figura 5.1: As três operações do método `Simplex`: reflexão (x^r), expansão (x^e), e contração (x^c).

Nesta figura, x^h representa o vértice correspondente ao maior valor da função objetivo. Os pontos x^o, x^r, x^e e x^c são calculados da seguinte forma

$$\begin{aligned} x^o &= \frac{1}{n} \sum_{\substack{1 \leq i \leq n \\ i \neq h}} x^i \\ x^r &= (1+\alpha)x^o - \alpha x^o \\ x^e &= \gamma x^r + (1-\gamma)x^o \\ x^c &= \beta x^h + (1-\beta)x^o \end{aligned}$$

onde α, γ, e β são parâmetros do método. Os valores sugeridos pelos autores são $\alpha = 1$, $\beta = 1/2$, $\gamma = 2$. (Nos nossos testes, entretanto, obtivemos melhores resultados com $\alpha = 0.896$, $\beta = 0.523$, $\gamma = 3.285$.). O método parte de um simplexo inicial, executando a seqüência de passos abaixo, até que atinja o critério de parada estabelecido:

1. Gere x^r fazendo uma operação de reflexão.

2. Se x^r é melhor que a melhor configuração corrente, então:

   2.1. Gere x^e fazendo uma operação de expansão.

   2.2. Se x^e é melhor que a melhor configuração corrente, então:

2.2.1. Troque a pior configuração por x^e.

2.3. Senão

2.3.1. Troque a pior configuração por x^r.

3. Senão, se x^r é pior que a segunda pior configuração, então

3.1. Se x^r é melhor que a pior configuração, então

3.1.1. Troque a pior configuração por x^r.

3.2. Gere x^c fazendo uma operação de contração.

3.3. Se x^c é melhor que a pior configuração, então

3.3.1. Troque a pior configuração por x^c.

3.4. Senão

3.4.1. Encolha o simplexo na direção da melhor configuração pelo fator 0.5.

4. Senão

4.1. Troque a pior configuração por x^r.

As principais vantagens deste método são a facilidade de programação e a sua aplicabilidade a funções bastante genéricas. Suas principais desvantagens são a convergência relativamente lenta e a possibilidade do simplexo encolher numa direção até ficar contido num hiperplano do $\mathbb{R}^n$, o que restringiria a busca a esse hiperplano.

Na verdade, na nossa implementação do método Nelder-Mead, modificamos o algoritmo de propósito de forma a permitir o uso de um simplexo sub-dimensional, cujo número de vértices m é menor que o número de variáveis n.

A motivação para esta mudança é diminuir o custo do algoritmo, pois o custo de um passo é proporcional ao produto mn, e o número de passos necessários para atingir o mínimo (com uma precisão fixa) cresce rapidamente com m. Mesmo os complexos mais simples produzem triangulações contendo várias centenas de vértices, o que significa que a função energia tem milhares de parâmetros. Por exemplo, no caso de uma configuração com 1000 vértices, teríamos $n = 3000$ variáveis; o simplexo completo teria $m = 3001$ pontos, e seu armazenamento exigiria uma matriz de 3001×3000 elementos (aproximadamente 30 megabytes de memória). Com nosso algoritmo modificado, poderíamos tratar esta configuração usando, por exemplo, um simplexo sub-dimensional com apenas $m = 100$ pontos, reduzindo o tamanho da matriz para $100 \times 3000 = 300.000$ elementos (aproximadamente 1 megabyte).

Naturalmente, esta mudança restringe a busca do mínimo a um sub-espaço do $\mathbb{R}^n$, que consiste das combinações afins dos m vértices do simplexo inicial. Este sub-espaço quase que certamente não contém o mínimo global da função energia. Entretanto, quase todas as configurações do espaço completo $\mathbb{R}^n$ tem energia alta. Para qualquer definição interessante de "suavidade" as superfícies "suaves" são um subconjunto muito pequeno de $\mathbb{R}^n$. Portanto, se as configurações escolhidas para formar o simplexo inicial forem previamente "amaciadas", nossa modificação estará de certa forma concentrando a busca numa parte especialmente "interessante" do $\mathbb{R}^n$. Na seção 5.4 veremos algumas técnicas que podem ser usada para esse "amaciamento" do simplexo inicial.

5.2.4 O método de Powell e Brent

O método de otimização que designamos por `Praxis` é baseado no algoritmo das direções principais de Powell [19, 32, 1], posteriormente melhorado por Brent [9].

O método original de Powell trabalha com um conjunto de $n+1$ configurações $p^{(0)}, \ldots, p^{(n)}$, e n direções $u^{(1)}, \ldots, u^{(n)}$. Inicialmente as direções $u^{(i)}$ são os vetores da base canônica, e $p^{(0)}$ é uma configuração inicial dada. O método parte de um ponto inicial $p^{(0)}$ e repete a seqüência de passos abaixo, até que a função pare de decrescer:

1. Para $i = 1, \ldots, n$, determine o mínimo da função a partir de p^{i-1} ao longo da direção $u^{(i)}$, e chame este ponto de $p^{(i)}$;
2. Para $i = 1, \ldots, n-1$, faça $u^{(i)} \leftarrow u^{(i+1)}$;
3. Faça $u^{(n)} \leftarrow p^{(n)} - p^{(0)}$;
4. Determine o mínimo da função a partir de $p^{(n)}$ na direção $u^{(n)}$, e guarde este ponto em $p^{(0)}$.

Infelizmente, há um problema com o método acima. Devido ao fato dele descartar, em cada estágio, a direção $u^{(1)}$, o conjunto de direções tende a ficar linearmente dependente. Quando isto acontece, a busca do mínimo da função fica restrito a um subespaço do $\mathbb{R}^n$. A versão proposta por Brent [9], que nós usamos, resolve este problema mantendo os vetores $u^{(i)}$ ortogonais durante o processo todo. Este ajuste naturalmente aumenta o custo de cada iteração, mas os ganhos em robustez e velocidade de convergência compensam este custo.

5.3 Comparação entre os métodos de otimização

Nesta seção descrevemos os testes que efetuamos para comparar o desempenho dos métodos descritos acima.

A triangulação utilizada nestes testes, que denominamos `tetra`, foi obtida a partir de um complexo celular com topologia de tetraedro, modelado com ladrilhos de ordem 3×3. A triangulação resultante (veja a figura 5.2) tem 110 vértices, 216 faces, e 324 arestas.

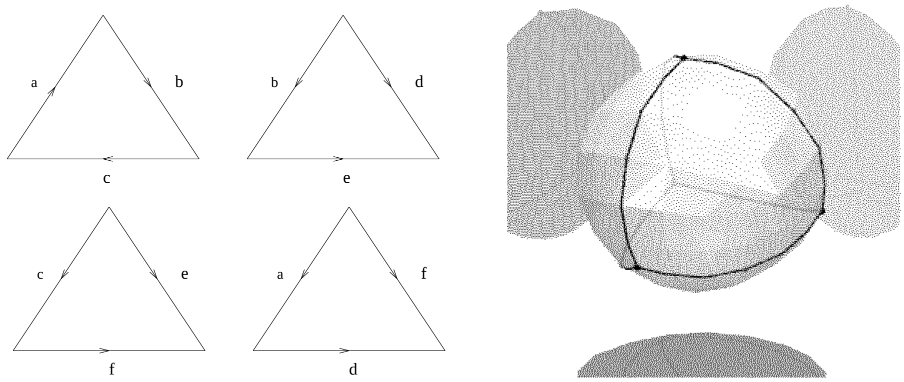


Figura 5.2: O complexo `tetra` utilizado nos testes.

Em todos os casos, a função energia utilizada foi uma combinação das energias $\mathcal{E}^{dobr}$ e $\mathcal{E}^{prox}$. Os testes foram realizados numa estação de trabalho *SUN SPARCstation 1+*.

5.3.1 Comparação entre `Simplex`, `Praxis` e `Coord`

Apesar dos livros texto recomendarem o método de descida pelo gradiente como tendo convergência mais rápida, no início de nossos testes nós evitamos usá-lo, pois acreditavamos que o cálculo do gradiente das nossas funções energia seria muito complexo e demorado, e que os métodos sem gradiente teriam mais chances de ultrapassar as numerosas barreiras entre os mínimos locais dessas funções.

Assim, implementamos e testamos inicialmente apenas o método do simplexo de Nelder-Mead (`Simplex`), o método das direções principais de Powell e Brent (`Praxis`), e o método de minimização coordenada a coordenada (`Coord`). É possível que o comportamento do `Simplex` seja devido as simplificações que adotamos. Infelizmente, não tivemos tempo para investigar este item.

As figuras 5.3 e 5.4 abaixo mostram a evolução da energia da melhor configuração obtida por cada método, em função do tempo de processamento (em segundos). Este tempo inclui tanto os cálculos da função energia quanto o tempo gasto pelo próprio algoritmo de otimização.

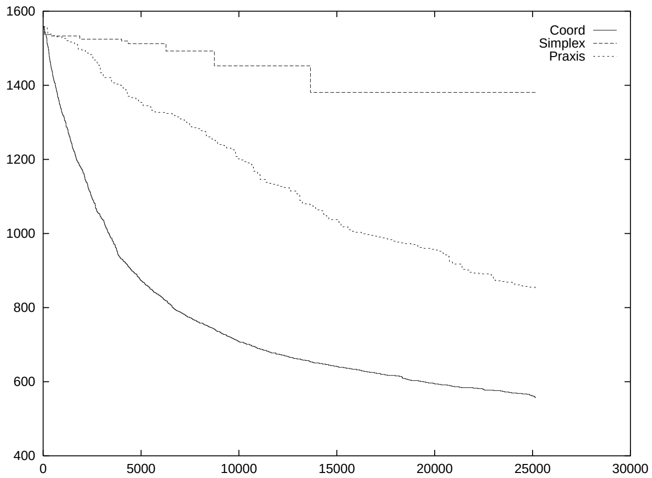


Figura 5.3: Comparação dos métodos `Simplex`, `Praxis`, e `Coord` partindo de uma configuração aleatória (energia $\times$ segundos de CPU).

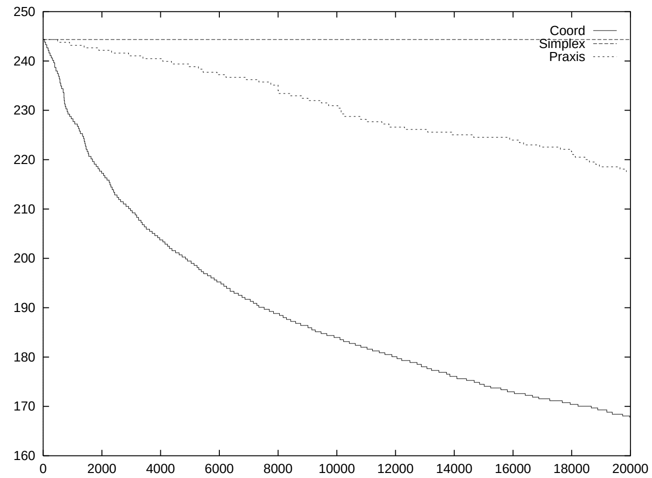


Figura 5.4: Comparação dos métodos `Simplex`, `Praxis`, e `Coord` partindo de uma configuração "amaciada" (energia $\times$ segundos de CPU).

Na figura 5.3, os três métodos foram aplicados a uma configuração inicial aleatória da triangulação `tetra`, onde cada coordenada era um número aleatório independente, uniformemente distribuído entre -1 e $+1$. Na figura 5.4 os três métodos foram aplicados a uma configuração inicial previamente "amaciada" pelos métodos heurísticos descritos na seção 5.4.

Como se pode notar, nos dois testes o método `Coord` progrediu muito mais rapidamente que os outros dois. Este resultado foi bastante inesperado; na verdade, nossa intenção ao implementar o método `Coord` era usá-lo apenas como ponto de referência nas comparações, pois a literatura dá a entender que ele é muito mais lento que os outros dois.

A explicação para este resultado paradoxal está nas peculiaridades da nossa função objetivo. Todas as funções energia que usamos são somas de um grande número de termos, sendo que cada termo depende de apenas alguns poucos vértices da configuração. A rotina que calcula a energia leva esse fato em conta, e só recalcula um termo se os vértices envolvidos mudaram de posição desde a chamada anterior. Portanto, no teste do método `Coord`, onde apenas um vértice muda de cada vez, cada cálculo da função energia era aproximadamente 10 vezes mais rápido do que nos testes de `Praxis` e `Simplex`.

Para confirmar esta explicação, mostramos nas figuras 5.5 e 5.6 os mesmos dados das figuras 5.3 e 5.4, mas usando como abscissa o número de chamadas da rotina de cálculo da energia, em vez do tempo de processamento. Como se pode ver, sob este critério o método `Praxis` é realmente melhor, como diz a literatura.

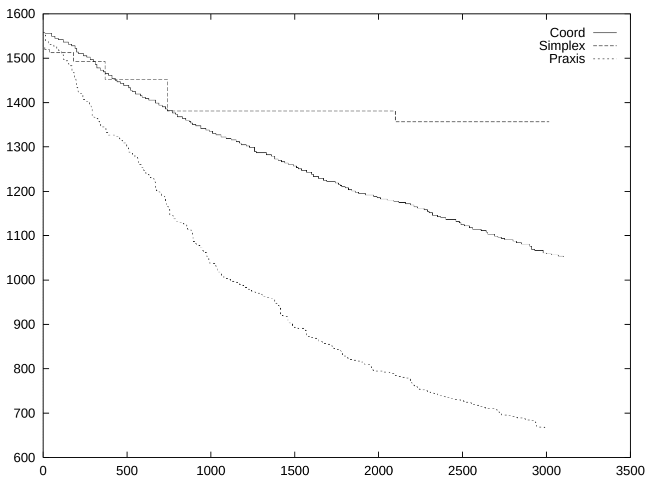


Figura 5.5: Comparação dos métodos `Simplex`, `Praxis`, e `Coord` partindo de uma configuração aleatória (energia $\times$ chamadas da função).

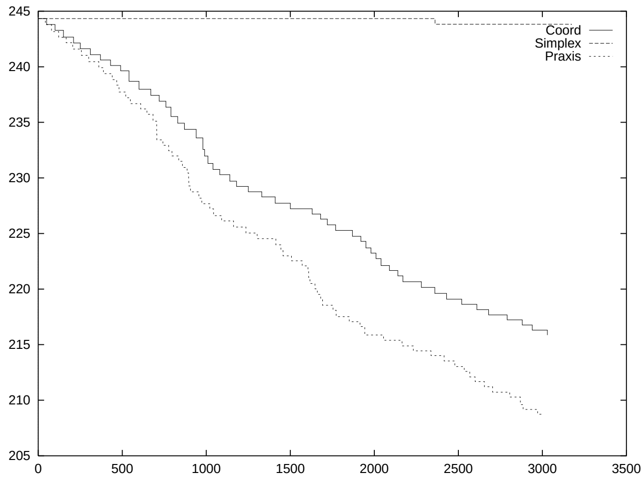


Figura 5.6: Comparação dos métodos `Simplex`, `Praxis`, e `Coord` partindo de uma configuração "amaciada" (energia $\times$ chamadas da função).

Note-se que os métodos `Praxis` e `Simplex` exigem memória proporcional a n^2 para uma função de n variáveis, enquanto que o método `Coord` usa memória proporcional a n apenas. No nosso caso, onde n está na casa dos milhares, esta é uma consideração importante. Em parte por este motivo, e em parte pelos resultados acima, desistimos de utilizar os métodos `Praxis` e `Simplex`.

5.3.2 Comparação entre `Grad` e `Coord`

Numa segunda etapa de testes, comparamos o desempenho do método de descida pelo gradiente(`Grad`) e minimização coordenada a coordenada (`Coord`).

Da mesma forma que o teste anterior, as figuras 5.7 e 5.8 mostram a evolução da energia da melhor configuração obtida por cada método, em função do tempo de processamento (em segundos). Ainda como no teste anterior, o gráfico da figura 5.7 corresponde a uma configuração aleatória enquanto o outro corresponde a uma configuração previamente "amaciada".

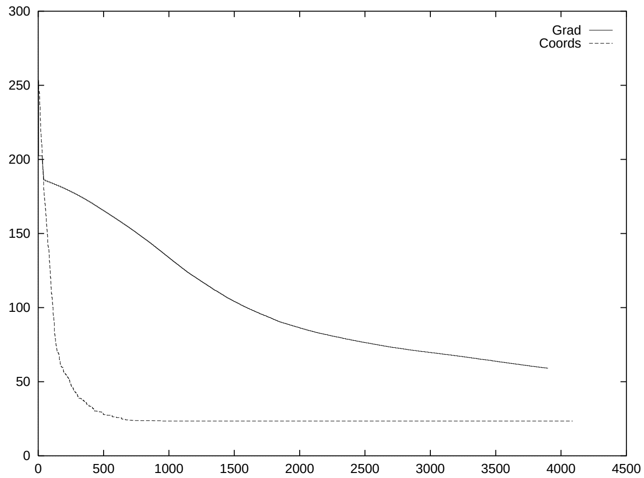


Figura 5.7: Comparação dos métodos `Grad` e `Coord` partindo de uma configuração aleatória (energia $\times$ segundos de CPU).

Como se pode notar, partindo de uma configuração aleatória, o método `Grad` leva muito mais tempo para se aproximar de um mínimo local. A razão é que, nas vizinhanças de uma configuração aleatória, a função possui muitos picos e vales, que atrapalham o funcionamento do método `Grad`. Nessas condições, a maior velocidade de cálculo da função, propiciada pelo método `Coord`, faz com que este seja o mais rápido dos dois.

Já no segundo gráfico, onde os métodos partem de uma configuração amaciada com pouco mínimos locais, a convergência do método `Grad` é mais rápida que a do `Coord`, como diz a literatura.

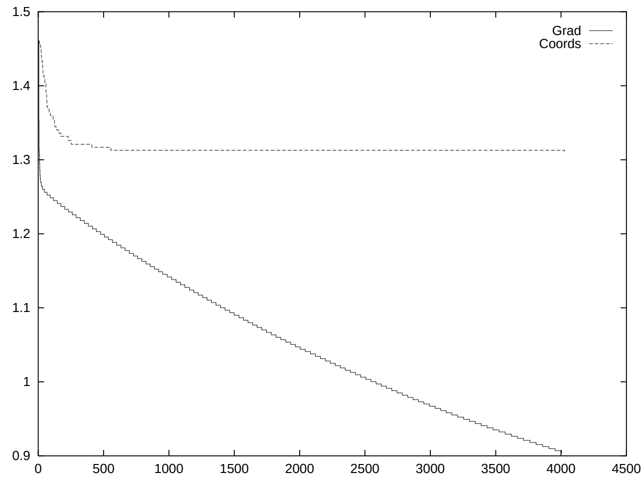


Figura 5.8: Comparação dos métodos `Grad` e `Coord` partindo de uma configuração "amaciada" (energia $\times$ segundos de CPU).

Esta análise é confirmada pelas duas figuras seguintes (5.9 e 5.10), que mostram a evolução da energia em função do número de iterações. Quando a comparação se faz por este critério, vemos que o método `Grad` é o melhor, mesmo partindo de uma configuração aleatória.

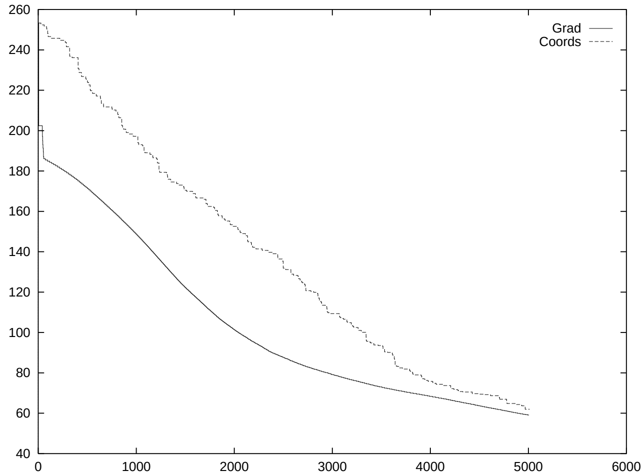


Figura 5.9: Comparação dos métodos `Grad` e `Coord` partindo de uma configuração aleatória (energia $\times$ chamadas da função).

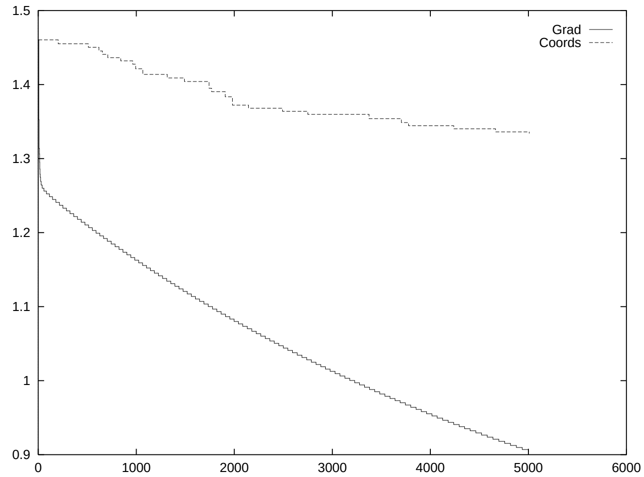


Figura 5.10: Comparação dos métodos `Grad` e `Coord` partindo de uma configuração "amaciada" (energia $\times$ chamadas da função).

Em resumo, no limite das nossas experiências, acreditamos que o método `Coord` é o mais eficiente para configurações aleatórias, enquanto que o método `Grad` é melhor para configurações suficientemente "amaciadas". Os métodos `Simplex` e `Praxis` parecem ser muito menos eficientes que ambos.

5.4 Métodos Heurísticos

Os métodos heurísticos são especializados para nosso problema em dois sentidos. Em primeiro lugar, eles pressupõem que os argumentos da função energia são coordenadas dos vértices de uma triangulação, e utilizam as relações de vizinhança entre esses vértices. Em segundo lugar, cada um desses métodos tenta minimizar uma função energia específica.

Todos os métodos heurísticos que usamos são iterativos e locais: a cada passo, eles ajustam a posição de um vértice e possivelmente de seus vizinhos imediatos. Uma aplicação completa de um método heurístico consiste em efetuar este ajuste uma vez para cada vértice da triangulação.

Estes métodos heurísticos são apenas aproximados: eles geralmente reduzem rapidamente a energia de configurações muito irregulares, mas têm efeito contrario em superfícies próximas do mínimo. Ou seja, a configuração obtida usando-se um método heurístico tem energia baixa, mas não se encontra num mínimo local. Portanto, os métodos heurísticos são úteis apenas no inicio da otimização, para gerar um ponto de partida para os métodos exatos. Além disso, geralmente não vale a pena aplicar mais do que algumas dezenas de iterações desses métodos.

5.4.1 Heurística de desdobramento

A heurística de desdobramento consiste em ajustar um vértice v de cada vez, mantendo os demais fixos, movendo-o para uma posição que aproximadamente minimiza a energia de dobramento das arestas da estrela de v (a união dos triângulos incidentes a v).

Para tornar o cálculo praticável, restringimos a nova posição do vértice v à reta $a + t\vec{n}$, onde a é o baricentro dos vértices vizinhos $v_0, v_1, \ldots, v_k$, e $\vec{n}$ é a normal aproximada da superfície no vértice v, calculada a partir desses vizinhos pelas fórmulas da seção 4.5. Veja a figura 5.11.

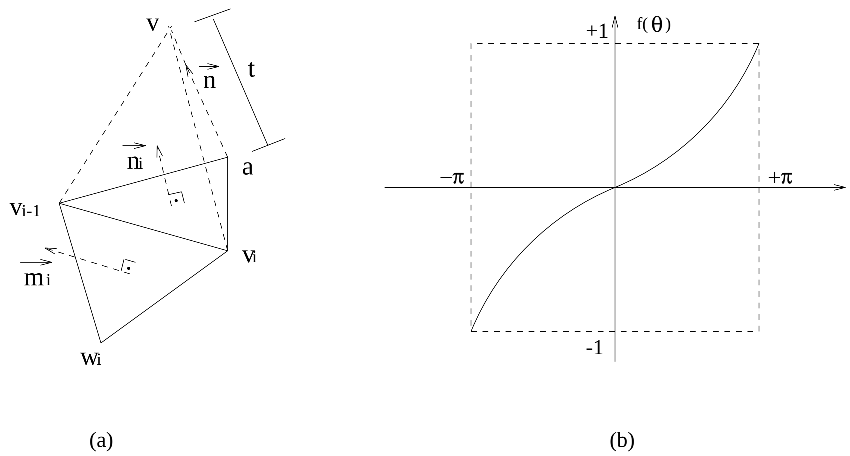


Figura 5.11: (a) Variáveis relevantes para o amaciamento. (b) A função $f(\theta)$.

O parâmetro t é escolhido de modo que ao deslocar v para $a + t\vec{n}$ estamos minimizando (aproximadamente) os ângulos diedrais externos às arestas $v_{i-1}v_i$.

Mais exatamente, para cada aresta $e_i = v_{i-1}v_i$, calculamos o ângulo diedral externo θ_i entre a face $v_{i-1}v_i a$ e a face wv_iv_{i-1} oposta à aresta e_i. Vetores n_i e m_i perpendiculares a esses dois triângulos são dados pelas fórmulas

$$\begin{aligned} d_i &= (v_i - v_{i-1}) \\ n_i &= di \times (a - v_i) \\ m_i &= (w_i - v_{i-1}) \times d_i \end{aligned}$$

de onde obtemos

$$\begin{aligned} \sin\theta_i &= \left(\frac{n_i}{\mid n_i \mid} \times \frac{m_i}{\mid m_i \mid}\right) \cdot \frac{d_i}{\mid d_i \mid} \\ \cos\theta_i &= \frac{n_i \cdot m_i}{\mid n_i \mid\mid m_i} \\ \theta_i &= \arctan(\sin\theta_i, \cos\theta_i) \end{aligned}$$

A energia de dobramento da aresta $v_{i-1}v_i$ seria mínima (nula) se o ângulo entre as faces $v_{i-1}v_iv$ e wv_iv_{i-1} fosse zero. Portanto, no que diz respeito a esta aresta, o valor ideal do parâmetro t deveria ser

$$t_i = h_i \tan\theta_i \tag{5.2}$$

onde h_i é a distância do baricentro a à aresta e_i,

$$h_i = \frac{|n_i|}{|v_i - v_{i-1}|} \tag{5.3}$$

Entretanto, a formula (5.2) da resultados indesejáveis quando o ângulo $|\theta_i|$ é grande. Em particular $\tan\theta_i$ é infinito ou negativo quando $\theta_i \geq \pi/2$. Portanto, em vez da formula (5.2) usamos

$$t_i = h_i f(\theta_i) \tag{5.4}$$

onde f é a função

$$f(\theta) = \left(\frac{1-\pi/2}{\pi^3}\right)\theta^3 + \frac{1}{2}\theta \tag{5.5}$$

Veja a figura 5.11(b). Note que $f(\theta)$ aproxima bem $\tan\theta$ para valores pequenos de θ.

O valor escolhido para o parâmetro t é a média dos valores ideais t_i apropriados para cada aresta, ponderados pelos comprimentos das arestas:

$$\frac{t = \sum_{i=1}^{k} |d_i| t_i}{\sum_{i-1}^{k} |di|} \tag{5.6}$$

Esta heurística tende a distribuir a energia de dobramento de maneira mais uniforme dentro da estrela de v, ou seja, a "amaciar" a superfície na vizinhança de v.

Ao aplicar-se essa heurística para todos os vértices, repetidas vezes, a energia de dobramento espalha-se por difusão em toda a superfície. Assim, a convergência é relativamente lenta. Intuitivamente, por analogia com processos físicos de difusão, podemos prever que o número de iterações necessárias para se obter uma configuração com qualidade determinada aumenta com o quadrado do diâmetro d da triangulação, no sentido da teoria de grafos. Por sua vez, para complexos típicos, o número de vértices tende a ser proporcional ao quadrado do diâmetro. Portanto, esperamos que o número de iterações seja proporcional ao número de vértices n, e o tempo total proporcional a n^2.

5.4.2 Heurística de equalização de ângulos

A heurística de ângulos consiste em ajustar os ângulos entre as m arestas que incidem num vértice u com o ângulo ideal $2\pi/m$. Esta heurística é aplicada a cada vértice u da triangulação, mantendo-o fixo e movendo os vértices vizinhos a u.

As novas posições dos vértices vizinhos são obtidas de modo que a projeção da sua respectiva aresta sobre o plano Φ que possui como normal o vetor $\vec{n}$, seja a que aproximadamente reduz a energia de ângulos. O vetor $\vec{n}$ é a normal aproximada da superfície no vértice u, como na heurística de desdobramento. Segue abaixo a descrição detalhada de como obter a nova posição dos vértices v_i.

Seja $N_u = \{v_0, v_1, \ldots, v_{m-1}\}$ o conjunto dos vértices vizinhos de u, numerados a partir de um vizinho v_0 escolhido aleatoriamente.

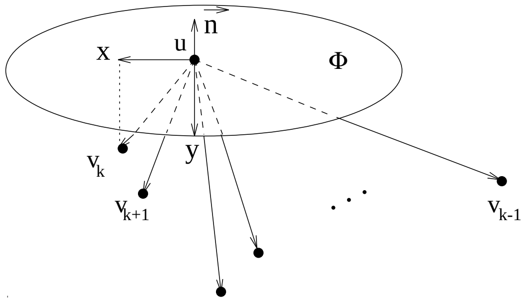


Figura 5.12: A estrela de um vértice u, o plano Φ e os eixos x e y definidos pelo vértice v_k.

Definimos eixos $\vec{x}$ e $\vec{y}$ sobre o plano Φ (veja figura 5.12) da seguinte forma:

$$\vec{y} = \frac{\vec{n} \times (v_0 - u)}{|\vec{n} \times (v_0 - u)|}$$

$$\vec{x} = \vec{y} \times \vec{n}$$

Então, para todo vizinho v_i, calculamos as projeções $v_i^{\perp}$ e $v_i^{||}$ do vetor $v_i - u$, respectivamente sobre a normal $\vec{n}$ e sobre o plano Φ:

$$v_i^{\perp} = ((v_i - u) \cdot \vec{n})\vec{n},$$

$$v_i^{||} = (v_i - u) - v^{\perp}$$

Com esses dados, calculamos a projeção "ideal" $w_i^{||}$ do mesmo vetor, supondo que ela tem o mesmo comprimento que $v_i^{||}$, mas forma ângulo $\theta_i = 2\pi/m$ com o eixo x:

$$w_i^{||} = \left|v_i^{||}\right|(\vec{x}\cos\theta_i + \vec{y}\sin\theta_i)$$

A nova posição de v_i é então

$$v_i = u + v_i^{\perp} + \alpha w_i^{||} + (1 - \alpha)v_i^{||}$$

onde α, um parâmetro da heurística, é um número entre 0 e 1. Nos nossos testes, verificamos que um valor adequado para α é 0.75.

5.4.3 Normalização

Um efeito colateral das heurísticas descritas acima é mudar ligeiramente o tamanho da configuração. Portanto, a aplicação repetida de uma tal heurística, sem maiores cuidados, levaria no limite a uma configuração degenerada, com todos os vértices coincidentes ou no infinito. Para corrigir este problema, após cada ciclo completo de uma heurística, nós aplicamos uma etapa de *normalização*, que amplia ou reduz a configura,cão de modo a manter seu tamanho e posição constante.

Mais precisamente, se os vértices são $\{v_1, \ldots, v_n\}$, a normalização consiste em efetuar os cálculos abaixo:

$$\begin{aligned} b &\leftarrow \frac{1}{n}\sum_{i=1}^{n} v_i \\ v_i &\leftarrow v_i - b(i+1,\ldots,n) \\ s &\leftarrow \frac{1}{n}\sum_{i=1}^{n} |v_i|^2 \\ v_i &\leftarrow \frac{v_i}{\sqrt{s}} \end{aligned}$$

Vale a pena observar que todas as heurísticas que usamos são puramente euclidianas, isto é, operações de translação, rotação, ou mudança de escala de $\mathbb{R}^3$ têm o mesmo efeito se aplicadas antes ou depois da heurística. Portanto, as etapas de normalização não alteram a forma da configuração limite das heurísticas, mas apenas mantêm seu tamanho.

Capítulo 6

Resultados

Neste capítulo apresentamos algumas realizações geométricas de complexos celulares obtidas usando nossa ferramenta. Além disso, apresentamos também algumas figuras que ilustram a influência das diversas energias na forma final da superfície.

Algumas das imagens abaixo (6.5 e 6.11) foram geradas utilizando-se o programa`X3D`, uma ferramenta visualização de poliedros para ambiente `UNIX/X11`. Devido as limitações dessa ferramenta, as arestas do complexo não foram indicadas.

As demais imagens foram produzidas pelo programa `POVRAY`, um traçador de raios de domínio público. A aparência suave dessas figuras é devida ao uso da técnica de Gouraud para eliminação de arestas na tonalização de superfícies poliédricas.

6.1 Toro

A figura 6.1 ilustra um modelo de colagem bem simples que denotamos por `torus`.

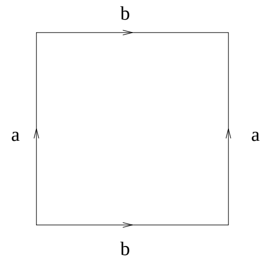


Figura 6.1: O modelo de colagem para o complexo `torus`

Trata-se do menor complexo que tem a topologia do toro, possuindo uma face de quatro lados, duas arestas, e um vértice. Para realizá-lo usamos ladrilhos de ordem 5×5, resultando numa triangulação (`torus-5`) com 200 triângulos, 300 arestas, e 100 vértices.

A figura 6.2 ilustra a otimização da energia desta triangulação. A primeira imagem mostra a configuração inicial, com coordenadas aleatórias; as três imagens seguintes mostram a configuração após 5, 30, e 100 iterações alternadas das heurísticas de desdobramento e equalização de ângulos.

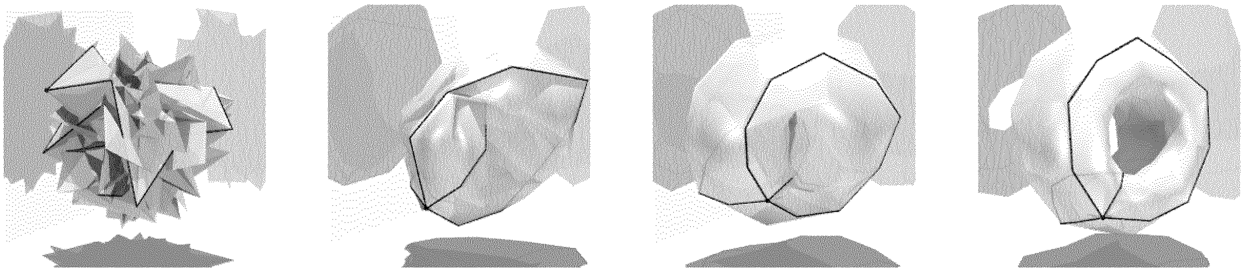


Figura 6.2: Realização heurística da triangulação `torus-5`.

Repetimos este teste cinco vezes, com configurações iniciais diferentes. Três das tentativas resultaram em superfícies com auto-interseções, e duas resultaram na solução ilustrada acima.

A figura 6.3 mostra novamente a configuração final da figura 6.2, e as configurações obtidas a partir da mesma pelo o método de otimização `Grad`, após respectivamente 100 e 1000 cálculos da função energia. A energia usada foi $\mathcal{E}^{dobr} + \mathcal{E}^{area} + \mathcal{E}^{excen} + 5\mathcal{E}^{prox}$.

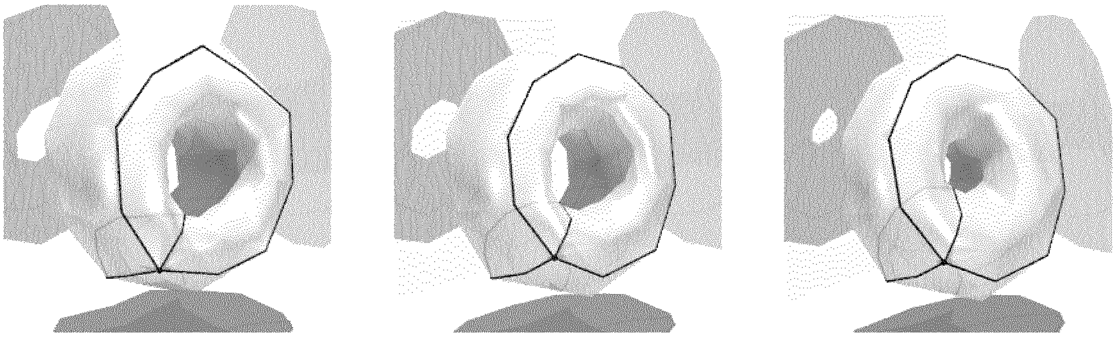


Figura 6.3: Otimização da energia da triangulação `torus-5`.

6.2 Garrafa de Klein

A figura 6.4 descreve o modelo de colagem do complexo celular `klein`, também com 1 face, 2 arestas, e 1 vértice, mas com a topologia da garrafa de Klein.

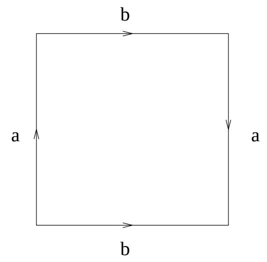


Figura 6.4: O modelo de colagem do complexo `klein`

Este complexo também foi realizado com ladrilhos de ordem 5×5, resultando numa triangulação `klein-5` com 200 triângulos, 300 arestas, e 100 vértices. A figura 6.5 mostra três etapas na evolução deste objeto: a configuração aleatória inicial, o estado após 5 aplicações da heurística de desdobramento, e o resultado após 10000 iterações do método `Simplex`.

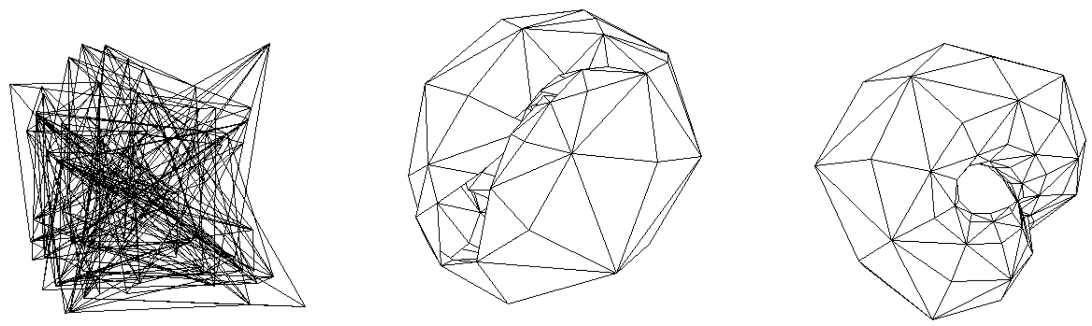


Figura 6.5: Evolução da triangulação `klein-5`.

A forma final acima pode ser entendida como um toro que foi cortado e costurado com um “trançamento”. Veja a figura 6.6.

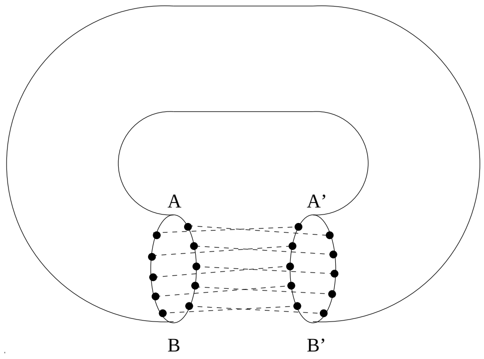


Figura 6.6: Estrutura do estágio final da figura 6.5.

Note que a superfície resultante não é completamente suave, pois sua curvatura nos pontos $A = A'$ e $B = B'$ é infinita. Este fato revela uma limitação de nossas funções energia, que não penalizam suficientemente estes defeitos.

6.3 Salsicha

A figura 6.7 descreve o modelo de colagem do complexo celular `sausage`: um tubo formado por 4 retângulos colados, fechado por duas faces de grau 2. O complexo tem 10 arestas e 6 faces.

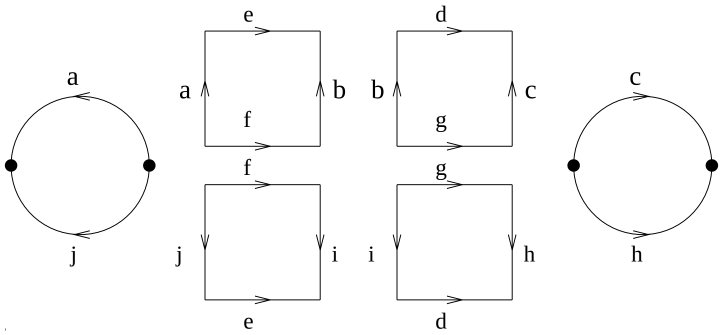


Figura 6.7: O modelo de colagem do complexo `sausage`

A modelagem do complexo `sausage` com ladrilhos de ordem 3×3 resultou na triangulação `sausage-3` com 360 triângulos, 540 arestas, e 182 vértices.

A figura 6.8 mostra a evolução deste modelo quanto submetido às heuristicas de desdobramento e de equalização de ângulos: a configuração aleatória inicial, o estado após 15 iterações de cada heurística, e o estado após 100 iterações.

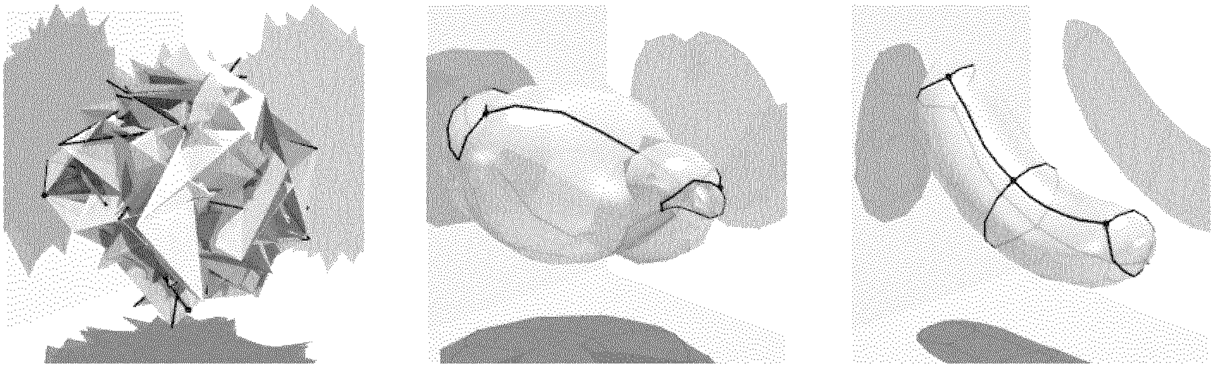


Figura 6.8: Evolução do modelo `sausage-3`.

Pode-se ver que neste caso também as heurísticas foram suficientes para eliminar todas as auto-intersecções; porém a superfície ainda tem uma curvatura desnecessária, e as faces nas extremidades estão relativamente pequenas.

A configuração da 6.9, abaixo, foi obtida submetendo-se a superfície final da figura 6.8 a 2000 iterações adicionais do método de otimização `Grad`, com a energia mista $\mathcal{E}^{dobr} + \mathcal{E}^{area} + \mathcal{E}^{ang} + \mathcal{E}^{excen}$. Pode-se ver que a área da superfície aumentou, epecialmente nas extremidades, devido à influência de $\mathcal{E}^{area}$; mas a curvatura supérflua ainda não foi eliminada.

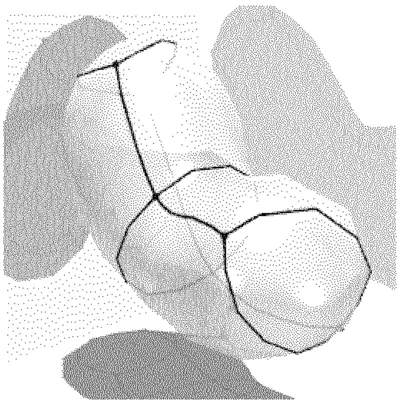


Figura 6.9: O modelo `sausage-3` otimizado.

6.4 Salame

A figura 6.10 descreve o modelo de colagem dos complexo celular `salami`, uma versão mais comprida do complexo `sausage`, com 18 arestas e 10 faces.

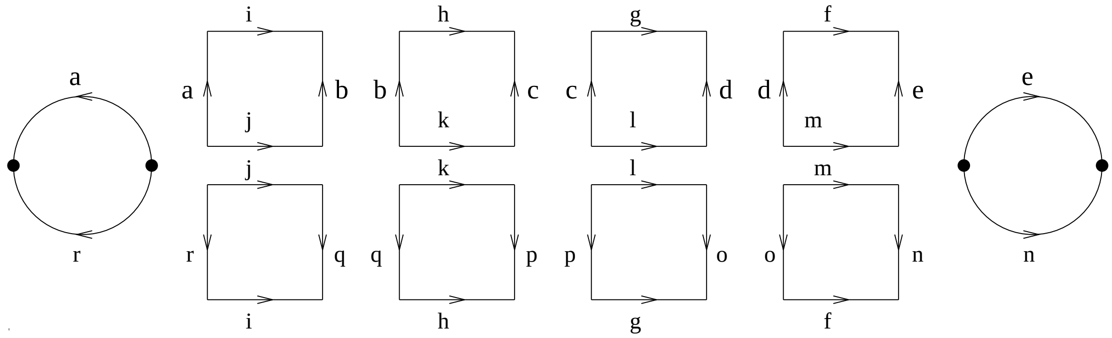


Figura 6.10: O modelo de colagem do complexo `salami`.

Este complex foi modelado com ladrilhos de ordem 5×5, resultando na triangulação `salami-5` com 1800 triângulos, 2700 arestas, e 902 vértices. A figura 6.11 mostra a otimização deste modelo: a configuração aleatória inicial, o estado após 5 aplicações da heurística de desdobramento, e o resultado após 10000 iterações do método `Simplex`.

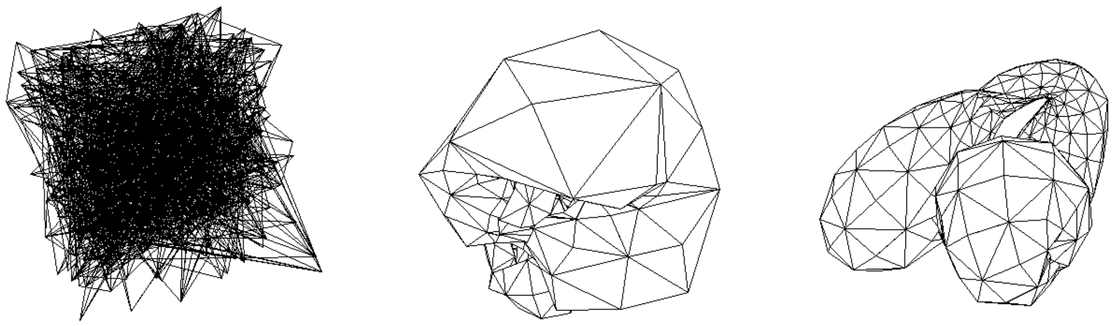


Figura 6.11: Evolução do modelo `salami-5`.

Como se pode ver desta figura, o aumento no número de variáveis retardou bastante a convergência, de modo que a configuração resultante está ainda mais longe da forma desejada, com um número ainda maior de trançamentos.

Só conseguimos progredir além deste ponto depois que implementamos a heurística de equalização de ângulos (seção 5.4.2) e substituímos o método `Simplex` por métodos mais eficientes, especialmente `Coord` e `Grad`.

6.5 Laranja

A figura 6.12 descreve o modelo de colagem para o complexo celular `orange`. A variedade deste complexo tem a topologia da esfera; o complexo consiste de dois vértices antípodas, ligados através de 7 arestas, formando 7 faces como se fossem gomos de uma laranja.

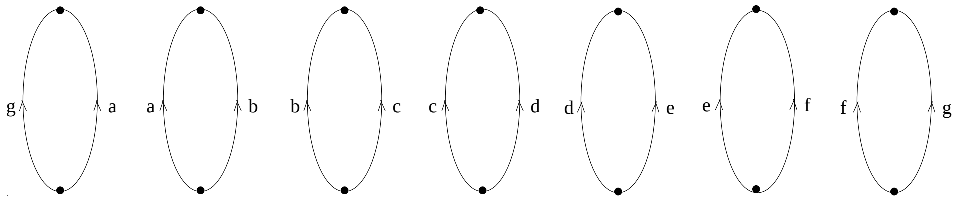


Figura 6.12: O modelo de colagem `orange`

Este complexo foi modelado com ladrilhos de ordem 3×3, resultando na triangulação `orange-3`, com 252 triângulos, 378 arestas, e 128 vértices. A Figura 6.13 mostra a configuração deste modelo após a aplicação de 100 passos das heurísticas de equalização de ângulos e desdobramento. Note que as arestas do complexo original se apresentam com formas homogêneas e igualmente distanciadas.

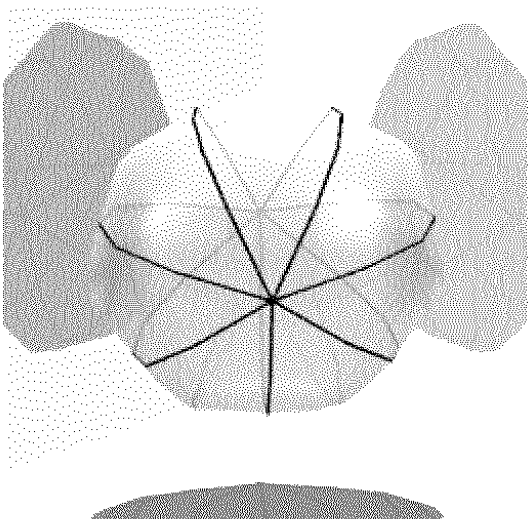


Figura 6.13: Configuração final do modelo `orange-3`.

6.6 Tritoro

A figura 6.14 descreve o modelo de colagem `tritorus`, com 12 faces, 24 arestas e 8 vértices, cuja superfície tem a topologia de um tritoro (uma esfera com três alças).

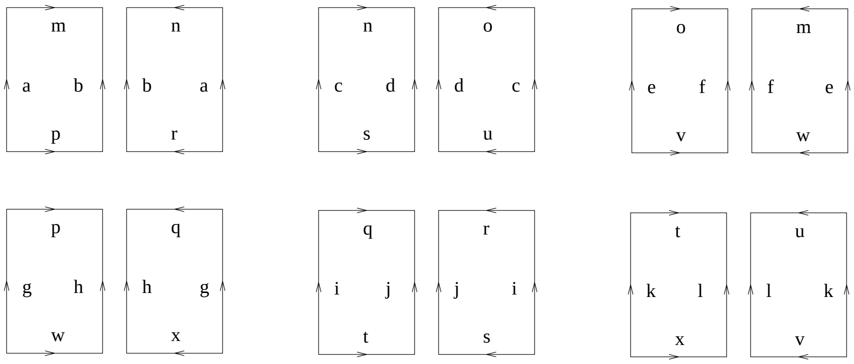


Figura 6.14: O modelo de colagem `tritorus`

Modelado com ladrilhos de ordem $k = 3$, este cmplexo produziu a triangulação `tritorus-3` com 864 triângulos, 1296 arestas, e 428 vértices.

Para realizar este exemplo, geramos 10 configurações aleatórias e para cada uma delas aplicamos 100 passos das heurísticas de equalização de ângulos e desdobramento, alternadamente. As configurações resultantes se apresentaram razoavelmente amaciadas e com uma forma próxima da ideal; porém 9 delas possuíam auto-intersecções. Nós manualmente selecionamos aquela sem auto-intersecções e aplicamos a ela alguns milhares de passos do método `Grad`. O resultado final pode ser visto na figura 6.15.

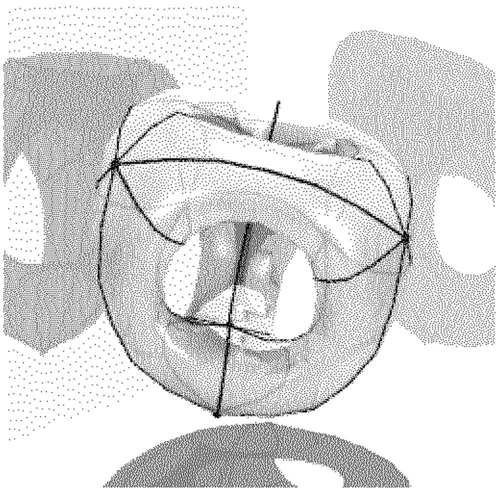


Figura 6.15: Configuração final do modelo `tritorus-3`.

6.7 Estrelas e não-estrelas

O próximo exemplo é particularmente interessante pois ilustra a motivação original do nosso trabalho, ou seja, a detecção de erros na topologia de modelos de fronteiras.

A figura 6.16 descreve o modelo de colagem do complexo celular `star`, com 15 faces, 25 arestas, e 12 vértices.

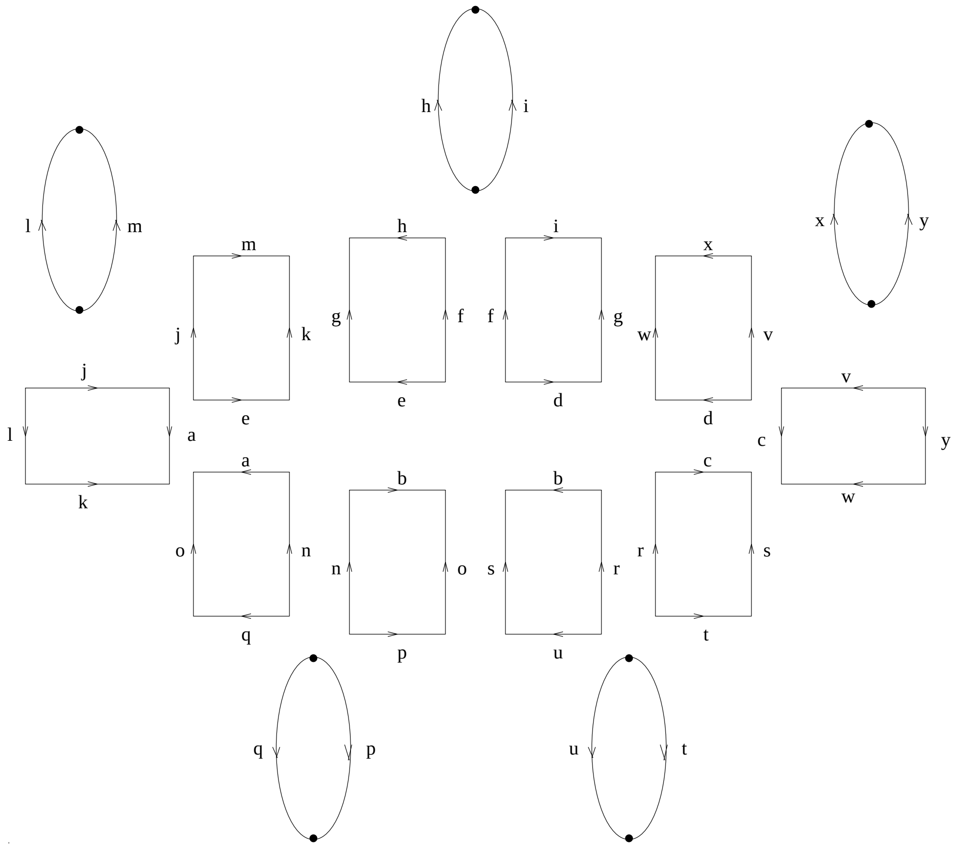


Figura 6.16: O modelo de colagem do complexo `star`

O complexo `star` pode ser entendido como uma variante do complexo `orange`, com 5 gomos em vez de 7, sendo que cada gomo é substituído por metade do modelo `sausage`, formando assim uma estrela de 5 pontas.

Na modelagem deste complexo usamos ladrilhos de ordem 3×3, resultando na triangulação `star-3` com 900 triângulos, 1350 arestas, e 452 vértices. A otimização deste modelo, usando as heurísticas e o método `Grad`, resultou na superfície da figura 6.17, muito diferente do que esperávamos:

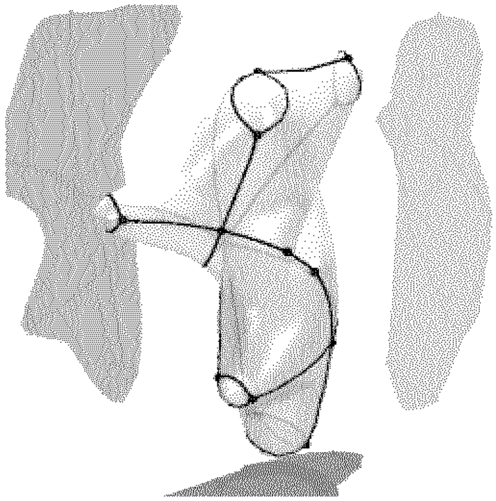


Figura 6.17: Primeira tentativa de construir o modelo `star-3`.

Esta imagem mostrou que o complexo que tínhamos construido não possuia a topologia desejada. Eventualmente descobrimos que a rotina de construção do mesmo incluia uma chamada incorreta da operação *Splice* (seção 3.2.2).

Depois de corrigir esse erro, e repetir a otimização, obtivemos a superfície ilustrada na figura 6.18.

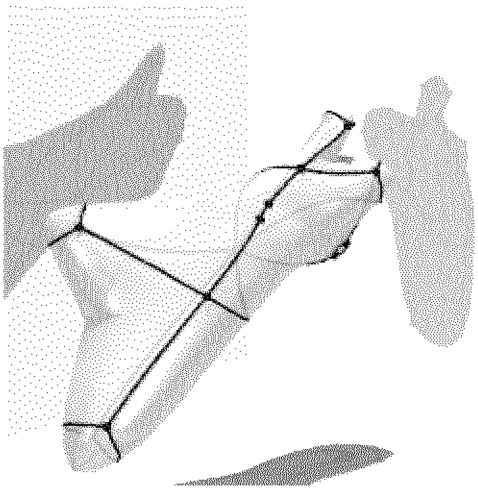


Figura 6.18: Segunda tentativa de construir o modelo `star-3`.

Como se pode notar, a topologia desta figura ainda não é a do complexo `star`; o que nos levou a descobrir um segundo erro no nosso algoritmo.

Após corrigir esse segundo erro, conseguimos finalmente uma superfície que se aproximava do desejado (figura 6.19). Note que, apesar dos cruzamentos, esta superfície esta bem próxima da forma ideal (uma estrela).

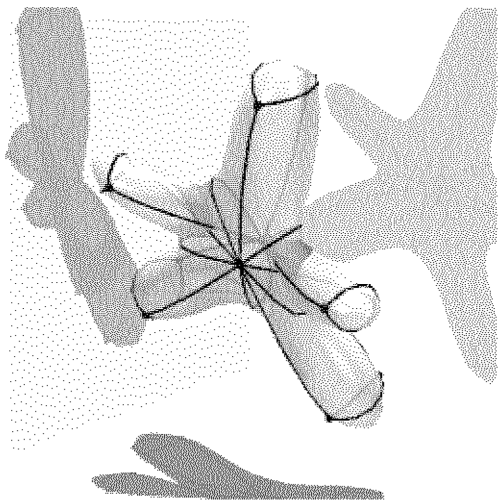


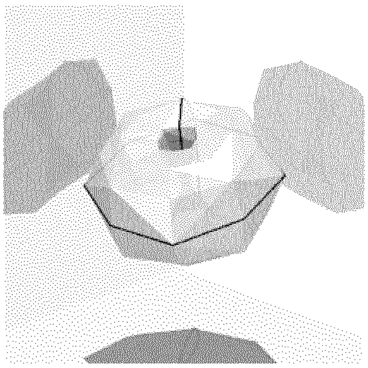
Figura 6.19: Configuração final obtida para modelo `star-3`. Agora o algoritmo de construção parece estar correto...

6.8 Influência das várias energias

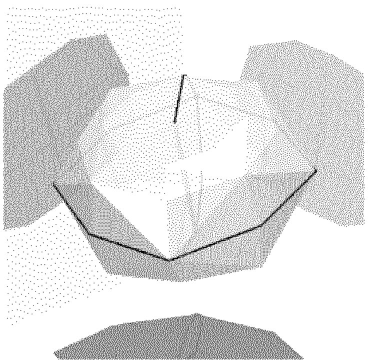
As figuras abaixo ilustram a variedade de formas que podem ser obtidas através da combinação de várias energias. Em todos os exemplos, o complexo `torus-3` (seção 6.1 foi submetido a 50 iterações da heurística de desdobramento e 20 iterações da heurística de equalização ângulos, e em seguida otimizado com 5000 iterações do método do `Grad`. As legendas em cada figura mostram as energias presentes em cada combinação e seus respectivos pesos.

6.8.1 Combinação $\mathcal{E}^{dobr} + \mathcal{E}^{excen} + \mathcal{E}^{prox}$

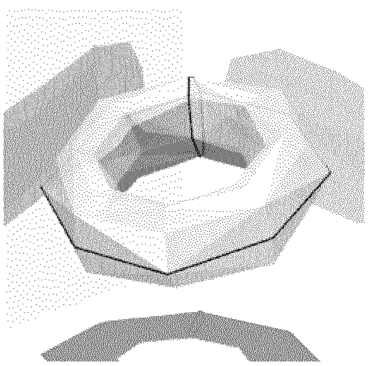
Esta é a combinação de energias que deu melhor resultado no complexo `torus`.



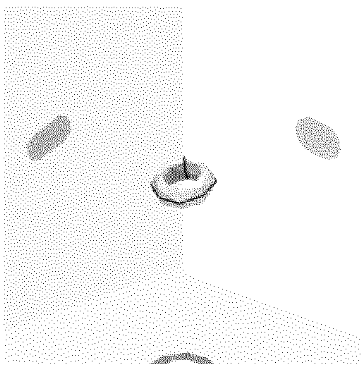
$0.33\mathcal{E}^{dobr} + 0.33\mathcal{E}^{excen} + 0.33\mathcal{E}^{prox}$



$0.83\mathcal{E}^{dobr} + 0.08\mathcal{E}^{excen} + 0.08\mathcal{E}^{prox}$



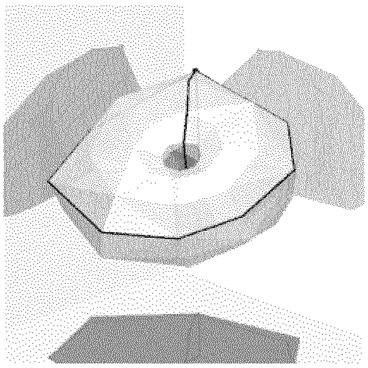
$0.08\mathcal{E}^{dobr} + 0.08\mathcal{E}^{excen} + 0.83\mathcal{E}^{prox}$



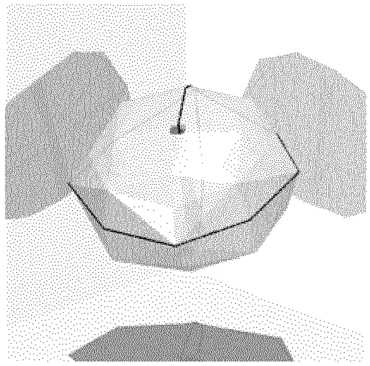
$0.08\mathcal{E}^{dobr} + 0.83\mathcal{E}^{excen} + 0.08\mathcal{E}^{prox}$

Note que, quando a energia $\mathcal{E}^{excen}$ está predominando sobre as outras duas, a figura tende a diminuir de tamanho. No limite, com apenas a $\mathcal{E}^{excen}$ presente, teríamos uma configuração degenerada, ou seja, todos os vértices coincidentes.

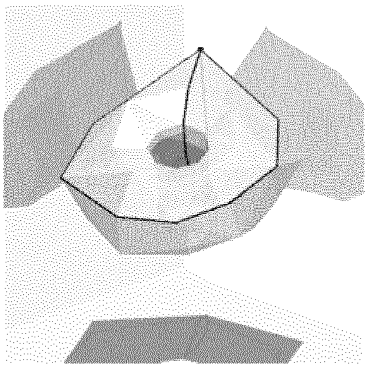
6.8.2 Combinação $\mathcal{E}^{dobr} + \mathcal{E}^{area} + \mathcal{E}^{ang}$



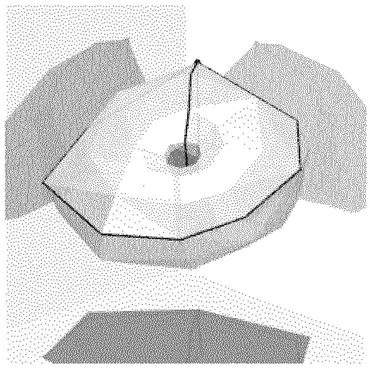
$0.33\mathcal{E}^{dobr} + 0.33\mathcal{E}^{area} + 0.33\mathcal{E}^{ang}$



$0.83\mathcal{E}^{dobr} + 0.08\mathcal{E}^{area} + 0.08\mathcal{E}^{ang}$



$0.08\mathcal{E}^{dobr} + 0.08\mathcal{E}^{area} + 0.83\mathcal{E}^{ang}$



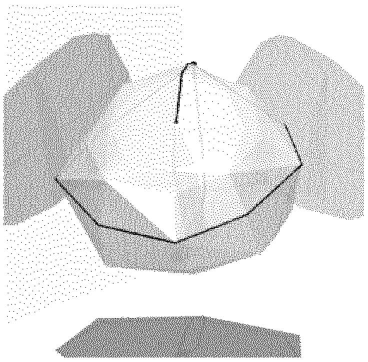
$0.08\mathcal{E}^{dobr} + 0.83\mathcal{E}^{area} + 0.08\mathcal{E}^{ang}$

Note que as energias $\mathcal{E}^{ang}$ e $\mathcal{E}^{area}$ em conjunto impedem que a minimização da energia $\mathcal{E}^{dobr}$ feche a abertura do toro. Por definição, a energia $\mathcal{E}^{dobr}$ diminui quando os ângulos diedrais entre as faces se aproximam de zero e/ou quando o tamanho das arestas se aproximam de zero. Assim, uma maneira de diminuir $\mathcal{E}^{dobr}$ é transferir toda a curvatura da superfície para as arestas nas bordas da abertura do toro, reduzindo ao mesmo tempo o comprimento dessas arestas para zero.

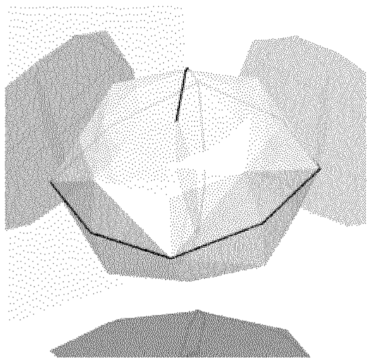
A energia $\mathcal{E}^{ang}$ impede que isso ocorra, pois tenta distribuir os ângulos ao redor de um vértice da forma mais homogênea possível, contribuindo para diminuir os ângulos diedrais entre as faces, na média; e $\mathcal{E}^{area}$ tenta manter a área dos ladrilhos com valor constante, forçando com que as arestas, na média, não sejam muito pequenas.

6.8.3 Combinação $\mathcal{E}^{dobr} + \mathcal{E}^{area} + \mathcal{E}^{prox}$

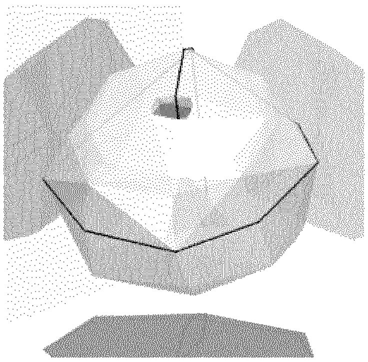
Nesta combinação é o termo $\mathcal{E}^{prox}$ que impede que a abertura do toro se feche. Note que quando diminuímos o peso da energia $\mathcal{E}^{prox}$, o diâmetro do toro se aproxima de zero.



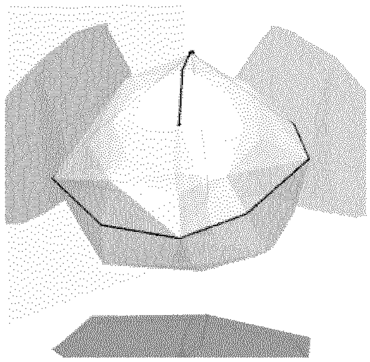
$0.33\mathcal{E}^{dobr} + 0.33\mathcal{E}^{area} + 0.33\mathcal{E}^{prox}$



$0.83\mathcal{E}^{dobr} + 0.08\mathcal{E}^{area} + 0.08\mathcal{E}^{prox}$



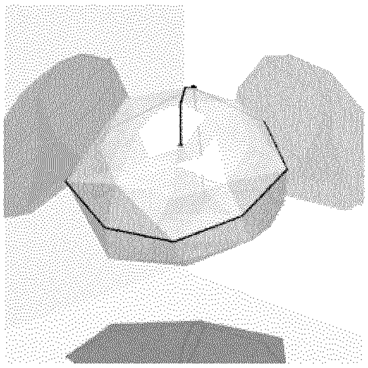
$0.08\mathcal{E}^{dobr} + 0.08\mathcal{E}^{area} + 0.83\mathcal{E}^{prox}$



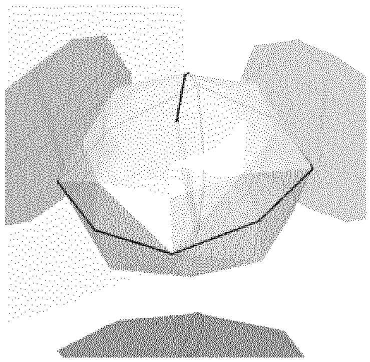
$0.08\mathcal{E}^{dobr} + 0.83\mathcal{E}^{area} + 0.08\mathcal{E}^{prox}$

6.8.4 Combinações com $\mathcal{E}^{dobr}$ e $\mathcal{E}^{excen}$

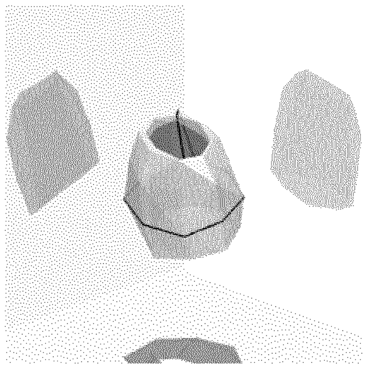
Novamente, note o efeito das energias $\mathcal{E}^{dobr}$ e $\mathcal{E}^{excen}$ na forma da superfície. Quando a energia $\mathcal{E}^{dobr}$ prevalece, a abertura do toro tende a diminuir; e quando $\mathcal{E}^{excen}$ prevalece, toda a superfície tende a encolher na direção do seu centro.



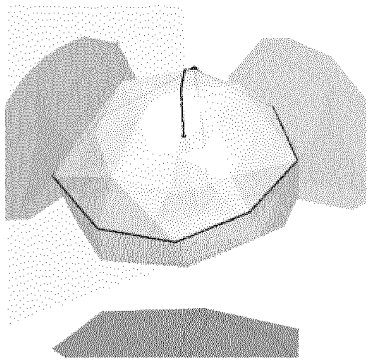
$0.33\mathcal{E}^{dobr} + 0.33\mathcal{E}^{area} + 0.33\mathcal{E}^{excen}$



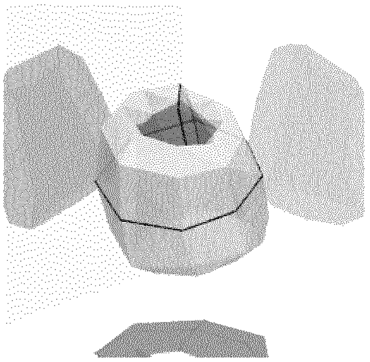
$0.83\mathcal{E}^{dobr} + 0.08\mathcal{E}^{area} + 0.08\mathcal{E}^{excen}$



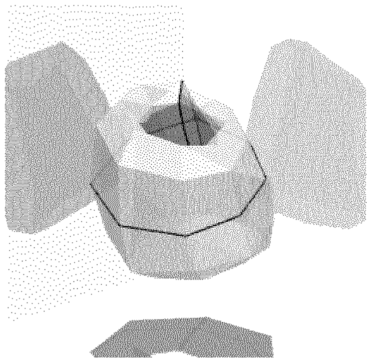
$0.08\mathcal{E}^{dobr} + 0.08\mathcal{E}^{area} + 0.83\mathcal{E}^{excen}$



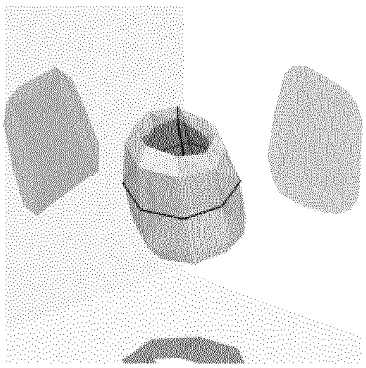
$0.08\mathcal{E}^{dobr} + 0.83\mathcal{E}^{area} + 0.08\mathcal{E}^{excen}$



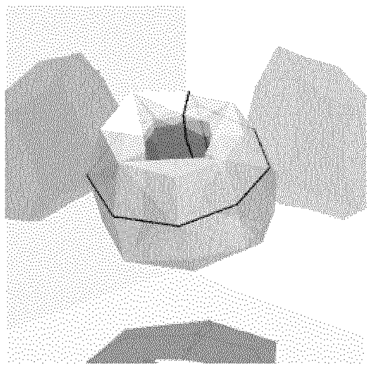
$0.33\mathcal{E}^{area} + 0.33\mathcal{E}^{ang} + 0.33\mathcal{E}^{excen}$



$0.83\mathcal{E}^{area} + 0.08\mathcal{E}^{ang} + 0.08\mathcal{E}^{excen}$



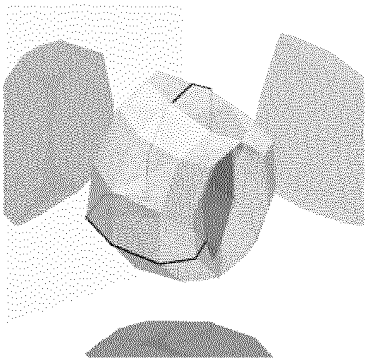
$0.08\mathcal{E}^{area} + 0.08\mathcal{E}^{ang} + 0.83\mathcal{E}^{excen}$



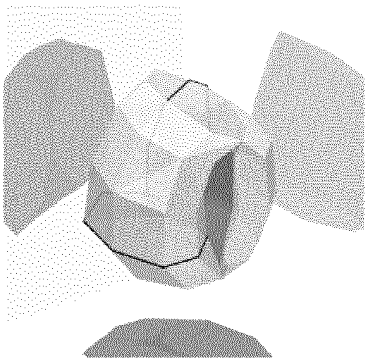
$0.08\mathcal{E}^{area} + 0.83\mathcal{E}^{ang} + 0.08\mathcal{E}^{excen}$

6.8.5 Combinação $\mathcal{E}^{area} + \mathcal{E}^{excen} + \mathcal{E}^{prox}$

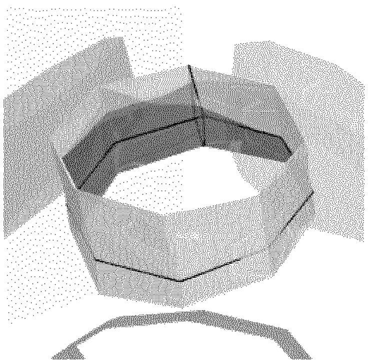
Note que nenhuma das energias presentes nessa combinação penaliza trançamentos. Outro ponto a ser observado é que a energia $\mathcal{E}^{prox}$ não conseguiu evitar um toro com espessura próxima a zero, ou seja, um cilindro de parede dupla.



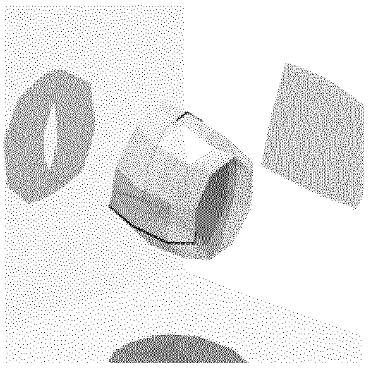
$0.33\mathcal{E}^{area} + 0.33\mathcal{E}^{excen} + 0.33\mathcal{E}^{prox}$



$0.83\mathcal{E}^{area} + 0.08\mathcal{E}^{excen} + 0.08\mathcal{E}^{prox}$



$0.08\mathcal{E}^{area} + 0.08\mathcal{E}^{excen} + 0.83\mathcal{E}^{prox}$



$0.08\mathcal{E}^{area} + 0.83\mathcal{E}^{excen} + 0.08\mathcal{E}^{prox}$

Capítulo 7

Conclusões

7.1 Os problemas tratados

Em nossa dissertação, tivemos que resolver problemas tais como: que modelagem adotar e como dar geometria as nossas superfícies. Escolhemos uma modelagem por triângulos que se mostrou simples e rápida de ser implementada, evitando os problemas advindos de uma modelagem baseada em retalhos polinomiais. Quanto ao problema de dar geometria à superfície, optamos por criar uma gama de funções de penalidade, *energias*, baseadas em critérios de beleza estipulados por nós e pela literatura corrente. A solução deste problema nos trouxe três novos problemas: que método de otimização usar, e qual combinação de energia usar e, que pesos cada energia deve ter.

Nossas funções energia mostraram possuir muitos mínimos locais. No estágio atual em que se encontram os métodos de otimização, não há um método que nos garanta a obtenção do ótimo procurado para este tipo de função. Isso nos obrigou a fazer muitos testes com vários métodos de otimização, além de adotarmos algumas heurísticas em conjunto com esses métodos. A escolha das energias também nos tomou um grande tempo já que foi necessário estudar os efeitos de cada uma das energias e de suas combinações sobre a forma final da superfície.

Cada um desses problemas citados por nós precisam ser melhor estudados e podem gerar, cada um, um trabalho específico sobre o assunto. A otimização de funções multimodais do $\mathbb{R}^n$ por si só já descreve um campo de estudo onde muitos trabalhos já foram publicados. Da mesma forma, a modelagem das superfícies e a escolhas das energias descrevem outros dois campos de estudo onde podemos encontrar muitos trabalhos específicos. Um dos grandes méritos de nosso trabalho talvez esteja em reunir tais problemas em um só trabalho, apresentando algumas soluções.

7.2 Os resultados

Os resultados obtidos por nós indicam que a visualização automática da topologia de complexos celulares, segundo a metodologia descrita nesta dissertação é perfeitamente factível e útil, pelo menos para complexos com algumas dezenas de arestas.

Nossos testes indicam também que o método de otimização `Grad`, baseado no princípio de descida pelo gradiente, é de modo geral muito mais eficiente do que os algoritmos `Simplex` e `Praxis`, que não usam o gradiente. Entretanto, este resultado só se aplica a configurações relativamente suavizadas e próximas do mínima. No início da otimização, quando a configuração ainda está aleatória, os métodos que se mostraram mais eficientes foram uma combinação de técnicas heurísticas e o método `Coord` de minimização coordenada-a-coordenada.

Há vários outros métodos de otimização — recozimento simulado, algoritmos genéticos, etc. — que poderiam ser aplicados ao nosso problema. Testes limitados feitos por nós indicam que apenas a implementação de um método como o do recozimento simulado, sem o auxílio de uma heurística, não proporciona resultados satisfatórios.

7.3 Sugestões e trabalhos futuros

No que diz respeito as energias, identificamos a necessidade de estudar com mais detalhes funções que penalizem trançamentos e pontos de curvatura infinita. A energia elétrica, da forma que está, ainda apresenta alguns problemas: além de barreiras de energia intransponíveis, a distribuição discreta de cargas que usamos não proporcionou os efeitos que gostaríamos de obter.

Podemos sugerir como melhorias ao nosso trabalho o estudo de novos métodos de otimização, um estudo mais aprofundado sobre as características de cada energia, o desenvolvimento de uma energia que penalize mais fortemente trançamentos e pontos de curvatura infinita e uma energia de perímetro de quartilho, que tende a manter o perímetro de uma quartilho próximo a um valor constante. Uma interface gráfica interativa, para o uso das ferramentas que nós desenvolvemos, traria maior comodidade para o usuário.

Apêndice A

Conceitos elementares de topologia

Esta seção tem como finalidade apresentar os conceitos básicos de topologia. Não é nossa pretensão nos aprofundarmos nestes conceitos. Para maiores detalhes, recomendamos consultar os livros textos padrões de topologia[2, 34, 24, 36].

A.1 Espaço topológico

Um *espaço topológico* é um par $(X, \mathcal{T})$ onde X é um conjunto não vazio e $\mathcal{T}$ é uma família de subconjuntos de X satisfazendo as seguintes regras:

- $\emptyset \in \mathcal{T}$;
- $X \in \mathcal{T}$;
- se $\mathcal{A} \subseteq \mathcal{T}$ então $\cup\mathcal{A} \in \mathcal{T}$;
- se $\mathcal{A} \subseteq \mathcal{T}$ e $\mathcal{A}$ é finito então $\cap\mathcal{A} \in \mathcal{T}$;

Os membros de $\mathcal{T}$ são chamados *conjuntos abertos* do espaço $(X, \mathcal{T})$, e a família $\mathcal{T}$ é a *topologia* do mesmo. Se $U \in \mathcal{T}$ então seu complemento $X \setminus U$ é um *conjunto fechado* do espaço.

No resto deste apêndice, vamos supor que $(X, \mathcal{T})$ é um espaço topológico.

A.2 Topologia da reta real

O exemplo mais importante de espaço topológico é $(\mathbb{R}, \mathcal{T}_{\mathbb{R}})$, onde $\mathbb{R}$ é o conjunto dos números reais, e um conjunto $U \subseteq \mathbb{R}$ pertence à família $\mathcal{T}_{\mathbb{R}}$ se e somente se todo ponto de U pertence a algum intervalo aberto de $\mathbb{R}$ que está totalmente contido em U. A família $\mathcal{T}_{\mathbb{R}}$ é a *topologia padrão* de $\mathbb{R}$.

A.3 Espaço produto

Se $(X_1, \mathcal{T}_1)$ e $(X_2, \mathcal{T}_2)$ são espaços topológicos, definimos o *espaço produto* $(X_1, \mathcal{T}_1) \times (X_2, \mathcal{T}_2)$ como sendo o par $(X_1 \times X_2, \mathcal{T})$, sendo que um subconjunto $U \subseteq X_1 \times X_2$ pertence a $\mathcal{T}$ se e somente para todo ponto p de U existem $U_1 \in \mathcal{T}_1$ e $U_2 \in \mathcal{T}_2$ tais que $p \in U_1 \times U_2 \subseteq U$.

O produto de espaços topológicos é associativo, e portanto esta definição pode ser estendida a produtos múltiplos

$$(X_1, \mathcal{T}_1) \times (X_2, \mathcal{T}_2) \times \cdots \times (X_n, \mathcal{T}_n)$$

e potências cartesianas $(X, \mathcal{T})^n$. Desta forma obtemos, por exemplo, a *topologia padrão* do espaço cartesiano $\mathbb{R}^n$, para todo $n > 1$.

A.4 Sub-espaços

Seja Y um subconjunto não vazio de X, e seja $\mathcal{T}(Y)$ a família $\{\, U \cap Y \ : \ U \in \mathcal{T} \,\}$. Prova-se que $(Y, \mathcal{T}(Y))$ também é um espaço topológico; ele é o *sub-espaço de* $(X, \mathcal{T})$ *induzido por* Y. A família $\mathcal{T}(Y)$ é a *topologia de* Y *relativa a* (ou *induzida por*) $(X, \mathcal{T})$.

O conceito de sub-espaços permite-nos construir facilmente muitos espaços topológicos interessantes: qualquer subconjunto do $\mathbb{R}^n$—um segmento, uma esfera, uma curva fechada, um disco com buracos—é automaticamente espaço topológico, com a topologia induzida pela topologia padrão do $\mathbb{R}^n$.

A.5 Espaço quociente

Se $\equiv$ é uma relação de equivalência sobre os pontos de um espaço $(X, \mathcal{T})$, definimos o *espaço quociente* $(X, \mathcal{T})/\equiv$ como sendo $(X', \mathcal{T}')$, onde X' é o conjunto $X/\equiv$ das classes de equivalência de X por $\equiv$; e $\mathcal{T}'$ é a família de todos os conjuntos $A/\equiv$, para todo $A \in \mathcal{T}$ que é união de classes de $X/\equiv$.

O conceito de espaço quociente foi usado implicitamente na seção 3.2.2, quando definimos o complexo representado por um modelo de colagem de polígonos. Nesse caso, o espaço $(X, \mathcal{T})$ original é a união dos polígonos isolados, um para cada face. A relação de equivalência define a maneira como as beiradas dessas faces devem ser coladas. Isto é, se o ponto x na fronteira de um polígono deve ser colado com o ponto y (na fronteira de outro polígono, ou do mesmo), então por definição $x \equiv y$. Por outro lado, se x pertence ao interior de um polígono, $x \equiv y$ se e somente se $x = y$. O espaço quociente $(X, \mathcal{T})/\equiv$ é a superfície do complexo representado.

Outro exemplo clássico de espaço quociente é o plano projetivo não orientado $\mathbb{P}^2$, que pode ser definido como o quociente da esfera unitária $\mathbb{S}^2$ de $\mathbb{R}^3$ (com a topologia natural) pela relação de equivalência que identifica cada ponto de $\mathbb{S}^2$ com seu antípoda.

A.6 Separabilidade

Dois pontos de um espaço topológico são *separáveis* se existe um conjunto aberto que contém um mas não o outro; caso contrário eles são *inseparáveis*. Um espaço é *separável* se todos os pares de pontos são separáveis.

Do ponto de vista topológico, dois pontos inseparáveis são essencialmente indistinguíveis. Não estaremos perdendo muita coisa interessante se restringirmos nossa atenção a espaços separáveis; pois o quociente de qualquer espaço por sua relação de inseparabilidade é um espaço separável, cujos abertos correspondem aos abertos do espaço original de maneira biunívoca e compatível com $\subseteq$.

A.7 Vizinhança, fecho e fronteira

Se p é um ponto de X, e V é um subconjunto de X tal que $p \in U \subseteq V$ para algum conjunto aberto U de $\mathcal{T}$, então dizemos que V é uma *vizinhança de* p na topologia $\mathcal{T}$.

Mais geralmente, se A é um subconjunto de X, dizemos que V é uma vizinhança de A em $\mathcal{T}$ se V for uma vizinhança de todo ponto p de A.

Se $A \subseteq X$, dizemos que um ponto $p \in X$ pertence ao *interior a* A na topologia $\mathcal{T}$ se e somente se A é uma vizinhança de p relativa a $\mathcal{T}$. Por outro lado, dizemos que p pertence ao *fecho* de A relativo a $\mathcal{T}$ se toda vizinhança de p encontra A.

Se denotarmos por A^- o interior de A, e A^+ seu fecho (numa determinada topologia $\mathcal{T}$), podemos provar que

$$X \setminus (X \setminus A)^+ = A^- \subseteq A \subseteq A^+ = X \setminus (X \setminus A)^-$$

O conjunto $A^+ \setminus A^-$ é por definição a *fronteira de* A na topologia $\mathcal{T}$; ele contém todo ponto $x \in X \setminus A$ tal que qualquer vizinhança de x encontra A.

A.8 Ponto de acumulação e limite

Seja A um subconjunto de X, e p um ponto de X. Dizemos que p é um *ponto de acumulação* (ou *ponto limite*) *de* A *em* $\mathcal{T}$ se e somente se qualquer vizinhança de p contem um número infinito de pontos de A.

Dizemos que um conjunto $A \subseteq X$ *incide* num conjunto $B \subseteq X$ se B possui um ponto de acumulação de A.

A.9 Continuidade e homeomorfismo

Uma função f de um espaço topológico $(X, \mathcal{T})$ para um espaço topológico $(X', \mathcal{T}')$, é dita *contínua* num ponto $x \in X$ se e somente se para cada cada vizinhança V de $f(x)$ em $\mathcal{T}'$, existe uma vizinhança U de x em $\mathcal{T}$ tal que $f(U) \subseteq V$. A função f é *contínua*, sem mais, se ela é contínua em todos os pontos de X.

Um *homeomorfismo* entre dois espaços $(X, \mathcal{T})$ e $(X', \mathcal{T}')$ é uma função bijetora contínua f de $(X, \mathcal{T})$ para $(X', \mathcal{T}')$, cuja inversa f^{-1} também é contínua. Diz-se então que os dois espaços são *homeomórficos* ou *topologicamente equivalentes*.

Uma *propriedade topológica* é uma propriedade de pontos e subconjuntos de um espaço $(X, \mathcal{T})$ que pode ser ser definida apenas em termos dos conjuntos abertos de $\mathcal{T}$ (e das operações da teoria de conjuntos). Um exemplo é o fato de um conjunto de pontos ser fechado: que significa, por definição, que o complemento dele é aberto. Outro exemplo é o fato de um espaço X ser *conexo*, que significa que não existe nenhum subconjunto próprio e não-vazio de X que é ao mesmo tempo fechado e aberto. Ainda outro exemplo é fato de um conjunto de pontos Z num espaço X *separar* dois pontos $u, v \in X \setminus Z$, significando que não existe nenhum sub-espaço de $X \setminus Z$ que seja conexo e contenha u e v.

Obviamente, um homeomorfismo f preserva todas as propriedades topológicas de pontos e subconjuntos de pontos. Em particular, se um espaço é conexo, qualquer espaço homeomorfo a ele também será conexo; se Z separa u e v, então $f(Z)$ separa $f(u)$ de $f(v)$. E assim por diante.

Bibliografia

[1] F. S. Acton. *Numerical Methods that work.* New York, Harper and Row, 1990.

[2] A. V. Arkhangel'skiĭ and L. S. Pontryagin. *General Topology I*, volume 17 of *Encyclopaedia of Mathematical Sciences.* Springer-Verlag, 1990.

[3] C. L. Bajaj. Smoothing polyhedra using implicit algebraic splines. In *SIGGRAPH'92 Conference Proceedings*, volume 26, pages 79–88, 1992.

[4] G. Di Battista, P. D. Eades, and R. Tamassia. Algorithms for drawing graphs: An annotated bibliography. Technical report, Departament of Computer Science, University of Newcastle, 1993.

[5] B. G. Baumgart. A polyhedron representation for computer vision. In *Proceedings of 1975 AFIPS National Computer Conference*, volume 44, pages 589–596, 1975.

[6] W. Baur and V. Strassen. The complexity of partial derivatives. *Theoretical Computer Science*, 22:317–330, 1983.

[7] K. A. Brakke. The surface evolver. *Experimental Mathematics*, 1(2):141–165, 1992.

[8] K. A. Brakke. *Surface Evolver Manual.* The Geometry Center, The Geometry Center, 1300 South Second Street, Minneapolis, MN 55454 and Susquehanna University, Selinsgrove, PA 17870, December 1993. Eletronic Address: brakke@geom.umn.edu.

[9] R. P. Brent. *Algorithms for Minimization without Derivatives.* Englewood Cliffs,N.J.: Prentice-Hall Inc., 1973.

[10] E. Brisson. Representing geometric structures in d dimensions: Topology and order. In *Proc.* 5^{th} *Annual ACM Symp. on Computational Geometry*, pages 218–227, 1989.

[11] R. Davidson and D. Harel. Drawing graphs nicely using simulated annealing. Technical report, Department of Applied Mathematics and Computer Science, The Weizmann Institute of Science, 1989.

[12] P. J. de Rezende and J. Stolfi. *Fundamentos de Geometria Computacional.* IX Escola de Computação - UFPE, 1994.

[13] W. L. F. Degen. Explicit continuity conditions for adjacent Bézier surface patches. *Computer Aided Geometric Design*, pages 181–189, 1990.

[14] T. D. DeRose. Necessary and sufficient conditions for tangent plane continuity of Bézier surfaces. *Computer Aided Geometric Design*, pages 165–179, 1990.

[15] D. P. Dobkin and M. J. Laszlo. Primitives for the manipulations of three-dimensional subdivisions. In *Proc. 3rd ACM Symp. on Computational Geometry*, pages 86–99, 1987.

[16] L. J. Guibas and J. Stolfi. Primitives for the manipulations of general subdivisions and the computation of Voronoi diagrams. *ACM Transactions on Graphics*, 4(2):74–123, april 1985.

[17] B. K. Horn. The curve of least energy. *ACM Transactions on Mathematical Software*, 9(4):441–460, December 1983.

[18] L. Hsu, R. Kusner, and J. Sullivan. Minimizing the squared mean curvature integral for surfaces in space forms. *Experimental Mathematics*, 1(3):191–207, 1992.

[19] D. A. H. Jacobs. *The State of the Art in Numerical Analysis.* Academic Press, 1977.

[20] M. Kallay and B. Ravani. Optimal twist vectors as a tool for interpolating a network of curves with a minimum energy surface. *Computer Aided Geometric Design*, pages 465–473, 1990.

[21] T. Kamada and S. Kawai. An algorithm for drawing general undirected graphs. *Information Processing Letters*, pages 7–15, April 1989.

[22] G. Kant. A more compact visibility representation. In *Graph Drawing '93 — ALCOM International Workshop on Graph Drawing and Topological Graph*, September 1993.

[23] P. Lienhardt. Subdivisions of n-dimensional spaces and n-dimensional generalized maps. In *Proc. 5^{th} Annual ACM Symp. on Computational Geometry*, pages 228–236, 1989.

[24] E. L. Lima. *Curso de Análise*, volume 2. Instituto de Matemática Pura e Aplicada, 1981.

[25] D. Lischinski. Incremental Delaunay triangulation. In P. S. Heckbert, editor, *Graphics Gems IV.* Academic Press, 1994.

[26] C. Loop and T. DeRose. Generalized B-spline surfaces of arbitrary topology. In *Proceedings of SIGGRAPH 90*, volume 24, pages 347–356, 1990.

[27] J. Meidanis and J. C. Setubal. *Uma Introdução à Biologia Computacional*. IX Escola de Computação, 1994.

[28] C. F. X. Mendonça and P. Eades. Learning aesthetics for visualization. In *Anais do XX Seminário Integrado de Software e Hardware*, pages 76–88, 1993.

[29] H. P. Moreton. Minimum curvature variation curves, networks, and surfaces for fair free-form shape design. Technical report, Computer Science Division, University of California, Berkeley, March 1993.

[30] H. P. Moreton and C. H. Séquin. Functional optimization for fair surface design. In *Proceedings of SIGGRAPH 92*, volume 26, pages 167–176, 1992.

[31] D. B. Parkinson and D. N. Moreton. Optimal biarc-curve fitting. *Computer-Aided Design*, 23(6):411–419, 1991.

[32] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T Vetterling. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, 1986.

[33] R. Unger and J. Moult. Finding the lowest free energy conformation of a protein is an np-hard problem: Proof and implications. *Bulletin of Mathematical Biology*, 55(6):1183–1198, 1993.

[34] L. E. Ward, Jr. *Topology: An Outline for a First Course*. Marcel Dekker, Inc., New York, 1972.

[35] F. Yamaguchi and T. Tokieda. A solid modelling system: Freedom-ii. *Computers & Graphics*, pages 225–232, 1983.

[36] W. H. Young and G. C. Young. *The Theory of Sets Of Points*. Chelsea, New York, 1972.