Visualização Automática de Complexos Celulares Arbitrários

Rober Marcone Rosi   Jorge Stolfi

DCC - IMECC - UNICAMP
Caixa Postal 606 - CEP 13081-970
Campinas SP – Brasil
{marcone,stolfi}@dcc.unicamp.br

1 de junho de 1994

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). Descrevemos aqui um programa que, dada apenas a estrutura topológica de um complexo celular (ou seja, as relacões 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). We describe here 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.

1 Introdução

1.1 Motivação

Representação de fronteira é uma técnica frequentemente usada em computação gráfica para modelar objetos tridimensionais, 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 modelo.

Na implementação de tais modelos, a experiência aconselha separar os aspectos *topológicos* (que dizem respeito aos contatos entre os vértices, arestas, e faces do objeto) e os aspectos *geométricos* (que incluem as coordenadas dos vértices, a forma das arestas e das faces, etc.). Esta separação facilita o desenvolvimento e depuração de algoritmos, e a modelagem dos objetos [18].

Um *complexo celular* (*bidimensional*) é uma estrutura matemática que captura os aspectos topológicos da superfície de um objeto tridimensional, ignorando toda informação geométrica. A estrutura consiste de um conjunto de elementos abstratos, representando as faces, arestas e vértices do objeto, e das relações de incidência e adjacência entre os mesmos.

1.2 Visualização da topologia

A topologia de um complexo pode ser bastante difícil de visualizar, mesmo quando o número de arestas e faces é pequeno. Por exemplo, a garrafa de Klein pode ser representada por um complexo celular com apenas uma face e duas arestas (veja figura 1). Topologias mais complexas, com um número maior de faces e arestas, são praticamente impossíveis de visualizar sem algum auxilio computacional.

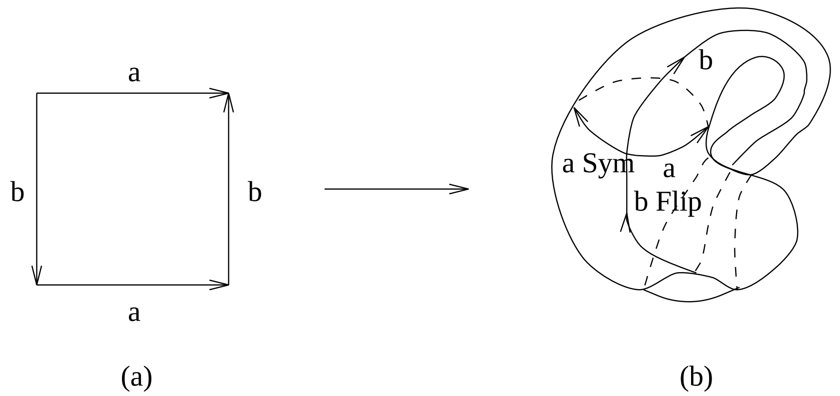


Figura 1: (a) A topologia da garrafa de Klein. (b) A representação desejada.

Esta dificuldade nos motivou a desenvolver uma ferramenta gráfica para a visualização automática da topologia de complexos celulares arbitrários. A entrada do programa é uma estrutura de dados que representa a topologia de um complexo. A saída é uma representação geométrica “bonita” do mesmo. Esta representação consiste de uma superfície de forma específica, mergulhada no espaço R^3, e de uma divisão da mesma em faces, vértices, e arestas, correspondentes aos elementos do complexo, com as mesmas relações de adjacência.

1.3 O que é uma representação bonita?

Para resolver este problema automáticamente, precisamos primeiro definir melhor o conceito de “representação bonita.” Informalmente, queremos uma superfície que, em primeiro lugar, permita a visualização fácil da topologia do complexo celular; e, em segundo lugar, seja estéticamente agradável.

Note-se a diferença entre a topologia do *complexo* e a topologia da *superfície*. 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 distinguíveis, 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 2(a) fica mais fácil de visualizar numa superfície alongada, como em (c), do que na esfera (b).

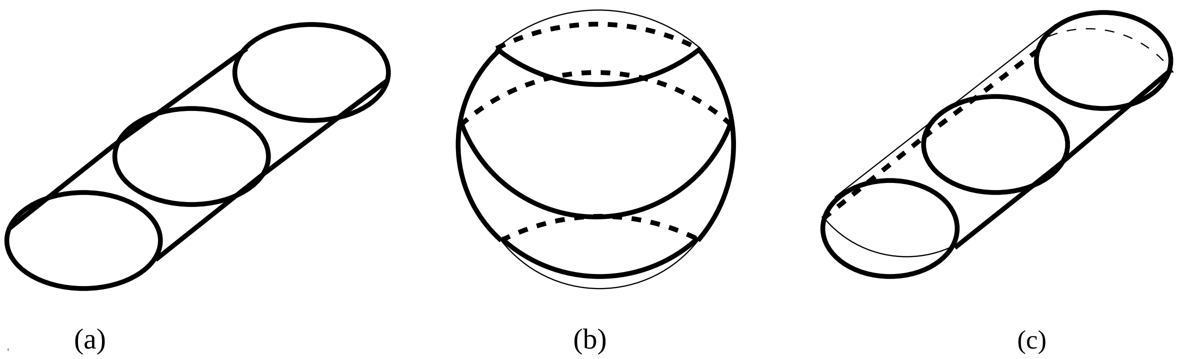


Figura 2: Um complexo celular (a), e duas representações (b) e (c).

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.

A fim de quantificar estes critérios, definimos uma função numérica da geometria da superfície, que chamamos 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 do complexo dado, aquela que tem energia mínima.

1.4 Trabalhos relacionados

O problema de desenho "bonito" de grafos no plano tem recebido grande atenção devido ao grande número de aplicações: projeto de VLSI, bancos de dados, algoritimos de animação, linguagens visuais, gerência de redes, ferramentas para desenvolvimento de software, etc. [12]. Apesar de nosso trabalho não ser o mesmo que o de desenho de grafos em 2 e 3 dimensões, nossa solução usa técnicas semelhantes às já usadas com sucesso nesta área; e, por outro lado, acreditamos que algumas das nossas idéias podem vir a ser úteis também para a visualização de tais grafos.

Nosso trabalho tem grandes semelhanças com certas técnicas de relaxação usadas em projeto industrial, para o amaciamento de juntas e cantos ("fairing" e "fileting") em modelos de sólidos [16]. 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 nossa, não têm que lidar com topologias muito complicadas, e partem de uma solução inicial bastante próxima à desejada.

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 [20, 15]. A relação vem do fato da nossa energia "estética" ser qualitativamente semelhante à energia fisico-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.

2 Modelo de visualização

2.1 Representação da topologia

Encontram-se na literatura muitas estruturas de dados destinadas a representar a topologia de complexos celulares [21, 5, 18, 7]. No nosso trabalho, resolvemos adotar a estrutura de dados *quad-edge* [13], que é similar às estruturas *winged edge* e *half-edge* [2], amplamente usadas em CAD e computação gráfica.

A principal vantagem da estrutura quad-edge, em relação às demais, é que ela permite a representação de complexos celulares não-orientáveis, como a garrafa de Klein ou o plano projetivo.

A estrutura quad-edge permite também representar complexos celulares com vértices de grau 1, arestas múltiplas unindo o mesmo par de vértices, arestas com origem e destino no mesmo vértice, e faces, múltiplas vezes incidentes ao mesmo vértice. Esta generalidade exige cuidados especiais na modelagem da superfície, como veremos mais adiante.

Por outro lado, a estrutura quad-edge exige que toda face seja topologicamente equivalente a um disco, sem buracos. Esta propriedade simplifica bastante o manuseio da estrutura e a modelagem da superfície.

2.2 Ladrilhos

Uma maneira óbvia de modelar a geometria da superfície seria modelar cada face do complexo por um retalho polinomial, implícito ou explícito, com restrições de continuidade entre retalhos adjacentes [14, 1]. 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.

Portanto, em vez de considerar a superfície como uma união de faces, nós a consideramos como uma união de *ladrilhos*, onde cada ladrilho é o quadrilátero definido pelos dois extremos de uma aresta e, e pelos extremos da aresta dual de e. Veja a figura 3.

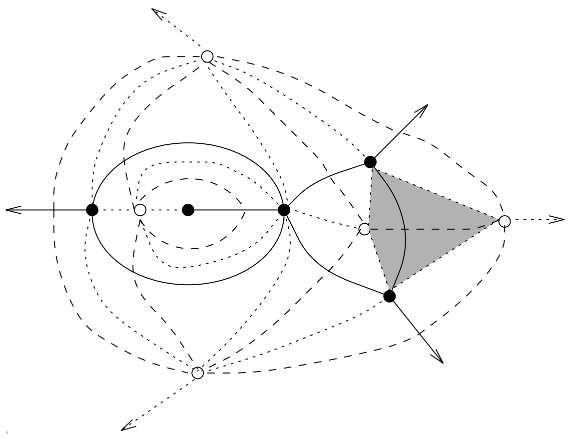


Figura 3: Um complexo celular bidimensional, mostrando arestas primais (linhas cheias), arestas duais (linhas tracejadas), e a subdivisão em ladrilhos (linhas pontilhadas).

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.

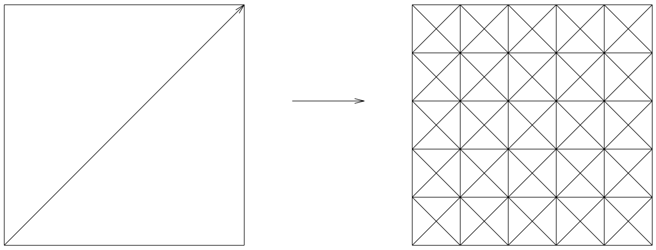
O problema reduz-se então à atribuir geometria a cada ladrilho, garantido continuidade, suavidade e não interferência.

2.3 Modelagem dos ladrilhos

Um possível modelo para cada ladrilho seria um retalho polinomial paramétrico [14, 16]. O problema desta solução é que teríamos que garantir a continuidade do plano tangente, e talvez da curvatura, ao longo das junções entre os retalhos. Ao mesmo tempo, precisamos evitar retalhos degenerados, cujas derivadas parciais se anulam. Isso é difícil de garantir no nosso caso, pois retalhos podem ser colados consigo mesmos.

Além disso, teriamos que usar fórmulas integrais complicadas para calcular a energia de cada retalho.

Em vista destas dificuldades, decidimos 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 4.



(a) (b)

Figura 4: O ladrilho de uma aresta e e a malha 5×5 correspondente.

Desta forma, substituímos o complexo celular original, com m arestas, por um complexo mais refinado, com $6mK^2$ arestas e $4mK^2$ faces triangulares. Portanto, desistimos de obter uma superfície com continuidade do plano tangente; em vez disso, nos contentamos com minimizar os ângulos entre triângulos vizinhos.

Na maioria dos nossos testes, adotamos $K = 5$. Constatamos que esse valor já permite obter superfícies bastante suaves, mesmo no caso de topologias complicadas.

2.4 Construindo a triangulação

Portanto, a primeira etapa do nosso algoritmo é construir uma grade triangulada $K \times K$ para cada aresta do complexo dado, como na figura 4. A seguir estes retalhos são colados entre si pelas bordas, conforme exigido pelas relações de incidência e adjacencia das respectivas arestas do complexo.

A colagem de retalhos é um tanto delicada, pois é preciso levar em conta que um retalho pode ser colado consigo mesmo, talvez até formando uma superfície não orientável. Via de regra, em tais casos o complexo que resulta da colagem pode possuir triângulos degenerados.

A figura 5 ilustra este problema, no caso da colagem de um retalho de grid $2{\times}2$ a si próprio (a). A figura (b) mostra o processo de colagem da borda inferior com a borda direita; a figura (c) mostra o resultado da colagem. Note em (c) que o complexo resultante tem dois triângulos degenerados — distintos, porém com os mesmos três vértices.

Tais triângulos são indesejáveis, pois eles são sempre coincidentes, quaisquer que sejam as coordenadas atribuídas aos vértices. Eles geralmente introduzem descontinuidades aparentes na superfície de energia mínima. Portanto, após cada colagem, precisamos localizar e eliminar quaisquer triângulos que tenham se tornado degenerados.

A figura (d) ilustra o resultado da eliminação dos triângulos degenerados da figura (c). Note que, em conseqüência desta eliminação, dois outros triângulos ficaram degenerados (em cinza), e também precisarão ser eliminados. Felizmente, pode-se provar que, se $K \geq 5$, este processo sempre termina, sem eliminar totalmente nenhum ladrilho.

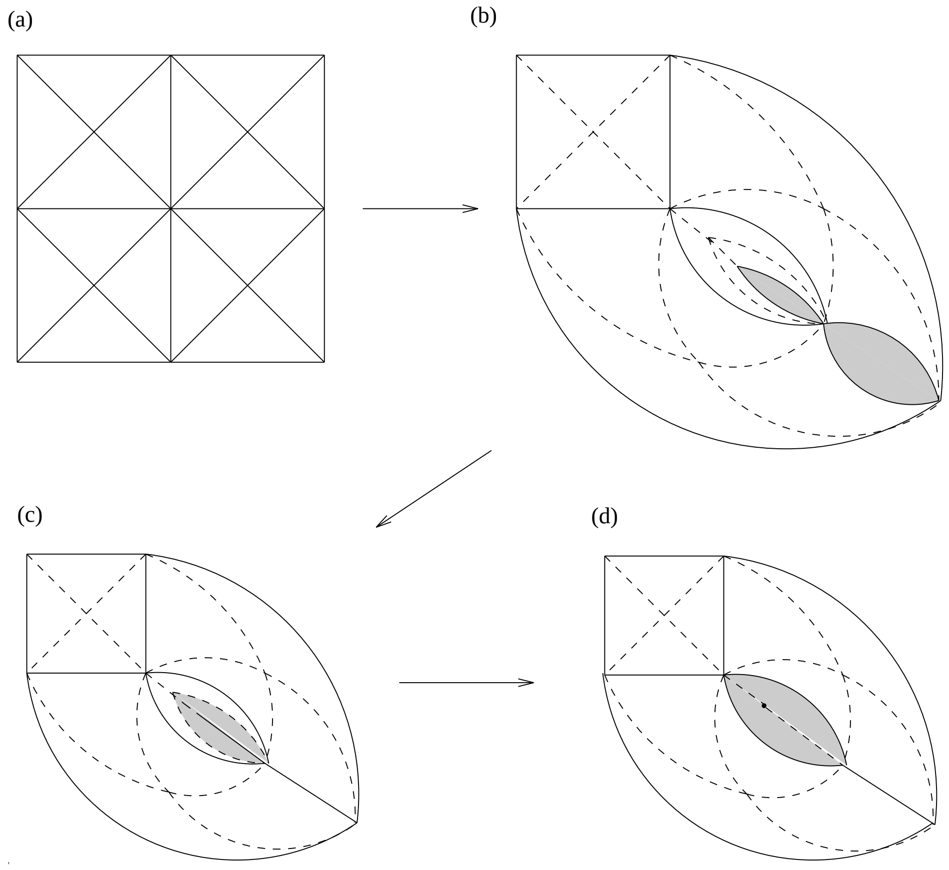


Figura 5: Colagem de um retalho a si mesmo, e eliminação de triângulos degenerados

3 A função energia

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 vértices $v_1, v_2, \ldots, v_n$ dos triângulos. Definimos, portanto, uma *configuração* da superfície como sendo uma tupla de n pontos $x_1, x_2, \ldots x_n$ do $\mathbf{R}^3$.

Na maioria dos nossos testes, definimos a energia de uma configuração como sendo uma combinação linear de duas parcelas:

- Uma *energia de dobramento*, calculada como se cada aresta fosse uma dobradiça, com uma mola que tende a manter a coplanaridade das duas faces adjacentes. Minimizar a energia de dobramento significa diminuir os ângulos entre triângulos adjacentes, e distribuir uniformemente a curvatura entre todas as arestas.

- uma *energia de proximidade*, calculada como se cada vértice fosse uma carga elétrica que repele os demais vértices. Minimizar a energia de proximidade significa afastar os vértices uns dos outros. Esta repulsão ajuda um pouco a desdobrar e alisar a superfície; mas sua principal função é evitar a superposição de partes independentes da superfície, e fazer com que as intersecções da superfície com si mesma sejam transversais em vez de rasantes.

Obviamente, quando estas energias são combinadas na energia total, elas podem ter efeitos opostos entre si.

3.1 A energia de dobramento

Definimos a energia de dobramento como sendo a soma dos quadrados dos ângulos diedrais externos de todas as arestas da triangulação. O ângulo diedral θ de uma aresta é calculado pela fórmula

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

onde $\vec{a}$ é o vetor definido pela aresta, e $\vec{b}$, $\vec{c}$ são os vetores das duas arestas vizinhas com mesma origem [4].

3.2 A energia de proximidade

A energia de proximidade é calculada como sendo a energia potencial eletrostática, supondo-se que cada vértice possui carga positiva 1.

Para evitar descontinuidades e picos infinitos na energia, que dificultam sua otimização, supomos que a carga de cada vértice está espalhada no espaço, em um raio comparável à distância média μ entre vértices adjacentes. Mais precisamente, a energia de proximidade é

$$EL = \sum_{1\leq i<j\leq n} \frac{1}{\sqrt{|x_i - x_j|^2 + \mu^2}}$$

3.3 Normalização

Se aplicadas sem nenhum critério, as funções energias que utilizamos podem atingir seu mínimo em configurações degeneradas pouco interessantes — por exemplo, com todos os pontos no infinito, ou todos concentrados num único ponto.

Podemos evitar estes problemas de duas maneiras: (1) normalizar; e (2) usar termos apropriados na função energia. Estes dois métodos correspondem à técnicas tradicionais de otimização, ou seja, restrições e funções de penalidades. No nosso caso, é essencial normalizar as coordenadas dos vértices de toda configuração, de modo a preservar a posição e tamanho da nuvem de pontos. Para tanto, subtraímos de cada vértice o baricentro (a média aritmética) de todos os vértices, e aplicamos em seguida um fator de escala calculado para deixar o comprimento quadrático médio igual a 1.

Esta normalização significa que a busca da configuração ótima é restrita a um sub-espaço do espaço $\mathbf{R}^{3n}$. Geométricamente, nós obteríamos o mesmo efeito da normalização, caso tivéssemos adicionado um terceiro termo à fórmula da energia — uma *energia de esticamento* — que tenderia a manter constante a distância média entre os pontos.

3.4 Trabalhos relacionados

A idéia de usar medidas de "feiura" análogas à energia potencial elétrica (proximidade) e elástica (esticamento) já foi usada por Kamada e outros para a visualização automática de grafos planares [11, 6]. A energia de curvatura (dobramento), geralmente definida como a integral do quadrado da curvatura, surge naturalmente no estudo de curvas e superfícies míminas [8, 10, 17, 3, 9, 16].

4 Otimização

Se a função energia fosse suficientemente suave, poderíamos usar a técnica proposta por Moreton [16] e outros: depois de obter uma forma inicial para a mesma usando alguma heurística, ajustar a mesma usando técnicas padrões de otimização não-linear baseadas no método do gradiente.

Infelizmente, nossa função energia tem uma estrutura bastante complexa, com uma grande quantidade de vales, descontinuidades, mínimos locais, separados por barreiras de alta energia.

A técnica de otimização que tem dado melhores resultados consiste em aplicar uma heurística de *amaciamento* local da superfície para gerar um repertório de configurações iniciais interessantes, seguida de um algoritmo clássico de otimização não-linear — o método do simplex de Nadler-Mead [19] — para explorar o espaço afim gerado por estas configurações iniciais.

Nossa implementação do método Nadler-Mead inclui algumas extensões triviais, como o uso de um simplexo sub-dimensional (que faz sentido no nosso caso, pois o espaço das configurações "amaciadas" tem dimensão muito menor que $3n$). Esta modificação, em particular, permite-nos trabalhar com triangulações contendo várias centenas de vértices — o que significa minimizar uma função não linear com milhares de parâmetros.

4.1 Heurística de amaciamento

A heurística de amaciamento 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, a nova posição do vértice v é restrita à reta $a + t\vec{n}$, onde a é o baricentro dos vértices vizinhos $u_0, u_1, \ldots, u_k$, e $\vec{n}$ é a normal aproximada da superfície no vértice v, também calculada a partir desses vizinhos. A posição t é escolhida de modo a minimizar (aproximadamente) o dobramento dos ângulos diedrais nas arestas $u_{i-1}u_i$.

O amaciamento 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 aplicarmos essa heurística para todos os vértices, repetidas vezes, a energia de dobramento espalha-se por difusão em toda a superfície. A convergência é relativamente lenta (por difusão), mas rápida o suficiente para produzir superfícies qualitativamente suaves.

5 Resultados

No estado atual da implementação, conseguimos em geral obter, após algumas centenas de iterações do algoritmo simplex, geometrias bastante aceitáveis para complexos celulares com duas ou três arestas (que dão origem a triangulações com algumas centenas de vértices).

As figuras 6, 7, e 8 mostram as superfícies calculadas pelo programa para um tetraedro (4 faces, 6 arestas, com ladrilhos de ordem 1×1), para um toro (1 face, 2 arestas, com ladrilhos de ordem 5×5), e para uma garrafa de Klein (idem). Em cada caso, a primeira imagem mostra a configuração inicial, com coordenadas aleatórias; a segunda imagem mostra a superfície obtida após 5 aplicações da heurística de amaciamento para todos os vértices; e a terceira imagem mostra o melhor resultado obtido pelo método simplex.

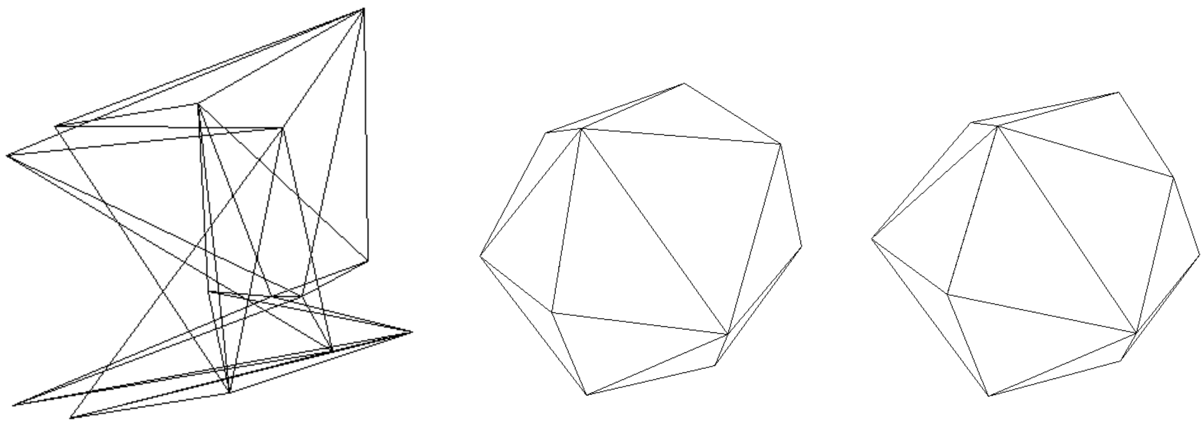


Figura 6: Tetraedro.

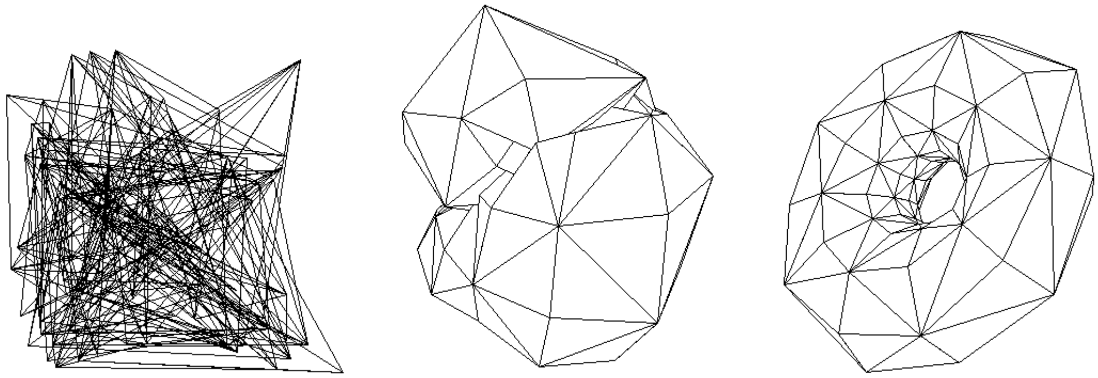


Figura 7: Toro.

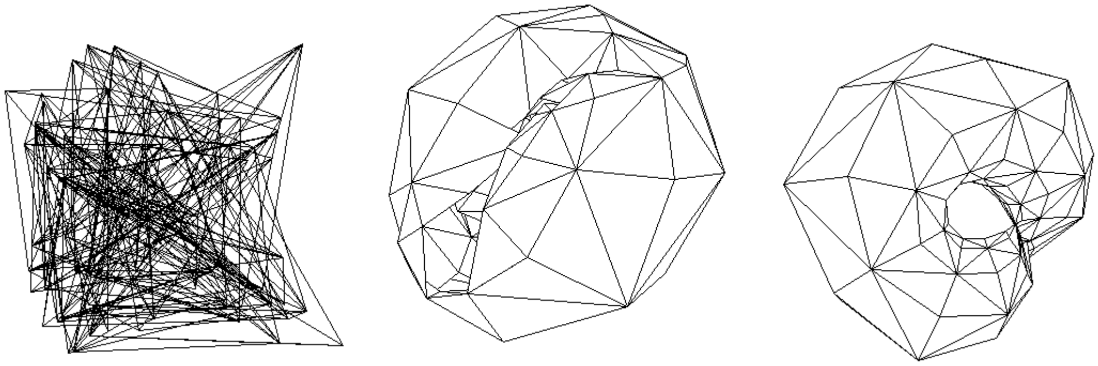


Figura 8: Garrafa de Klein.

Para complexos maiores, a convergência ainda é muito lenta, e as superfícies produzidas ainda apresentam vários problemas: auto-intersecções desnecessárias, triângulos de tamanhos não uniformes, superfícies com pregas de curvatura infinita, etc.

Por exemplo, a figura 9 mostra a superfície obtida para um complexo com a topologia de uma "salsicha" (18 arestas, 10 faces, usando ladrilhos de ordem 5×5), após várias horas de processamento. Note que a superficie resultante ainda tem algumas auto-intersecções, além de não ter a forma ótima (que deveria ser um longo cilindro com calotas nas duas extremidades).

Reduzindo o número de arestas para 10, e usando ladrilhos de ordem 3×3, obtivemos após um par de horas a superfície ilustrada na figura 10. Note que as auto-intersecções diminuíram, e a forma é mais próxima à esperada.

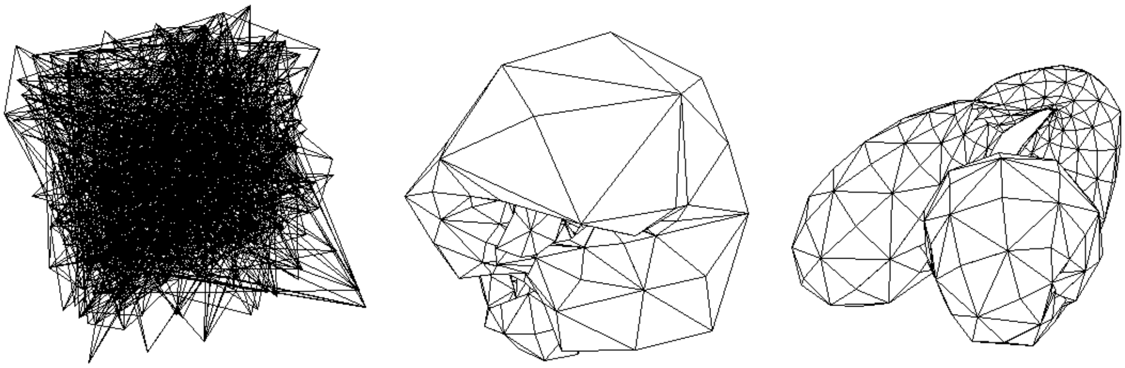


Figura 9: Salsicha 1.

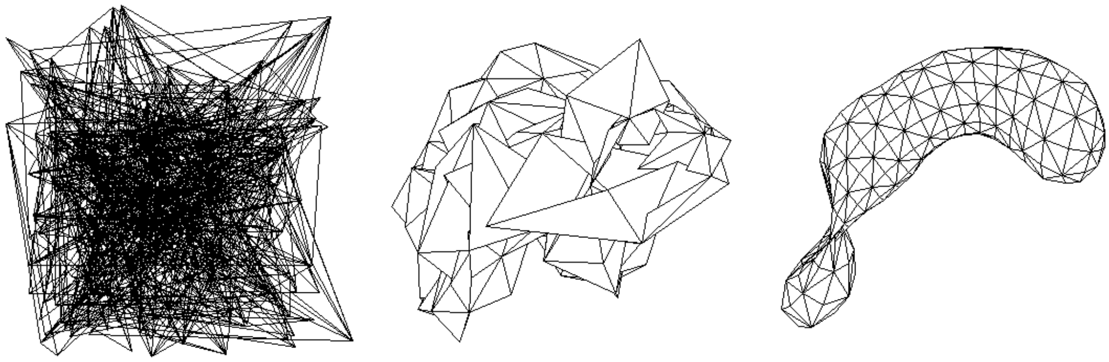


Figura 10: Salsicha 2.

6 Conclusões

Nossa experiência até o momento indica que a visualização automática da topologia de complexos celulares, segundo a metodologia descrita neste artigo é perfeitamente factível e útil, pelo menos para subdivisões com algumas dezenas de arestas.

Nossa implementação atual ainda está bastante lenta, mas já identificamos muitas melhorias óbvias que devem acelerar a convergência, pelo menos por uma ordem de grandeza. Em particular, há vários outros métodos de otimização — recozimento simulado, algoritmos genéticos, métodos de gradiente, etc. — que parecem ser mais adequados ao nosso problema do que o algoritmo de Nadler-Mead, mas que ainda não tivemos ocasião de implementar.

Referências

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

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

[3] Kenneth A. Brakke. The surface envolver. *Experimental Mathematics*, 1(2):141–165, 1992.

[4] Kenneth 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.

[5] David P. Dobkin and Michel J. Laszlo. Primitives for the Manipulations of Three-Dimensional Subdivisions. *Proc. 3rd ACM Symp. on Comp. Geometry*, pages 86–99, june 1987.

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

[7] Erik Brisson. Representing Geometric Structures in d Dimensions: Topology and Order. *Proc. 5^{th} Annual ACM Symp. on Computational Geometry*, pages 218–227, june 1989.

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

[9] Lucas Hsu, Rob Kusner, and John Sullivan. Minimizing the squared mean curvature integral for surfaces in space forms. *Experimental Mathematics*, 1(3):191–207, 1992.

[10] Michael Kallay and Bahram Ravani. Optimal twist vectors as a tool for interpolation a network of curves with a minimum energy surface. *Computer Aided Geometric Design*, pages 465–473, 1990.

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

[12] Goos Kant. A more compact visibility representation. In *Graph Drawing '93, ALCOM International Workshop PARIS - 1993 on Graph Drawing and Topological Graph*, September 1993.

[13] Leonidas Guibas and Jorge Stolfi. Primitives for the Manipulations of General Subdivisions and the Computation of Voronoi Diagrams. *ACM Transactions on Graphics*, 4(2):74–123, april 1985.

[14] Charles Loop and Tony DeRose. Generalized b-spline surface of arbitrary topology. In *Proceedings of SIGGRAPH 90. In Computer Graphics*, volume 24, pages 347–356, august 1990.

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

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

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

[18] Pascal Lienhardt. Subdivisions of N-Dimensional Spaces and N-Dimensional Generalized Maps. *Proc.* 5^{th} *Annual ACM Symp. on Computational Geometry*, pages 228–236, june 1989.

[19] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T Vetterling. *Numerical Recipes: The art of scientific computing.* Cambridge University Press, 1986.

[20] Ron Unger and John 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.

[21] Fujio Yamaguchi and Toshiya Tokieda. A solid modelling system: Freedom-ii. *Comput. & Graphics*, pages 225–232, 1983.