

# Processamento de Imagens usando Grafos

Prof. Alexandre Xavier Falcão

Segundo semestre de 2004

## 1 Segmentação por poda de árvores

Os paradigmas de (i) detecção de objetos  $\kappa$ -conexos com um conjunto de sementes internas e de (ii) segmentação por partição ótima baseada em um conjunto de sementes possuem as seguintes analogias.

Em ambos os casos, uma imagem de diferenças deve ser criada de tal forma que as dissimilaridades intra-objetos sejam menores do que as dissimilaridades inter-objetos (objeto-fundo). Isto pode ser feito explicitamente, com uma imagem do tipo gradiente, ou implicitamente com uma função de custos adequada. A função  $f_{\max}$  tem um papel importante em ambos os casos, pois penaliza todos os caminhos que cruzam as bordas, enquanto permite caminhos tão longos quanto possível em regiões de baixas dissimilaridades. Portanto, a imagem de diferenças pode ser vista como um relevo, onde o objeto é representado por um vale e sua borda por uma cadeia de montanhas.

Na abordagem (i), criamos buracos nas sementes internas e afundamos o relevo em água, sem deixar que esta transborde para o exterior do objeto. Estamos supondo que o objeto será definido pela extensão do lago que se forma em seu interior e que o nível  $k$  da água será suficiente para cobrir todas as montanhas internas. Caso contrário, a segmentação falha.

Na abordagem (ii), temos sementes internas e externas, e afundamos o relevo inteiro sem deixar que as águas do objeto se misturem com as do fundo. Neste caso, não importa que existam montanhas internas (externas) mais altas do que as da borda, desde que nenhum caminho d'água proveniente do exterior (interior) as atinjam. Porém, se as sementes externas estiverem isoladas em vales de tal forma que a altura mínima da água para atingir pixels vizinhos externos da borda do objeto seja a mesma para sementes internas e externas (i.e. platôs na imagem de custos), a segmentação pode falhar (e.g. ocorrendo um vazamento de água interna para o exterior e/ou invasão de água externa para o interior). Isto porque estes pixels devem ser divididos de alguma forma entre as sementes que os atingem (situação de *tie-breaking*, que pede por uma definição mais exata do que a simples descrição do processo de inundação com barreiras).

Observe que, em ambos os casos, o ponto crítico é a **altura mínima** da cadeia de montanhas que separa objeto e fundo. Normalmente, a cadeia tem altura irregular e são **poucos os pixels com altura mínima**. Se as sementes com buracos forem colocadas apenas no interior do objeto e o relevo for completamente afundado em água, a água vazará para o fundo pelos

pixels de borda com altura mínima (*leaking pixels*). Este novo paradigma é diferente dos outros dois, pois assume que os demais pixels do fundo serão atingíveis por caminhos que passam por esses poucos pixels de vazamento. Isto é verdade sempre que a altura do relevo nos arredores do objeto for menor ou igual a altura mínima da borda. Mesmo que a altura do relevo nos arredores do objeto seja maior, o importante é que esta altura seja menor do que a altura da borda adjacente, para evitar que a água transborde por partes mais altas antes que os arredores sejam completamente inundados pela água que passa pelos pixels de vazamento. Note que, mesmo nas situações onde as duas abordagens anteriores falham, esta terceira abordagem pode funcionar.

No contexto da IFT, a primeira abordagem reduz o problema a uma limiarização da imagem de custos e a segunda abordagem visa a partição da imagem de raízes. O terceiro paradigma explora a topologia do mapa de predecessores (floresta ótima). Sabendo que são poucos os pixels de vazamento e que os caminhos ótimos que atingem o fundo passam por esses pixels, o objetivo é analisar o mapa de predecessores para identificar os pixels de vazamento e podar suas subárvores de forma tal que a floresta resultante defina o objeto. **Uma observação importante é que como todo caminho ótimo tem prefixo ótimo, todos os caminhos ótimos que atingem o fundo passando por um mesmo pixel de vazamento terão o mesmo prefixo até o início do vazamento.** Esta propriedade pode ser usada para encontrar os pixels de vazamento.

Os caminhos em  $\hat{P}$  podem ser visualizados superpondo o **número de descendentes** de cada pixel (com cor avermelhada) e o brilho da imagem original. Vamos observar que caminhos que atingem o fundo terão tonalidade de vermelho mais forte do que caminhos com término no objeto. No caso da política FIFO, os caminhos de vazamento normalmente sofrem uma **queda abrupta** no número de descendentes ao passar para o fundo. Isto vai ocorrer sempre que os custos dos caminhos ótimos que atingem os pixels externos nos arredores do pixel de vazamento forem os mesmos, visto que a política de desempate FIFO se degenera em busca em largura. Uma falha pode ocorrer, por exemplo, quando a borda do objeto está muito próxima de uma borda externa, fazendo com que a busca em largura se inicie apenas mais adiante, fora do objeto. Os caminhos de vazamento podem ser identificados também pela limiarização no número de descendentes de cada pixel em  $\hat{P}$ , e os pixels de vazamento podem ser localizados pela derivada segunda ao longo desses caminhos. A política LIFO não gera a queda abrupta do número de descendentes, mas reduz o número de caminhos de vazamento tornando-os mais evidentes.

Após executar uma IFT com  $f_{\max}$ , adjacência  $A$  e sementes  $S$  no interior do objeto. O número de descendentes de cada pixel em  $\hat{P}$  pode ser calculado em tempo linear, conforme algoritmo a seguir.

### Algoritmo linear para contagem de descendentes:

Entrada: Imagem  $\hat{P} = (D_I, P)$  de predecessores e adjacência  $A$ .

Saída: Imagem  $\hat{D} = (D_I, D)$  do número de descendentes. Inicialmente  $D(p) = 0$  para todo  $p \in D_I$ .

1. Para todo pixel  $p$  em  $D_I$  faça
2. Se  $P(p) = nil$  então  $NSons(\hat{P}, \hat{D}, A, p)$ .

**Algoritmo**  $NSons(\hat{P}, \hat{D}, A, p)$ :

Entrada: Imagem  $\hat{P} = (D_I, P)$  de predecessores, imagem  $\hat{D} = (D_I, D)$  do número de descendentes, adjacência  $A$ , e pixel  $p$ .

Saída: Número de descendentes do pixel  $p$ .

1. Para todo pixel  $q \in A(p)$  faça
2. Se  $P(q) = p$ , então  $D(p) \leftarrow D(p) + NSons(\hat{P}, \hat{D}, A, q) + 1$ .
3. Retorne  $D(p)$ .

Este algoritmo utiliza busca em profundidade (recursão) para contar o número de nós de cada árvore da floresta. Durante o processo, o número de descendentes de cada nó é gerado em uma imagem  $\hat{D}$ . A informação em  $\hat{D}$  é processada para encontrar os caminhos de vazamento e os pixels de poda em  $\hat{P}$  (aqueles que devem ser removidos da floresta junto com suas subárvores). A remoção dessas subárvores é trivial, portanto o restante do algoritmo fica como exercício.

## 2 Exercícios

1. Elabore diferentes mecanismos de identificar caminhos de vazamento e localizar os pontos de poda.
2. Implemente e teste o paradigma de segmentação por poda usando  $f_{\max}$  e os mecanismos que você inventou na questão anterior.
3. Prove que os caminhos que atingem o fundo passando por um dado pixel de vazamento têm prefixo comum com término neste pixel.