计算机辅助手术讲座（11）

Image Guided Surgery (11)

灰度的数学形态学(3)

Mathematical morphology in gray scale (3)

顾 力栩 (*Lixu Gu*)

上海交通大学 Med-X研究院

2009.12

Conditional Dilation

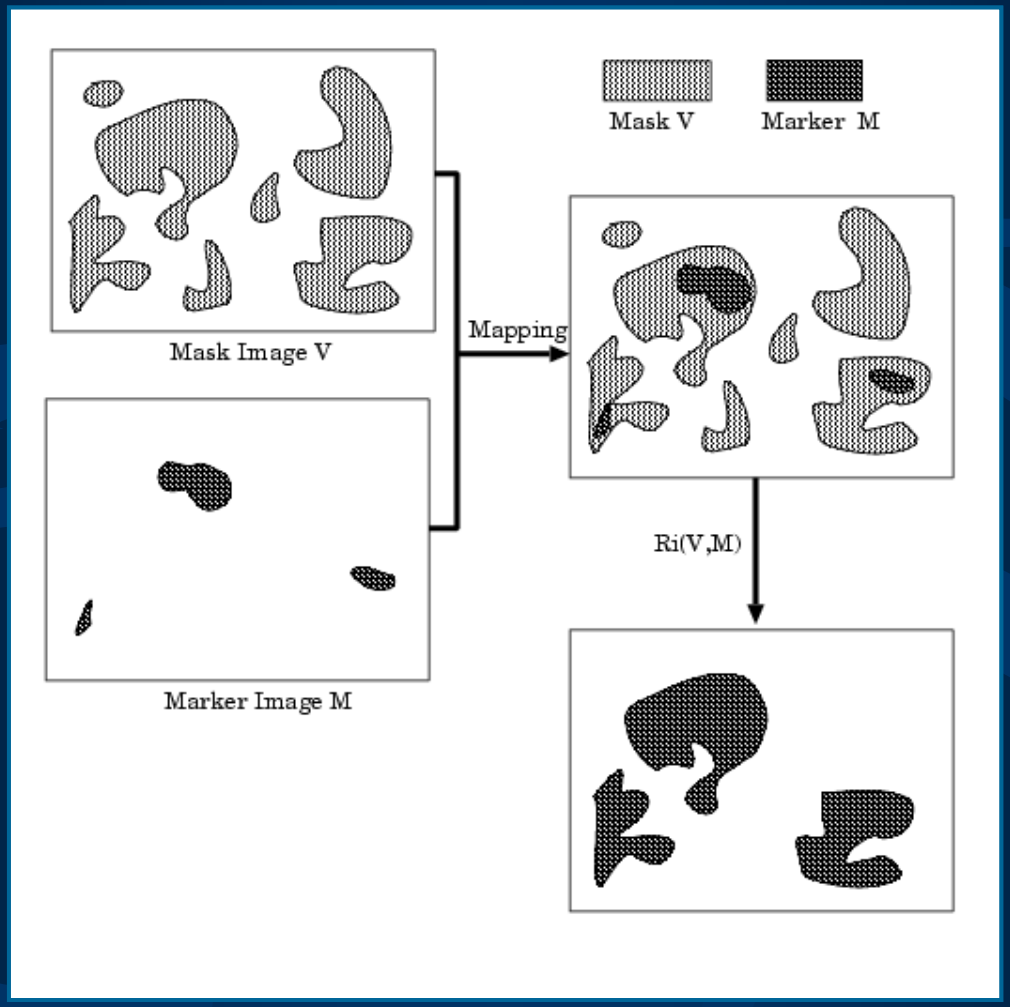
- Conditional Dilation : a special recursive dilation operation. It is also known as *Geodesic Dilation* or *Morphological Reconstruction* used for restoring destroyed objective regions.
  - ❖ Let M and V ($M \subseteq V$) be two binary images defined as "marker" and "mask", respectively.
  - ❖ Conditional dilation $R_i(M,V)$ is defined as:

$$R_i(M,V) = (M \overset{i}{\oplus} K) \bigcap V,$$

$$until \quad R_i(M,V) = R_{i-1}(M,V)$$

  - ❖ Marker M is only allowed to grow in the region restricted by mask V.

How it works



Morphological Reconstruction

- Algorithm for binary reconstruction:

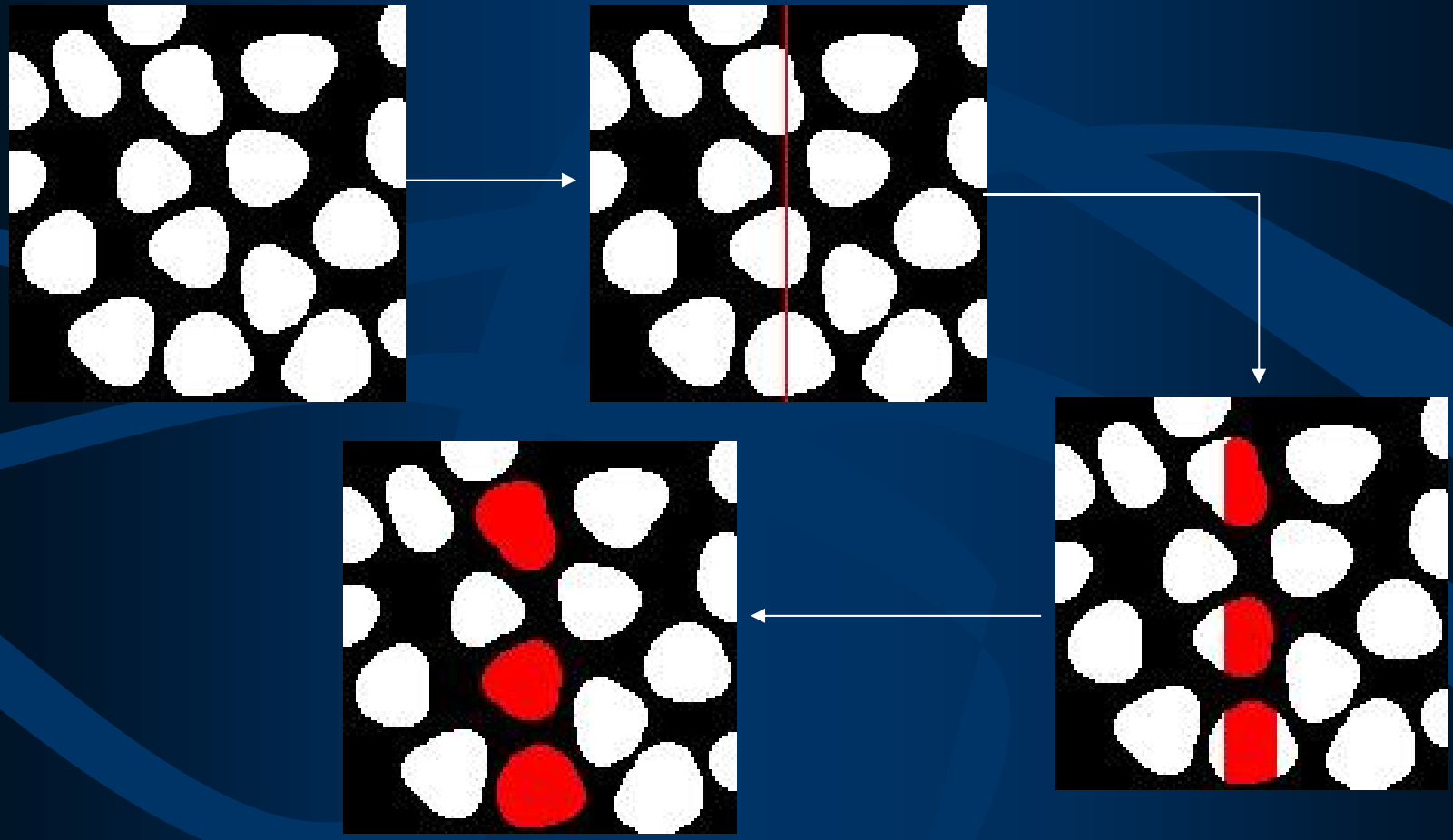
1. $M = V \circ K$, where K is any SE.
2. $T = M$,
3. $M = M \oplus K_i$, where $i=4$ or $i=8$,
4. $M = M \cap V$, [*Take only those pixels from M that are also in V .*]
5. if $M \neq T$ then go to 2,
6. else stop;

Original (V)

Opened (M)

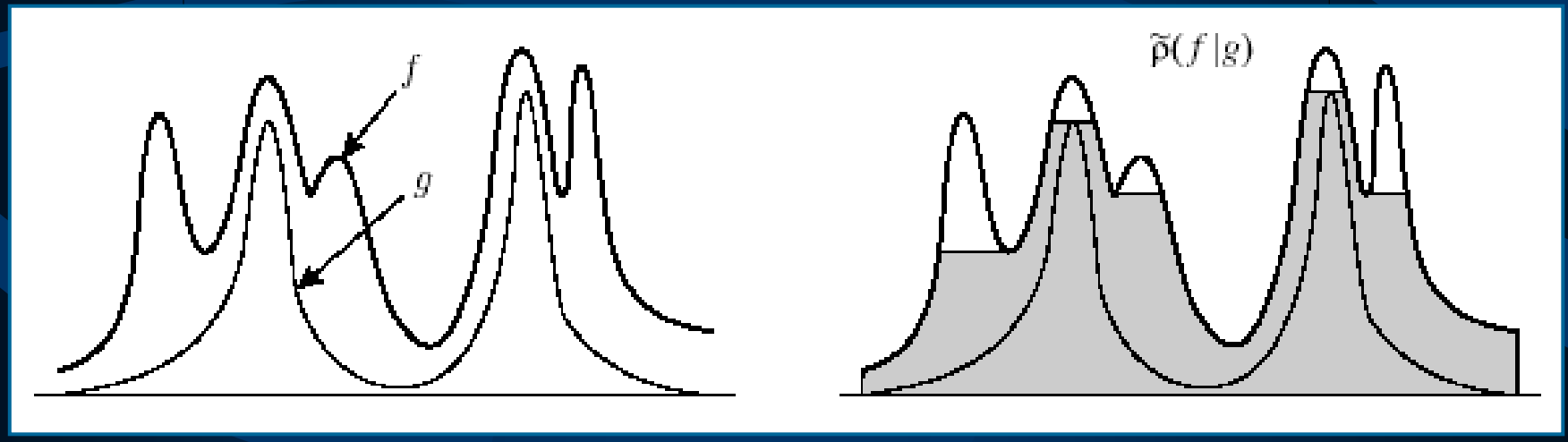
Reconstructed (T)

Conditional Dilation

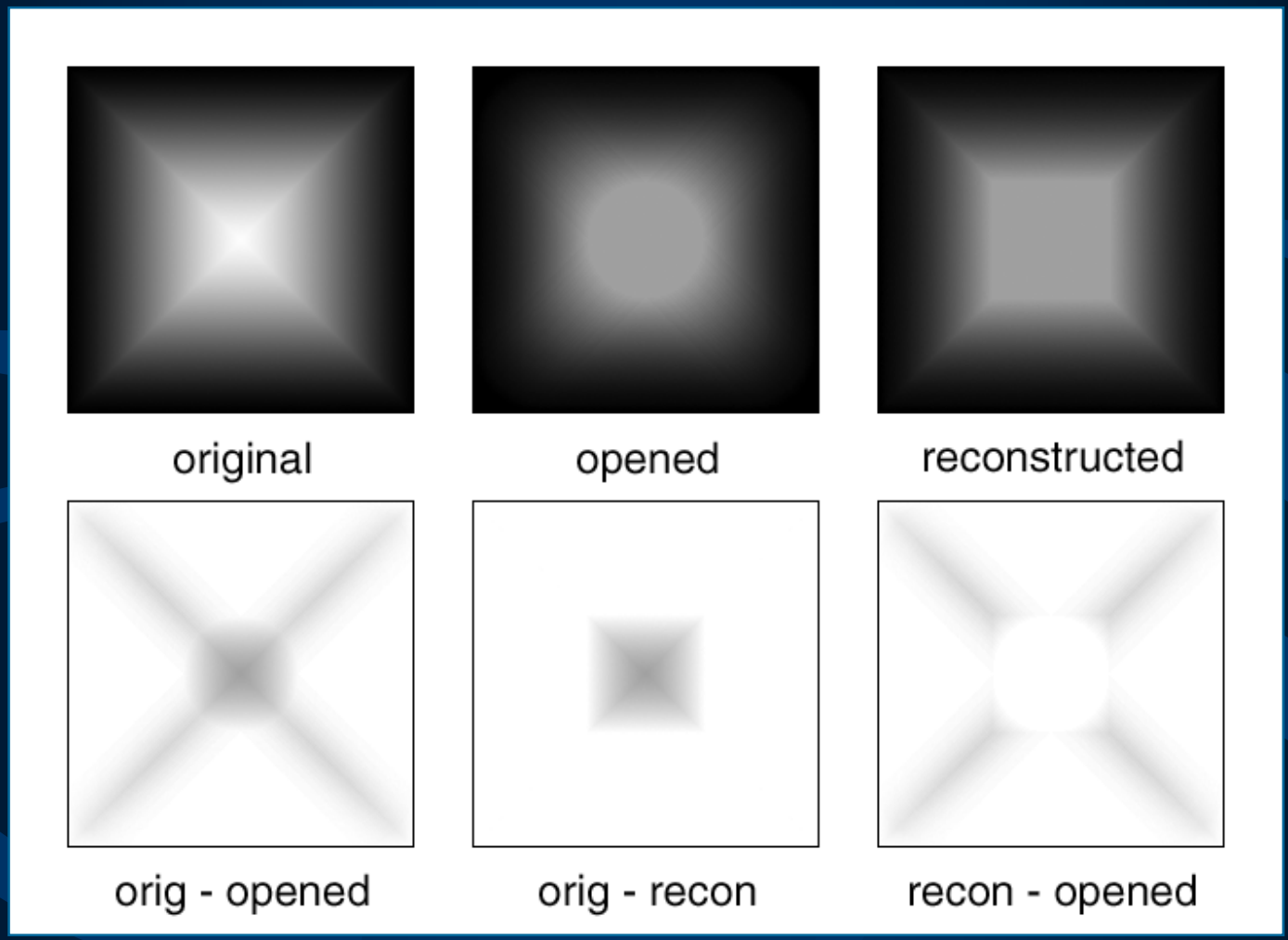


Grayscale Reconstruction

- Grayscale Reconstruction:
  - Step1: perform grayscale dilation on marker *g*.
  - Step2: check every gray value in dilated result is not exceed the restriction of mask *f*.
  - Step3: repeat step1 and 2, until *g* getting stable.

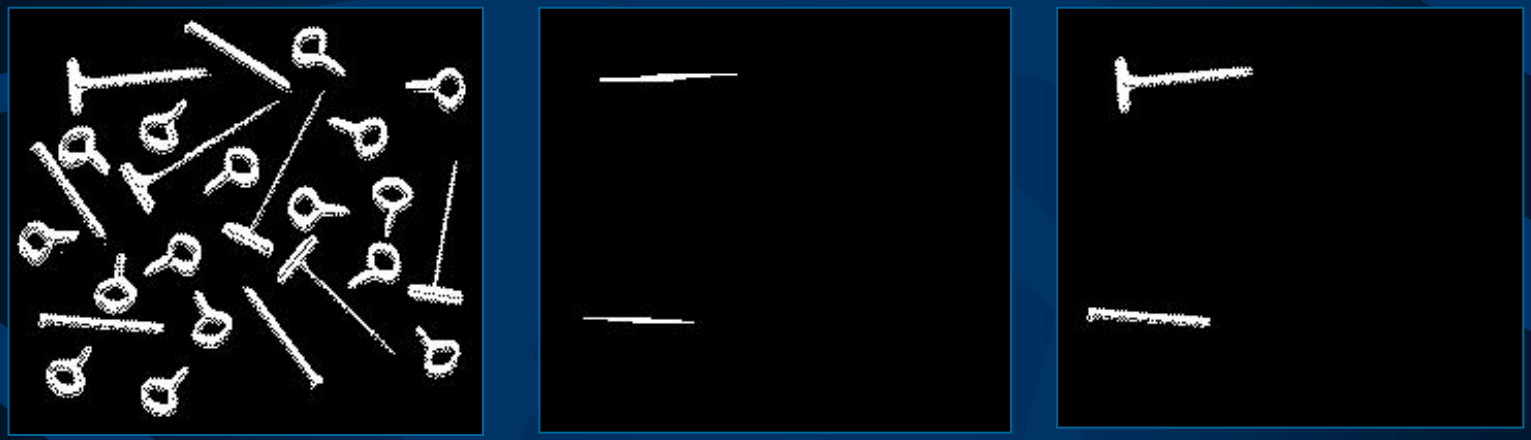


Grayscale Reconstruction



Grayscale Reconstruction

- Grayscale Opening by Reconstruction (OBR):
  1. Smooth image or detect seeds by grayscale opening operation
  2. Recover the objective regions by grayscale reconstruction



f $f \circ g\, r_{30X1} K_{line}$ OBR

Grayscale Reconstruction

- Grayscale Closing by Reconstruction(CBR):
  1. Smooth image by grayscale closing operation
  2. Recover the objective regions by grayscale reconstruction

Grayscale Reconstruction

Source

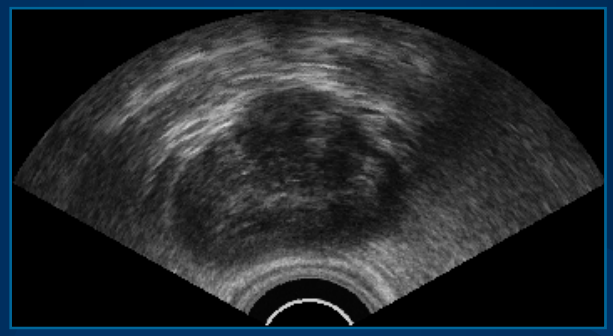
$r_{10}K_{disk}$

OBR

CBR

$r_{20}K_{disk}$

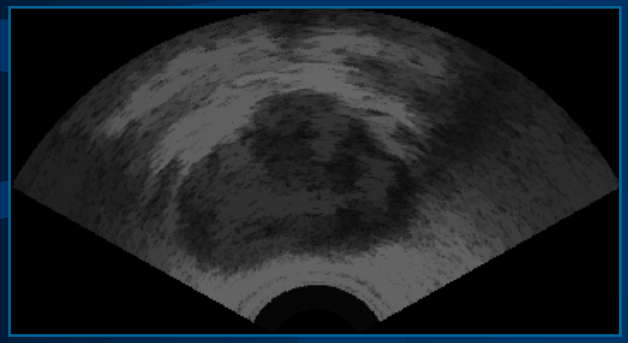
Grayscale Reconstruction

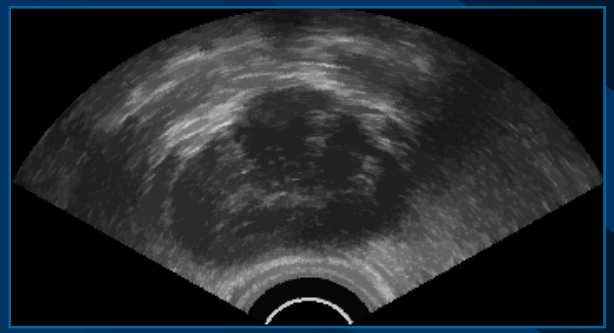


OBR

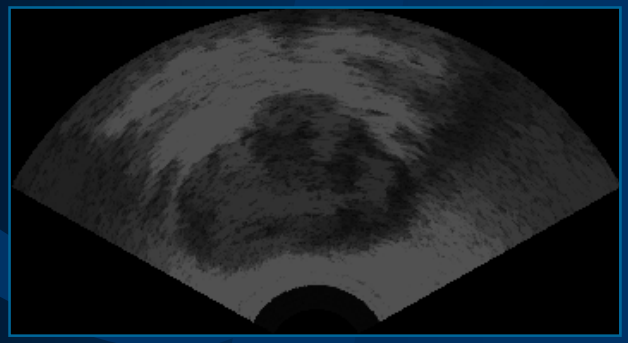
CBR

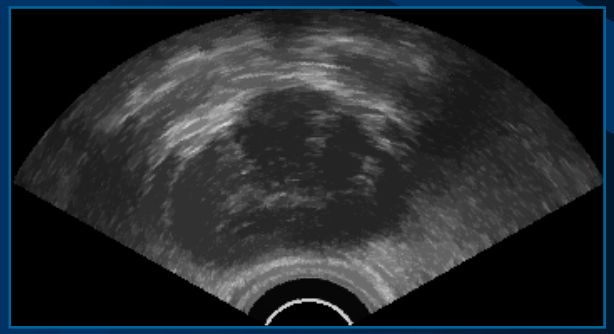
r_5K_{disk}



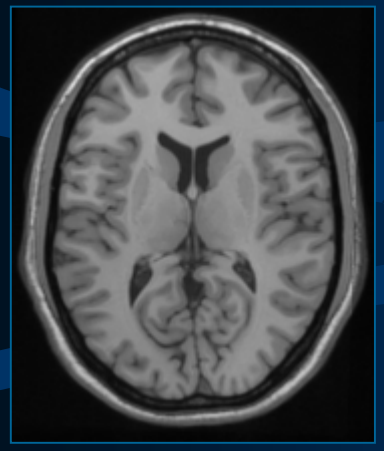


$r_{10}K_{disk}$

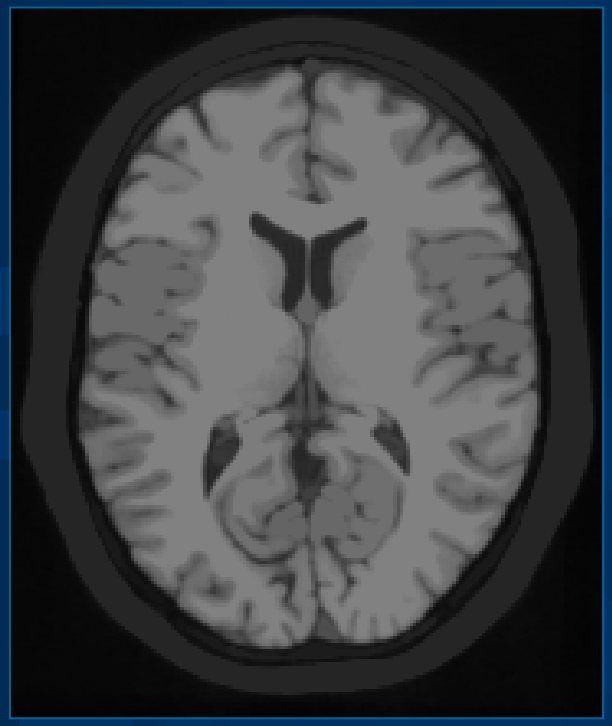




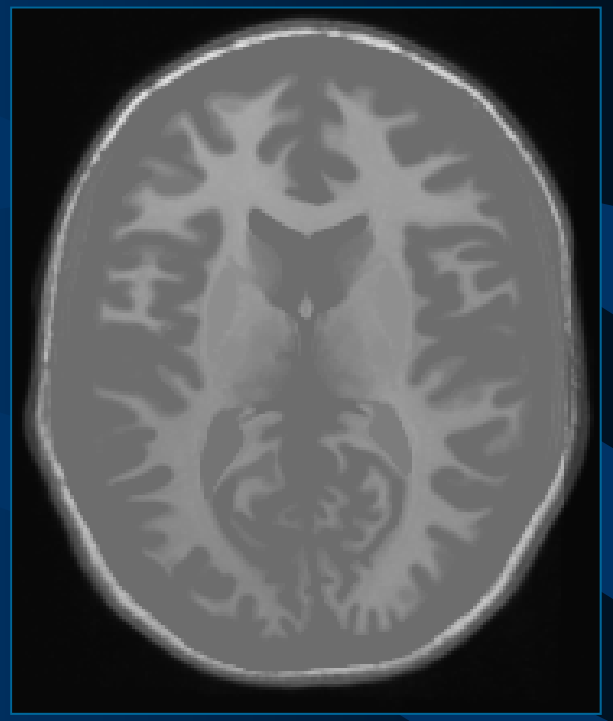
Grayscale Reconstruction



Source

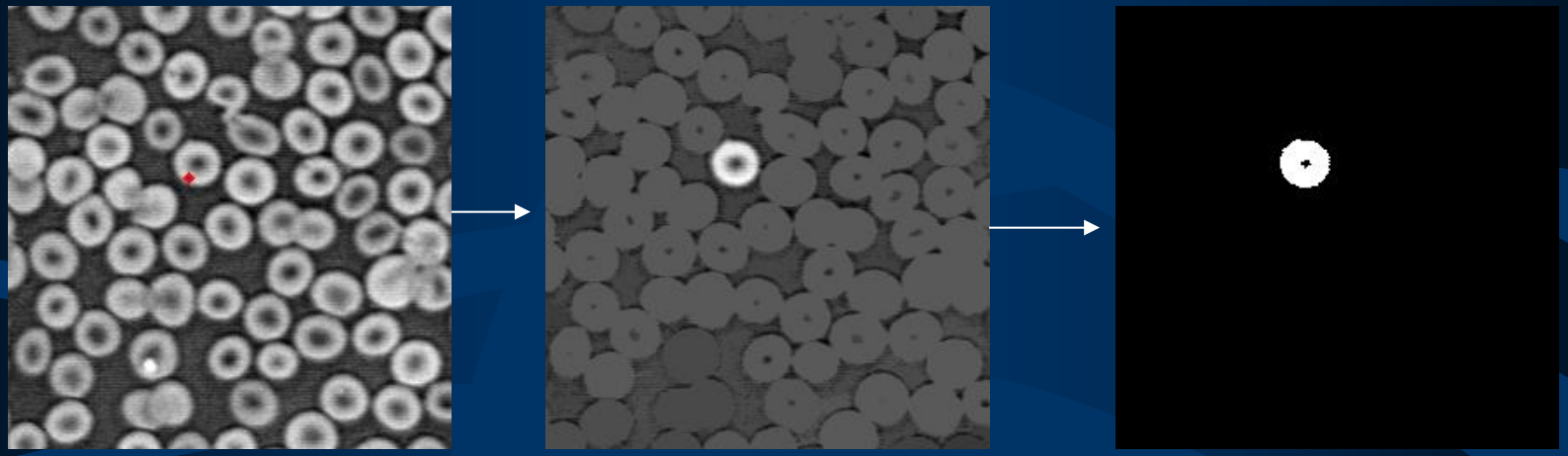


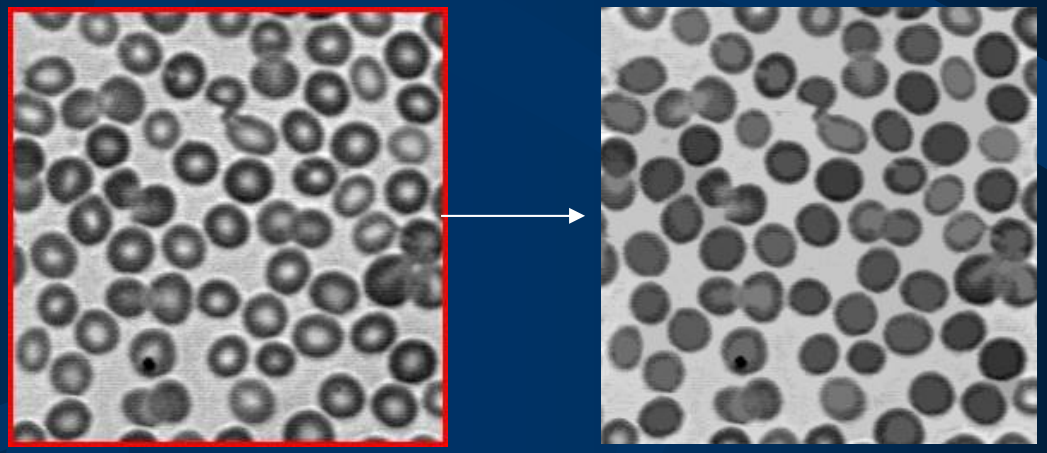
OBR by $r_{10}K_{disk}$



CBR by $r_{10}K_{disk}$

Grayscale Reconstruction





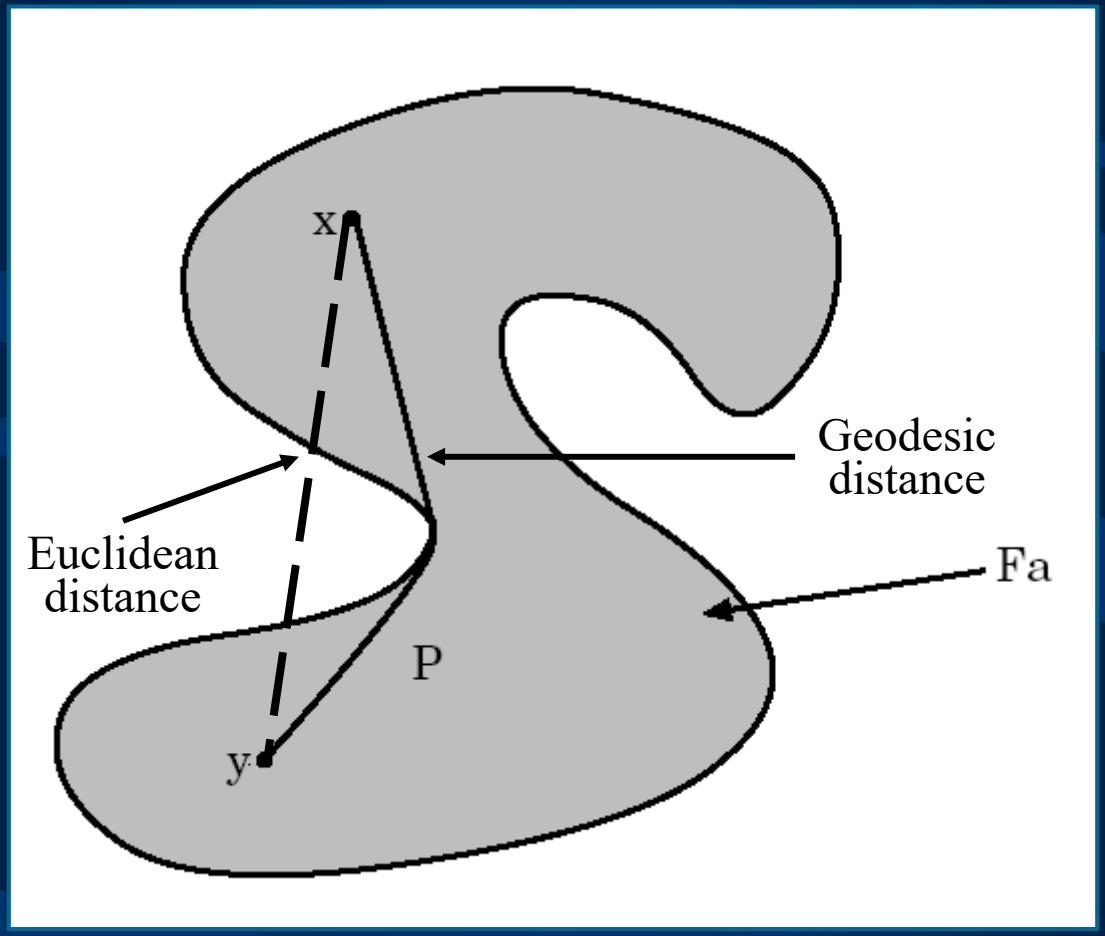
Geodesic Distance

- Geodesic Distance: Let F_a be a connected region. The geodesic distance $d_{Fa}(x,y)$ between two pixels x and y in F_a is the infimum of the length of the paths P which join x and y and are totally included in F_a:

$$d_{F_a}(x,y) = \wedge\{l(P)\}$$

where, "$\wedge$" stands for the infimum and $l(P)$ is the length of the path P.

Geodesic Distance *vs.* Euclidean Distance

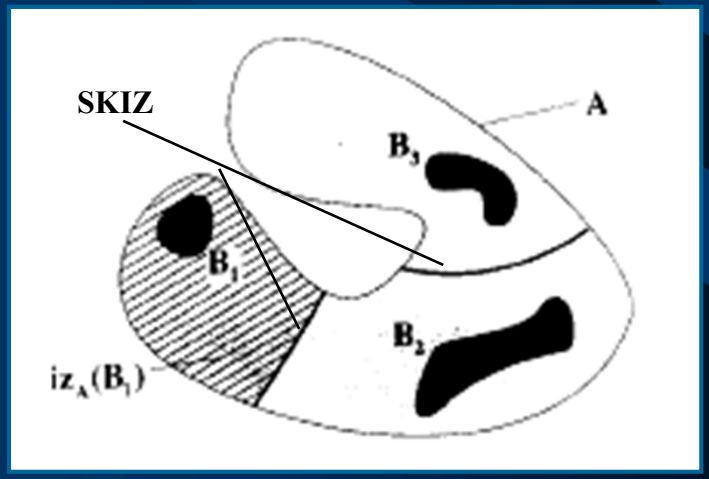


Geodesic Influence Zone

- Geodesic Influence Zone (IZ): Suppose a region A contains a set B made of several connected components $B_1, B_2, \ldots, B_k$. *IZ* is denoted by $iz_A(B_i)$ and defined as:

$$iz_A(B_i) = \{p \in A, \forall j \in [1,k]/\{i\}, d_A(p,B_i) < d_A(p,B_j)\}$$

- $iz_A(B_i)$ is the locus of the points of A whose geodesic distance to B_i is smaller than the distances to any other components of B.



SKIZ

- Skeleton by Influence Zone (SKIZ): the points of A which do not belong to any *IZ*, constitute the *SKIZ* of B inside A, denoted as $SKIZ_A(B)$:

$$SKIZ_A(B) = A / IZ_A(B)$$

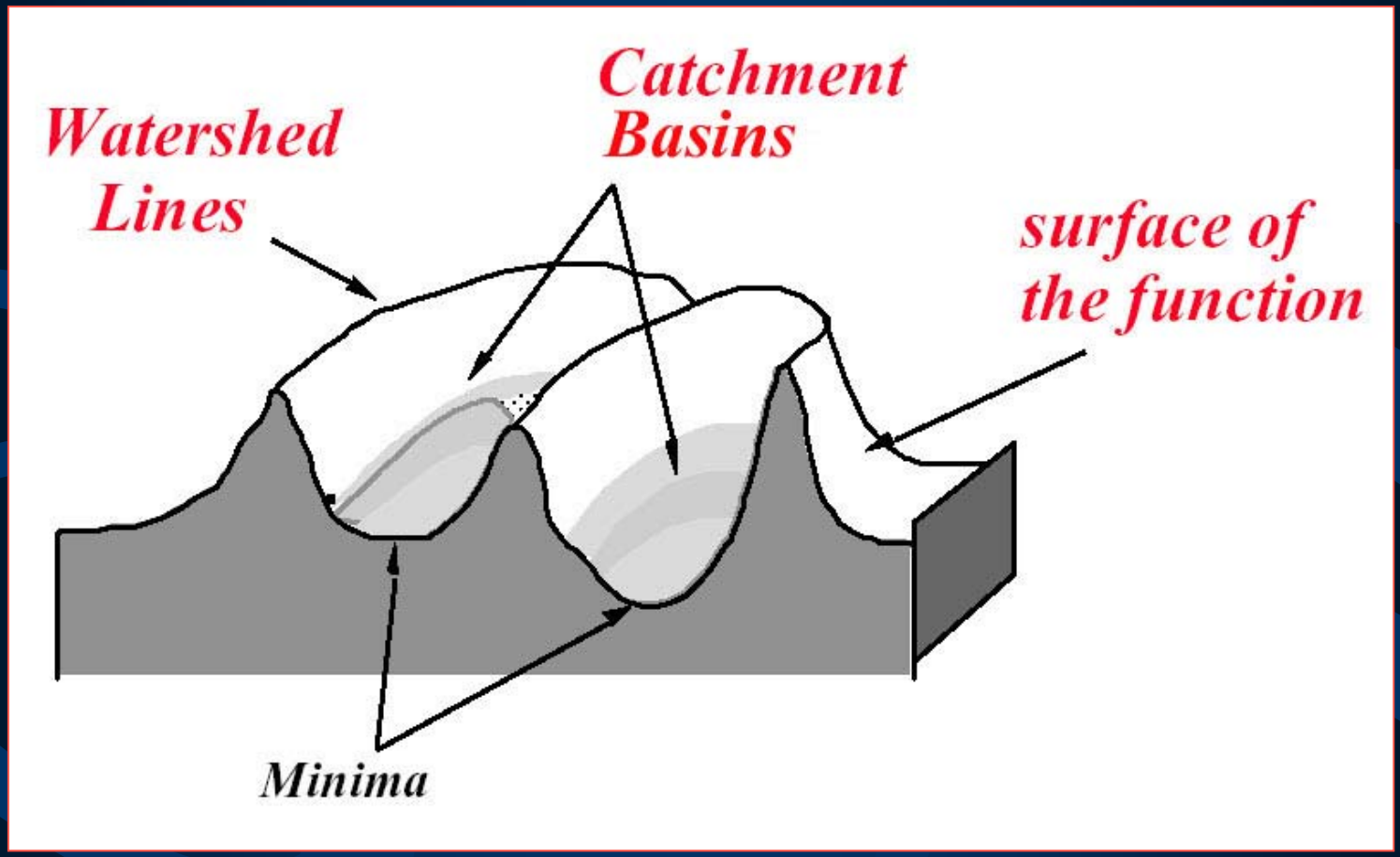
$$with \quad IZ_A(B) = \bigcup_{i \in [1..k]} iz_A(B_i)$$

  - "A/B" means all members in A except those in B $(B \subset A)$

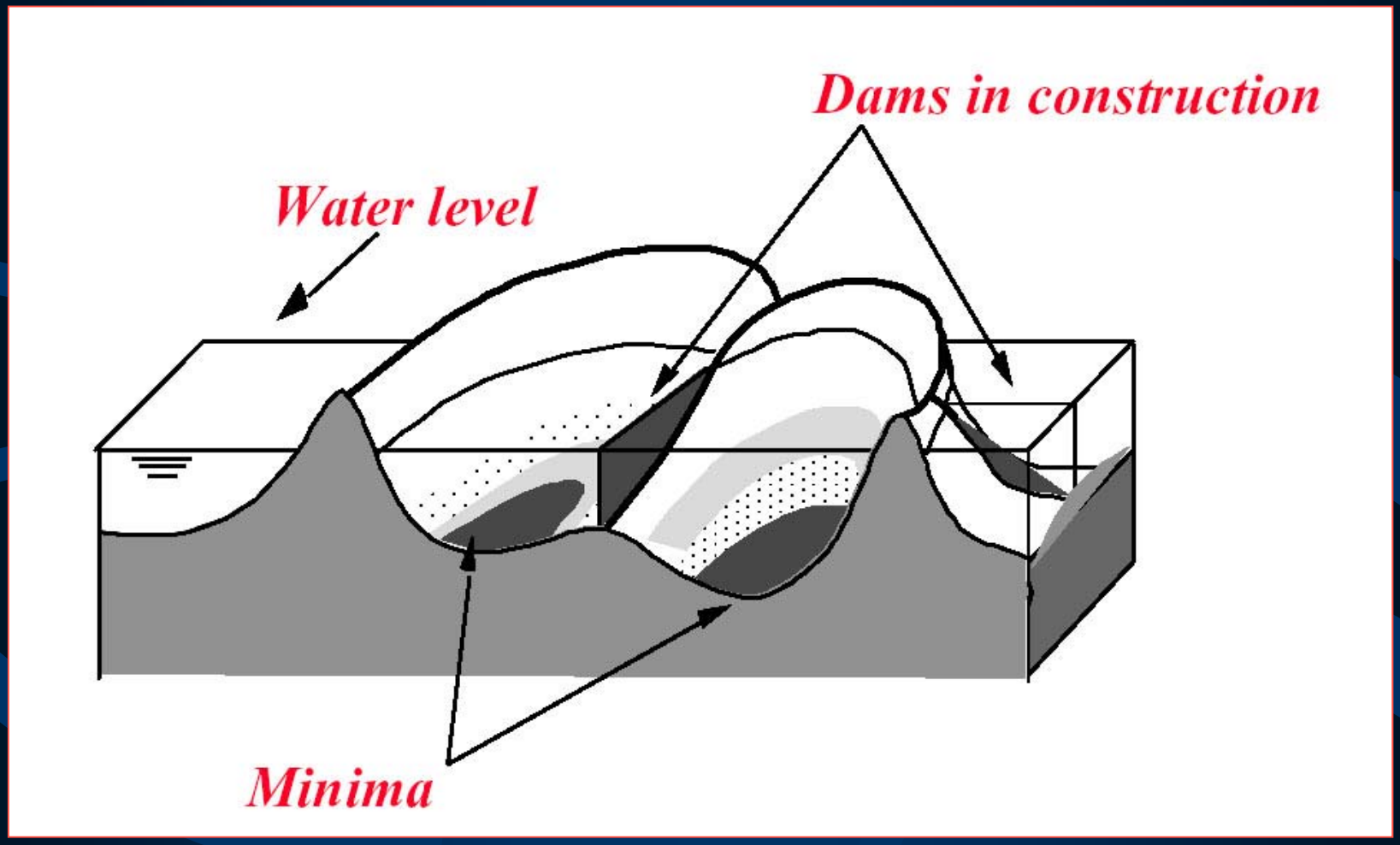
Watershed

The Great Divide

Watershed – Catchment Basin



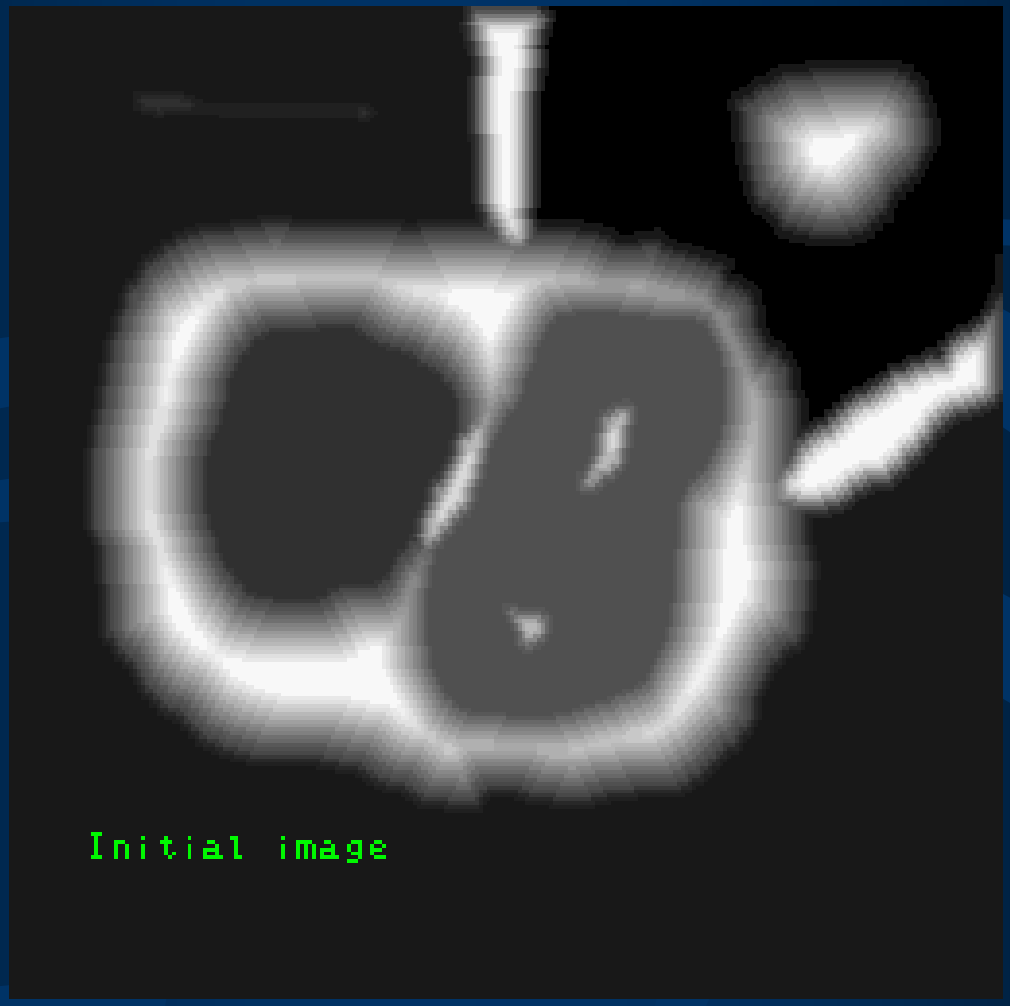
Watershed - Dam



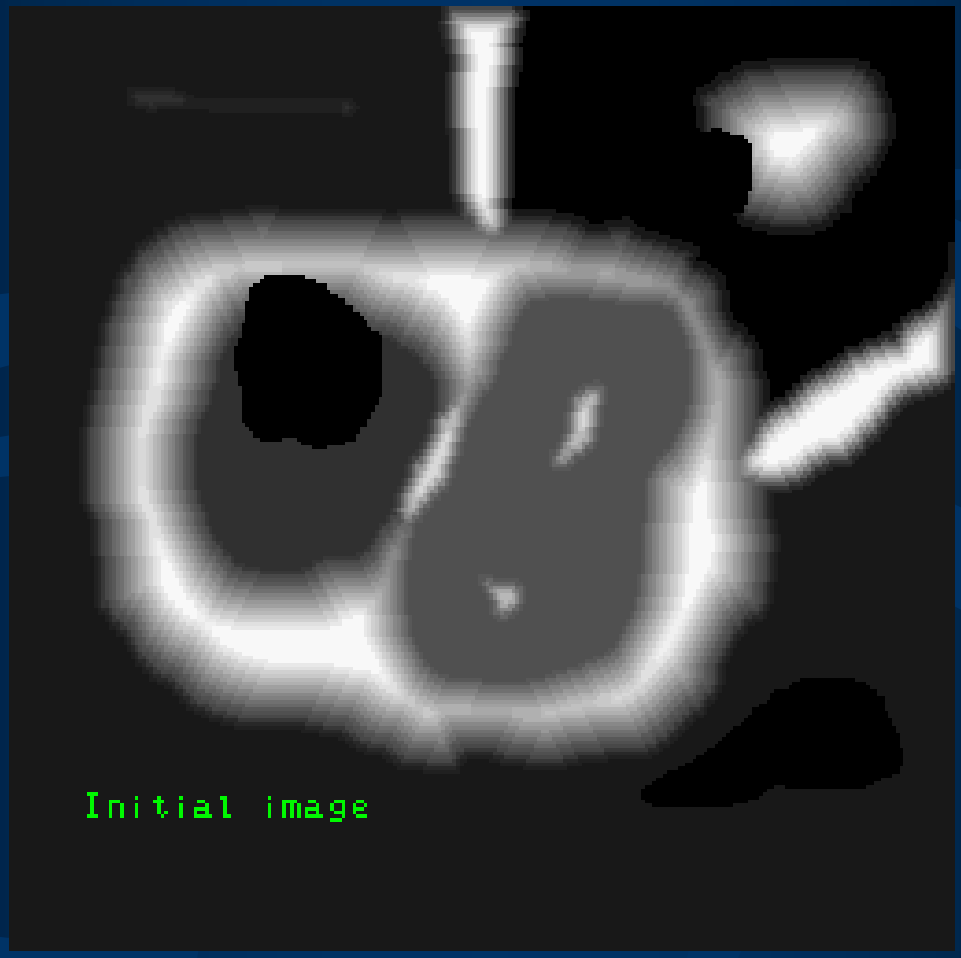
Watershed

- How it works:
  1. Sort all the pixels in an image by their intensities
  2. Find the minimum intensity pixel as the start point (it's gray level as the initial threshold level)
  3. Increase threshold level by 1:
     1) If find another local minimum point, add it to minimum list, and calculate the SKIZ with other existing minimum points
     2) Otherwise, calculate SKIZ within existing minimum list.
  4. Repeat 3 , until all the pixels are sorted to basins or threshold exceed maximum intensity.

Watershed Demo

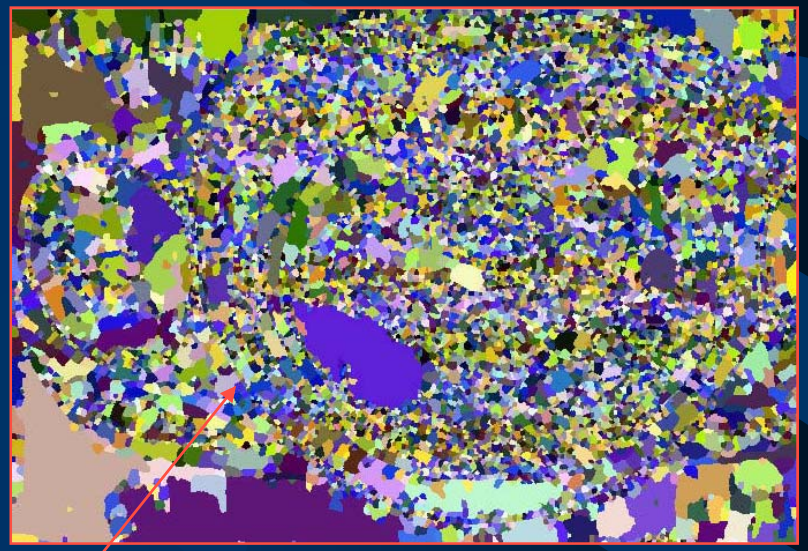


Marked Watershed - Demo



Watershed - Example

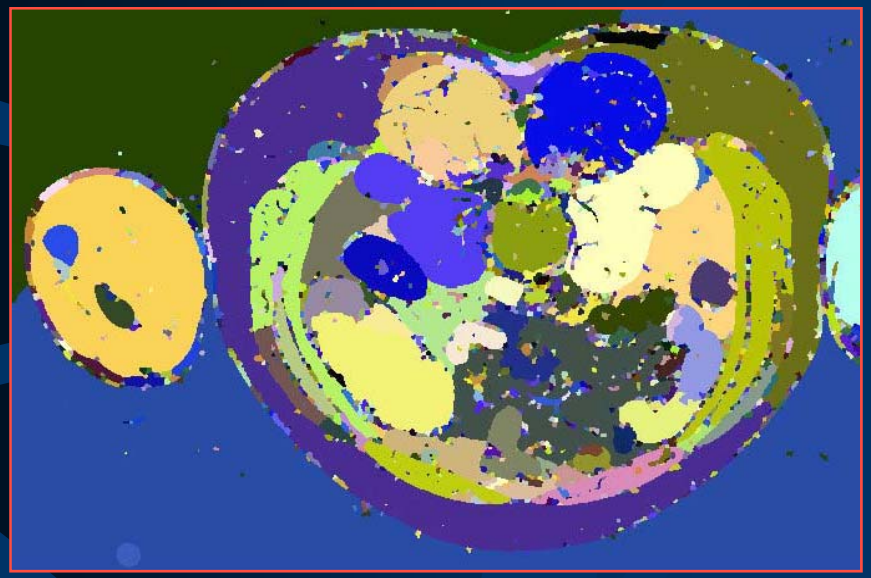
Source Image



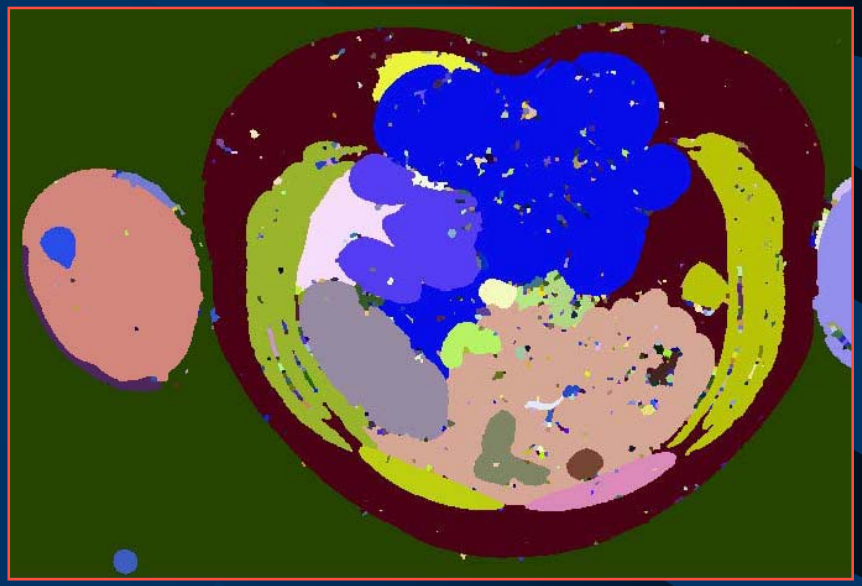
Watershed

Over-segmented

Watershed - Example



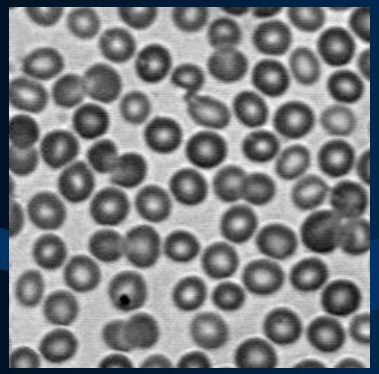
Merged by increase flood level



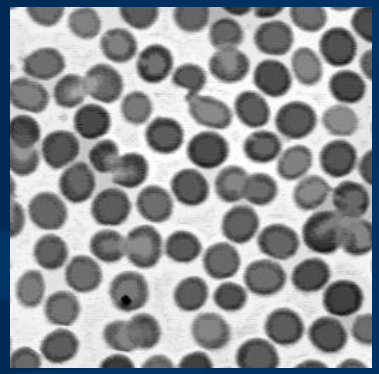
Final Result

Application

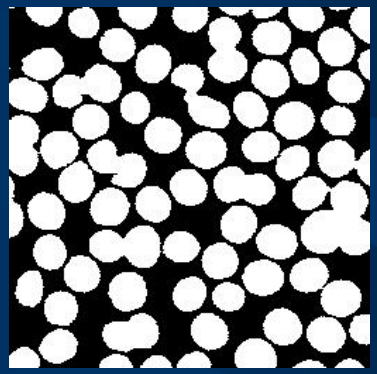
- **Extract blood cells and separate them:**



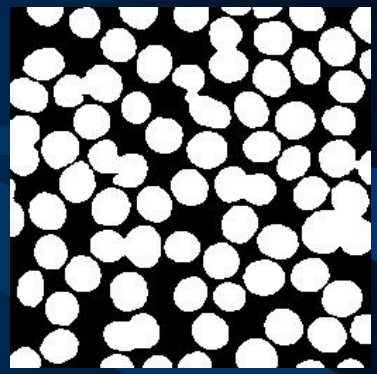
Source



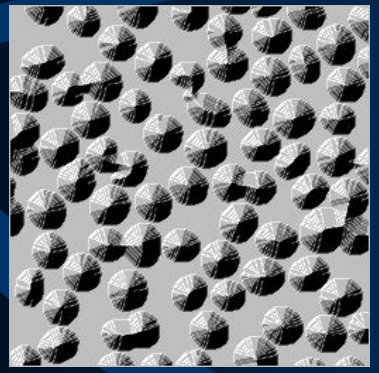
Smoothed



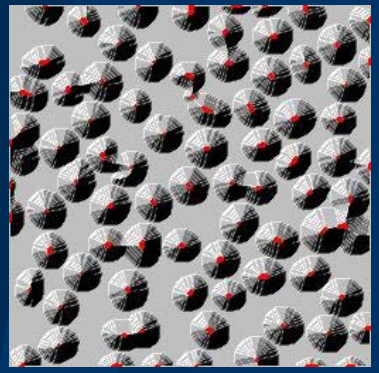
Binary



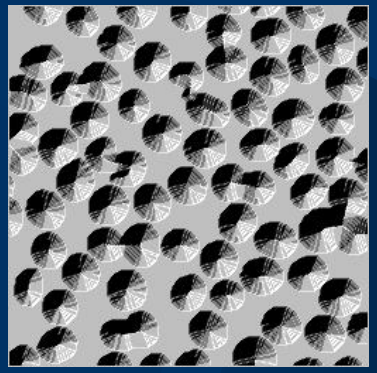
Smoothed



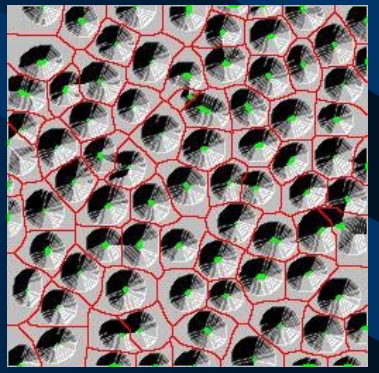
Distance Transform



Maximum points

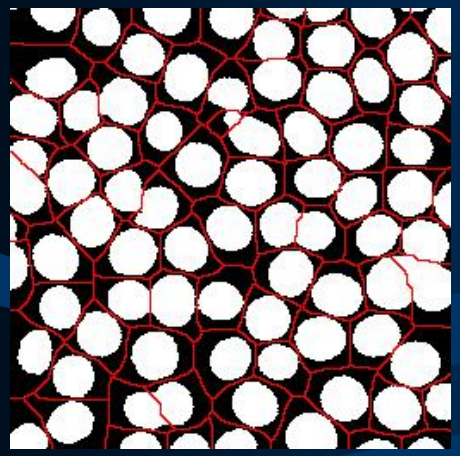


SKIZ

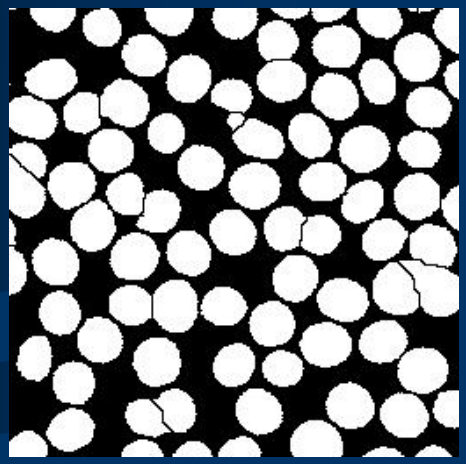


Watershed

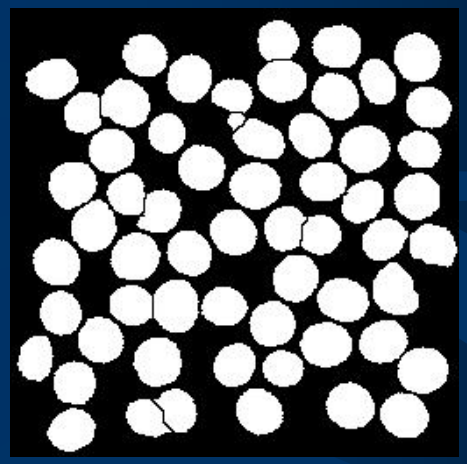
Application



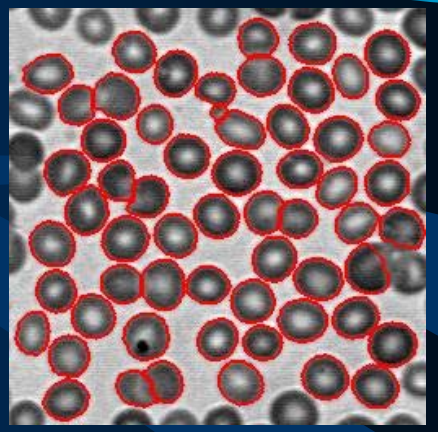
The cell binary image overlay with watershed lines



Cells separated by watershed lines



The Cells that touch the frames are removed



Final

Exercise

Grayscale Operations

$$f = \begin{bmatrix} * & 0 & 2 & 2 & 2 & 1 \\ * & 1 & 2 & 6 & 2 & 1 \\ * & 0 & 6 & 7 & 2 & 1 \\ * & 1 & 1 & 6 & 1 & * \\ * & 1 & 0 & 2 & 2 & 1 \\ * & * & * & * & * & * \end{bmatrix}$$

$$g = \begin{bmatrix} 0 & 3 \\ 3_{\Delta} & 4 \end{bmatrix}$$

Grayscale Operations

$$f \oplus g = \max\{ f_{0,1} + 0,\ f_{0,0} + 3,\ f_{1,1} + 3,\ f_{1,0} + 4 \}$$

$$D_g(f,g) = \begin{bmatrix} * & 0 & 3 & 5 & 5 & 5 & 4 \\ * & 3 & 5 & 6 & 9 & 6 & 5 \\ * & 4 & 6 & 9 & 10 & 6 & 5 \\ * & 3 & 9 & 10 & 11 & 6 & 5 \\ * & 4 & 5 & 9 & 10 & 5 & 4 \\ * & 4 & 5 & 5 & 6 & 6 & 5 \\ * & * & * & * & * & * & * \end{bmatrix}$$

Grayscale Operations

$$f \$ g = \min\{ f_{-1,0} - 4,\ f_{-1,-1} - 3,\ f_{0,0} - 3,\ f_{0,-1} - 0 \}$$

$$E_g(f,g) = \begin{bmatrix} * & -2 & -1 & -2 & -3 \\ * & -3 & 2 & -2 & -3 \\ * & -3 & -2 & -3 & * \\ * & -4 & -3 & -2 & * \\ * & * & * & * & * \end{bmatrix}$$

Grayscale Operations

$$f \circ g = (f \ \$ \ g) \oplus g$$

$$f \circ g = \begin{bmatrix} * & -2 & 1 & 2 & 1 & 0 \\ * & 1 & 2 & 5 & 2 & 1 \\ * & 0 & 5 & 6 & 2 & 1 \\ * & 0 & 1 & 2 & 1 & * \\ * & -1 & 0 & 1 & 2 & * \\ * & * & * & * & * & * \end{bmatrix}$$

Projects

Project1

- Write your own code to realize dilation, erosion,opening and closing operations in grayscale.
- Requirement:
  - Design your own UI and display I/O images
  - Try to apply fast operations in case

Project2

- Write code to realize the next functions:
  - Morphological edge detection
  - Morphological Reconstruction
    - Conditional dilation in binary image
    - Gray scale Reconstruction
  - Morphological gradient
- Requirement:
  - Design your own UI and display I/O images
  - Try to apply fast operations in case

Discussion