

Processamento de Imagens usando Grafos

Prof. Alexandre Xavier Falcão

Aula 03

1 Funções de conexidade

Uma função de conexidade $f(\pi_p)$ associa um valor escalar a qualquer caminho no grafo $G = (D_I, A)$, com base em propriedades da imagem ao longo deste caminho. Em segmentação, por exemplo, propriedades locais da imagem (vetor de atributos) e globais do objeto desejado (textura, cor e forma) podem ser exploradas no cálculo de $f(\pi_p)$ para indicar a força de conexidade entre seus nós terminais através do caminho π_p .

Uma função de conexidade é especificada por uma regra de inicialização e uma regra de extensão do caminho. Exemplos são:

$$\begin{aligned} f_{\max}(\langle q \rangle) &= H(q) \\ f_{\max}(\pi_p \cdot \langle p, q \rangle) &= \max\{f_{\max}(\pi_p), w(p, q)\} \end{aligned} \quad (1)$$

$$\begin{aligned} f_{\min}(\langle q \rangle) &= H(q) \\ f_{\min}(\pi_p \cdot \langle p, q \rangle) &= \min\{f_{\min}(\pi_p), w(p, q)\} \end{aligned} \quad (2)$$

$$\begin{aligned} f_{\text{sum}}(\langle q \rangle) &= H(q) \\ f_{\text{sum}}(\pi_p \cdot \langle p, q \rangle) &= f_{\text{sum}}(\pi_p) + w(p, q) \end{aligned} \quad (3)$$

$$\begin{aligned} f_{\text{euc}}(\langle q \rangle) &= \begin{cases} 0 & \text{se } q \in S \\ +\infty & \text{otherwise} \end{cases} \\ f_{\text{euc}}(\pi_p \cdot \langle p, q \rangle) &= \|q - R(p)\|^2 \end{aligned} \quad (4)$$

onde $H(q)$ é um valor inicial, $R(p) = \text{org}(\pi_p)$, $w(p, q)$ é um peso de arco ($w(s, t) \geq 0$ em f_{sum}), e $S \subset D_I$ é um conjunto de sementes. Funções f_{\max} e f_{\min} são usadas para operações duais. A restrição de sementes em f_{euc} pode ser aplicada também à f_{\min} , f_{\max} , e f_{sum} por atribuir valores finitos $H(q)$ apenas para pixels $q \in S$.

Essas funções serão usadas em operações de filtragem de imagem, segmentação de imagem, representação e descrição de objetos, e classificação de padrões usando **caminhos ótimos** em $G = (D_I, A)$. Um caminho $\pi_p \in G$ é ótimo quando $f(\pi_p) \leq f(\tau_p)$ (ou $f(\pi_p) \geq f(\tau_p)$) para qualquer outro caminho τ_p independente de sua origem. Estamos, portanto, interessados em encontrar a função $V(q)$ que armazena o valor do caminho ótimo com término em cada pixel

q .

$$V(q) = \min_{\forall \pi_q \in (D_I, A)} \{f(\pi_q)\} \quad (5)$$

$$V(q) = \max_{\forall \pi_q \in (D_I, A)} \{f(\pi_q)\}. \quad (6)$$

Para funções f_{\max} , f_{sum} , e f_{euc} , nós buscamos minimizar $V(q)$, enquanto a maximização de $V(q)$ é feita para f_{\min} . As funções f_{\max} (f_{\min}) e f_{sum} são casos particulares de funções f_{mi} monotonicamente incrementais (decrementais).

$$\begin{aligned} f_{mi}(\langle q \rangle) &= H(q), \\ f_{mi}(\pi_p \cdot \langle p, q \rangle) &= f_{mi}(\pi_p) \odot (p, q), \end{aligned} \quad (7)$$

onde \odot é uma operação binária entre um caminho e um arco que satisfaz as condições

$$(M1) \quad \pi_p \geq \tau_p \Rightarrow \pi_p \odot (p, q) \geq \tau_p \odot (p, q),$$

$$(M2) \quad \pi_p \odot (p, q) \geq \pi_p,$$

para $\pi_p, \tau_p \in (D_I, A)$ e quaisquer arcos $(p, q) \in A$. Uma característica essencial deste modelo de função é que \odot depende apenas do valor de π_p , e não de qualquer outra propriedade deste caminho. Entretanto, a função f_{euc} é mais geral e depende de $R(p)$.

2 Transformada Imagem-Floresta

A transformada imagem-floresta (IFT - *Image Foresting Transform*) reduz problemas de processamento de imagem baseados em conexidade ao cálculo de uma **floresta de caminhos ótimos** P a partir de (D_I, A) e uma função de conexidade — i.e., $P(q)$ é uma função **sem ciclos** que associa a todo pixel $q \in D_I$ um valor $P(q) = \text{nil}$, se $\langle q \rangle$ for ótimo (**raiz da floresta**), ou um predecessor $P(q) = p$ no caminho ótimo $P^*(q)$ (o qual pode ser obtido percorrendo os predecessores até a raiz).

Na IFT, as raízes são os mínimos (máximos) locais da função V de conexidade ótima. Estas raízes competem entre elas e cada raiz conquista os nós mais fortemente conexos a ela do que a qualquer outra, particionando a imagem em uma floresta onde cada raiz forma uma árvore de caminhos ótimos (Figura 1).

A floresta de caminhos ótimos é uma partição ótima da imagem, que pode ser representada pela propagação das raízes da floresta aos nós das respectivas árvores (mapa $R(q)$ de raízes). Portanto, a IFT se aplica à segmentação baseada em regiões e classificação de padrões, que são problemas de partição de conjuntos. Ao resolver a partição ótima, a IFT também resolve o problema de encontrar uma função ótima $V(q)$ que associa a todo pixel $q \in D_I$, o valor de um caminho ótimo com término em q (Equações 5 e 6). Existem operadores de filtragem, representação e descrição de objetos que exploram este atributo. Os caminhos ótimos em $P(q)$ também têm sido usados em segmentação por contornos e por regiões (neste caso explorando propriedades combinatórias da floresta). Em resumo, a idéia da IFT é reduzir o problema ao processamento local de atributos da floresta resultante.

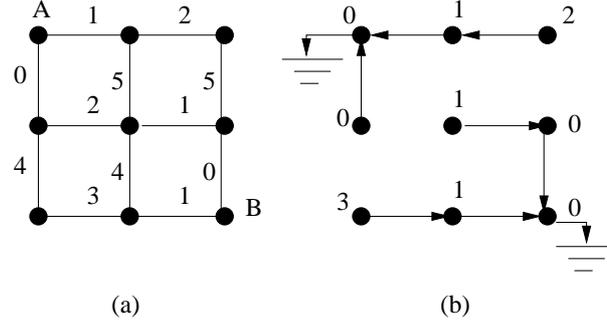


Figura 1: (a) Um grafo com adjacência-4 onde os números indicam pesos dos arcos e os nós A e B são sementes. (b) Uma floresta de caminhos ótimos para f_{\max} , quando $H(q) = 0$, se $q \in \{A, B\}$, e $H(q) = +\infty$ no caso contrário.

3 Algoritmo geral

A IFT essencialmente generaliza o algoritmo de Dijkstra para funções de conectividade, onde para qualquer nó $q \in D_I$, existe um caminho ótimo π_q que é ou trivial, ou tem a forma $\pi_p \cdot \langle p, q \rangle$ onde

(C1) $f(\pi_p) \leq f(\pi_q)$,

(C2) π_p é ótimo, e

(C3) para qualquer caminho ótimo τ_p , $f(\tau_p \cdot \langle p, q \rangle) = f(\pi_q)$.

Note que estas condições são aplicadas apenas a caminhos ótimos. O algoritmo é apresentado abaixo.

Algoritmo geral da IFT:

Entrada: Imagem $\hat{I} = (D_I, I)$, adjacência A , e função f de conectividade.

Saída: Imagens $\hat{V} = (D_I, V)$ de conectividade, $\hat{P} = (D_I, P)$ de predecessores, e $\hat{R} = (D_I, R)$ de raízes.

Auxiliares: Fila Q de prioridades e variável tmp .

1. Para todo pixel $q \in D_I$ faça
2. $V(q) \leftarrow f(\langle q \rangle)$, $P(q) \leftarrow nil$, e $R(q) \leftarrow q$.
3. Se $V(q) \neq +\infty$, insira q em Q .
4. Enquanto $Q \neq \emptyset$ faça
5. Remova um pixel p de Q cujo valor $V(p)$ é mínimo.
6. Para todo $q \in A(p)$, tal que $V(q) > V(p)$, faça

7. $tmp \leftarrow f(P^*(p) \cdot \langle p, q \rangle)$.
8. Se $tmp < V(q)$ faça
9. Se $V(q) \neq +\infty$, remova q de Q .
10. $V(q) \leftarrow tmp, P(q) \leftarrow p, R(q) \leftarrow R(p)$, e insira q em Q .

O algoritmo inicia com todos os caminhos triviais (linhas 1 e 2). Na linha 3, todos os candidatos a raízes (sementes) são inseridos na fila Q . As raízes são os mínimos de V , primeiramente detectados na linha 5. Pois, ao remover qualquer nó p de Q na linha 5, o caminho $P^*(p)$ é ótimo. As demais linhas calculam e propagam caminhos melhores para os nós adjacentes, seguindo uma ordem não-decrescente de conectividade e aplicando a seguinte regra de expansão: se $f(\pi_p \cdot \langle p, q \rangle) < f(\pi_q)$ (linha 8), então $\pi_q \leftarrow \pi_p \cdot \langle p, q \rangle$ (linha 10).

4 Exemplo em segmentação

A Figura 2 ilustra um exemplo onde a IFT é usada para segmentação de imagem por região. Marcadores rotulados S são colocados na imagem para indicar pixels de objeto e fundo. A função f_{\max} é escolhida com $H(q) = 0$, para $q \in S$, $H(q) = +\infty$ no caso contrário. O peso $w(p, q)$ das arestas é calculado de forma a ser o maior possível na borda do objeto desejado em relação aos pesos dentro e fora. Os marcadores de rótulos distintos competem entre si e o objeto é definido pela floresta de caminhos ótimos com raízes nos marcadores internos.

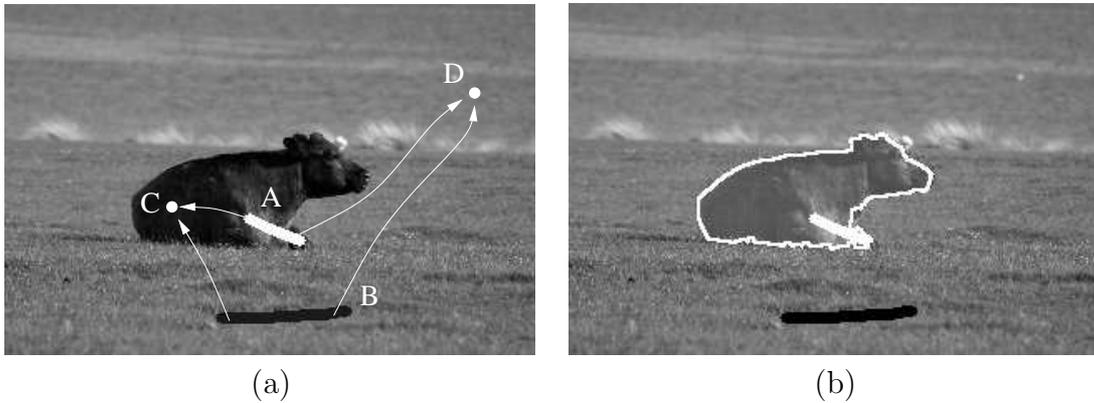


Figura 2: (a) Caminhos ótimos para pixels internos, como C , vem de A (marcador branco) e não de B (marcador preto). O oposto é válido com relação a pixels externos, como D . (b) A borda entre as florestas ótimas interna e externa.

A Figura 3 ilustra o processo ordenado de crescimento de regiões a partir das sementes em S para um objeto simples usando a mesma função f_{\max} e regras de inicialização.

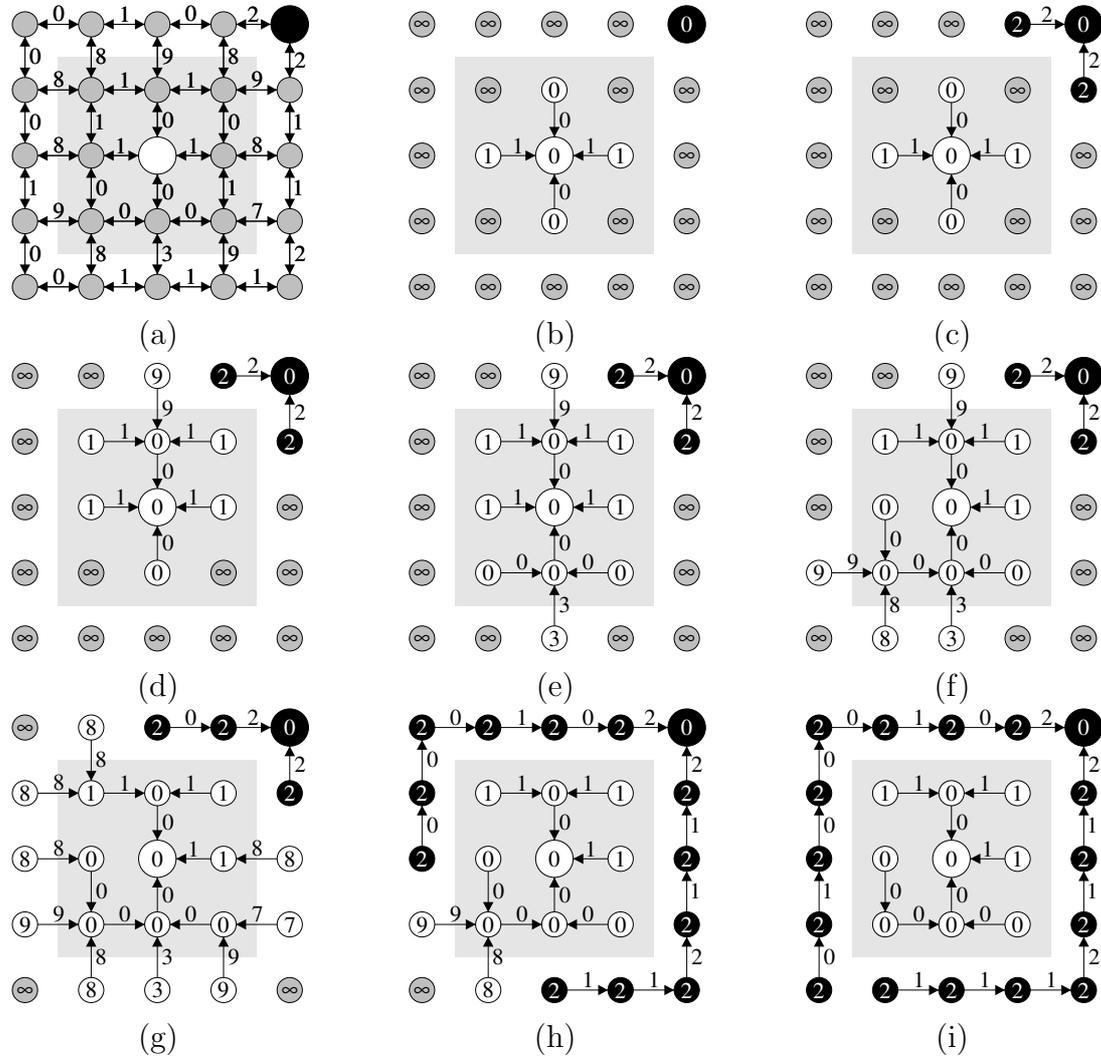


Figura 3: (a) Um grafo com adjacência-4 e sementes $S = \{(2, 2), (4, 0)\}$. (b)-(f) Floresta nas 5 primeiras iterações. Floresta após (g) 12, (h) 20, e (i) 25 iterações.