Introdução ao Processamento de Imagem Digital (MO443/MC920)

Prof. Alexandre Xavier Falcão

Material de 2005

1 Transformada Imagem-Floresta

Na aula de Introdução à Topologia Digital (aula 4) vimos que uma relação de adjacência A entre pixels define um grafo na imagem, onde os pixels são os nós, $(p, q) \in A$ é uma aresta entre pixels adjacentes e um pixel q é conexo a um pixel p se existir um caminho de p a q composto de pixels adjacentes no grafo. A Transformada Imagem-Floresta (IFT de *Image Foresting Transform*) explora esta representação para reduzir problemas de processamento de imagens baseados em conectividade em um problema de floresta de caminhos de custo mínimo (caminhos ótimos).

O custo de um caminho é calculado por uma função dependente da aplicação e com base em propriedades locais da imagem— tais como brilho, gradiente, e posição de pixel ao longo do caminho. Para uma função de custo adequada, relação de adjacência irreflexiva e um dado conjunto de pixels sementes, a IFT associa a cada pixel da imagem um caminho de custo mínimo, particionando a imagem em uma floresta de caminhos ótimos onde cada árvore tem como raiz um pixel semente e como nós os pixels da imagem mais "conexos" com a raiz do que com qualquer outra semente, em algum sentido apropriado. A IFT gera como resultado uma imagem anotada, onde cada pixel tem associado o predecessor no caminho ótimo, o custo deste caminho e o pixel raiz (ou algum rótulo associado à raiz).

Exemplos de problemas redutíveis a uma IFT são segmentação por transformada de watershed, segmentação baseada em conectividade *fuzzy*, filtragem e segmentação por reconstrução morfológica, segmentação por crescimento de regiões, segmentação por perseguição de borda, cálculo de caminhos geodésicos, transformadas de distância, representação por esqueletos multiescala, estimação de pontos de saliência em curvas, e cálculo da dimensão fractal multiescala de uma curva.

Observe que alguns casos não são facilmente relacionados com um problema de partição da imagem (e.g. reconstrução morfológica, perseguição de bordas, pontos de saliência). Porém, na maioria dos casos, a solução é obtida pela simples escolha de parâmetros da IFT seguida de um processamento local da imagem anotada, em tempo proporcional ao número de pixels. Este resultado é obtido com uma extensão do algoritmo de Dijkstra para múltiplas fontes e função de custo de caminho mais geral.

1.1 Exemplo simples

Suponha, por exemplo, a segmentação de um objeto em uma imagem 2D em tons de cinza, onde um pixel semente o é selecionado no interior do objeto e outro pixel semente o' é selecionado fora. Supondo que o objeto tem distribuição homogênea de brilho diferente do exterior, a função de custo de caminho pode ser o valor máximo das diferenças absolutas entre os valores dos pixels adjacentes ao longo do caminho. Podemos definir o valor de conectividade de um pixel p com relação ao objeto como o custo do caminho ótimo de o a p e com relação ao fundo como o custo do caminho ótimo de o' a p. Um pixel da imagem será classificado como pertencente ao objeto se sua conectividade com o objeto for maior do que sua conectividade com o fundo. Se a segmentação funcionar, a árvore de caminhos ótimos com raiz o representará o objeto desejado.

1.2 Funções de custo de caminho

Cada problema requer a escolha de uma função f, a qual associa um custo, que provêm de um conjunto V ordenado de valores com valor máximo denotado por $+\infty$, para cada caminho. Para garantir uma floresta de caminhos ótimos, esta função deve ser suave. Isto é, para todo pixel $q \in D_I$, onde D_I é o conjunto dos pixels de uma imagem $\hat{I}$, deve existir um caminho ótimo π terminando em q que deve satisfazer as condições estabelecidas em artigo submetido recentemente para a IEEE TPAMI
(http://www.math.wvu.edu/ kcies/SubmittedPapers/SS17DijkstraCharacterization.pdf).
O exemplo mais comum é a função de custo aditiva, a qual satisfaz

$$\begin{aligned} f_{sum}(\langle q \rangle) &= h(q), \\ f_{sum}(\pi \cdot \langle p, q \rangle) &= f_{sum}(\pi) + w(p, q), \end{aligned} \tag{1}$$

onde $(p, q) \in A$, π é qualquer caminho terminando em p, $h(q)$ é um custo inicial (*handicap cost*) fixo para qualquer caminho iniciando em q, e $w(p, q)$ é um peso não negativo associado ao arco (p, q).

Um outro exemplo é a função de custo $f_{\max}$, a qual satisfaz

$$\begin{aligned} f_{\max}(\langle q \rangle) &= h(q), \\ f_{\max}(\pi \cdot \langle p, q \rangle) &= \max\{f_{\max}(\pi), w(p, q)\}, \end{aligned} \tag{2}$$

onde $h(q)$ e $w(p, q)$ são fixos mas arbitrários.

De uma forma geral, os exemplos acima pertencem à classe de funções monotônicas-incrementais (MI), as quais satisfazem

$$\begin{aligned} f(\langle q \rangle) &= h(q), \\ f(\pi \cdot \langle p, q \rangle) &= f(\pi) \odot (p, q), \end{aligned} \tag{3}$$

onde $h(q)$ é arbitrário e $\odot : V \times A \to V$ é uma operação binária que satisfaz as condições

(M1) $x' \geq x \Rightarrow x' \odot (p, q) \geq x \odot (p, q)$, (M2) $x \odot (p,q) \geq x$,

para quaisquer $x, x' \in V$ e qualquer $(p,q) \in A$, onde $\odot$ depende apenas do custo de π e não de qualquer outra propriedade de π.

Alguns operadores requerem funções mais gerais que as funções MI. Este é o caso da função f_{euc} usada em problemas que envolvem a transformada de distância Euclideana, a qual satisfaz

$$f_{euc}(\pi \cdot \langle p,q \rangle) = d^2(org(\pi), q), \tag{4}$$

onde d é a distância Euclideana entre dois pixels, $org(\pi)$ é o pixel inicial do caminho π. A suavidade de f_{euc}, porém, vai depender da relação de adjacência A adotada.

1.3 Pixels sementes

O conjunto $S \subseteq D_I$ de pixels sementes restringe a busca por caminhos ótimos que iniciam em S. Isto equivale a modificar a função f de custo para f^S, a qual satisfaz

$$f^S(\pi) = \left\{ \begin{array}{ll} f(\pi), & \text{se } org(\pi) \in S, \\ +\infty, & \text{no caso contrário.} \end{array} \right\} \tag{5}$$

No caso particular de funções MI, isto também é equivalente a definir $h(q) = +\infty$ para pixels $q \notin S$. Note que se f for MI, então f^S será MI e portanto suave. Infelizmente, isto não é necessariamente verdade se f for suave, mas não for MI. Por exemplo, f^S_{euc} com A igual à vizinhança-4 é suave para $|S| \leq 2$, mas não é suave para $|S| \geq 3$.

Observe que todas as raízes da IFT são pixels sementes, mas nem todas sementes se transformam em raízes da IFT, pois o custo de um caminho trivial $\langle q \rangle$, onde $q \in S$, pode ser maior que o custo de um outro caminho iniciado em S com término em q.

1.4 Imagem anotada

Um mapa P de predecessores é uma função que associa a cada pixel $q \in D_I$ ou outro pixel $p \in D_I$, $(p,q) \in A$, ou uma marca $nil \notin D_I$. No segundo caso, q é dito ser uma raiz do mapa. Uma floresta espalhada é um mapa de predecessores que não contém ciclos— i.e., aquele que leva todo pixel para nil em um número finito de iterações. Para qualquer pixel $q \in D_I$, a floresta P define um caminho $P^*(q)$ recursivamente como $\langle q \rangle$, se $P(q) = nil$, e $P^*(p) \cdot \langle p,q \rangle$ se $P(q) = p \neq nil$.

A IFT calcula essencialmente uma floresta P de caminhos ótimos— i.e. uma floresta espalhada onde $P^*(q)$ tem custo mínimo para todo $q \in D_I$. Para fins de eficiência, a IFT também gera um mapa de custos C e um mapa de raízes L, onde $C(q)$ é o custo do caminho ótimo até q e $L(q)$ é o pixel inicial deste caminho (ou algum rótulo associado a ele).

2 Exercícios

1. Mostre que toda função MI é suave.

2. Mostre que f_{euc}^{S} com vizinhança-4 é suave para $|S| \leq 2$.
3. Mostre que f_{euc} não é MI.