MAC 747 GEOMETRIA COMPUTACIONAL

PARTIÇÃO DE UM POLÍGONO EM POLÍGONOS MONÓTONOS

1. Introdução

O algoritmo que veremos para particionar um polígono em polígonos monótonos (ou trapézios) usa a técnica da linha de varredura (*sweepline technique*).

2. Event points

Os pontos eventos (*event points*) serão os pontos do polígono. O *event-point schedule* será determinado estaticamente e consistirá de uma lista ordenada $\mathcal{E}$ dos pontos do polígono em ordem decrescente da y-coordenada.

3. Sweepline structure

A *sweepline structure* S manterá as arestas do polígono que estão sendo intersectadas pela linha de varredura L em ordem das x-coordenadas dos pontos de intersecção (análogo ao que foi feito no algoritmo `Intersecção-de-Segmentos`). Além disso, para a partição em polígonos monótonos (para a partição em trapézios esses vértices não serão necessários), a estrutura manterá os vértices do polígono que são vértices de suporte das arestas superiores dos trapézios correntes (i.e. trapézios para os quais, até o momento, não se conhece o vértice de suporte da aresta inferior). Alguns desses vértices são pontas interiores para baixo do polígono que ainda precisam ser eliminadas, os outros são vértices que ajudarão a eliminar eventuais pontas interiores para cima.

A *sweepline structure* deve permitir operações de inserção, remoção e busca de segmentos e vértices em tempo $O(\log n)$ por operação. Para isso podemos usar como estrutura de dados uma árvore de busca balanceada como, por exemplo, árvore AVL, árvore 2-3, ...

4. Atualização da sweepline structure

A atualização da *sweepline structure* S depende do tipo do evento. Suponha que v está entre as arestas a e b sobre L e que v é um vértice compartilhado pelas arestas c e d.

Caso 1: O segmento c está acima de L e d abaixo de L.
Remova c da estrutura S e insira d.

$$(\dots, a, c, b, \dots) \Longrightarrow (\dots, a, d, b, \dots)$$

Caso 2: Ambos c e d estão acima de L.
Remova c e d da estrutura S.

$$(\dots, a, c, d, b, \dots) \Longrightarrow (\dots, a, b, \dots)$$

Caso 3: Ambos c e d estão abaixo de L.
Insira c e d na estrutura S.

$$(\dots, a, b, \dots) \Longrightarrow (\dots, a, c, d, b, \dots)$$

5. Algoritmo de varredura

1. $S \leftarrow \emptyset$. [Inicialização da *sweepline structure* S.]
2. Ordene os vértices em ordem decrescente de y-coordenadas e coloque-os na lista ordenada $\mathcal{E}$
   [Este passo gasta tempo $O(n \log n)$.]
3. Para cada vértice v na lista ordenada $\mathcal{E}$ faça
   Encontre a aresta a e b imediatamente a esquerda e direita de v e sejam c e d as arestas que compartilham v como extremidade.
   Caso 1: c está acima de L e d abaixo.
   Seja u o vértice de suporte da aresta superior do trapézio que tem o vértice v como vértice de suporte da aresta inferior.
   Remova c de S e insira d em seu lugar.
   Remova u da estrutura S e insira v em seu lugar.
   [Note que v foi inserido na *sweepline structure* pois agora v passa a ser o vértice de suporte da aresta superior de um trapézio.]
   Se u for uma ponta interior para baixo então trace a diagonal vu.
   [Eliminação de uma eventual ponta interior para baixo.]
   Caso 2: v é ponta interior para baixo (c e d estão acima de L.)
   Sejam u e w os vértices de suporte das arestas superiores dos dois trapézios que têm v como vértice de suporte da aresta inferior.
   [Note que um desses trapézios tem c como uma de suas arestas laterais e o outro tem d como uma de suas arestas laterais.]
   Remova de S os segmentos c e d.
   Remova de S os vértices u e w e insira o vértice v no lugar.
   [Note que v é uma ponta interior que passa a ser o vértice de suporte da aresta superior de um trapézio—esta ponta interior será eliminada quando o vértice de suporte da aresta de inferior do trapézio for encontrada.]
   Se u é ponta para baixo então trace a diagonal vu.
   Se w é ponta para baixo então trace a diagonal vw.
   [Eliminação de eventuais pontas interiores para baixo.]
   Caso 3: v é ponta interior para cima (c e d estão abaixo de L).
   Seja u o vértice de suporte da aresta superior do trapézio que tem v como vértice de suporte da aresta inferior.
   Insira os segmentos c e d em S.
   Remova u de S e o ponto v.
   [Note que v passa a ser o vértice de suporte da aresta superior de dois trapézios.]
   Trace a diagonal entre vu.
   [Eliminação de uma ponta interior para cima.]

6. Análise do algoritmo

Para cada vértice v precisamos encontrar a aresta imediatamente à esquerda e a aresta imediatamente à direita em L. Para fazer isso, sabendo as coordenadas dos pontos extremos de cada aresta, podemos calcular onde a linha de varredura corta a aresta e comparar com a y-coordenada do ponto. Precisamos fazer esta operação n vezes (n é o número de vértices do polígono) a um custo de $O(\log n)$ por vez (se usarmos uma árvore de busca balanceada para manter a *sweepline structure*).

O tempo gasto em cada um dos casos do passo 3 é $O(\log n)$ (inserções, buscas e remoções a um custo de $O(\log n)$ por operação). Isso nos da uma complexidade de tempo de $O(n \log n)$ para todo o algoritmo, pois o passo 3 será executado n vezes.