Morphological Processing

In graphics, **morphing** is changing forms

In general, ***Morphology*** is the study of forms

⇓

Morphological processing is about manipulating forms/shapes/structures in an image

Morphological operations

- Morphological operations are neighbourhood operations carried out in the spatial domain

- Based on mathematical morphology
  - set theoretical framework
  - originally for binary images
  - extended for grey scale images

- **Applications**
  - extract information about *forms* and *structures*
  - shaping and filtering of forms and structures

Morphological processing

Consists essentially of two steps:

1. *Probe* a given object in x[m,n] with a structuring element (**se**)

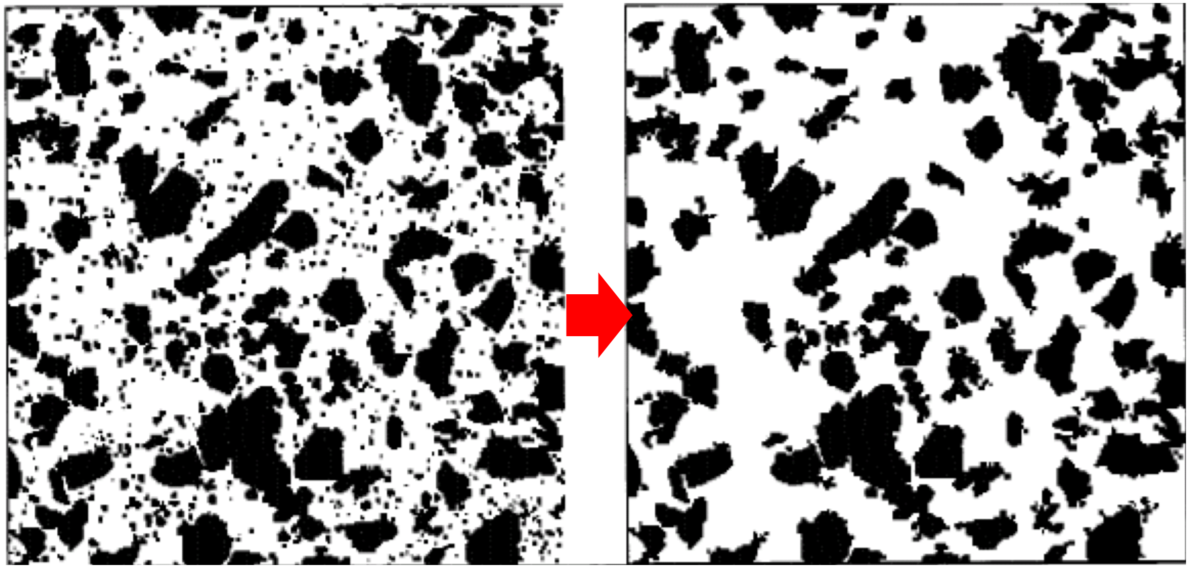
2. Find how the **se** fits with the object

3. Based on the fit do one of the two:
   - **a.** **change pixel values** (hence, the shape) of objects
   - **b.** **extract information** about the form of object

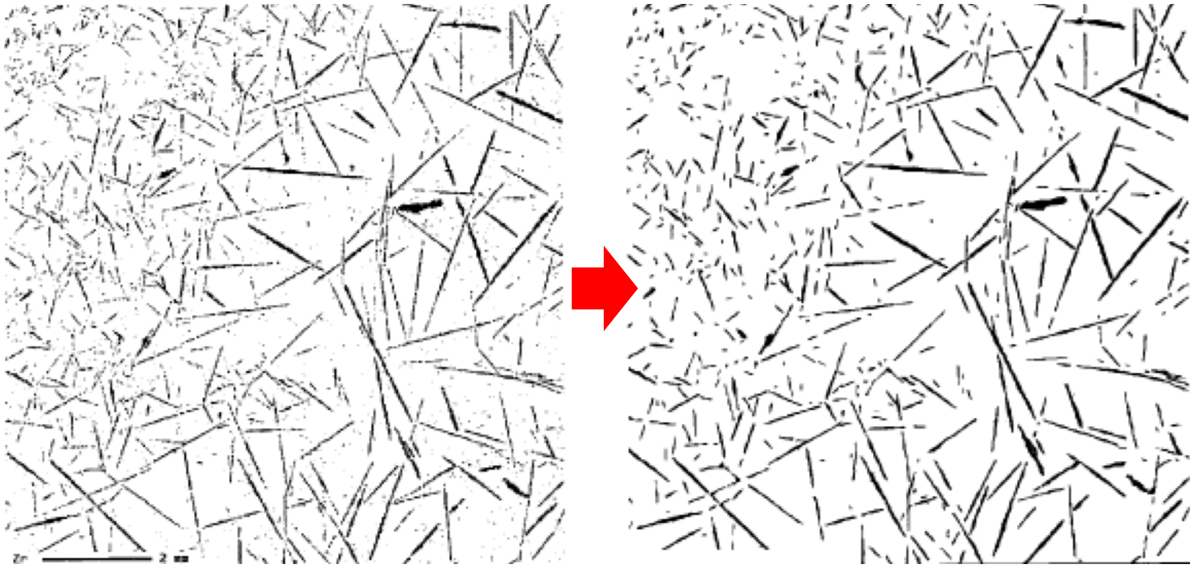
Role of structuring element

- Varying the {size, shape} of **se**
  - yields *different kinds of information* about the object
  - *alters the shape* of the forms in different ways

Applications- Filtering

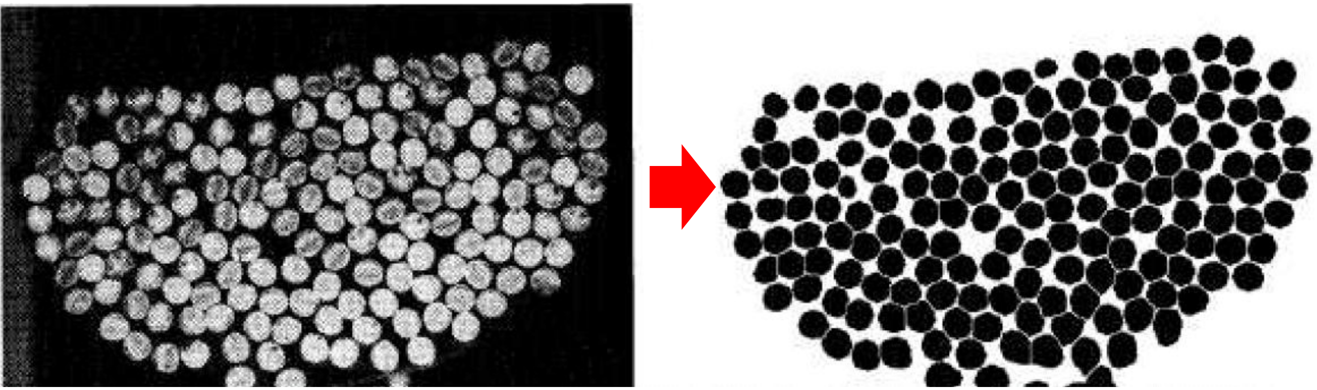


Removal of small blobs

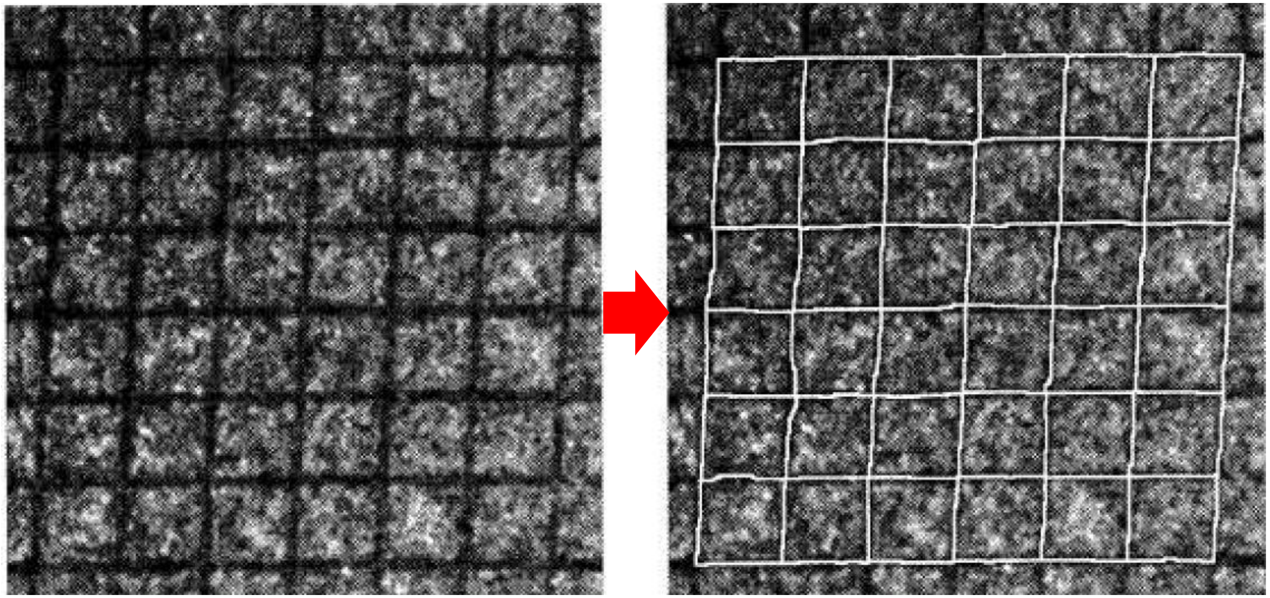


Extraction and grouping of linear objects

Applications- Segmentation

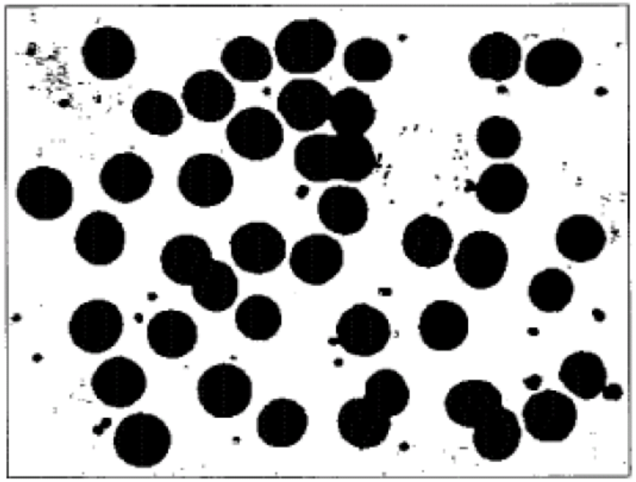


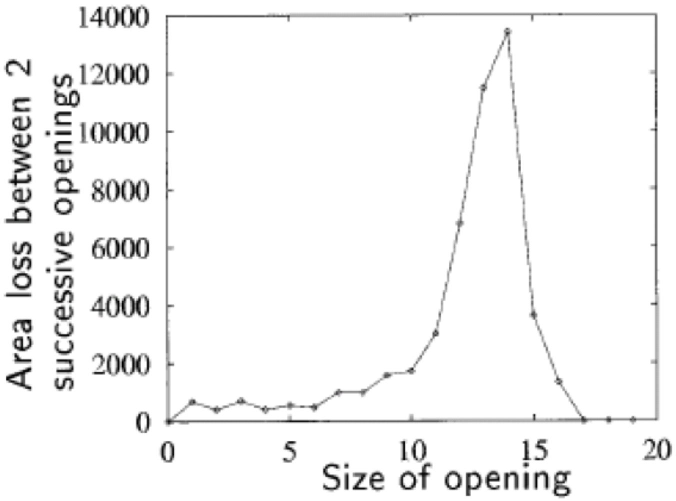
Separation of connected blobs



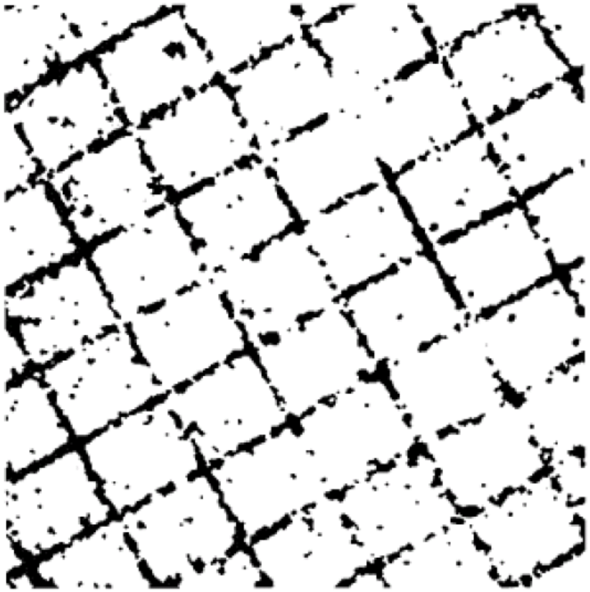
Extraction of grid lines

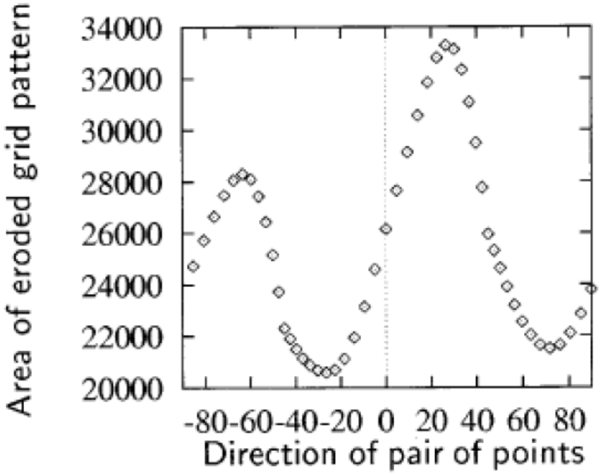
Applications- Measurement





Analysis of connected blobs area

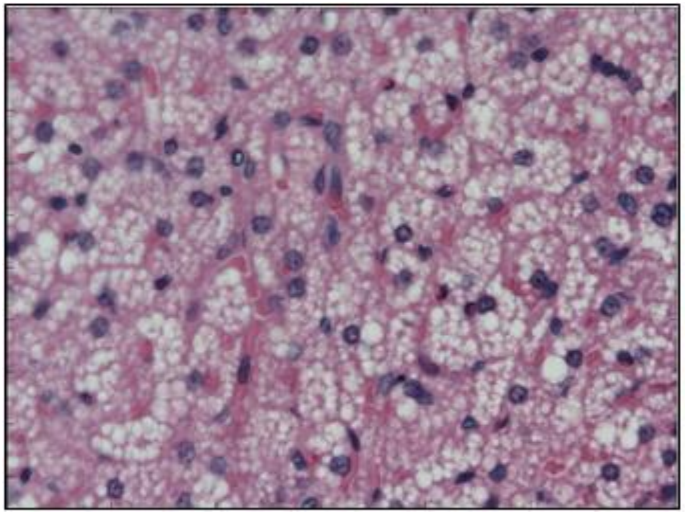


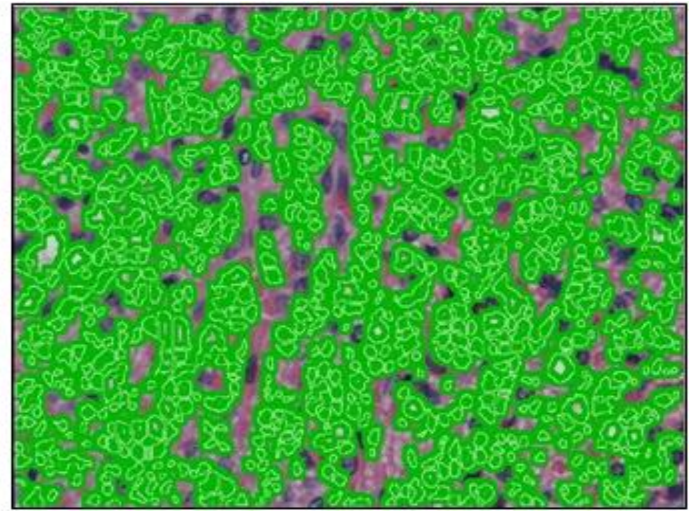


Analysis of line directions

Digital pathology

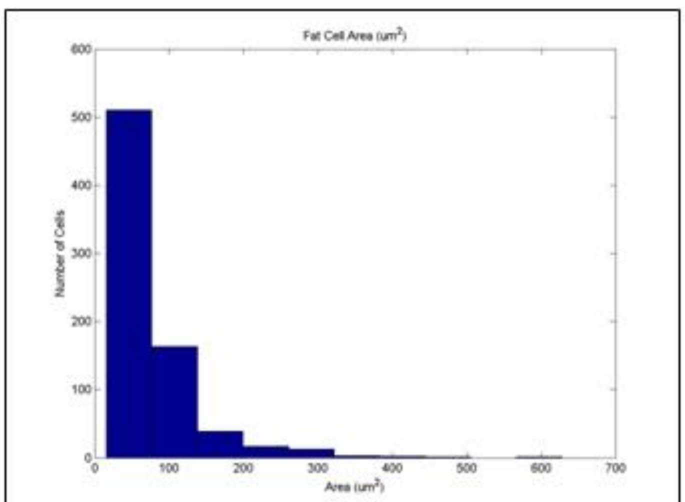
Given image

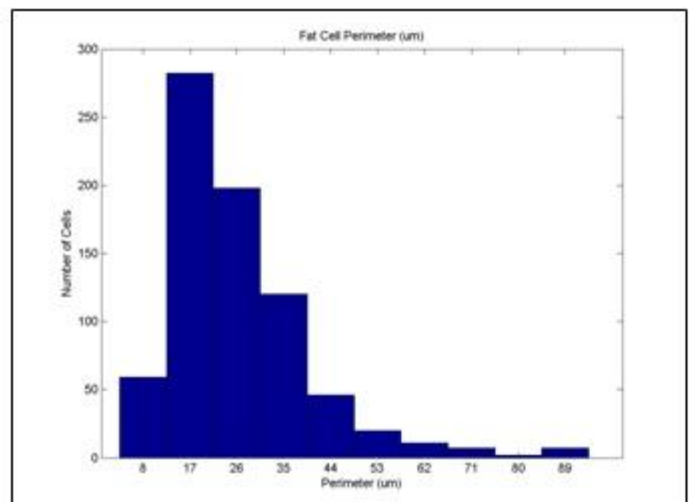




Segmented result

Quantitative analysis of results





Morphological processing on Binary images

Definitions

Given sets $A=\{a\}$, and $B=\{b\}$ and vector x

- Translation by x : $A_x = \{a+x \mid \forall\, a \in A\}$
- Reflection: $-B = \{-b \mid \forall\, b \in B\}$
- Complement: $A^c = \{d \mid d \notin A\}$
- Difference: $A\text{-}B = \{d \mid d \in A,\ d \notin B\} = A \cap B^c$

Examples

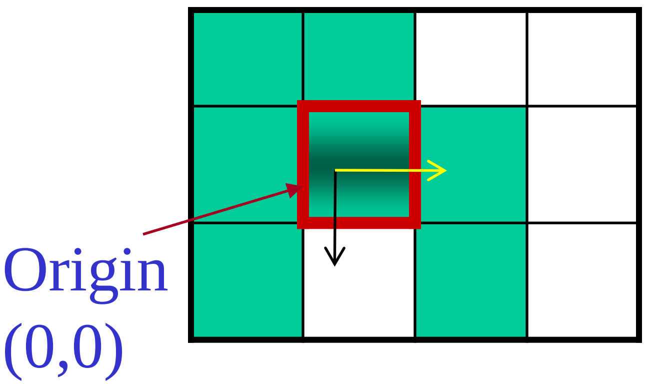
$A_x = \{a+x \mid \forall\ a \in A\}$

$-B = \{-b \mid \forall\ b \in B\}$

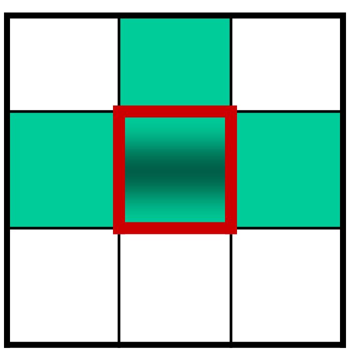
Difference: $A - B = \{d \mid d \in A,\ d \notin B\} = A \cap B^c$

Note: green pixels are object pixels

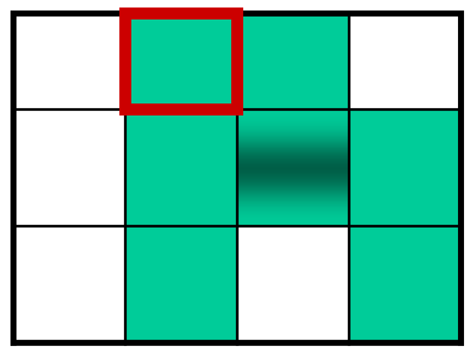
Set A



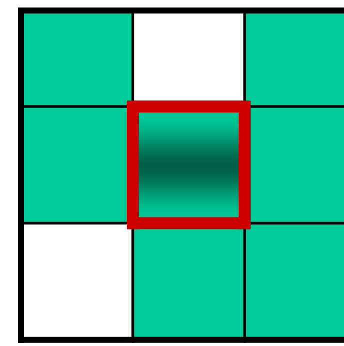
Set B

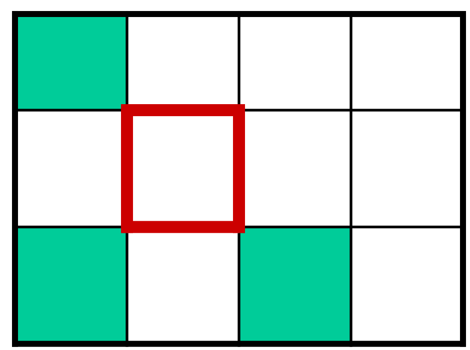


$A_x \mid x = (1,1)$



-A





A- B

Basic operations

- Erosion
- Dilation
- Erosion followed by dilation = Opening
- Dilation followed by erosion = Closing

Basic operations - definitions

Erosion – shrinks an object

$$X \ominus B = \{x \mid B_x \subseteq X\}$$

- Set of all locations x such that **se** is within X

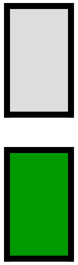
Dilation – enlarges an object

$$X \oplus B = \{x \mid -B_x \cap X \neq \phi\}$$

- Set of all locations x such that **-se** has at least 1 pixel within X

Examples

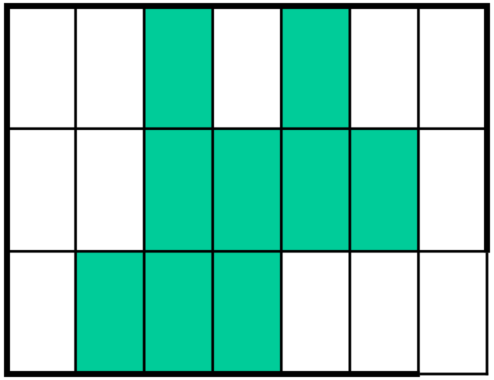
$\{x \mid B_x \subseteq X\}$



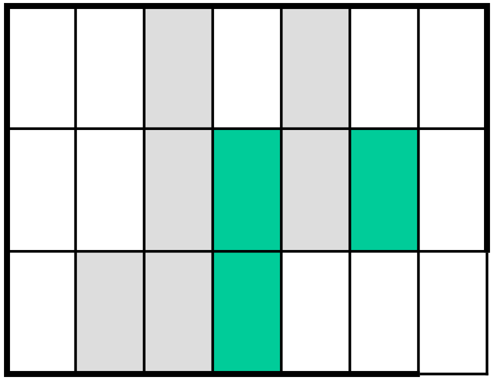
Eroded pixel

Dilated pixel

Object A

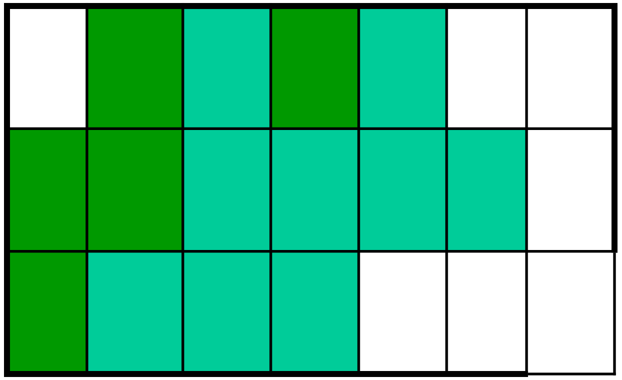


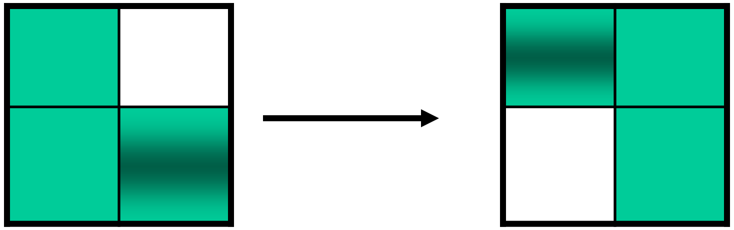
After erosion



$\{x \mid -B_x \cap X \neq \phi\}$

After dilation

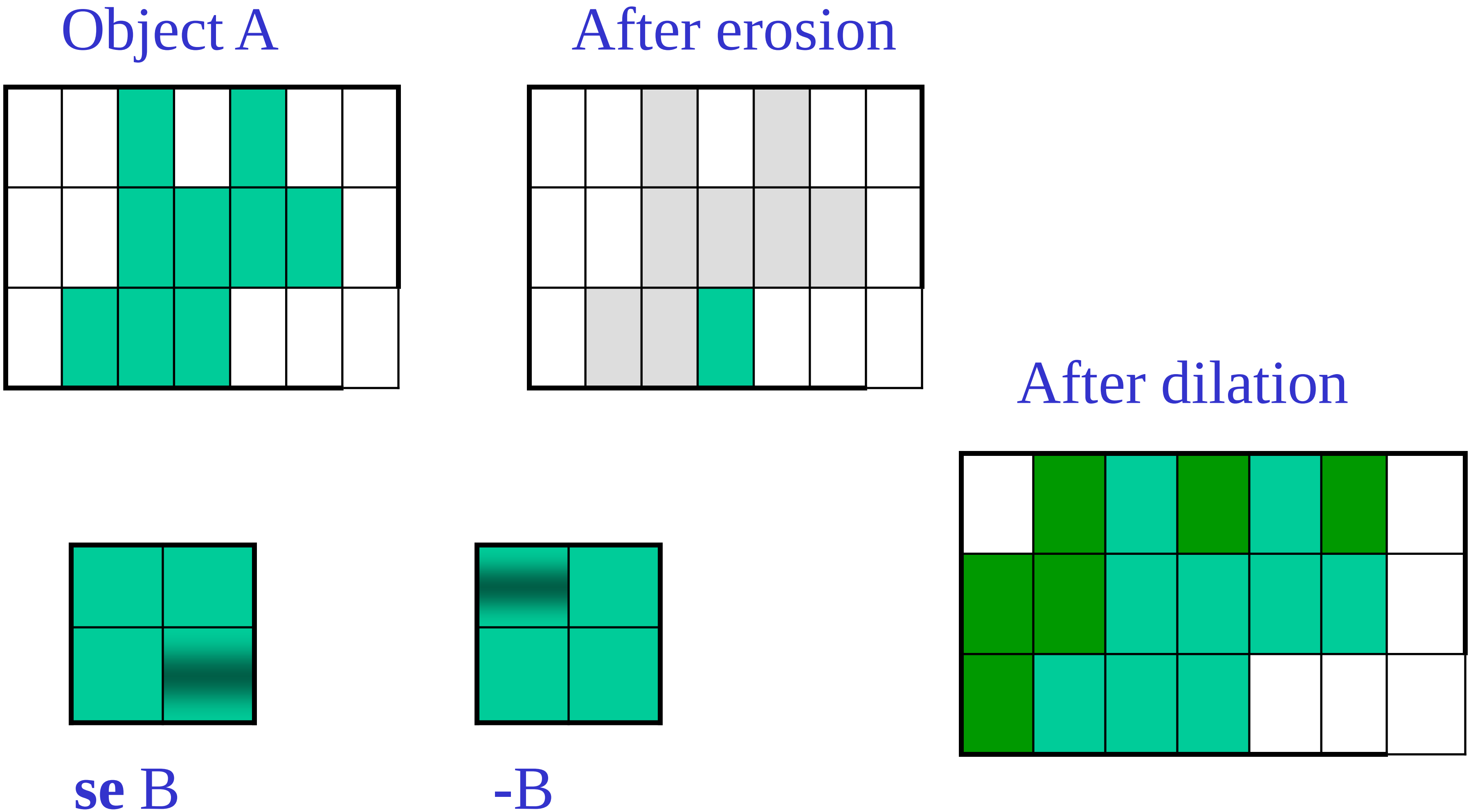




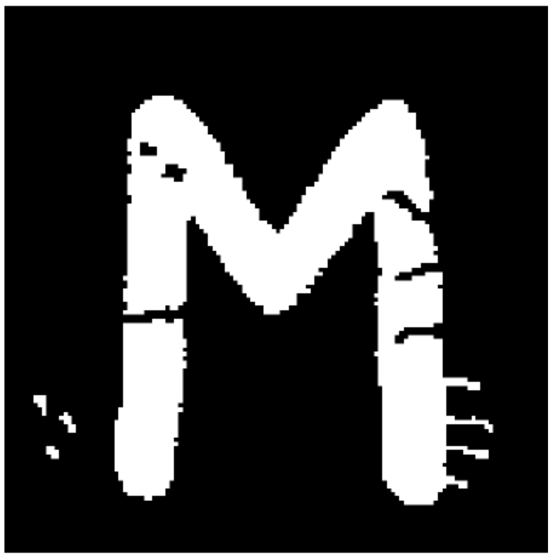
se B

-B

Effect of changing se



Binary image example



Dilation
(expansion)

Erosion
(contraction)

White pixels represent object

Basic operations - definitions..contd.

- **Opening** – erosion followed by dilation

$$(X \ominus B) \oplus B = X \circ B$$

  - Smoothes contours, fills in small islands

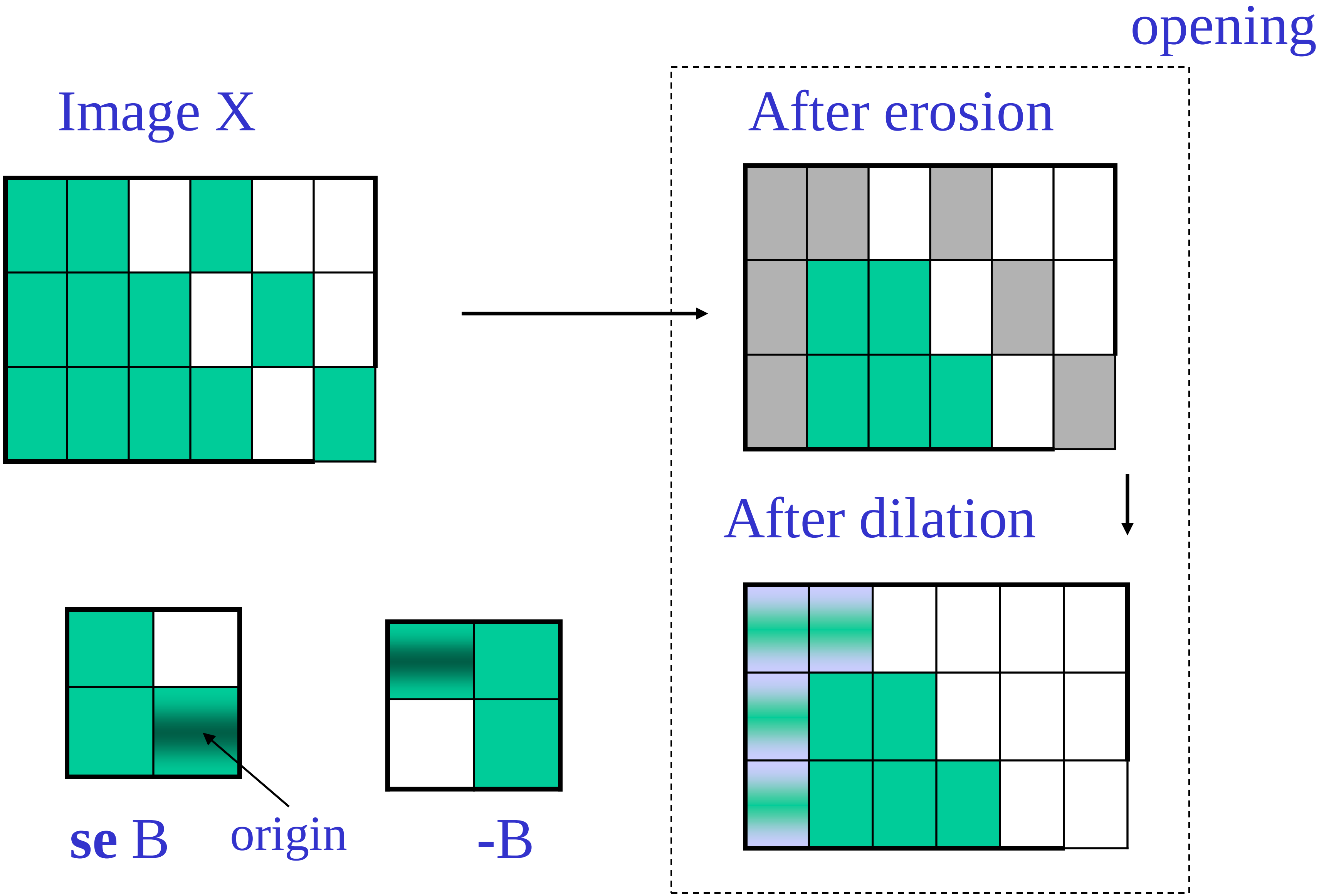
- **Closing** – dilation followed by erosion

$$(X \oplus B) \ominus B = X \bullet B$$

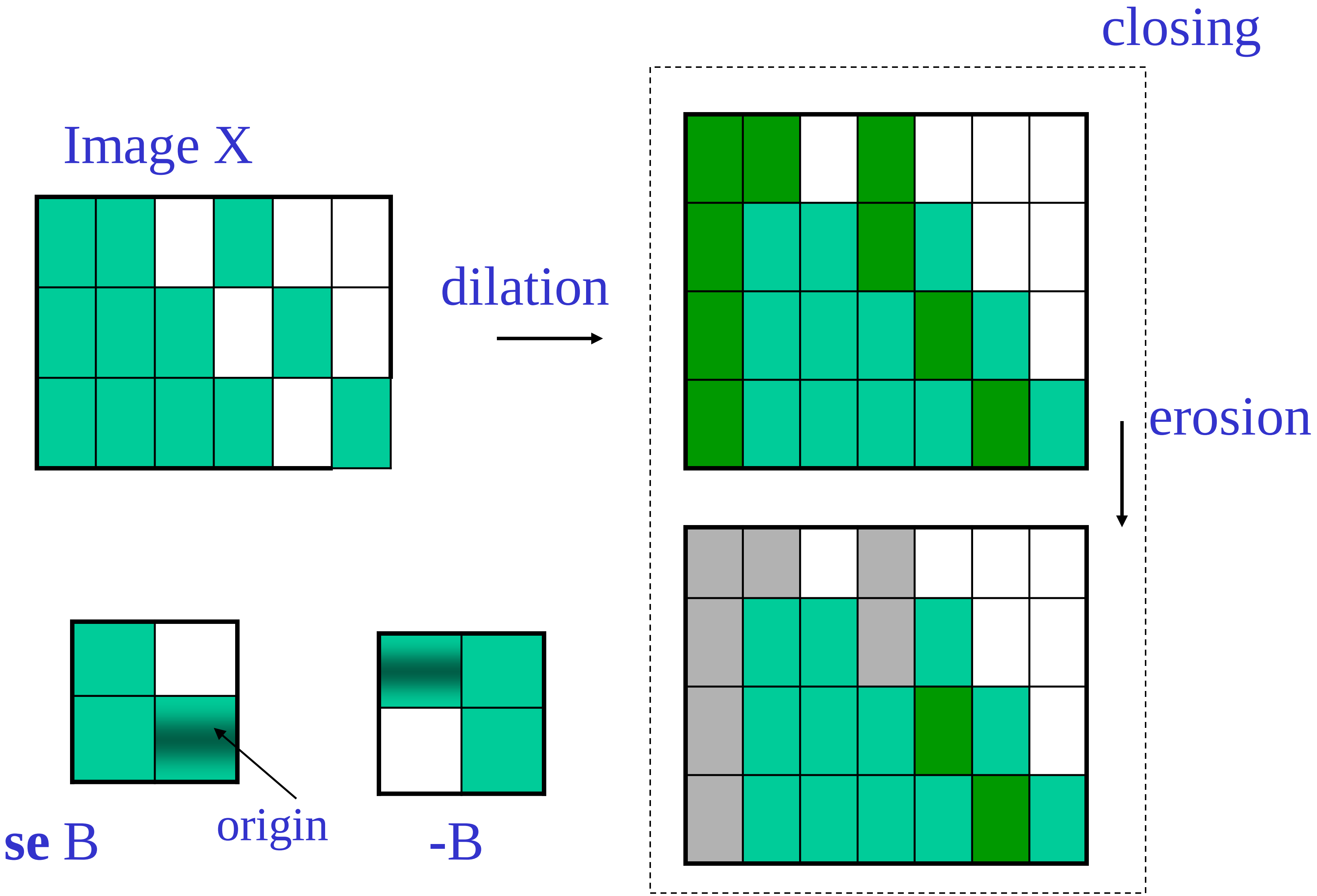
  - Blocks narrow channels and thin holes

Opening - Example

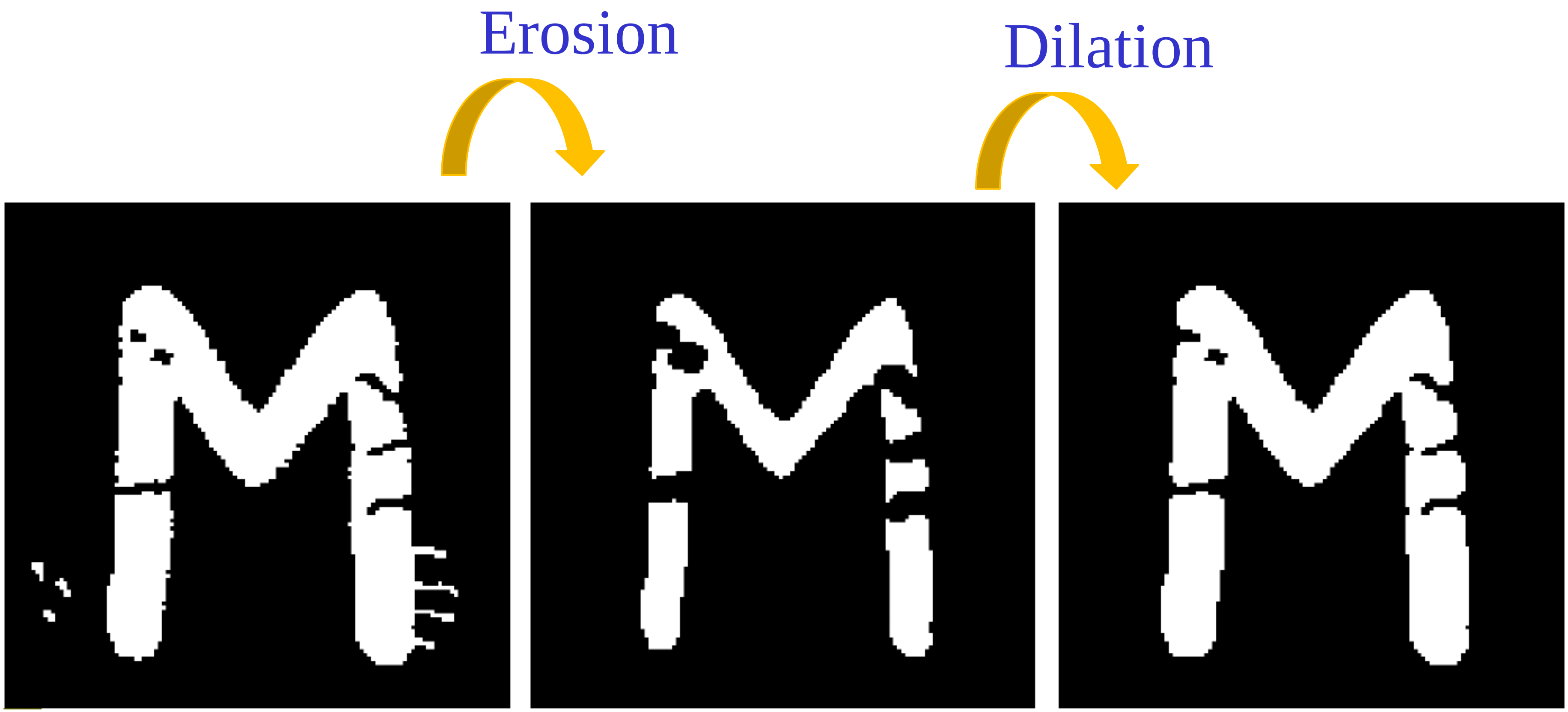
Eroded pixel



Closing - example



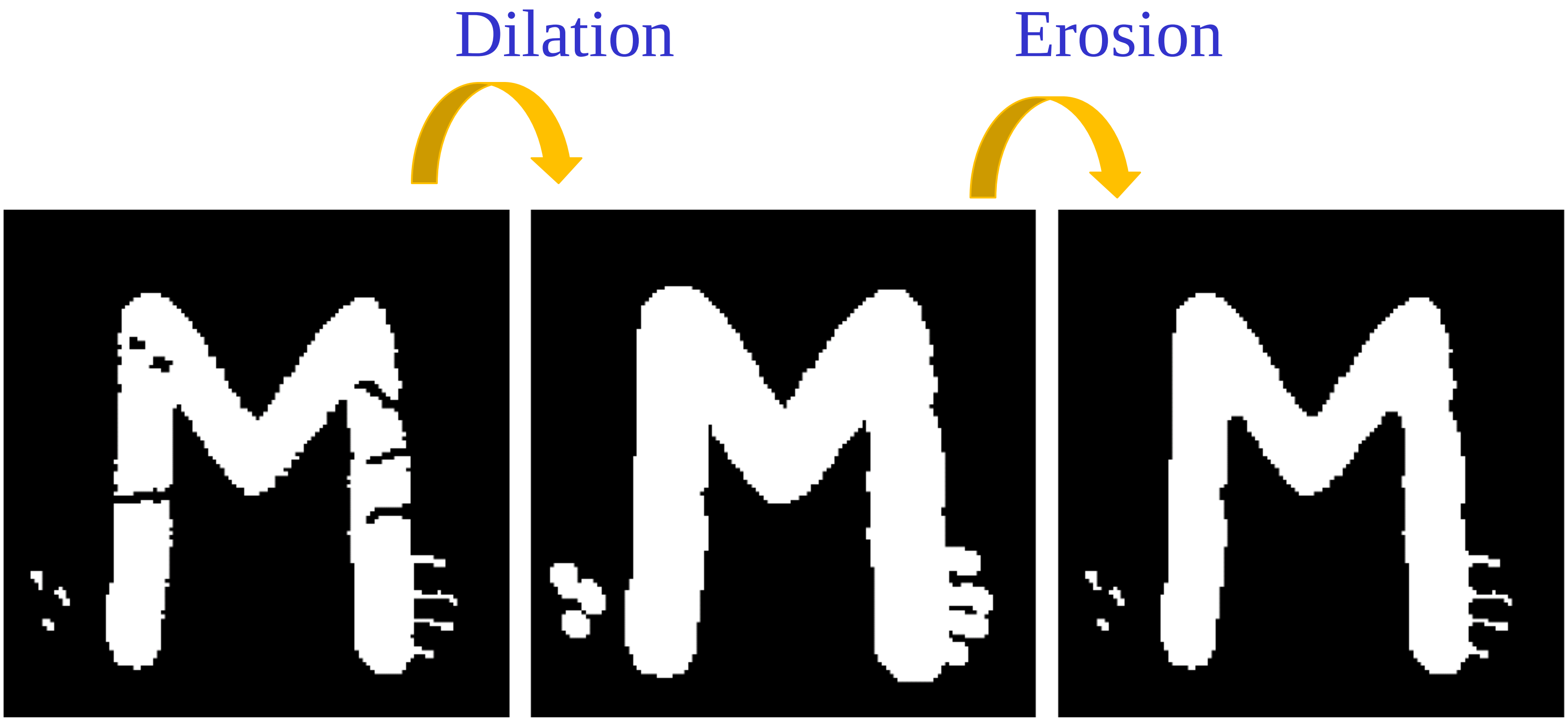
Opening = Erosion + Dilation



Opening Effect:

- Remove small objects and spurs
- Reset size of the object

Closing = Dilation + Erosion



Closing Effect:

- Fills small holes and cracks
- Reset size of the object

Properties for Opening and Closing

$$A \circ B = (A \Theta B) \oplus B$$

$A \circ B$ is a subset of A

$(A \circ B) \circ B = A \circ B \leftarrow$ *Repeated opening has no effect!*

$$A \bullet B = (A \oplus B) \Theta B$$

A is a subset of $A \bullet B$

$(A \bullet B) \bullet B = A \bullet B \leftarrow$ *Repeated closing has no effect!*

Hit or Miss transform

$$X \circledast B = (X \ominus B_1) \cap (X^c \ominus B_2)$$

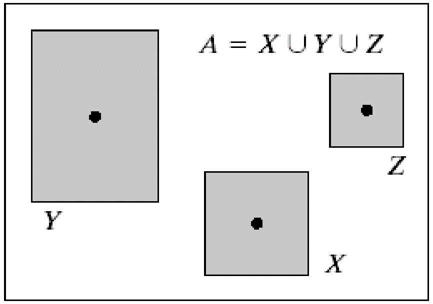
B = (B_1, B_2)

B_1 : object part

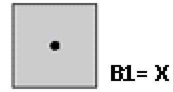
B_2 : local background

HMT Algorithm (uses two *se* **):**

1. Find where the object part (*se*1) ‘fits’ *X*
2. Find where the local background (*se*2) ‘fits’ X^C
3. Take their intersection



B_2



Set A ={X, Y, Z}

- Objective: identify center of shape X

Define Window W enclosing shape X

B_1= X ***object***

B_2= (W-X) ***background***

1. **Erode A with B_1**
2. **Complement (A) and erode with B_2**

Find the intersection of results of 1 and 2

$$A \circledast B = (A \ominus X) \cap [A^c \ominus (W - X)]$$

Find shape B in A

Find matches for B_1

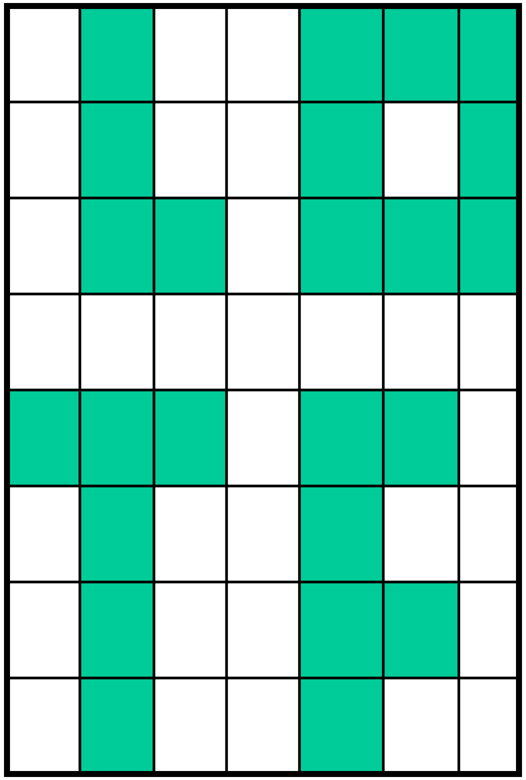
Find matches for local Background B_2

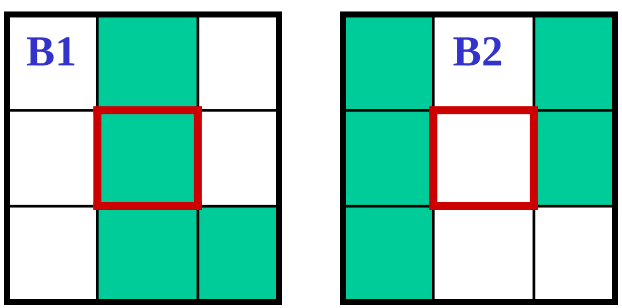
Hit or miss transform - example

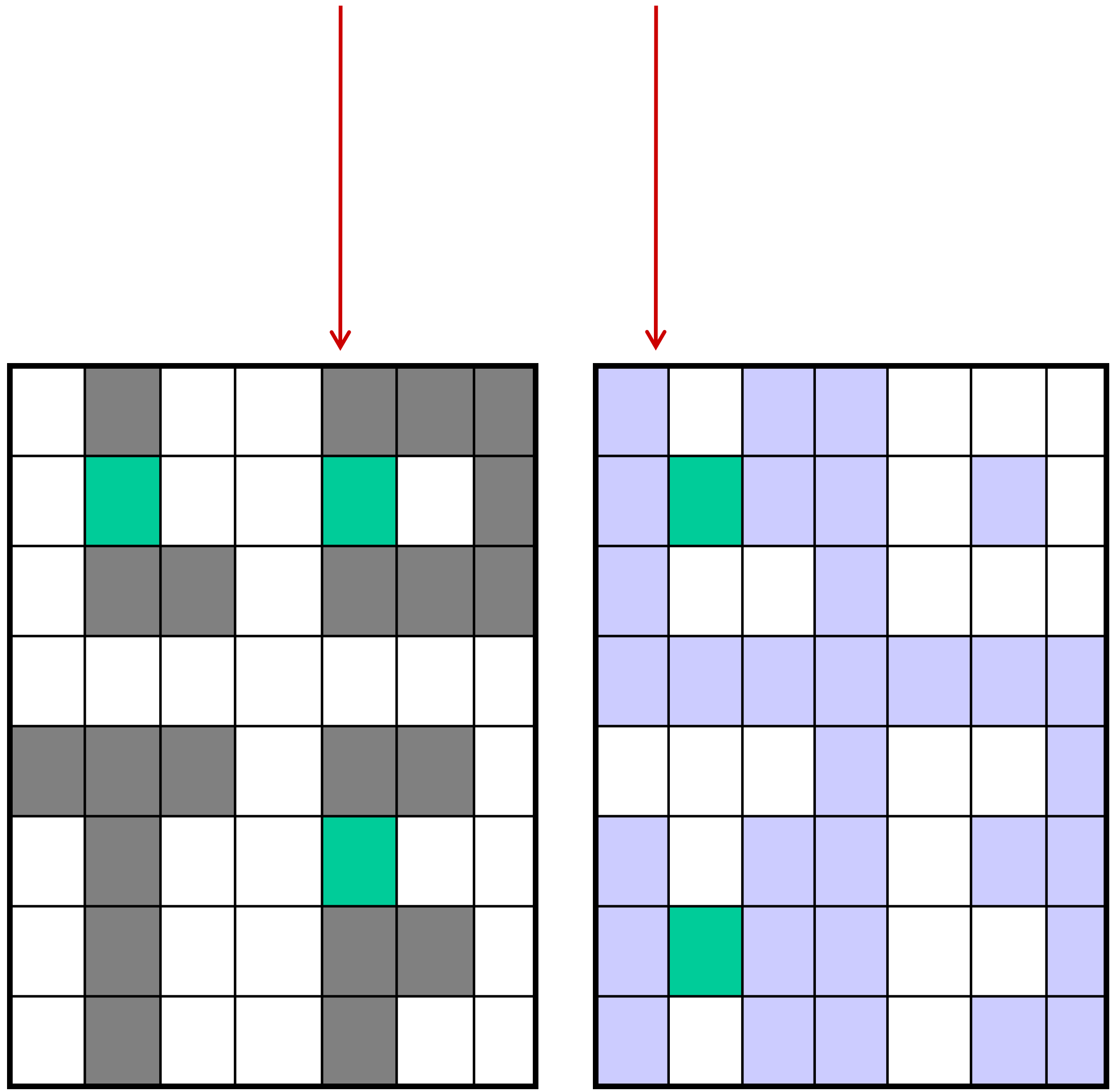
A : all green pixels

A^c : all white pixels

$$A \circledast B = (A \ominus B1) \cap (A^c \ominus B2)$$

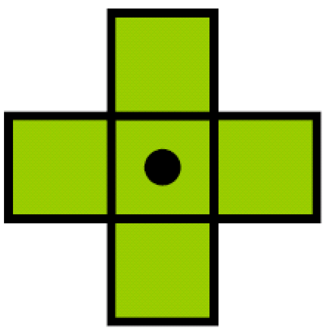




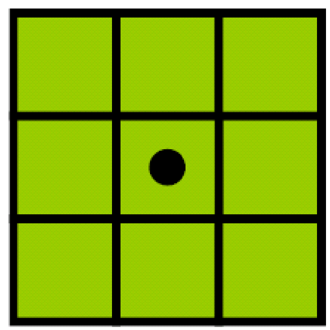


Some common Structuring Elements

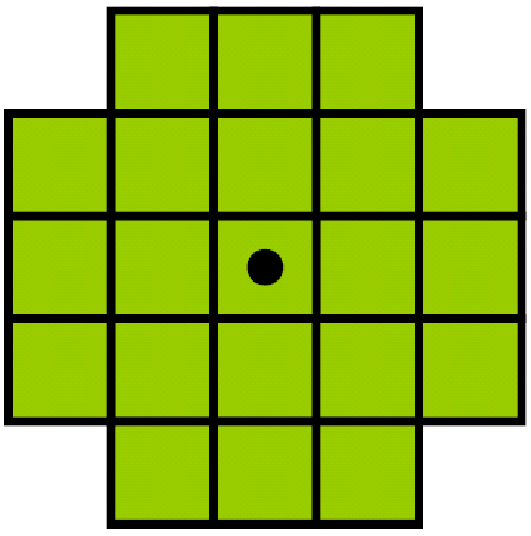
- se can be viewed as a binary filter kernel



d(4)



d(8)



d(oct)

The origin is marked with a point.

MORPHOLOGICAL ALGORITHMS

Task of interest

Detect and recognise objects

1. Based on shape of an object
   - Extract object's contour or boundary

2. Based on a minimal representation for a shape
   - Extract its skeleton

Solution: Design algorithms using basic operations of erosion, dilation, opening, closing, HMT

Morphological algorithms

- **Boundary extraction**: difference between object and its eroded version

$$\beta(X) = X - (X \ominus B)$$

  - This extracts internal boundary

$$\beta(X) = (X \oplus B) - X$$

  - This extracts internal boundary

Boundary detection $X-(X \ominus B)$

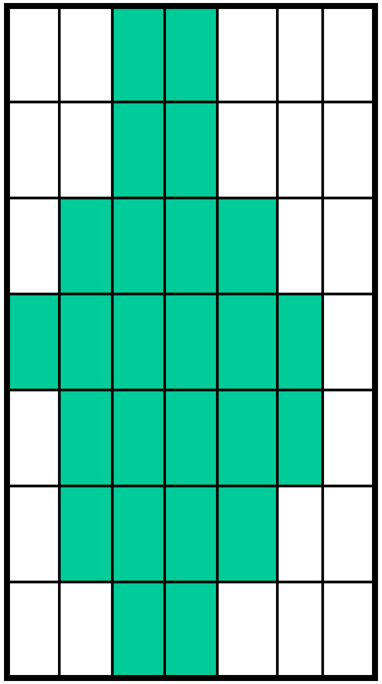
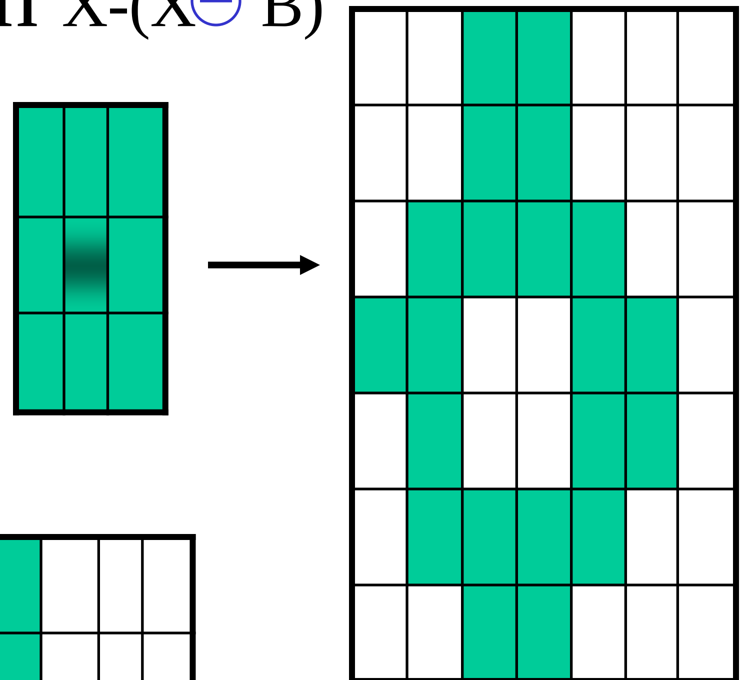


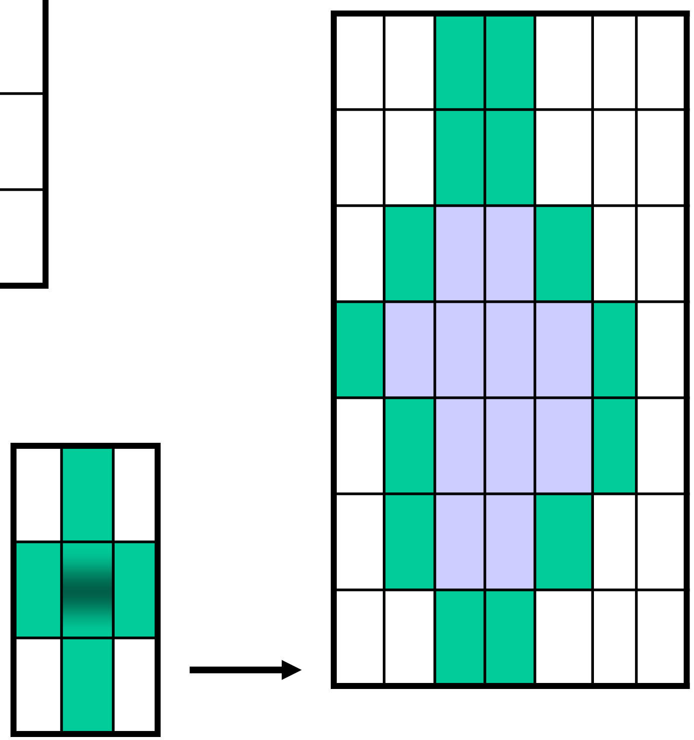
Image X

With B_1



$\beta_1(X)$

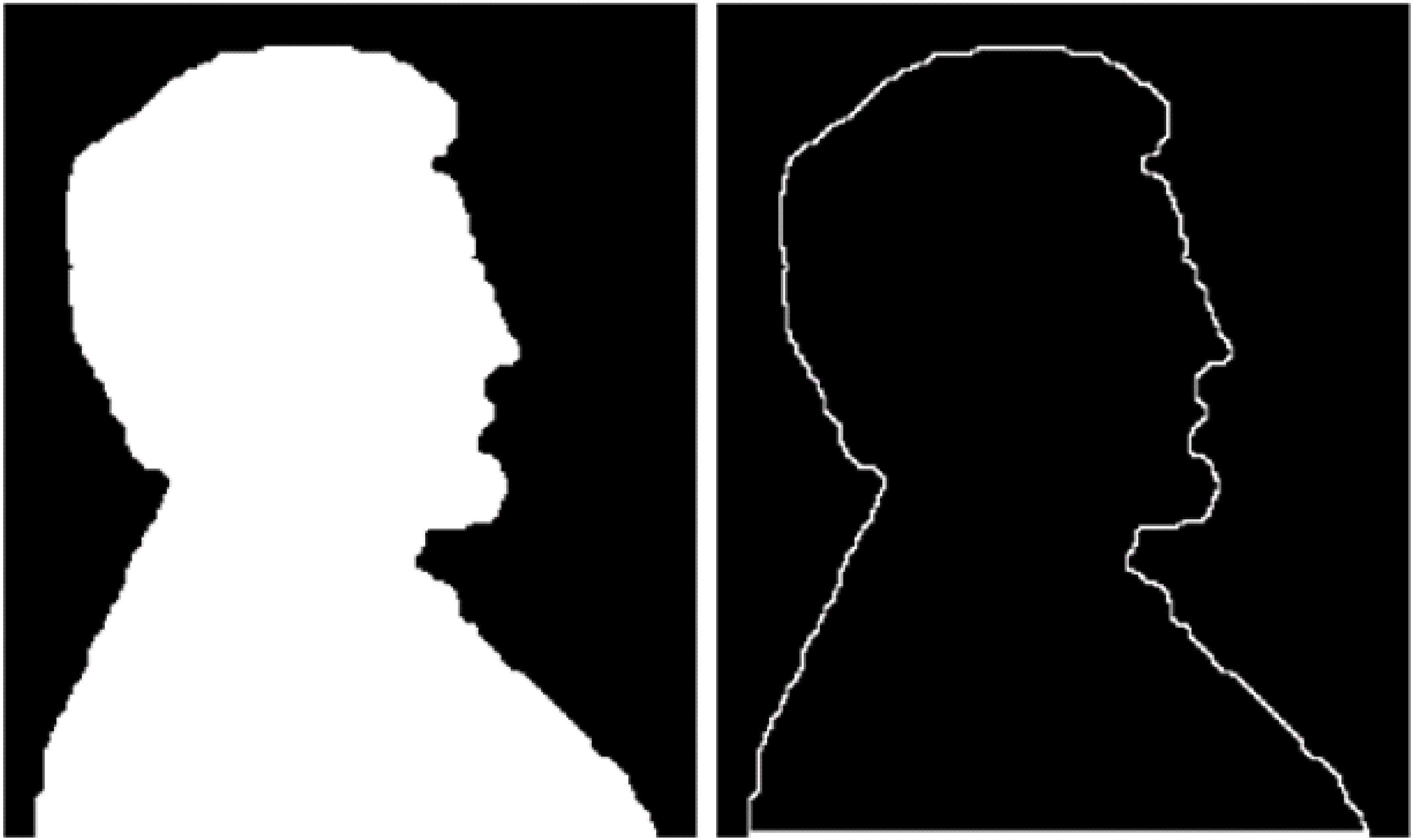
With B_2



$\beta_2(X)$

Centre pixels are those retained after erosion of X by B

Binary example of boundary extraction



Thinning and thickening

- **Thinning**: $X \otimes B = X - X \circledast B$

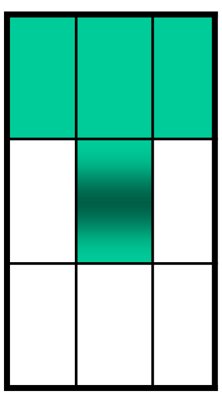
- **Thickening**: $X \cup X \circledast B$
  - Can be done by thinning the background

Skeletonisation

- A skeleton is a *minimal* representation for structures
- Provides topological and metric information
  - End points, nodes, holes; branch length, angles, etc
- Several approaches exist

Methods for skeletonisation

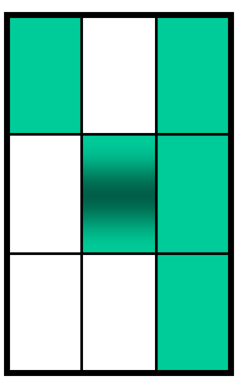
- **Basic approach** – conditional erosion
  - remove pixels only if they don't split a region
  - can use a "fate table" for implementation



Removal of centre pixel

Splits a region

Does not split a region



Methods for skeletonisation

Method 1

- Using iterative erosion

$$S_k(X) = \bigcup_p (X \Theta pB) - ((X \Theta pB) o B)$$

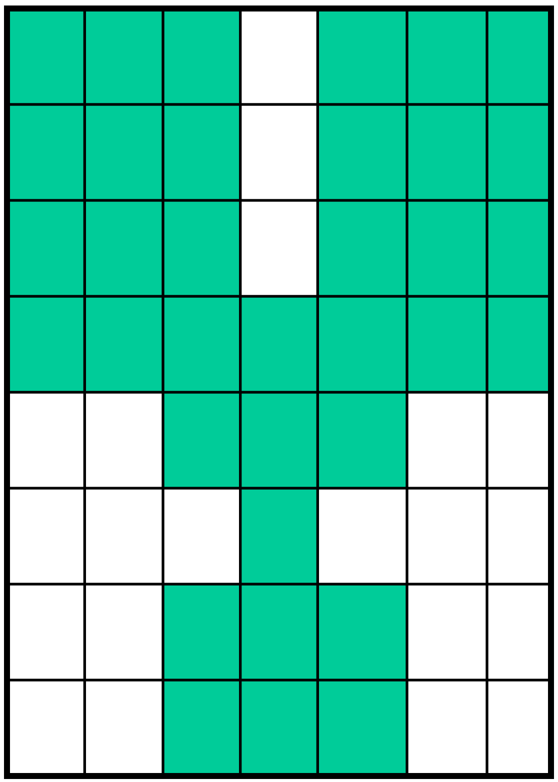
$$S(X) = \bigcup_k S_k(X)$$ Skeleton

S_k : Skeleton subset

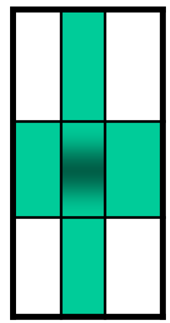
Θ : Erosion operator

Drawbacks : very sensitive to change in shape; difficult to determine the right number of iterations

Skeletonisation - example

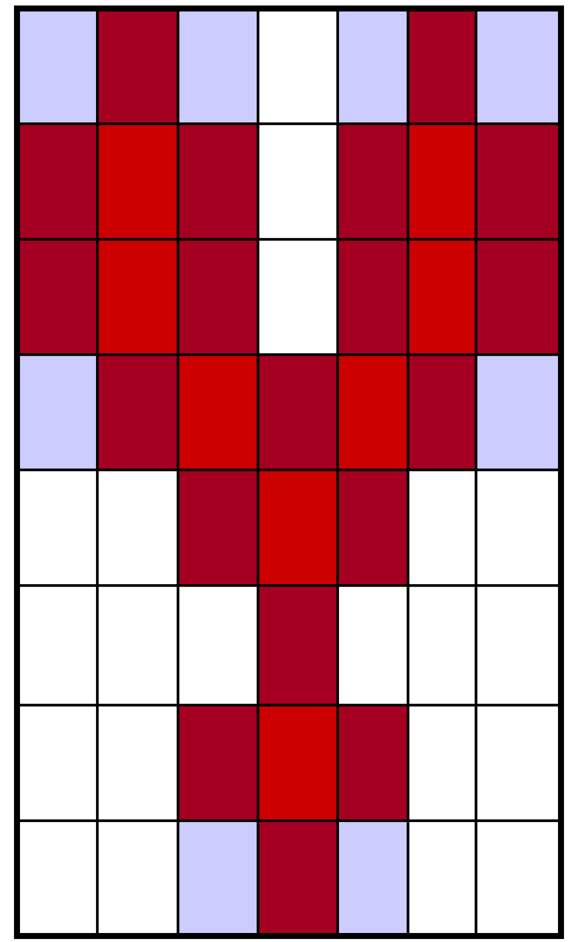


X

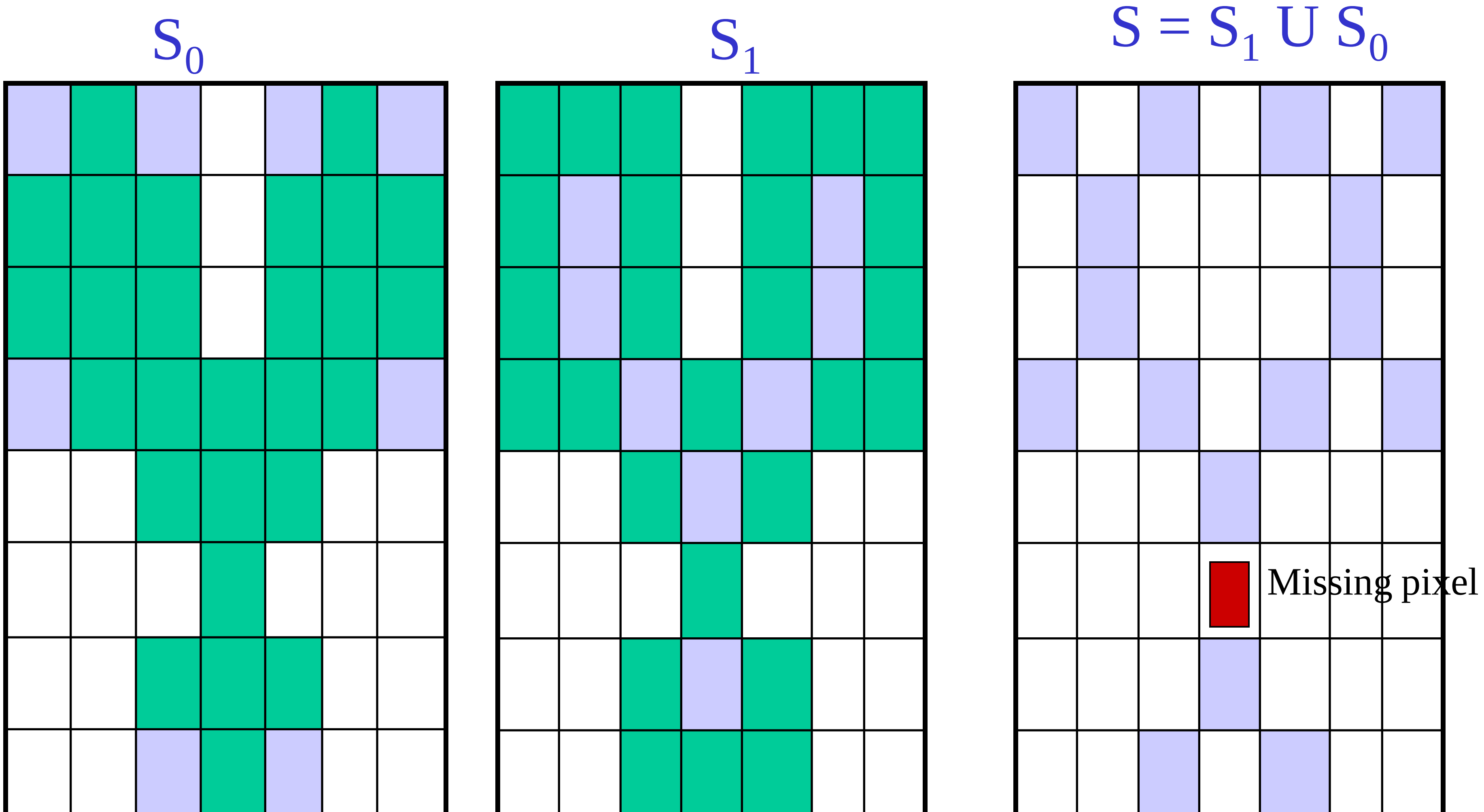


B

Light red: $X \ominus B$
Red: XoB
Blue: S_0



Skeletonisation – example contd.

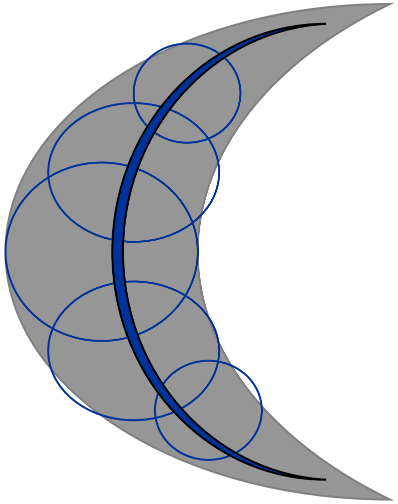


Methods for skeletonisation

Method 2

- Fit a set of maxballs inside the object
  - **Max ball** is a ball of max radius that just fits within X
- Skeleton = {Centre points of the set of max balls}

- **Drawback:** Can result in disjoint skeleton



Distance Transform

- A map/image which gives the *distance from a pixel to the nearest background*
  - different distance measures D can be used
  - also called a Euclidean Distance Map (even when D is not Euclidean dist.!)

Applns.

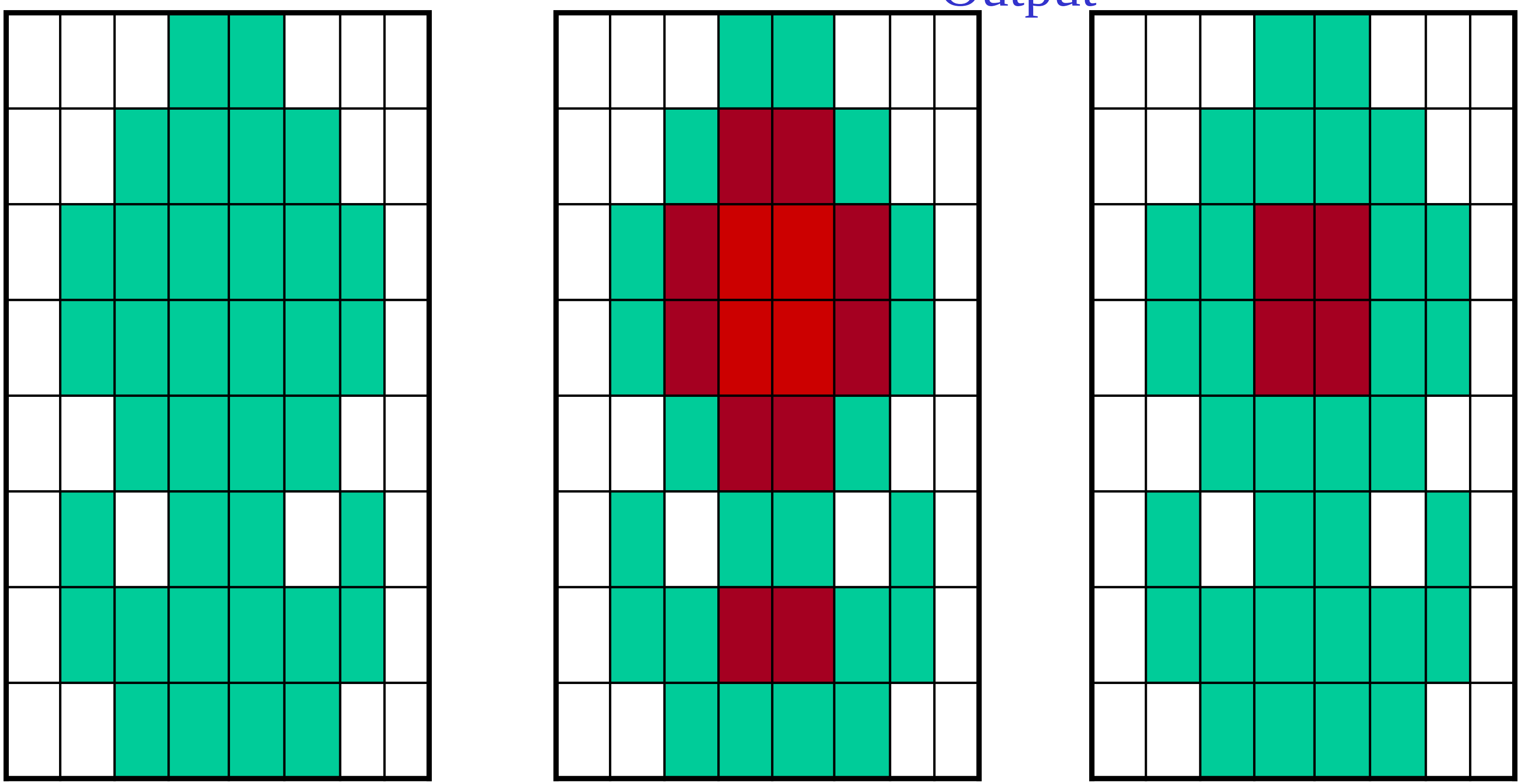
- feature measurement
- binary to grey or colour image conversion
- skeletonisation
- cluster analysis, etc.

EDM examples

Light red: d = 3 pixels; Dark red: d = 2 pixel; Green: d = 1

Input

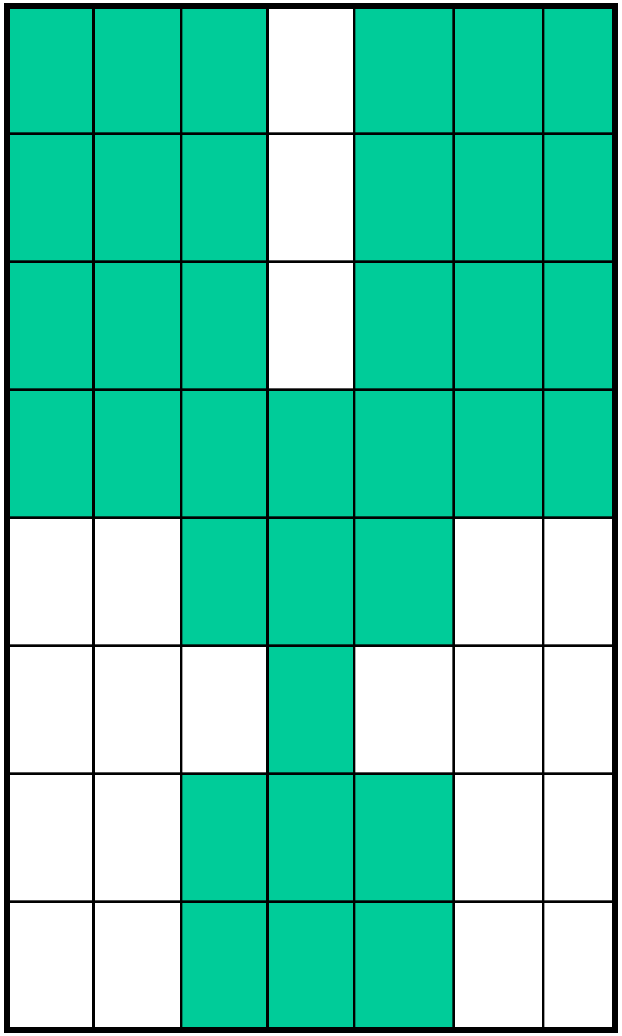
Output

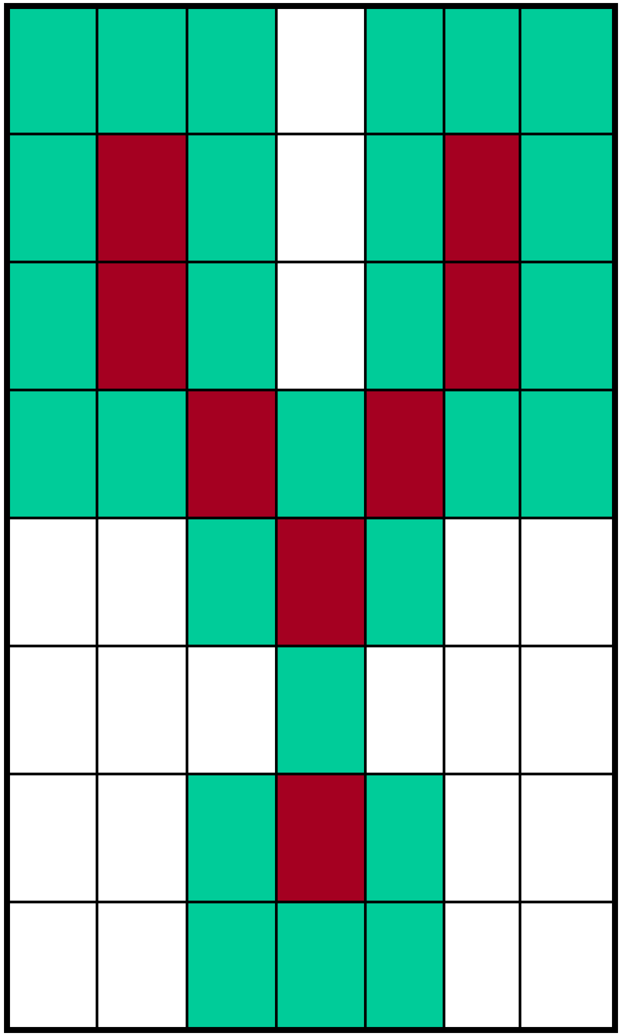


With D_4

With D_8

Skeletonisation using DM

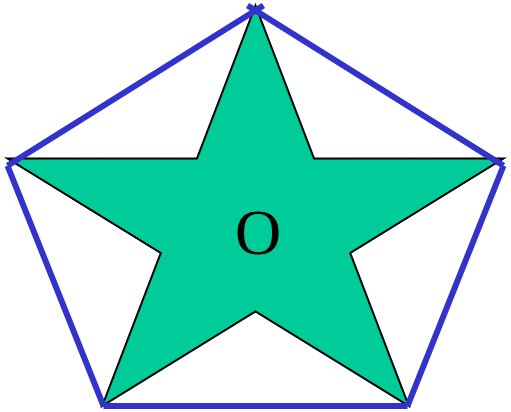




Skeleton using (D_4)

Convex hull

- Region R : $\{x_i\}$
- Region R is convex if straight lines connecting x_i and x_{i+1} is in R.
- Convex hull of an object O is the smallest convex set containing O



Convex Hull of O is a pentagon

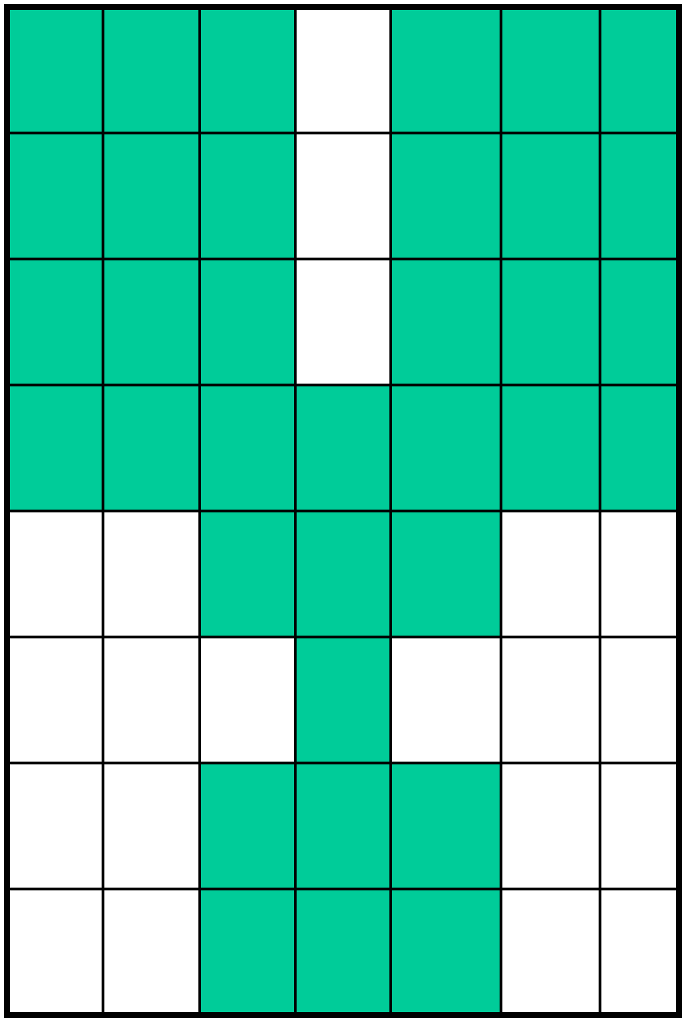
Algo for finding convex hull of O

1. For every pixel *i* find the number n_i of its neighbours which belong to the object

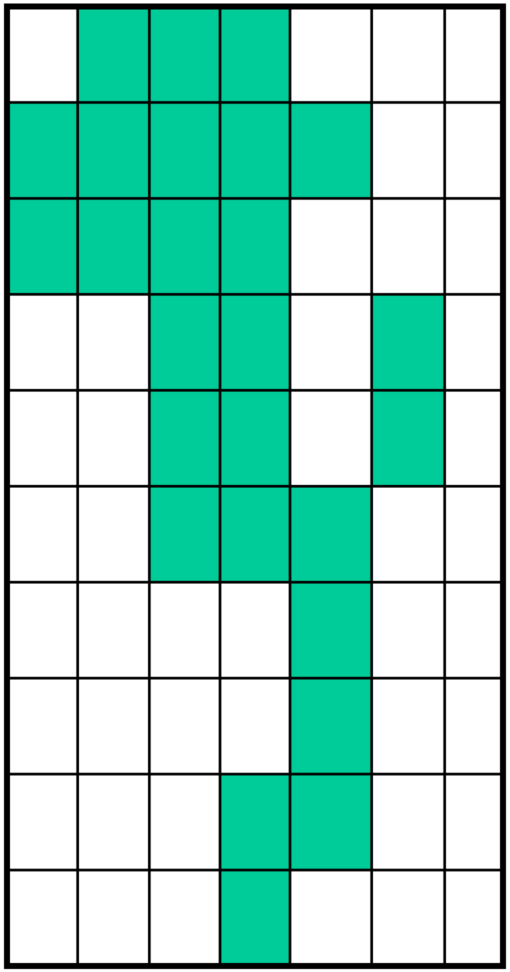
2. If $n_i > 3$ then mark the object pixel *I*

3. Repeat 1 and 2 until there are no pixels with more than 3 neighbours

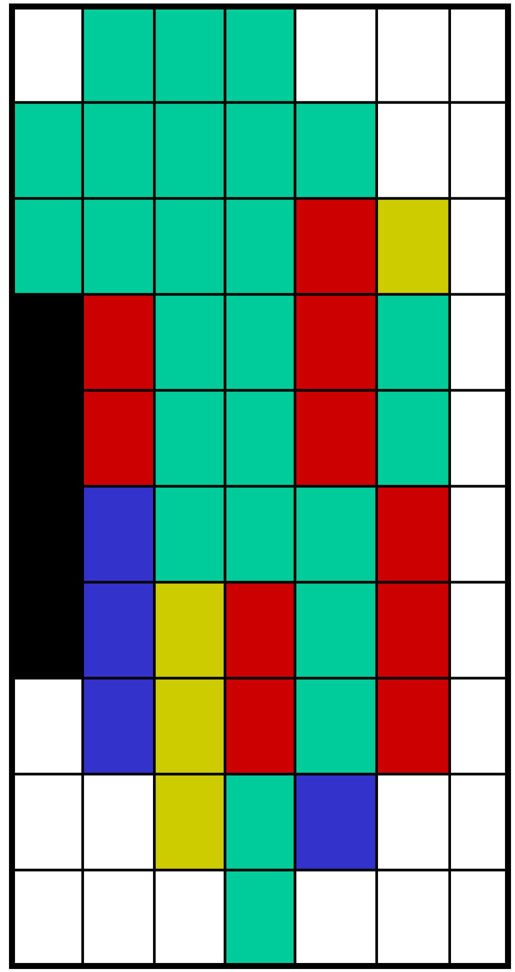
Examples



Example 2



Original image

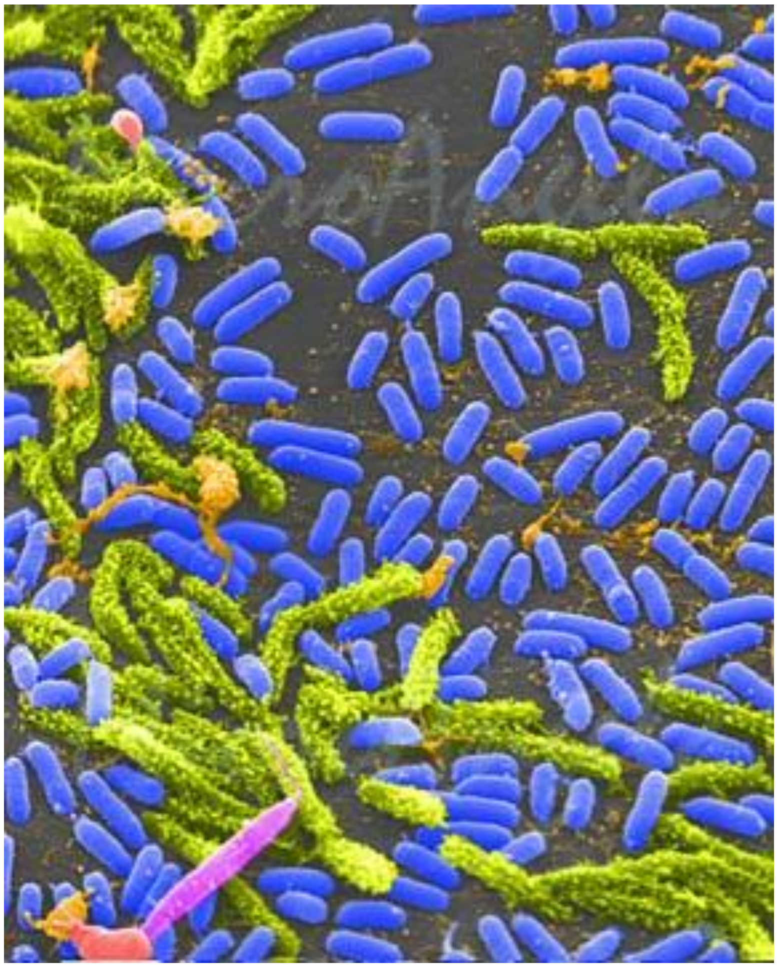


Convex hull

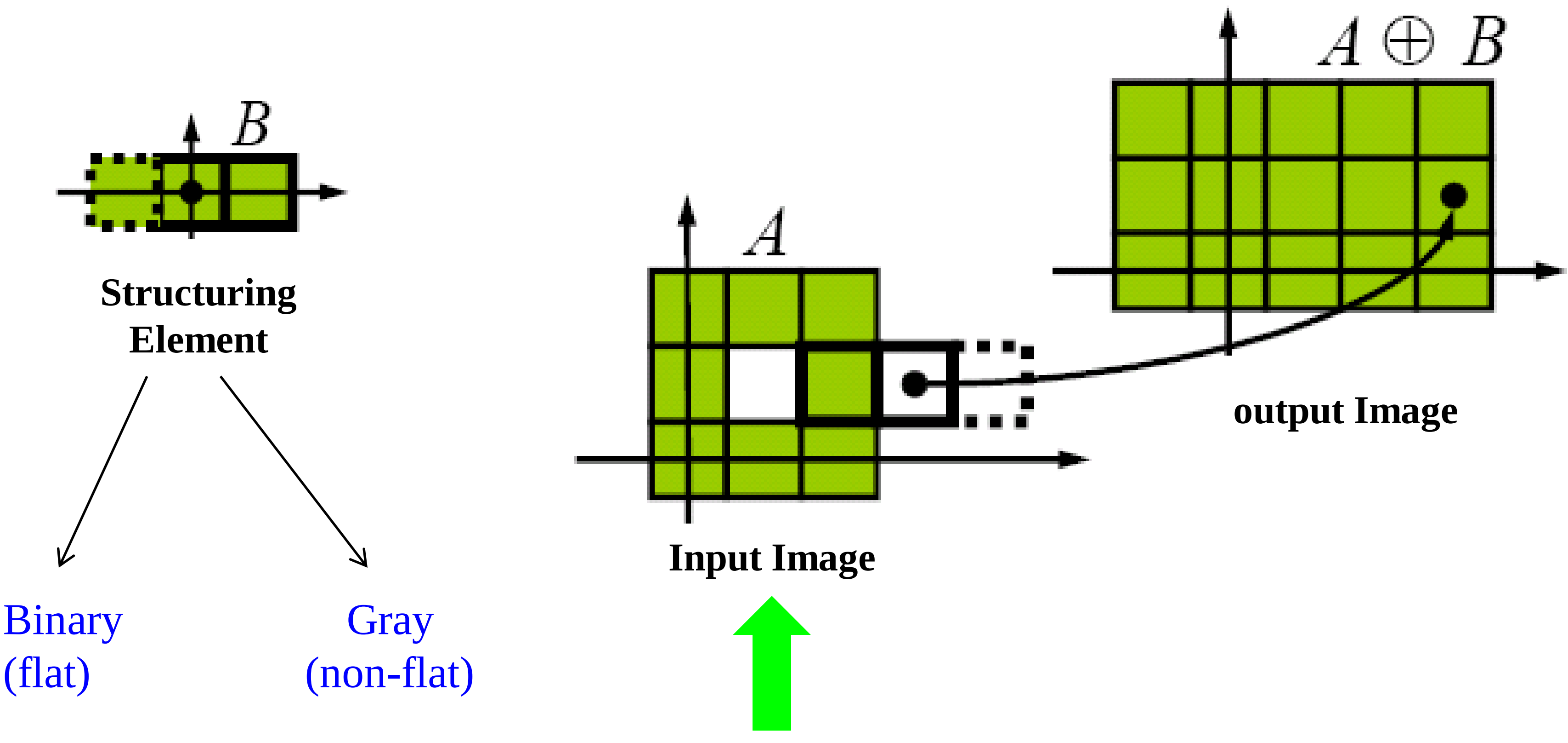
Pass #	colour
1	Red
2	Green
3	Blue
4	Black

Sample task in pathology: Count the number of different bacteria in image

- Microscopy images are acquired at 400x to 1000x



Gray level morphological operations



Gray-level image Only

Binary (flat) Structuring Element

- Greyscale erosion and *dilation* use rank operations (rank filtering)

Definition: replace current pixel in X with minimum or *maximum* value in the window represented by the **se B**

  - shape of B determines the neighbourhood size over which ranking is done

Gray(non-flat) structuring Element

- B is a *grayscale* image defined over a domain D_B

Dilation: find max after adding value of B and X

- ➢ centre pixel = max{X[m-i,n-j] + B[i,j]; i,j $\in D_B$}

$$(f \oplus b)(s,t) = \max\{ f(s-x, y-t) + b(x,y) \mid (s-x),(t-y) \in D_f; (x,y) \in D_b \}$$

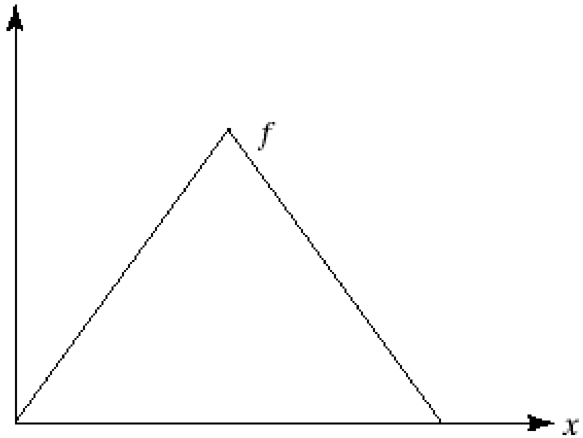
Erosion: find min after subtracting value of B from X

- ➢ centre pixel = min{X[m-i,n-j] - B[i,j]; i,j $\in D_B$}

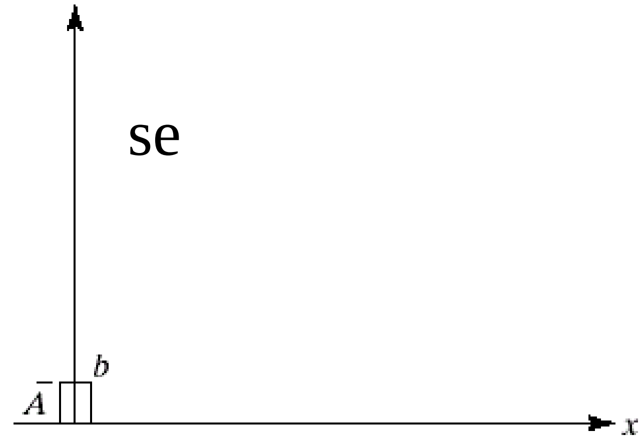
$$(f \oplus b)(s,t) = \min\{ f(s+x, y+t) - b(x,y) \mid (s+x),(t+y) \in D_f; (x,y) \in D_b \}$$

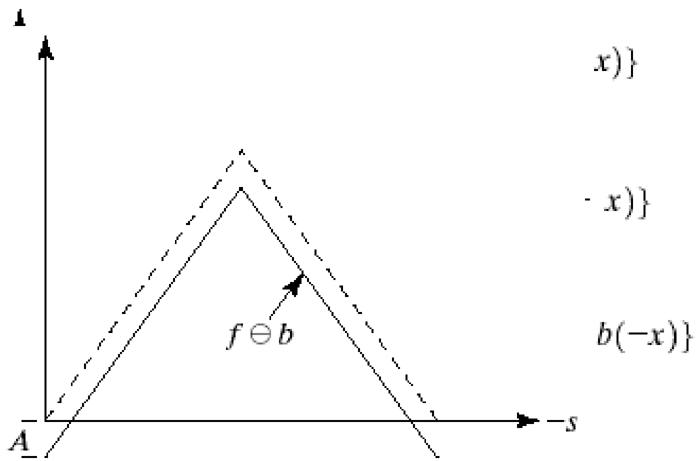
Greyscale dilation and erosion– 1D

Line image

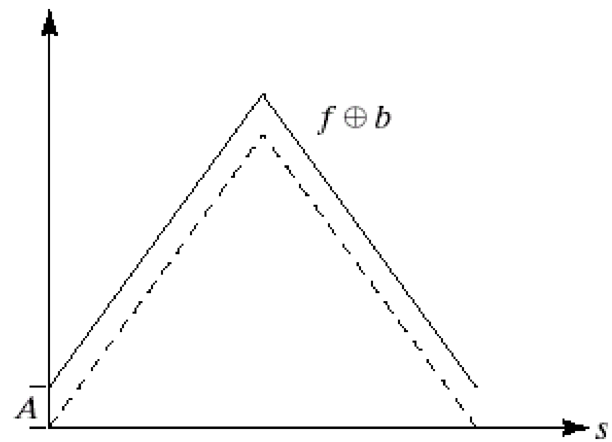


se





Eroded image



Dilated image

Cameraman

Original

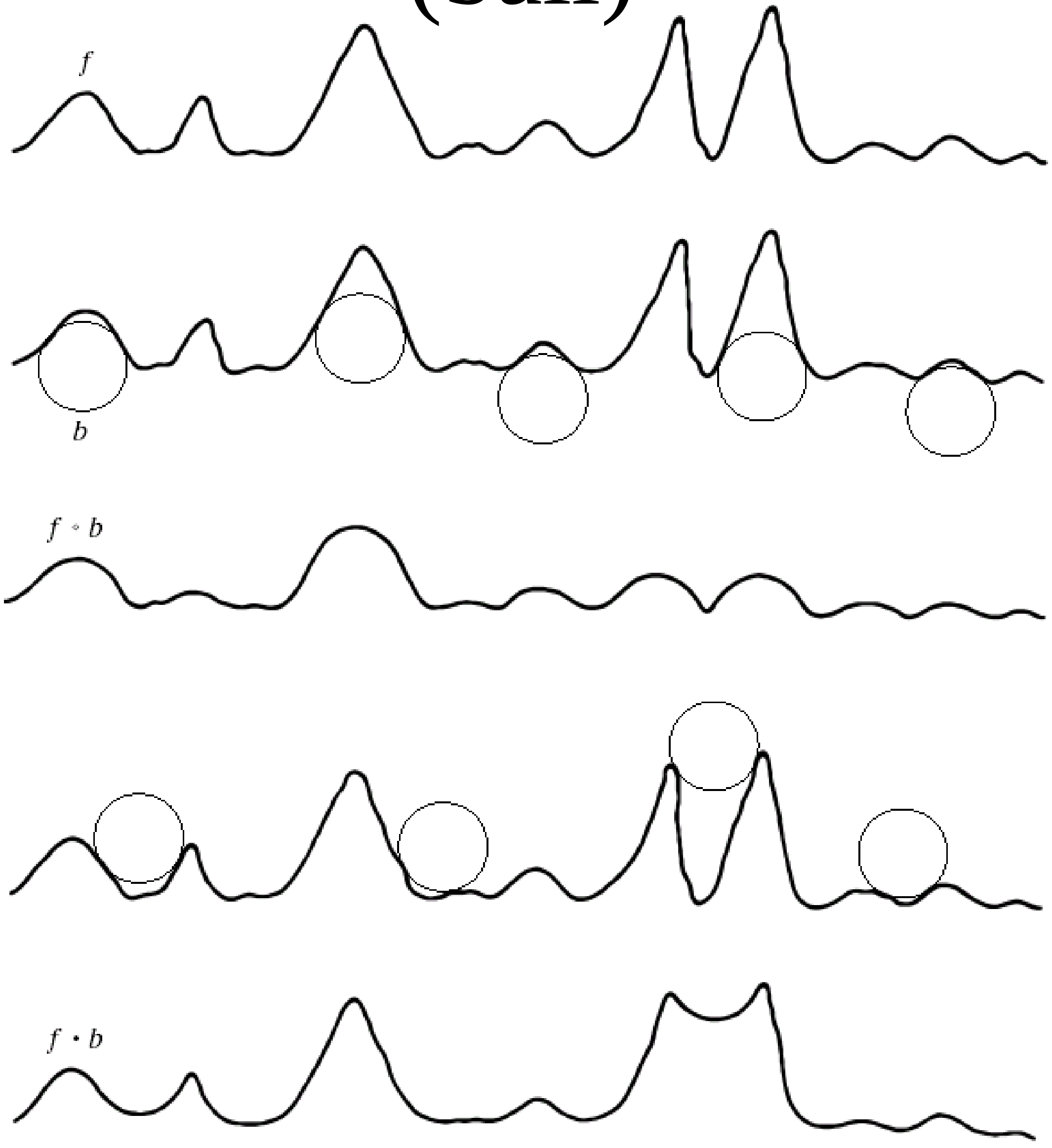
Dilated

(**se** = rolling ball)

Cameraman

(**se** = rolling ball)

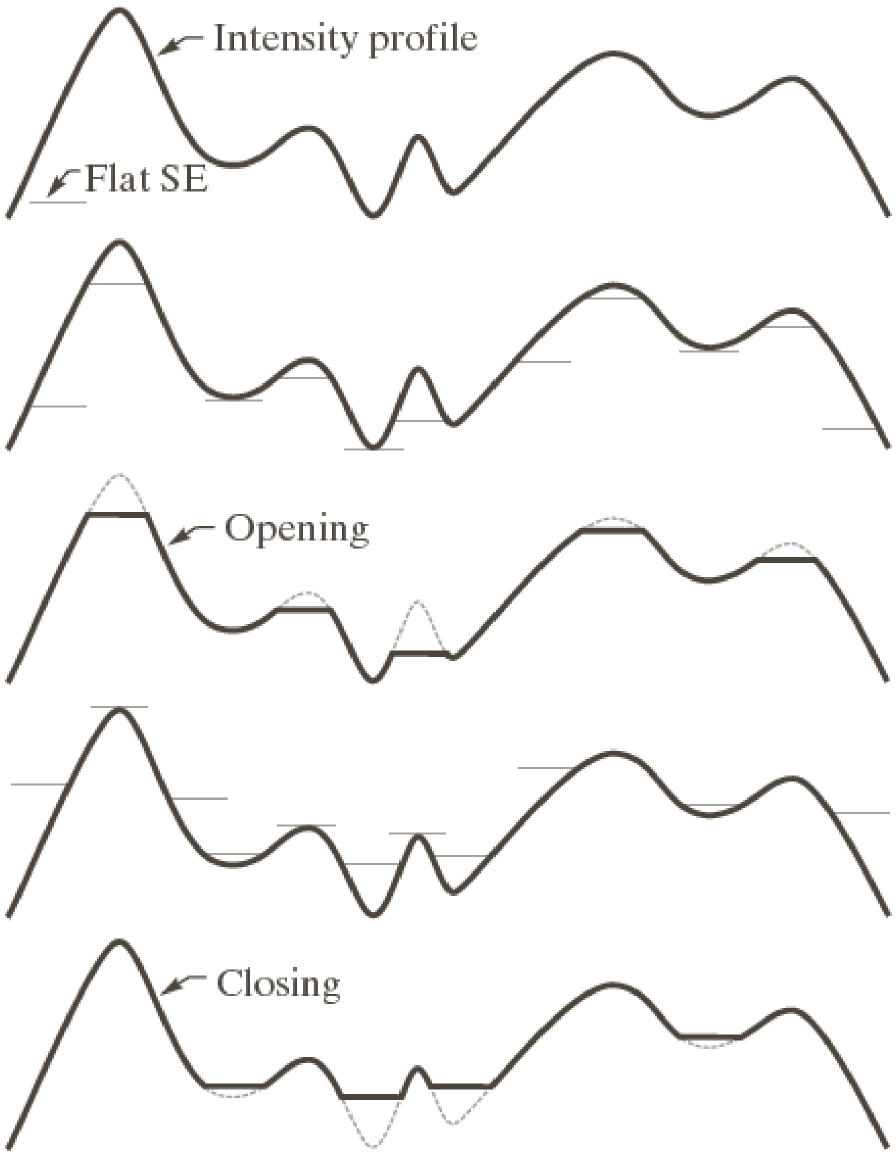
Opening and Closing: non-flat SE (ball)



Opening Result

Closing Result

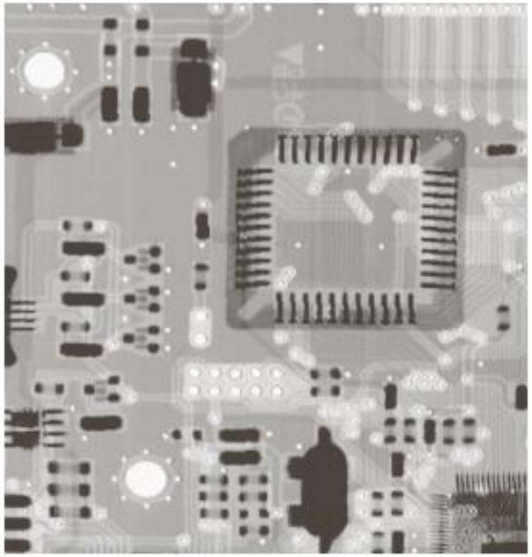
Opening and Closing: flat SE



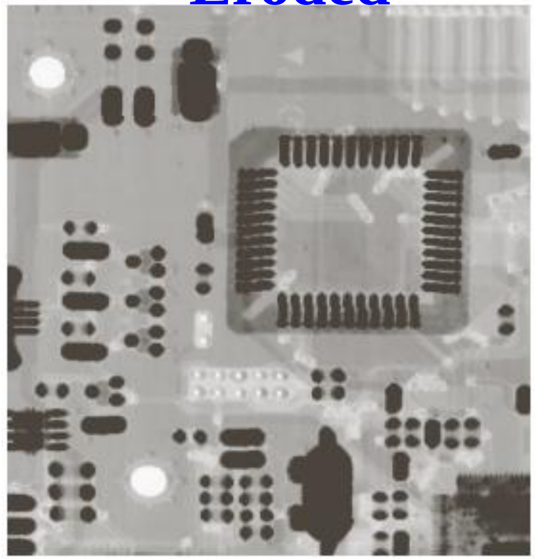
Opening Result

Closing Result

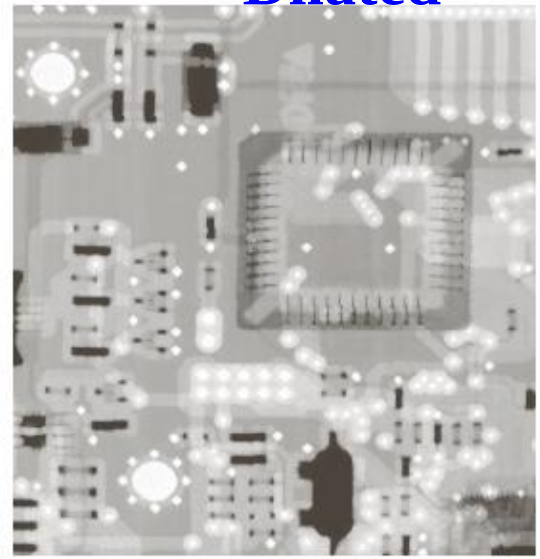
Input - PCB **Output**

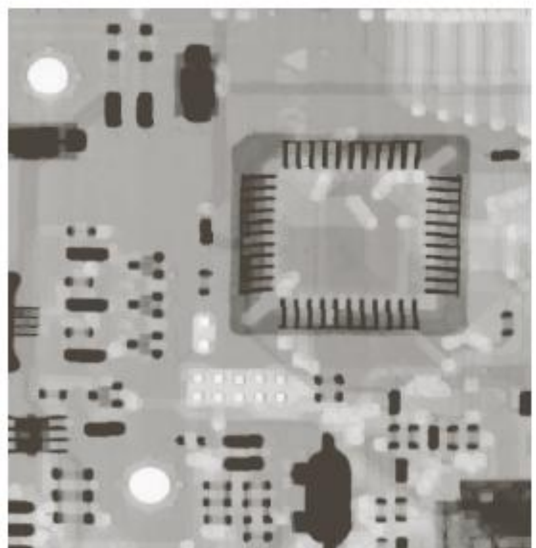


Eroded

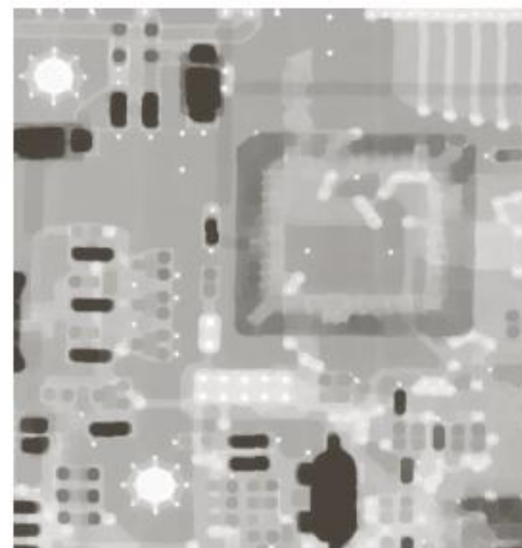


Dilated





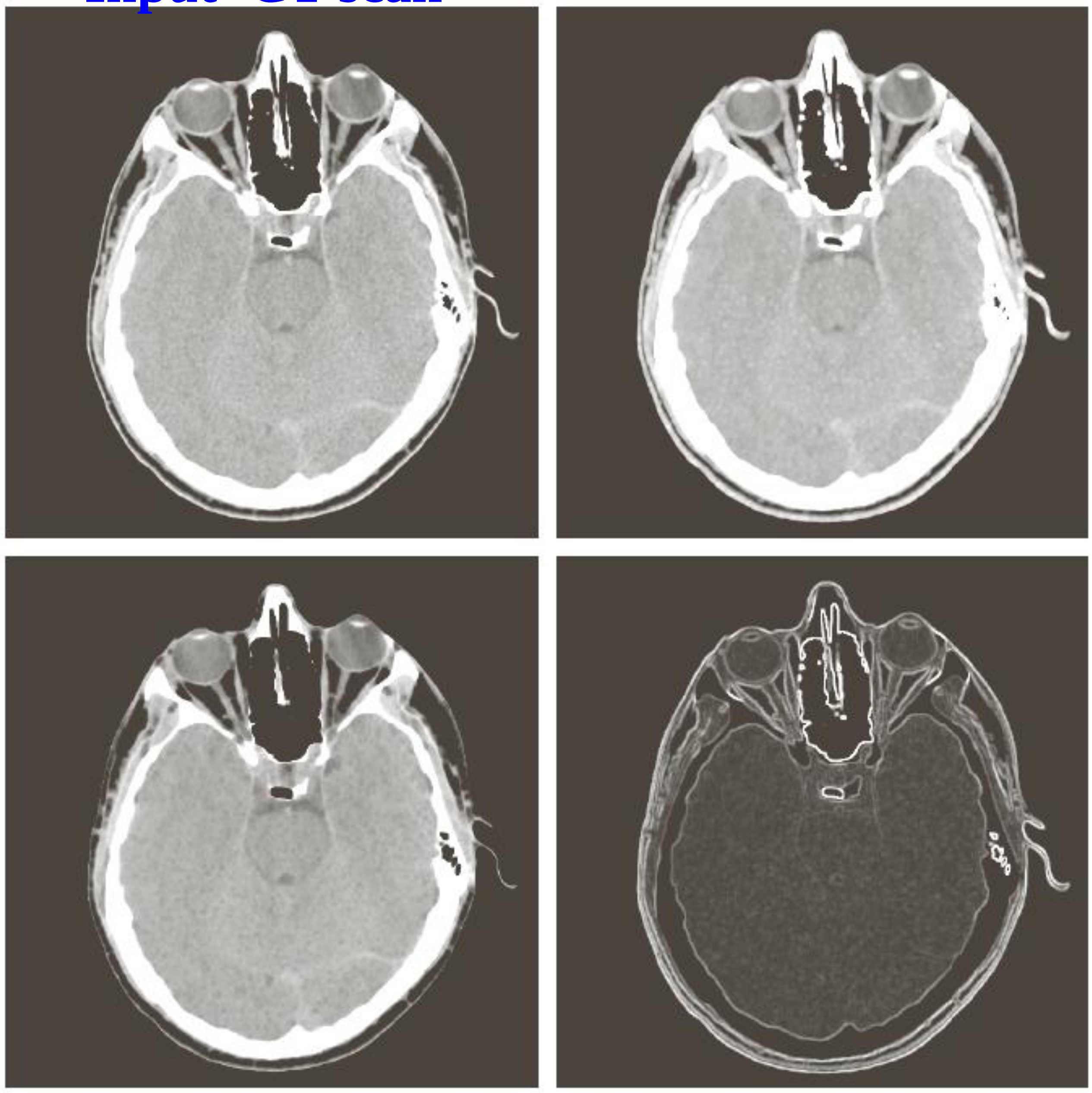
Opened



Closed

Gonzalez and Woods 3rd ed

Input- CT scan **Dilated**



Erode
d

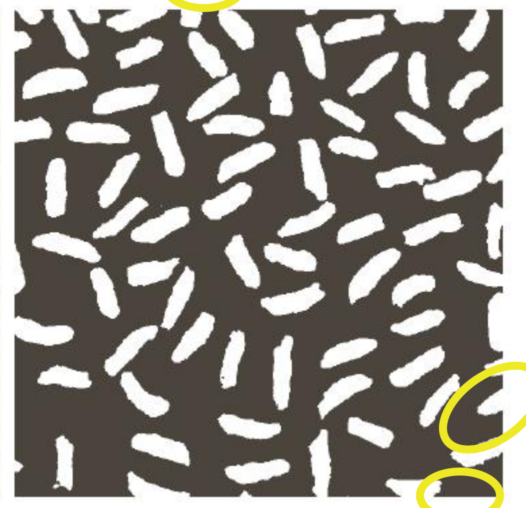
D-E = gradie

Gonzalez and Woods 3rd ed

Morphological top hat transform

Input- rice grains binarised input

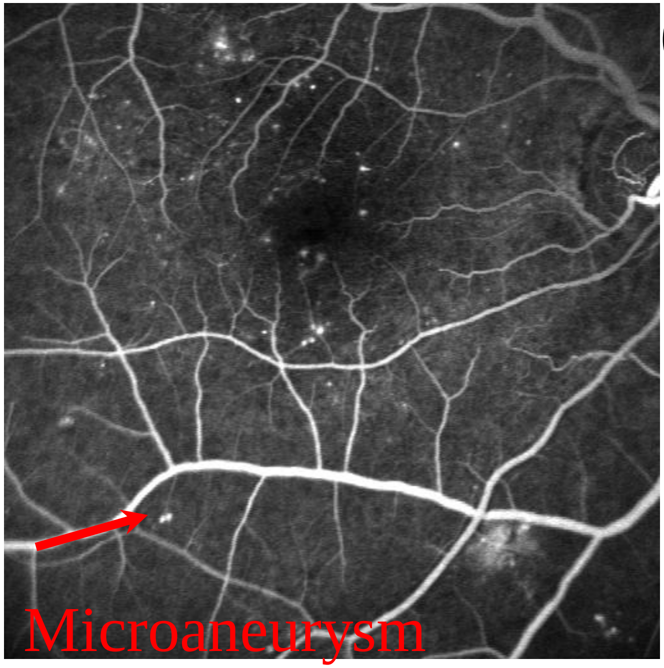
O2: opened result



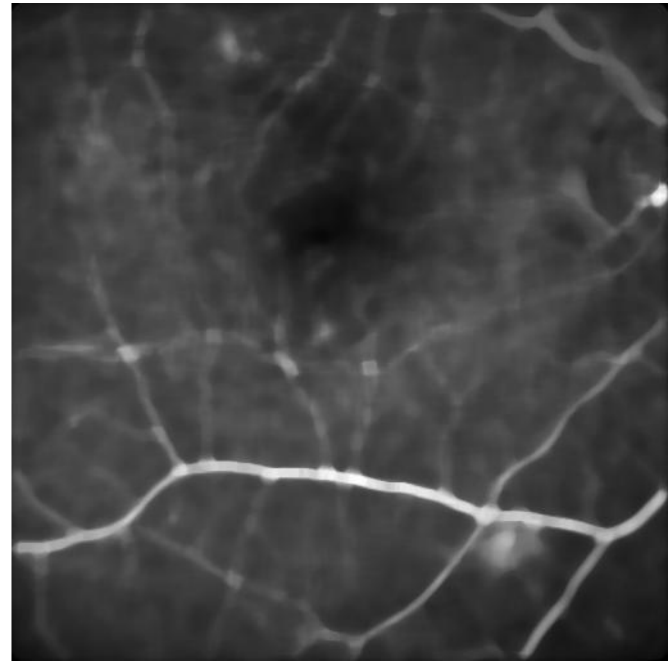
Used for illumination correction **(Input - O2)** → **binarised**

Gonzalez and Woods 3rd ed

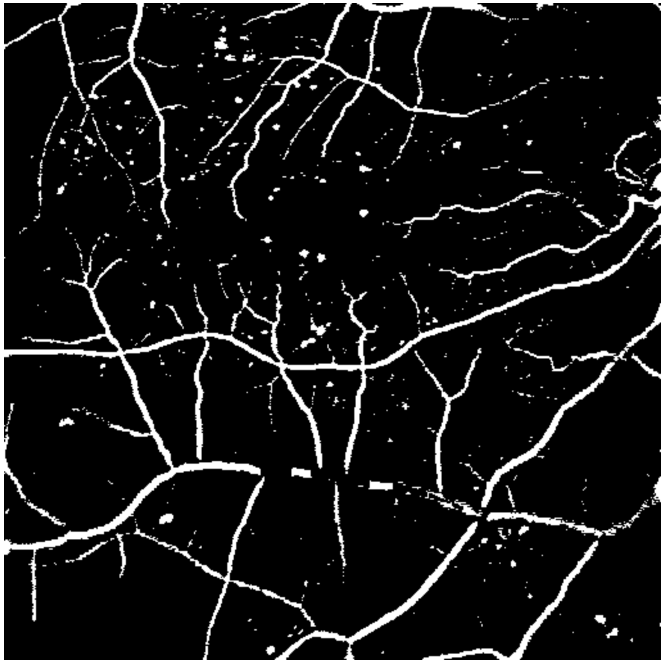
Case Study



Input



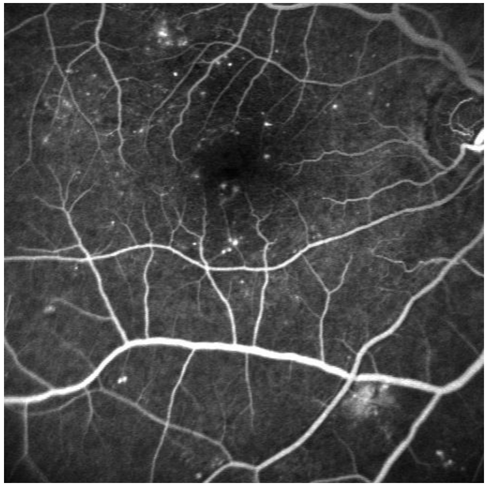
Median Filtering (13 X 13)

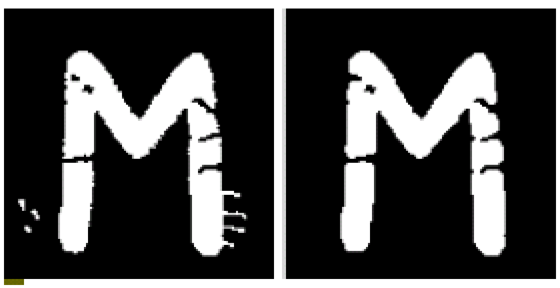


I-M

Similar size vessels are causing problem !!

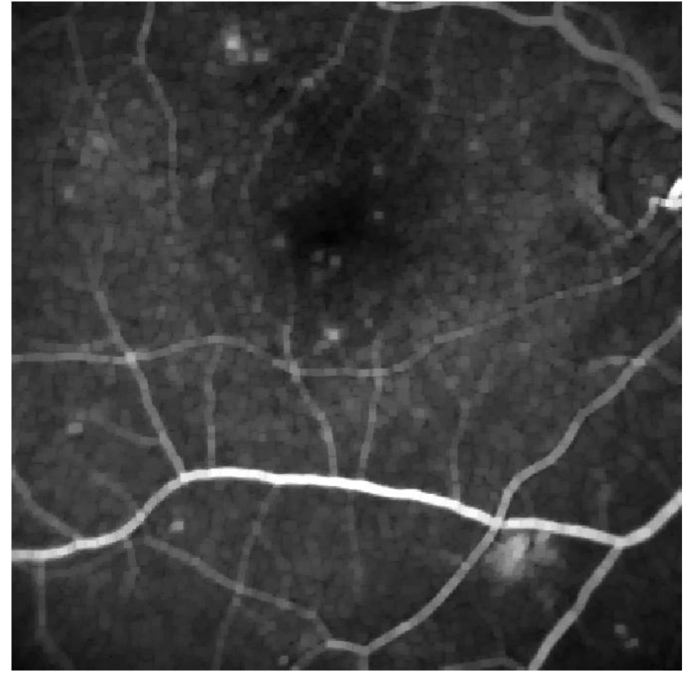
Can view as small sized noise regions



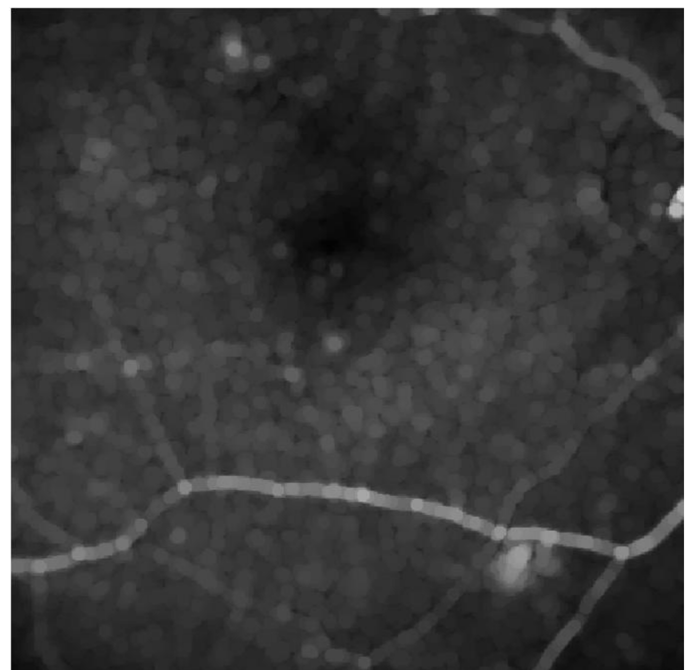


Opening ?

Vessels are also getting removed !!
Can we use some properties of vessels to distinguish them from MA?

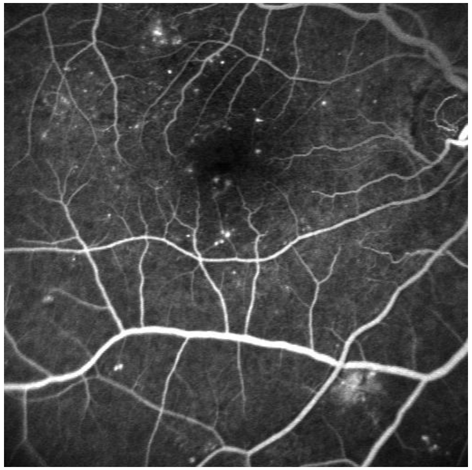


SE: Disk of 3

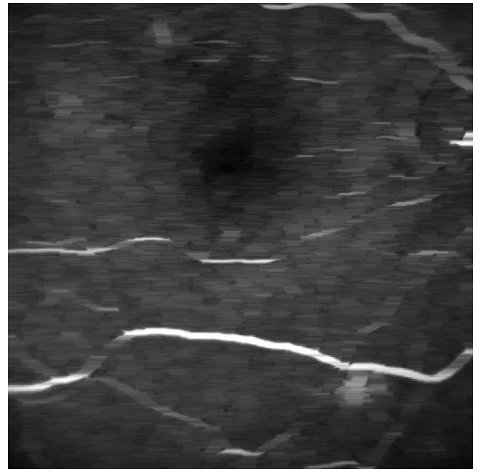


SE: Disk of 5

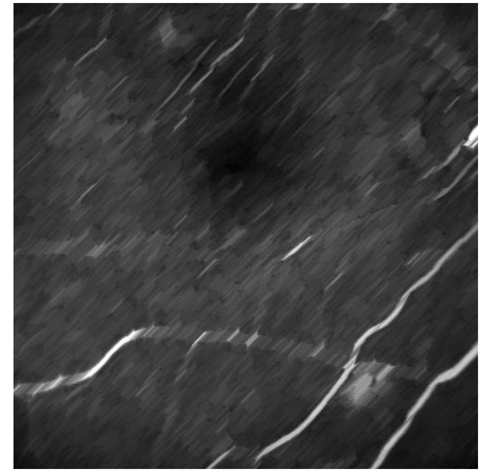
Objective: first remove vessels !!



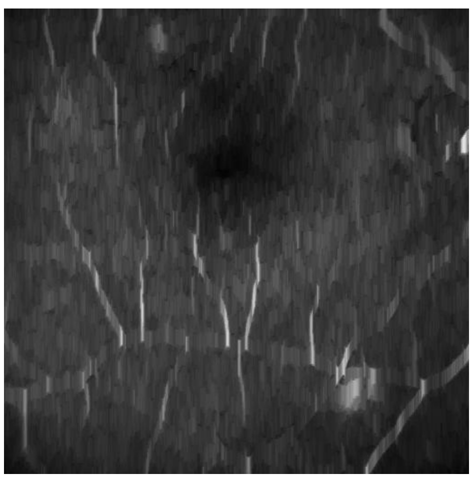
Input



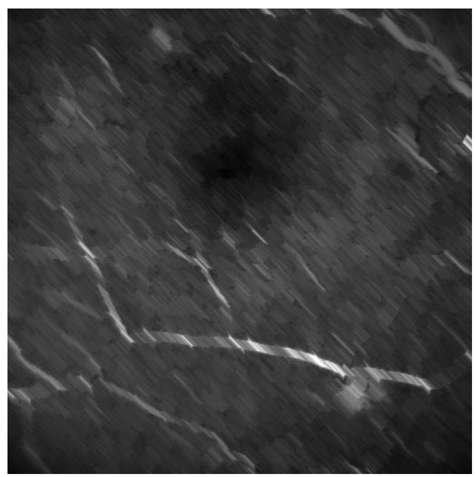
Linear SE= 0 degree, 15



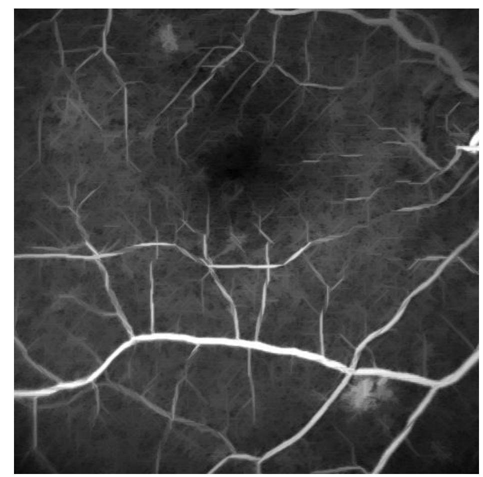
Linear SE= 45 degree, 15



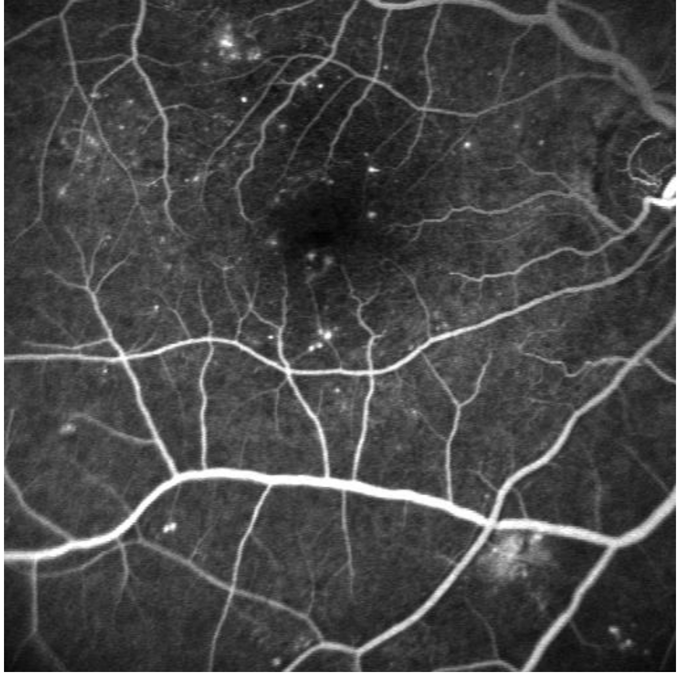
Linear SE= 90 degree, 15



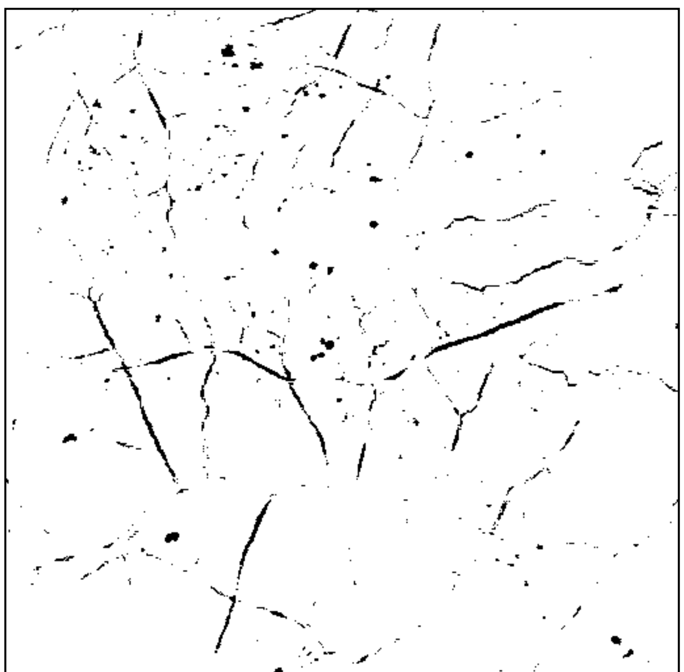
Linear SE= 135 degree, 15



MAX image (0, 45, 90, 135)

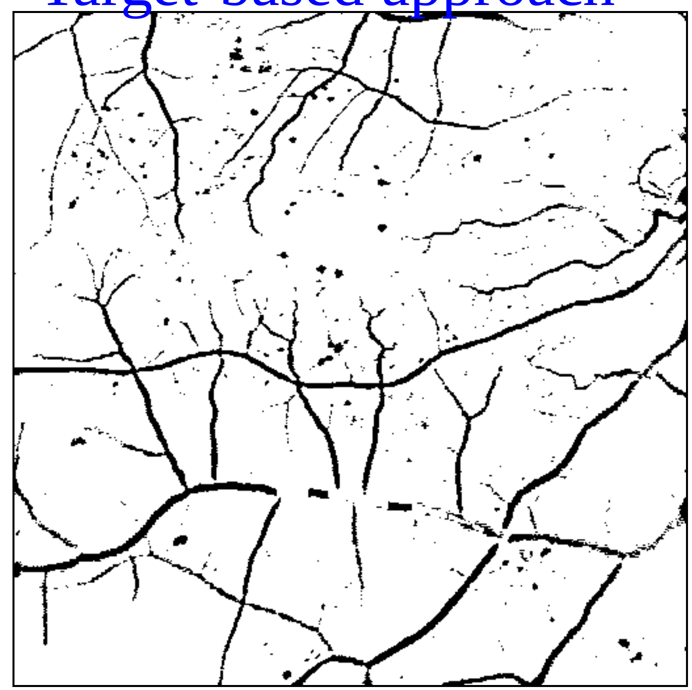


I – Processed Image



After Vessel Removal

Target-based approach



After MA removal (median)