

# Digital Image Processing

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

# 1, 0, Black, White?

Throughout all of the following slides whether 0 and 1 refer to white or black is a little interchangeable

All of the discussion that follows assumes segmentation has already taken place and that images are made up of 0s for background pixels and 1s for object pixels

After this it doesn't matter if 0 is black, white, yellow, green.....

# What Is Morphology?

Morphological image processing (or *morphology*) describes a range of image processing techniques that deal with the shape (or morphology) of features in an image

Morphological operations are typically applied to remove imperfections introduced during segmentation, and so typically operate on bi-level images

# Quick Example

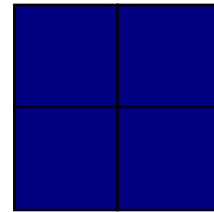
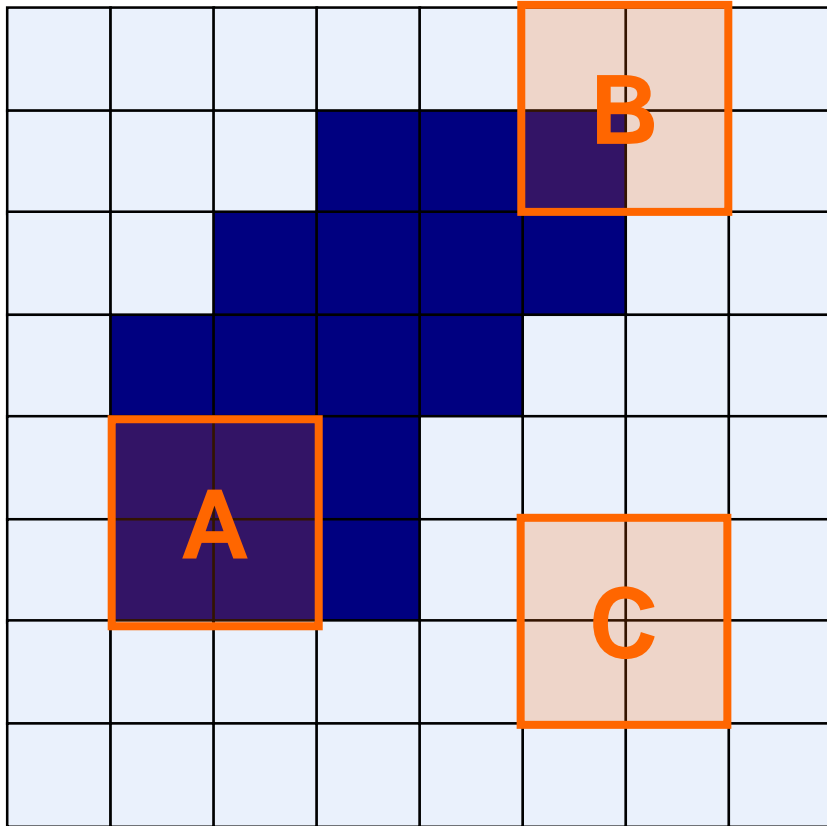


Image after segmentation



Image after segmentation and morphological processing

# Structuring Elements, Hits & Fits



Structuring Element

**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	<b>1</b>	1
1	1	1

0	1	0
1	<b>1</b>	1
0	1	0

0	0	1	0	0
0	1	1	1	0
1	1	<b>1</b>	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>B</b>	1	1	1	0	<b>C</b>	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	<b>A</b>	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

Fundamentally morphological image processing is very like spatial filtering

The structuring element is moved across every pixel in the original image to give a pixel in a new processed image

The value of this new pixel depends on the operation performed

There are two basic morphological operations: **erosion** and **dilation**

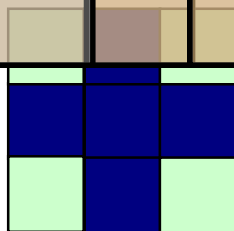
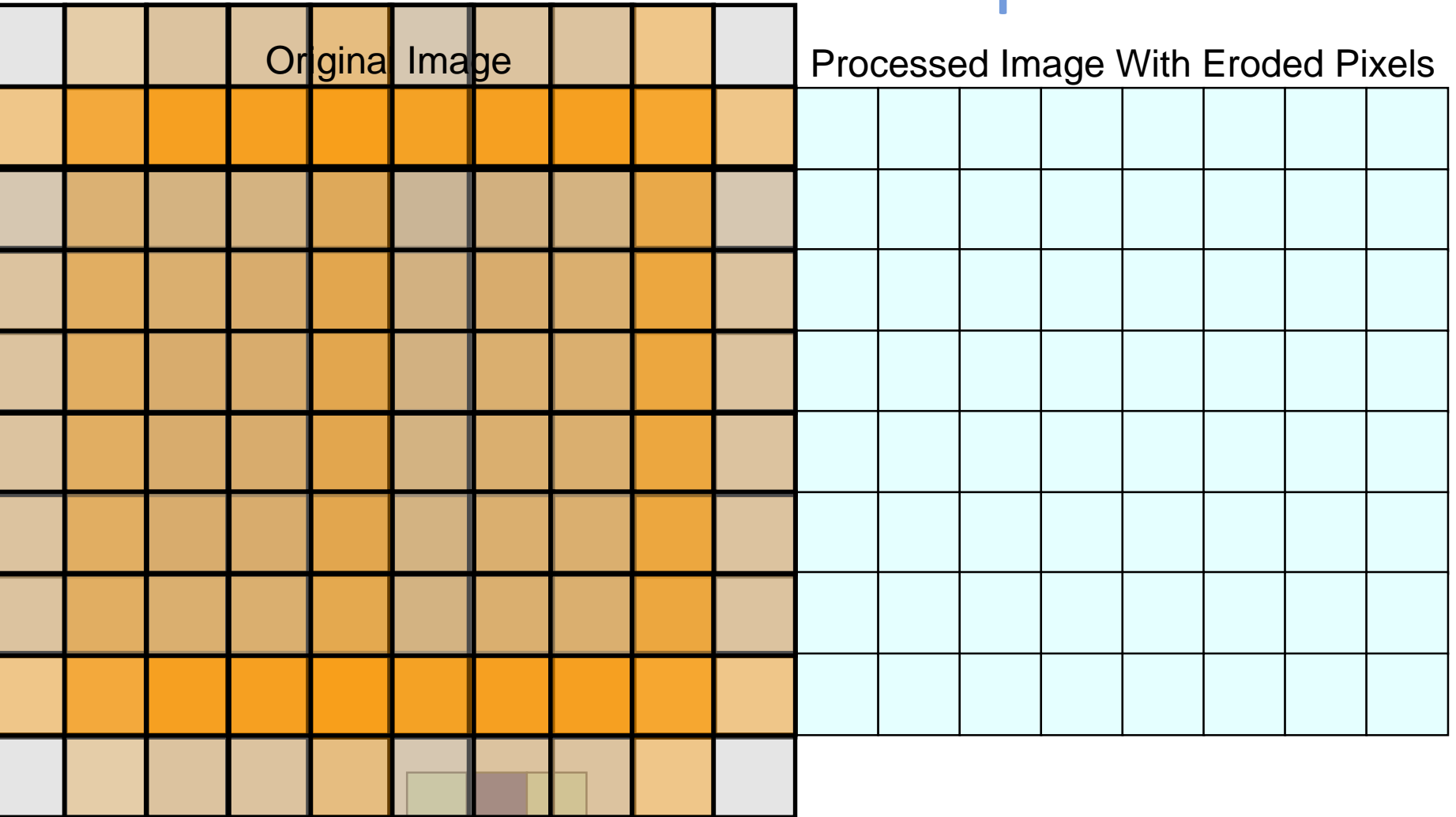
# Erosion

Erosion of image  $f$  by structuring element  $s$  is given by  $f \ominus s$

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

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

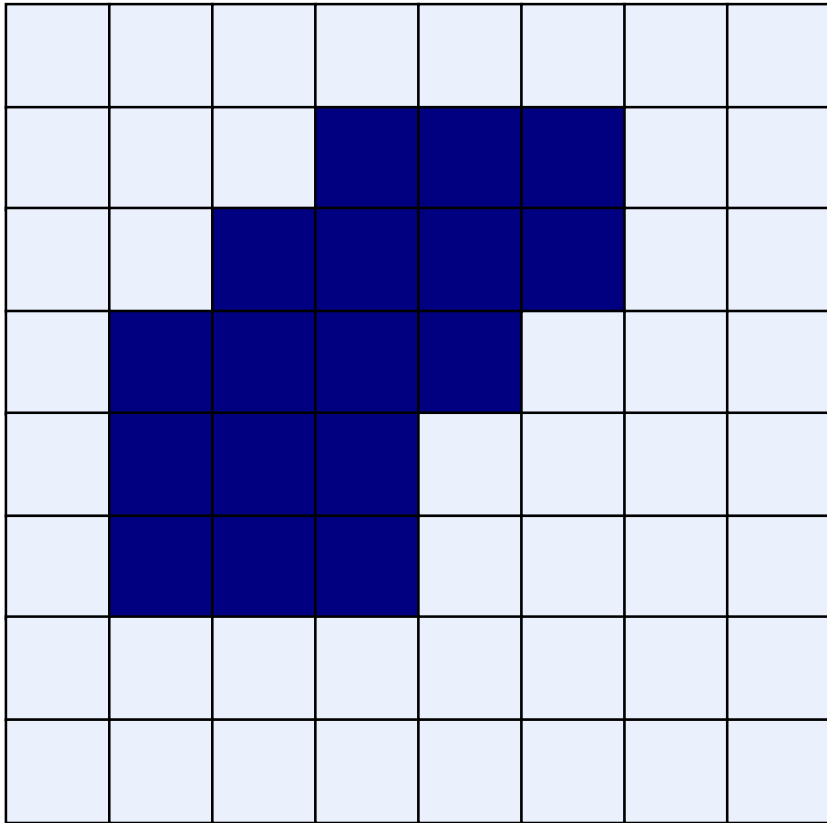
# Erosion Example



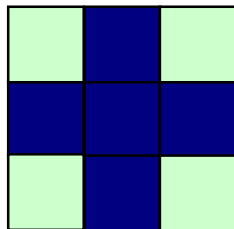
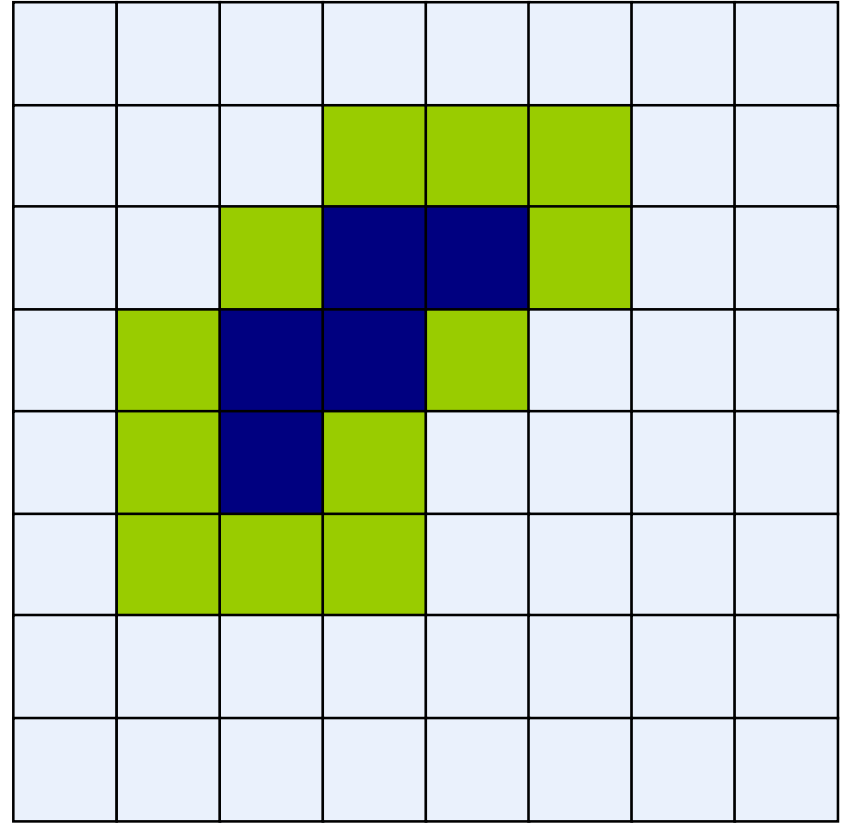
Structuring Element

# Erosion Example

Original Image



Processed Image

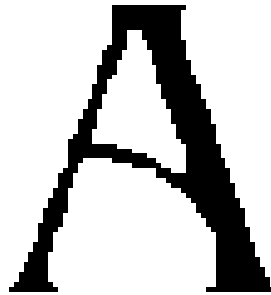


Structuring Element

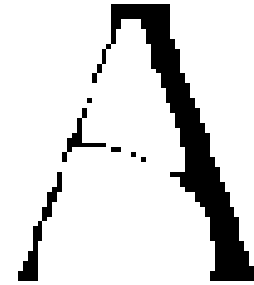
# Erosion Example 1



Original image



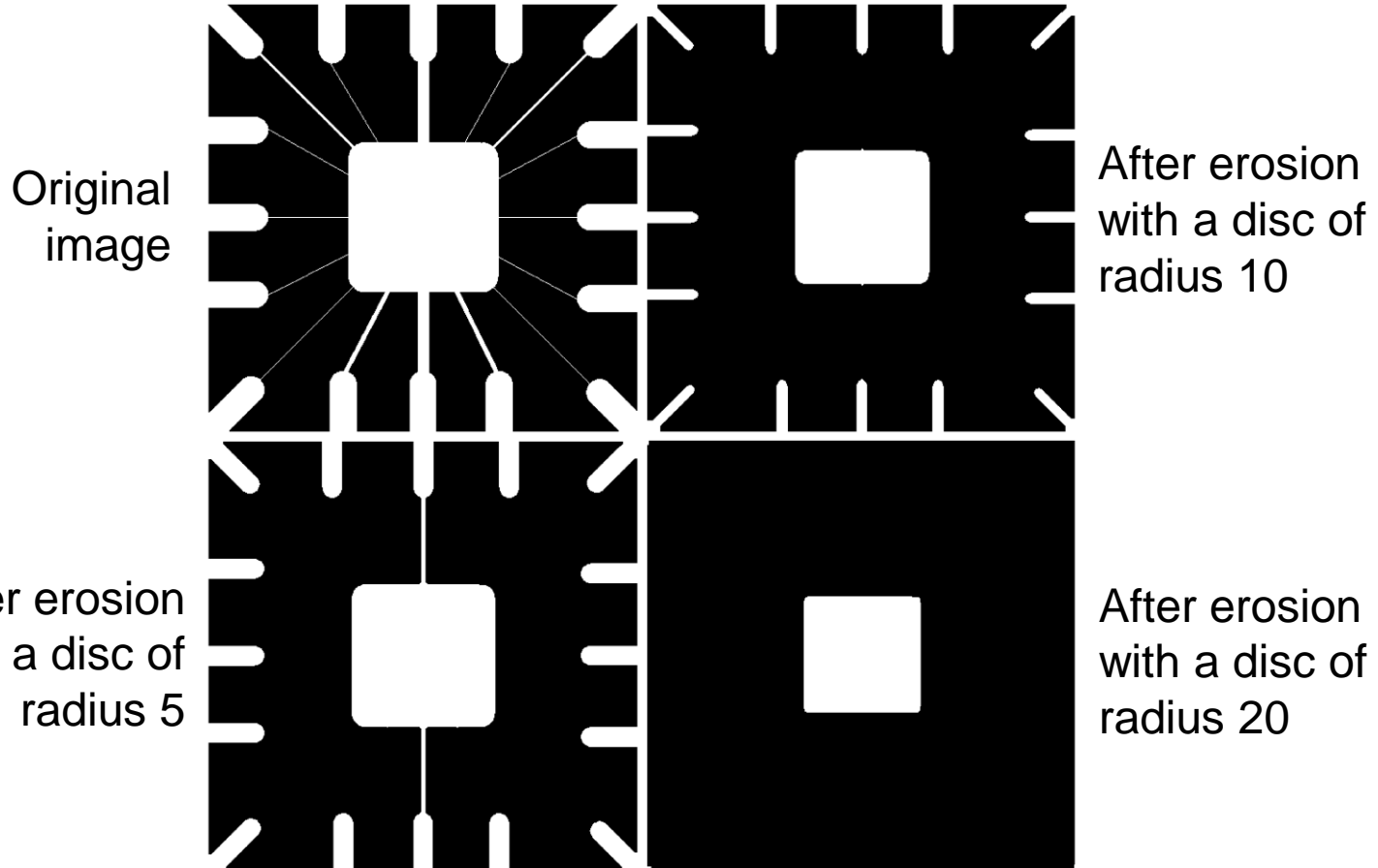
Erosion by 3\*3  
square structuring  
element



Erosion by 5\*5  
square structuring  
element

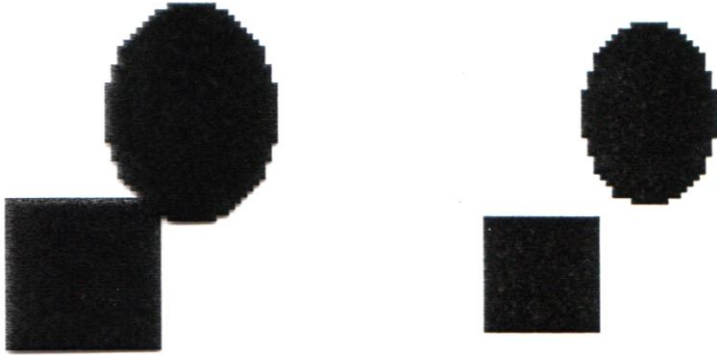
**Watch out:** In these examples a 1 refers to a black pixel!

# Erosion Example 2

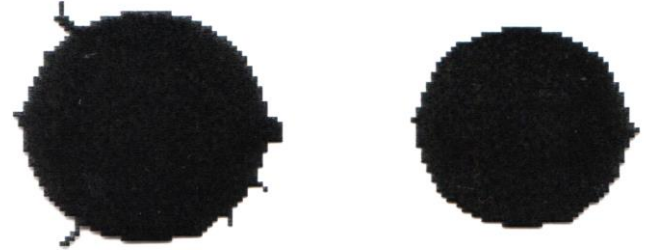


# What Is Erosion For?

Erosion can split apart joined objects



Erosion can strip away extrusions



**Watch out:** Erosion shrinks objects

# Dilation

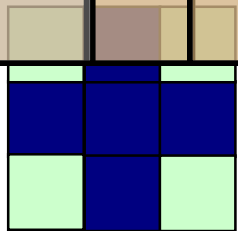
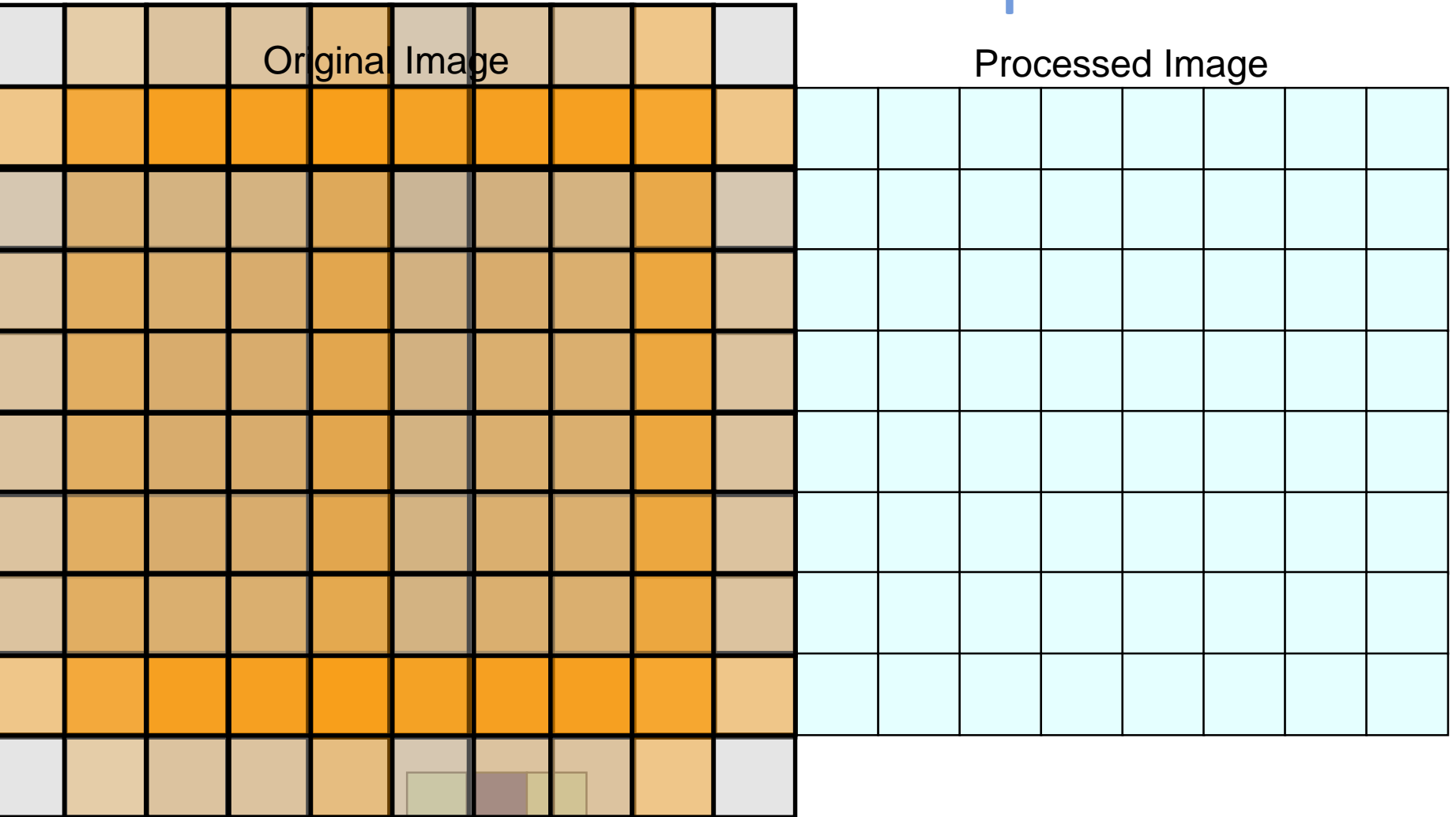
Dilation of image  $f$  by structuring element  $s$  is given by  $f \oplus s$

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

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



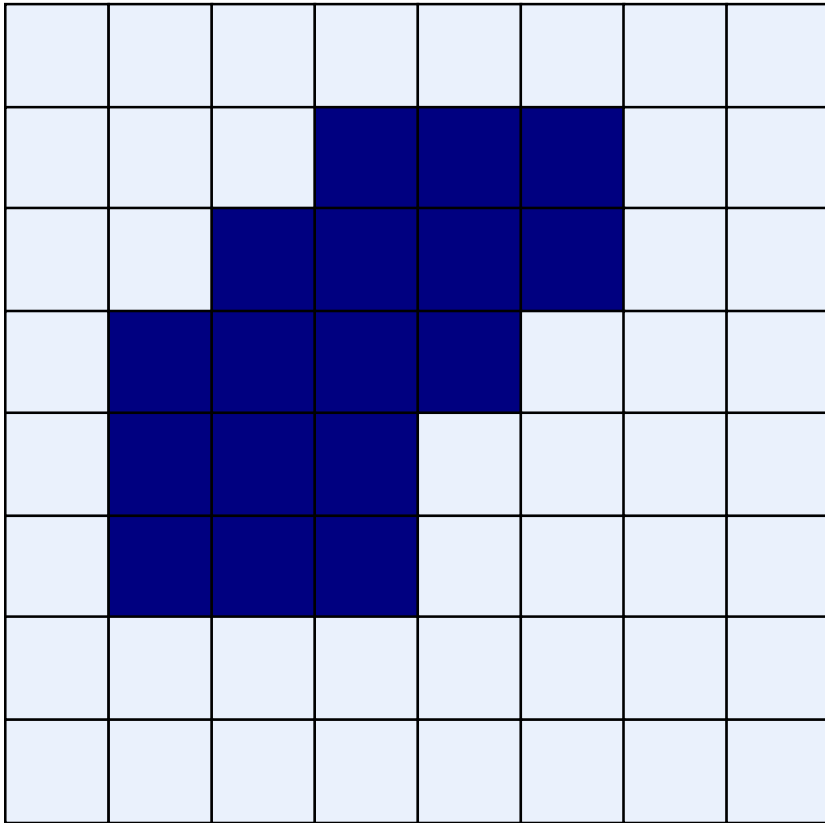
# Dilation Example



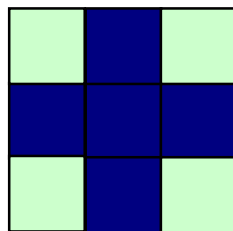
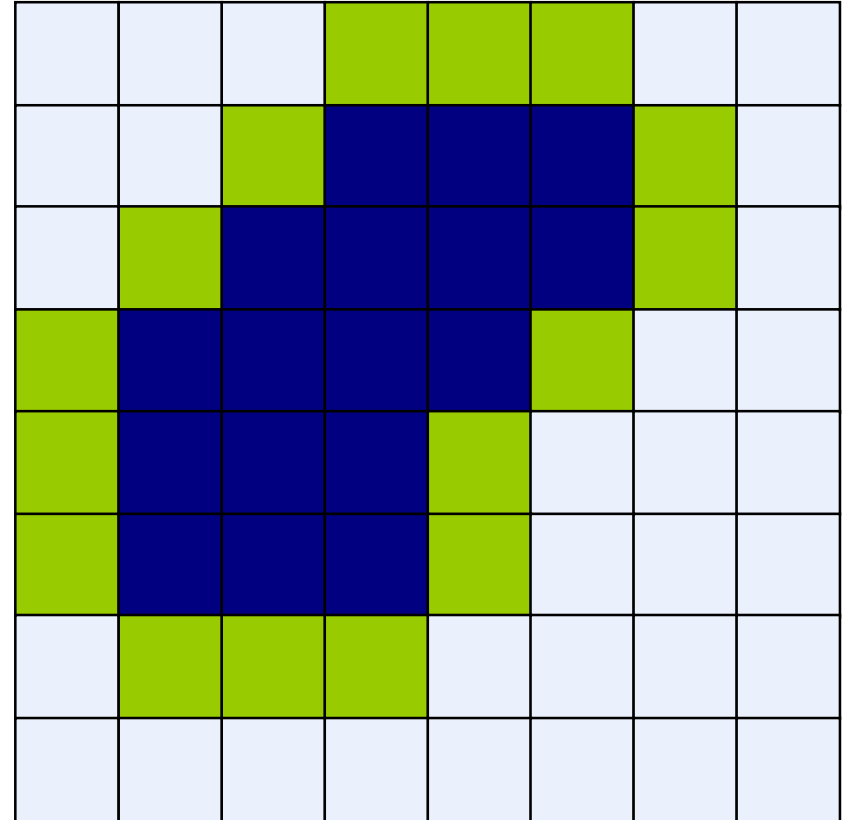
Structuring Element

# 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

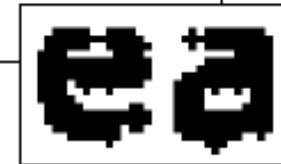
Original image

Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.



After dilation

**Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.**



0	1	0
1	1	1
0	1	0

Structuring element

# What Is Dilation For?

Dilation can repair breaks



Dilation can repair intrusions



**Watch out:** Dilation enlarges objects

# Compound Operations

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

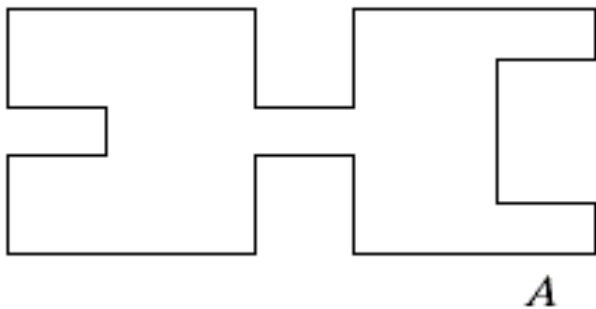
The most widely used of these *compound operations* are:

- Opening
- Closing

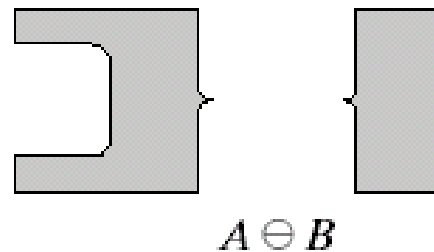
# Opening

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

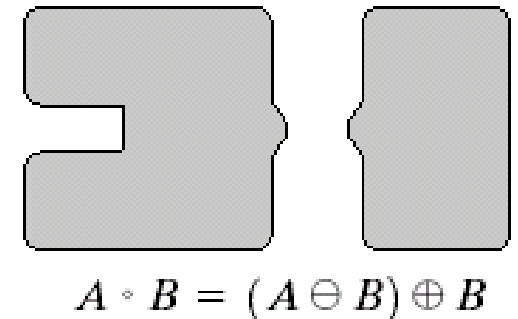
$$f \circ s = (f \ominus s) \oplus s$$



Original shape



After erosion



After dilation  
(opening)

Note a disc shaped structuring element is used

# Opening Example

Original  
Image

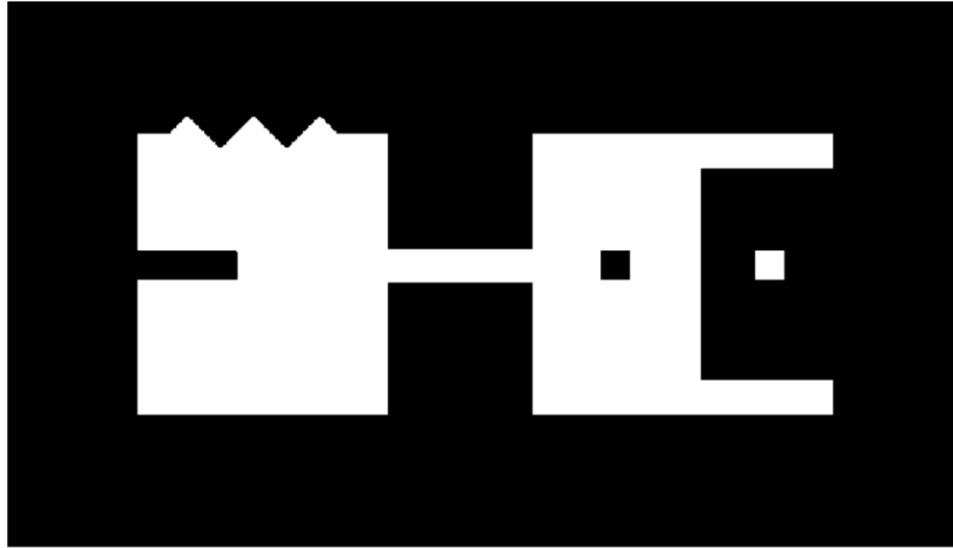
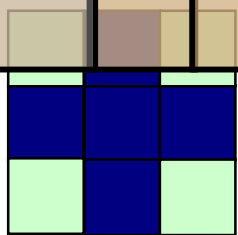
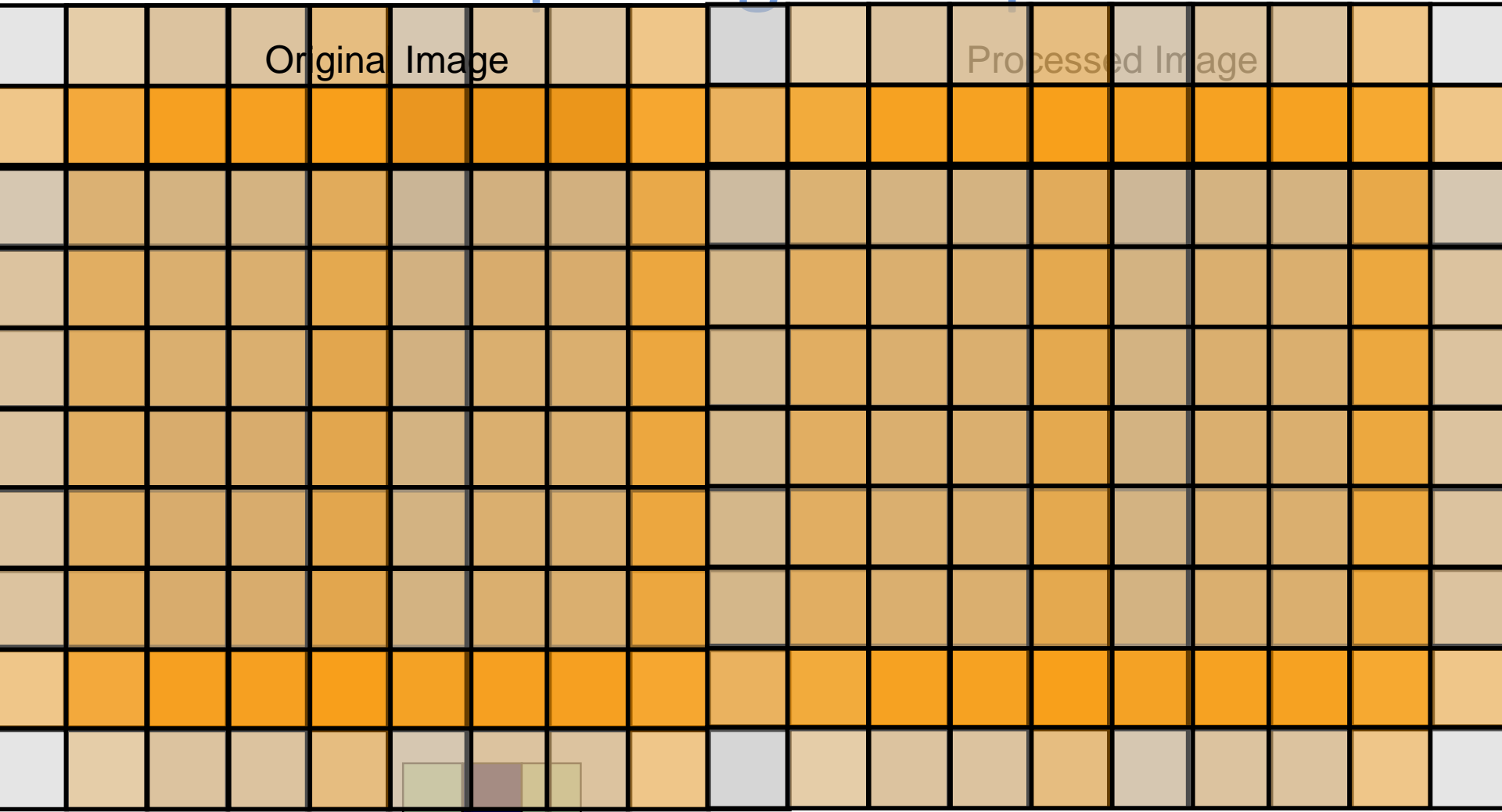


Image  
After  
Opening





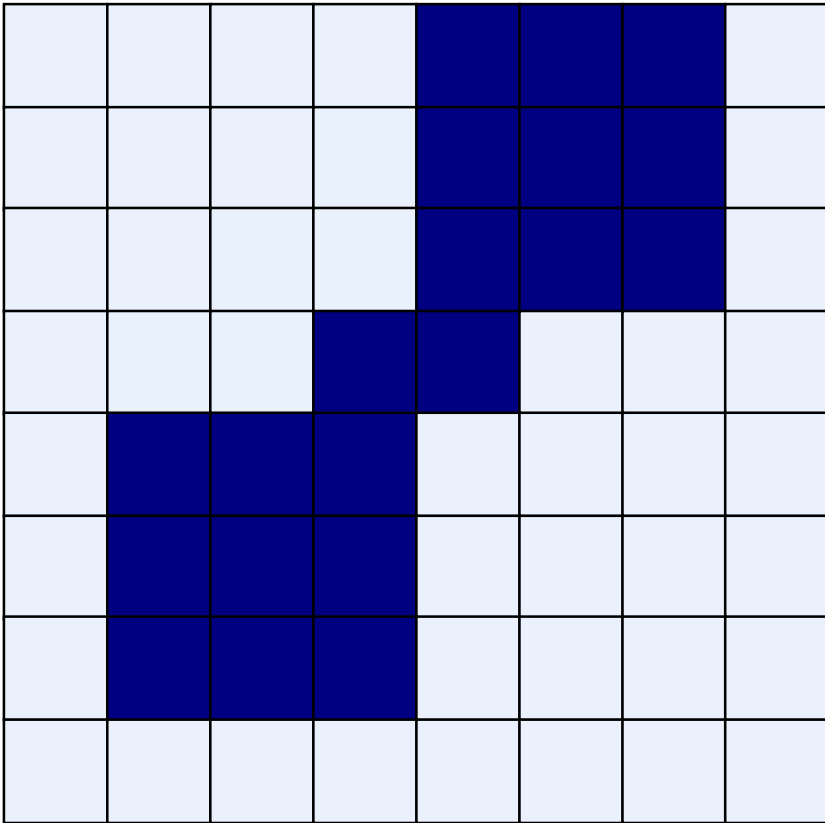
# Opening Example



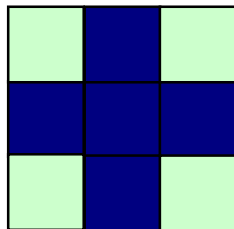
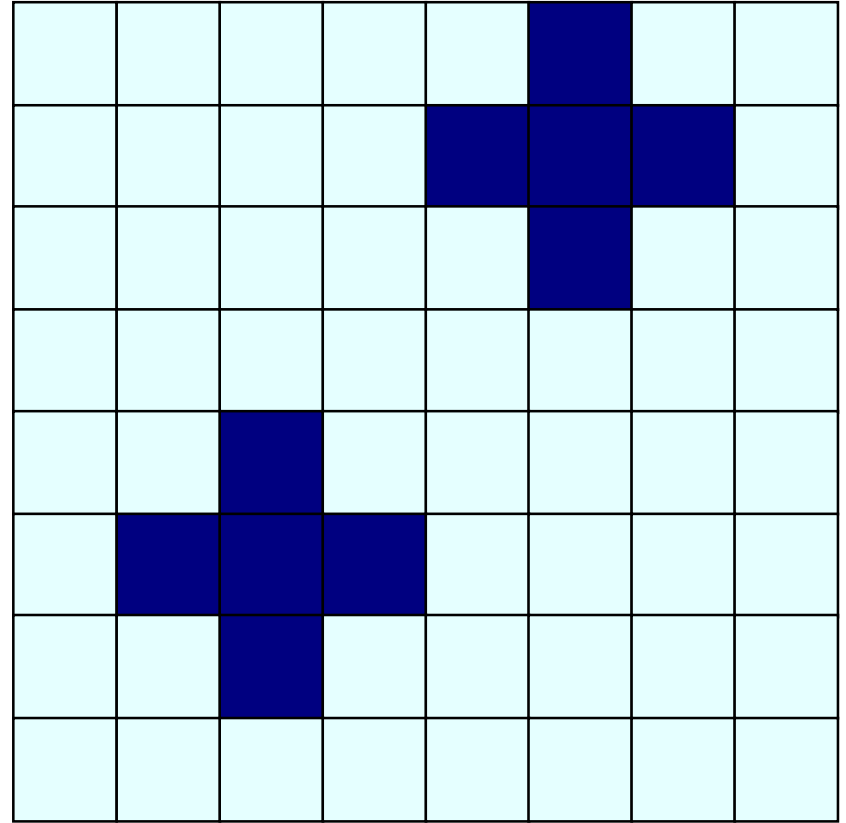
Structuring Element

# Opening Example

Original Image



Processed Image

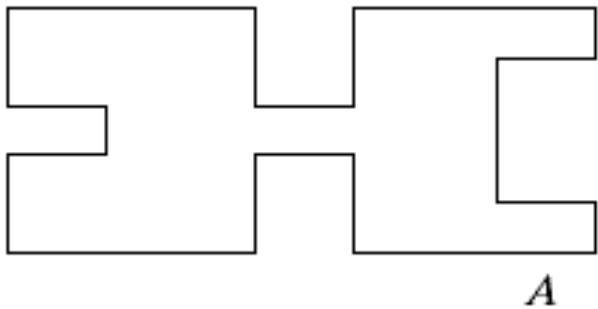


Structuring Element

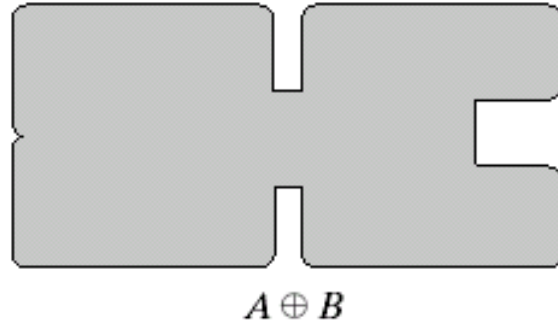
# Closing

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

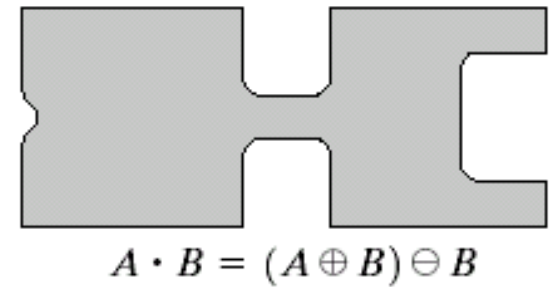
$$f \bullet s = (f \oplus s) \ominus s$$



Original shape



After dilation



After erosion  
(closing)

Note a disc shaped structuring element is used

# Closing Example

Original Image

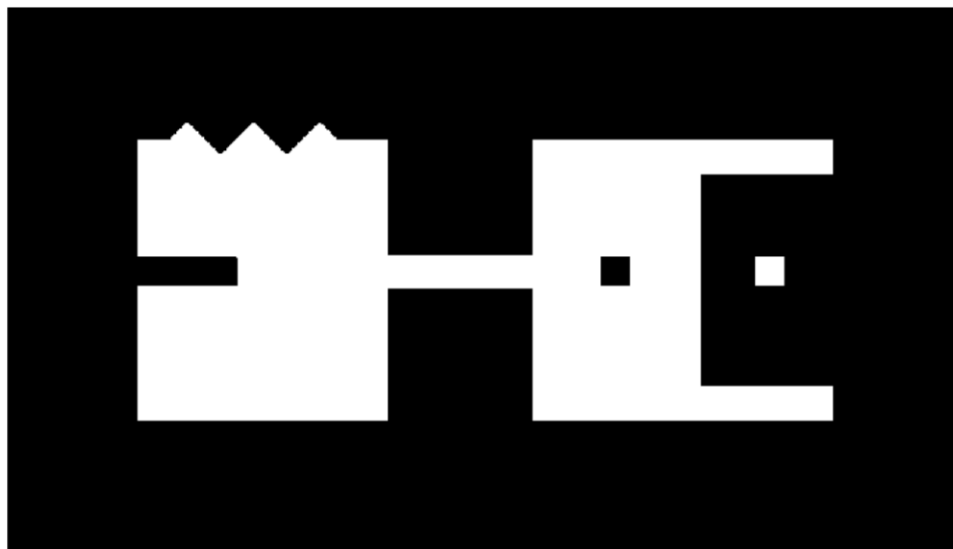
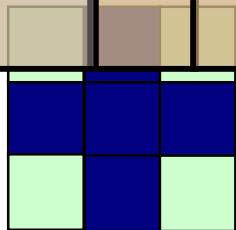
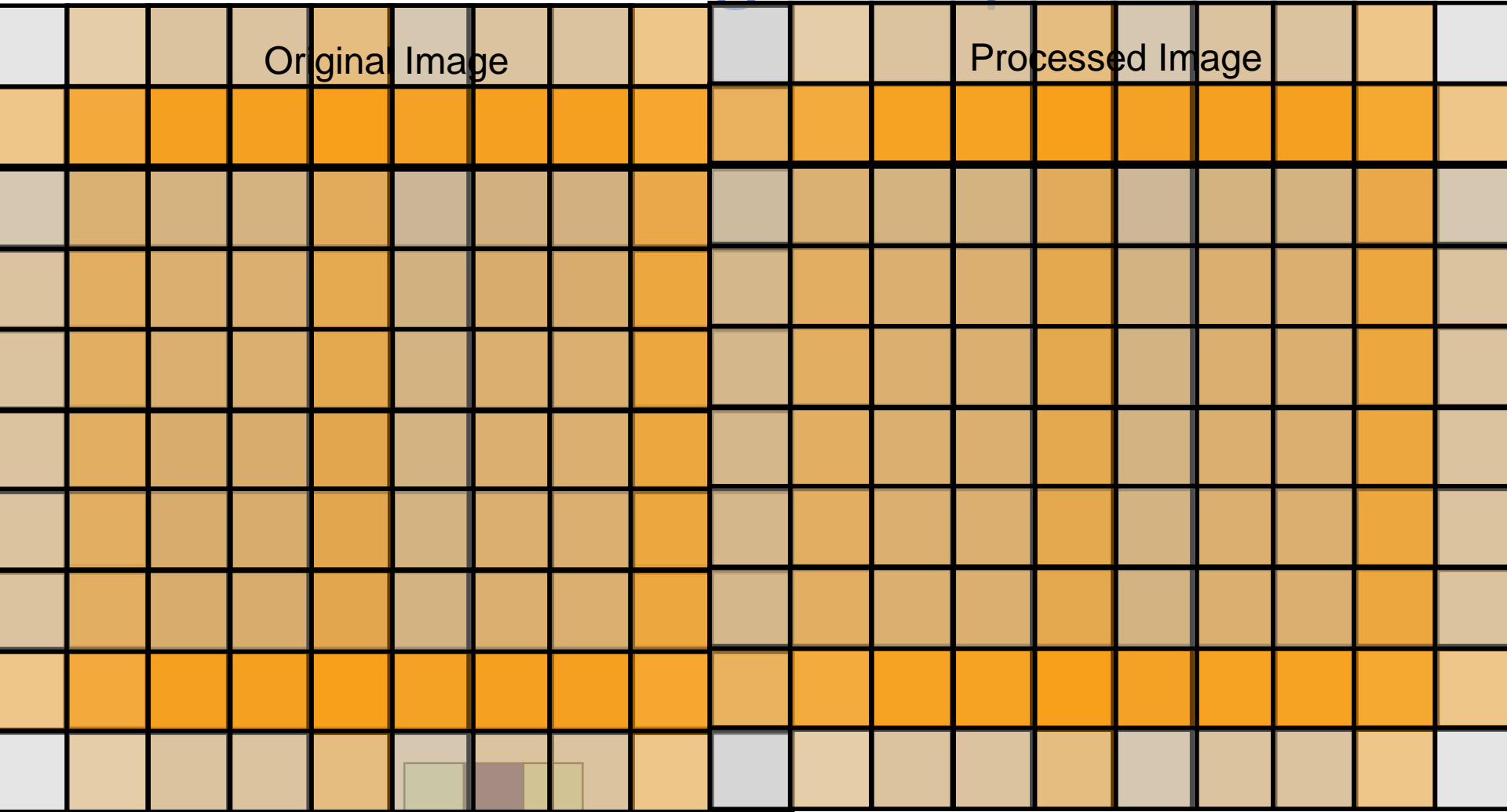


Image After Closing



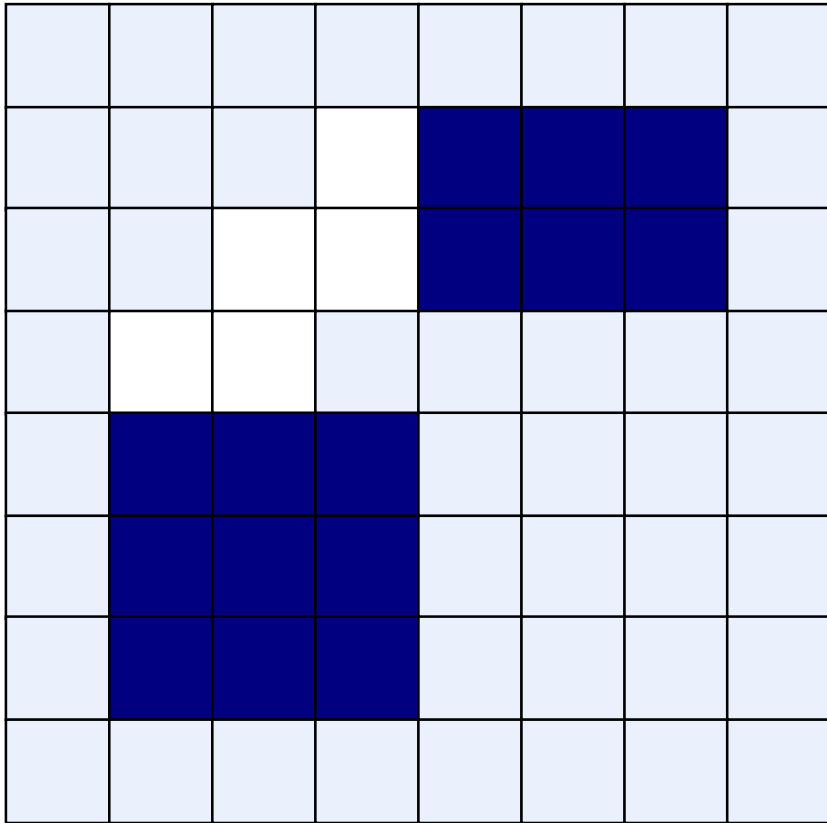
# Closing Example



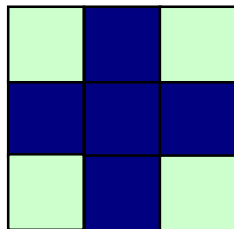
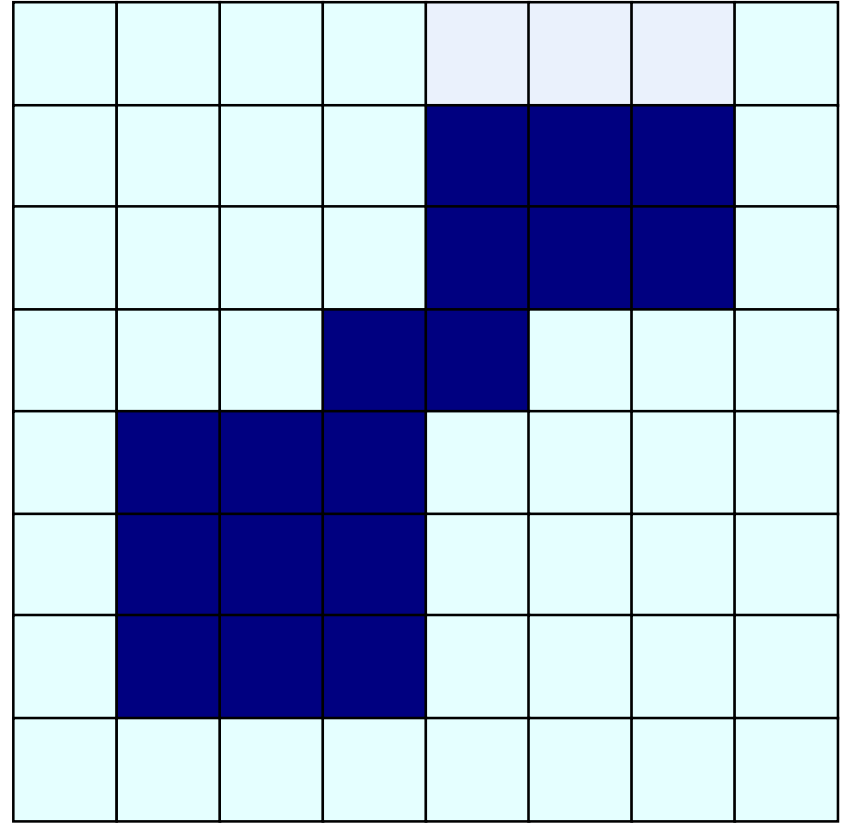
Structuring Element

# Closing Example

Original Image

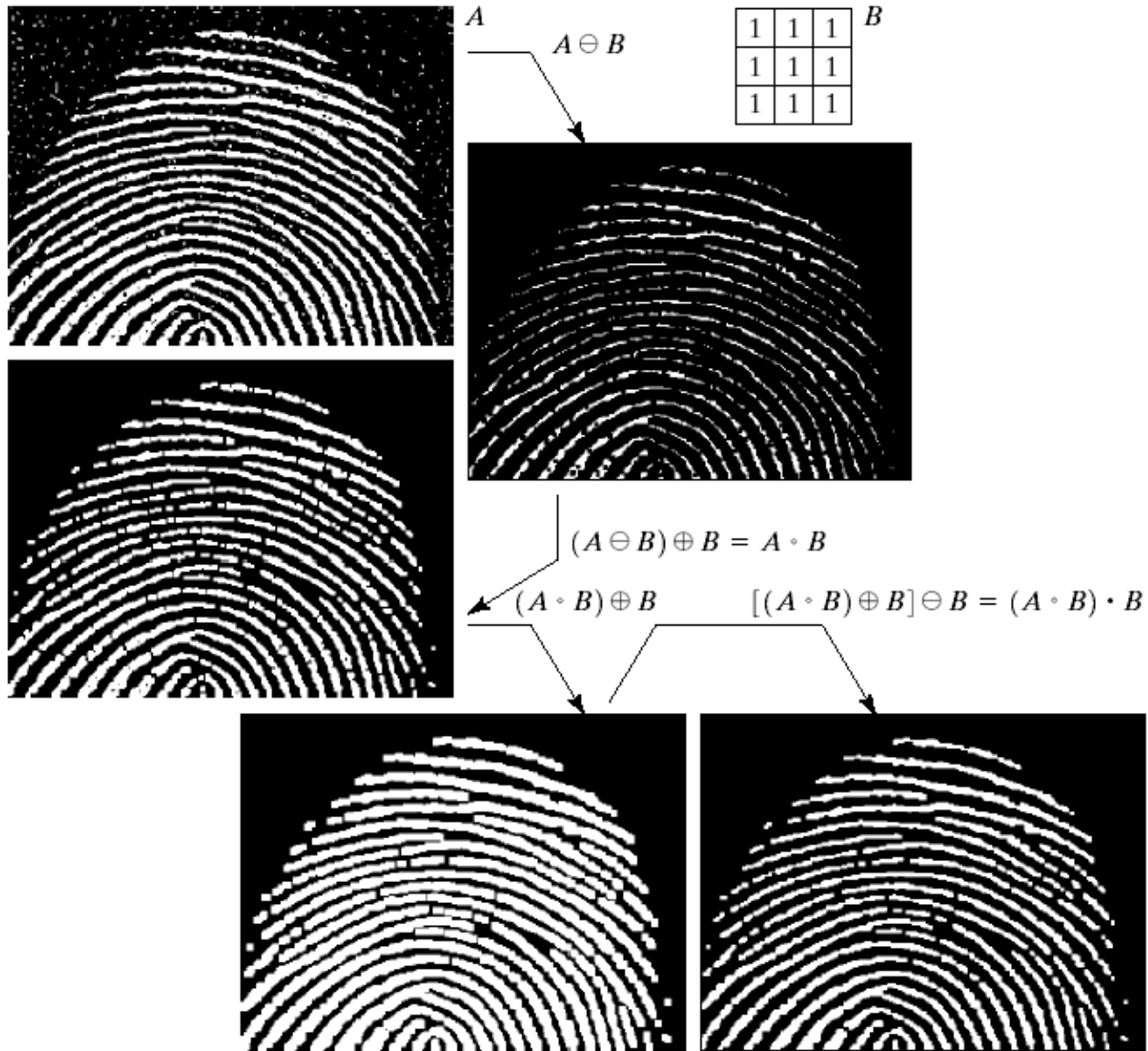


Processed Image



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
- > Skeletonisation

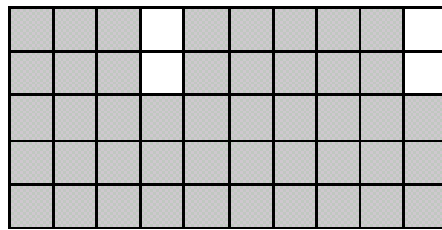


# Boundary Extraction

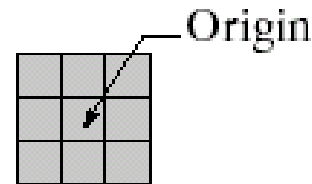
Extracting the boundary (or outline) of an object is often extremely useful

The boundary can be given simply as

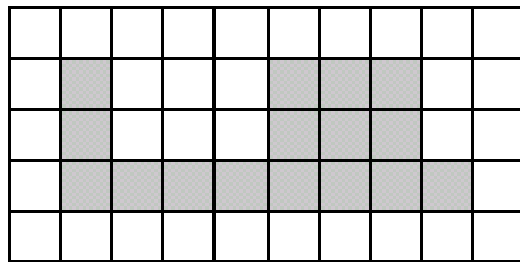
$$\beta(A) = A - (A \ominus B)$$



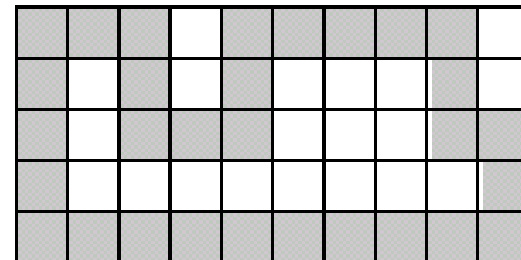
*A*



*B*



*A ⊖ B*



*β(A)*

# Boundary Extraction Example

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



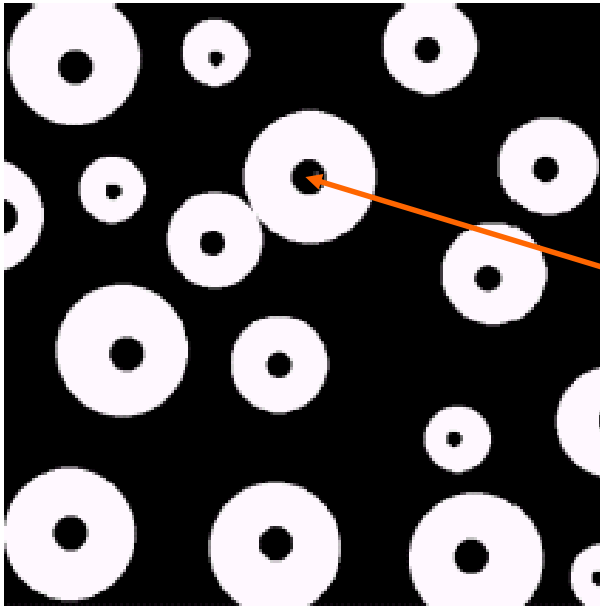
Original Image



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

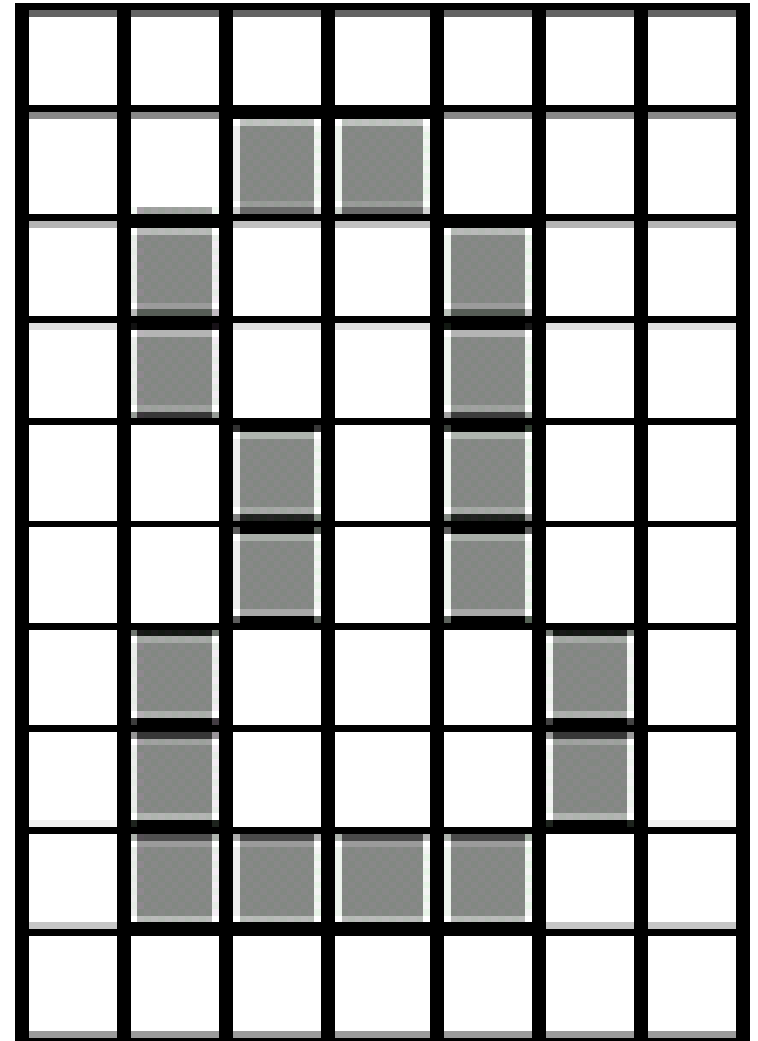
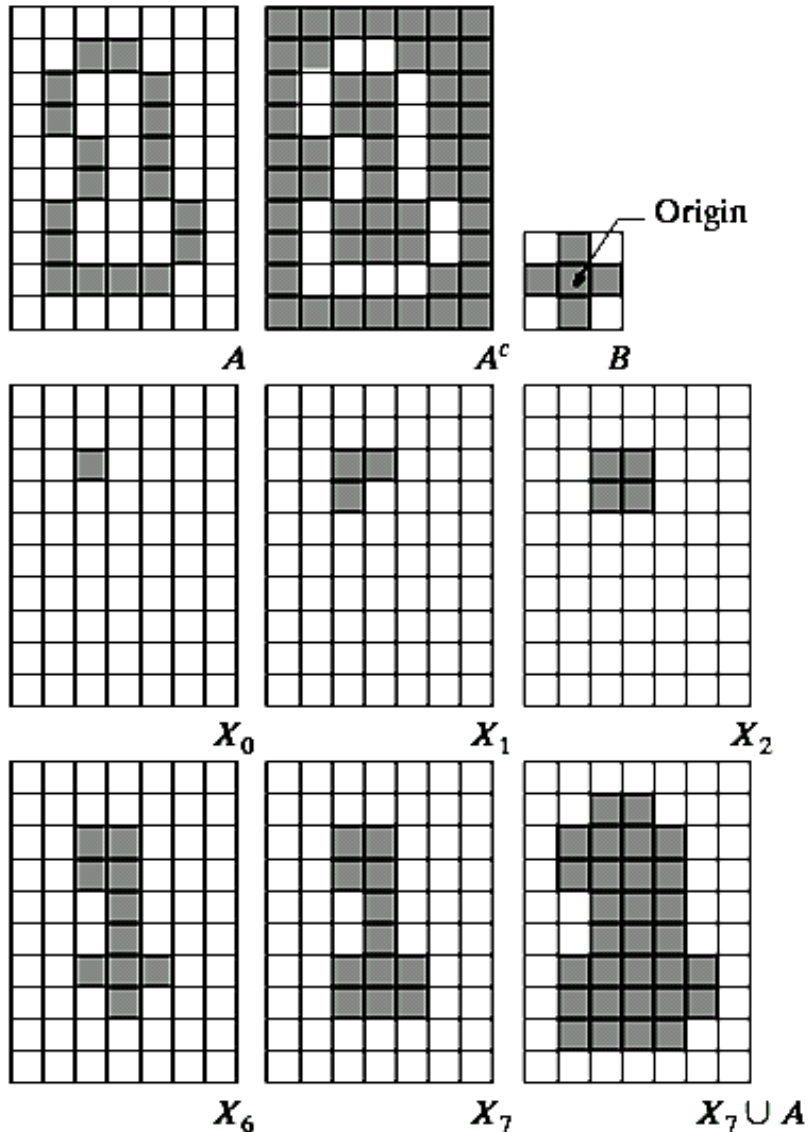
$$X_k = (X_{k-1} \oplus B) \cap A^c \quad k = 1, 2, 3, \dots$$

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

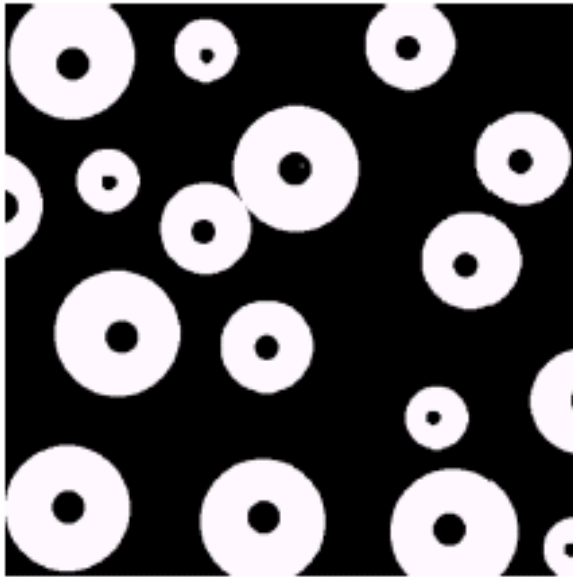
This equation is applied repeatedly until  $X_k$  is equal to  $X_{k-1}$

Finally the result is unioned with the original boundary

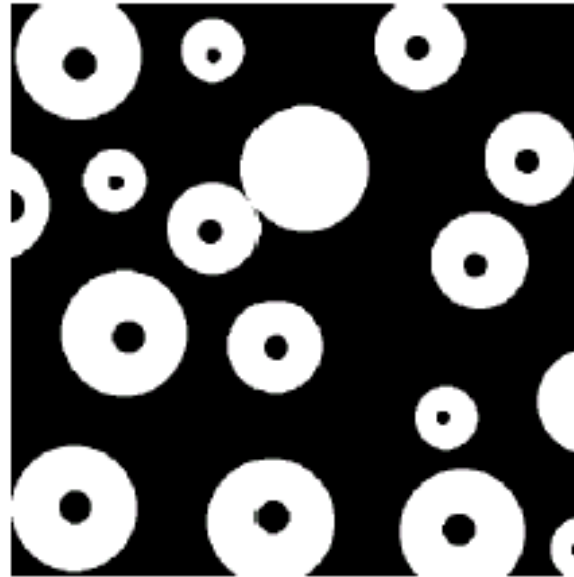
# Region Filling Step By Step



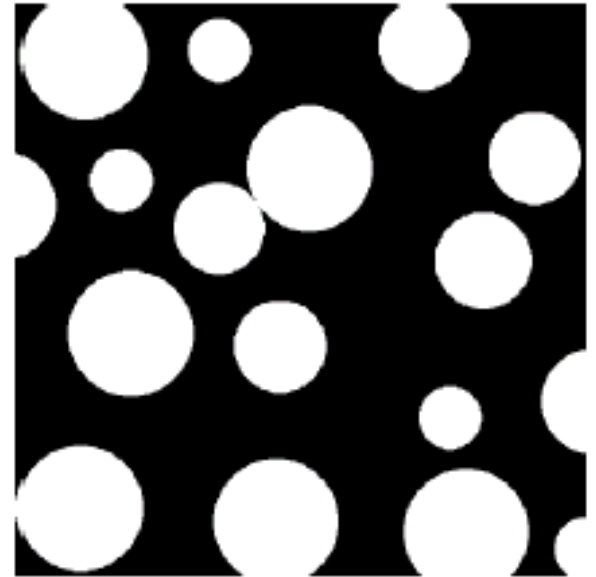
# Region Filling Example



Original Image



One Region Filled



All Regions Filled



# 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