# Filtragem Conexa

Prof. Dr. Paulo A. V. de Miranda

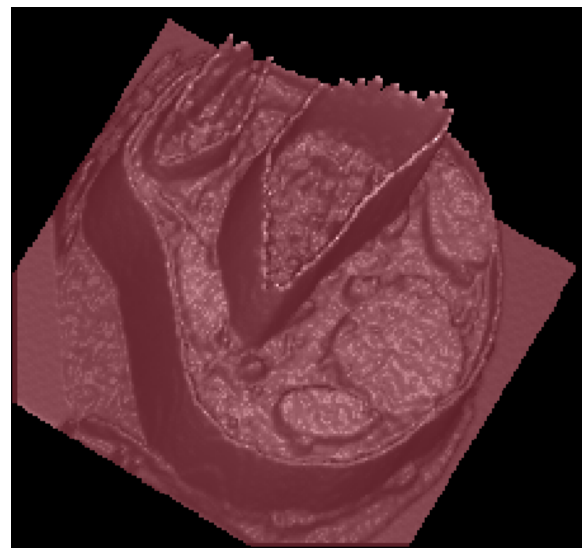
pmiranda@vision.ime.usp.br

Instituto de Matemática e Estatística (IME),

Universidade de São Paulo (USP)

# Operador conexo

Considere uma imagem em níveis de cinza $\hat{I} = (\mathcal{D}_I, I)$ representada por uma superfície topográfica em que os níveis $I(p)$ indicam a altitude do ponto $p$ no relevo.



Exemplo de imagem de entrada vista como uma superfície topográfica.

# Operador conexo

- Este relevo possui domos, bacias e platôs.
- Um platô (*flat zone*) neste relevo é um componente conexo maximal, onde todos os pixels possuem o mesmo valor (mesma altitude).
- Um operador $\psi$ é dito conexo se e somente se qualquer par de pixels pertencentes a um dado platô em $\hat{I}$ também pertencem a um mesmo platô em $\psi(\hat{I})$.
- A principal vantagem é que a operação conexa não cria falsas bordas, como a filtragem linear, apenas elimina bordas entre platôs.

# Operador conexo

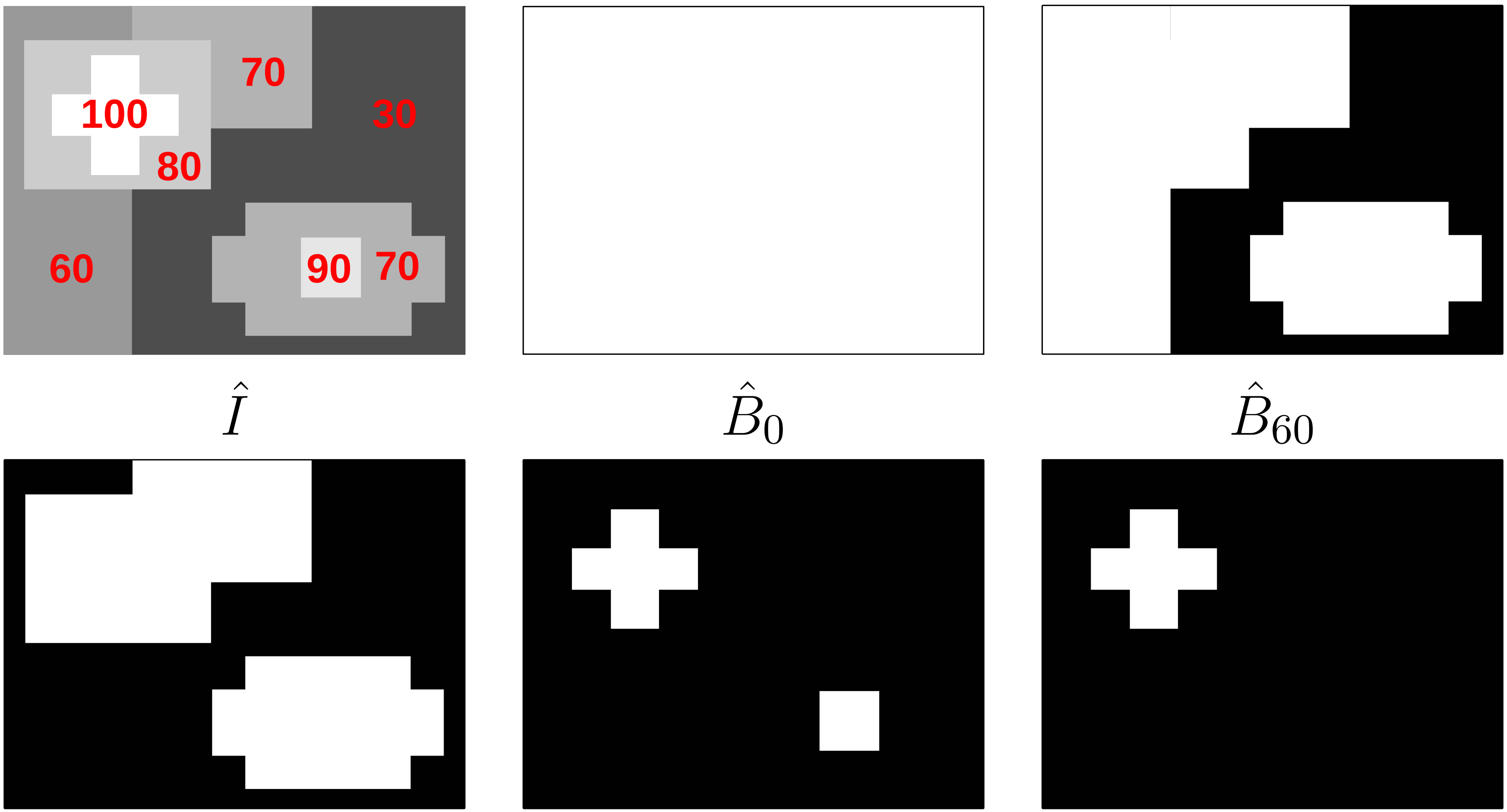
Outra forma de visualizar uma operação conexa é através da decomposição por limiar:

- A imagem $\hat{I}$ decomposta por limiar forma um conjunto $\mathcal{T}_{\hat{I}}$ de imagens binárias $\hat{B}_l = (\mathcal{D}_I, B_l), l = 0, 1, \ldots, I_{max}$, onde $B_l(p) = 1$ se $I(p) \geq l$, e $0$ no caso contrário ($I_{max} = \max_{\forall p \in \mathcal{D}_I} \{I(p)\}$).
- Operadores conexos apenas eliminam ou unem componentes conexos deste conjunto.

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.

# Operador conexo

Exemplo de decomposição por limiar:




$\hat{I}$ $\hat{B}_0$ $\hat{B}_{60}$

$\hat{B}_{70}$ $\hat{B}_{90}$ $\hat{B}_{100}$

# Transformada de watershed

A transformada de watershed pode ser vista como uma IFT com função de custo de caminho:

$$
\begin{aligned}
f_{peak}(\langle t \rangle) &= H(t) \\
f_{peak}(\pi_s \cdot \langle s, t \rangle) &= \max\{f_{peak}(\pi_s), I(t)\}
\end{aligned}
$$

onde $H(t) < I(t)$ (imposição de marcadores), se $t \in \mathcal{S}$, e $H(q) = +\infty$ no caso contrário.

- As bacias sem sementes se transformam em platôs na imagem $\hat{V}$ de custos, com valores iguais à altura das águas que as encontram.
- As raízes da floresta serão os pixels sementes, que estarão em mínimos regionais da imagem de custos.

# Transformada de watershed

A transformada de watershed pode ser vista como uma IFT com função de custo de caminho:

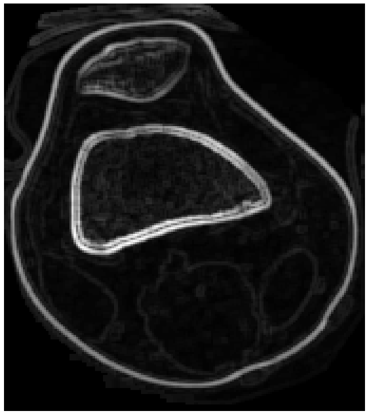
$$\begin{aligned} f_{peak}(\langle t \rangle) &= H(t) \\ f_{peak}(\pi_s \cdot \langle s, t \rangle) &= \max\{f_{peak}(\pi_s), I(t)\} \end{aligned}$$

onde $H(t) < I(t)$ (imposição de marcadores), se $t \in \mathcal{S}$, e $H(q) = +\infty$ no caso contrário.

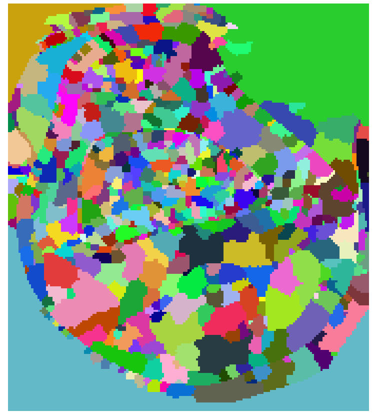
- Note que as linhas de watershed podem ser obtidas da imagem $\hat{R}$ de raízes (e.g., a linha composta por todos os pixels $p$ com raiz $R(p) \neq R(q)$ para algum $q$ vizinho-4 de $p$, gera espessura máxima de 2 pixels).

# Transformada de watershed

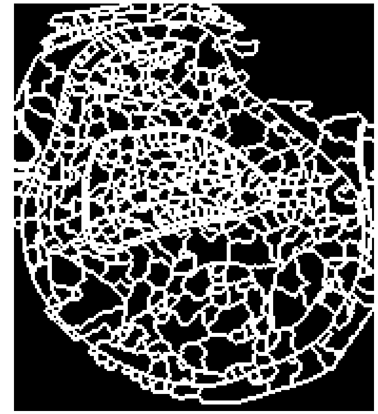
Exemplo onde as sementes são os mínimos regionais.



$\hat{I}$

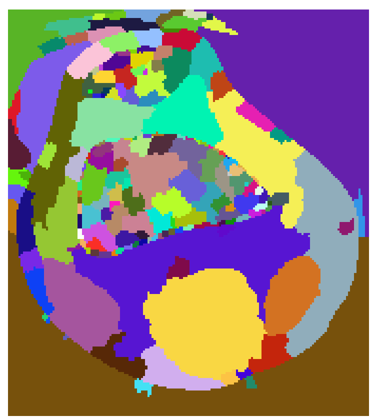


$\hat{L}$

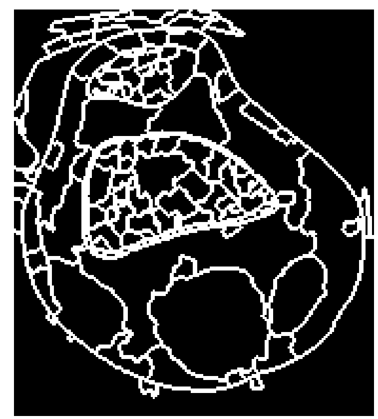


linhas de watershed

Imagem $\hat{I}$ filtrada



$\hat{L}$



linhas de watershed

# Transformada de watershed

A detecção automática de sementes:

- pixels obtidos por limiarização,
- pixels da borda da imagem,
- mínimos/máximos regionais.

# Transformada de watershed

Os mínimos podem ser obtidos via IFT com função $f_{ini}$:

$$
\begin{aligned}
f_{ini}(\langle t \rangle) &= I(t), \text{para todo } t \in \mathcal{D}_I, \\
f_{ini}(\pi_s \cdot \langle s, t \rangle) &= \begin{cases} f_{ini}(\pi_s), & \text{se } I(s) \leq I(t), \\ +\infty, & \text{caso contrário.} \end{cases}
\end{aligned}
$$

- Com a política FIFO, uma imagem binária dos mínimos regionais pode ser gerada associando $1$ a pixels raízes e $0$ aos demais.
- Com a política LIFO, vamos obter exatamente um pixel por mínimo regional.

# Filtragem Conexa

Filtros conexos para preenchimento de bacias podem ser obtidos usando a IFT com função de custo de caminho:

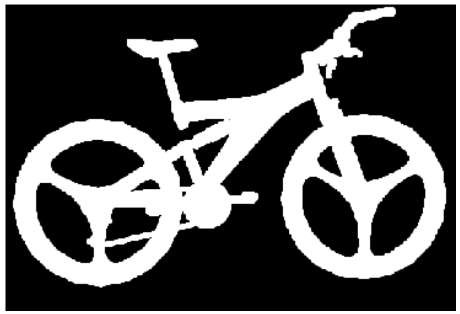
$$
\begin{aligned}
f_{peak}(\langle t \rangle) &= H(t) \\
f_{peak}(\pi_s \cdot \langle s, t \rangle) &= \max\{f_{peak}(\pi_s), I(t)\}
\end{aligned}
$$

onde $H(t) = I(t)$, se $t \in \mathcal{S}$, e $H(q) = +\infty$ no caso contrário.

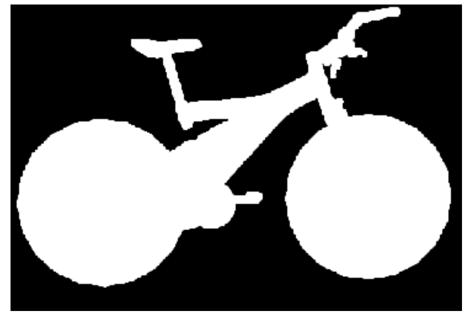
- O mapa de custo gera uma imagem filtrada $\hat{V}$, que simplifica a imagem por preencher as bacias não marcadas por $\mathcal{S}$.
- No caso em que $\mathcal{S}$ é a borda da imagem:
  - Buracos escuros de objetos claros, em fundos escuros, são preenchidos com brilho claro em $\hat{V}$ (**closing of holes**).

# Filtragem Conexa

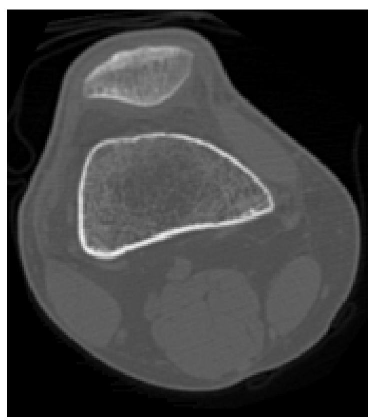
Fechamento de buracos (**closing of holes**):



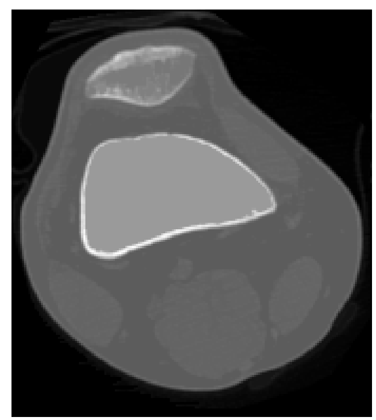
$\hat{I}$



$\hat{V}$



$\hat{I}$



$\hat{V}$

# Filtragem Conexa

Filtros conexos para preenchimento de bacias podem ser obtidos usando a IFT com função de custo de caminho:

$$\begin{aligned} f_{peak}(\langle t \rangle) &= H(t) \\ f_{peak}(\pi_s \cdot \langle s, t \rangle) &= \max\{f_{peak}(\pi_s), I(t)\} \end{aligned}$$

onde $H(t) = I(t)$, se $t \in \mathcal{S}$, e $H(q) = +\infty$ no caso contrário.

- No caso de política FIFO, existe uma simplificação que pode ser adotada. Quando um pixel $p$ encontra um adjacente $q$ e lhe oferece um caminho de melhor conexidade, este adjacente $q$ nunca foi inserido na fila. Então, não é necessária a remoção da fila para atualização de atributos.

# Filtragem por reconstrução morfológica

A reconstrução morfológica é uma operação que envolve duas imagens de entrada, uma máscara $\hat{I} = (\mathcal{D}_I, I)$ e uma marcadora $\hat{J} = (\mathcal{D}_I, J)$ e um elemento estruturante planar (i.e., relação de adjacência $\mathcal{A}$). A reconstrução é dita:

- **superior** quando $J(p) \geq I(p)$ para todo $p \in \mathcal{D}_I$, e
- **inferior** quando $J(p) \leq I(p)$ para todo $p \in \mathcal{D}_I$.

# Reconstrução superior

Dependendo de como a imagem $\hat{J}$ é gerada a partir da imagem $\hat{I}$, temos origem a diferentes métodos:

- **Filtragem de h-bacias:**
  - $\hat{J}$ é obtido somando-se um valor $h > 0$ às intensidades de $\hat{I}$ (i.e., $J(p) = I(p) + h$),
  - preenche bacias de altura menor que $h$.
- **Fechamento por reconstrução:**
  - $\hat{J}$ é obtido aplicando-se em $\hat{I}$ um fechamento morfológico (dilatação seguida de erosão por $\mathcal{A}$).
  - bacias menores que o elemento estruturante são fechadas.

# Reconstrução superior

A implementação da reconstrução superior é uma IFT com minimização de $V$ (resultado da reconstrução) usando função $f_{rsup}$:

$$\begin{aligned} f_{rsup}(\langle t \rangle) &= J(t) \\ f_{rsup}(\pi_s \cdot \langle s, t \rangle) &= \max\{f_{rsup}(\pi_s), I(t)\} \end{aligned}$$

# Reconstrução superior e Watershed

Vamos apresentar um algoritmo de IFT com política FIFO que calcula a reconstrução superior em $V$, e associa em $L$ um rótulo distinto para cada mínimo de $V$:

$$
\begin{aligned}
f_{rsws}(\langle t \rangle) &= \begin{cases} J(t), & \text{se } t \in \mathcal{R}, \\ J(t)+1, & \text{caso contrário.} \end{cases} \\
f_{rsws}(\pi_s \cdot \langle s, t \rangle) &= \max\{f_{rsws}(\pi_s), I(t)\},
\end{aligned}
$$

onde $J(t) \geq I(t)$, e $\mathcal{R}$ é o conjunto de raízes da floresta, que é descoberto durante o algoritmo da seguinte forma: se $P(s) = nil$ quando $s$ é removido de $Q$, então $s$ é raiz da floresta. Quando as raízes $s$ são encontradas, nós baixamos o valor delas $V(s)$ para $J(s)$.

# Reconstrução superior e Watershed

**Algoritmo 1 -** Algoritmo de reconstrução superior com watershed

ENTRADA: Imagens $\hat{I} = (\mathcal{D}_I, I)$, $\hat{J} = (\mathcal{D}_I, J)$, com $J(q) \geq I(q)$ para $\forall q \in \mathcal{D}_I$, e adjacência $\mathcal{A}$.

SAÍDA: Reconstrução superior $\hat{V} = (\mathcal{D}_I, V)$, e rótulo por watershed $\hat{L} = (\mathcal{D}_I, L)$.

AUXILIAR: Fila de prioridade $Q$, predecessores $\hat{P} = (\mathcal{D}_I, P)$, variáveis $tmp$ e $l = 1$, e vetor de $estado$ inicialmente zerado.

# Reconstrução superior e Watershed

**Algoritmo 1** — ALGORITMO DE RECONSTRUÇÃO SUPERIOR COM WATERSHED (CONT.)

1. ***Para Cada*** $t \in \mathcal{D}_I$, ***Faça*** $P(t) \leftarrow nil$ e $V(t) \leftarrow J(t)+1$. ***Se*** $V(t) \neq +\infty$, ***Então*** insira $t$ em $Q$.
2. ***Enquanto*** $Q \neq \emptyset$, ***Faça***
3. *Remova um pixel* $s$ *de* $Q$ *cujo valor* $V(s)$ *seja minimo, e faça* $estado(s) \leftarrow 1$.
4. ***Se*** $P(s) = nil$, ***Então***
5. $V(s) \leftarrow J(s)$, $L(s) \leftarrow l$, e $l = l + 1$.
6. ***Para Cada*** $t \in \mathcal{A}(s)$, *tal que* $estado(t) = 0$, ***Faça***
7. $tmp \leftarrow \max\{V(s), I(t)\}$.
8. ***Se*** $tmp < V(t)$, ***Então***
9. ***Se*** $V(t) \neq +\infty$, ***Então*** *remova* $t$ *de* $Q$.
10. $P(t) \leftarrow s$, $V(t) \leftarrow tmp$, $L(t) \leftarrow L(s)$ *e insira* $t$ *em* $Q$.

# Artigo para leitura

*A.X. Falcão, B. S. da Cunha, and R. A. Lotufo*,
**Design of connected operators using the image foresting transform**,
In Proc. of SPIE on Medical Imaging, volume 4322, pages 468-479. SPIE, 2001.

`http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=906355`