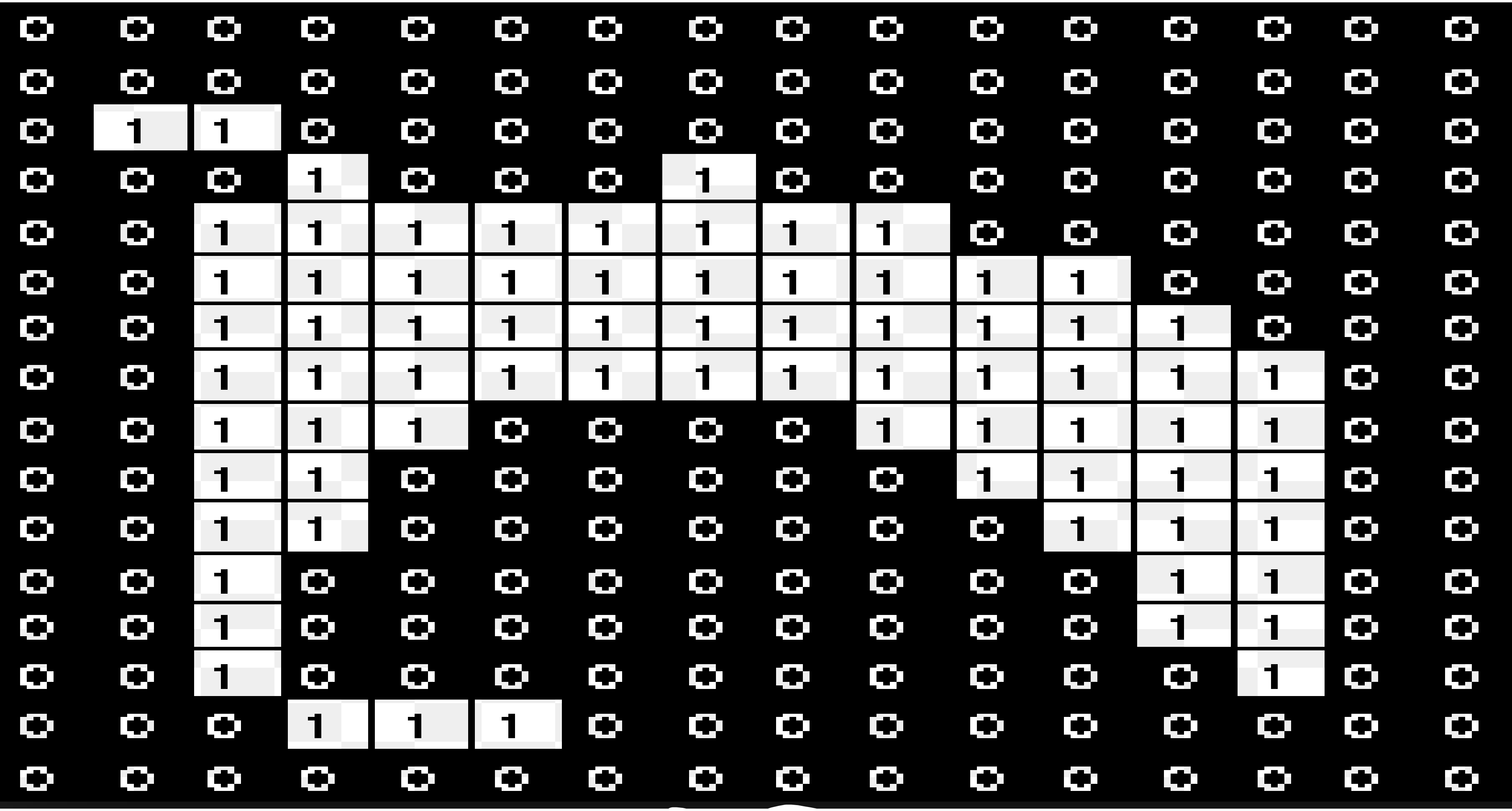
Morphological Image Processing



What Is Morphology?

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

Morphological Image Processing

- Morphological Operators
  - Used generally on binary images, e.g., background subtraction results!
  - Used on gray value images, if viewed as a stack to binary images.
- Good for, e.g.,
  - Noise removal in background
  - Removal of holes in foreground / background

Morphological Operations

- Structuring Element
- Erosion
- Dilation
- Opening
- Closing
- Hit-and-miss Operation, Thinning, Thickening

Structuring Element (Kernel)

- Structuring Elements can have varying sizes.
- Usually, element values are 0,1 and none(!).
- Structural Elements have an origin.
- For thinning, other values are possible.
- Empty spots in the Structuring Elements are *don't care*'s!

Box

1	1	1
1	1	1
1	1	1

Disc

	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	0	
1		0

1	1	1
1	10	1
1	1	1

Examples of structuring elements

Structuring Element (Kernel)

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

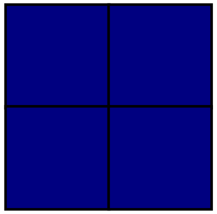
1	1	1
1	**1**	1
1	1	1

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

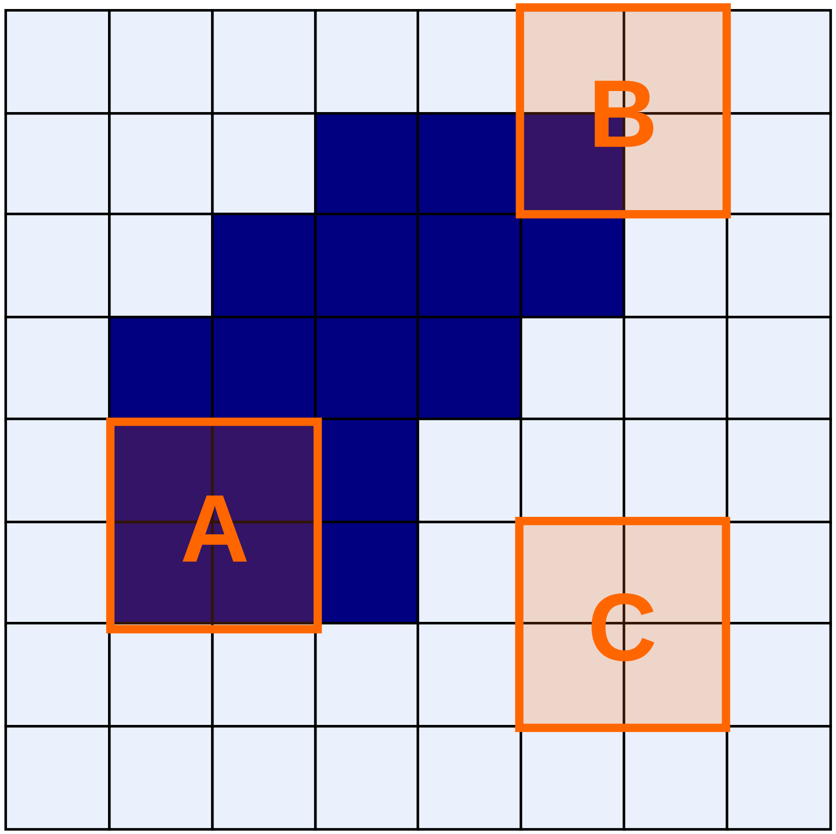
0	0	1	0	0
0	1	1	1	0
1	1	**1**	1	1
0	1	1	1	0
0	0	1	0	0

Examples of structuring elements

Structuring Elements: Hits, Fits & Miss



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.

Miss: All ***on pixels*** in the structuring element do not cover ***on pixels*** in the image.

All morphological processing operations are based on these simple ideas

Fundamental Operations

- Fundamentally morphological image processing is very much 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.**

Dilation & Erosion

- Basic operations.

- Are dual to each other:
  - Dilation enlarges Foreground, shrinks Background
  - Erosion shrinks Foreground, enlarges Background

Dilation

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:

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

Dilation

- **Dilation** is the set of all points in the image, where the structuring element "touches" the foreground.

- Consider each pixel in the input image
  - If the structuring element touches the foreground image, write a "1" at the origin of the structuring element!
- Input:
  - Binary Image
  - Structuring Element, containing only 1s!!

Morphological Dilation

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

Output Image

	1								

Morphological Dilation

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

Output Image

	1	0							

Morphological Dilation

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

Output Image

	1	0	1						

Morphological Dilation

Input image	1	0	0	0	1	1	1	0	1	1

Structuring Element	1	1	1

Output Image		1	0	1	1					

Morphological Dilation

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

Output Image

	1	0	1	1	1				

Morphological Dilation

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

Output Image

	1	0	1	1	1	1			

Morphological Dilation

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

Output Image

	1	0	1	1	1	1	1		

Morphological Dilation

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

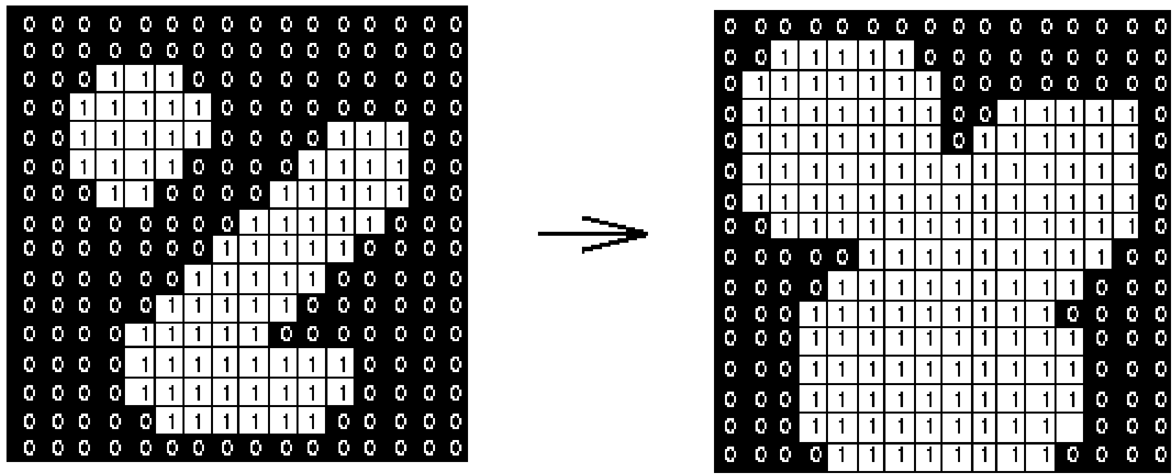
1	1	1

Output Image

	1	0	1	1	1	1	1	1	

Morphological Dilation: Example

- **Dilation** is an important morphological operation



- Applied **Structuring Element**:

1	1	1
1	1	1
1	1	1

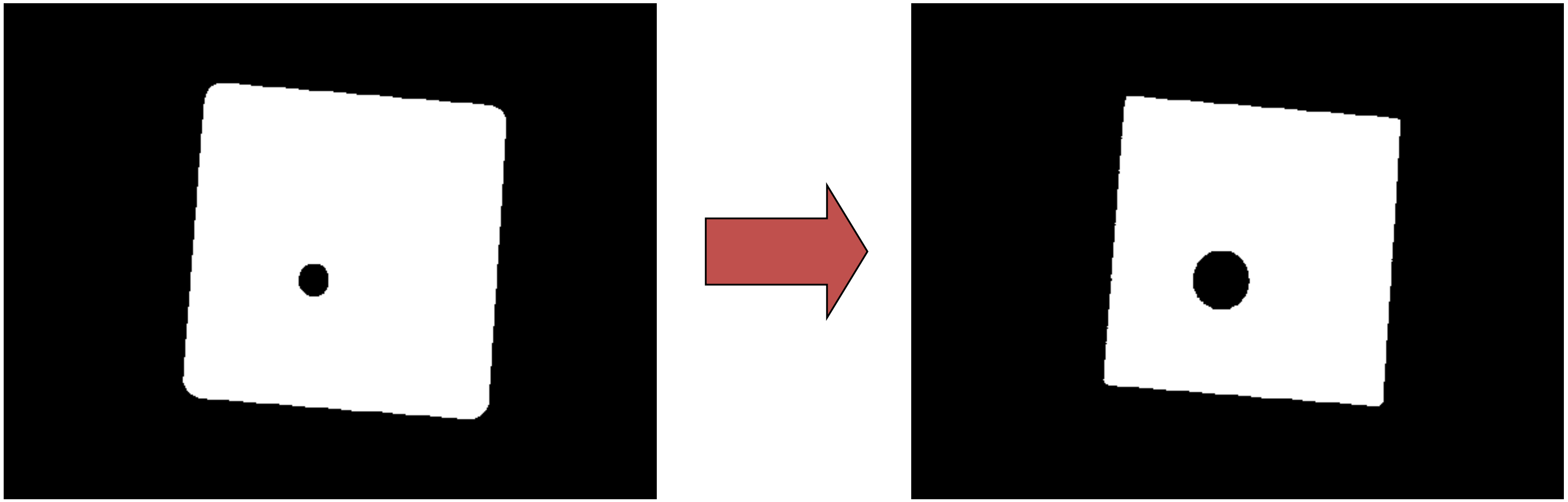
Set of coordinate points =

{ (-1, -1), (0, -1), (1, -1),

(-1, 0), (0, 0), (1, 0),

(-1, 1), (0, 1), (1, 1) }

Morphological Dilation: Example



Erosion

Morphological 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:

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

Morphological Erosion

Input image	1	0	0	0	1	1	1	0	1	1
Structuring Element	1	1	1							
Output Image		0								

Morphological Erosion

Input image	1	0	0	0	1	1	1	0	1	1
Structuring Element		1	1	1						
Output Image		0	0							

Morphological Erosion

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

Output Image

	0	0	0						

Morphological Erosion

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

Output Image

	0	0	0	0					

Morphological Erosion

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

Output Image

	0	0	0	0	1				

Morphological Erosion

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

Output Image

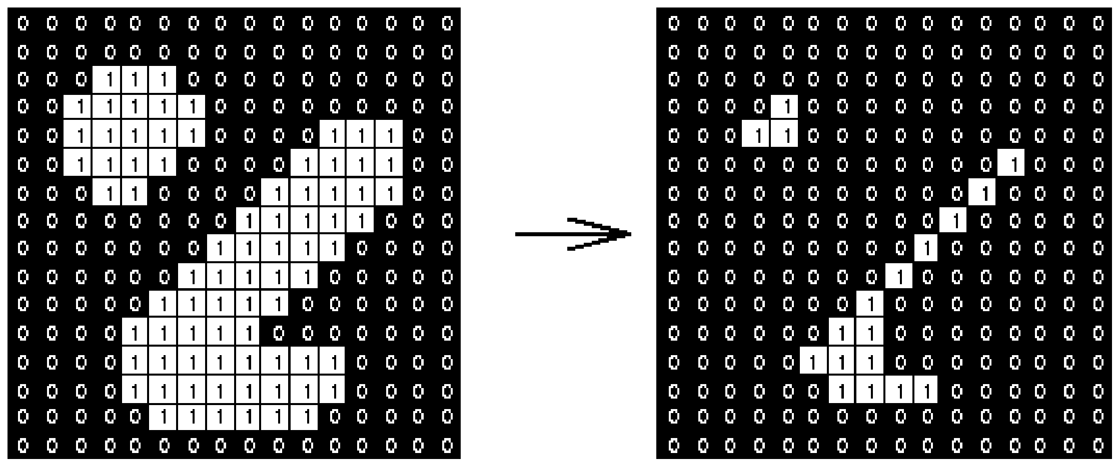
	0	0	0	0	1	0	0		

Morphological Erosion

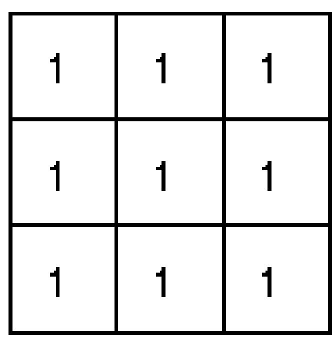
Input image	1	0	0	0	1	1	1	0	1	1
Structuring Element								1	1	1
Output Image		0	0	0	0	1	0	0	0	

A first Example: Erosion

- **Erosion** is an important morphological operation



- Applied **Structuring Element**:



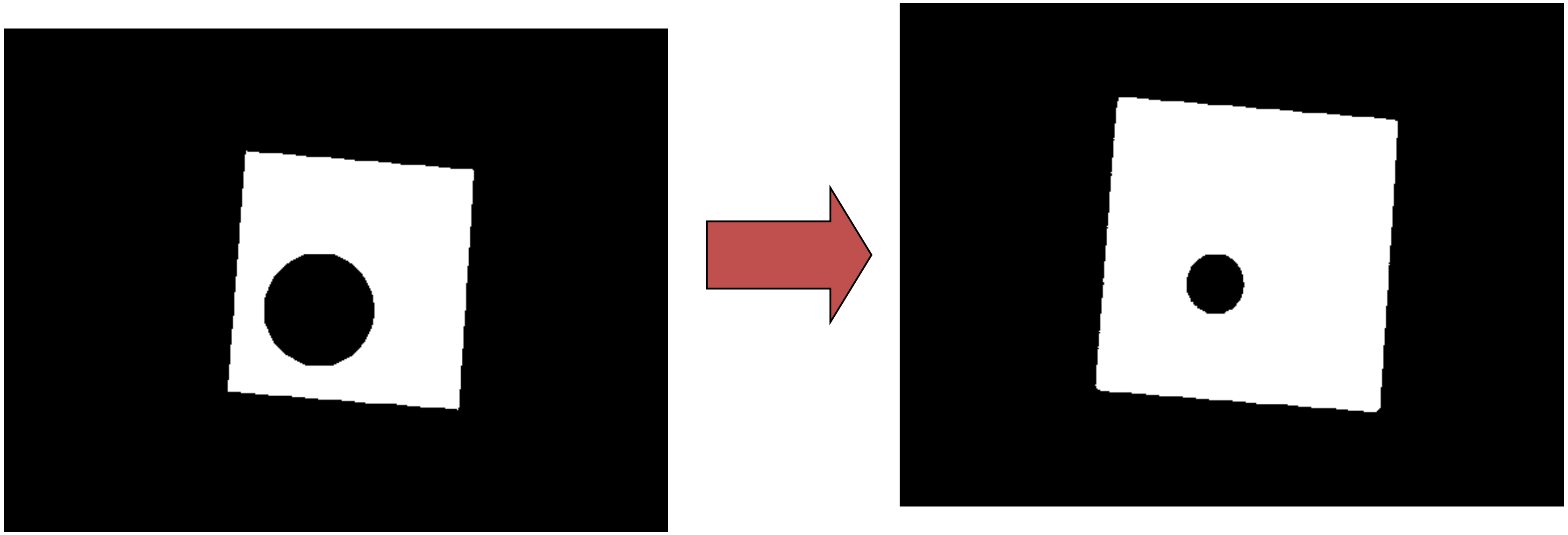
Set of coordinate points =

{ (-1, -1), (0, -1), (1, -1),

(-1, 0), (0, 0), (1, 0),

(-1, 1), (0, 1), (1, 1) }

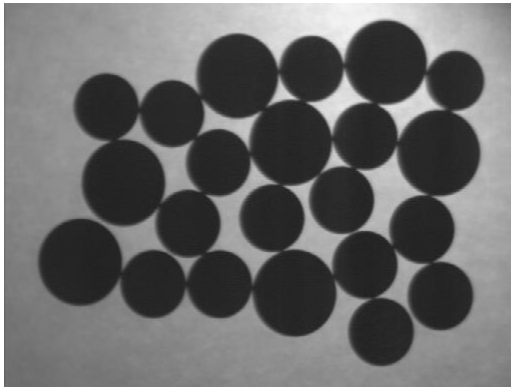
Another example of erosion

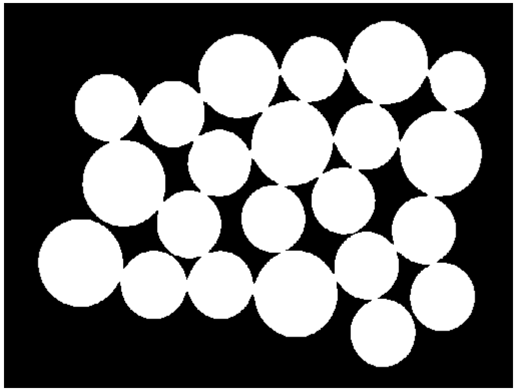


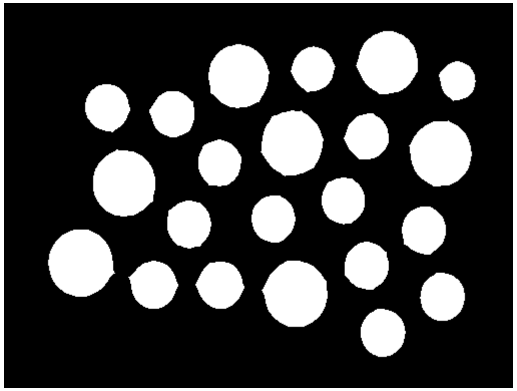
Erosion on Gray Value Images

Counting Coins

- Counting coins is difficult because they touch each other!
- Solution: Binarization and Erosion separates them!







Opening & Closing

- Important operations
- Derived from the fundamental operations
  - Dilation
  - Erosion
- Usually applied to binary images, but gray value images are also possible
- Opening and closing are dual operations

Opening

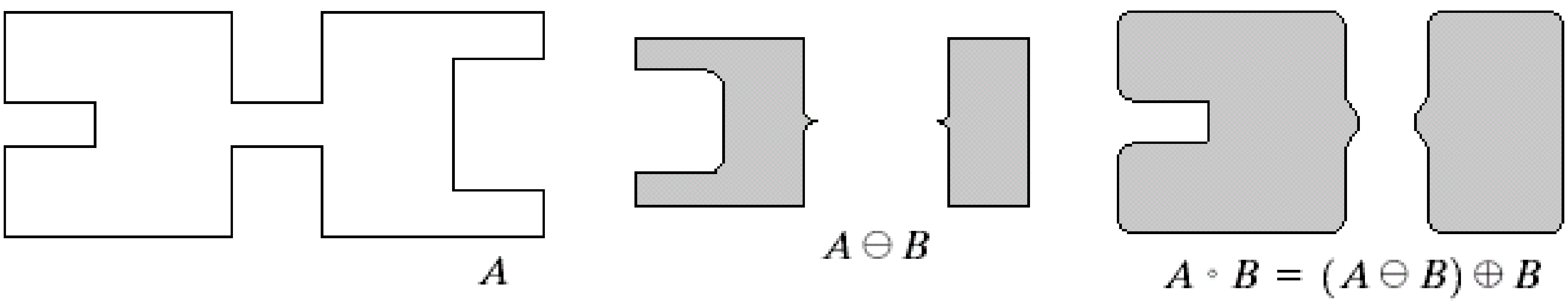
Opening

- Similar to Erosion
  - Spot and noise removal
  - Less destructive
- **Erosion next dilation**
- S*ame structuring element for both operations.*
- Input:
  - Binary Image
  - Structuring Element, containing only 1s!
- Smoothens contours, breaks narrow isthmuses, and eliminates small island and sharp peaks.

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)

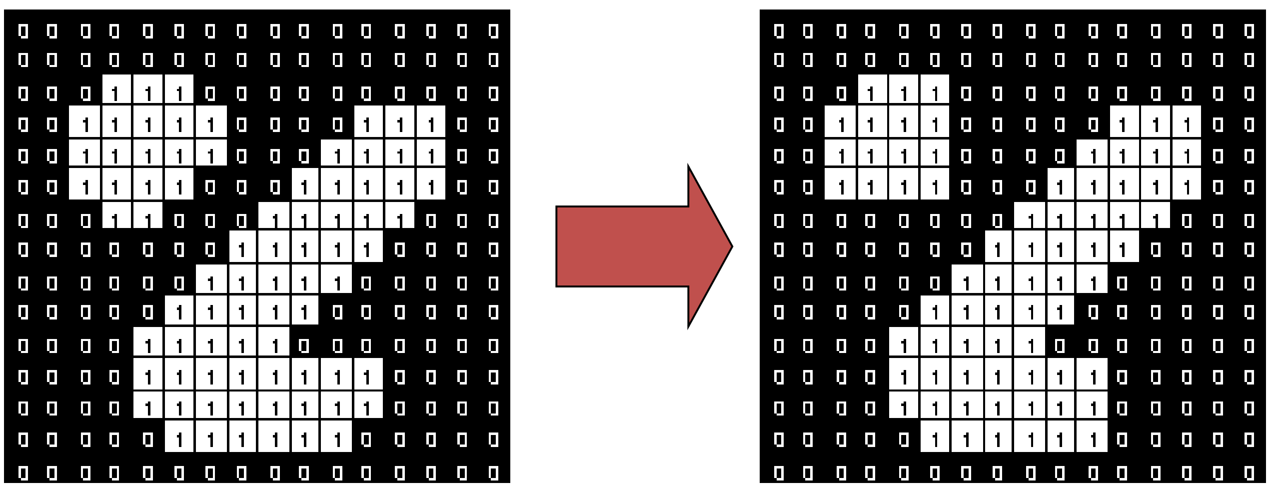
Note: A disc shaped structuring element is used

Opening

- Take the structuring element (SE) and slide it around *inside* each foreground region.
  - All pixels which can be covered by the SE with the SE being entirely within the foreground region will be preserved.
  - All foreground pixels which *cannot* be reached by the structuring element without lapping over the edge of the foreground object will be eroded away!
- Opening is **idempotent:** Repeated application has no further effects!

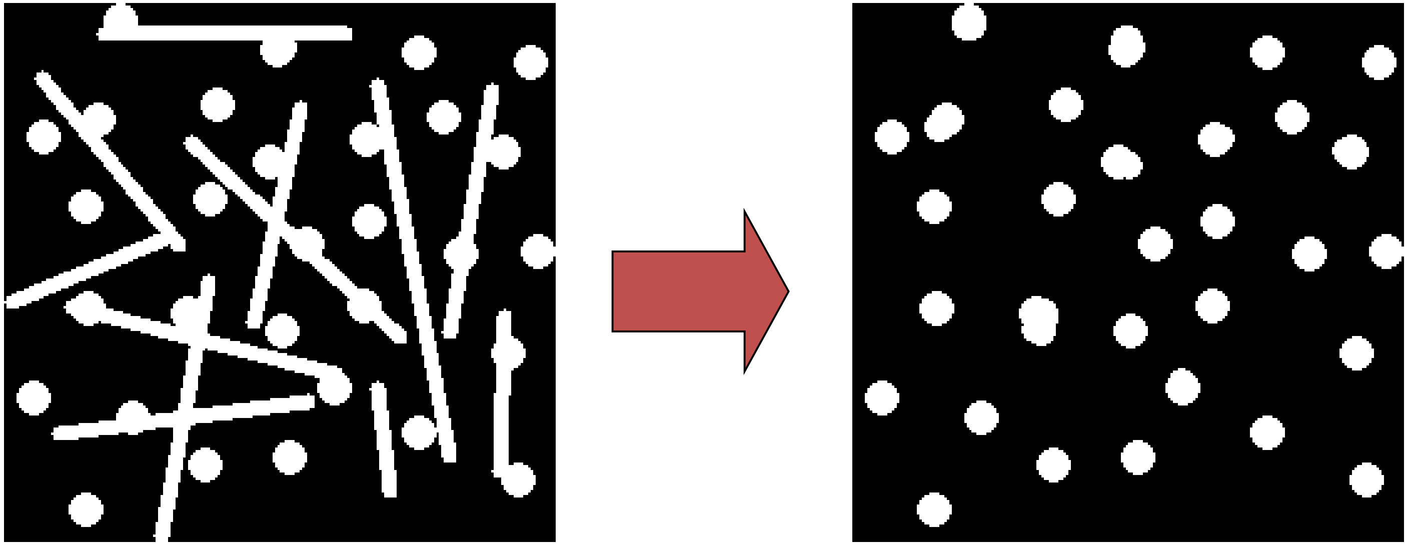
Opening: Example

- Structuring element: 3x3 square



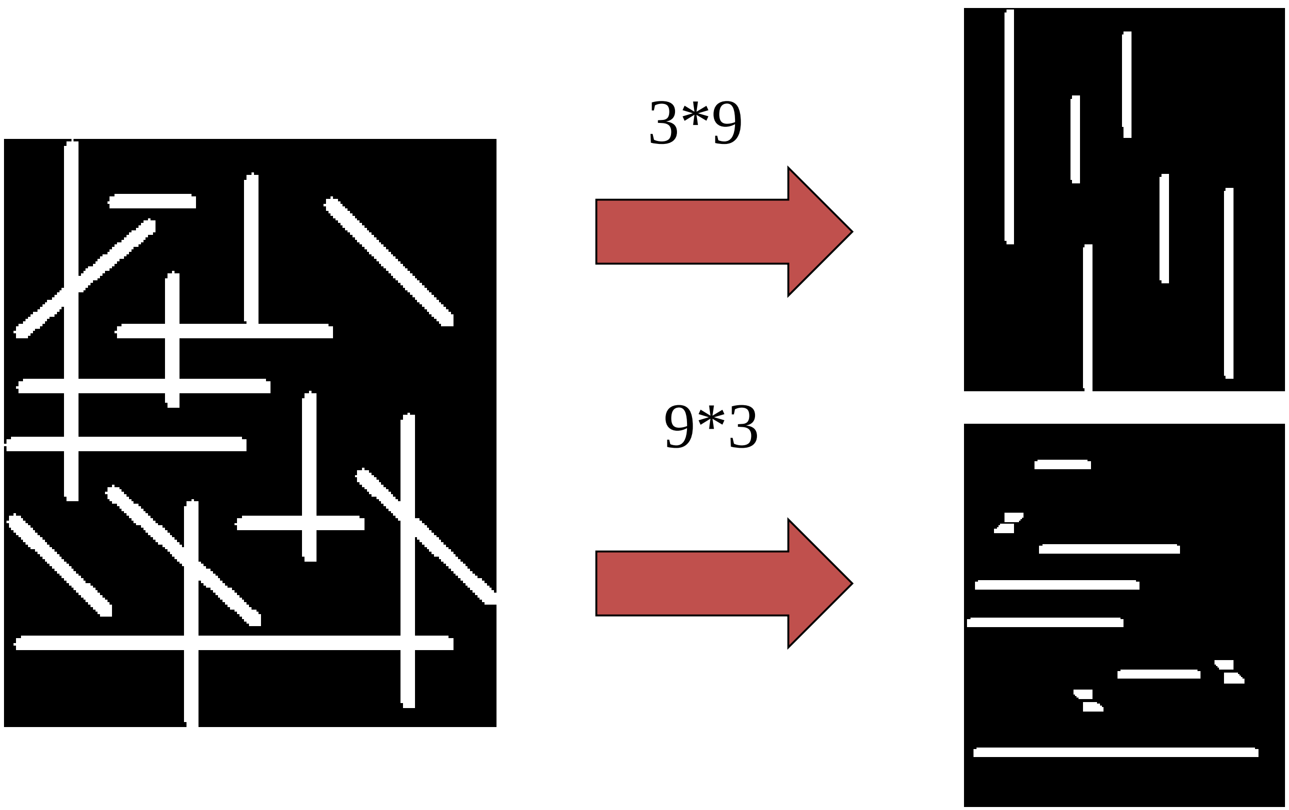
Opening: Example

- Opening with a 11 pixel diameter disc



Opening: Example

- 3x9 and 9x3 Structuring Element



Closing

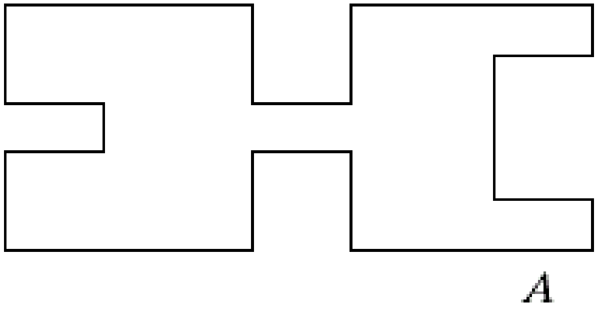
Closing

- Similar to Dilation
  - Removal of holes
  - Tends to enlarge regions, shrink background
- Closing is defined as a Dilatation, followed by an Erosion *using the same structuring element for both operations*.
- **Dilation next erosion**!
- Input:
  - Binary Image
  - Structuring Element, containing only 1s!
- Smoothens contours, fuses narrow breaks and long thin gulfs and eliminates small holes.

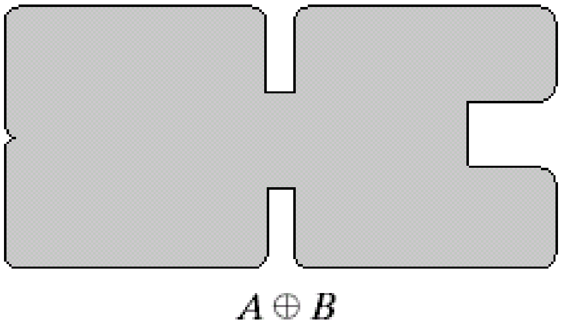
Closing

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

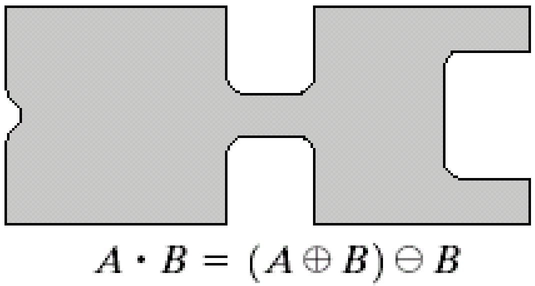
$$f \bullet s = (f \oplus s) \ominus s$$



Original shape



After dilation



After erosion (closing)

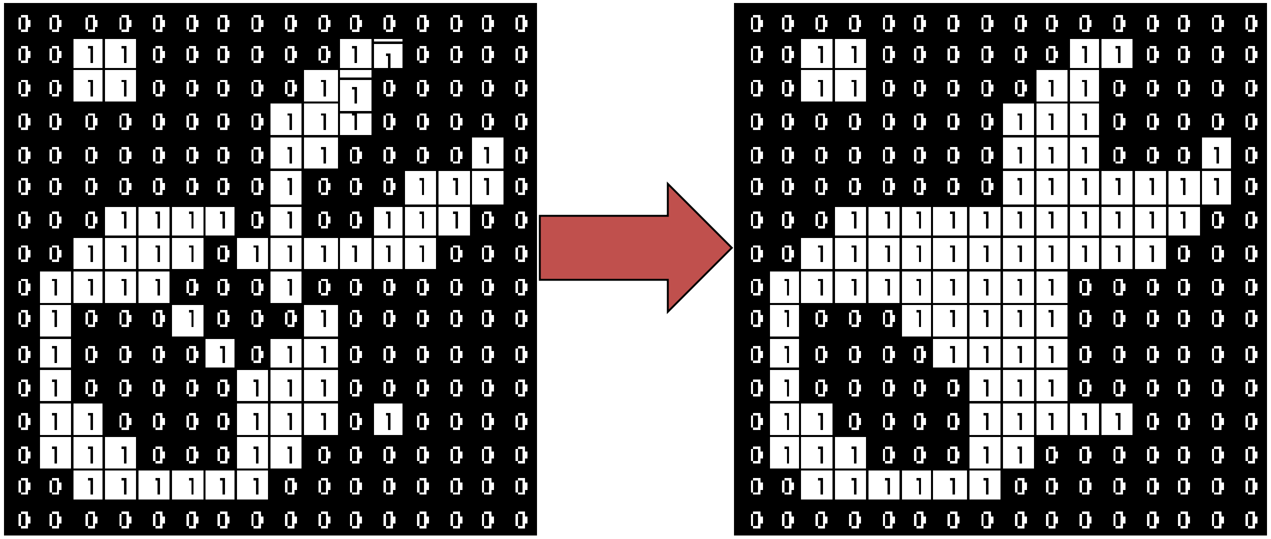
Note: A disc shaped structuring element is used

Closing

- Take the structuring element (SE) and slide it around *outside* each foreground region.
  - All background pixels which can be covered by the SE with the SE being entirely within the background region will be preserved.
  - All background pixels which cannot be reached by the structuring element without lapping over the edge of the foreground object will be removed.
- Opening is **idempotent:** Repeated application has no further effects!

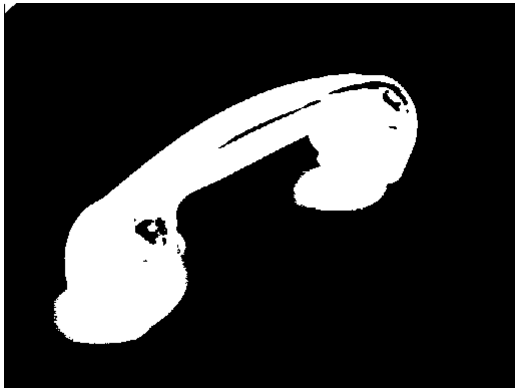
Closing Example

- Structuring element: 3x3 square

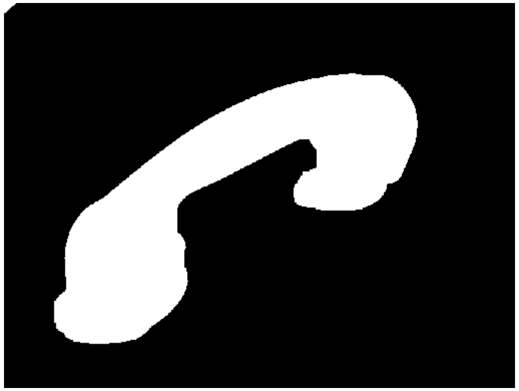


Closing Example

1. Threshold
2. Closing with disc of size 20



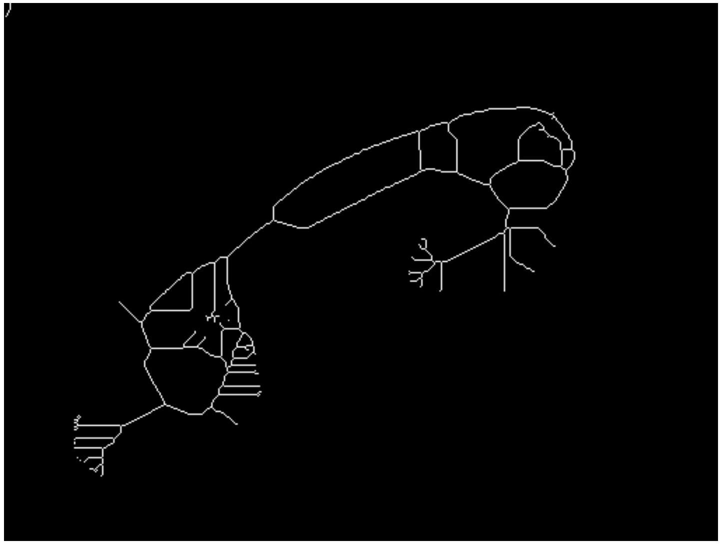
Thresholded



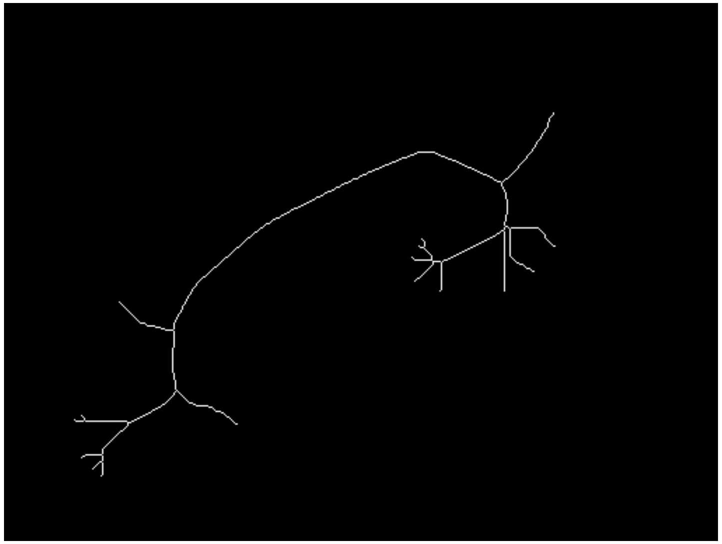
Closed

Closing Example

- Good for further processing: E.g. Skeleton operation looks better for closed image!



Skeleton of Thresholded



Skeleton of Thresholded and next closed

Opening and Closing

- Opening is the *dual* of closing
- *i.e.* opening the foreground pixels with a particular structuring element
- is equivalent to closing the background pixels with the same element.

HIT and MISS

HIT & MISS Transform

- Used to look for particular patterns of foreground and background pixels
- Very simple object recognition
- All other morphological operations can be derived from it!!
- Input:
  - Binary Image
  - Structuring Element, containing 0s and 1s!!

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

HIT & MISS Transform

- Example for a Hit-and-miss Structuring Element
- Contains 0s, 1s and *don't care*'s.
- Usually a "1" at the origin!

	1	
0	1	1
0	0	

HIT & MISS Transform

- Similar to Pattern Matching:
- If foreground and background pixels in the structuring element ***exactly match*** foreground and background pixels in the image, then the pixel underneath the origin of the structuring <u>element is set to the foreground color.</u>

HIT & MISS Transform: Corner Detection

- Structuring Elements representing four corners

	1	
0	1	1
0	0	

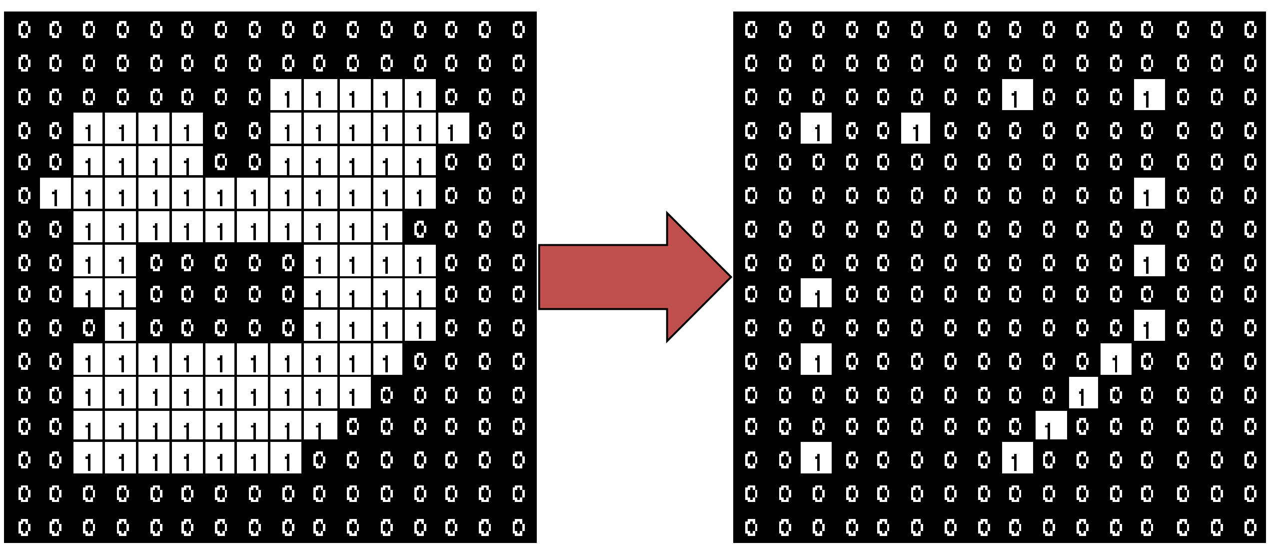
	1	
1	1	0
	0	0

	0	0
1	1	0
	1	

0	0	
0	1	1
	1	

HIT & MISS Transform: Corner Detection

- Apply each Structuring Element
- Use OR operation to combine the four results



Thinning

Thinning

1. Used to remove selected foreground pixels from binary images

2. After edge detection, lines are often thicker than one pixel.

3. Thinning can be used to thin those line to one pixel width.

Thinning Definition

- Let ***K*** be a kernel and ***I*** be an image

$$A \otimes B = A - (A \circledast B)$$

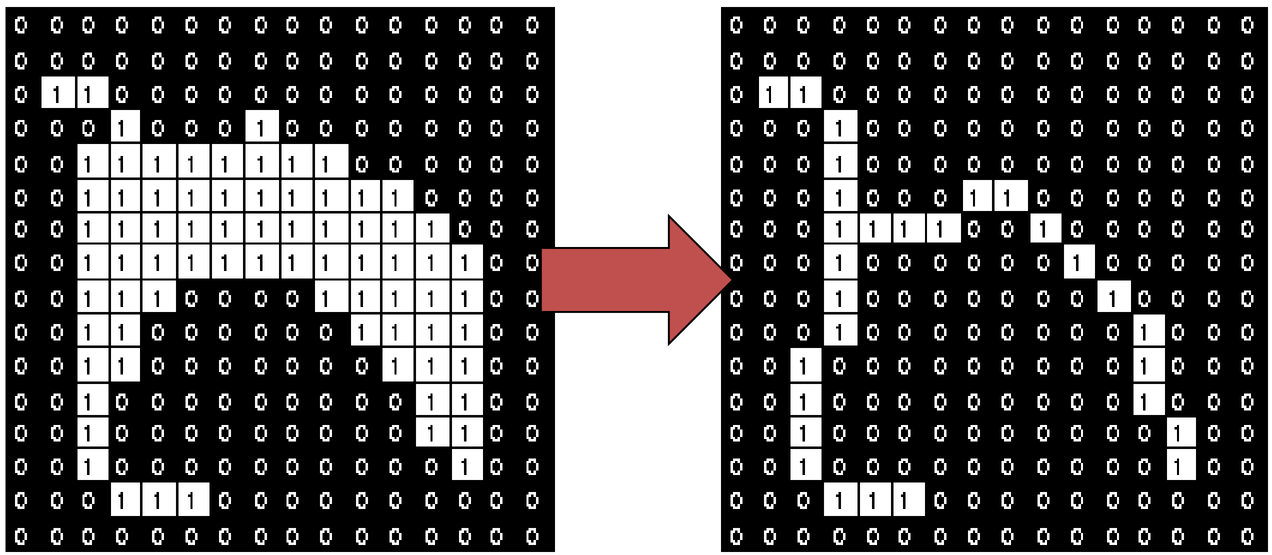
- If foreground and background fit the structuring element exactly, then the pixel at the origin of the SE is set to 0.

- Note that the value of the SE at the origin is 1 or *don't care*!

Thinning Definition

0	0	0
	1	
1	1	1

	0	0
1	1	0
	1	



We use two Hit-and-miss Transforms

Thickening

Thickening

- Used to grow selected regions of foreground pixels
- E.g. applications like approximation of *convex hull*

Thickening Definition

- Let ***K*** be a kernel and ***I*** be an image

$$A \odot B = A \cup (A \circledast B)$$

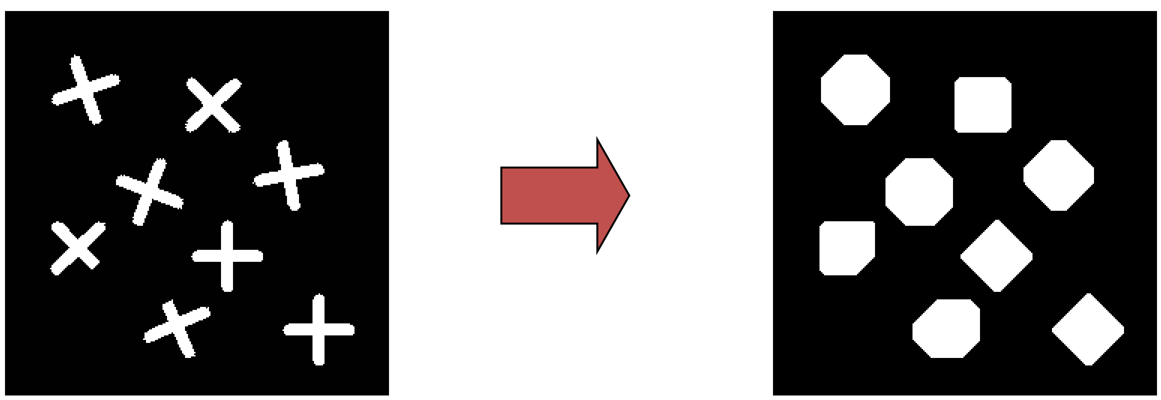
- If foreground and background match exactly the SE, then set the pixel at its origin to 1!

- Note that the value of the SE at the origin is 0 or *don't care*!

Thickening Example

1	1	
1	0	
1		0

	1	1
	0	1
0		1



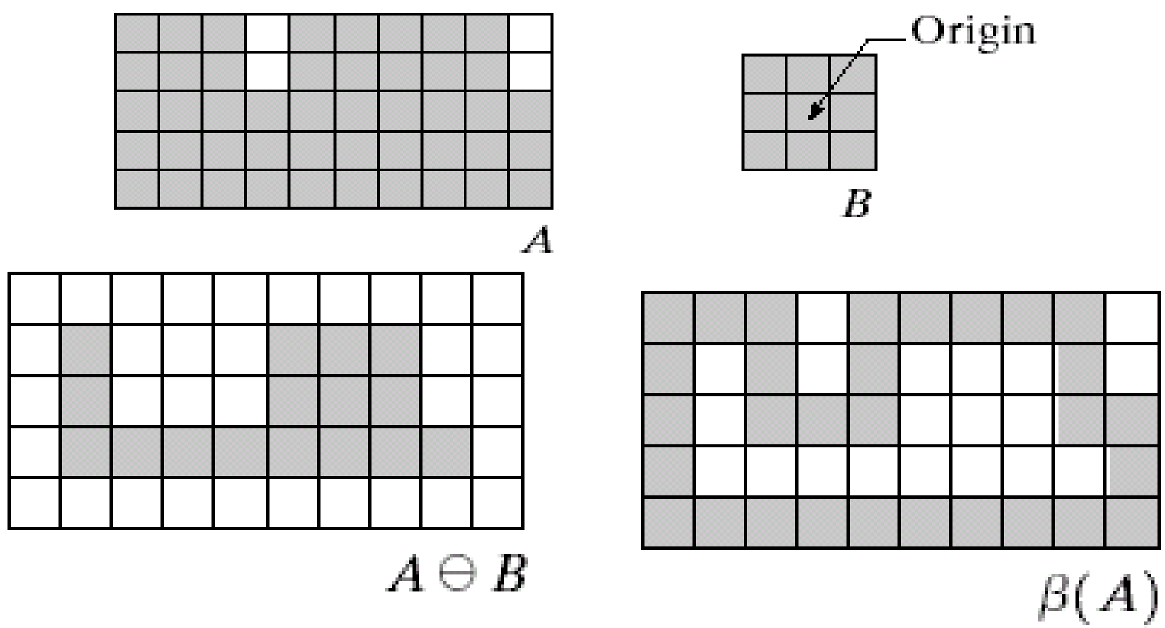
Boundary Extraction

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

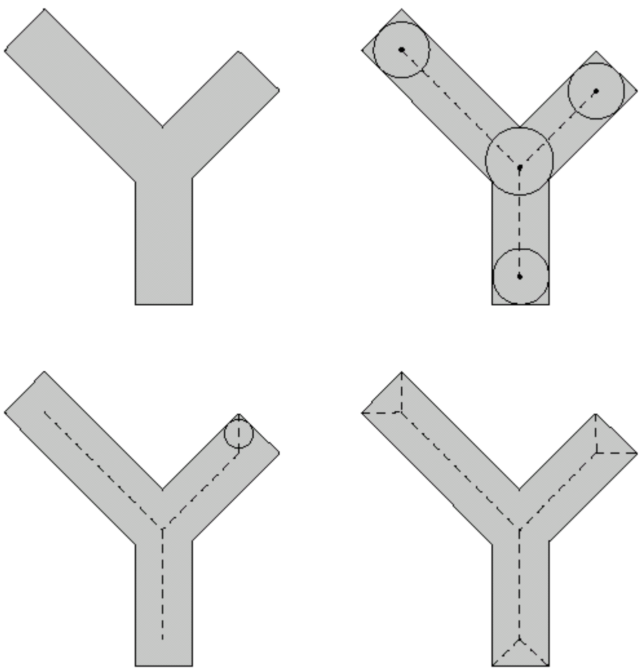
Original Image

Extracted Boundary

Skeletonization

Skeletons

Skeletonization is a process for reducing foreground regions to a skeletal remnant that largely preserves the extent and connectivity of the original region while throwing away most of the original foreground pixels.



$$S(A) = \bigcup_{k=0}^{K} S_k(A)$$

$$S_k(A) = (A - kB) - (A - kB) \circ B$$

Skeletons Example