Morphological Operations

Siyue Yu
siyue.yu02@xjtlu.edu.cn

Last lectures

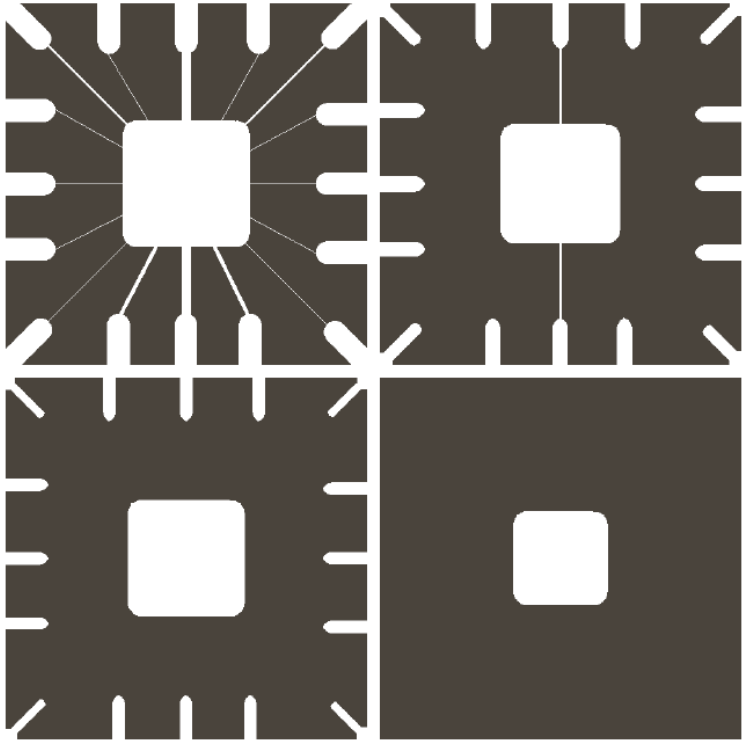
- Introduction to compression
  - Why to do compression
  - Data VS Information, Entropy
  - Compression ratio
  - Concept of lossless and lossy compression
  - Types of redundancy
- Coding Redundancy
  - Fixed-Length Code
  - Variable length codes
    - Shannon-Fano code
    - Huffman code
- Interpixel redundancy
  - Predictive coding
- Psychovisual redundancy
  - Transform coding
- JPEG

About this lecture

- In this lecture we introduce the most important operations based on morphological image processing, just to give you the intuitive feeling.

About this lecture

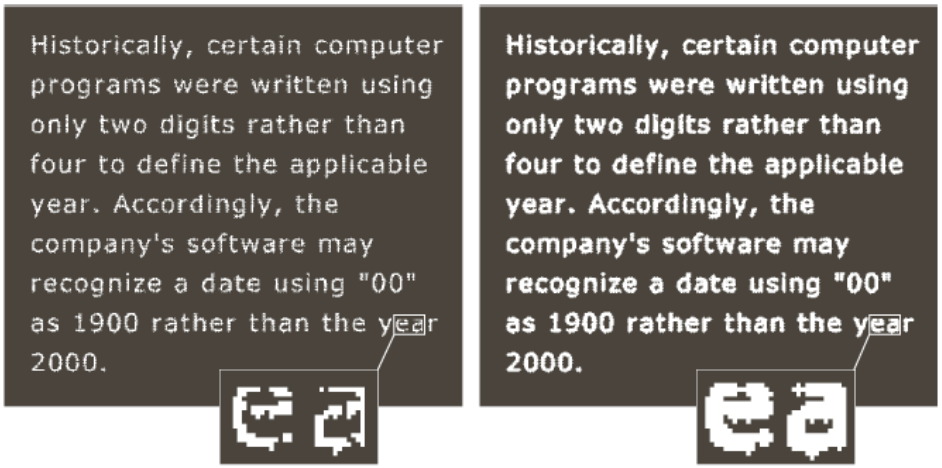
- How to get the right-bottom image from the left-top one?



Erosion

About this lecture

- How to get the right image from the left one?



0	1	0
1	1	1
0	1	0

Dilation

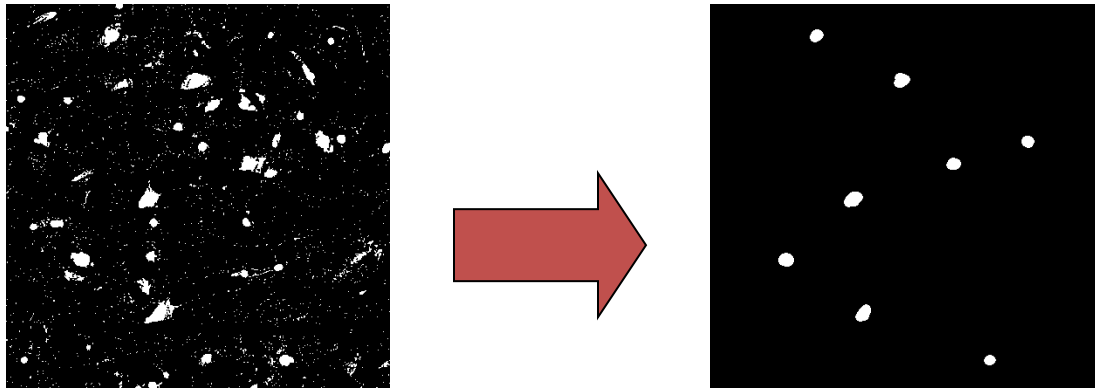
About this lecture

- How to get the right image from the left one?

Boundary Extraction

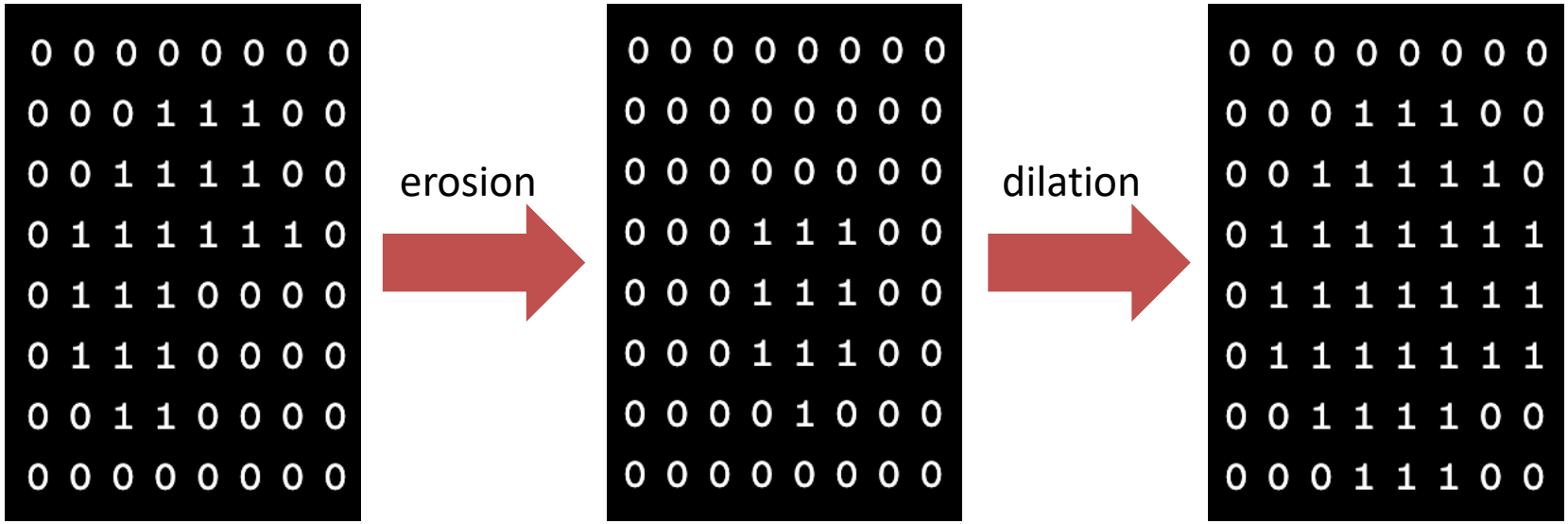
About this lecture

- How to get the right image from the left one?



Opening

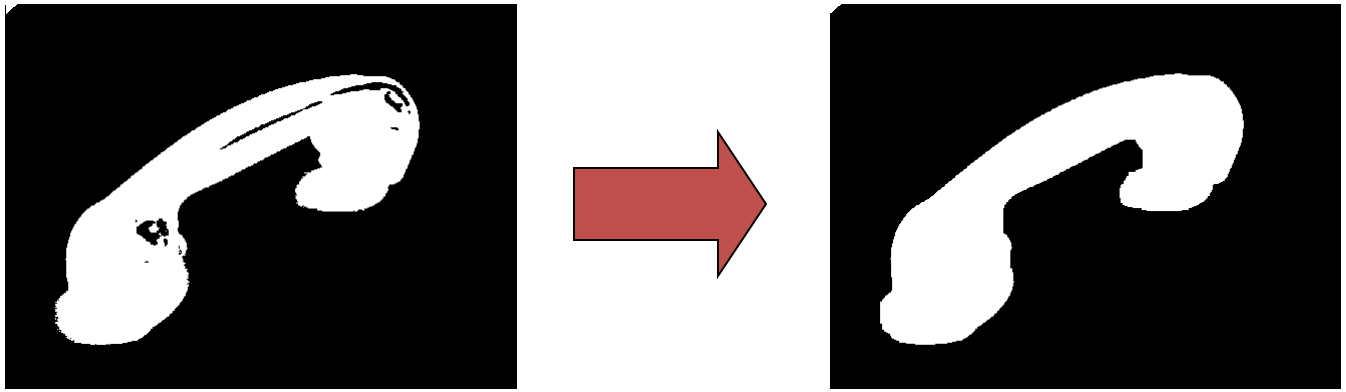
Opening - example



1. Suppose we have a binary image of a circle with some noise
2. Applying an erosion operation with a small kernel can remove the noise
3. However, this erosion operation has also caused the edges of the circle to become more jagged. We can smooth out these edges by applying a dilation operation with a larger kernel
4. Now, the circle has been extracted more smoothly and any small noise or artifacts have been removed.

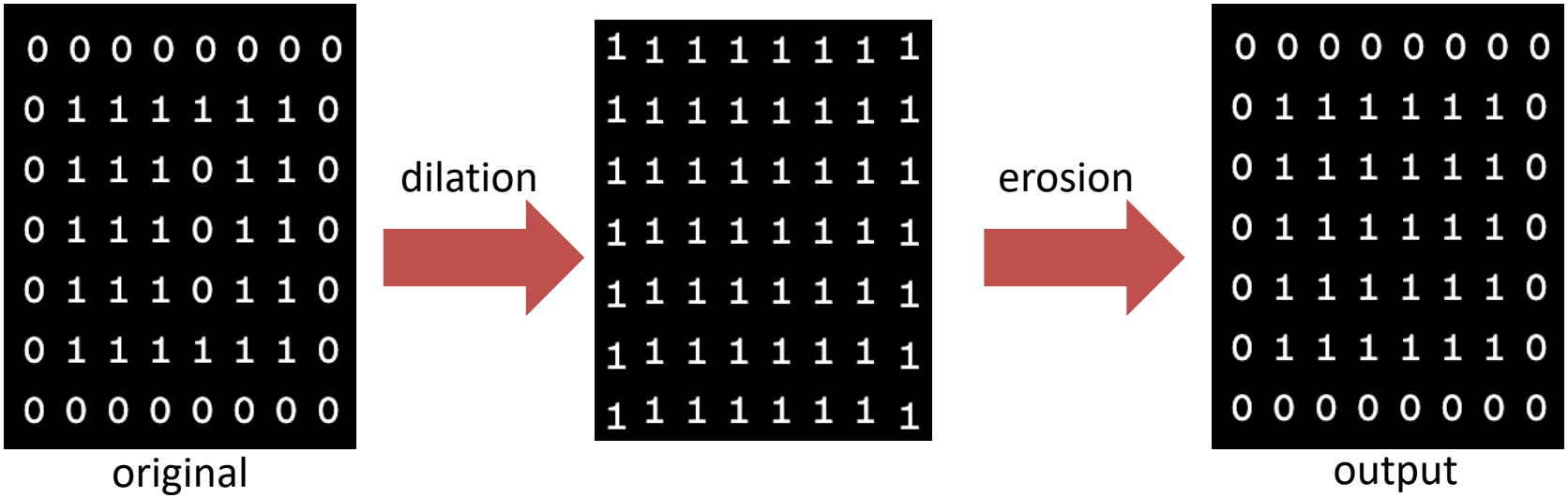
About this lecture

- How to get the right image from the left one?



Closing

Closing - example



original output

1. Suppose we have a binary image of a rectangle with a gap in the middle:
2. Applying a dilation operation with a small kernel can fill in the gap:
3. However, this dilation operation has also caused the edges of the rectangle to become more blurred. We can sharpen these edges by applying an erosion operation with a larger kernel:
4. Now, the rectangle has been closed and any gaps or holes have been filled in.

About this lecture

- How to detect the character I and T?

Hit-and-miss

Binary images

- A binary image (bi-level) is a digital image that has only two possible values for each pixel.
  - Typically black (‘0’) for background (BG) color and white (‘1’) for foreground (FG).
  - The color used for the object(s) in the image is the FG color while the rest of the image is the BG color.

Binary images

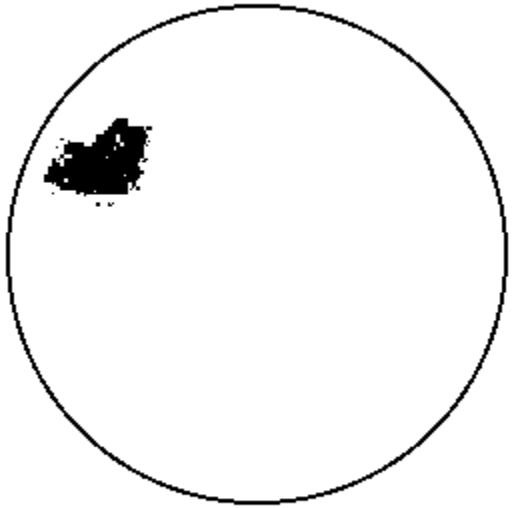
- Binary images often arise in digital image processing as masks or as the result of certain operations such as segmentation, thresholding, and detection.
- Some input/output devices, such as laser printers, fax machines, and bilevel computer displays, can only handle bilevel images.

Binary mask for bleeding detection in WCE

- Bleeding detection in the WCE (Wireless Capsule Endoscopy) images

Binary mask for bleeding detection in WCE

- Thresholding to detect “bleeding color”



  - RGB → HSI (color space) → blood detection → threshold and masking (please note FG is black here)

Binary mask for bleeding detection in WCE

- Bleeding detection in the WCE images

Introductory example about Morphological Operations

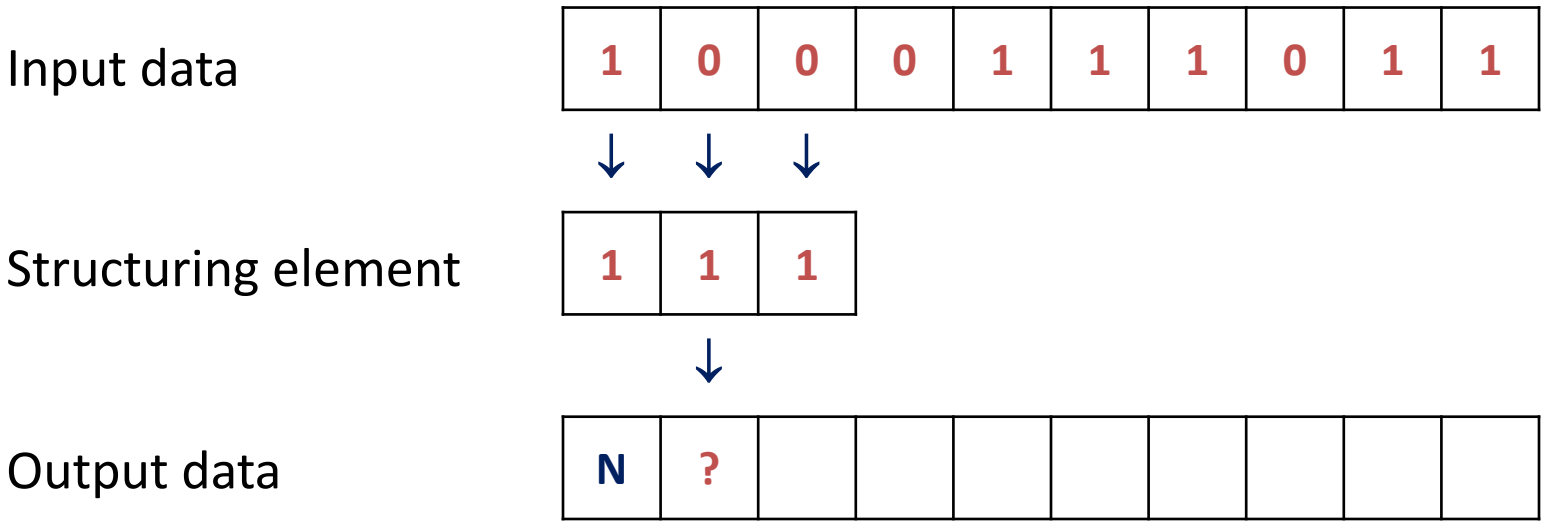
Example of morphological operation

Input data

1	0	0	0	1	1	1	0	1	1

Output data

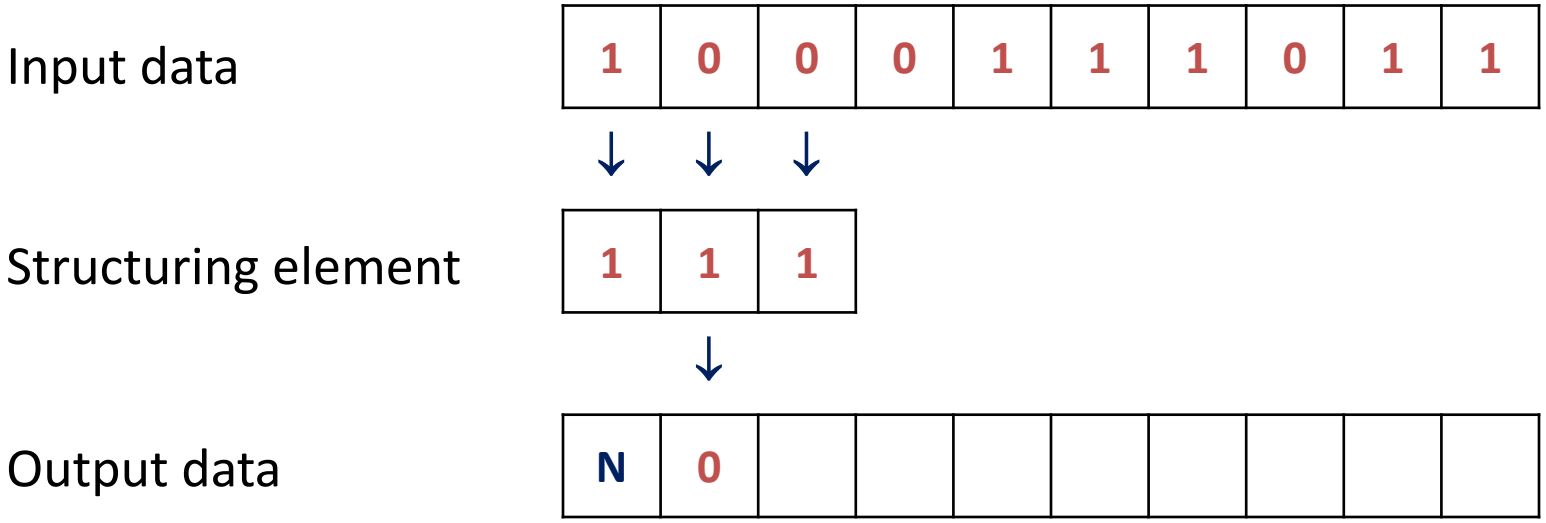
Example of morphological operation



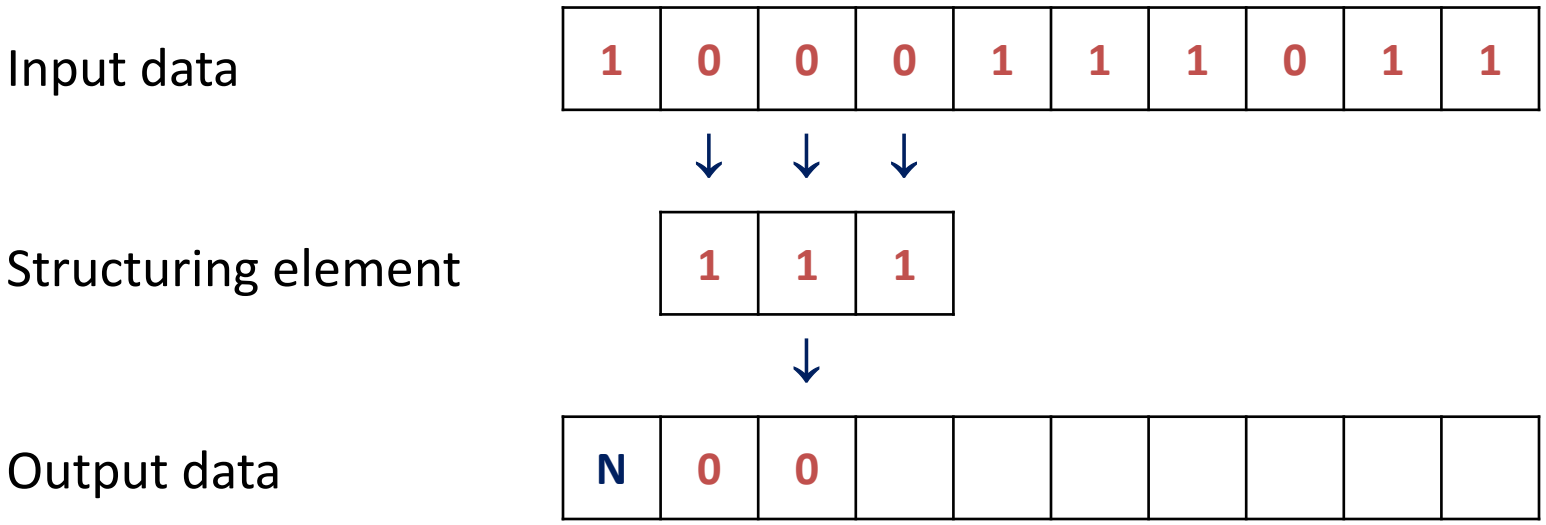
– The structuring element (SE) is like a window that we process through it the input signal

Example of morphological operation

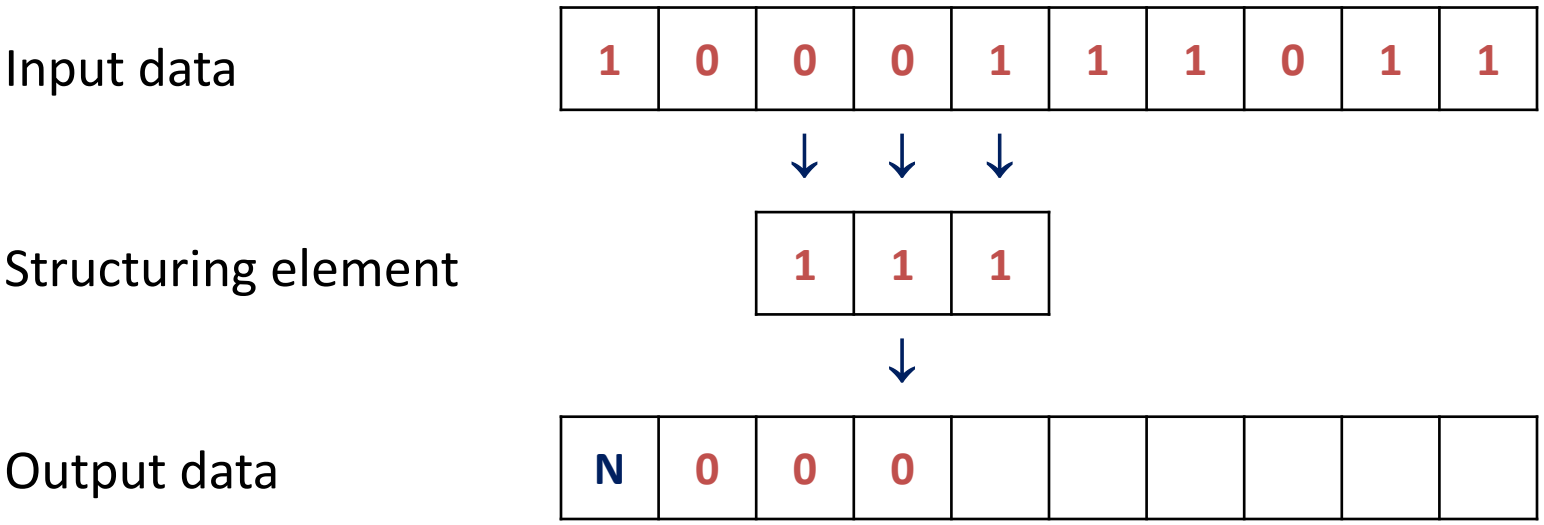
– If any of the input pixel's neighborhood is ‘0’ the output pixel is set to ‘0’.



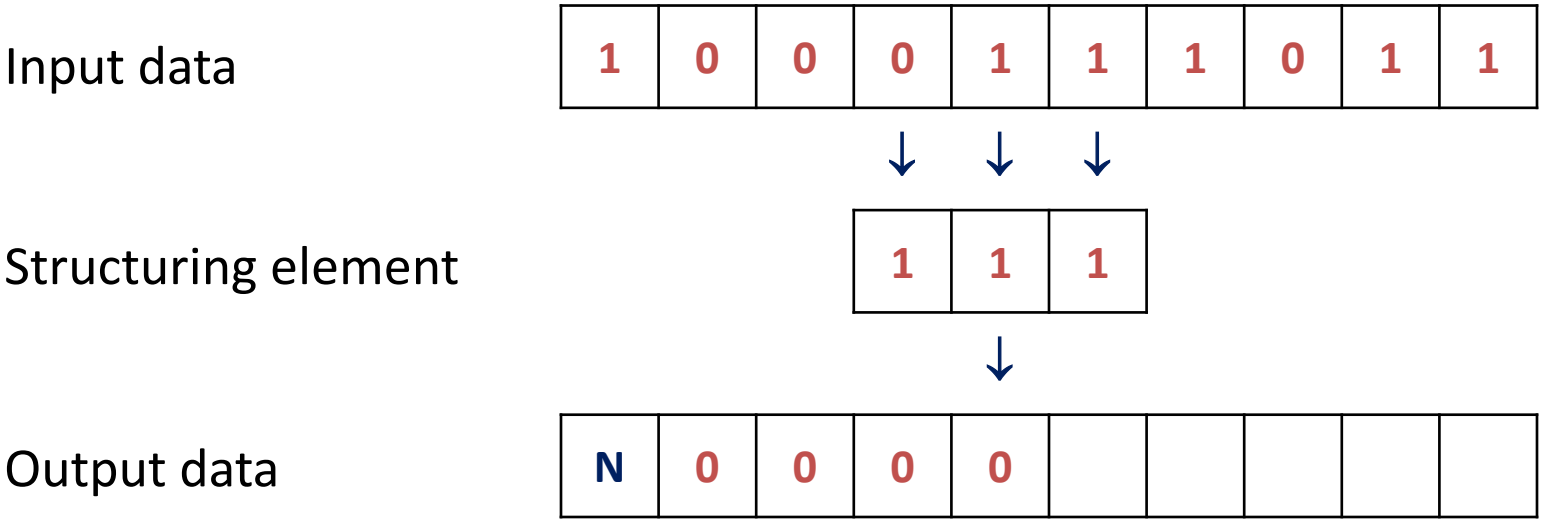
Example of morphological operation



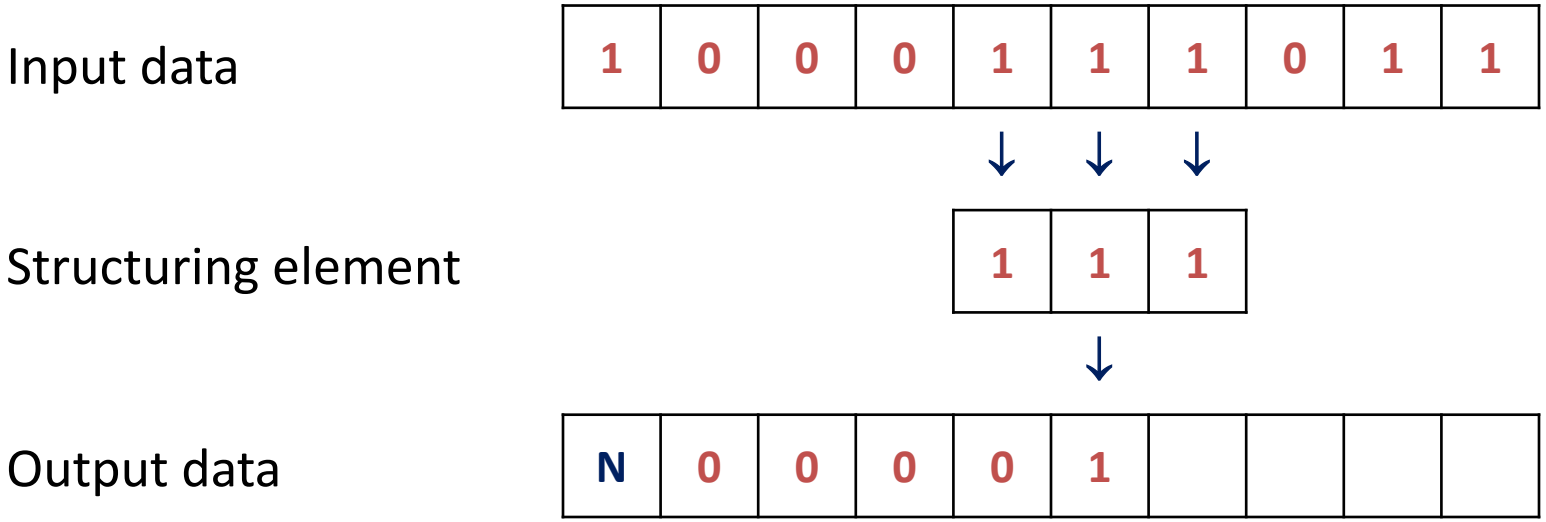
Example of morphological operation



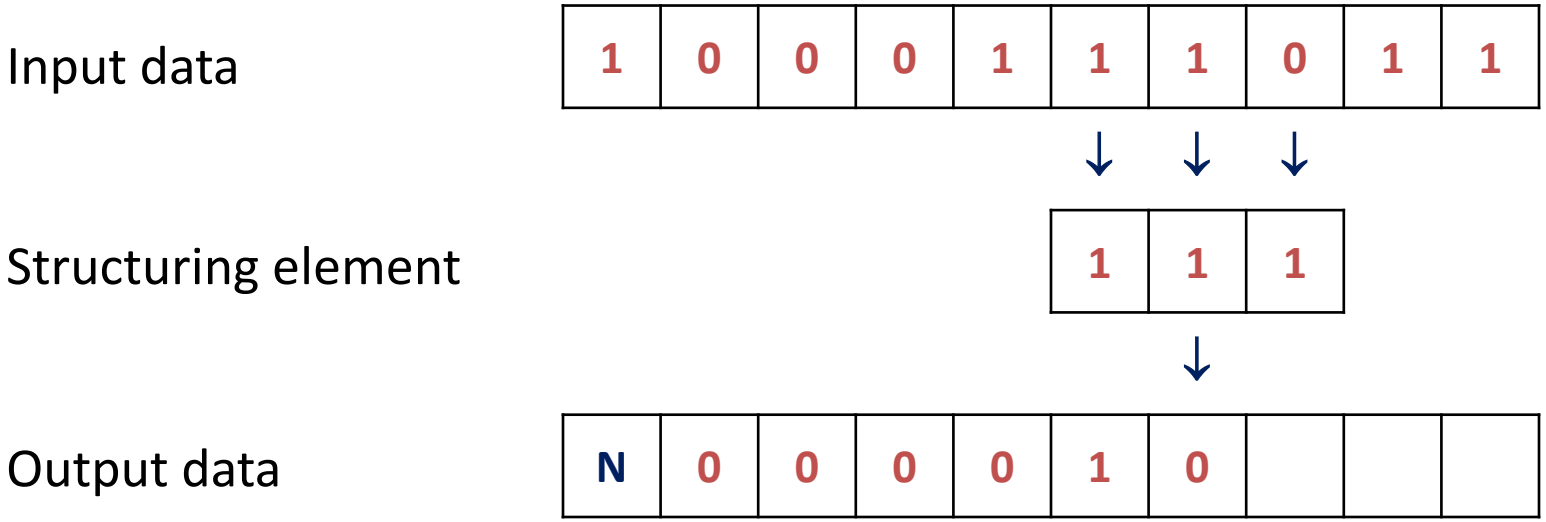
Example of morphological operation



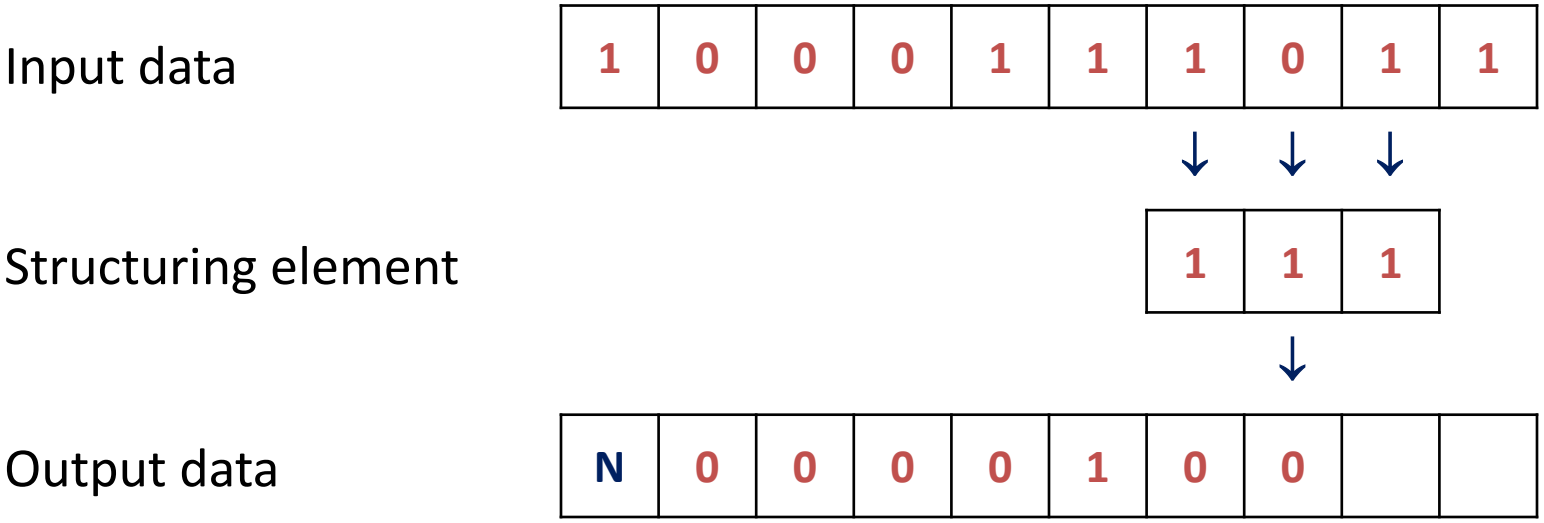
Example of morphological operation



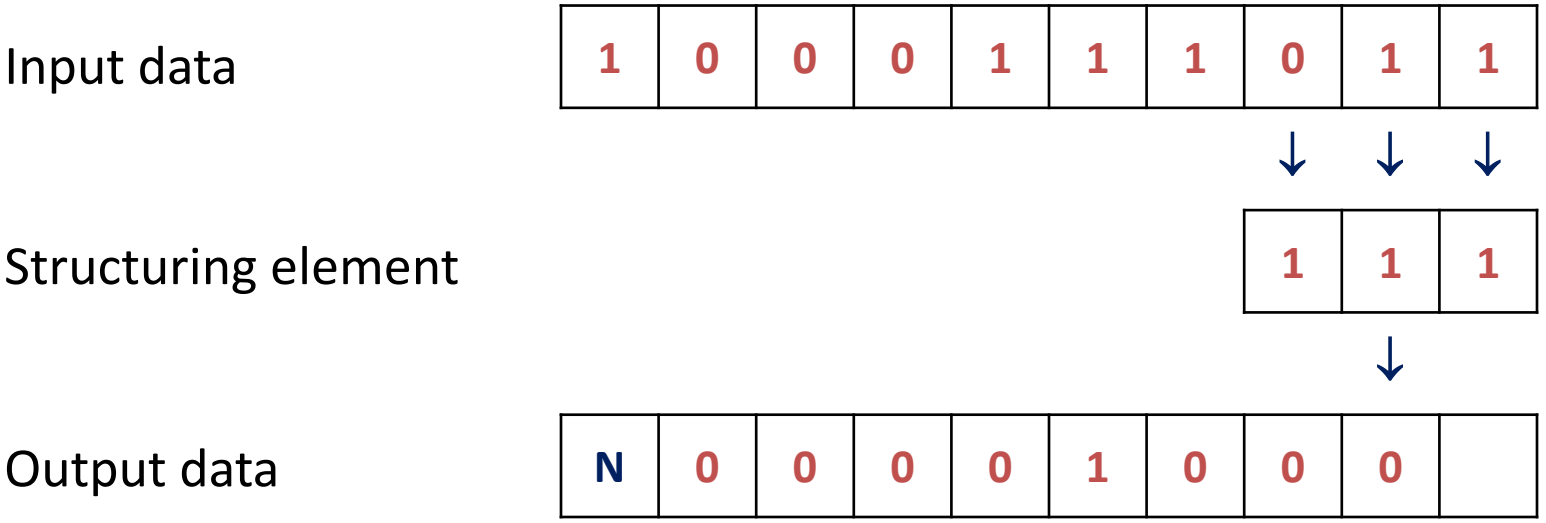
Example of morphological operation



Example of morphological operation



Example of morphological operation



Example of morphological operation

Input data

1	0	0	0	1	1	1	0	1	1

Structuring element

Output data

N	0	0	0	0	1	0	0	0	N

Example of morphological operation

- Do you have a proper name to describe this morphological operation ?

Input data

1	0	0	0	1	1	1	0	1	1

Structuring element

Output data

N	0	0	0	0	1	0	0	0	N

Morphological Operators

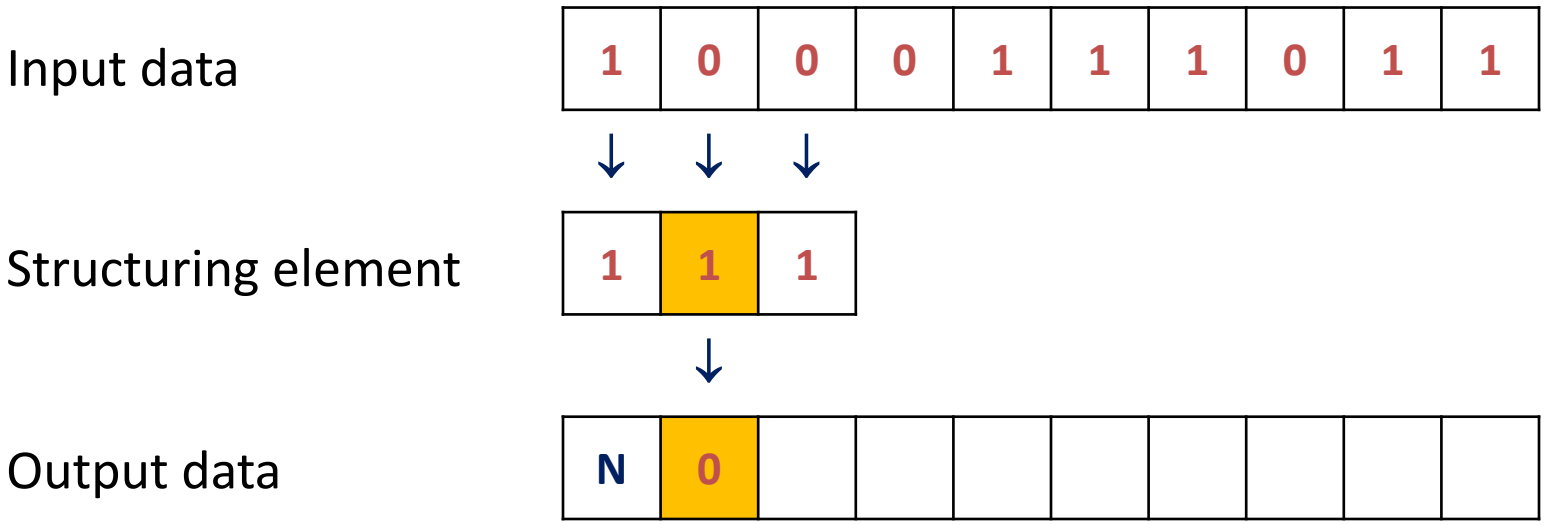
- In a morphological operation, the value of each pixel in the output image is based on the value of the input pixels in the neighborhood defined by the SE.
- By choosing the size and shape of the structuring element, it is possible to construct a morphological operation that is sensitive to specific shapes in the input image.

The structuring elements

- The structuring element (kernel) is a matrix that defines the neighborhood of interest of the pixel:
  - '1' values define the neighborhood (of interest).
  - '0' values in SE is equivalent to don't use the corresponding input pixel!
  - Structural Elements have an origin, which identifies the pixel being processed.

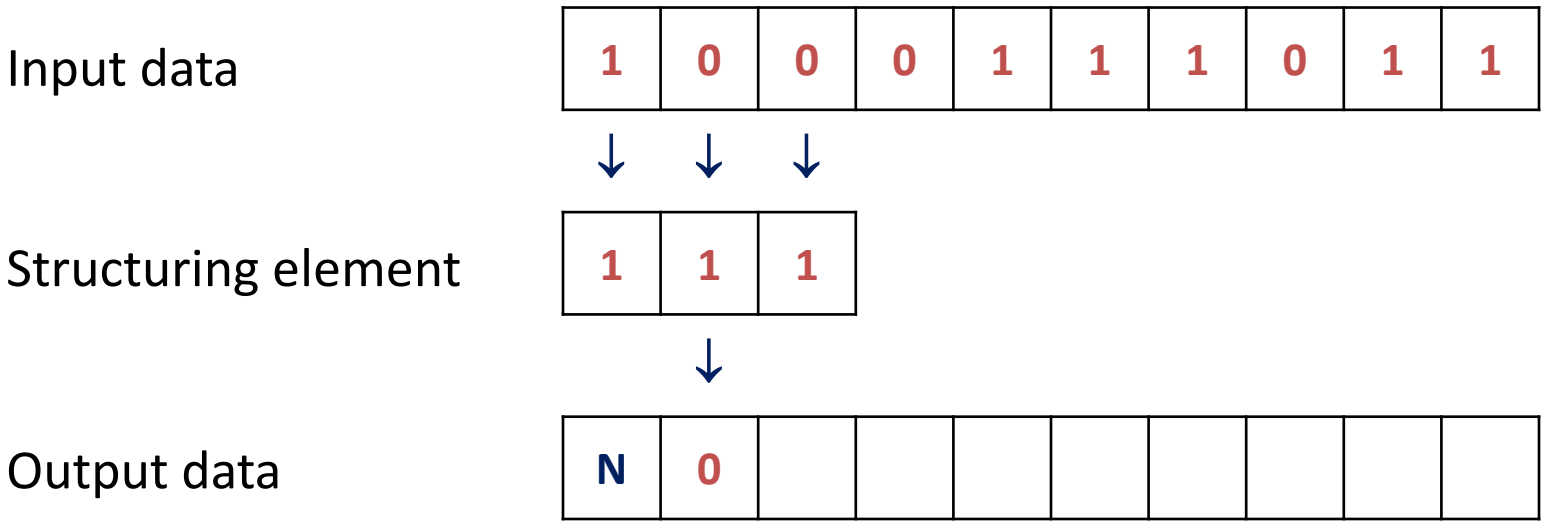
The structuring elements

– The origin of the SE



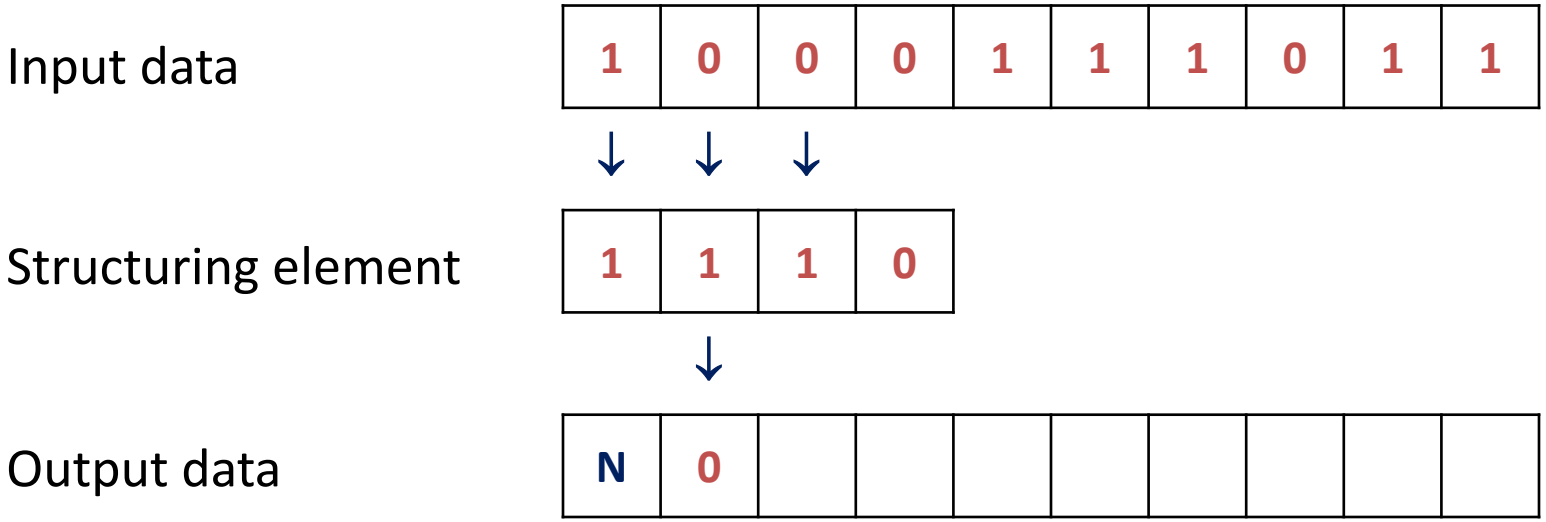
The structuring elements

– ‘1’ values define the neighborhood (of interest).



The structuring elements

– ‘0’ in SE is equivalent to don’t use the corresponding input pixel

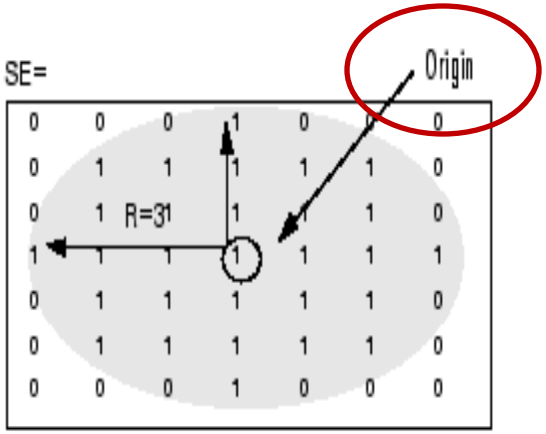


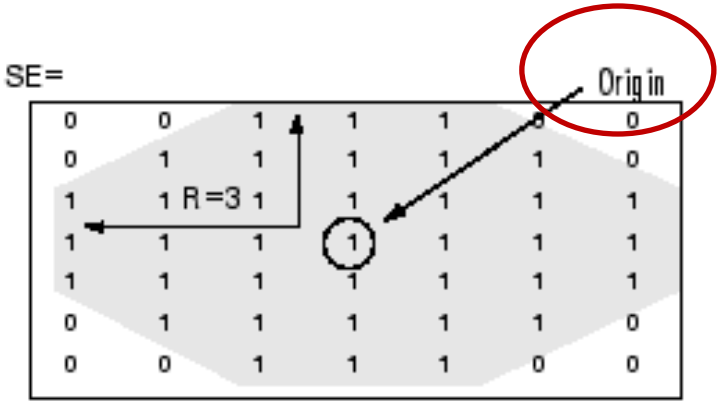
The structuring elements

- Structuring Elements can have any arbitrary shape and size, they are typically much smaller than the image being processed.
- Typically the structuring element has the same size and shape as the objects we want to process in the input image.
  - For example, to find lines in an image we could use a linear structuring element.

The structuring elements

- In Matlab the command :
  - SE = strel(shape,parameters)

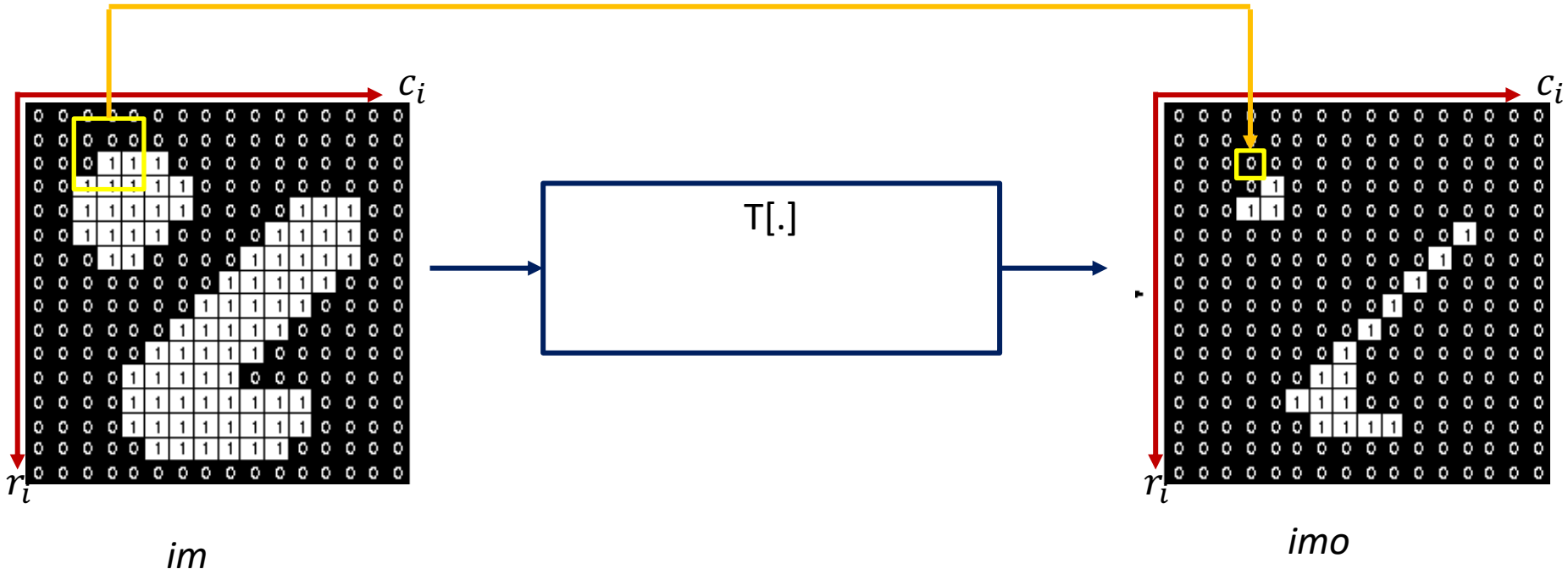





```
SE1 = strel('disk',3);
SE2=strel('octagon',3);
```

Morphological Operators

- Block-to-pixel mapping
  - Operates on a **set of 'neighbourhoods'** of each processed pixel



Erosion

What is Erosion ?

- Erosion is an important morphological operation
  - Erosion removes pixels on object boundaries.
  - The number of removed pixels from the objects depends on the size and shape of the structuring element used to process the image.

What is Erosion ?

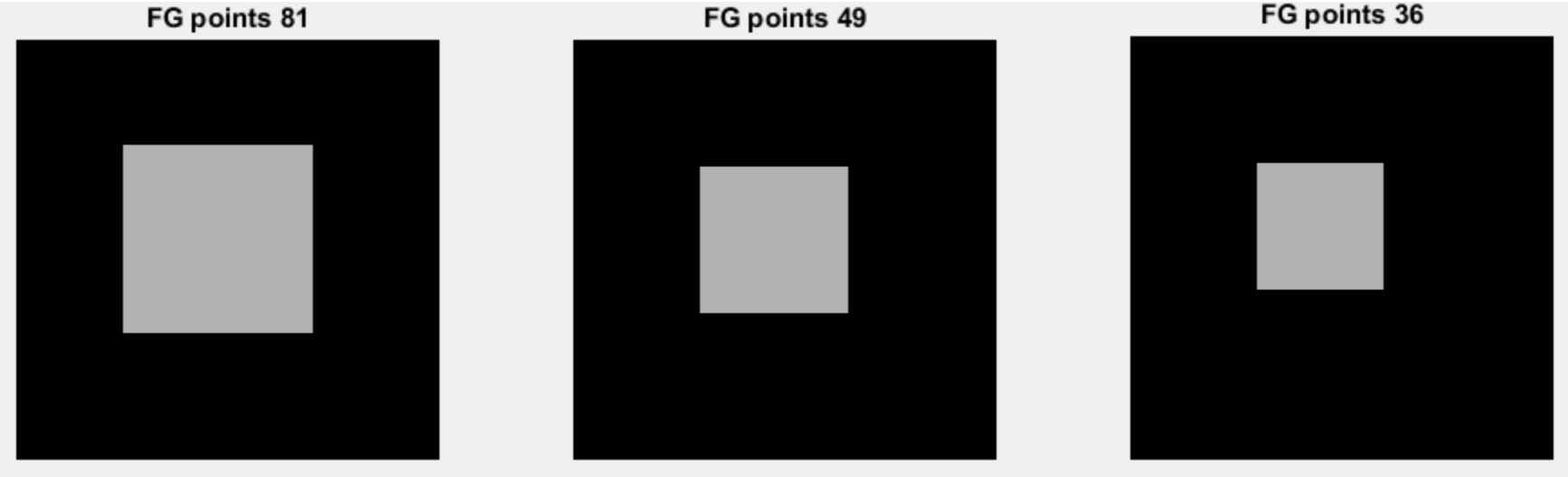
- Erosion is the set of all points in the image, where the structuring element “fits into”.
- If the structuring element fits in the foreground object, write “1” at the origin of the structuring element in the output image!

Where Erosion is used for ?

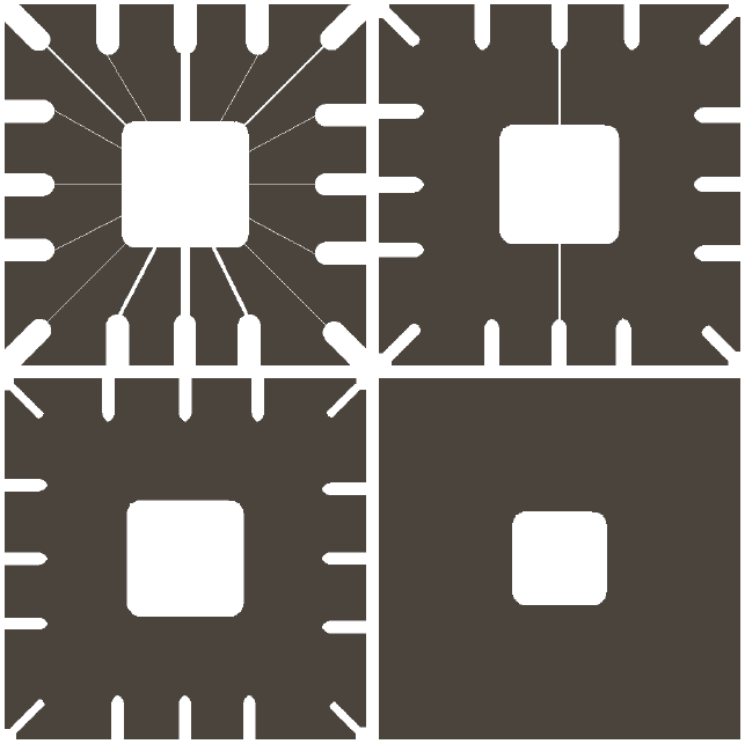
- Used to:
  - Make boundary of objects smooth
  - Remove sparse small objects,
  - In simple application of pattern matching

One example of Erosion

- See the different size of SE



Another example of Erosion



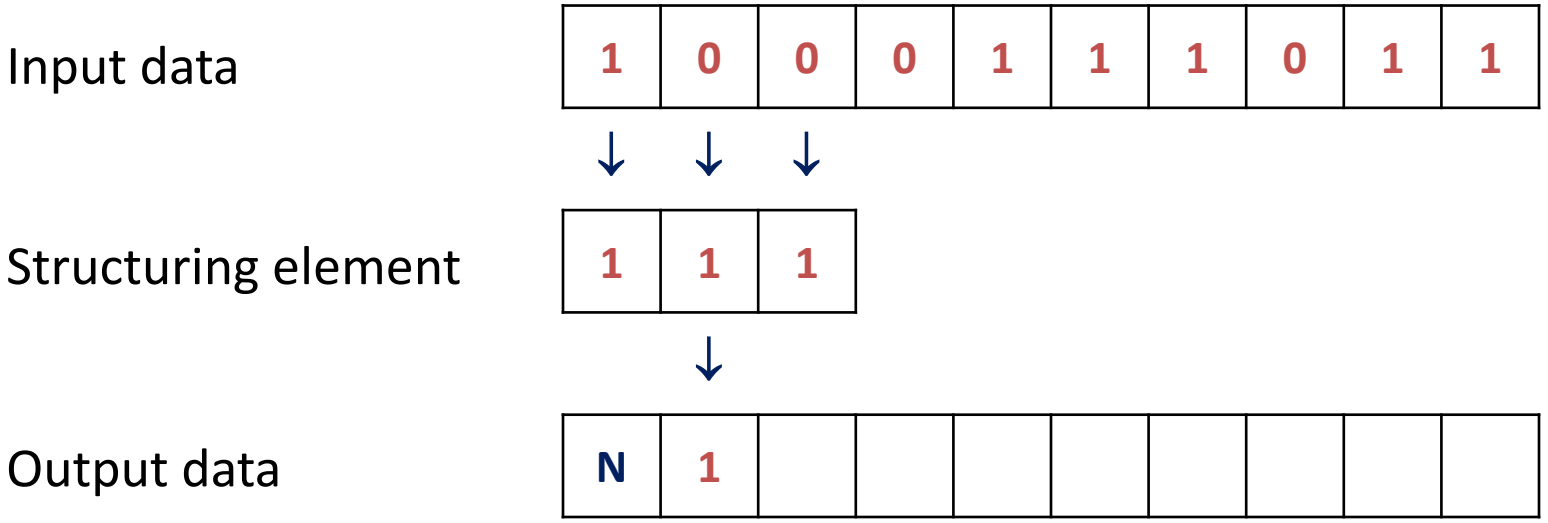
a b
c d

FIGURE 9.5 Using erosion to remove image components. (a) A 486 × 486 binary image of a wire-bond mask. (b)–(d) Image eroded using square structuring elements of sizes 11 × 11, 15 × 15, and 45 × 45, respectively. The elements of the SEs were all 1s.

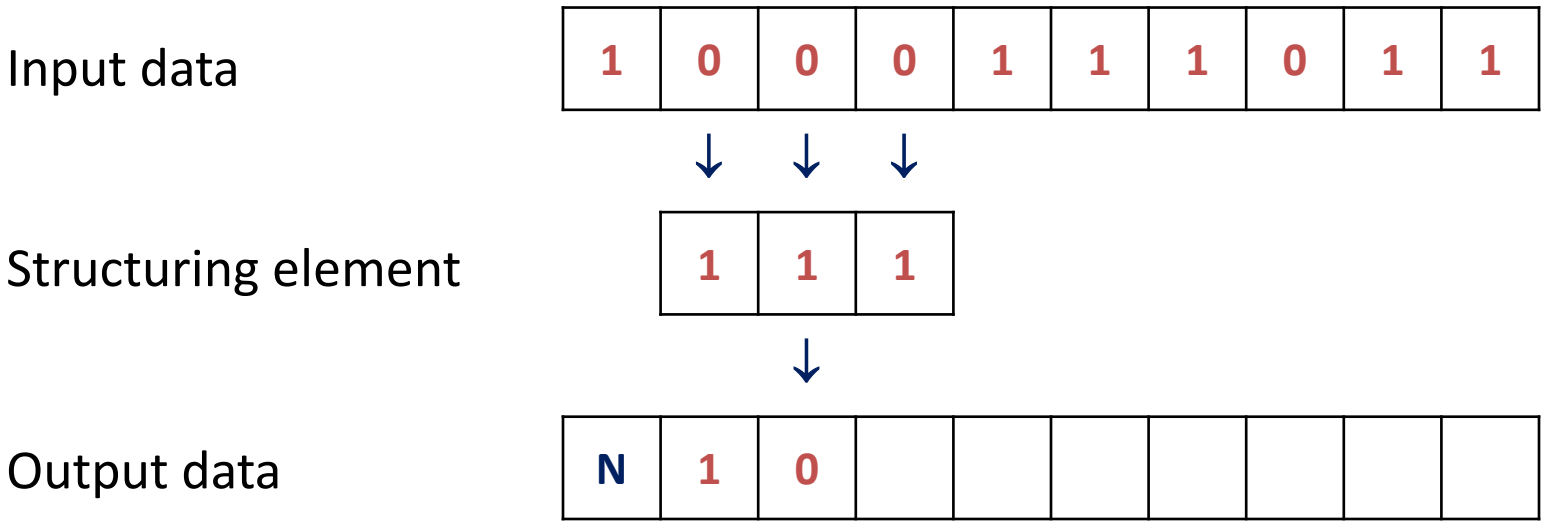
Dilation

Example of Dilation

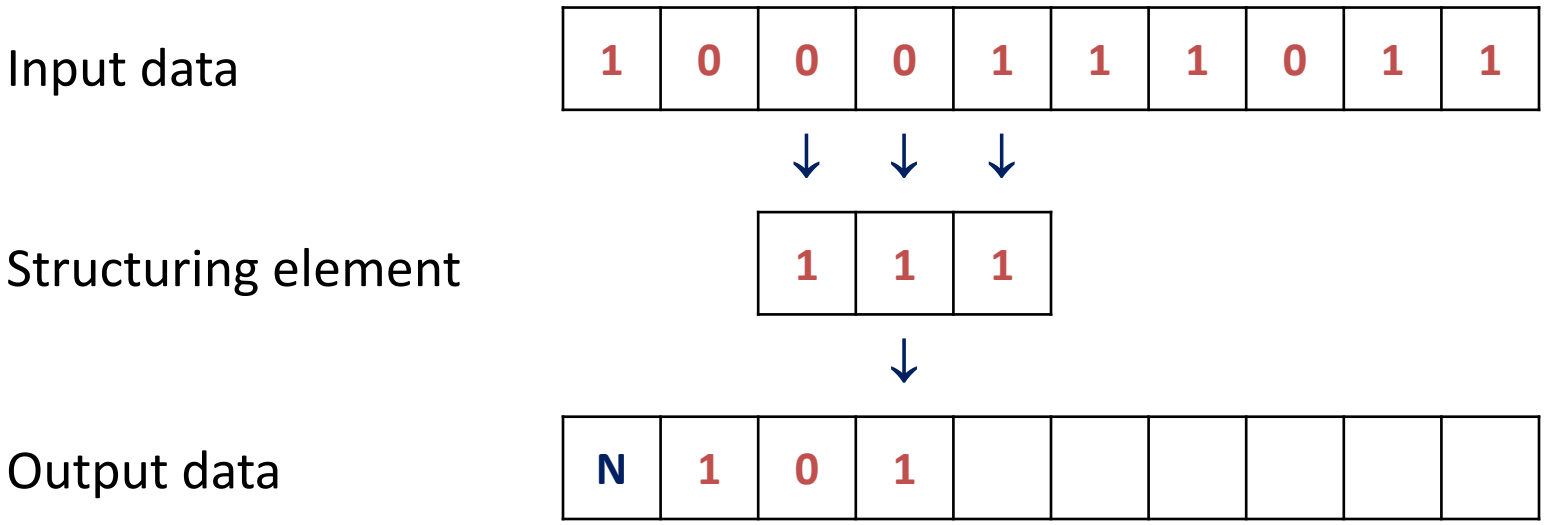
– If one of the pixels in the neighborhood defined by the SE is ‘1’ then the value of the output pixel is set to ‘1’



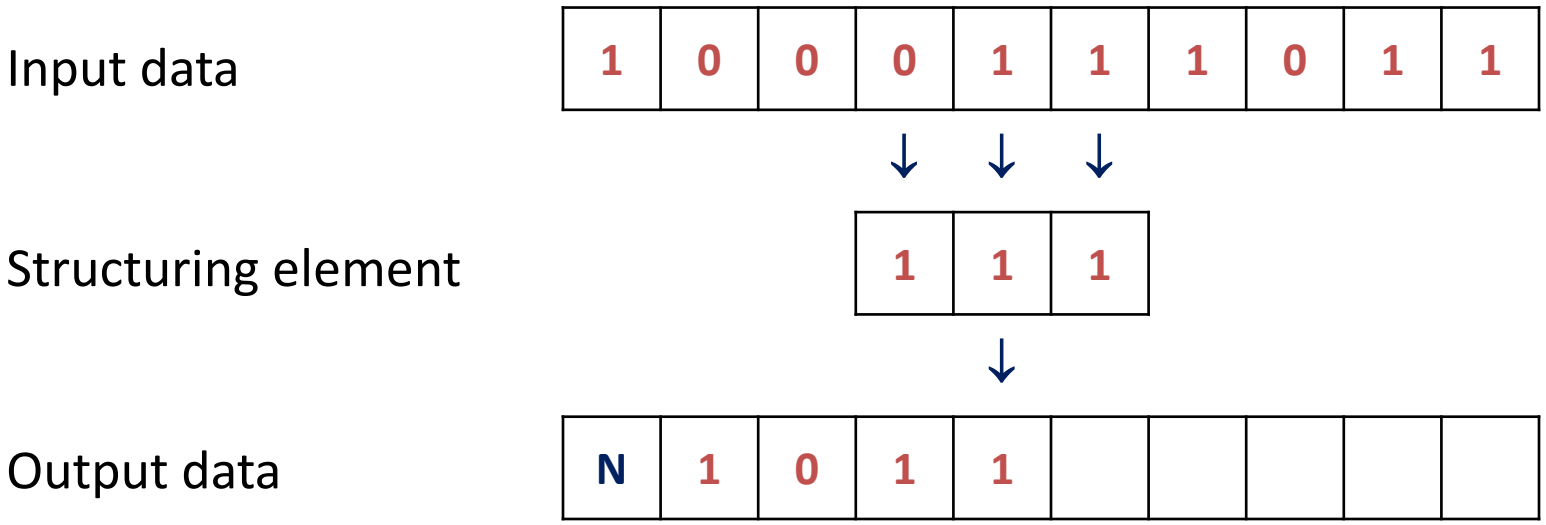
Example of Dilation



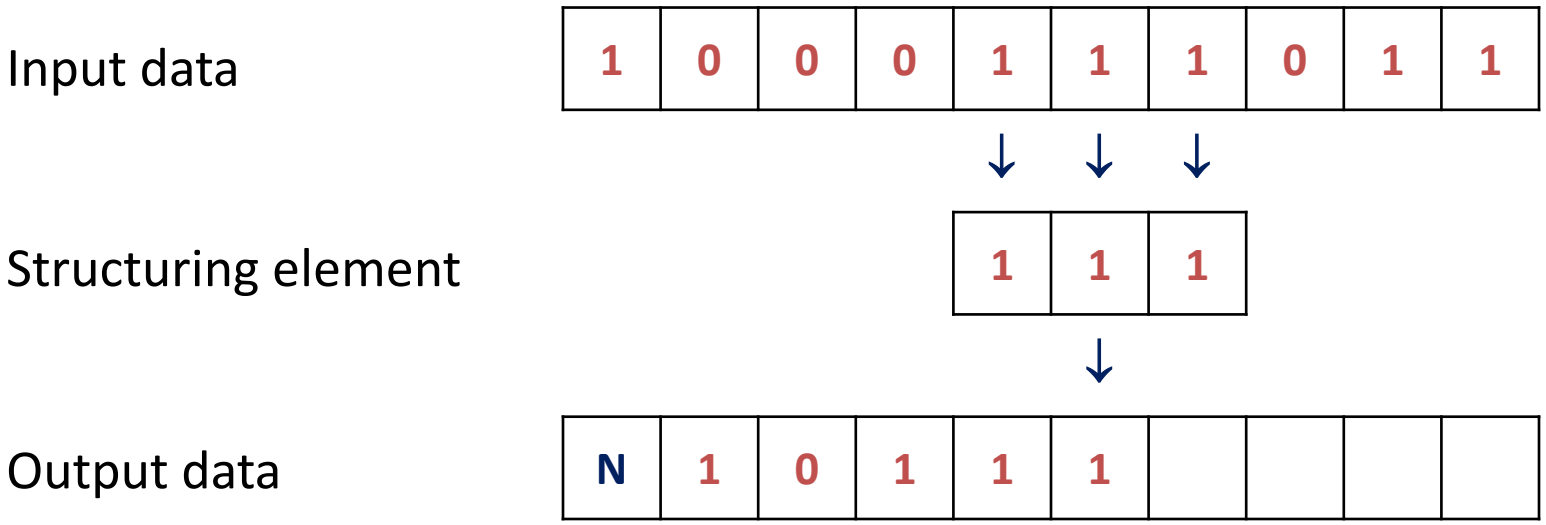
Example of Dilation



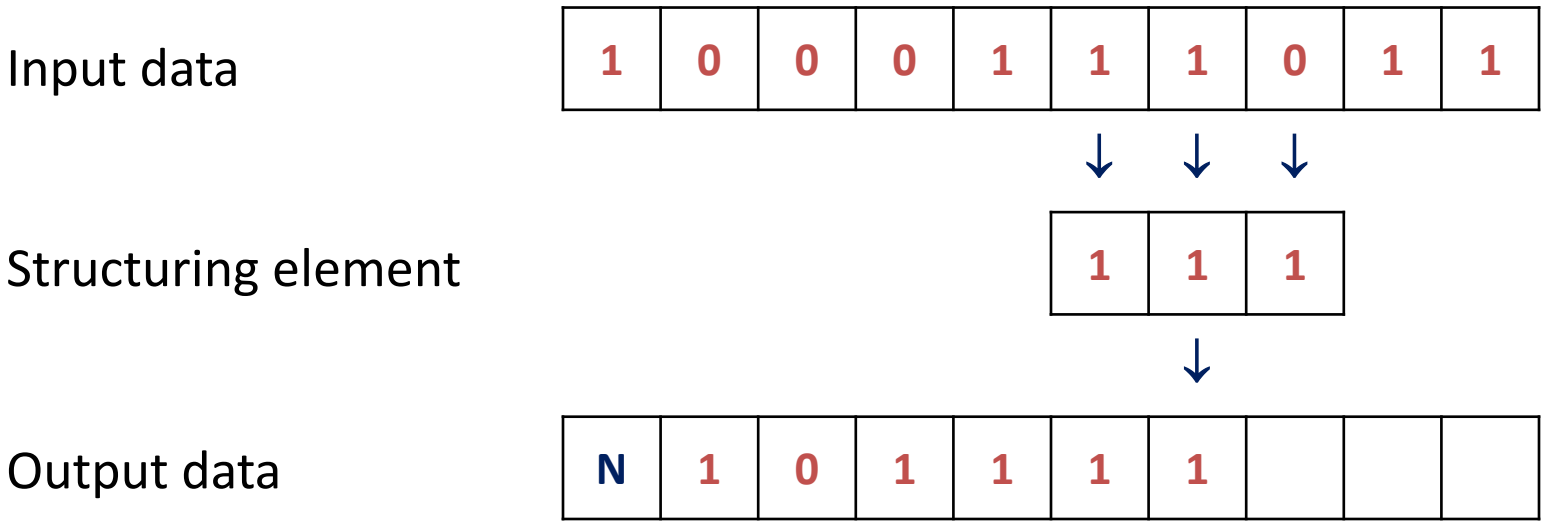
Example of Dilation



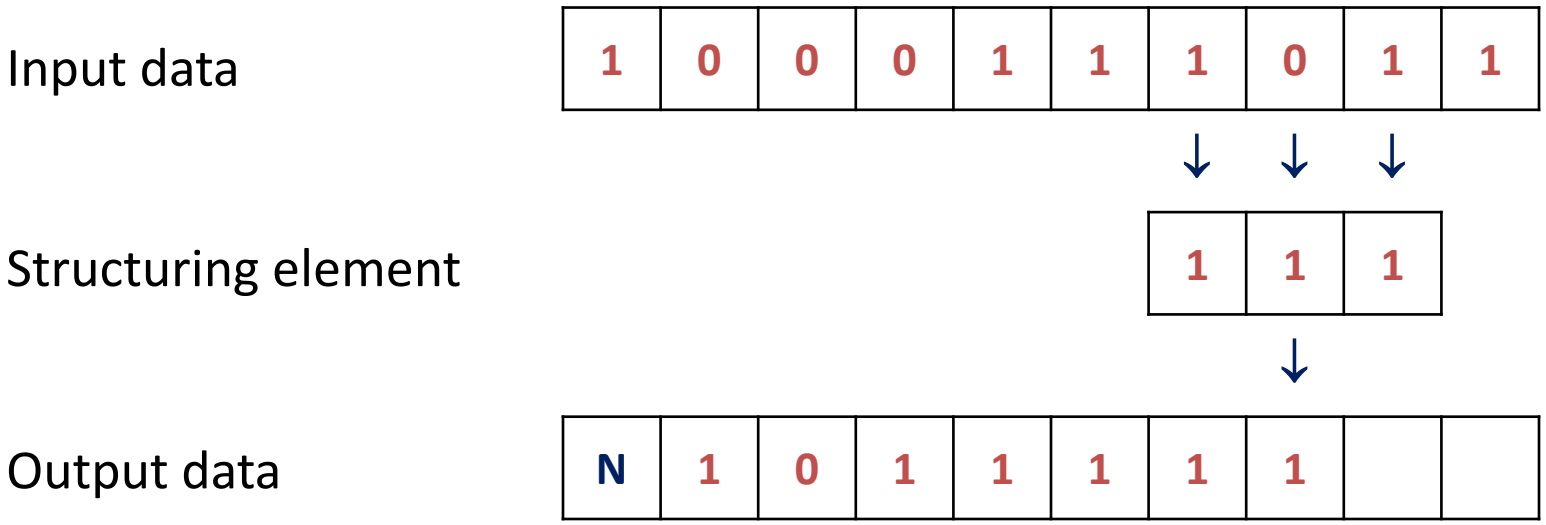
Example of Dilation



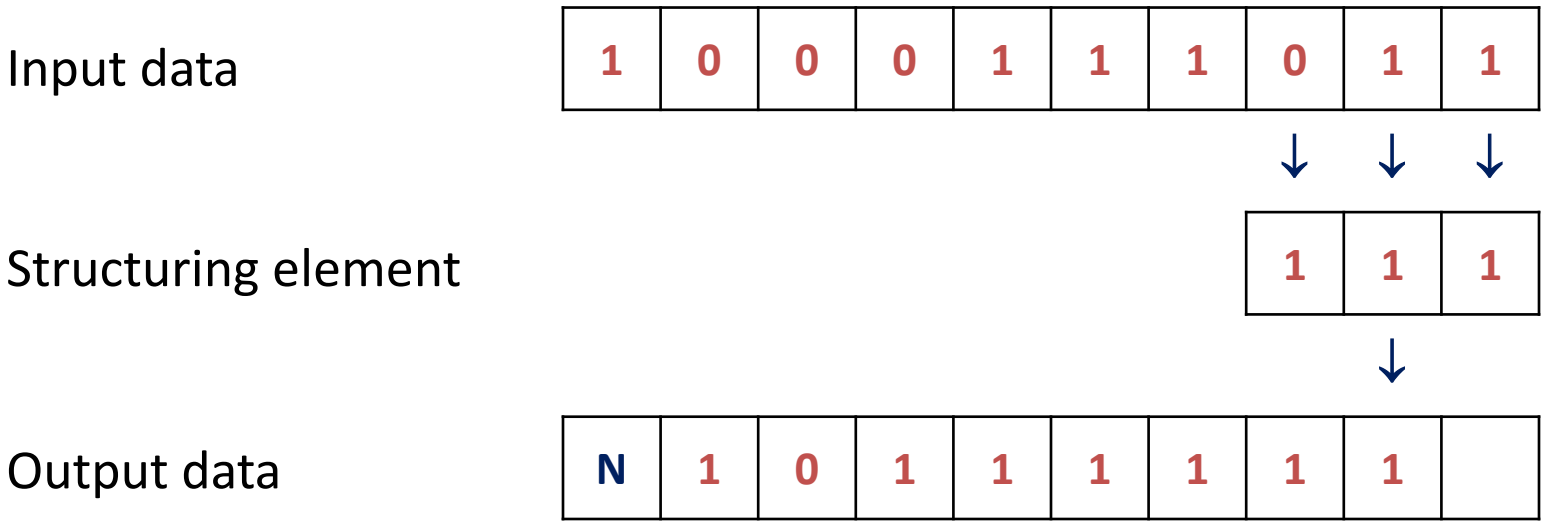
Example of Dilation



Example of Dilation



Example of Dilation



Example of Dilation

Input data

1	0	0	0	1	1	1	0	1	1

Structuring element

Output data

N	1	0	1	1	1	1	1	1	N

What is Dilation ?

- Dilation is an important morphological operation
  - Dilation adds pixels to the boundaries of objects in an image.
  - The number of pixels added to the objects depends on the size and shape of the Structuring Element.

What is Dilation ?

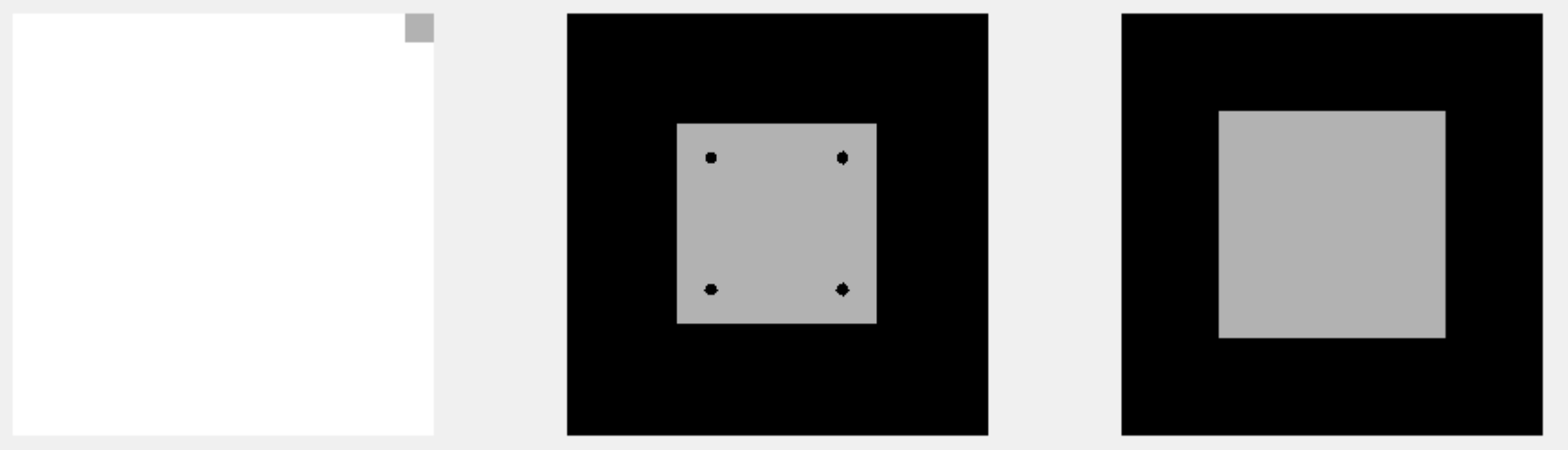
- Dilation is the set of all points in the image, where the structuring element “touches” the foreground.
- If the structuring element touches the foreground object, write “1” at the origin of the structuring element in the output image!

Where Dilation is used for ?

- To merge (connect) sparse objects

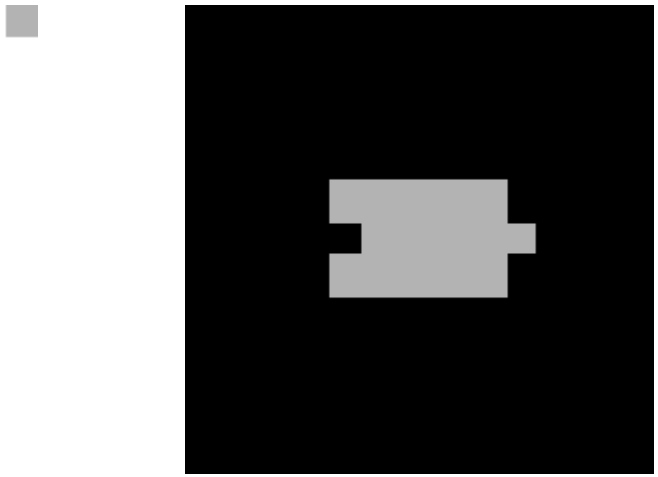
Where Dilation is used for ?

- Filling small holes



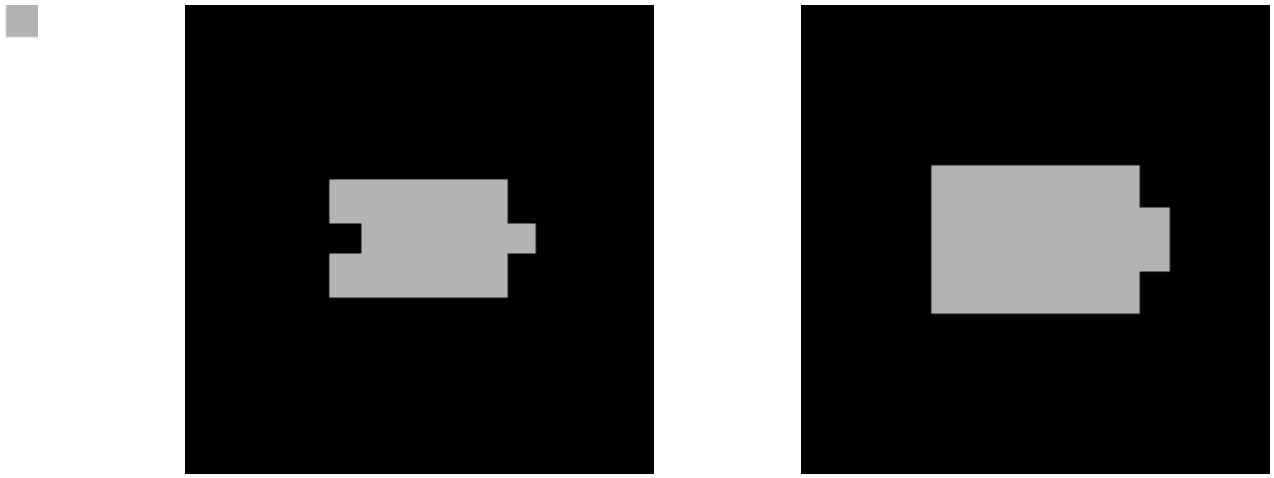
Do it yourself

- Try it yourself with Matlab!



Do it yourself

- Why do you need this operation ? Quality inspection?



- Why do you need this operation ? Quality inspection?

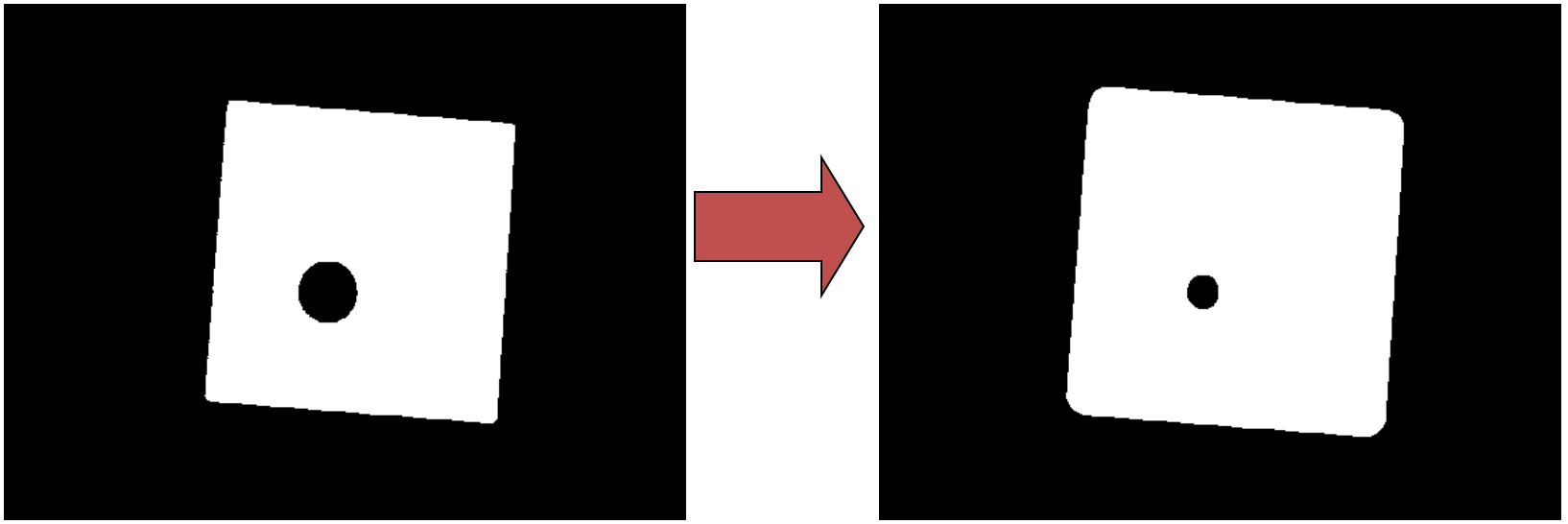
Image restoration: Dilation can be used to restore or reconstruct an image that has been damaged or degraded by noise or other artifacts. By expanding the regions of the image that still contain useful information, dilation can help to fill in missing or damaged parts of the image.

Image enhancement: Dilation can be used to enhance or exaggerate certain features in an image, such as edges, boundaries, or other details. By expanding the regions of the image that contain these features, dilation can help to make them more prominent or visible.

Object detection: Dilation can be used to detect objects in an image that are partially occluded or obscured by other objects or noise. By expanding the regions of the image that contain the object, dilation can help to make it easier to detect and distinguish from other features in the image.

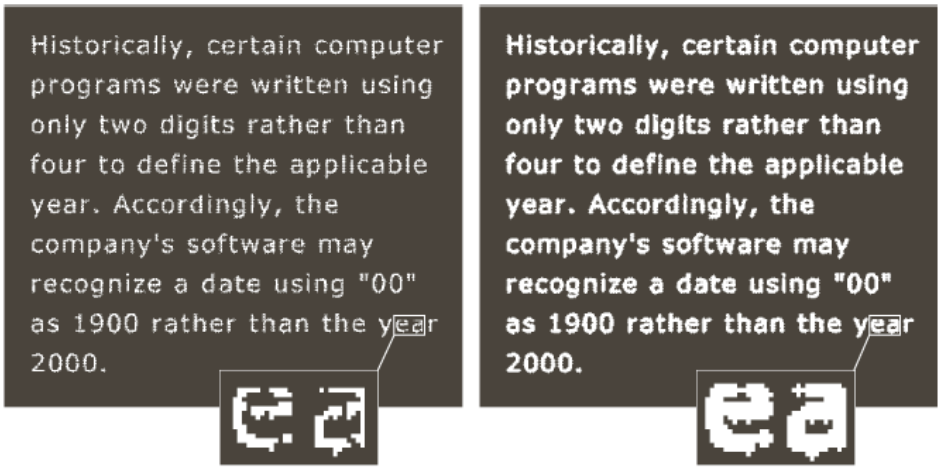
Binary image processing: Dilation is commonly used in binary image processing to fill in gaps or holes in an object or to connect separate objects that are close together. By expanding the regions of the image that correspond to the object, dilation can help to make it more complete or connected.

Example: Dilation



- Image get lighter, more uniform intensity

Example: Dilation



0	1	0
1	1	1
0	1	0

a c
b

FIGURE 9.7
(a) Sample text of poor resolution with broken characters (see magnified view).
(b) Structuring element.
(c) Dilation of (a) by (b). Broken segments were joined.

From Gonzalez and Woods book on Image Processing

Dilation and Erosion

- Dilation and Erosion are basic morphological operations
- Are dual to each other:
  - Erosion shrinks foreground, enlarges Background
  - Dilation enlarges foreground, shrinks background

Erosion and Dialation

im =

0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	0
0	1	0	0	0	0	1	0
0	1	1	1	1	1	1	0
0	1	1	1	1	1	1	0
0	1	1	1	1	1	1	0
0	1	1	1	1	1	1	0
0	0	1	1	1	1	0	0

se =

1	1	1
1	1	1
1	1	1

Do it yourself

Padding the boundary with the same value as the neighbor if it is needed.

Padding

se =

1	1	1
1	1	1
1	1	1

im =

0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	0
0	1	0	0	0	0	1	0
0	1	1	1	1	1	1	0
0	1	1	1	1	1	1	0
0	1	1	1	1	1	1	0
0	1	1	1	1	1	1	0
0	0	1	1	1	1	0	0

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	1	0	0
0	0	1	1	1	1	1	1	0	0
0	0	1	1	1	1	1	1	0	0
0	0	1	1	1	1	1	1	0	0
0	0	1	1	1	1	1	1	0	0
0	0	0	1	1	1	1	0	0	0
0	0	0	1	1	1	1	0	0	0

Padding the boundary with the same value as the neighbor if it is needed.

Erosion Result

im =

0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	0
0	1	0	0	0	0	1	0
0	1	1	1	1	1	1	0
0	1	1	1	1	1	1	0
0	1	1	1	1	1	1	0
0	1	1	1	1	1	1	0
0	0	1	1	1	1	0	0

imo =

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	1	1	1	1	0	0
0	0	1	1	1	1	0	0
0	0	0	1	1	0	0	0
0	0	0	1	1	0	0	0

Dilation Result

im =

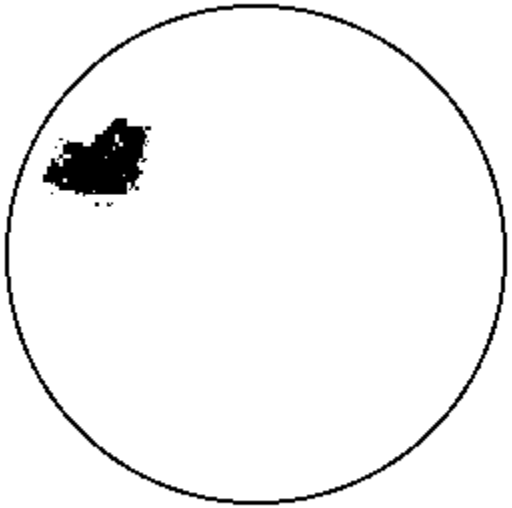
0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	0
0	1	0	0	0	0	1	0
0	1	1	1	1	1	1	0
0	1	1	1	1	1	1	0
0	1	1	1	1	1	1	0
0	1	1	1	1	1	1	0
0	0	1	1	1	1	0	0

imo =

1	1	1	0	0	1	1	1
1	1	1	0	0	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1

Binary mask for bleeding detection in WCE

- This is the "intermediate" (not final) bleeding mask



Noise
Small hole

Binary mask for bleeding detection in WCE

- Which morphological operation we used here to generate the final mask ?

Counting Coins

- How many coins (Jiao) are there ?

Counting coins is difficult because they touch each other!

Solution: binarization and Erosion separates them!

code

Boundary extraction

Boundary extraction

- Boundary extraction (sometime edge detection) can be obtained by :
  - Dilate input image
  - Subtract input (XOR logical operation) image from dilated image

Boundary extraction

- Boundary extraction (sometime edge detection) can be obtained by :
  - Dilate input image
  - Subtract input (XOR logical operation) image from dilated image

The truth table for XOR can be expressed as follows:

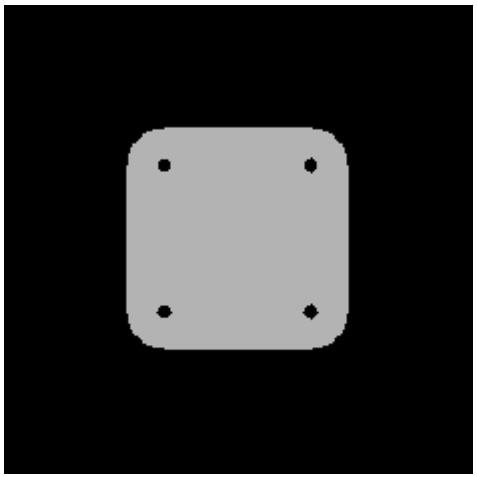
Input 1	Input 2	Output
0	0	0
0	1	1
1	0	1
1	1	0

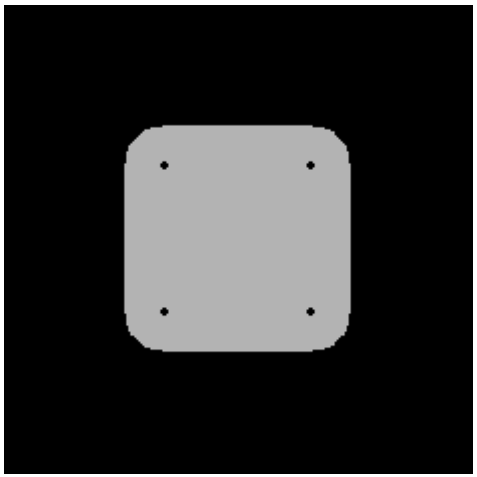
Boundary extraction

- Boundary extraction (sometime edge detection) can be obtained by :
  - Dilate input image
  - Subtract input (XOR logical operation) image from dilated image
- How to control the width of the extracted boundary?
- Could we use the erosion then subtracting ?
  - What will happen with small objects if we use this approach ?

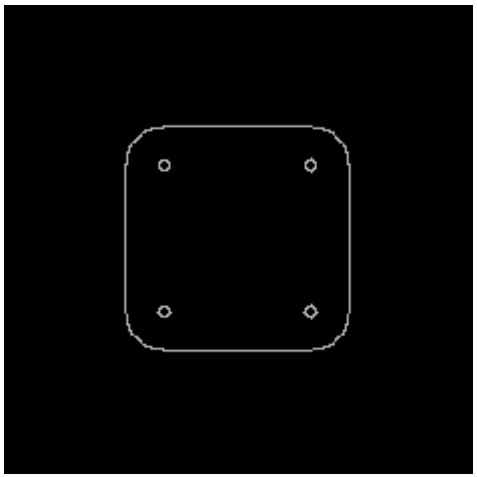
Example: Boundary extraction

– Extract features from boundary





xor



code

- # Why do you need this operation?

Object recognition: Boundary extraction can be used to extract features or characteristics of an object in an image, such as its shape, size, or orientation. This information can be used to recognize or identify the object, or to distinguish it from other objects in the image.

Image segmentation: Boundary extraction can be used as a pre-processing step for image segmentation, which involves dividing an image into multiple regions or objects. By isolating the boundaries of these regions, it becomes easier to separate them and extract features from each region.

Object tracking: Boundary extraction can be used to track the movement or motion of an object in a video stream over time. By extracting the boundary of the object in each frame, it becomes possible to track its position, velocity, and other properties.

Image enhancement: Boundary extraction can be used to enhance or emphasize certain features or structures in an image, such as edges, corners, or other details. By highlighting the boundaries of these features, it becomes easier to distinguish them from other parts of the image and to analyze their properties.

code

Example: Boundary extraction

- 

a b

FIGURE 9.14
(a) A simple binary image, with 1s represented in white. (b) Result of using Eq. (9.5-1) with the structuring element in Fig. 9.13(b).

From Gonzalez and Woods book on Image Processing

Thanks