

Processamento de Imagens usando Grafos

Prof. Alexandre Xavier Falcão

Aula 05

1 Transformada Imagem-Floresta Diferencial

Sequências de IFTs para um mesma relação de adjacência e uma mesma função de conectividade, mas conjuntos sementes S_t que variam de acordo com a iteração $t = 1, 2, \dots, n$, podem ser calculadas de forma diferencial. Isto é, visitando apenas os nós em D_I que requerem mudança no caminho ótimo. Em segmentação interativa, por exemplo, esta estratégia é fundamental para reduzir o tempo de resposta a correções de segmentação feitas pelo usuário. O método também tem sido explorado no projeto de *hardware* para operadores de imagem baseados na IFT e no cálculo paralelo da IFT em máquinas com múltiplos processadores e em sistemas distribuídos.

A Figura 1a ilustra um exemplo em segmentação interativa, onde o marcador A indica sementes de objeto e os marcadores B e C indicam sementes de fundo. As cores branco e preto indicam diferentes rótulos $\lambda(p)$ associados aos pixels p desses marcadores. O objeto deve ser definido pela floresta de caminhos ótimos com raízes em marcadores de objeto. A floresta de A , no entanto, invade o fundo como indica a seta. A Figura 1b mostra que a segmentação pode ser corrigida através da adição de um marcador D no fundo. Na verdade, o marcador C não é necessário e, portanto, sua floresta pode ser removida na mesma iteração. A segmentação da segunda iteração envolverá competição entre pixels sementes de D e os **pixels de fronteira** das florestas dos marcadores A e B (i.e., pixels com nós adjacentes na floresta de C que está sendo removida). A correção é diferencial, pois apenas os pixels na região de vazamento e na zona de influência de C são processados (Figure 1c).

A regra de inicialização $f(\langle q \rangle)$ deve ser finita para sementes em \mathcal{S} , o qual muda a cada iteração com a adição de novos marcadores. Os demais pixels devem ser inicializados com valor infinito na primeira iteração e, nas iterações subsequentes, apenas quando eles pertencem a florestas removidas. Os pixels de fronteira, portanto, mantêm os valores $V(q)$ da iteração anterior.

O algoritmo diferencial é apresentado abaixo, onde F é o conjunto de pixels de fronteira e M_t é o conjunto de pixels que pertencem a árvores marcadas para remoção na iteração t . Note que o ideal é selecionar um pixel p na imagem e remover a floresta do marcador que contém a raiz de p . Isso pode ser feito no algoritmo abaixo, inserindo os pixels deste marcador em M_t .

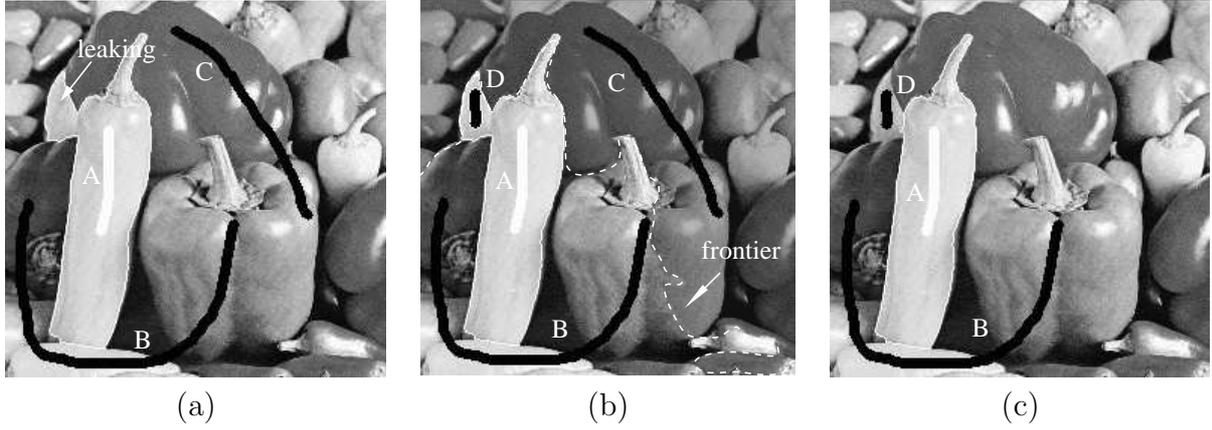


Figura 1: Segmentação por IFT diferencial. (a) Marcadores são selecionados dentro (A) e fora (B e C) do objeto. A segmentação resultante mostra o vazamento da floresta de A (seta). (b) Um marcador de fundo (D) é selecionado para correção da segmentação e o marcador C é selecionado para remoção. (c) Segmentação resultante da competição entre D e os pixels de fronteira (linhas tracejadas) das florestas enraizadas em A e B, os quais têm adjacentes na floresta de C.

Algoritmo IFT-diferencial:

Entrada: Imagem $\hat{I} = (D_I, I)$, adjacência A , função f de conectividade, imagens $\hat{V} = (D_I, V)$, $\hat{P} = (D_I, P)$ e $\hat{R} = (D_I, R)$ (para iteração $t = 1$, $V(q) = +\infty$, $R(q) = q$ e $P(q) = nil$), e conjuntos S_t e M_t .

Saída: Imagens $\hat{V} = (D_I, V)$ e $\hat{P} = (D_I, P)$ para o instante t .

Auxiliares: Fila Q de prioridades com política FIFO, variável tmp , arranjo de cor para indicar o status do nó em relação a Q e conjunto F de pixels de fronteira.

1. Faça $(\hat{V}, \hat{P}, F) \leftarrow RemoveArvores(\hat{V}, \hat{P}, \hat{R}, A, M_t)$.
2. Enquanto $F \neq \emptyset$ faça
3. Remova um pixel q de F .
4. Se $q \notin S_t$, então insira q em Q e faça $cor(q) \leftarrow cinza$.
5. Enquanto $S_t \neq \emptyset$ faça
6. Remova um pixel q de S_t .
7. Se $f(\langle q \rangle) < V(q)$, então $V(q) \leftarrow f(\langle q \rangle)$, $P(q) \leftarrow nil$, $R(q) \leftarrow q$, $cor(q) \leftarrow cinza$ e insira q em Q .
8. Enquanto $Q \neq \emptyset$ faça

9. Remova um pixel p de Q cujo valor $V(p)$ é mínimo e faça $cor(q) \leftarrow preto$.
10. Para todo $q \in A(p)$, tal que $cor(q) \neq preto$, faça
11. $tmp \leftarrow f(P^*(p) \cdot \langle p, q \rangle)$.
12. Se $tmp < V(q)$ ou $P(q) = p$ faça
13. Se $cor(q) = cinza$, remova q de Q .
14. $V(q) \leftarrow tmp$, $P(q) \leftarrow p$, $R(q) \leftarrow R(p)$, $cor(q) \leftarrow cinza$ e insira q em Q .

Algoritmo *RemoveArvores*:

Entrada: Imagens $\hat{V} = (D_I, V)$, $\hat{P} = (D_I, P)$ e $\hat{R} = (D_I, R)$, adjacência A , e conjunto M_t .

Saída: Imagens $\hat{V} = (D_I, V)$ e $\hat{P} = (D_I, P)$ atualizadas após remoção de árvores, e conjunto F de pixels de fronteira.

Auxiliares: Fila FIFO T e conjunto M de raízes de árvores marcadas para remoção.

1. Enquanto $M_t \neq \emptyset$ faça
2. Remova um pixel q de M_t .
3. Faça $r \leftarrow R(q)$.
4. Se $V(r) \neq +\infty$ então insira r em T e em M , e faça $V(r) \leftarrow +\infty$ e $P(r) \leftarrow nil$.
5. Enquanto $T \neq \emptyset$ faça
6. Remova um pixel p de T .
7. Para todo $q \in A(p)$ faça
8. Se $P(q) = p$ então $V(q) \leftarrow +\infty$ e $P(q) \leftarrow nil$, e insira q em T .
9. Se não, mas se $R(q) \notin M$, então insira q em F .

O algoritmo *RemoveArvore* reinicializa os mapas V e P , percorrendo em largura cada árvore removida da floresta, e retorna em F os pixels de fronteira.

A principal diferença entre o algoritmo IFT-diferencial e o algoritmo original é o teste do predecessor, $P(q) = p$, na linha 12. Este teste é executado positivamente, apenas quando $V(q) = tmp$. Isto garante que, quando uma semente s é adicionada, todos os pixels q alcançáveis a partir de s por caminhos ótimos $\pi = P^*(p) \cdot \langle p, q \rangle$ com $f(\pi) \leq V(q)$ serão reavaliados (porém, quando $f(\pi) = V(q)$, as linhas 13 e 14 só serão executadas se $P(q) = p$). Sem este teste, o mapa de raízes pode ficar desatualizado. Isto é, a nova semente pode conquistar uma região inteira a menos de uma ilha cujos pixels ainda apontarão para a raiz velha. Uma questão interessante é que provavelmente os caminhos ótimos não tenham os seus sufixos a partir de q alterados. Isto tornaria o teste do predecessor desnecessário se a imagem de raízes não fosse usada. Porém, as linhas 3 e 9 do algoritmo *RemoveArvore* usam a informação de raiz, a qual teria que ser calculada a partir do mapa P .

2 Tarefa

Utilize o filtro de suavização da primeira tarefa com três níveis de filtragem nas bandas Y, Cb e Cr (ou Lab) de imagens coloridas para constituir um vetor de características $\vec{F}(p) = (F_1(p), F_2(p), \dots, F_{12}(p))$ para cada pixel p , onde o valor original de cada banda e os três valores após a filtragem formam os valores $F_i(p)$, $i = 1, 2, \dots, 12$. Considere um grafo imagem usando adjacência circular de raio ρ e a função de conexidade f_{\max} ,

$$f_{\max}(\langle q \rangle) = \begin{cases} 0 & \text{se } q \in S \\ +\infty & \text{no caso contrário} \end{cases} \quad (1)$$

$$f_{\max}(\pi_p \cdot \langle p, q \rangle) = \max\{f_{\max}(\pi_p), \|\vec{F}(q) - \vec{F}(p)\|\}, \quad (2)$$

onde S é um conjunto de pixels (x, y) sementes, as quais são selecionadas dentro e fora de um objeto desejado. Faça um programa para ler uma imagem colorida, um arquivo texto com os pixels sementes internos e externos, e o valor de ρ . Seu programa deve transformar a imagem para o espaço YCbCr (ou Lab) e calcular a IFT para f_{\max} usando os algoritmos com políticas de desempate FIFO e LIFO. O objeto deve ser extraído da imagem \hat{L} de rótulos, que deve ser gravada na saída do programa. Avalie o seu programa com diferentes valores de ρ e sementes S . Responda qual dessas políticas é mais adequada para segmentação de imagens.

Outra opção nesta tarefa é usar o algoritmo diferencial e ir acrescentando as sementes à medida que são necessárias. Por exemplo, você pode iniciar com uma semente dentro e a borda da imagem como marcador externo; visualizar a imagem de saída; e decidir as novas sementes internas ou externas que vão corrigir a segmentação. O programa fica em laço até o usuário indicar sementes $(x, y) = (-1, -1)$. Neste caso, você deve ainda escrever o algoritmo da IFT diferencial com política de desempate LIFO. Portanto, esta opção valerá 2 vezes mais pontos do que a primeira.