Morphological Image Processing

Contents

Once segmentation is complete, morphological operations can be used to remove imperfections in the segmented image and provide information on the form and structure of the image.

In this lecture we will consider

- What is morphology?
- Simple morphological operations
- Compound operations
- Morphological algorithms

What Is Morphology?

The term *Morphology* means the form and structure of an object or the arrangements and interrelationships between the parts of an object.

Morphological image processing (or *morphology*) refers to a range of image processing techniques that deal with the shape (or morphology) of objects in an image.

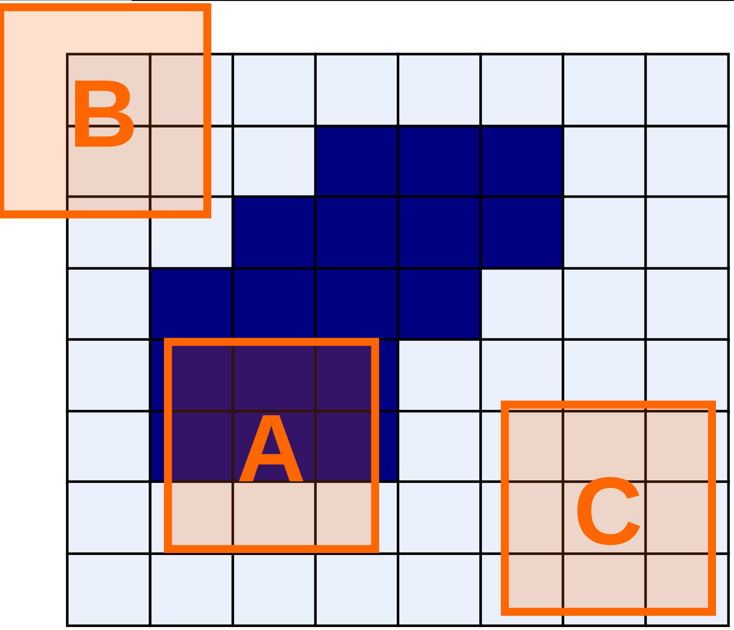
Morphological operations are typically applied to remove imperfections introduced during segmentation, and so typically operate on binary images.

Quick Example

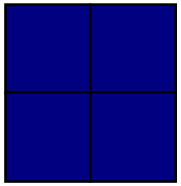
Image after segmentation

Image after segmentation and morphological processing

Structuring Elements, Hits & Fits



Structuring Element: a shape of any size and type.



Fit: All *on pixels* in the structuring element cover *on pixels* in the image.

Hit: Any *on pixel* in the structuring element covers an *on pixel* in the image.

All morphological processing operations are based on these simple ideas.

Structuring Elements

Structuring elements can be any size and make any shape.

However, for simplicity we will use rectangular structuring elements with their origin at the middle pixel.

1	1	1
1	**1**	1
1	1	1

0	1	0
1	**1**	1
0	1	0

0	0	1	0	0
0	1	1	1	0
1	1	**1**	1	1
0	1	1	1	0
0	0	1	0	0

Fitting & Hitting

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0	0	0	0	0
0	0	1	B 1	1	1	1	0	C 0	0	0	0
0	1	1	1	1	1	1	1	0	0	0	0
0	1	1	1	1	1	1	1	0	0	0	0
0	0	1	1	1	1	1	1	0	0	0	0
0	0	1	1	1	1	1	1	1	0	0	0
0	0	1	1	1	1	1	A 1	1	1	1	0
0	0	0	0	0	1	1	1	1	1	1	0
0	0	0	0	0	0	0	0	0	0	0	0

1	1	1
1	1	1
1	1	1

Structuring Element 1

0	1	0
1	1	1
0	1	0

Structuring Element 2

Fundamental Operations

Morphological operations process images based on shapes. Morphological operations apply a structuring element to an input image, creating an output image of the same size. The structuring element is moved across every pixel in the original image to give a new pixel in the output image.

In a morphological operation, the value of each pixel in the output image is based on a comparison of the corresponding pixel in the input image with its neighbors. The value of this new pixel depends on the operation performed.

There are two basic morphological operations:

- **Dilation**
- **Erosion.**

Dilation

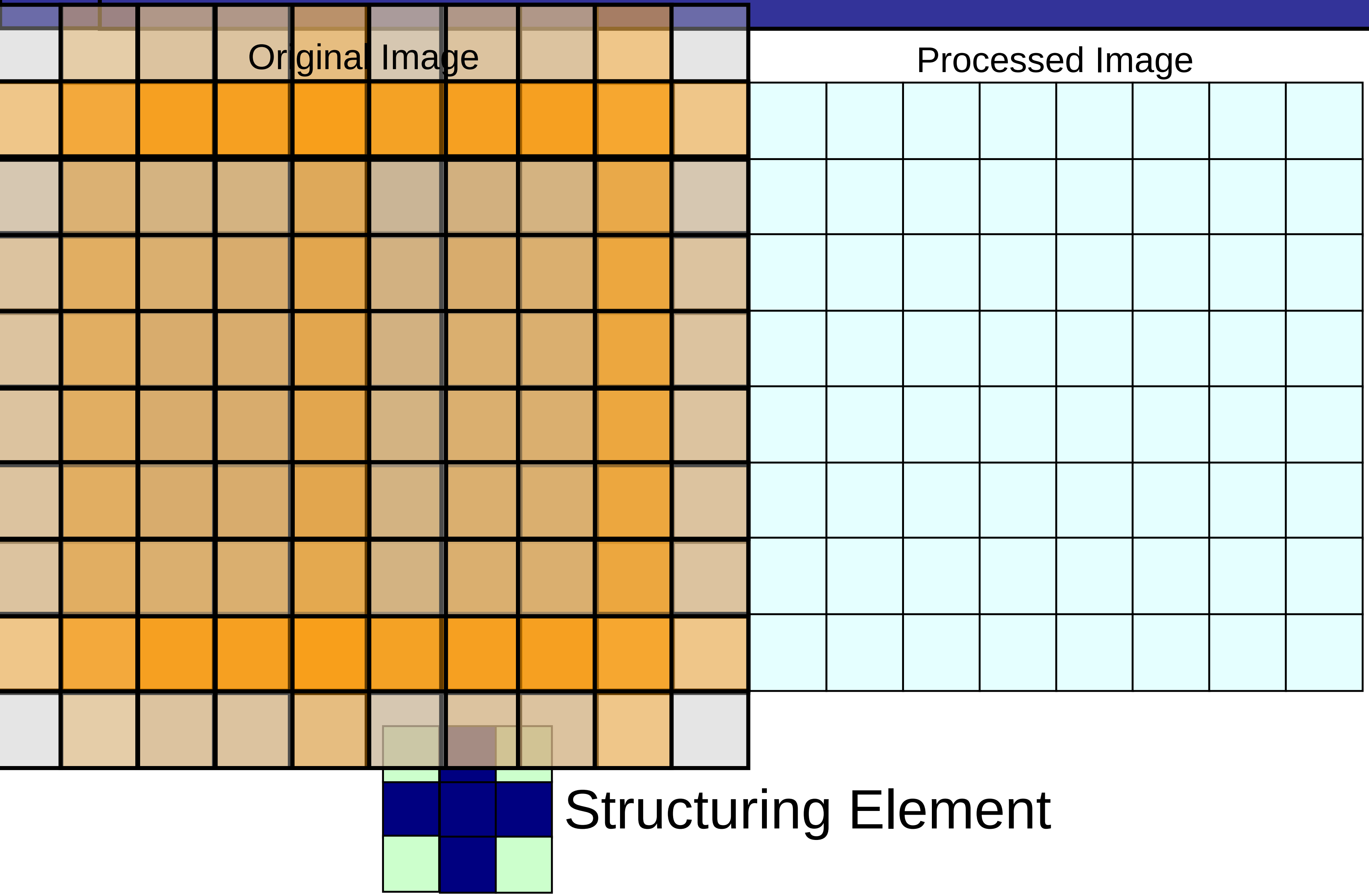
Dilation is an operation that thickens or enlarges objects in a binary image.

Dilation of image f by structuring element s is given by: $f \oplus s$.

The structuring element s is positioned with its origin at (x, y) and the new pixel value (dilated pixel) is determined using the rule:

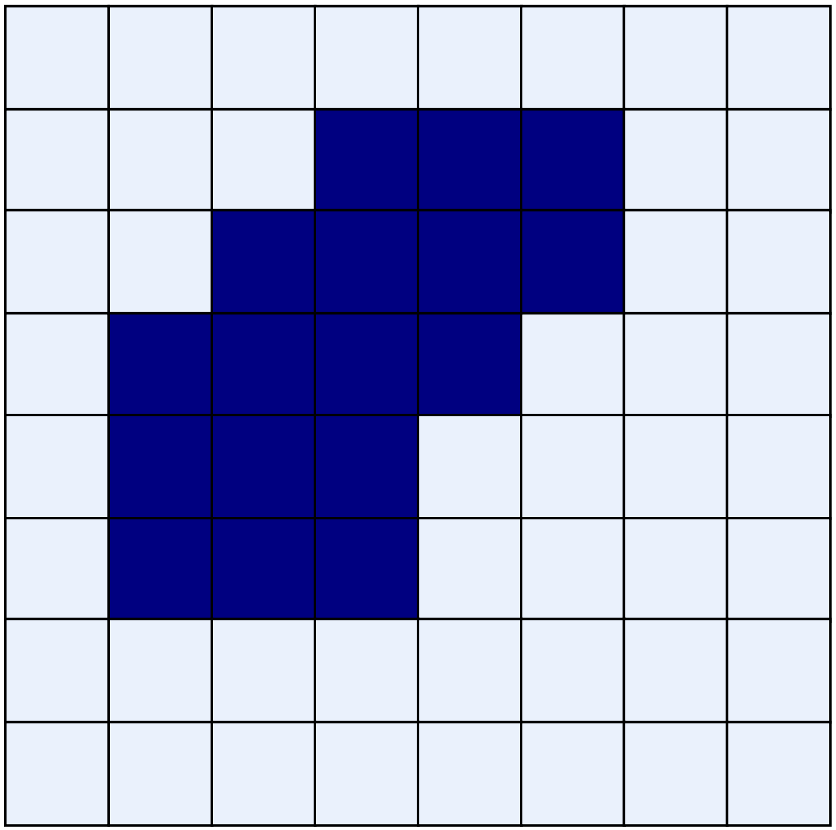
$$g(x, y) = \begin{cases} 1 \text{ if } s \text{ hits } f \\ 0 \text{ otherwise} \end{cases}$$

Dilation Example

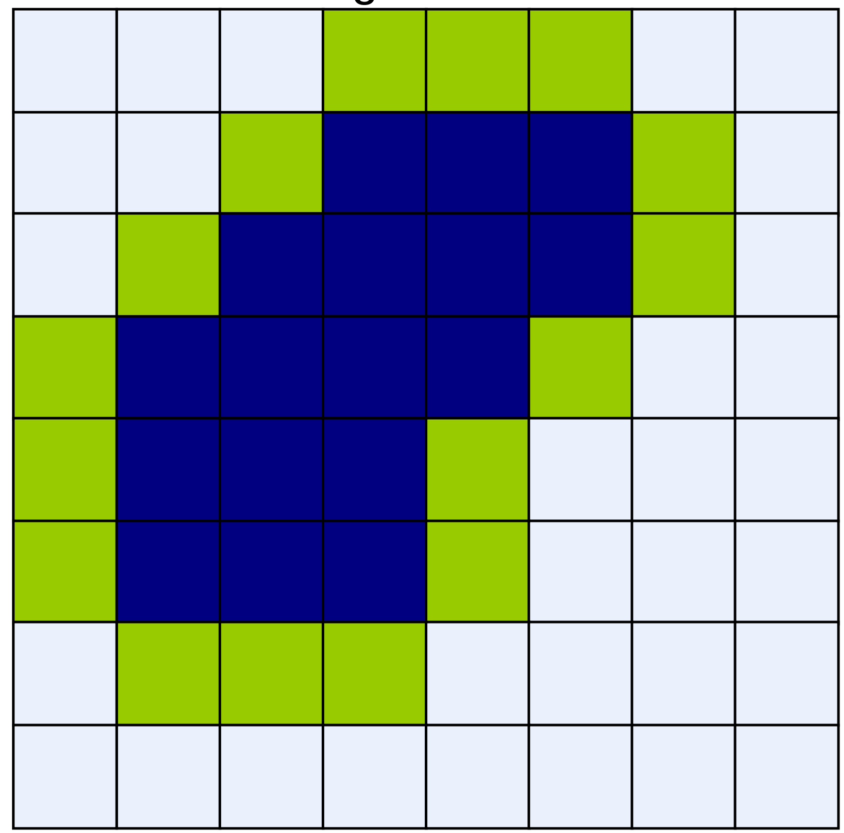


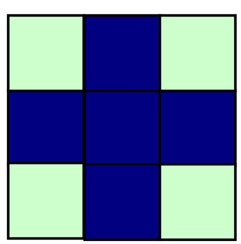
Dilation Example

Original Image



Processed Image With Dilated Pixels





Structuring Element

Dilation Example 1

Original image

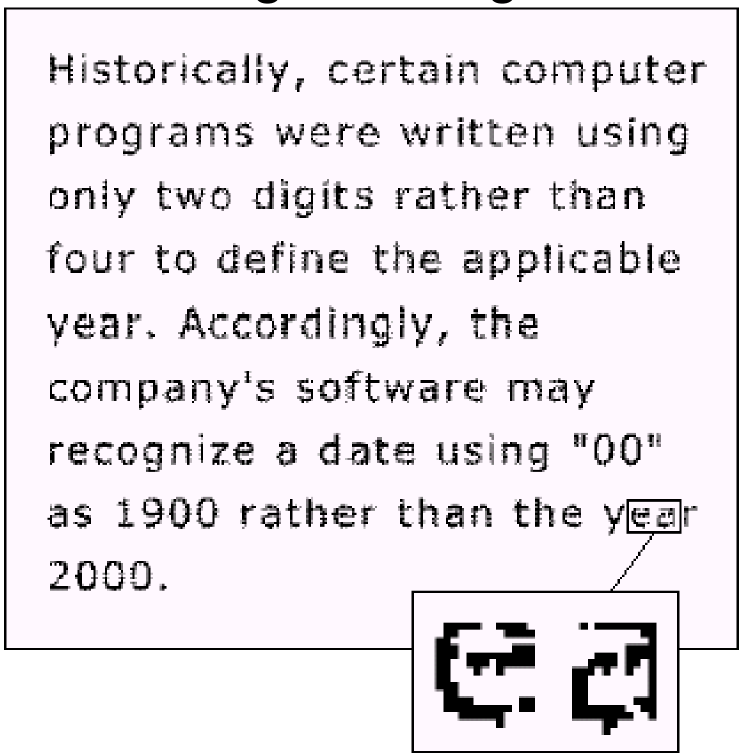
Dilation by 3*3 square structuring element

Dilation by 5*5 square structuring element

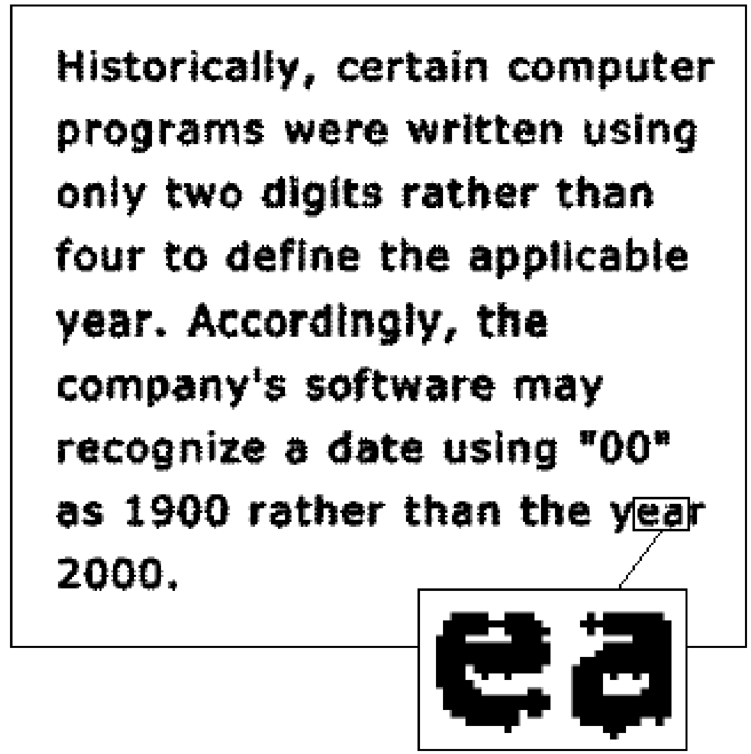
Watch out: In these examples a 1 refers to a black pixel!

Dilation Example 2

Original image



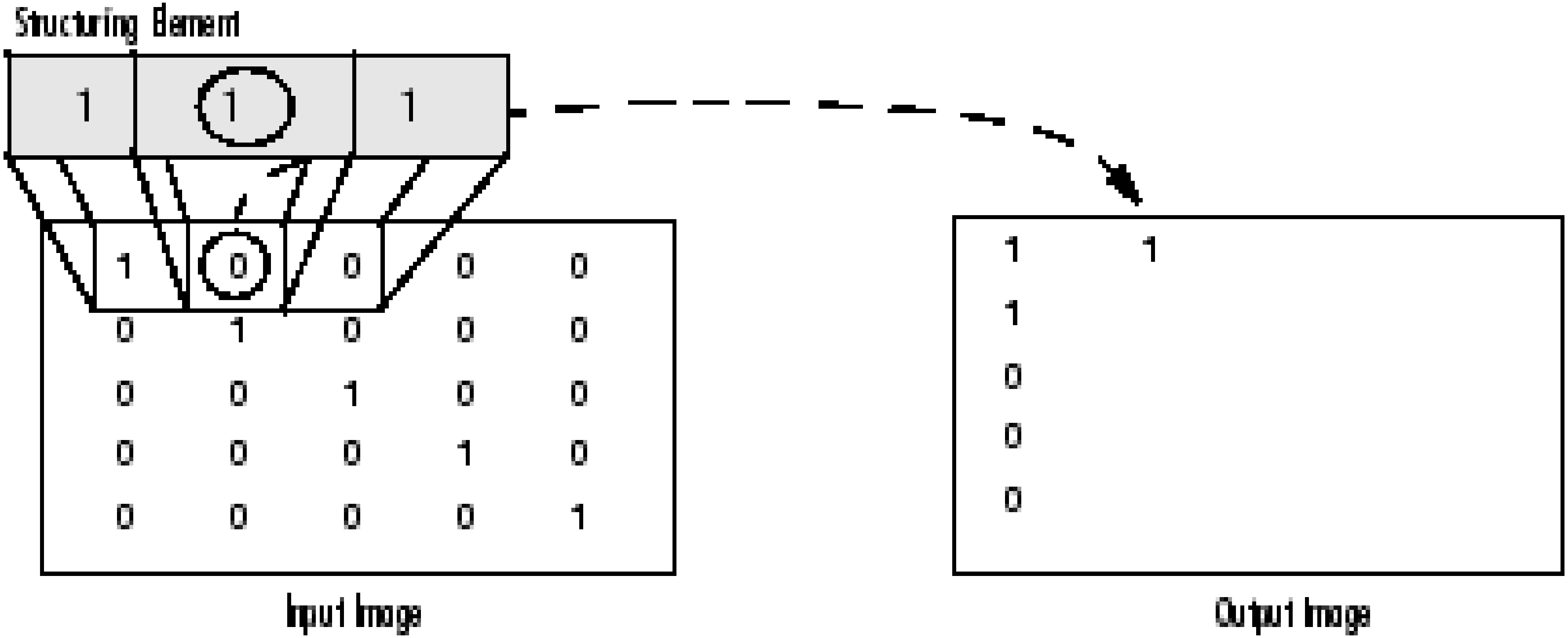
After dilation



0	1	0
1	1	1
0	1	0

Structuring element

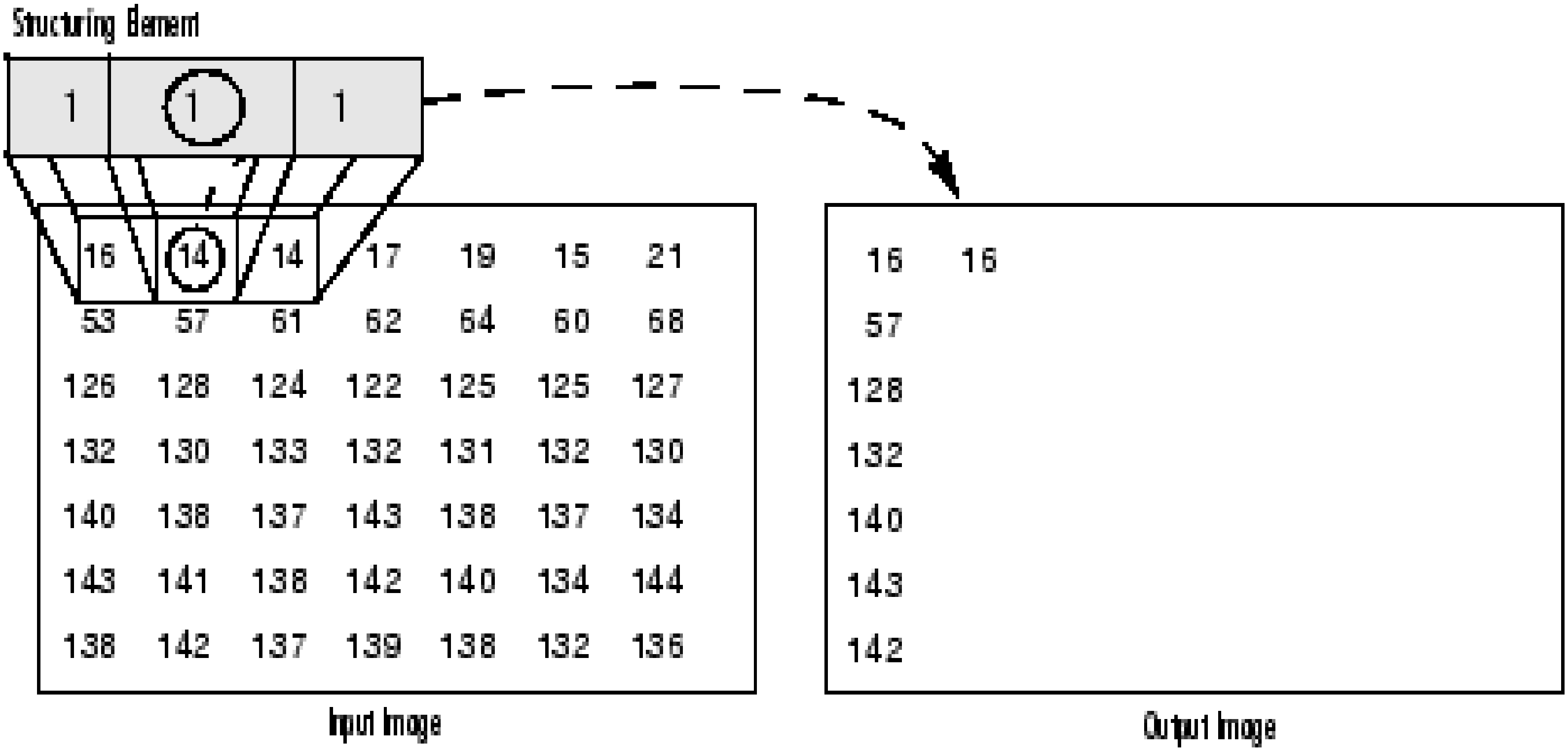
Dilation Example 3



In morphological dilation of a binary image, if any of the pixels is 1 within the neighborhood, the output pixel is set to 1.

Dilation of a Grayscale Image

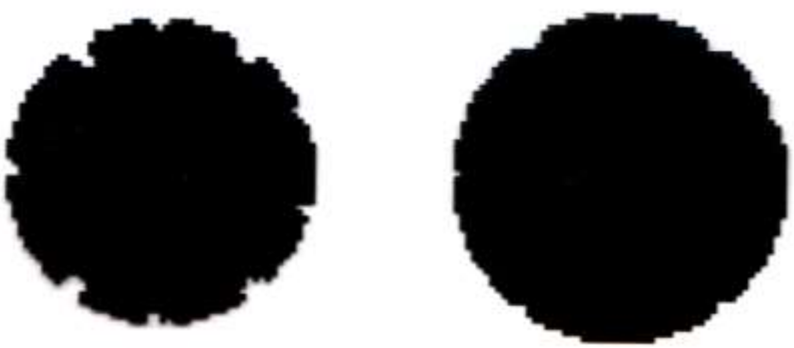
The value of the output pixel is the ***maximum*** value of all the pixels in the input pixel's neighborhood.



What Is Dilation For?

Dilation can repair breaks:

Dilation can repair intrusions:



Watch out: Dilation enlarges objects.

Erosion

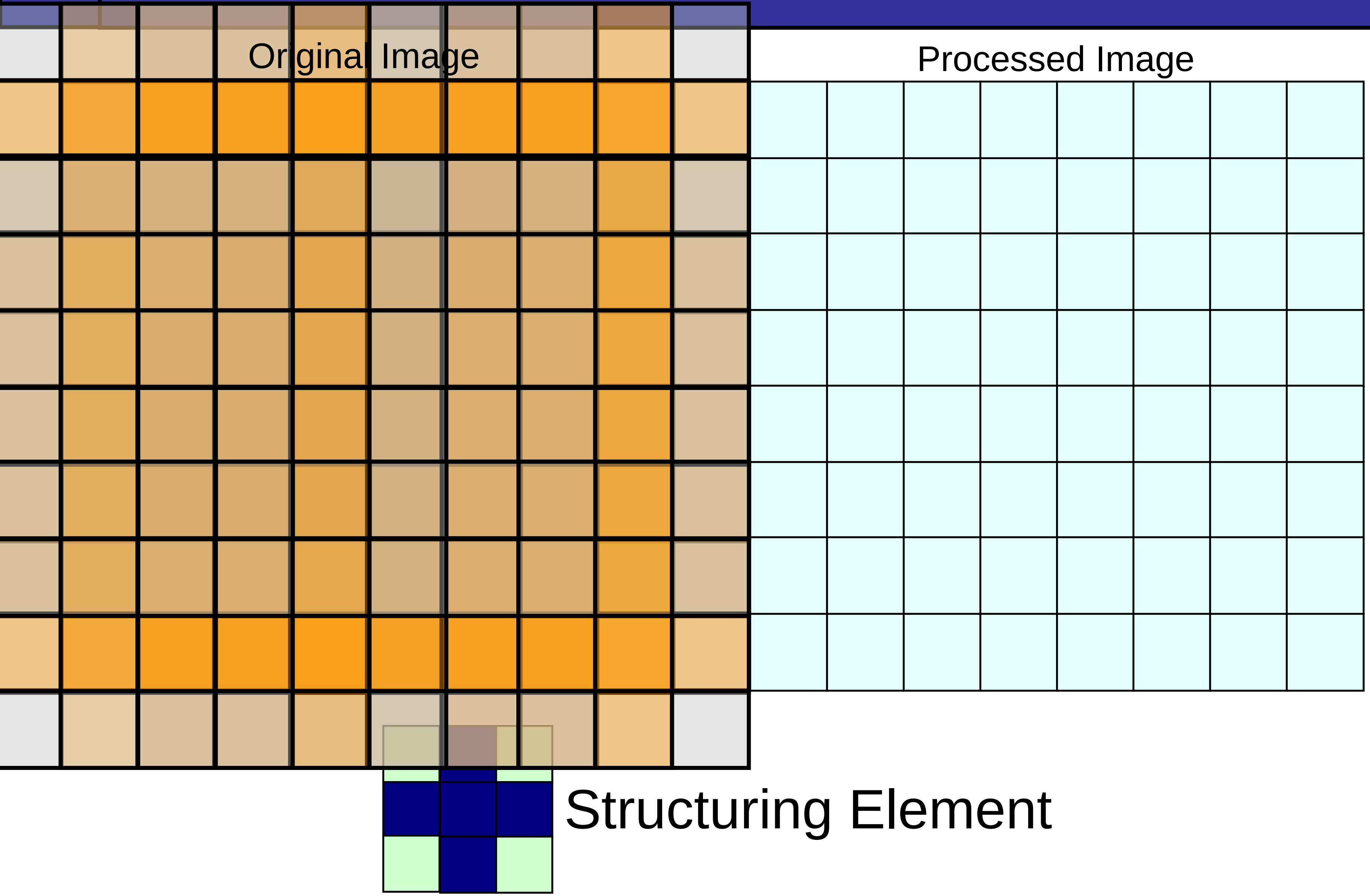
Erosion operation shrinks or thins objects in a binary image.

Erosion of image $\boldsymbol{f}$ by structuring element $\boldsymbol{s}$ is given by, $\boldsymbol{f} \ominus \boldsymbol{s}$.

The structuring element s is positioned with its origin at (x, y) and the new pixel value is determined using the rule:

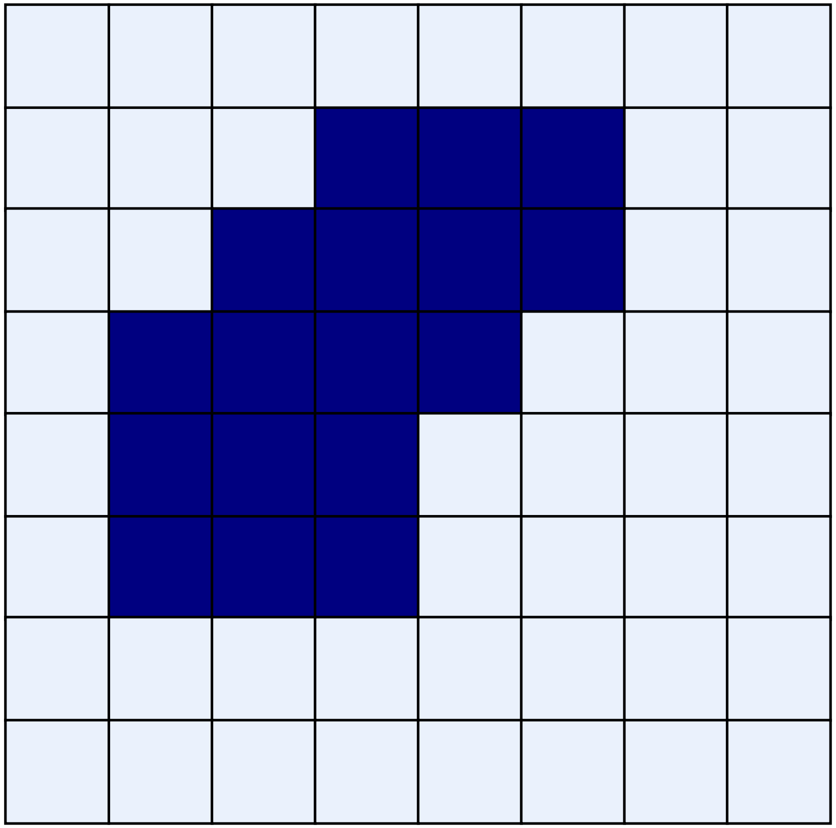
$$g(x, y) = \begin{cases} 1 \text{ if } s \text{ fits } f \\ 0 \text{ otherwise} \end{cases}$$

Erosion Example

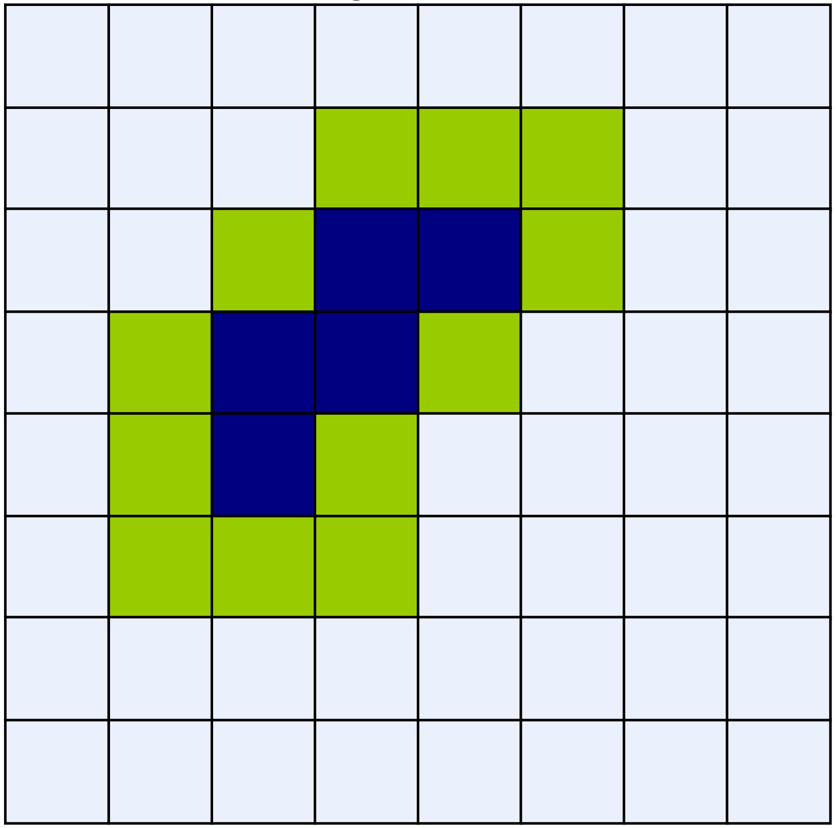


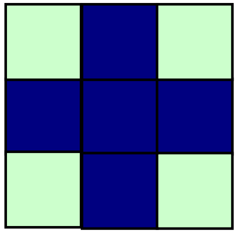
Erosion Example

Original Image



Processed Image With Eroded Pixels

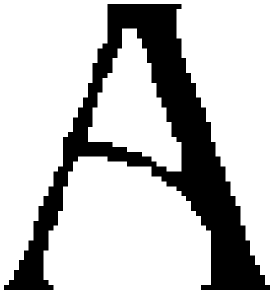




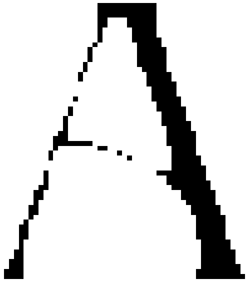
Structuring Element

Erosion Example 1

Original image



Erosion by 3*3 square structuring element



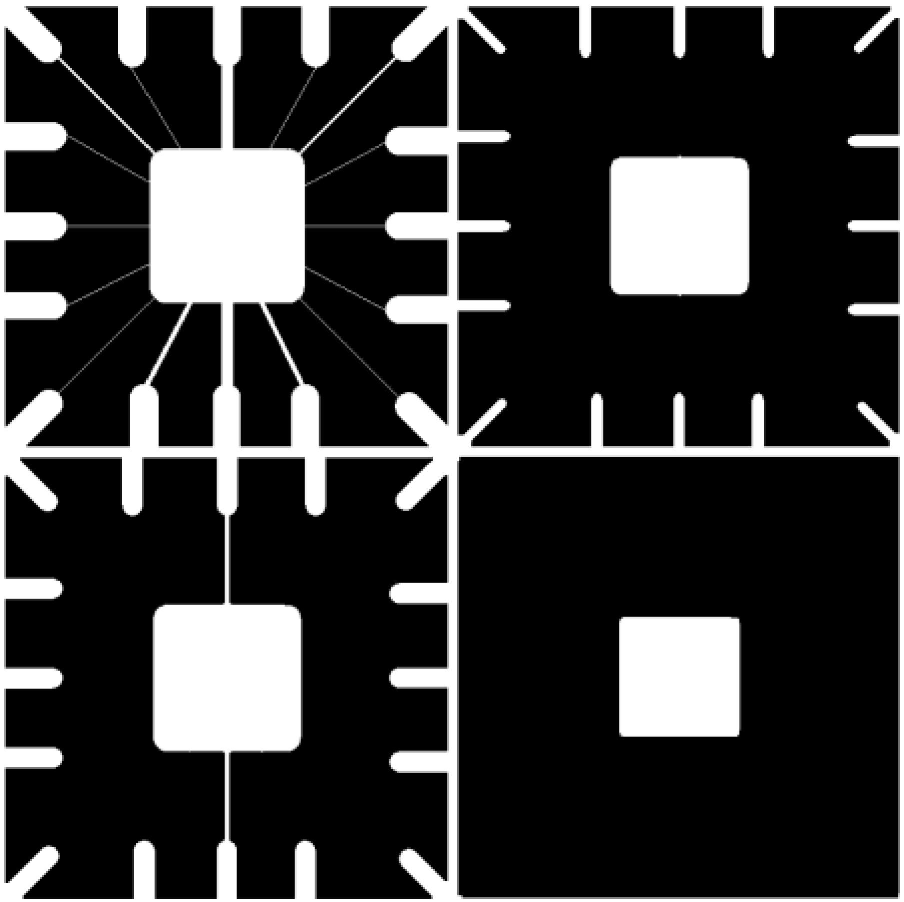
Erosion by 5*5 square structuring element

Watch out: In these examples a 1 refers to a black pixel!

Erosion Example 2

Original image

After erosion with a disc of radius 10

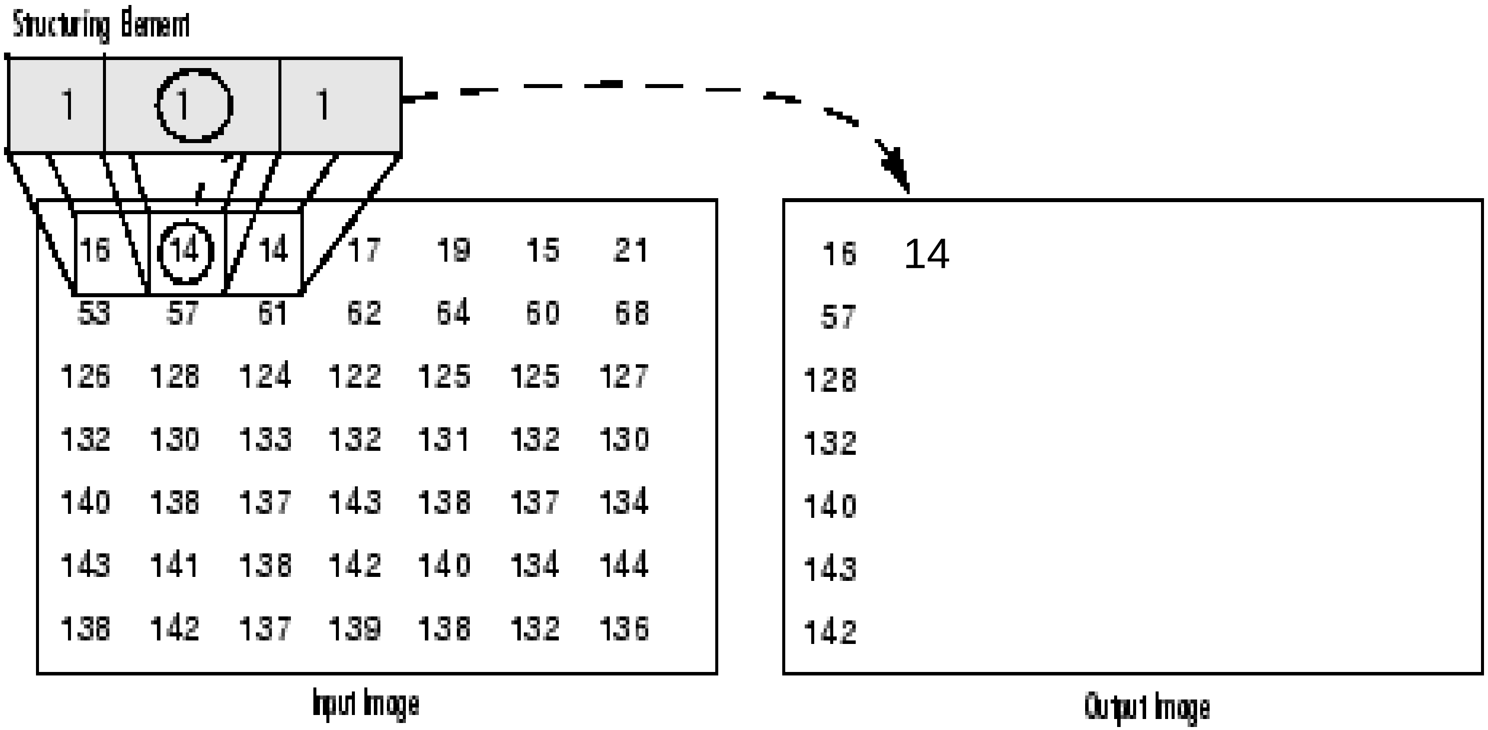


After erosion with a disc of radius 5

After erosion with a disc of radius 20

Erosion of a Grayscale Image

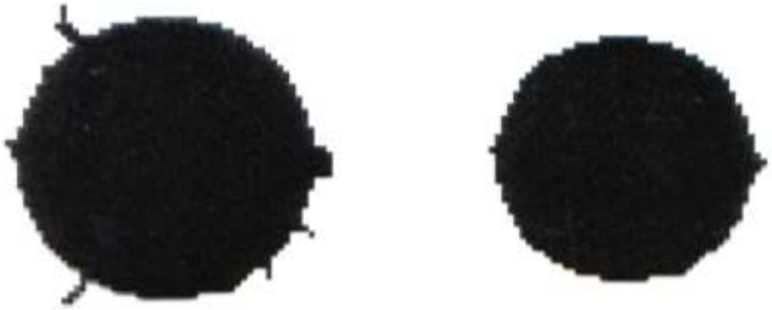
The value of the output pixel is the ***minimum*** value of all the pixels in the input pixel's neighborhood.



What Is Erosion For?

Erosion can split apart joined objects:

Erosion can strip away extrusions;



Watch out: Erosion shrinks or thins objects.

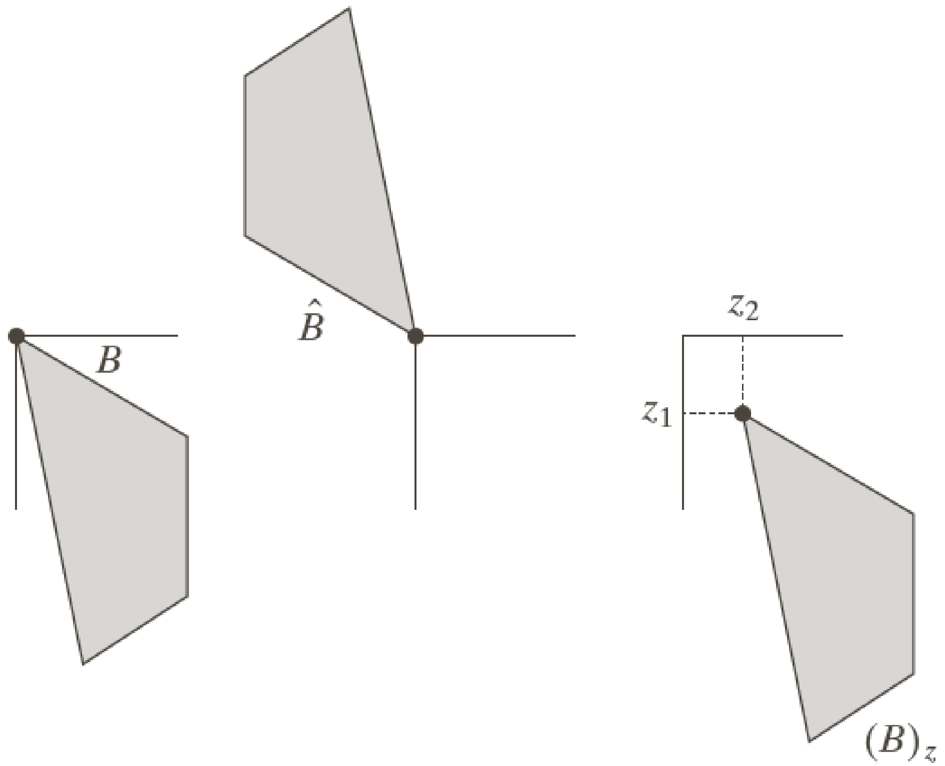
Processing Pixels at Image Borders (Padding Behavior)

- Morphological functions position the origin of the structuring element, its center element, over the pixel of interest in the input image. For pixels at the edge of an image, parts of the neighborhood defined by the structuring element can extend past the border of the image.
- To process border pixels, the morphological functions assign a value to these undefined pixels, as if the functions had padded the image with additional rows and columns. The value of these padding pixels varies for dilation and erosion operations. The following table describes the padding rules for dilation and erosion for both binary and grayscale images.

Operation	**Rule**
Dilation	Pixels beyond the image border are assigned the ***minimum*** value afforded by the data type. For binary images, these pixels are assumed to be set to 0. For grayscale images, the minimum value is 0.
Erosion	Pixels beyond the image border are assigned the ***maximum*** value afforded by the data type. For binary images, these pixels are assumed to be set to 1. For grayscale images, the maximum value is 255.

Mathematical Notation

Operation	Equation	Comments (The Roman numerals refer to the structuring elements in Fig. 9.33.)
Translation	$(B)_z = \{w \mid w = b + z, \text{ for } b \in B\}$	Translates the origin of B to point z.
Reflection	$\hat{B} = \{w \mid w = -b, \quad \text{for } b \in B\}$	Reflects all elements of B about the origin of this set.
Complement	$A^c = \{w \mid w \notin A\}$	Set of points not in A.
Difference	$A - B = \{w \mid w \in A, w \notin B\} = A \cap B^c$	Set of points that belong to A but not to B.
Dilation	$A \oplus B = \left\{z \mid (\hat{B}_z) \cap A \neq \varnothing\right\}$	"Expands" the boundary of A. (I)
Erosion	$A \ominus B = \left\{z \mid (B)_z \subseteq A\right\}$	"Contracts" the boundary of A. (I)
Opening	$A \circ B = (A \ominus B) \oplus B$	Smoothes contours, breaks narrow isthmuses, and eliminates small islands and sharp peaks. (I)



a b c

FIGURE 9.1
(a) A set, (b) its reflection, and (c) its translation by z.

Compound Operations

More interesting morphological operations can be performed by performing combinations of erosions and dilations.

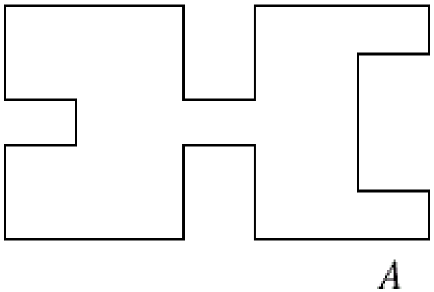
The most widely used of these *compound operations* are:

- Opening
- Closing

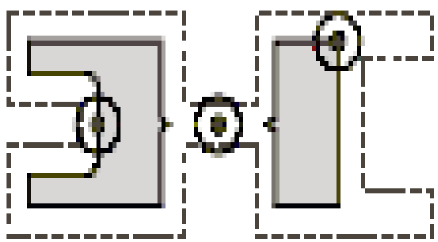
Opening

The opening of image f by structuring element s, denoted $f \circ s$ is simply an erosion followed by a dilation:

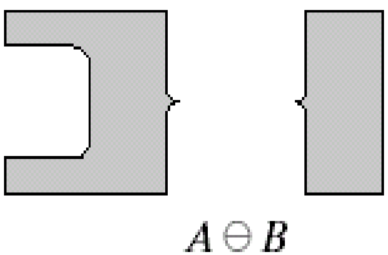
$$f \circ s = (f \ominus s) \oplus s$$



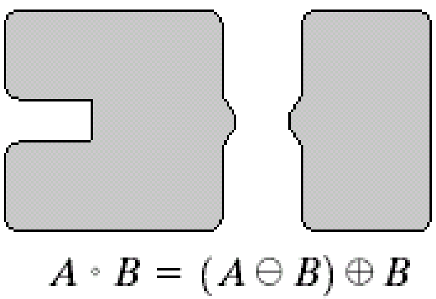
Original shape



Structuring elements
(Small circle in various positions)



After erosion



After dilation
(opening)

Opening Example

Original Image

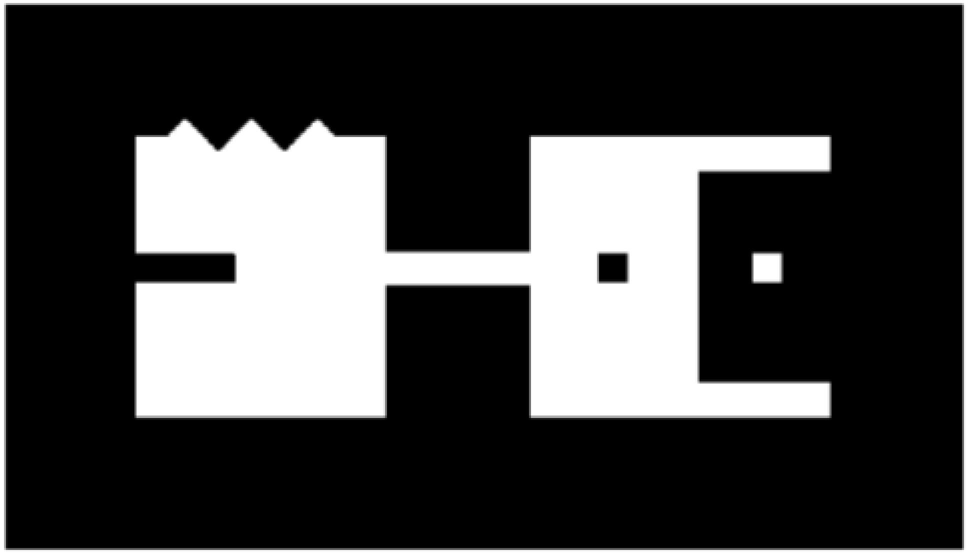
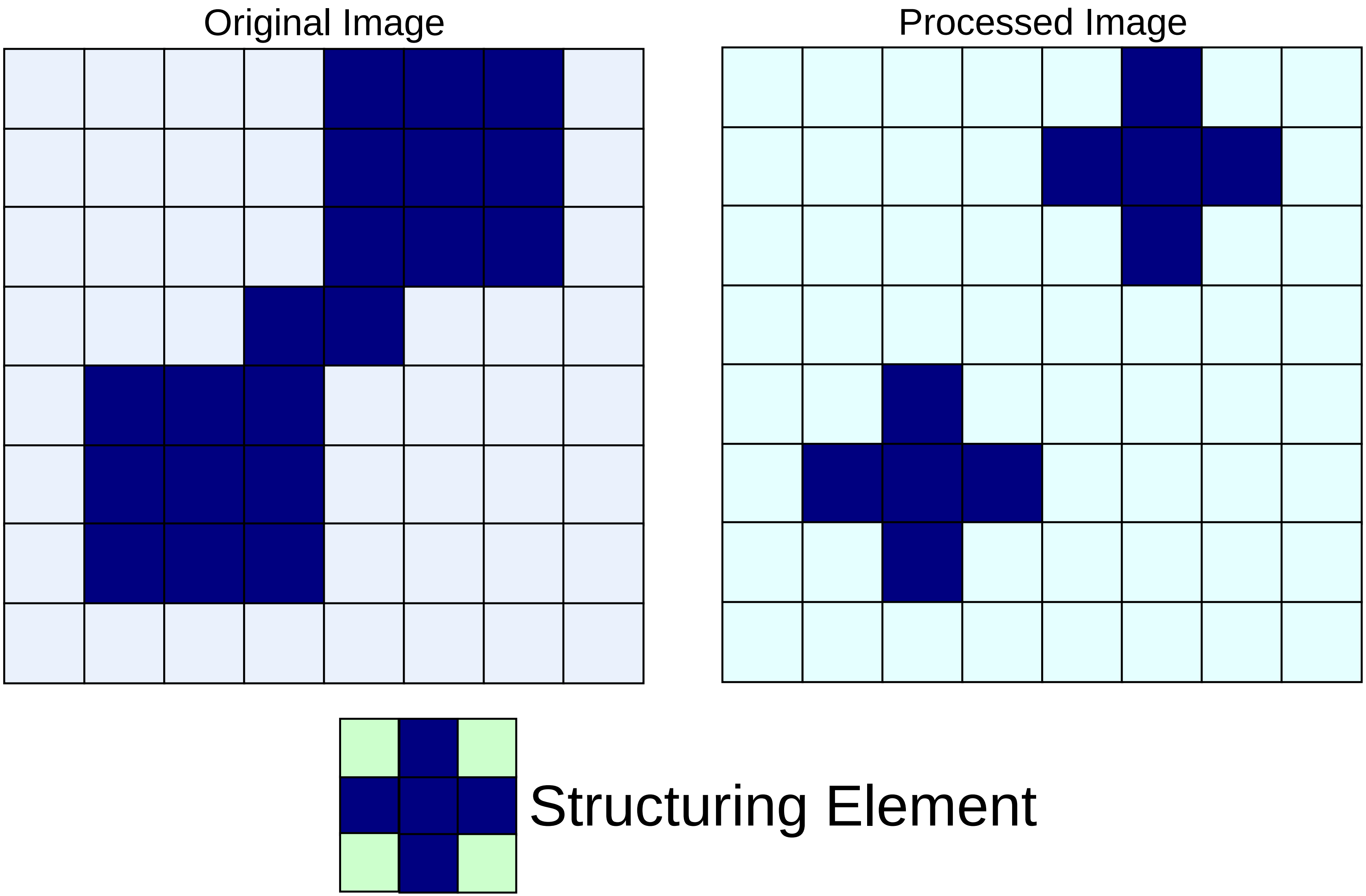


Image After Opening

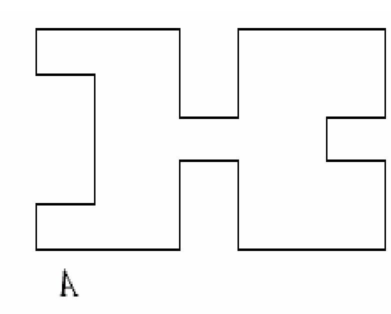
Opening Example



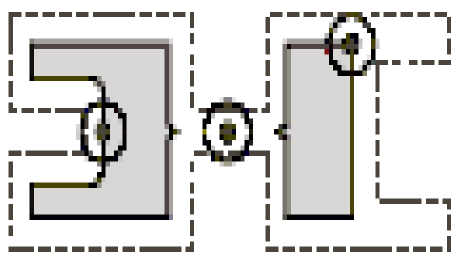
Closing

The closing of image *f* by structuring element *s*, denoted $f \bullet s$ is simply a dilation followed by an erosion

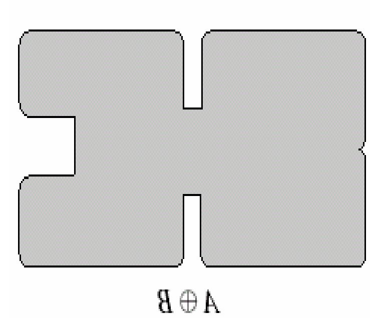
$$f \bullet s = (f \oplus s) \ominus s$$



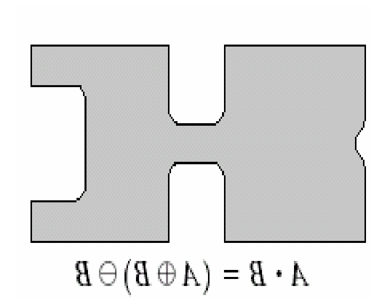
Original shape



Structuring elements
(Small circle in various positions)



After dilation



After erosion
(closing)

Closing Example

Original Image

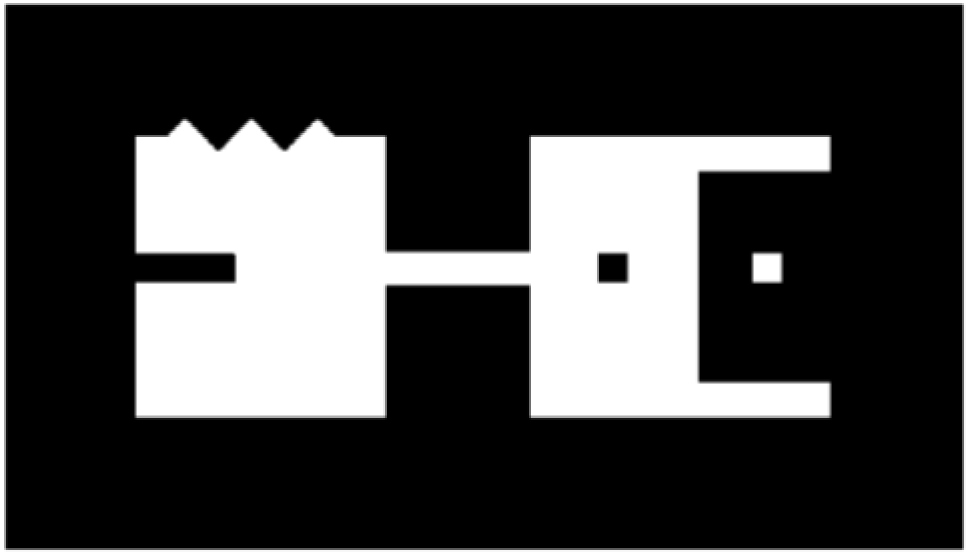
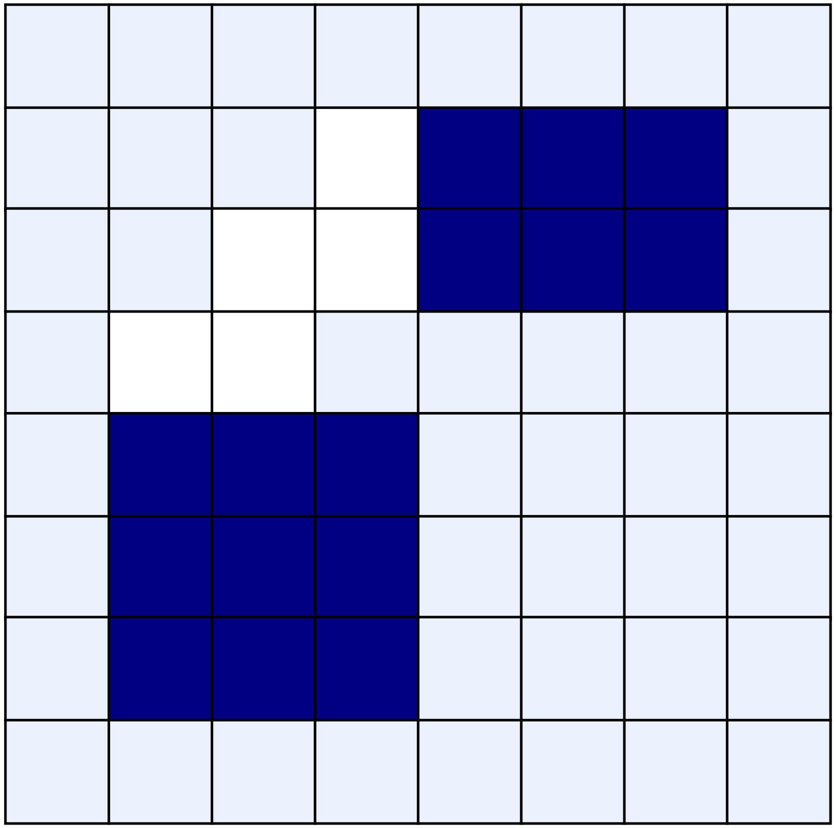


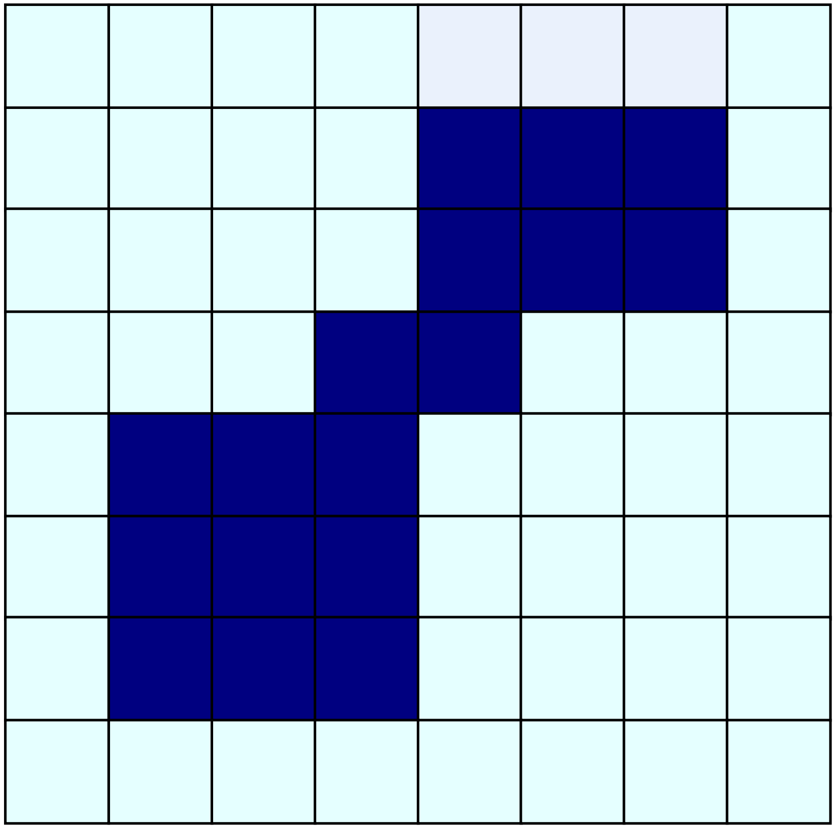
Image After Closing

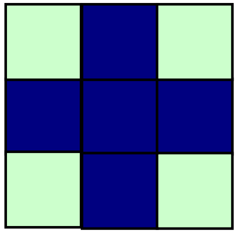
Closing Example

Original Image



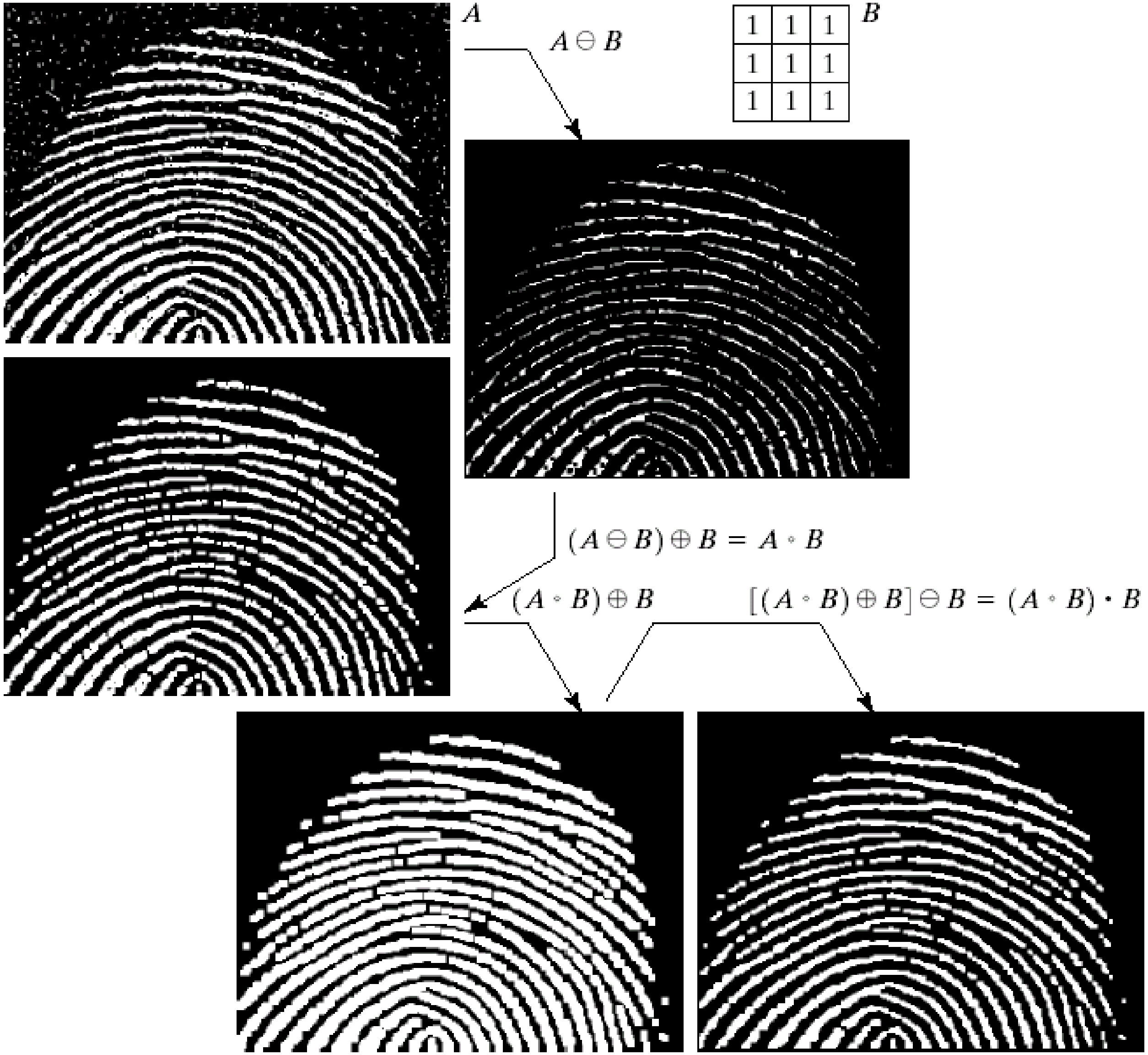
Processed Image





Structuring Element

Morphological Processing Example



Morphological Algorithms

Using the simple technique we have looked at so far we can begin to consider some more interesting morphological algorithms

We will look at:

- Boundary extraction
- Region filling

There are lots of others as well though:

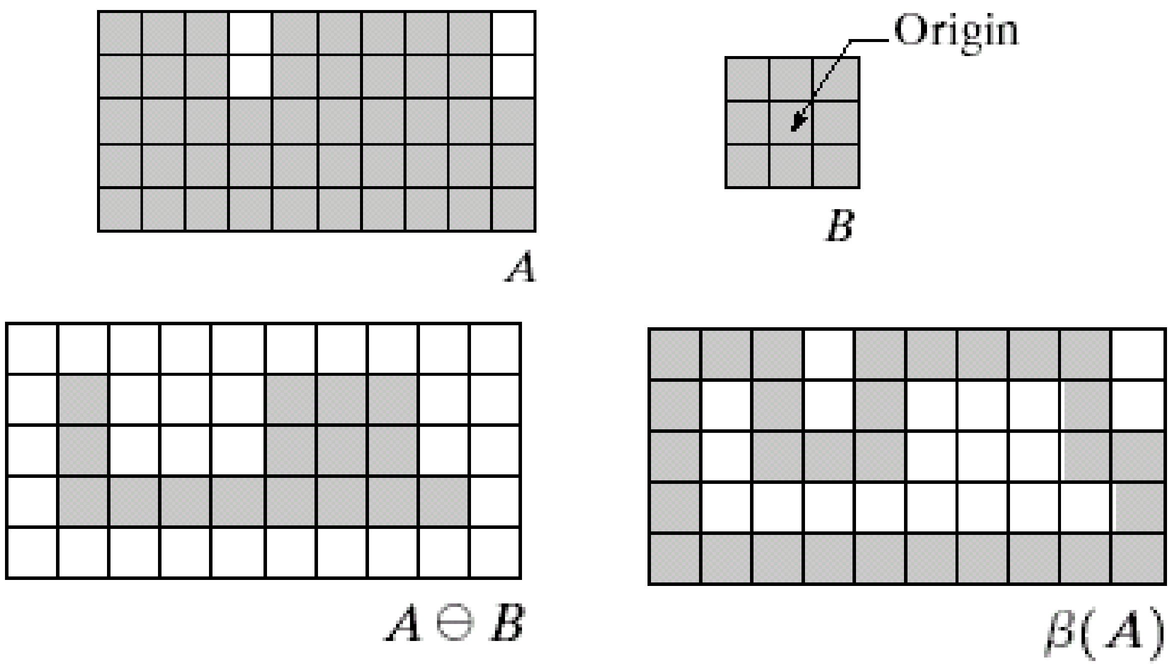
- Extraction of connected components
- Thinning/thickening
- Skeleton (Skeletonisation)
- Pruning

Boundary Extraction

Extracting the boundary (or outline) of an object is often extremely useful

The boundary can be given simply as

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



Boundary Extraction Example

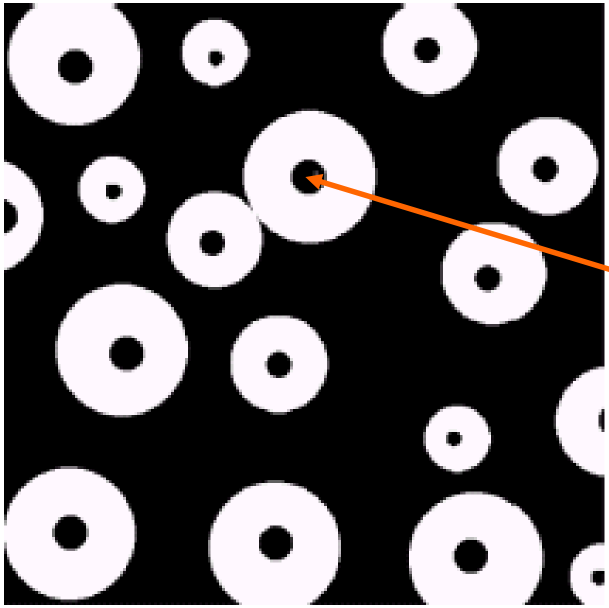
A simple image and the result of performing boundary extraction using a square 3*3 structuring element

Original Image

Extracted Boundary

Region Filling

Given a pixel inside a boundary, *region filling* attempts to fill that boundary with object pixels (1s)



Given a point inside here, can we fill the whole circle?

Region Filling (cont…)

The key equation for region filling is

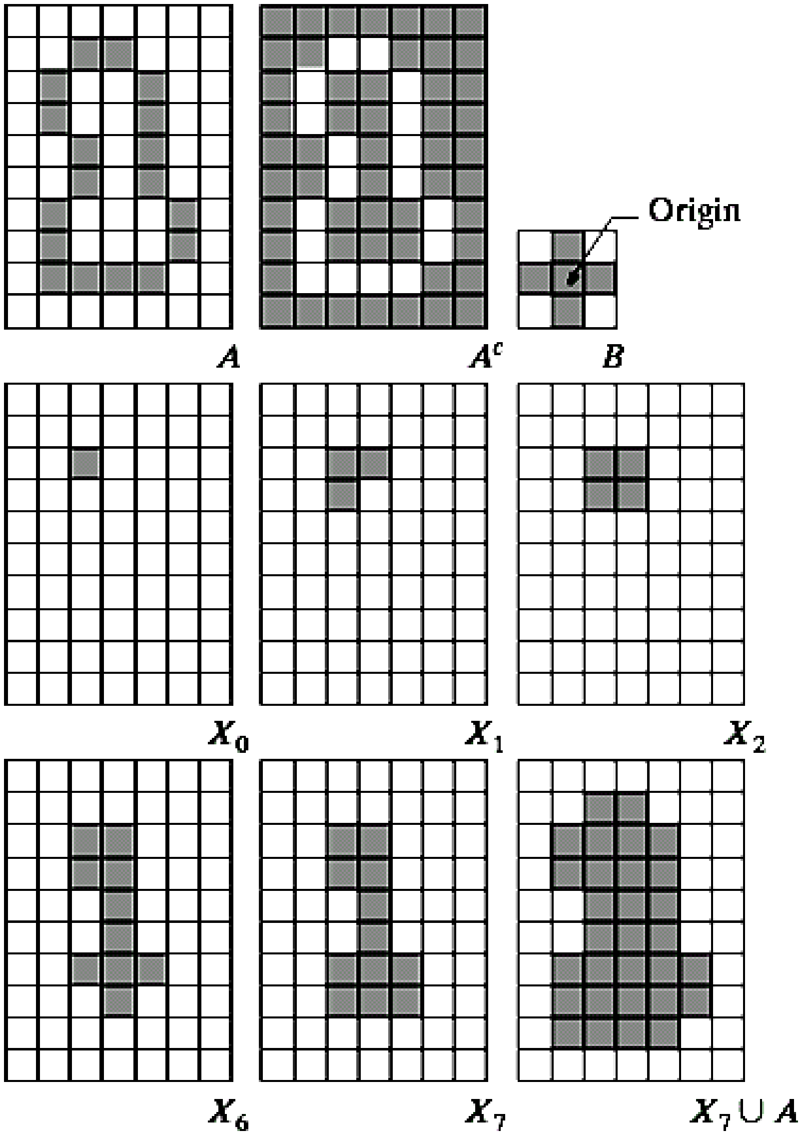
$$X_k = (X_{k-1} \oplus B) \cap A^c \qquad k = 1,2,3.....$$

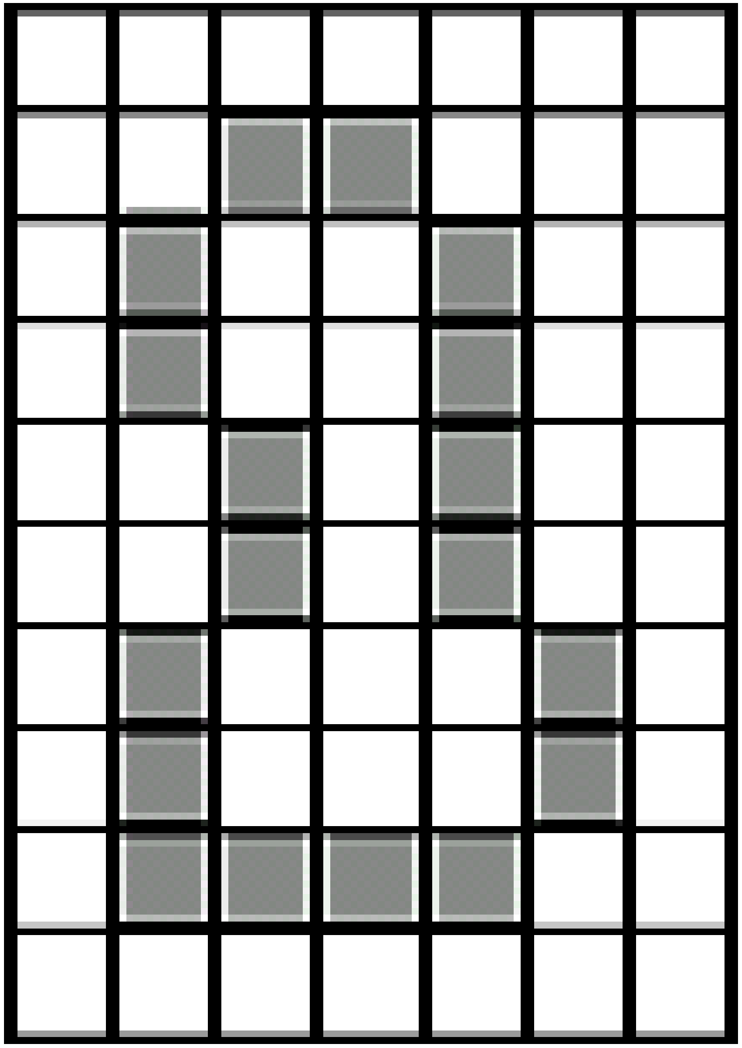
Where X_0 is simply the starting point inside the boundary, B is a simple structuring element and A^c is the complement of A

This equation is applied repeatedly until X_k is equal to X_{k-1}

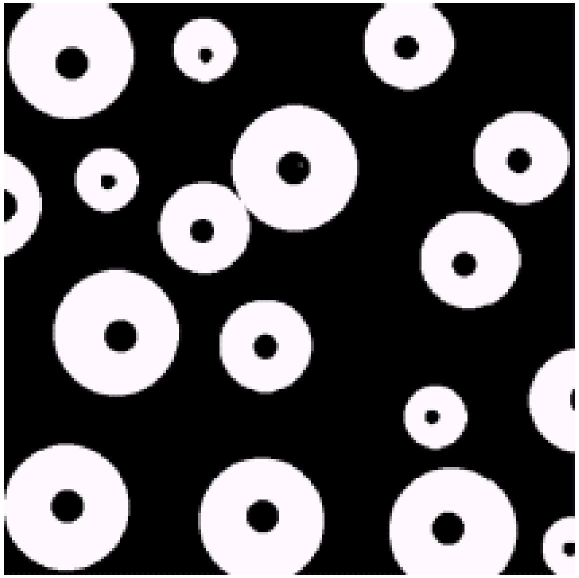
Finally the result is unioned with the original boundary

Region Filling Step By Step

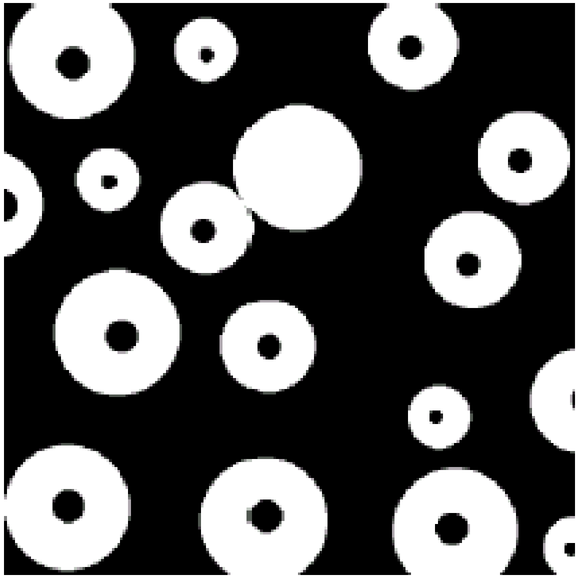




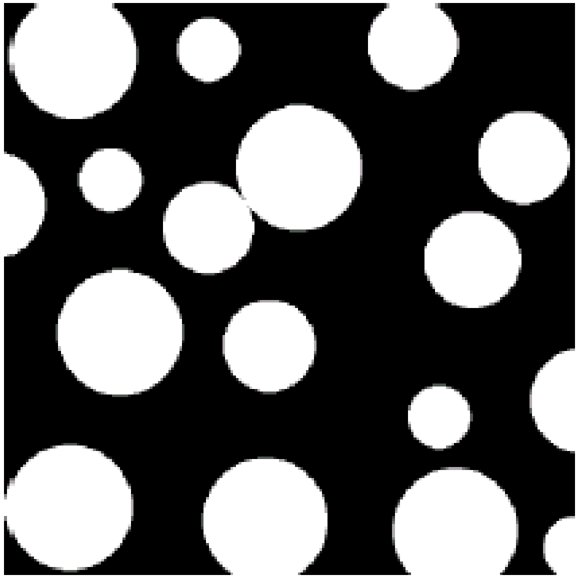
Region Filling Example



Original Image



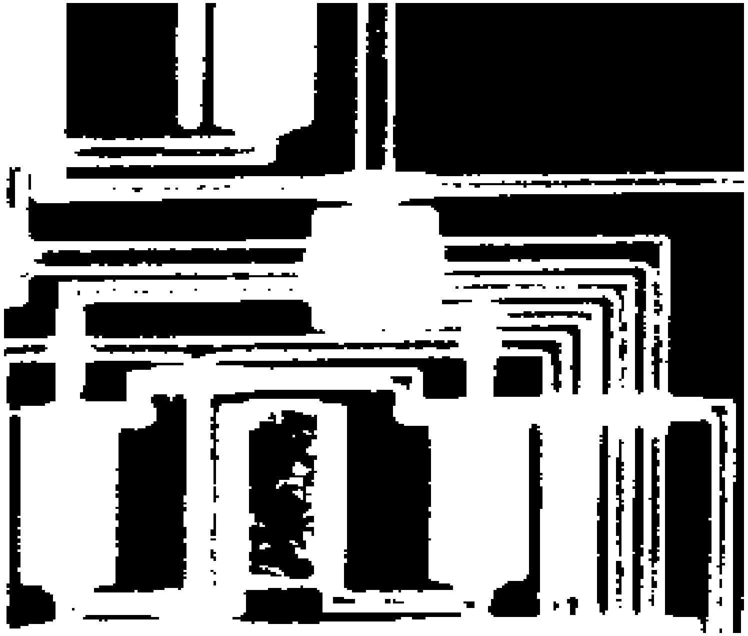
One Region Filled



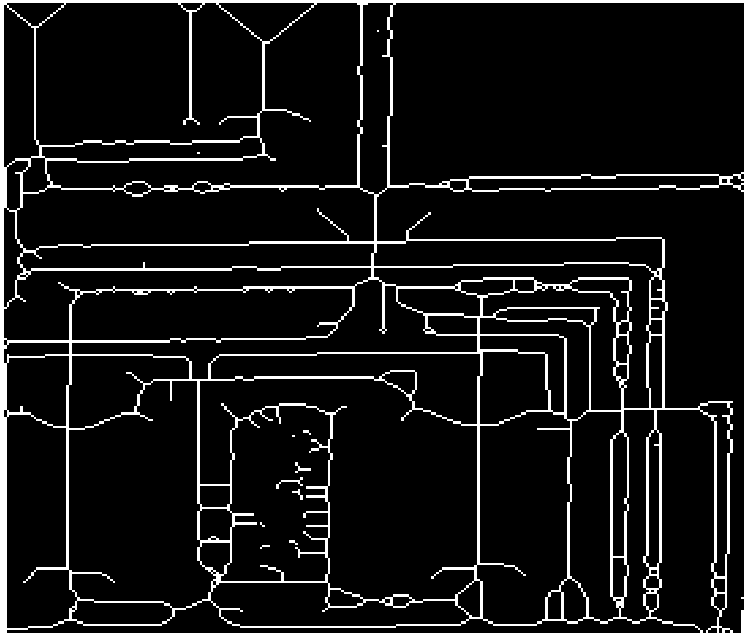
All Regions Filled

Skeletonization

- To reduce all objects in an image to lines, without changing the essential structure of the image, we use the process known as *skeletonization*.
- *Check Lantuéjoul's formula*



Original Image

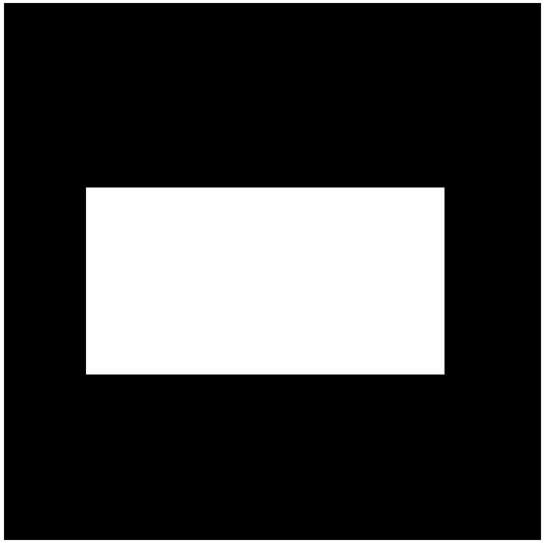


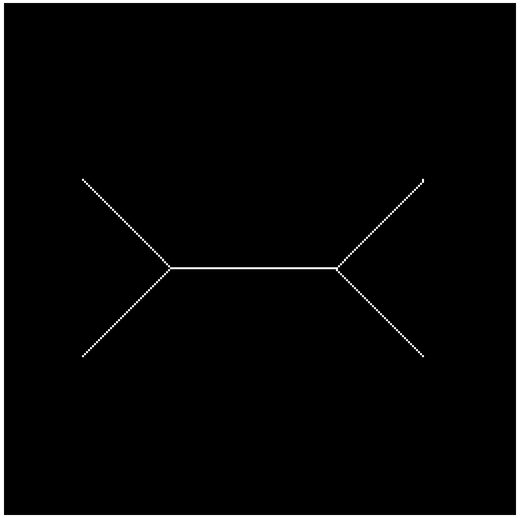
Skeletonization of Image

Skeletonization

- Input

Skeleton





Summary

The purpose of morphological processing is primarily to remove imperfections added during segmentation

The basic operations are *erosion* and *dilation*

Using the basic operations we can perform *opening* and *closing*

More advanced morphological operation can then be implemented using combinations of all of these