

# Computer Image Processing

## Lecture 9

Morphological operations

# Mathematical morphology

**Mathematical morphology** is a theory and technique for the analysis and processing of geometrical structures.

It was originally developed for binary images, and was later extended to grayscale functions and images.

The research began in the 1960s at the Center de Morphologie Mathématique at the Paris School of Mines (Fontainbleau) - G. Matheron, J. Serra.

# Binary images as sets

Binary images could be treated as sets – object in the picture is a set of all elements (pixels) which value is equal to 1.

Translation by a vector  $t = (t_x, t_y)$

$$A_t = \{a + t : a \in A\}$$

The set  $A$  translated by a vector  $t$  – coordinates of each image pixel  $(x, y)$  are transformed to  $(x + t_x, y + t_y)$ .

# Binary images as sets

Translation by a vector  $t = (t_x, t_y)$

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ \underline{0} & 0 & 0 & 0 & 0 \end{bmatrix}$$

Set A

$$t = (1,1)$$

$$A_t = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \underline{0} & 0 & 0 & 0 & 0 \end{bmatrix}$$

Set A after translation  
by the vector  $t$

Underlined element denotes the center of coordinate system.

# Binary images as sets

Symmetrical mirroring of the set relative to the origin of the coordinate system

$$A^* = \{-a : a \in A\}$$

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & \underline{1} & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Set A

$$A^* = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & \underline{1} & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Set A after mirroring

Underlined element denotes the origin of coordinate system.

# Binary images as sets

## Minkowski addition

$$A \oplus B = \{p : p = a + b, a \in A, b \in B\}$$

$a$  and  $b$  are pairs (vectors) belonging to different sets

Example:

$$A = \{(0,0), (0,1), (-1,0)\}$$

$$B = \{(0,0), (0,1), (1,1)\}$$

$$A \oplus B = \{(0,0), (0,1), (1,1), (0,2), (1,2), (-1,0), (-1,1)\}$$

# Binary images as sets

## Minkowski addition

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & \underline{1} & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 1 \\ \underline{1} & 1 \end{bmatrix}$$

$$A \oplus B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & \underline{1} & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

# Binary images as sets

## Minkowski subtraction

$$A \ominus B = (A' \oplus B^*)'$$

where:

$A' = \{a' : a' \notin A\}$  - complement of set A

$B^* = \{-b : b \in B\}$  - mirroring of set B relative to the origin of the coordinate system

$$A \ominus B = \{a : a + b \in A, \text{ for each } b \in B\}$$



# Binary images as sets

## Minkowski subtraction

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & \underline{1} & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} \underline{1} & 1 \end{bmatrix}$$

$$A \ominus B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & \underline{0} & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

# Morphological operations

Basic operations:

- dilation,
- erosion,
- opening,
- closing.

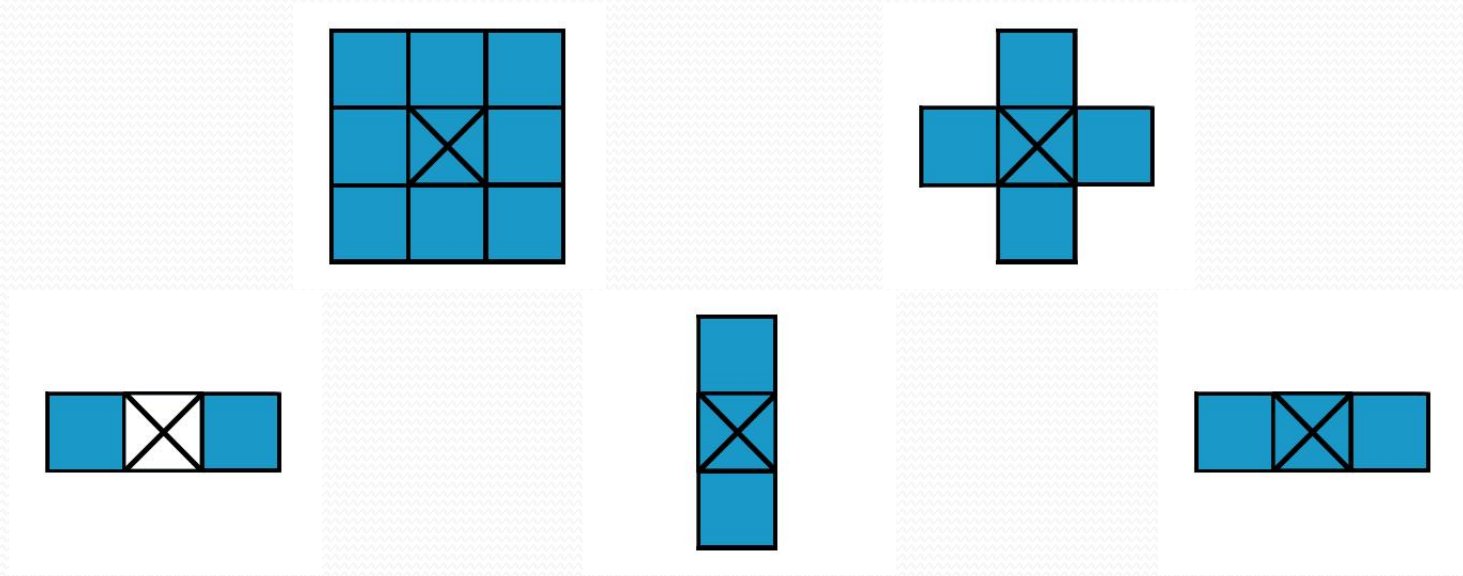
Other operations:

- thickening/thinning,
- morphological skeleton,
- pruning transform,
- hit-or-miss
- ...

# Morphological operations

The basic idea in binary morphology is to probe an image with a simple, pre-defined shape, drawing conclusions on how this shape fits or misses the shapes in the image.

This simple "probe" is called the **structuring element** and is itself a binary image.



# Morphological operations

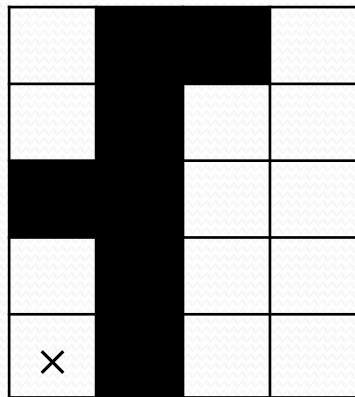
The difference between morphological operations and other operations on the image:

- **contextless operations** – do not take into consideration neighbouring pixels,
- **context operations (filters)** – calculations are performed on each pixel of the image with taking into account neighbouring pixels,
- **morphological operations** – performed only on pixels, whose surroundings are equal to the pattern.

# Morphological operations

**Dilation** - is an operation of extending the base set, which combines two sets with the use of a vector addition. The dilation of sets  $A$  and  $B$  leads to the creation of a set of all  $p$  points representing vector sums of any pairs of elements of sets  $A$  and  $B$ .

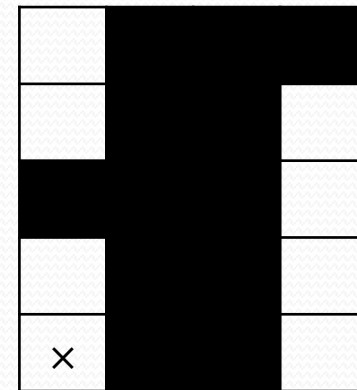
$$A \oplus B = \{p : p = a + b, a \in A, b \in B\}$$



$A$

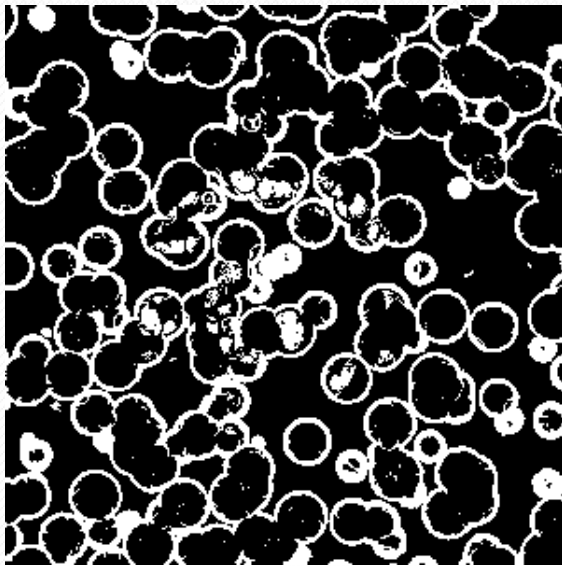


$B$



$A \oplus B$

# Morphological operations



Source image

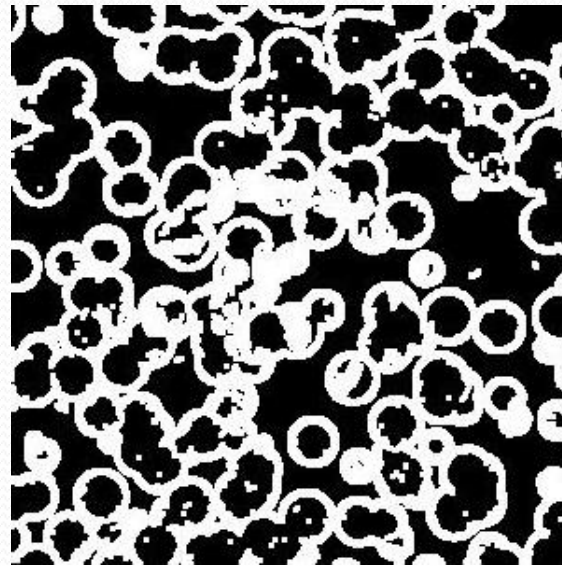
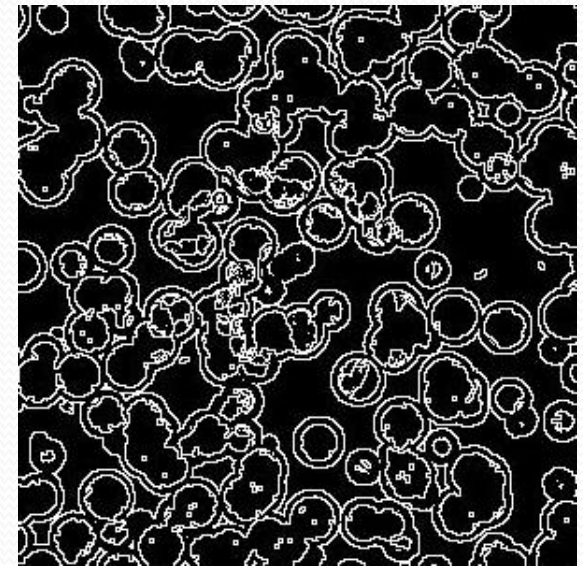


Image after dilation



Added pixels

# Morphological operations

Dilation is used to fill narrow gaps and small holes in the image, connect parts located close to each other, generalize the image and smooth the edges.

Easiest way to describe it is to imagine the same text is written with a thicker pen.

If a structuring element  $3 \times 3$  is used for dilation, the object is enlarged by a 4-contour.

# Morphological operations



Source image



Image after five-times dilation with the use of structuring element 3x3.



# Morphological operations



Source image



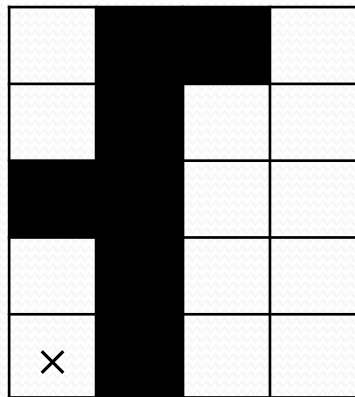
Image after five-times dilation with the use of structuring element  $3 \times 3$ .

# Morphological operations

**Erosion** - means a set of all points  $a$  such that  $a$  shifted by  $b$  is contained in  $A$

$$A \ominus B = \{a : a + b \in A, \text{ for each } b \in B\}$$

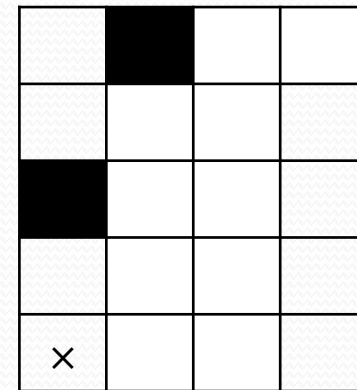
The result is determined only by the points  $a$ , for which  $a + b \in A$



$A$

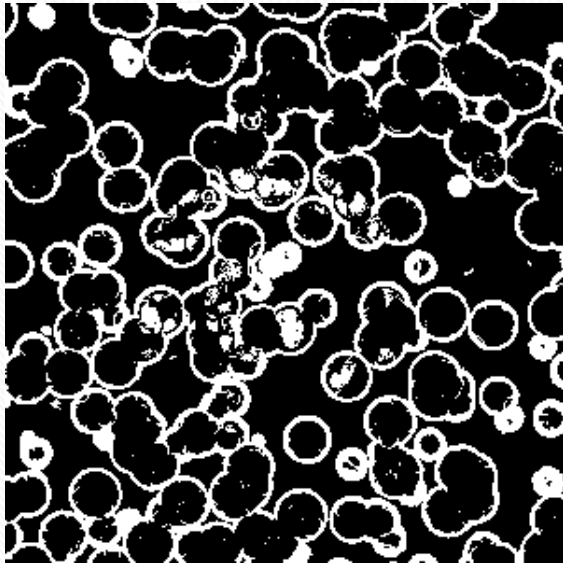


$B$



$A \ominus B$

# Morphological operations



Source image

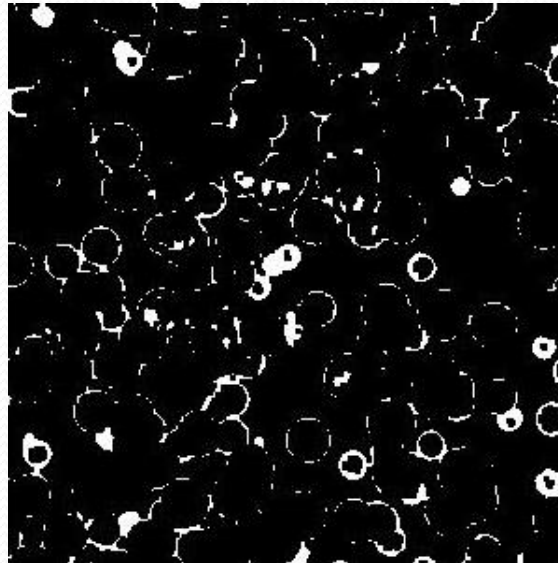
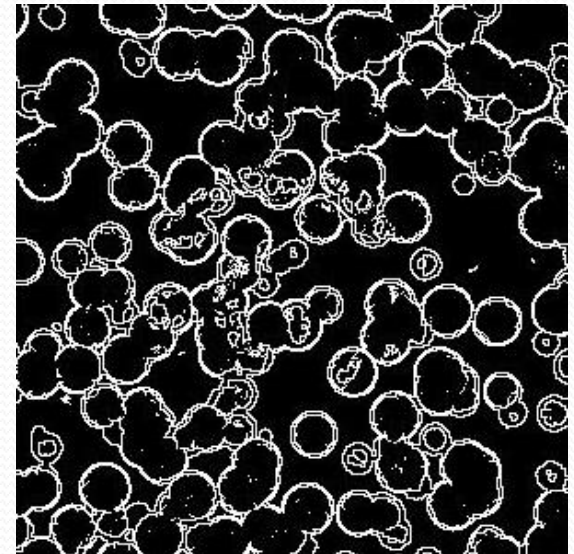


Image after erosion



Removed pixels

# Morphological operations

Erosion is used to simplify the structure of the object, remove small details, smooth the edge, divide the object compound into several simpler components.

Erosion is an additive transformation, i.e. the effect of complex erosion is the same as the effect of successive straight transformations.

The use of a longitudinal shaped structuring element allows to distinguish longitudinal shapes in the image in a direction similar to the direction of the structuring element.

# Morphological operations



Source image



Image after five-times erosion with the use of structuring element 3x3.



# Morphological operations



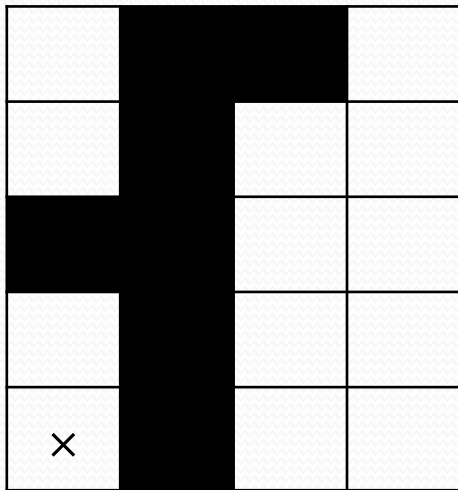
Source image



Image after five-times erosion with the use of structuring element 3x3.

# Morphological operations

Comparison of the effect of morphological operations



Source image

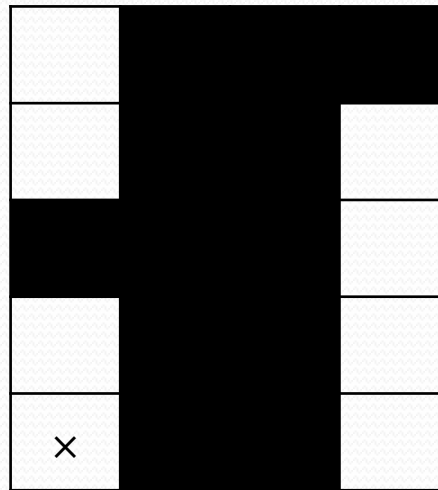


Image after dilation

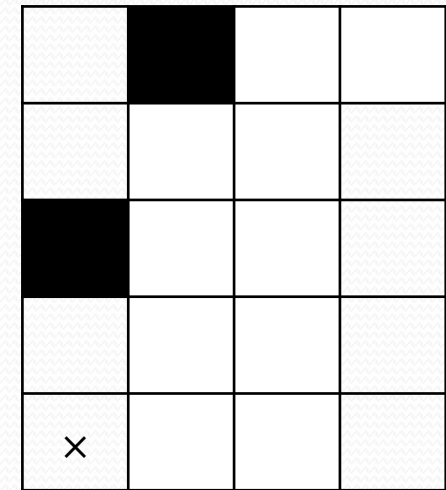
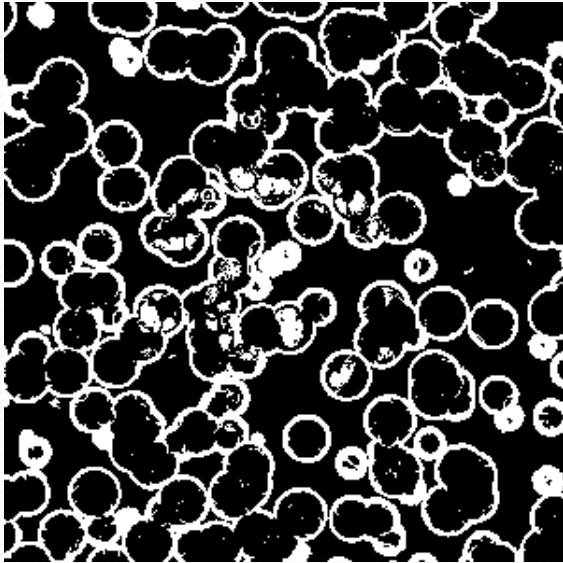


Image after erosion

# Morphological operations

Comparison of the effect of morphological operations



Source image

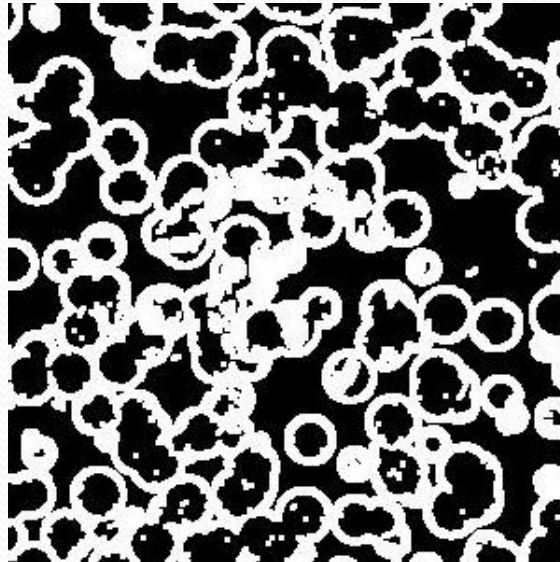


Image after dilation

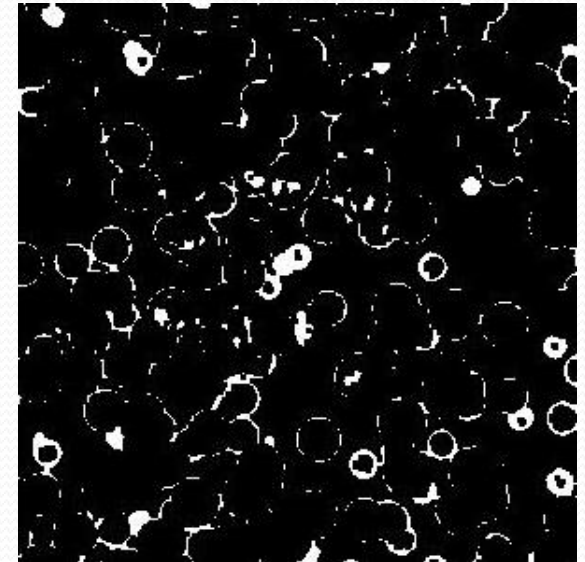


Image after erosion



# Morphological operations

## Opening and closing

Erosion and dilation are not fully reversible operations. Used sequentially, they do not give as a result the source image, as it is usually significantly simplified.

**Opening** = erosion + dilation  $A \circ B = (A \ominus B) \oplus B$

**Closing** = dilation + erosion  $A \cdot B = (A \oplus B) \ominus B$

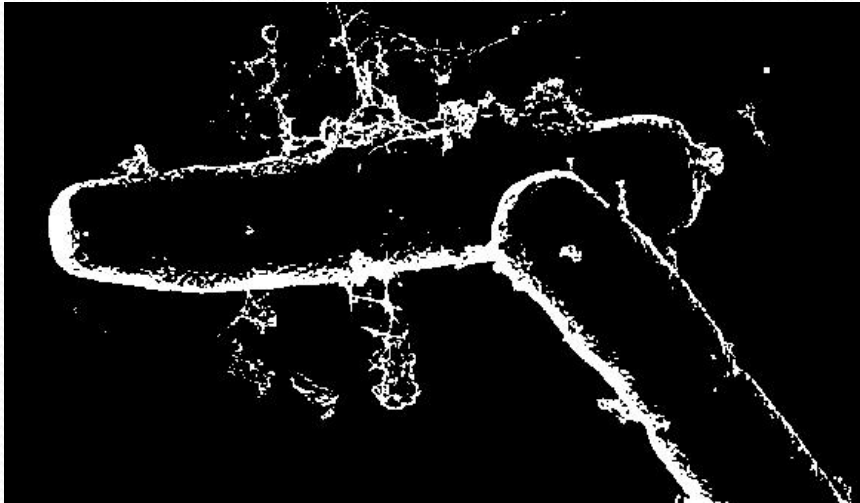
# Morphological operations

The point of **opening** is to move the circle (structuring element) on the inside of the edge of the figure and rejecting all those points that can not be reached by the wheel.

That transformation is used to remove small objects, depressions and bulges, to disconnect objects with constrictions.

# Morphological operations

## Opening



Source image



Image after opening

# Morphological operations



Source image



Image after opening

# Morphological operations



Source image



Image after opening

# Morphological operations

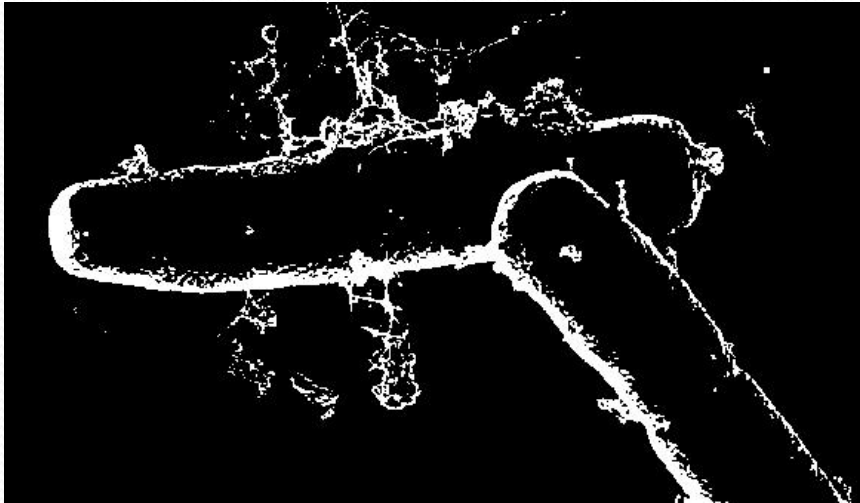
The point of **closing** is to move the circle (structuring element) on the outside of the edge of the figure and adding to it all those points that can not be reached by the wheel.

An operation that can be used to fill gaps and small holes inside the object.

Opening and closing do not change the shape or area of objects with large dimensions and smooth contour.

# Morphological operations

## Closing



Source image

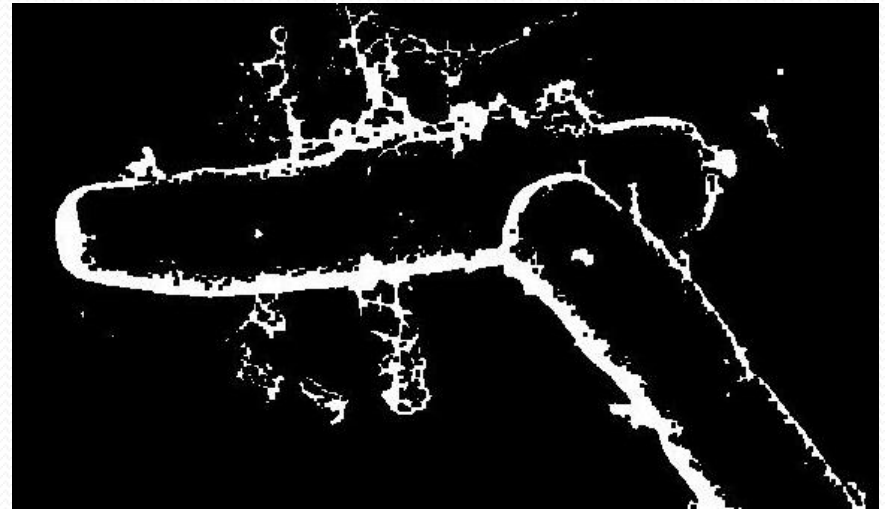


Image after closing



# Morphological operations



Source image



Image after closing



# Morphological operations



Source image



Image after closing

# Morphological operations

Comparison between opening and closing

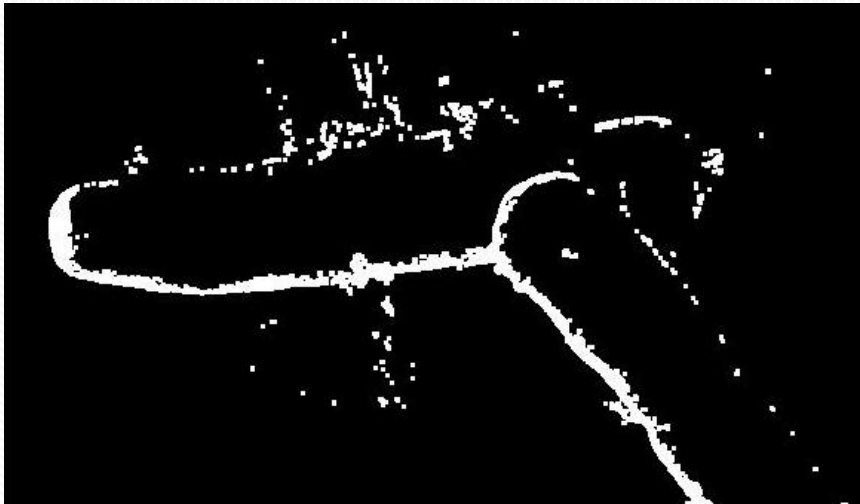


Image after opening



Image after closing

# Morphological operations

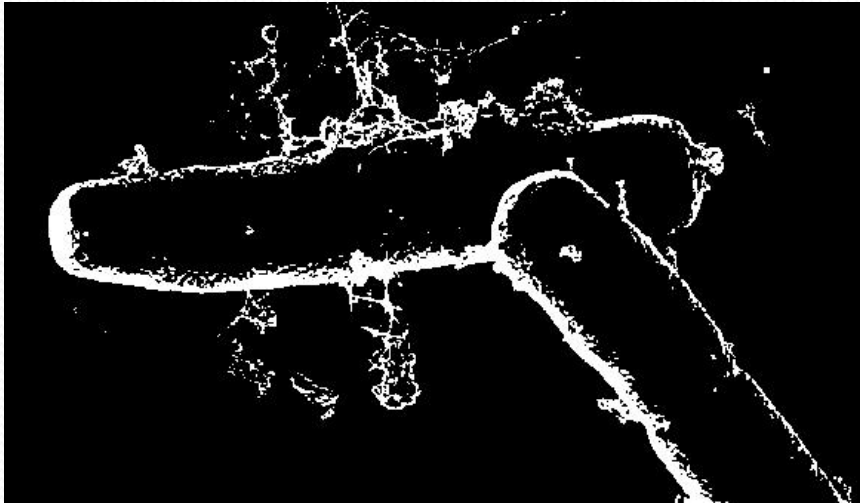


Image after opening



Image after closing

# Morphological operations



Source image

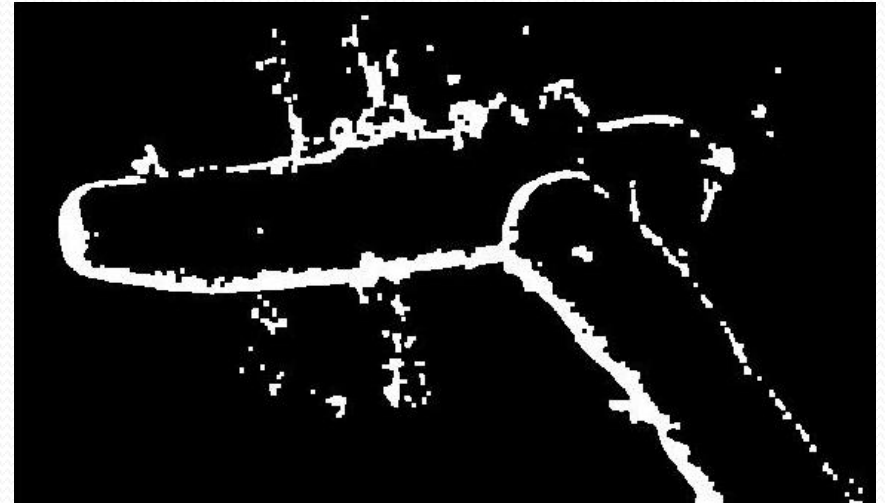


Image after opening and closing

# Morphological operations

**Hit-or-miss** transform is an operation that detects a given configuration (or pattern) in a binary image, using the morphological erosion operator and a pair of disjoint structuring elements. The result of the hit-or-miss transform is the set of positions where the first structuring element fits in the foreground of the input image, and the second structuring element misses it completely.

# Morphological operations

## Hit-or-miss

Element  $B = (B_1, B_2)$  on the assumption  $B_1 \cap B_2 = \emptyset$  can be presented in the form of one pattern in the way that by the value of 1 the pixels of the element are supposed to hit the pixels of the object with the value 1 and by 0 those that have to hit the surroundings, i.e. the image pixels with the value 0.

$$HTM(A, B) = \{p : B_1 \subset A \text{ i } B_2 \subset A'\}$$

# Morphological operations

## Hit-or-miss

In Matlab following elements of the pattern denotes:

1 – points to hit the analyzed object

-1 – points to miss it

0 – points without matter

-1	-1	-1
-1	1	-1
0	0	0

# Morphological operations

## Hit-or-miss

Examples of structuring elements that can be used to detect characteristic elements of an image:

-1	-1	-1
-1	1	-1
-1	-1	-1

isolated points

-1	-1	-1
-1	1	-1
0	0	0

endpoints

1	0	0
0	1	0
1	0	1

nodes