

Localização no Plano

Claudio Esperança
Paulo Roma



O Problema

- Dada uma subdivisão poligonal do plano com n vértices e um ponto de consulta q , determinar qual face contém q
 - q pode coincidir também com arestas ou vértices
- Assume-se que a subdivisão é representada através de uma estrutura de dados adequada
 - DCEL *double-connected edge list*
 - *Half-edge data structure*



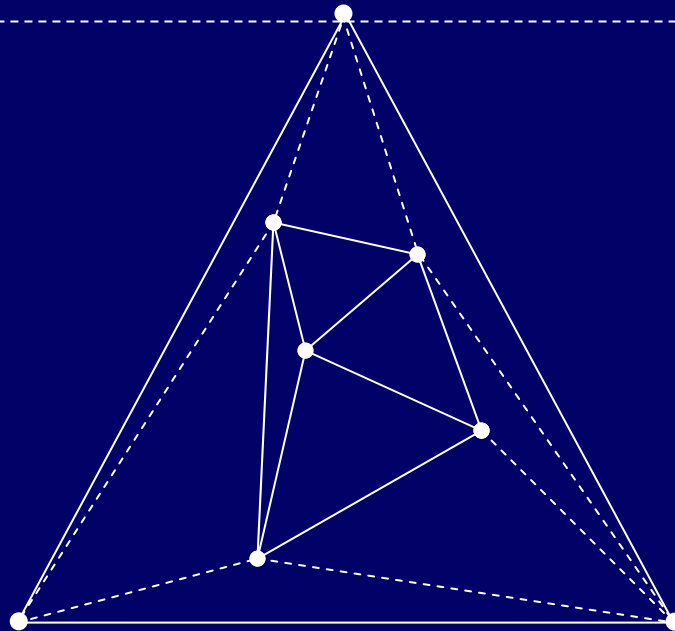
O Problema

- *k-d-trees* e *quadtrees* resolvem o problema razoavelmente bem na prática, mas boas cotas de complexidade nunca foram estabelecidas
- Outras estruturas com
 - Tempo $O(\log n)$ e espaço $O(n \log n)$
 - Tempo $O(\log^2 n)$ e espaço $O(n)$
- Triangulações hierárquicas (Kirkpatrick)
 - Tempo $O(\log n)$ e espaço $O(n)$
 - Interessante, mas constante é alta
 - Pouco prática
 - Variante randomizada é prática



Algoritmo de Kirkpatrick

- Assume que subdivisão é uma triangulação e face externa é um triângulo
 - Triangular se faces são polígonos quaisquer
 - Criar um triângulo grande que contenha o fecho convexo
 - Triangular a região entre os dois



Estrutura de Kirkpatrick

- A estrutura de Kirkpatrick é uma seqüência de triangulações $T_0, T_1, \dots, T_k, k = O(\log n)$
 - T_0 é a triangulação original
 - T_k consiste apenas do triângulo externo
 - Cada triângulo de T_{i+1} intersecta um número constante de triângulos de T_i
 - Na verdade, um número de triângulos limitado por uma constante



Construção da Estrutura

- Dado T_i , construir T_{i+1} de tal forma que
 - T_{i+1} tenha uma fração (constante) dos triângulos de T_i
 - Vai garantir que o número total de triangulações é logarítmico
 - Cada triângulo de T_{i+1} cubra um número constante de triângulos de T_i
 - Permite que consulta seja logarítmica
- Idéia
 - Remover alguns vértices (e arestas incidentes) de T_i e retriangular as cavidades resultantes



Construção da Estrutura

- Na escolha dos vértices a serem retirados, duas restrições devem ser observadas
 - Grau constante: cada vértice deve ter grau $\leq d$
 - Ao deletar o vértice, a cavidade será retriangulada com no máximo $d - 2$ triângulos
 - Cada novo triângulo poderá cobrir no máximo d triângulos da triangulação mais fina
 - Conjunto Independente: vértices a serem deletados não podem ser adjacentes
 - Cavidades não se comunicam e podem ser retrianguladas independentemente
 - Triangulação de cada cavidade leva tempo constante



Construção da Estrutura

- **Lema:** Dado um grafo planar com n vértices, existe um conjunto independente com no mínimo $n/18$ vértices de grau menor ou igual a 8. Este conjunto pode ser computado em tempo $O(n)$
 - Na prática, os números são melhores que estes, mas não existe prova

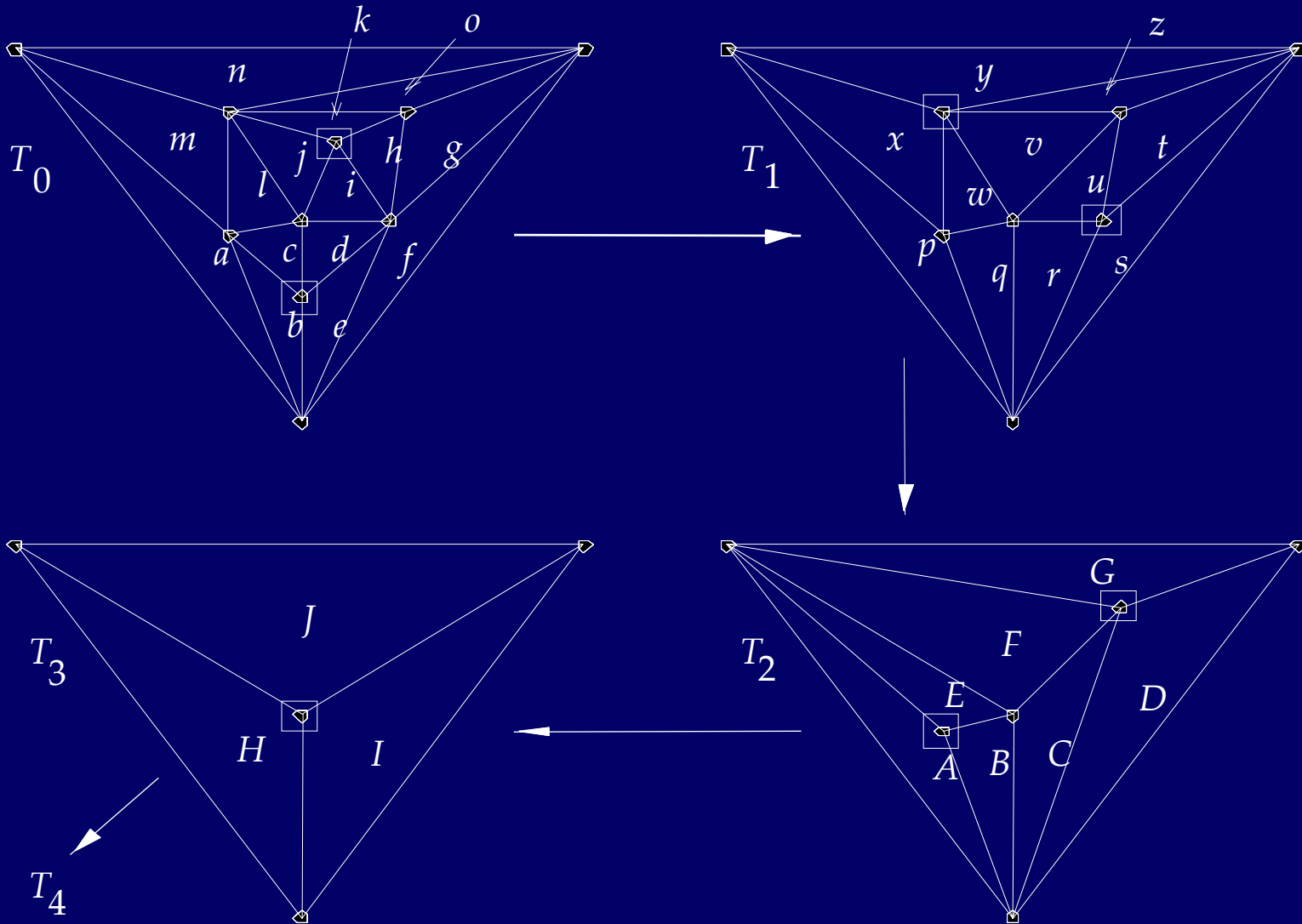


Construção da Estrutura

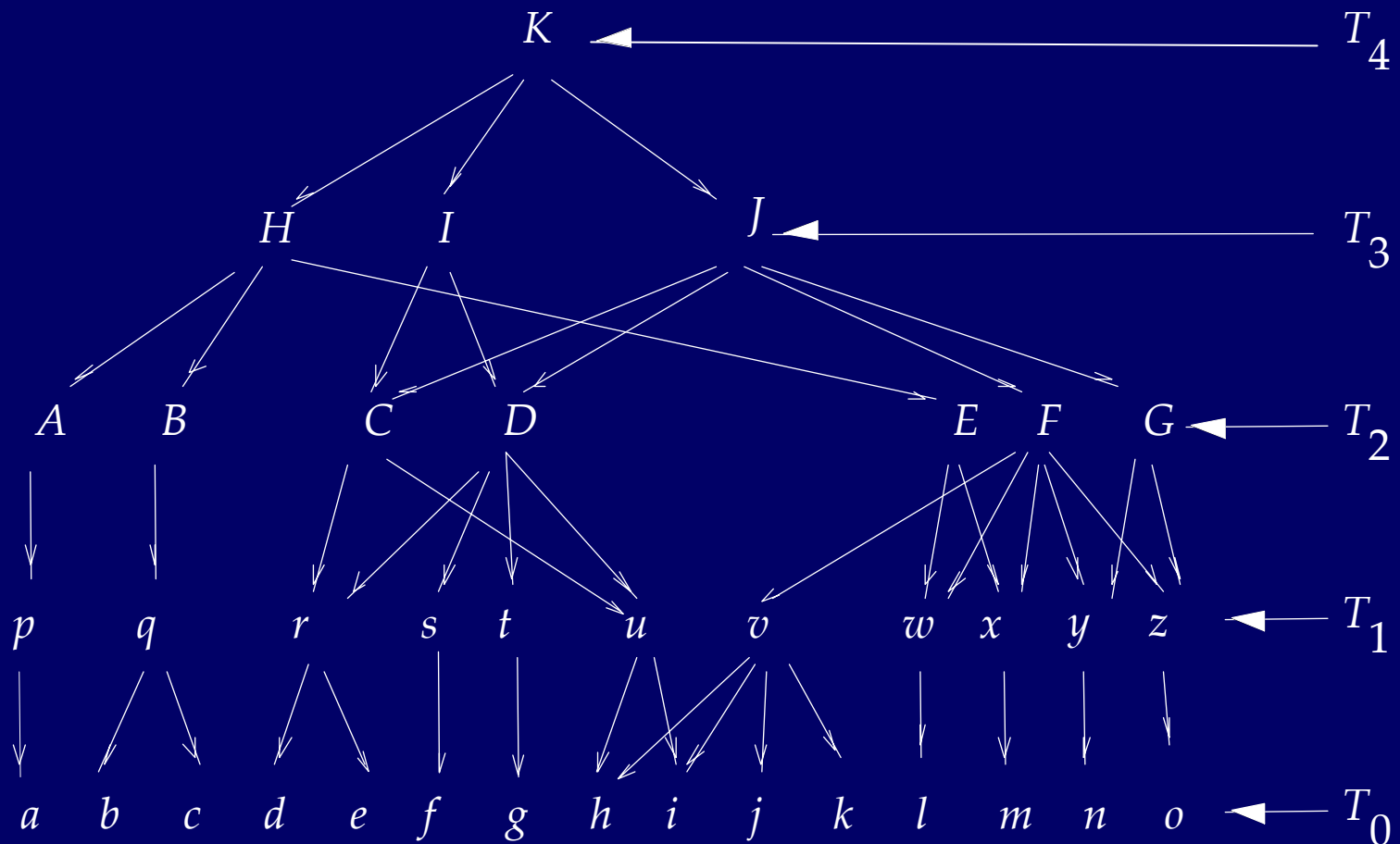
- Começamos com T_0 , a triangulação original
- Escolhe-se um conjunto independente de vértices com grau ≤ 8
 - Os vértices do triângulo mais externo nunca são escolhidos
- Deleta-se os vértices do conjunto independente e retriangula-se as cavidades formadas
- Cada triângulo da nova triangulação vai intersectar no máximo 8 triângulos da triangulação anterior
- No total teremos k triangulações
$$k = \log_{18/17} n \approx 12 \log_2 n$$
- Na verdade, pode-se provar que se escolhermos sempre os vértices de menor grau $k \approx 4.5 \log_2 n$



Construção da Estrutura



Grafo Acíclico Direcionado

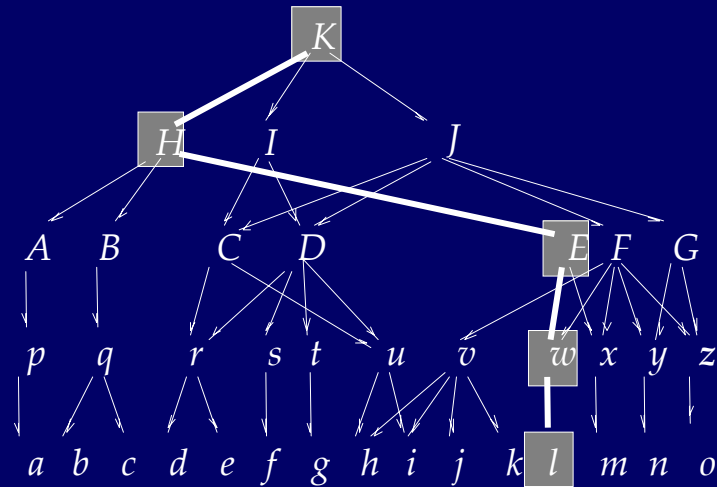
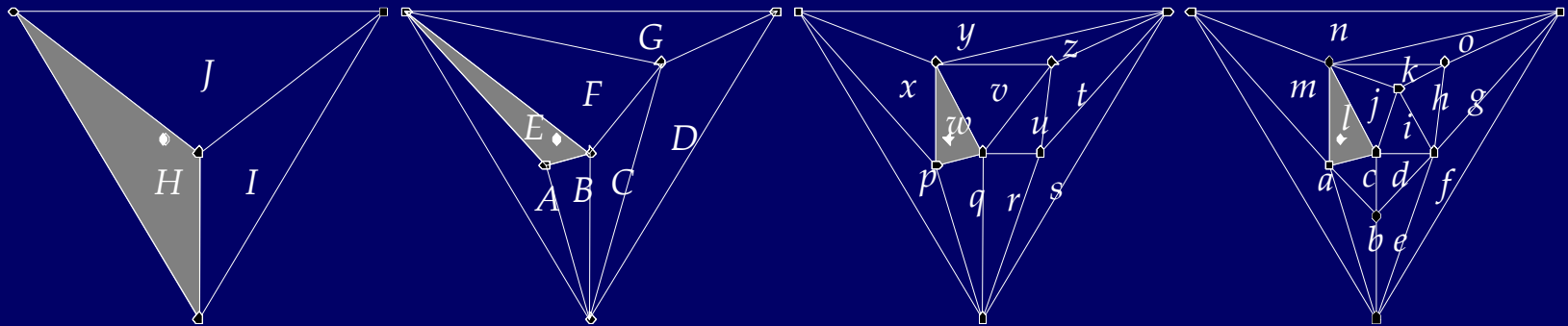


Algoritmo de busca

- Testa-se o ponto q contra T_k
 - Se q não pertence a T_k então a busca termina sem sucesso
 - Caso contrário, testa-se q contra todos os triângulos de T_{k-1} (um número limitado)
 - Seja t o triângulo intersectado de T_{k-1}
 - Então, t intersecta um número limitado de triângulos de T_{k-2} , que são por sua vez testados contra q
 - O algoritmo prossegue até encontrar o triângulo procurado em T_0
- Como em cada triangulação fazemos um número constante de testes e o número de triangulações é logarítmico, o algoritmo é $O(\log n)$



Algoritmo de Busca



Algoritmo p/ montar o conjunto independente

- Marca-se todos os vértices de grau ≥ 9
- Enquanto existir um vértice não marcado
 - Escolher um vértice não marcado v
 - Adicionar v ao conjunto independente
 - Marcar v e os vértices adjacentes a v
- Complexidade $O(n)$
 - Vértices não marcados são postos numa lista
 - Adjacência é determinada com uma DCEL ou equivalente



Prova do Lema – Grau médio

- Fórmula de Euler garante que em uma triangulação de um grafo planar

$$e = 3n - 6$$

- Somando os graus de todos os vértices, cada aresta é contada 2 vezes

$$\sum_v \text{grau}(v) = 2e = 6n - 12 < 6n$$

- Portanto o grau médio de uma triangulação planar é 6



Prova do Lema – vértices c/ grau ≤ 8

- Triangulação tem ao menos $n / 2$ vértices com grau ≤ 8
- **Prova:** suponha o contrário
 - $n / 2$ vértices têm grau 9 ou maior
 - Os demais vértices têm no mínimo grau 3
 - Portanto a soma de todos os graus é

$$9 \frac{n}{2} + 3 \frac{n}{2} = 6n$$

- ... o que contraria a suposição



Prova do Lema

- Quando o algoritmo começa, ao menos $n / 2$ vértices não estão marcados
- Ao escolhermos um vértice v (não marcado) este é marcado assim como os vértices adjacentes a v
 - 9 vértices marcados no máximo
- Logo, podemos escolher no mínimo $(n / 2) / 9 = n / 18$ vértices não marcados

