Processamento de Imagens usando Grafos

Prof. Alexandre Xavier Falcão

Segundo semestre de 2004

1 Transformada Imagem-Floresta Diferencial

Em algumas operações de processamento de imagens, tais como a segmentação, é desejável calcular seqüências de IFTs, mantendo fixas a função de custo e a relação de adjacência, para instâncias subsequentes de conjuntos sementes S_t, $t = 1, 2, \ldots, n$. Mais ainda, podemos remover árvores marcadas por pixels em conjuntos M_t, $t = 1, 2, \ldots, n$, a cada instante t.

Podemos então imaginar um processo de segmentação, por exemplo, onde a cada instante t os conjuntos S_t e M_t são informados, e uma IFT é calculada de forma diferencial com relação à floresta no instante $t-1$. As sementes em S_t vão conquistar regiões de pixels mais conexos com essas sementes do que com as raízes da floresta em $t-1$. As árvores com pixels em M_t serão removidas e suas regiões disponibilizadas para a nova conquista por parte das raízes restantes e das sementes em S_t. Essas raízes são representadas por pixels das árvores não removidas que são adjacentes a pixels das árvores removidas. Estes pixels são denotados **pixels de fronteira**. O processo pára quando a floresta resultante satisfaz a segmentação desejada.

Este processo é válido para funções suaves f de custo e é ilustrado abaixo usando política de desempate FIFO. Adjacências simétricas são normalmente desejáveis, para garantir que todos os pixels com custo infinito serão visitados a cada iteração.

Algoritmo IFT-diferencial:

Entrada: Imagem $\hat{I} = (D_I, I)$, adjacência A, função f de custo de caminho, imagem $\hat{C} = (D_I, C)$ de custos, imagem $\hat{P} = (D_I, P)$ de predecessores, imagem $\hat{R} = (D_I, R)$ de raízes, e conjuntos S_t e M_t. Na primeira vez, $C(p) = +\infty$ e $P(p) = nil$ para todo $p \in D_I$.
Saída: Imagem $\hat{C} = (D_I, C)$ de custos, imagem $\hat{P} = (D_I, P)$ de predecessores, e imagem $\hat{R} = (D_I, R)$ de raízes atualizadas para o instante t.
Auxiliares: Fila Q de prioridades com política FIFO, variável tmp, e conjunto F de pixels de fronteira.

1. Faça $(\hat{C}, \hat{P}, F) \leftarrow RemoveArvores(\hat{C}, \hat{P}, \hat{R}, A, M_t)$.
2. Enquanto $F \neq \emptyset$ faça
3. Remova um pixel q de F.

4. Se $q \notin S_t$, então insira q em Q.

5. Enquanto $S_t \neq \emptyset$ faça

6. Remova um pixel q de S_t.

7. Se $f(\langle q \rangle) < C(q)$, então $C(q) \leftarrow f(\langle q \rangle)$, $P(q) \leftarrow nil$, $R(q) \leftarrow q$, e insira q em Q.

8. Enquanto $Q \neq \emptyset$ faça

9. Remova um pixel p de Q cujo custo $C(p)$ é mínimo.

10. Para todo $q \in A(p)$, tal que $C(q) \geq C(p)$, faça

11. $tmp \leftarrow f(P^*(p) \cdot \langle p, q \rangle)$.

12. Se $tmp < C(q)$ ou $P(q) = p$ faça

13. Se $q \in Q$, remova q de Q.

14. $C(q) \leftarrow tmp$, $P(q) \leftarrow p$, $R(q) \leftarrow R(p)$, e insira q em Q.

Algoritmo $RemoveArvores$**:**

Entrada: Imagem $\hat{C} = (D_I, C)$ de custos, imagem $\hat{P} = (D_I, P)$ de predecessores, imagem $\hat{R} = (D_I, R)$ de raízes, adjacência A, e conjunto M_t.
Saída: Imagem $\hat{C} = (D_I, C)$ de custos e imagem $\hat{P} = (D_I, P)$ de predecessores 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 $C(r) \neq +\infty$ então insira r em T e em M, e faça $C(r) \leftarrow +\infty$ 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 $C(q) \leftarrow +\infty$ 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 C 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 $C(q) = tmp$. 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 custo $f(\pi) \leq C(q)$ serão reavaliados (porém, quando $f(\pi) = C(q)$, 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.

Para segmentação, podemos associar uma função $f_{\max}$ (ou variações discutidas nas aulas anteriores) e rótulos distintos para os objetos (incluindo o fundo). Após a IFT, esses rótulos são associados aos pixels da imagem de raízes, trocando-se a raiz pelo rótulo da semente correspondente.

2 Exercícios

1. Prove que o teste do predecessor é desnecessário para garantir a corretude da floresta $\hat{P}$.

2. Implemente e teste o algoritmo acima com as variações da função f_{max} vistas nas aulas anteriores. Verifique o ganho de eficiência em relação ao algoritmo original da IFT.

3. Suponha que desejamos separar um objeto do resto da imagem. Elabore um critério global de avaliação da floresta ótima, um critério de seleção de novas sementes S_t, e um critério de remoção de árvores (escolha de M_t) para que esta segmentação se complete em um número finito de iterações.