

Processamento de Imagens usando Grafos

Prof. Alexandre Xavier Falcão

Aula 07

1 Filtragem Conexa

Uma imagem cinza $\hat{I} = (D_I, I)$ pode ser vista como um relevo discreto onde a altura de cada ponto $p \in D_I$ é a intensidade $I(p)$. Este relevo possui domos, bacias e platôs (*flat zones* — componentes conexos de mesma intensidade e área máxima). A filtragem conexa simplifica a imagem por remover domos ou preencher bacias de forma a aumentar o número e/ou área dos platôs. Um operador ψ é dito conexo se e somente se qualquer par de pixels em um dado platô em \hat{I} pertencer a um mesmo platô em $\psi(\hat{I})$.

A principal vantagem é que a operação conexa não cria falsas bordas, apenas elimina bordas. Filtros conexos podem ser aplicados para simplificar uma imagem a ser segmentada, extrair atributos multi-escala da imagem para segmentação, como pós-processamento para remover componentes binários indesejados após a segmentação, e até mesmo como descritores de granulometria dos objetos.

Um platô é dito **mínimo regional (máximo regional)** se a intensidade dos pixels nos platôs vizinhos for estritamente maior (menor) que a intensidade no platô. Filtros conexos para preenchimento de bacias podem ser obtidos com a seleção de mínimos regionais em um conjunto S e a minimização do mapa de conectividade V usando a função f_{bas} .

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

$$f_{bas}(\pi_p \cdot \langle p, q \rangle) = \max\{f_{bas}(\pi_p), I(q)\}. \quad (2)$$

Esta função calcula em $V(q)$ a menor altura que um indivíduo em S deve subir para caminhar de S a q . O caminho ótimo $P^*(q)$ é obtido pela IFT, mas estamos interessados na imagem filtrada $\hat{V} = (D_I, V)$. Em 2D, a adjacência é normalmente 4, mas a idéia se estende para qualquer adjacência. A operação dual maximiza o mapa de conectividade V usando a função f_{dom} para S contendo pixels de máximos regionais selecionados.

$$f_{dom}(\langle q \rangle) = \begin{cases} I(q) & \text{se } q \in S \\ -\infty & \text{no caso contrário} \end{cases} \quad (3)$$

$$f_{dom}(\pi_p \cdot \langle p, q \rangle) = \min\{f_{dom}(\pi_p), I(q)\}. \quad (4)$$

Esta função calcula em $V(q)$ a maior altura que um indivíduo em S deve descer para caminhar de S a q .

O conjunto S pode ser, por exemplo, a borda da imagem. No caso de imagens de fundo escuro e objetos claros com bacias escuras, estas são preenchidas com brilho claro em f_{bas} . O efeito oposto é obtido com f_{dom} .

2 Filtragem por reconstrução morfológica

A reconstrução morfológica é uma operação conexa, monotônica e idempotente, que envolve duas imagens de entrada, uma máscara $\hat{I} = (D_I, I)$ e uma marcadora $\hat{J} = (D_I, J)$ e um elemento estruturante planar (i.e., relação de adjacência A). A reconstrução é dita **superior** quando $J(p) \geq I(p)$ para todo $p \in D_I$ e **inferior** quando $J(p) \leq I(p)$ para todo $p \in D_I$.

2.1 Reconstrução superior

A imagem \hat{J} pode ser gerada a partir da imagem \hat{I} somando-se um valor $h > 0$ às intensidades de \hat{I} ou aplicando-se um fechamento morfológico (dilatação seguida de erosão por A). No primeiro caso, a reconstrução preenche bacias de altura menor que h (**filtragem de h-bacias**) e, no segundo caso, bacias com área menor que o elemento estruturante são fechadas (**fechamento por reconstrução**). A implementação da reconstrução superior é uma IFT com minimização de V (resultado da reconstrução) usando função f_{rsup} ,

$$f_{rsup}(\langle q \rangle) = J(q) \quad (5)$$

$$f_{rsup}(\pi_p \cdot \langle p, q \rangle) = \max\{f_{rsup}(\pi_p), I(q)\}. \quad (6)$$

No caso 2D, a adjacência pode ser 4.

2.2 Reconstrução inferior

A imagem \hat{J} pode ser gerada a partir da imagem \hat{I} subtraindo-se um valor $h > 0$ das intensidades de \hat{I} ou aplicando-se uma abertura morfológica (erosão seguida de dilatação por A). No primeiro caso, a reconstrução remove domos de altura menor que h (**filtragem de h-domos**) e, no segundo caso, domos com área menor que o elemento estruturante são removidos (**abertura por reconstrução**). A implementação da reconstrução inferior é uma IFT com maximização de V (resultado da reconstrução) usando função f_{rinf} ,

$$f_{rinf}(\langle q \rangle) = J(q) \quad (7)$$

$$f_{rinf}(\pi_p \cdot \langle p, q \rangle) = \min\{f_{rinf}(\pi_p), I(q)\}. \quad (8)$$

No caso 2D, a adjacência pode ser 4.

3 Aspectos de implementação

Considerando o algoritmo da IFT com política de desempate FIFO, as raízes da floresta serão os mínimos de V na reconstrução superior. Cada pixel de um dado mínimo gerará assim uma árvore de caminhos ótimos. Se adotarmos adjacência-4, por exemplo, os mínimos formarão componentes conexos-4. A política de desempate LIFO fará com que cada mínimo tenha uma única raiz, representada pelo último pixel deste mínimo a sair da fila. Neste caso, o número de mínimos é o número de raízes.

No algoritmo da IFT com política FIFO, existe uma simplificação que pode ser adotada no caso das funções f_{bas} e f_{dom} . Quando um pixel p encontra um adjacente q e lhe oferece um caminho de melhor conectividade, este adjacente q nunca foi inserido na fila. Então, não é necessária a remoção da fila para atualização de atributos.

Um outro aspecto interessante é que após a filtragem, o mapa V preenche a bacia (remove o domo) na altura $V(q) = J(q)$ de cada raiz q . Bacias (domos) sem raízes devem ser totalmente preenchidas (eliminados), mas as bacias (domos) das raízes deveriam voltar ao brilho original em um novo mapa V' . Isso pode ser feito pelo seguinte pós-processamento: Para todo $p \in D_I$, se $V(p) = V(R(p))$ então $V'(p) \leftarrow I(p)$.

4 Nivelamento

Filtros alternados sequenciais (e.g., fechamento seguido de abertura, abertura seguida de fechamento seguido de abertura, etc), filtros lineares e outros tipos de filtros podem ser aplicados à imagem \hat{I} gerando uma imagem \hat{J} , na qual as intensidades são maiores que as de \hat{I} para alguns pixels e menores para outros. Neste caso, o **nivelamento** (*leveling*) força uma filtragem conexa segundo o algoritmo abaixo.

1. Calcula $\hat{J}' = (D_I, J')$ pela dilatação de \hat{J} usando adjacência A seguida do mínimo entre $\hat{I} = (D_I, I)$ e a imagem dilatada.
2. Encontra a reconstrução inferior \hat{V}_i com marcadora \hat{J}' e máscara \hat{I} .
3. Calcula $\hat{I}' = (D_I, I')$ pela erosão de \hat{I} usando adjacência A seguida do máximo entre $\hat{V}_i = (D_I, V_i)$ e a imagem erodida.
4. Encontra a reconstrução superior $\hat{V}_s = (D_I, V_s)$ com marcadora \hat{I}' e máscara \hat{V}_i .

A imagem \hat{V}_s é a imagem resultante.

5 Reconstrução Superior e Transformada de Watershed

Uma transformada de *watershed* é uma operação que simula a inundação de um relevo, quando fazemos um buraco em cada mínimo e mergulhamos ele em um lago, mas não deixamos águas provenientes de mínimos distintos se misturarem. As linhas de watershed são barreiras que construímos entre as zonas de influência desses mínimos e elas podem ser obtidas no mapa de

rótulos L , se propagarmos um rótulo por mínimo. A mesma IFT com f_{bas} que calcula uma transformada de watershed em L , com um pixel por mínimo em S , gera no mapa V a filtragem conexa correspondente. Esta observação também é válida no caso dual com f_{dom} .

Nesta seção, vamos apresentar um algoritmo de IFT com política de desempate FIFO que calcula a reconstrução superior em V e associa em L um rótulo distinto para cada mínimo de V . Lembremos que a IFT primeiro retira da fila os mínimos de V e depois propaga os caminhos ótimos a partir deles. Ao retirarmos o primeiro pixel p de mínimo da fila, nós queremos garantir que ele será rotulado com uma função $\lambda(p)$ (e.g, atribui um rótulo subsequente de 1 a n para cada mínimo) e este rótulo será propagado na sua zona de influência. Gostaríamos também que a propagação rotulasse todos os pixels da imagem. A função de conectividade fica então

$$f_{rsws}(\langle q \rangle) = \begin{cases} I(q) & \text{se } q \in \mathcal{R} \\ J(q) + 1 & \text{no caso contrário.} \end{cases} \quad (9)$$

$$f_{rsws}(\pi_p \cdot \langle p, q \rangle) = \max\{f_{rsws}(\pi_p), I(q)\}, \quad (10)$$

onde \mathcal{R} é o conjunto de raízes da floresta (um pixel por mínimo), que é descoberto durante o algoritmo da seguinte forma: se $P(p) = nil$ quando p é removido de Q , então p é raiz da floresta. Inicialmente, portanto, $f_{rsws}(\langle q \rangle) = J(q) + 1$ para todos os pixels $q \in D_I$ e quando as raízes são encontradas, nós baixamos o valor delas para $I(q)$.

Um caso particular é quando existem sementes $S \subset D_I$ e $J(q) = +\infty$, se $q \notin S$. Neste caso, a função $\lambda(q)$ pode associar o rótulo conhecido de q em S . O algoritmo é apresentado abaixo.

Algoritmo de reconstrução superior com watershed:

Entrada: Imagens $\hat{I} = (D_I, I)$, $\hat{J} = (D_I, J)$, $J(q) \geq I(q)$ para todo $q \in D_I$, adjacência A e função de rotulação λ .

Saída: Reconstrução superior $\hat{V} = (D_I, V)$, segmentação por watershed $\hat{L} = (D_I, L)$ e floresta de caminhos ótimos $\hat{P} = (D_I, P)$.

Auxiliares: Fila de prioridade Q e variável tmp .

1. Para todo $q \in D_I$, faça $P(q) \leftarrow nil$ e $V(q) \leftarrow J(q) + 1$.
2. Se $V(q) \neq +\infty$, então insere q em Q .
3. Enquanto $Q \neq \emptyset$ faça
 4. Remova p de Q tal que $V(p)$ é mínimo.
 5. Se $P(p) = nil$, então $V(p) \leftarrow I(p)$ e $L(p) \leftarrow \lambda(p)$.
 6. Para todo $q \in A(p)$, tal que $V(q) > V(p)$, faça
 7. Calcule $tmp \leftarrow \max\{V(p), I(q)\}$.
 8. Se $tmp < V(q)$, então

9. Se $V(q) \neq +\infty$, remove q from Q .
10. Faça $P(q) \leftarrow p$, $V(q) \leftarrow tmp$, $L(q) \leftarrow L(p)$ e insere q em Q .

A Figura 1 ilustra uma aplicação em segmentação para este algoritmo, onde \hat{I} é uma imagem de gradiente. O mesmo raciocínio se aplica à reconstrução inferior e transformada dual de watershed.

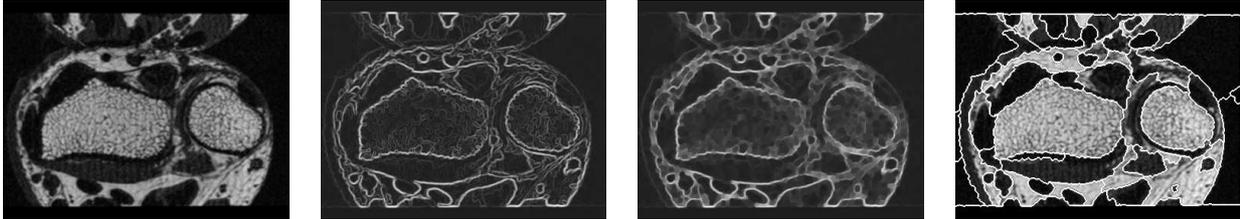


Figura 1: (a) Uma imagem de ressonância magnética de um pulso. (b) A imagem de gradiente \hat{I} e (c) seu fechamento morfológico $\hat{J}+1$ usando adjacência circular com raio 2.5. (d) Segmentação \hat{L} calculada por IFT com adjacência 3.5 e função f_{rsws} .

6 Extração de atributos por filtragem multi-escala

Vimos na aula passada que podemos segmentar a imagem usando marcadores de objeto e fundo e um vetor de atributos $\vec{F}(p)$ para cada pixel p . Este vetor de atributos foi criado com filtragem por difusão, mas poderia ser criado com filtragem por reconstrução. Podemos aplicar, por exemplo, filtros Gaussianos com valores diferentes variâncias seguidos de nivelamento, filtros de fechamento por reconstrução para diferentes raios de adjacência, filtros de abertura por reconstrução para diferentes raios de adjacência, etc, e no final associar os resultados dessas filtragens ao vetor \vec{F} .