

Novos Resultados em Triangulações (Triangulação Gulosa)

LUIZ MARCOS GARCIA GONÇALVES

LCG - Laboratório de Computação Gráfica, COPPE - Sistemas / UFRJ
21945-970, Rio de Janeiro, RJ, Brasil, Caixa Postal 68511
lmarcos@lcg.ufrj.br, lmarcos@cs.umass.edu

Abstract: This paper describes a data structure with a restricted set of topological operations (creation, subdivision and flip) used to construct planar triangulations. We discuss the implementation of algorithms to build some common triangulations (greedy, Delaunay and adaptive rectangular) based on that structure. In order to allow the use of such a structure in the implementation of the Greedy Triangulation (GT), we define a new concept of GT and present an algorithm to build it. This GT algorithm has the same complexity as the best known algorithm for the traditional GT.

Keywords: Greedy Triangulation (GT), Delaunay triangulation (DT), Adaptive rectangular triangulation (RA), Triangulation data structure.

1 INTRODUÇÃO

Este trabalho apresenta uma estrutura com três operações topológicas (criação, subdivisão e flip) usada para construir uma triangulação no plano a partir de um conjunto irregular de pontos. Usando esta estrutura, realizamos a implementação de três triangulações comuns na literatura (Greedy, de Delaunay e retangular adaptativa). Para tornar possível (ou mais fácil) o uso da estrutura para a implementação da Triangulação Gulosa (GT), estabelecemos um critério guloso diferente dos usuais e apresentamos um novo conceito para a GT, segundo este critério. A complexidade deste algoritmo é igual à do melhor algoritmo conhecido para a triangulação gulosa tradicional. Os algoritmos implementados são apresentados e os resultados são discutidos nas últimas seções.

2 TRIANGULAÇÕES

Uma triangulação de um conjunto de pontos P é um conjunto máximo de segmentos de reta com extremos em P que não se interceptam mutuamente [01]. As triangulações estão entre as formas de decomposição do espaço mais utilizadas pelo fato dos triângulos serem estruturas lineares extremamente simples que facilitam muitas tarefas de exploração do espaço, além de serem muito usadas para representação de funções bivariadas.

As triangulações mais conhecidas são a de custo mínimo, a de Delaunay e a gulosa. A de custo mínimo (menor somatório de arestas) possui boas propriedades numéricas, porém, encontrar esta triangulação é um problema de complexidade ainda não resolvida (pode ser NP-completo). Entre os problemas recentemente abordados envolvendo triangulação, podemos citar a

busca de uma definição para a complexidade da triangulação de custo mínimo [04, 05, 06, 08, 09, 10, 11], métodos para medir a distância entre triangulações [03] e a busca de triangulações para determinadas aplicações [07, 12, 13, 14, 15, 18], restritas ou não segundo critérios definidos. A triangulação retangular adaptativa (originalmente introduzida como CFK adaptativa) apresenta interessantes propriedades de semi-regularidade e adaptação (ver [17] e [23]). Todas estas triangulações podem ser determinadas a partir de uma amostra de pontos do espaço ou a partir de uma outra triangulação genérica por flips [02, 03], subdivisões e remoções de triângulos.

2.1 Triangulações Retangulares

A triangulações J1 e K1 resultam de subdivisões de malhas quadradas regulares onde os quadrados da malha são subdivididos por uma de suas diagonais. Na K1, tomam-se as diagonais paralelas entre si e na J1 alternadas (como na bandeira do Reino Unido). Nessas triangulações regulares, todos os triângulos são retângulo-isósceles. As triangulações retangulares adaptativas (RA), generalizações dessas, são aquelas que envolvem apenas triângulos retângulo-isósceles (em qualquer arranjo e de quaisquer tamanhos).

Uma vasta família de triangulações RA (a que nos restringiremos aqui) engloba aquelas que podem ser obtidas a partir de triangulações J1 ou K1 por sucessivas operações básicas de subdivisão por refinamento. O refinamento resulta da subdivisão de triângulos em dois cortando-os através de um segmento unindo o ponto médio de suas hipotenusas aos seus respectivos vértices opostos (o que mantém a natureza geométrica dos triân-

gulos). Para preservar a estrutura de triangulação, o refinamento de um triângulo requer em geral a subdivisão (simultânea) de outros. No entanto, é possível obter-se qualquer triangulação RA (refinada) apenas por sucessivos refinamentos de núcleos [15] de triangulações J1 ou K1 como ilustra-se a seguir.

A Fig. 1 ilustra uma triangulação RA classificando os seus triângulos de acordo com os conceitos núcleo, fonte e bacia introduzidos em [15] e que se seguem.

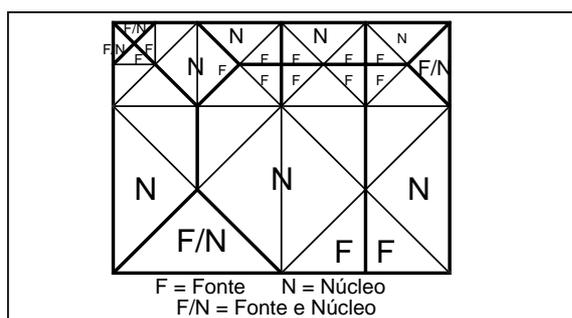


Figura 1 - Elementos da triangulação RA

a) Núcleo (N): conjunto formado por dois triângulos com uma hipotenusa comum; denomina-se núcleo (degenerado) também a todo triângulo com sua hipotenusa no bordo da triangulação. O refinamento mais elementar de uma triangulação RA corresponde à subdivisão (ao meio) dos triângulos de um de seus núcleos. Por refinamento de núcleos, produz-se todos os possíveis refinamentos de triangulações RA.

b) Caminho de refinamento (C): é uma sequência de triângulos em que cada triângulo é vizinho de seu sucessor por sua hipotenusa; o caminho maximal que se inicia em um triângulo, termina necessariamente em um triângulo pertencente a um núcleo; o seu núcleo terminal. Se um triângulo deve ser subdividido ao meio, como parte de um refinamento, é minimamente necessário subdividir-se ao meio todos os triângulos que o sucedem em seu caminho até um núcleo terminal. A essa sequência de subdivisões denomina-se *refinamento básico do triângulo*. Note-se que todo refinamento básico pode ser decomposto em uma sequência de refinamentos de núcleo: refina-se reiteradamente o núcleo terminal do caminho maximal do triângulo, até que ele próprio seja subdividido.

c) Fonte (F): todo triângulo cujos catetos não são hipotenusas de seus vizinhos. Esses triângulos são iniciais a qualquer caminho em que ocorrem.

d) Bacia de um núcleo: conjunto de todos os triângulos cujos caminhos terminam em um núcleo. Na figura 1, as bacias estão delimitadas por traços mais espessos. Em [23], descreve-se um esquema iterativo genérico para construir-se triangulações RA minimais segundo crité-

rios especificados. Tipicamente determina-se a triangulação RA refinada minimal que atende aos requisitos de um problema caracterizados por um critério de aprovação fundado nos triângulos: a triangulação satisfaz os requisitos se cada triângulo atende ao critério de aprovação. O método de construção da triangulação RA minimal envolve exclusivamente refinamentos básicos sucessivos de uma triangulação J1 de partida. Aplicações específicas desse esquema são encontradas em [15] e [17]. A triangulação RA ilustrada na seção 4.1 foi construída por meio desse esquema.

2.2 Triangulação de Delaunay (DT)

Dado um conjunto S de pontos no plano, um triângulo t com vértices em S satisfaz ao *critério de Delaunay* quando a circunferência pelos três vértices de t (o circuncírculo de t) não contiver nenhum outro ponto de S em seu interior. Uma *triangulação de Delaunay de S* é uma triangulação cujos triângulos atendem ao critério de Delaunay. Algumas propriedades e aplicações da triangulação de Delaunay podem ser encontradas em [18, 20, 21, 22]

Dentre os vários algoritmos para determinar a triangulação de Delaunay de um conjunto de pontos, ressaltam-se o de inserção incremental de pontos, o método da bolha, o método usando flips de arestas a partir de uma triangulação qualquer de partida e o “divide and conquer” (o de melhor complexidade).

A implementação do método descrita na seção 4.2 assemelha-se ao método da bolha, onde a partir de uma aresta de um triângulo no bordo da triangulação corrente, procura-se um ponto que, com a aresta, atenda ao critério de Delaunay.

2.3 Triangulação Gulosa (GT)

No conceito original, a triangulação gulosa de um conjunto S com n pontos é a triangulação obtida a partir de um conjunto vazio de arestas e adicionando-se a cada passo a aresta de menor comprimento compatível entre dois pontos de S . Uma aresta compatível é definida como uma aresta que não intercepta nenhuma das arestas anteriormente adicionadas. Conjectura-se que a triangulação gulosa seja uma boa aproximação para a triangulação de custo mínimo (menor somatório de arestas). Mais detalhes a respeito desta triangulação podem ser encontrados em [01, 07, 12, 18, 19].

Em [01], Aicholzer et Al apresentam várias propostas de algoritmos GT, descrevendo, além do algoritmo original em que são consideradas arestas gulosas, apresentado acima, vários outros algoritmos usando conceitos de triângulos gulosos.

3 ESTRUTURA TOPOLÓGICA E MODELOS

A estrutura de dados usada baseia-se em relações de adjacências e de posição. Os aspectos topológicos e geométricos são separados. Com esta separação, é possível a generalização da parte topológica e a simples redefinição dos métodos do modelo geométrico para as diversas geometrias. Os principais objetos desta estrutura são mostrados na figura 2. A estrutura topológica é a mesma, independente do modelo de triangulação usado (GT, DT ou RA). Chamamos esta estrutura topológica genérica de triângulo. Ocorre comunicação entre a geometria (informações de triângulo) e a topologia (triângulo) apenas para solicitação de operações a serem executadas (criação, subdivisão e flip).

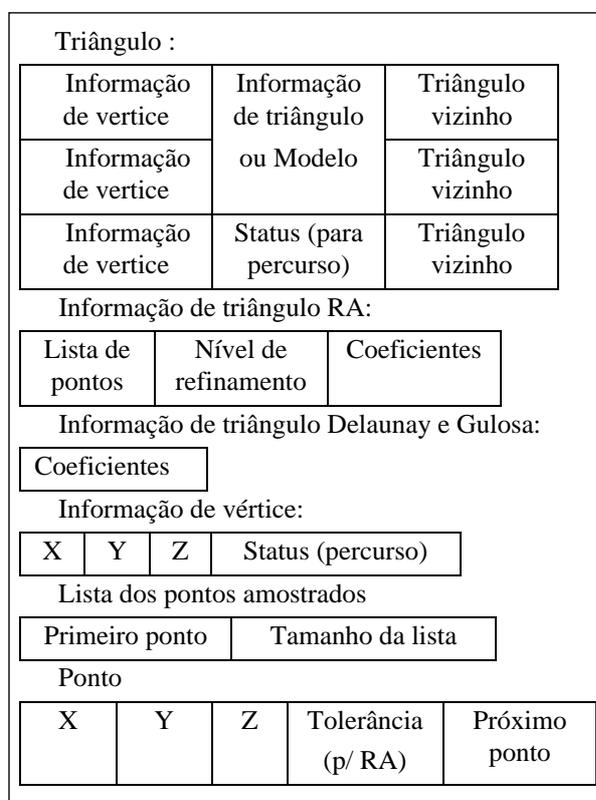


Figura 2- Estrutura Básica para Triangulações

O objeto triângulo “sabe” quem são seus vértices e quem são seus vizinhos, ambas informações numa certa ordenação, digamos no sentido horário. Assim, a triangulação contém informações topológicas sem redundâncias, normalmente presentes em outras estruturas.

O objeto modelo (informações de triângulo) mantém as informações necessárias para modelagem de uma superfície sobre a triangulação (ver [14, 15, 16, 17]). Estas informações são basicamente os coeficientes de uma superfície polinomial por partes definida em

cada triângulo e outras estruturas necessárias para calcular a aproximação desta superfície. O modelo gerencia ainda os procedimentos de criação, subdivisão e flip, mantendo a parte geométrica compatível.

3.1 Operações Topológicas Genéricas

As três operações topológicas da classe triângulo usadas para criar as triangulações, são resumidas a seguir:

a) Criação de um triângulo. Cria um objeto triângulo a partir de três informações de vértices.

b) Subdivisão de um triângulo a partir de um índice para uma de suas arestas e um novo vértice dado. O efeito geométrico da subdivisão é a retirada de uma aresta e a inserção de três novas arestas. A primeira aresta inserida une o vértice oposto à aresta removida ao novo vértice, e as outras duas unem os vértices da aresta removida ao novo vértice. Note que mesmo que o novo vértice não esteja sobre a aresta removida, a consistência topológica será mantida. De fato, este “truque” será usado na criação das triangulações, para inserção de novos vértices. Se o triângulo possuir vizinho pela aresta removida, este vizinho deverá ser subdividido também. As novas vizinhanças (colagem) serão compatibilizadas. Note que um ou dois triângulos novos são implicitamente criados e adicionados à triangulação por esta operação.

c) Troca de uma aresta comum a dois triângulos adjacentes, pela outra definida pelos vértices opostos de ambos. Esta operação é mais conhecida por “flip”.

Note que o resultado geométrico das operações de subdivisão e flip pode não ser uma triangulação (no plano), mas o resultado topológico certamente será. A operação de colagem está implícita na operação de subdivisão, uma vez que esta é necessária para compatibilizar a vizinhança (ligar novos triângulos criados aos antigos) e será necessária no caso em que uma das arestas inseridas já esteja na triangulação, como é o caso no fechamento da borda da triangulação.

4 IMPLEMENTAÇÕES

Realizamos a implementação usando orientação a objetos (compilador Gnu c++) e no ambiente X-Windows, usando as bibliotecas Xlib (X11) para Work-Station Sun (OS Solaris). Nas seções seguintes serão apresentados de forma resumida os algoritmos usados na implementação de cada tipo de triangulação.

4.1 Implementação da Triangulação RA

O algoritmo para criação da triangulação retangular adaptativa resume-se nos seguintes passos:

1) Determinar os 4 pontos de um quadrado inicial que envolva toda a nuvem de pontos (bounding box);

2) Triangular os vértices do quadrado inicial mais o ponto central deste (em quatro triângulos J1). Este passo subdivide-se nos seguintes:

a) criar o primeiro triângulo com 3 vértices quaisquer dos 4 cantos do quadrado;

b) subdividir este triângulo em dois passando como novo vértice o ponto restante (canto) do quadrado inicial;

c) trocar a diagonal do quadrado (hipotenusa dos dois triângulos) pela outra diagonal;

d) subdividir destes dois triângulos em 4 passando como novo vértice o ponto central da diagonal do quadrado (hipotenusa em ambos).

3) Refinar onde os critérios de refinamento exigirem, sempre subdividindo os triângulos pelo meio de sua hipotenusa (esta será sempre a aresta de índice 0).

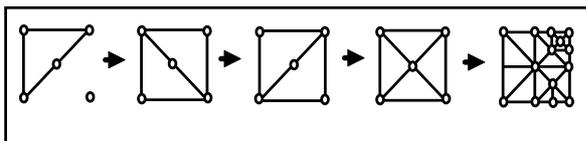


Figura 3 - Realizando a triangulação RA

A figura 3 (acima) mostra a construção da triangulação RA.

A operação de troca após a subdivisão em dois triângulos é realizada para que a diagonal do quadrilátero (hipotenusa nos dois triângulos) receba o índice 0. Assim, durante a etapa de refinamento, a subdivisão de um triângulo sempre poderá ser processada na aresta de índice 0 (hipotenusa).

Se for considerada a existência de alturas em cada ponto e uma superfície matemática (que pode ser constituída por retalhos de Bézier em cada triângulo) que melhor represente o conjunto de pontos (pode ser aproximada pelo método dos mínimos quadrados), pode-se estabelecer um erro como sendo uma distância medida entre cada altura e a superfície aproximada. Assim, o refinamento pode ser efetuado, por exemplo, no triângulo que possuir o maior erro quadrático entre esta superfície e o conjunto de pontos internos. O critério de parada pode ser uma tolerância para o erro (distância acima definida) atingida em todos os pontos da amostra. Convém ressaltar que este método pode não convergir para a melhor solução, mas, em se tratando de terrenos, certamente, as tolerâncias serão atingidas. Note que após uma subdivisão, a superfície pode mudar segundo o critério adotado, localmente ou globalmente. Em nosso caso, usamos o critério local.

Assim, para cada triângulo subdividido, deve ser recalculada pelos mínimos quadrados a nova superfície aproximada apenas nos triângulos envolvidos na subdivisão.

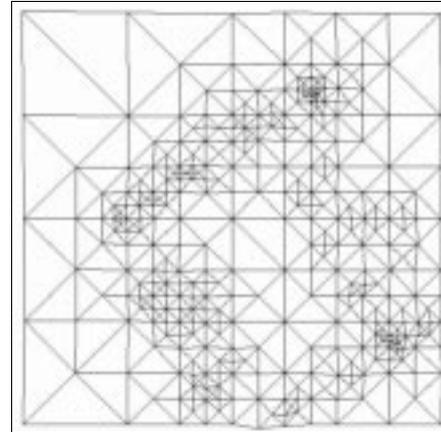


Figura 4 - Triangulação RA

A figura 4 mostra a triangulação RA de uma amostra de 43.000 pontos aproximadamente, usando o algoritmo implementado. As regiões com mais triângulos são edifícios residenciais em forma retangular. O terreno é mais ou menos plano sem ondulações acentuadas (exceto nos edifícios). Nota-se pelas regiões com menos triângulos.

4.2 Triangulação de Delaunay

O algoritmo implementado para criação da triangulação de Delaunay de um conjunto dado de pontos pode ser resumido nos seguintes passos:

- 1) Tomar um ponto qualquer da triangulação;
- 2) Determinar o ponto mais próximo deste;
- 3) Para estes dois pontos, determinar o primeiro ponto que atenda a restrição de Delaunay, e criar o primeiro triângulo;
- 4) Para cada aresta do novo triângulo, determinar um ponto exterior que atenda o critério de Delaunay;
- 5) Caso haja o ponto exterior, subdividir o triângulo pela aresta considerada e o ponto encontrado e trocar (flip) a aresta formada pelo vértice oposto à aresta removida e o novo vértice (de índice 0), visando restabelecer a triangulação de Delaunay. As questões relacionadas à compatibilização das vizinhanças (colagem) são resolvidas na subdivisão, já apresentada.
- 6) Repetir iterativamente os passos 4 e 5 para o novo triângulo inserido na triangulação pelo passo 5 (subdivisão e troca) até que não hajam mais pontos que atendam o critério de Delaunay.

A figura 5 mostra a determinação de um triângulo de Delaunay vizinho a um dado triângulo por uma de

suas arestas, ilustrando dois casos que podem ocorrer (quadrilátero convexo ou não). Na configuração dada na parte inferior da figura 5 (quadrilátero não convexo), com a subdivisão realizada na 1ª etapa do passo 5, será removida uma aresta e dentre as novas arestas inseridas na triangulação, uma delas faz com que a geometria resultante desta etapa não seja uma triangulação, com arestas se interceptando. Isto pode ser isto na 2ª fase da parte inferior da figura 5. Porém, topologicamente a triangulação está mantida. Com a realização da 2ª etapa do passo 5 (troca da aresta inserida), a aresta retirada na subdivisão certamente voltará para a triangulação. Isto pode ser visto na 3ª fase da parte inferior da figura 5. Note que com isto, a geometria errada da fase 2 será corrigida restabelecendo geometricamente a triangulação de Delaunay.

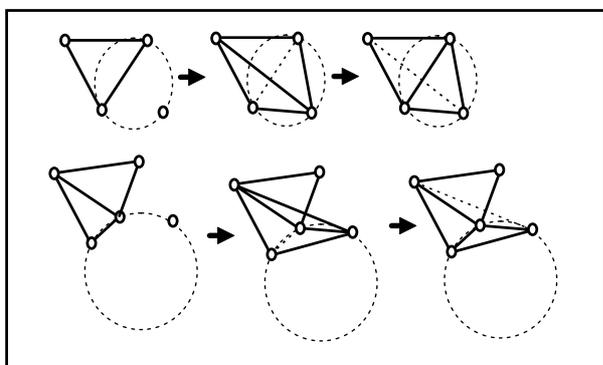


Figura 5 - Realizando a triangulação de Delaunay

Este algoritmo segue a mesma estratégia do método da bolha, expandindo-se para os lados. Dependendo do sentido considerado na estrutura topológica, a expansão pode se dar para fora ou para dentro. Resultados podem ser visualizados na figura 6.

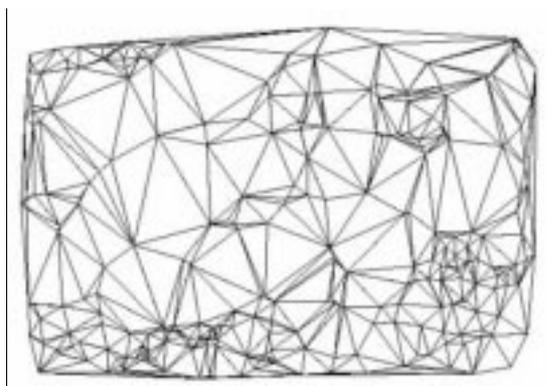


Figura 6 - Triangulação de Delaunay

4.3 Nova Proposta para a Triangulação Gulosa

Já apresentada, a idéia da triangulação gulosa é inserir na triangulação a cada passo a menor aresta compatível. Note que na estrutura apresentada não se tem explicitamente o conceito de aresta, mas sim o conceito

de vértices e triângulos. Para evitar a necessidade de um objeto aresta, introduzimos este algoritmo guloso. Numa fase inicial, cria-se uma lista para cada ponto, contendo os outros $n-1$ pontos ordenados pelas suas distâncias ao ponto. Sendo p_1 e p_2 os pontos extremos de uma aresta que já esteja na triangulação (a primeira aresta pode ser dada por qualquer ponto e seu mais próximo), o critério guloso é sintetizado da seguinte forma:

- toma-se da lista ordenada de p_1 o primeiro ponto (mp_1). Este ponto é o mais próximo de p_1 ;
- toma-se da lista ordenada de p_2 o primeiro ponto (mp_2). Este ponto é o mais próximo de p_2 ;
- se estes pontos coincidirem, está determinado o ponto que atende o critério guloso;

d) se não coincidirem, o seguinte método é usado para "jogar fora" um dos dois: se a distância entre mp_1 e p_2 for menor que a distância entre mp_2 e p_1 e além disso o ponto mp_2 não estiver dentro do triângulo (poderia ser círculo) definido por (p_1, p_2, mp_1) , então o ponto mp_2 é descartado, tomando-se o seguinte da lista de p_2 (o seguinte mais próximo de p_2); caso contrário, o ponto mp_1 é descartado tomando-se o seguinte da lista de p_1 (o seguinte mais próximo de p_1). Se não houver ponto na lista ordenada para um dos dois pontos, encerra-se a procura não havendo o ponto que atenda o critério, caso contrário retorna-se ao passo c). Note que só podem ser comparadas distâncias para os pontos que estejam de um mesmo lado do segmento (p_1, p_2) .

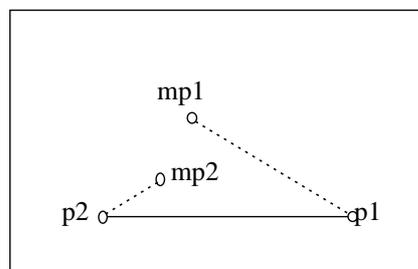


Figura 7 - Caso ilustrativo do critério guloso

Observe que não seria suficiente apenas comparar a distância mp_2-p_1 com mp_1-p_2 e descartar o ponto referente à maior delas, pois isto pode gerar um resultado em que o triângulo formado com o ponto encontrado possua algum ponto em seu interior, o que não é compatível. A figura 7 ilustra um caso onde o próprio ponto descartado estaria dentro do triângulo formado. Considerando apenas o critério das distâncias, o ponto mp_2 seria descartado e tomado o próprio mp_1 como mais próximo de p_2 . Assim, mp_2 ficaria dentro do triângulo $p_1-p_2-mp_1$. Com o critério de verificação da interioridade ao possível triângulo formado, o ponto

mp1 será descartado e o triângulo p1-p2-mp2 estará definido, o que é condizente.

Usando este critério, o algoritmo para criação da triangulação gulosa segue os seguintes passos:

- 1) Tomar um ponto qualquer da amostra (p1);
- 2) Determinar o ponto mais próximo de p1 (p2), obtendo a primeira aresta da triangulação gulosa (p1,p2);
- 3) Determinar um terceiro ponto usando o critério guloso e criar o primeiro triângulo da triangulação gulosa, tendo estes 3 pontos como vértices;
- 4) Para cada aresta do novo triângulo, encontrar um ponto externo usando o critério guloso;
- 5) Subdividir o triângulo pela aresta e o ponto encontrado e trocar (flip) a aresta formada pelo vértice oposto à aresta removida e o novo vértice inserido (de índice 0). Esta troca retornará a aresta antiga na triangulação (esta passa a ter índice 0), restabelecendo a triangulação gulosa; um novo triângulo será criado nesta subdivisão. Tal qual no algoritmo para a triangulação de Delaunay, a operação de subdivisão se encarrega da compatibilização das vizinhanças.
- 6) Repetir os passos 4 e 5 iterativamente, até que não se tenha mais pontos atendendo ao critério guloso, ficando definida a triangulação gulosa.

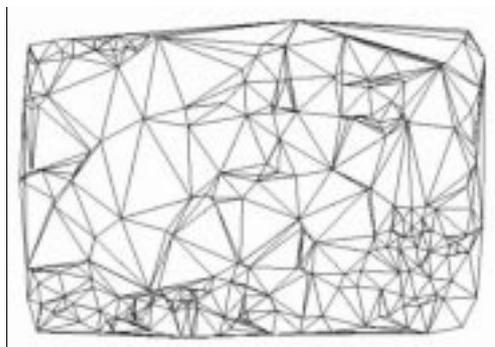


Figura 8 - Triangulação gulosa

A figura 8 mostra o resultado visual usando este algoritmo guloso, para a mesma amostra de pontos da triangulação de Delaunay mostrada na figura 6. Note que em algumas regiões, as triangulações resultantes de ambos algoritmos são diferentes.

Sendo n o número de pontos, para cada ponto temos $n-1$ arestas e para cada dois pontos dados há $2(n-2)$ pares de arestas que convergem para um mesmo ponto. Ordenando-se estas segundo algum critério, e escolhendo as duas primeiras desta ordenação, desde que não haja algum ponto no interior do triângulo formado, de certa forma estamos estabelecendo algum algoritmo guloso. Em nosso algoritmo o critério usado, além do de proximidade foi o teste do triângulo (da

mesma forma poderia ser o teste do círculo). De certa forma, o algoritmo mistura os conceitos de Delaunay com o critério guloso original.

5 ANÁLISE DOS ALGORITMOS

Em todas as implementações realizadas, algumas particularidades foram omitidas. Por exemplo, na triangulação gulosa, como saber que um triângulo não interceptará outro já determinado ou como evitar loops e casos em que o ponto determinado já está na triangulação, seja como vértice de um triângulo, seja como parte de uma aresta de um triângulo a ser vizinho do triângulo determinado ou como parte de duas arestas de dois triângulos nesta mesma situação anterior. Estas particularidades, bem como as questões de vizinhança (colagem) nos processos de subdivisão e *flip* foram todas resolvidas no trabalho de implementação, com testes simples. A implementação de algoritmos recursivos facilitou esta tarefa de controle. Os algoritmos foram apresentados apenas com os principais pontos, visando sua fácil compreensão.

Não faz sentido tentar estabelecer a complexidade para triangulação RA, uma vez que esta é dependente do critério de parada do processo de refinamento.

A complexidade para o algoritmo de Delaunay apresentado é $O(n^2)$. O número de arestas de uma triangulação obedece a $2n - 3 \leq |T(P)| \leq 3n - 6$, sendo n o número de pontos da amostra. Para determinar o ponto externo a cada aresta de um triângulo que atenda ao critério de Delaunay, deve-se percorrer todos os n pontos da amostra. Desta forma, a complexidade resultante é quadrática.

Quanto à complexidade do algoritmo guloso, para cada aresta de cada triângulo encontrado, o passo de determinar o ponto externo que atenda ao critério guloso, pode percorrer $2(n-2)$ pontos ordenados da lista de ambos vértice, o que resulta em $2(n-2)$ operações. Como o número de arestas é dado por $2n - 3 \leq |T(P)| \leq 3n - 6$, isto deixa a complexidade do processo de triangular quadrática. A complexidade final é dada pelo processo de ordenação inicial, necessário para determinar a lista ordenada dos pontos mais próximos de cada ponto, ou seja, é $O(n^2 \log n)$.

6 CONCLUSÕES E PERSPECTIVAS FUTURAS

A finalidade principal do trabalho foi o desenvolvimento de uma base triangular que contivesse apenas as informações topológicas, sendo genérica, independente do modelo ou geometria. As operações de criação de triângulo, de subdivisão a partir de uma aresta e um ponto dado (com colagem implícita) e de

flip (troca) de uma determinada aresta foram suficientes para a determinação das triangulações apresentadas no texto. Repare que a triangulação de Delaunay poderia ser encontrada a partir da gulosa, por *flips*. É claro que tal algoritmo seria mais complexo.

Não encontramos na literatura [01, 07, 12, 13, 18, 19] a respeito nenhuma proposta de triangulação gulosa semelhante ao nosso algoritmo, apresentando assim este como uma nova proposta de triangulação gulosa, uma vez que de certa forma, usamos gulodice. A complexidade do método implementado compara-se à do melhor algoritmo de que se tem notícia [19] para a tradicional. No caso da triangulação de Delaunay, pode-se estudar formas de redução da complexidade que está muito longe da melhor, bem como outras alternativas de realizar o algoritmo ou otimizações neste, mas isto foge ao escopo deste trabalho.

Estas implementações podem e foram usadas como base para análise visual ou numérica de resultados de sistemas cuja a saída seja dada por pontos num espaço 3D, bem como em modelagem geométrica de superfícies. Como exemplo deste último, temos a reconstrução de terrenos a partir de imagens.

Como próximos trabalhos, pretendemos estabelecer uma melhor base matemática para o algoritmo guloso apresentado (em todos os testes realizados, o resultado foi uma triangulação), bem como realizar medidas e comparações com a triangulação de custo mínimo para se ter idéia da distância a esta.

10 AGRADECIMENTOS

O desenvolvimento desta pesquisa contou com o apoio financeiro da CAPES e do Projeto PROTEM II - GEOTEC do CNPQ e com a colaboração dos Professores Antonio Oliveira (orientação) e Ronaldo Marinho Persiano (orientação e revisao).

11 BIBLIOGRAFIA

- [01] O. Aicholzer, F. Aurenhammer and G. Rote. New Greedy Triangulations (Extended Abstract). Proc. 12^o Symposium on Computacional Geometry. Germany, 1996.
- [02] F. Hurtado, M. Noy and J. Urrutia. Flipping edges in triangulations of polygons and point sets. 12^o ACM Computacional Geometry Symposium. 1995.
- [03] S. Hanke, T. Ottmann and S. Schuierer. The edge-flipping distance of triangulations. Proc. 12^o ACM Symposium on Computacional Geometry. 1995.
- [04] M. R. Garey and D. S. Johnson. Computers and Intractability. A Guide to the Theory of NP-completeness. W. H. Freeman, 1979.
- [05] F. Hurtado and M. Noy. The graph of triangulations of a convex polygon, Tech. Rep. MA2-IR-94-13, Universitat Politècnica de Catalunya. 1994.
- [06] L. S. Heath and S. V. Pemmaraju. New Results for the Minimum Weight Triangulation Problem. Technical Report. Virginia Polytechnic Institute and State University. Blacksburg. 1992. Algorithmica (12), pg 533-552. 1994.
- [07] E. L. Lloyd. On triangulations of a set points in the plane. Proceedings of the Eighteen IEEE Symposium on Foundations of Computer Science, 18, pg 228-240.
- [08] A. Lingas. A New Heuristic for the Minimum Weight Triangulation. SIAM Journal of Algebraic and Discrete Methods, 8, pg 646 - 658. 1987.
- [09] M. T. Dickerson, S. McElfresh and M. Montague. New Algorithms and Empirical Findings on Minimum Weight Triangulation Heuristics. Proceedings of the 11th Annual Symposium on Computacional Geometry. 1995.
- [10] M. T. Dickerson and M. Montague. A (Usualy?) Connected Subgraph of the Minimum Weight Triangulation. Extended Abstract. Proceedings of the 12 Annual ACM Symposium on Computational Geometry. 1996.
- [11] J. M. Keil. Computing a Subgraph of the Minimum Weight Triangulation. Computational Geometry: Theory and Applications, n^o 4, pg 13 - 26. 1994.
- [12] A. Lingas. The greedy and Delaunay triangulations are not bad in the average case and minimum weight triangulation of multi-connected polygons is NP-complete. Foundations of Computational Theory, Lecture Notes in Computer Science 158, pg 270-284. Springer-Verlag, Heidelberg. 1983.
- [13] Cláudio T. SILVA, Joseph S. B. MITCHELL, Arie E. KAUFMAN. Automatic Generation of Triangular Irregular Networks Using Greedy Cuts. Stony Brook, NY, 1995.
- [14] L. M. G. GONÇALVES e A. A. F. OLIVEIRA. Modelagem de Terreno com Triangulação CFK Adaptativa. Anais do IX Sibgrapi, pg 337-338. 1996.
- [15] L. P. BUENO. Minimização da Energia de Deformação na Modelagem de Terreno. Dsc Tese. COPPE-UFRJ, 1996.
- [16] G. FARIN. Triangular Bernstein-Bézier Patches. Computer Aided Geometric Design, V. 3, 83-127. North-Holland, 1986.

- [17] Ronaldo C. M. PERSIANO and Antonio A. L. JUNIOR. Boundary Evaluation of CSG Models by Adaptative Triangulation. Proceedings of CSG 94 Conference - Set-theoretic Solid Modeling: Techniques and Application. 1994.
- [18] D. G. Kirkpatrick. A note on Delaunay and optimal triangulations. Information Processing Letters, 10, pg 127-128. 1980.
- [19] M. T. Dickerson, R. L. Drysdale, S. McElfresh and E. Wetzi. Fast Greedy Triangulations. Proceedings of the 10th Annual Symposium on Computational Geometry. Pg 211-220. 1994
- [20] C. LAWSON. Properties of n -Dimensional Triangulations. Computer Aided Geometric Design, 3, 231-246. 1986.
- [21] D. WATSON. Computing the n -Dimensional Delaunay Tessellation with Application to Voronoi Polytopes. The Computer Journal, 24, 167-172. 1981.
- [22] L. GUIBAS and J. STOLFI. Primitives for the Manipulation of General Subdivisions and the Computation of Voronoi Diagrams. ACM Transaction on Graphics 4, 75-123. 1985.

- [23] J. M. R. Neves and R. M. Persiano. Visualizing Scalar Fields Represented by Adaptive Square Triangulations. Proceedings of the X Brazilian Symposium on Computer Graphics and Image Processing, 1997.

- [24] R. M. Persiano, J. L. Comba, and V. Barbalho: An adaptive triangulation scheme and construction. Anais do SIBGRAPI'93, Recife, 1993.