Processamento de Imagens usando Grafos

Prof. Alexandre Xavier Falcão

Segundo semestre de 2004

1 Perseguição de bordas

Uma borda de objeto em uma imagem é caraterizada por uma discontinuidade de propriedades da imagem, internas e externas ao objeto em uma vizinhança da borda. Portanto, uma função de custo de caminho para perseguição de bordas deve capturar esta discontinuidade associando **custos baixos** para pixels adjacentes sobre a borda desejada, e custos altos no caso contrário. Dados dois pixels, o como origem e d como destino, sobre a borda, o custo do caminho ótimo de o a d deve ser um segmento de borda. Na prática, porém, uma borda consiste de vários segmentos ótimos formando um contorno fechado.

Genericamente, podemos considerar k conjuntos S_i, $i = 1, 2, \ldots, k$, de pixels sementes selecionados sobre uma borda, tais que $S_1 = S_k$ possui um único pixel e os pixels em S_i, $i = 2, 3, \ldots, k-1$, pertencem a uma região pequena de incerteza (e.g. uma linha que cruza a borda, uma marca circular sobre a borda). Um segmento ótimo de borda de S_i para S_{i+1}, $i = 1, 2, \ldots, k-1$, pode ser formado por n caminhos de custo mínimo que atingem todos os pixels de S_{i+1}. Neste caso, teríamos um segmento ótimo de **borda *fuzzy***. Para simplificar, vamos considerar apenas um único caminho ótimo 4- ou 8-conexo entre cada par S_i e S_{i+1}.

Portanto, uma borda é um **contorno ótimo** que passa pela seqüência de conjuntos S_i, $i = 1, 2, \ldots, k$, de pixels sementes. Isto requer $k - 1$ IFTs para calcular os caminhos de S_i a S_{i+1}, $i = 1, 2, \ldots, k-1$.

A **escolha da função de custo** deve levar em conta dois aspectos relevantes:

1. Bordas próximas com o mesmo grau de discontinuidade devem ser evitadas com pré-processamento ou explorando propriedades, tais como a **orientação** da discontinuidade. Por exemplo, custos baixos podem ser atribuídos para arcos de borda no grafo, cujo o lado esquerdo tem brilho menor/maior que o brilho do lado direito.

2. Na ausência de informação de borda, custos altos podem ser atribuídos a alguns arcos da borda, e portanto, a função de custo deve ser **robusta** com relação a estes casos. Por exemplo, a função $f_{\max}$ **não é adequada**, pois basta um arco com custo máximo para que o conjunto de pixels de destino possa ser atingido por diversos caminhos ótimos, passando por outras regiões da imagem longe da borda.

Esta reflexão nos leva a funções de custo do tipo f_{sum} com política FIFO, tais como f_{ctrack} abaixo:

$$\begin{aligned} f_{ctrack}(\langle q \rangle) &= h(q) \\ f_{ctrack}(\pi \cdot \langle p, q \rangle) &= f_{ctrack}(\pi) + (K - \max\{G(p,q) \cdot \eta(p,q), 0\}), \end{aligned} \tag{1}$$

onde $h(q) = 0$, se $q \in S_1$ na primeira iteração, ou $h(q)$ é o custo final do pixel q na iteração anterior, se $q \in S_i$ e $i > 1$, ou $h(q) = +\infty$ no caso contrário, independente da iteração; $G(p,q)$ é um vetor gradiente estimado no ponto médio do arco (p,q); $\eta(p,q)$ é o arco (p,q) rotacionado de 90 graus no sentido anti-horário; e K é um limite superior para $|G(p,q) \cdot \eta(p,q)|$. O vetor $G(p,q)$ deve ser tal que arcos sobre a borda orientada do objeto possuam valores baixos de custo e os demais arcos fiquem com valores altos de custo.

Note que a cada iteração, o algoritmo pode parar o cálculo da IFT quando o último pixel de S_{i+1} sai da fila Q. O contorno final pode ser obtido das respectivas $k-1$ imagens de predecessores $\hat{P}_{k-1}, \ldots, \hat{P}_1$, mas também é possível modificar o algoritmo da IFT para usar um único conjunto de imagens de custos e de predecessores. Antes de iniciar uma iteração i, os segmentos ótimos que atingiram os conjuntos S_j, $j \leq i$, nas iterações anteriores são candidatos a fazerem parte do contorno ótimo final. Portanto, estes segmentos devem permanecer com os custos e os predecessores calculados, enquanto os demais pixels da imagem devem ter seus custos reinicializados para $+\infty$. Um cuidado especial, porém, deve ser tomado na última iteração $k-1$, pois o pixel em S_1 também deve ter seu custo reinicializado para $+\infty$ e o algoritmo pára quando ele sai da fila Q. O contorno ótimo é obtido percorrendo os predecessores deste pixel até encontrá-lo novamente (veja outros variantes na aula 23 do curso MO443).

2 Exercício

Escreva um algoritmo para obter um contorno ótimo que passa pelos conjuntos ordenados de pixels sementes S_i, $i = 1, 2, \ldots, k-1$, onde S_1 tem um único pixel, usando a função f_{ctrack} acima. Use uma única imagem de predecessores e uma única imagem de custos.

Solução proposta por Fábio Cappabianco e Leonel Gayard.

Algoritmo para perseguição de bordas.

Entrada: Imagem $\hat{I} = (D_I, I)$, sementes $\{S_1, S_2, \ldots, S_{k-1}\}$ sobre a borda do objeto, onde S_1 possui um único pixel s_1, função de custos f_{ctrack}, e adjacência-8 A.
Saída: Imagem binária $\hat{B} = \{D_I, B\}$, tal que $B(p) = 1$ se p pertence à borda, e $B(p) = 0$ no caso contrário.
Auxiliar: Fila de prioridade Q com política FIFO, imagem de custos $\hat{C} = \{D_I, C\}$, imagem de predecessores $\hat{P} = \{D_I, P\}$, e variáveis tmp e n.

1. Para todo pixel $p \in D_I$ faça $B(p) \leftarrow 0$.
2. Faça $C(s_1) \leftarrow 0$ e $P(s_1) \leftarrow nil$.
3. Para $i \leftarrow 1$ até $i = k - 1$ faça
4.   Se $i = k - 1$ faça $C(s_1) \leftarrow \infty$.
5.   Para toda semente $s \in S_i$ faça
6. $PintaPrefixo(\hat{B}, \hat{P}, s)$ e insira s em Q.
7.   Para todo pixel $p \in D_I$, tal que $B(p) = 0$ faça $C(p) \leftarrow +\infty$ e $P(p) \leftarrow nil$.
8.   Faça $n \leftarrow |S_{i\%(k-1)+1}|$.
9.   Enquanto $n > 0$ faça
10.   Remova um pixel p de Q cujo custo $C(p)$ é mínimo.
11.   Se $p \in S_{i\%(k-1)+1}$, então $n \leftarrow n - 1$.
12.   Para todo $q \in A(p)$ tal que $C(p) < C(q)$ faça
13. $tmp \leftarrow f_{ctrack}(P^*(p) \cdot \langle p, q \rangle)$.
14.   Se $tmp < C(q)$, então
15.   Se $C(q) \neq +\infty$, remova q de Q.
16. $C(q) \leftarrow tmp$, $P(q) \leftarrow p$, e insira q em Q.
17.   Esvazie Q.
18. $PintaPrefixo(\hat{B}, \hat{P}, s_1)$.

Obs. $i\%(k-1)$ significa o resto da divisão inteira de i por $k-1$, e $|S_i|$ significa a cardinalidade do conjunto S_i.

Função $PintaPrefixo(\hat{B}, \hat{P}, p)$**:**

Entrada: Imagens $\hat{B} = (D_I, B)$ e $\hat{P} = (D_I, P)$, e último pixel p do prefixo.
Saída: Imagem $\hat{B} = \{D_I, B\}$ com prefixo pintado.

1. Se $B(p) = 0$, então
2. $\quad B(p) \leftarrow 1$.
3. $\quad$ Se $P(p) \neq nil$, então $PintaPrefixo(\hat{B}, \hat{P}, P(p))$.