Processamento de Imagens usando Grafos

Prof. Alexandre Xavier Falcão

Segundo semestre de 2004

1 Reconstrução Morfológica

Na aula 6, mencionamos que as reconstruções morfológicas podem ser calculadas com variantes da função f_{peak} e política FIFO. Considere, portanto, uma IFT com função f_{srec}:

$$\begin{aligned} f_{srec}(\langle q \rangle) &= J(q), \\ f_{srec}(\pi \cdot \langle p, q \rangle) &= \max\{f_{srec}(\pi), I(q)\}, \end{aligned} \tag{1}$$

onde $\hat{I}$ é a imagem máscara, $\hat{J}$ é a imagem marcadora, e $J(q) \geq I(q)$ para todo $q \in D_I$. Esta IFT retorna na imagem de custos $\hat{C} = (D_I, C)$ o resultado da **reconstrução morfológica superior** de $\hat{I}$ a partir de $\hat{J} = (D_I, J)$. Esta operação "preenche bacias" da imagem $\hat{I}$ e não cria falsas bordas, como ocorre tipicamente no caso de filtros lineares. O algoritmo simplificado é mostrado abaixo.

Reconstrução morfológica superior:

Entrada: Imagens $\hat{I} = (D_I, I)$, $\hat{J} = (D_I, J)$, $\hat{J} \geq \hat{I}$, e adjacência A.
Saída: Imagem $\hat{C} = (D_I, C)$ resultante da reconstrução.
Auxiliares: Fila Q de prioridades com política FIFO e variável tmp.

1. Para todo pixel $q \in D_I$ faça
2. $\quad$ $C(q) \leftarrow J(q)$ e insira q em Q.
3. Enquanto $Q \neq \emptyset$ faça
4. $\quad$ Remova um pixel p de Q cujo custo $C(p)$ é mínimo.
5. $\quad$ Para todo $q \in A(p)$, tal que $C(q) > C(p)$, faça
6. $\quad\quad$ $tmp \leftarrow \max\{C(p), I(q)\}$.
7. $\quad\quad$ Se $tmp < C(q)$ faça
8. $\quad\quad\quad$ Remova q de Q, $C(q) \leftarrow tmp$, e insira q em Q.

O caso dual é a **reconstrução morfológica inferior** ("corta domos"), que pode ser calculada com função f_{irec}:

$$\begin{aligned} f_{irec}(\langle q\rangle) &= K - J(q), \\ f_{irec}(\pi\cdot\langle p,q\rangle) &= \max\{f_{irec}(\pi), K - I(q)\}, \end{aligned} \tag{2}$$

onde K é o valor máximo em $\hat{I}$ e $J(q) \leq I(q)$ para todo $q \in D_I$. Neste caso, o resultado $\hat{C}' = (D_I, C')$, onde $C'(q) = K - C(q)$ para todo $q \in D_I$, da reconstrução é o complemento da imagem $\hat{C}$ de custos da IFT.

Também podemos ter reconstruções restritas a um conjunto $S \subset D_I$ de sementes. A reconstrução superior, por exemplo, usando a função f^S_{srec},

$$\begin{aligned} f^S_{srec}(\langle q\rangle) &= J(q), \text{ se } q \in S \text{ e } +\infty \text{ no caso contrário.} \\ f^S_{srec}(\pi\cdot\langle p,q\rangle) &= \max\{f^S_{srec}(\pi), I(q)\}, \end{aligned} \tag{3}$$

requer as seguintes modificações no algoritmo acima. As linhas 1 e 2 mudam para

1. Faça $C(q) \leftarrow +\infty$ para todo pixel $q \in D_I$.
2. Enquanto $S \neq \emptyset$ faça
3. Remova q de S, $C(q) \leftarrow J(q)$ e insira q em Q.

e a linha 8 muda para

9. $C(q) \leftarrow tmp$ e insira q em Q.

Esses algoritmos podem ser usados em vários filtros morfológicos (**veja a aula 16 do curso MO443**). Em particular, as reconstruções superior e inferior usando conjuntos de sementes podem ser usadas para reduzir o problema de segmentação de alguns objetos a uma simples limiarização da imagem reconstruída (veja exemplo no tutorial da IFT).

Outro variante interessante é a **reconstrução superior (inferior) local**. Esta reconstrução "preenche" ("corta") apenas bacias (domos) marcados por pixels sementes com brilho estritamente maior (menor) que o brilho da imagem original (Veja exemplos no tutorial da IFT). Por exemplo, a função de custo da reconstrução superior é f^S_{lsrec}:

$$\begin{aligned} f^S_{lsrec}(\langle q\rangle) &= \begin{cases} J(q), & \text{se } J(q) > I(q) \text{ e } q \in S, \\ +\infty, & \text{no caso contrário.} \end{cases} \\ f^S_{lsrec}(\pi\cdot\langle p,q\rangle) &= \begin{cases} f^S_{lsrec}(\pi), & \text{se } f^S_{lsrec}(\pi) > I(q), \\ +\infty, & \text{no caso contrário.} \end{cases} \end{aligned} \tag{4}$$

Note que, a reconstrução preenche apenas a bacia que contém o pixel q até o nível $J(q)$. Porém, se as zonas de influência de duas sementes se encontram com níveis diferentes, a semente de menor nível domina a de maior nível.

Neste curso, vamos introduzir um novo operador, **alagamento regional** (*regional flooding*), que tem o mesmo efeito de uma reconstrução superior local, quando as zonas de influências das sementes **não se encontram**, e tem um efeito de inundação da superfície, onde a semente de maior nível domina a de menor nível, quando suas zonas de influência se encontram. Ou seja, imagine que derramamos água em bacias com sementes $q \in S$ até o nível especificado por $J(q) > I(q)$, porém este nível pode ser diferente para cada semente. Neste sentido, quando a água transborda de uma bacia para uma vizinha com nível menor, preenchemos a bacia vizinha até o maior nível. Este operador requer função de custo f_{rf}^{S}, onde K é o brilho máximo de $\hat{J}$:

$$
\begin{aligned}
f_{rf}^{S}(\langle q \rangle) &= \begin{cases} K - J(q), & \text{se } J(q) > I(q) \text{ e } q \in S, \\ +\infty, & \text{no caso contrário.} \end{cases} \\
f_{rf}^{S}(\pi \cdot \langle p, q \rangle) &= \begin{cases} f_{rf}^{S}(\pi), & \text{se } f_{rf}^{S}(\pi) < K - I(q), \\ +\infty, & \text{no caso contrário.} \end{cases}
\end{aligned} \tag{5}
$$

Note que, em ambos os casos, a imagem de raízes pode ser usada para definir objetos representados pelas bacias. Porém, quando o objetivo é a filtragem, regiões não conquistadas pela IFT ficarão com custo infinito. No caso do alagamento regional, teremos ainda regiões conquistadas com custo complementar em relação a K. Neste caso, podemos gerar uma imagem filtrada aplicando um processamento local na imagem de custos $\hat{C} = (D_I, C)$ para todo pixel $p \in D_I$: i.e., $C(p) \leftarrow I(p)$, se $C(p) = +\infty$, e $C(p) \leftarrow K - C(p)$, se $C(p) \neq +\infty$.

O alagamento regional pode ser simplificado no seguinte algoritmo.

Filtragem por alagamento regional:

Entrada: Imagens $\hat{I} = (D_I, I)$ e $\hat{J} = (D_I, J)$, sementes $S \subset D_I$, e adjacência A.
Saída: Imagem $\hat{C} = (D_I, C)$ resultante da filtragem.
Auxiliares: Fila Q de prioridades com política FIFO.

1. Faça $C(q) \leftarrow I(q)$ para todo $q \in D_I$.
2. Enquanto $S \neq \emptyset$ faça
3. Remova q de S.
4. Se $J(q) > I(q)$ então
5. $C(q) \leftarrow J(q)$ e insira q em Q.
6. Enquanto $Q \neq \emptyset$ faça
7. Remova um pixel p de Q cujo custo $C(p)$ é **máximo**.
8. Para todo $q \in A(p)$, tal que $C(p) > C(q)$, faça
9. Se $q \in Q$, remova q de Q.
10. $C(q) \leftarrow C(p)$ e insira q em Q.

Observe que quando as zonas de influência de duas sementes se encontram com alturas diferentes, a mais alta prevalece, e que isto é naturalmente tratado com a remoção de pixels com custo máximo em Q. Note também que o alagamento regional tem uma operação dual, para cortar domos selecionados por pixels sementes.

Uma das aplicações do alagamento regional é o fechamento de bacias por área. Este algoritmo requer um variante da IFT-watershed clássica, para calcular as bacias que serão preenchidas e seus respectivos níveis, e um alagamento regional para efetivamente preencher essas bacias. A primeira parte simula a inundação da imagem toda, unindo as bacias em uma imagem $\hat{R}$ de representantes à medida que o nível da água sobe. Durante este processo, o algoritmo calcula os níveis para fechamento das bacias em uma imagem $\hat{J}$ e as áreas inundadas em $\hat{A}r$. O nível de fechamento de cada bacia pára de ser atualizado quando a água chega a uma altura em que a área inundada é maior ou igual a um limiar de área dado. A segunda parte do algoritmo é o alagamento regional de $\hat{J}$, usando como sementes as raízes da IFT anterior (**um pixel por mínimo regional**) obtidas da imagem de predecessores $\hat{P}$.

Fechamento por área:

Entrada: Imagens $\hat{I} = (D_I, I)$ e limiar T_a de área.
Saída: Imagem $\hat{J} = (D_I, J)$ resultante da filtragem.
Auxiliares: Fila Q de prioridades com política FIFO, adjacência-4 A, variável tmp, e imagens de custos $\hat{C} = (D_I, C)$, areas $\hat{A}r = (D_I, Ar)$, representantes $\hat{R} = (D_I, R)$, e predecessores $\hat{P} = (D_I, P)$.

1. Para todo pixel $q \in D_I$, faça $P(q) \leftarrow nil$, $C(q) \leftarrow I(q) + 1$, $R(q) \leftarrow q$, $A(q) \leftarrow 0$, $J(q) \leftarrow I(q)$ e insira q em Q.
2. Enquanto $Q \neq \emptyset$ faça
3.   Remova um pixel p de Q cujo custo $C(p)$ é mínimo.
4.   Faça $r_p \leftarrow Representante(\hat{R}, p)$.
5.   Se $A(r_p) \leq T_a$ e $J(r_p) < I(p)$, então $J(r_p) \leftarrow I(p)$.
6.   Faça $A(r_p) \leftarrow A(r_p) + 1$.
7.   Para todo $q \in A(p)$, faça
8.   Se $C(q) > C(p)$, então
9. $tmp \leftarrow \max\{C(p), I(q)\}$.
10.   Se $tmp < C(q)$ faça
11.   Remova q de Q, $C(q) \leftarrow tmp$, $P(q) \leftarrow p$, $R(q) \leftarrow r_p$ e insira q em Q.
12.   Se não, então

13. Se $q \notin Q$, então

14. Faça $r_q \leftarrow Representante(\hat{R}, q)$.

15. Se $r_p \neq r_q$, então

16. Se $A(r_q) \leq T_a$ e $J(r_q) < I(p)$, então $J(r_q) \leftarrow I(p)$.

17. Se $A(r_p) < A(r_q)$, então $tmp \leftarrow r_p$, $r_p \leftarrow r_q$, e $r_q \leftarrow tmp$.

18. Faça $R(r_q) \leftarrow r_p$ e $A(r_p) \leftarrow A(r_p) + A(r_q)$.

19. Se $P(p) = nil$ e $P(q) = nil$, então $P(r_q) \leftarrow r_p$.

20. Para todo pixel $q \in D_I$, faça

21. Se $P(q) = nil$, então insira q em Q.

22. Enquanto $Q \neq \emptyset$ faça

23. Remova um pixel p de Q cujo $J(p)$ é **máximo**.

24. Para todo $q \in A(p)$, tal que $J(p) > J(q)$, faça

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

26. $J(q) \leftarrow J(p)$ e insira q em Q.

A função *Representante* é a mesma da aula 5. A união de bacias adjacentes é realizada nas linhas 17 e 18, de forma tal que a maior bacia sempre fica como representante das duas. Antes disso, a linha 16 atualiza o nível da bacia vizinha para que acompanhe o nível atual da água, caso seu nível de fechamento esteja ainda sendo calculado. A linha 19 garante um pixel por mínimo regional de cada bacia. As linhas 20 a 26 calculam a reconstrução local.

Uma observação importante: A implementação do algoritmo acima requer a função $ResetQueue(Q)$ entre as linhas 19 e 20. Esta função reinicializa valores de controle da fila Q.

2 Exercícios

1. Qual é a função de custo da operação dual do alagamento regional? Escreva o seu algoritmo simplificado.

2. Implemente e verifique os resultados dos algoritmos acima para cada filtro mencionado na aula 16 do curso MO443 e para os filtros de fechamento e abertura por área. Compare os resultados com o fechamento e a abertura por área usando árvores de componentes.

3. Como ficaria o algoritmo de fechamento por área, caso substituíssemos Q por uma fila de prioridades com política $LIFO$?