06.09.2019

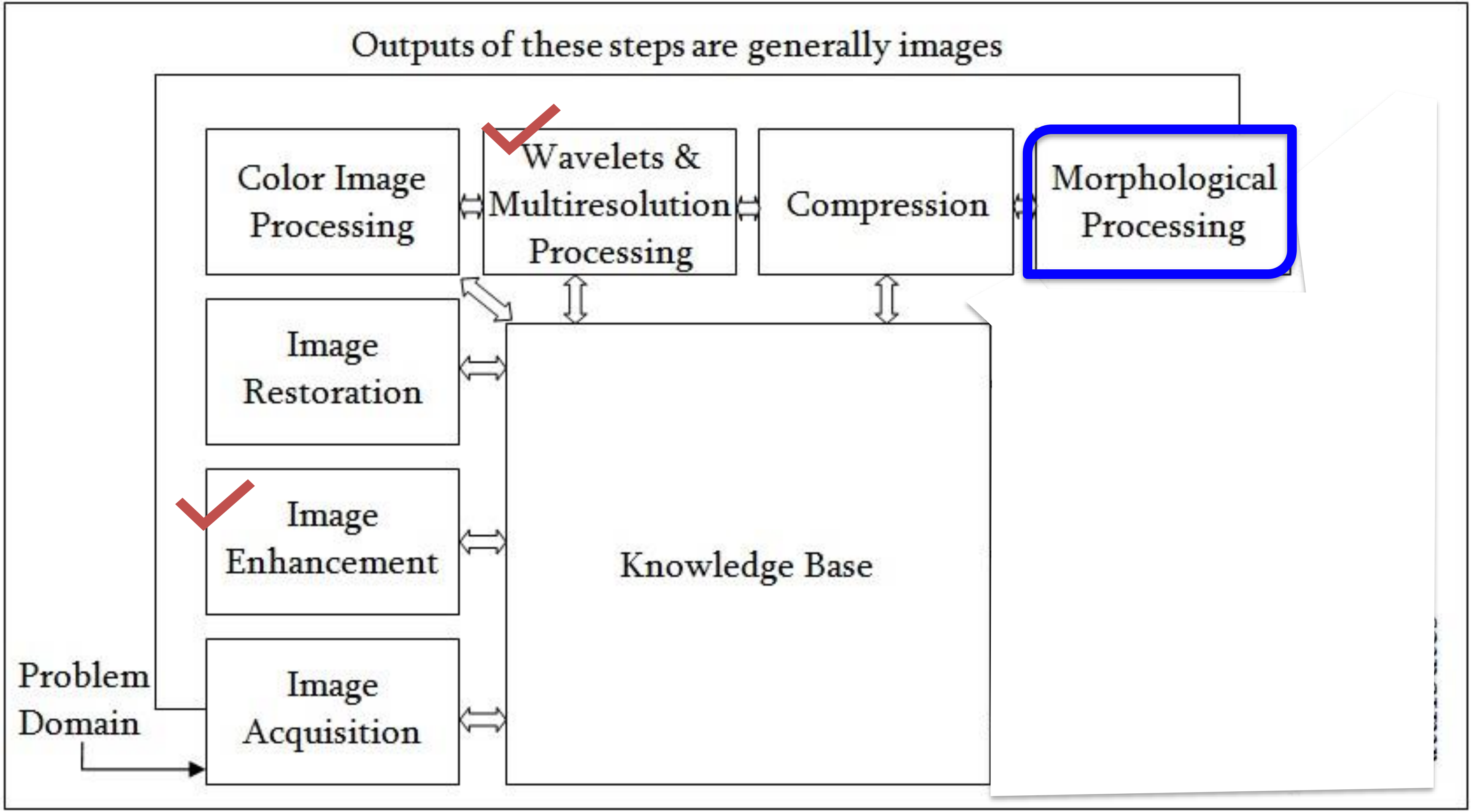
Digital Image Processing (CSE/ECE 478)

Lecture-11: Morphological Operations

Ravi Kiran

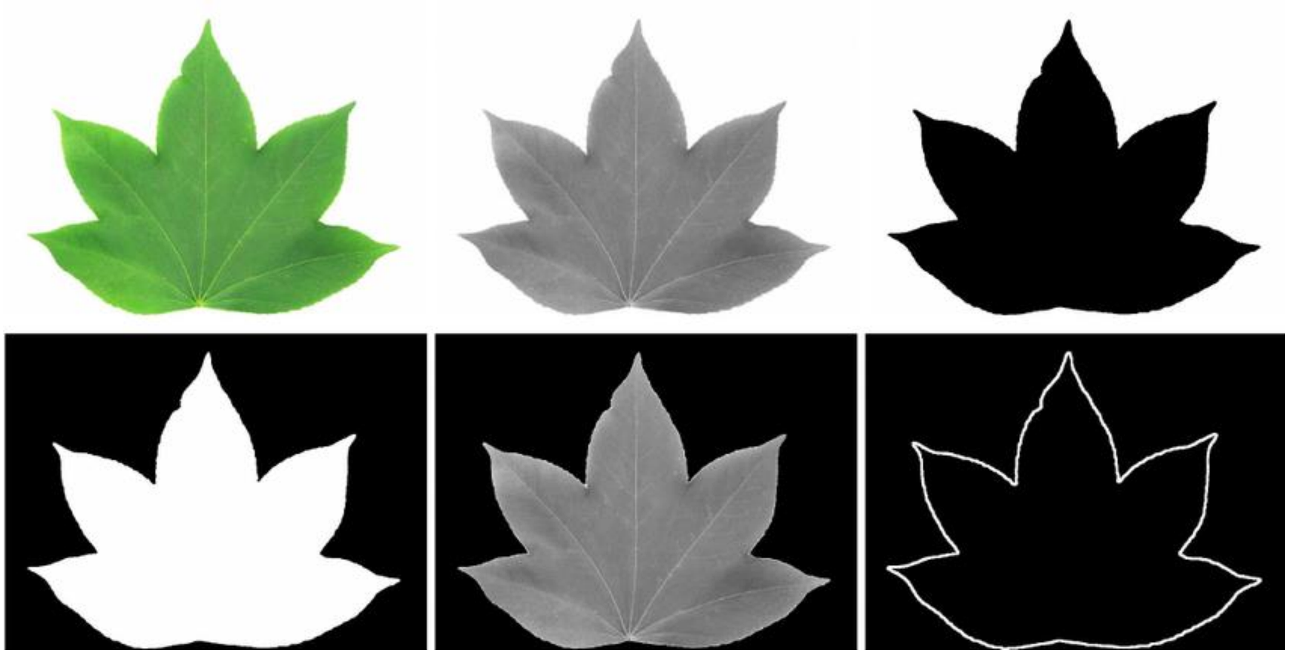
Center for Visual Information Technology (CVIT), IIIT Hyderabad

Slide credits: Volker Krüger, Rune Andersen, Roger S. Gaborski

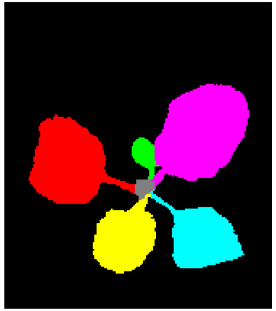

Outputs of these steps are generally images
Color Image Processing
Wavelets & Multiresolution Processing
Compression
Morphological Processing
Image Restoration
Image Enhancement
Knowledge Base
Problem Domain
Image Acquisition

Binary Images

Plant Phenotyping



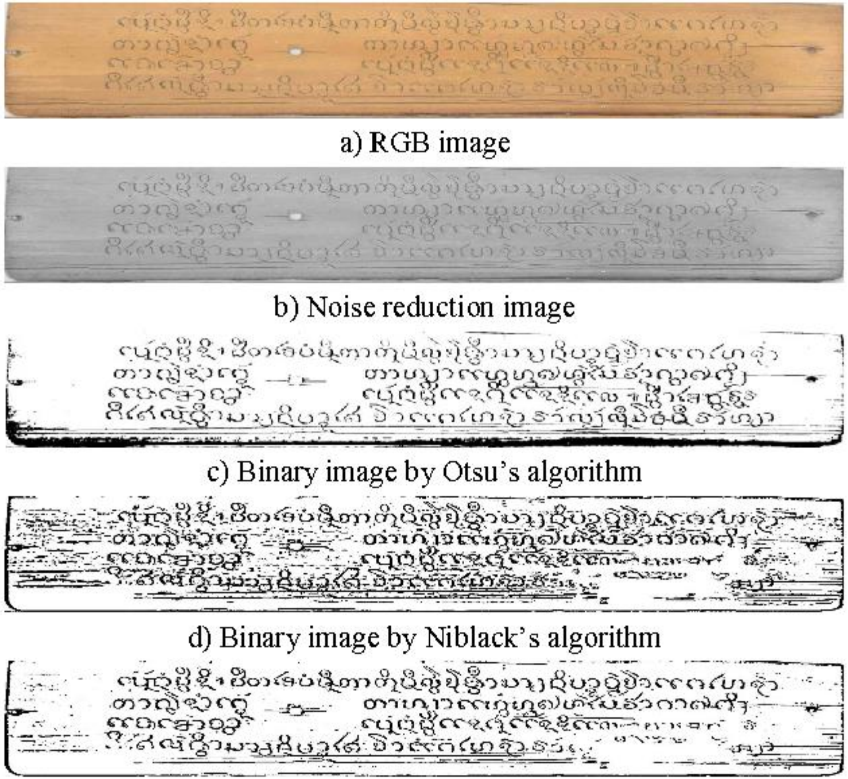
Plant Phenotyping



Recognizing Scene Text

https://cvit.iiit.ac.in/research/projects/cvit-projects/scene-text-understanding

Document Image Analysis



a) RGB image

b) Noise reduction image

c) Binary image by Otsu's algorithm

d) Binary image by Niblack's algorithm

e) Binary image by Sauvola's algorithm

Figure 2. Samples of palm leaf images

Background Subtraction

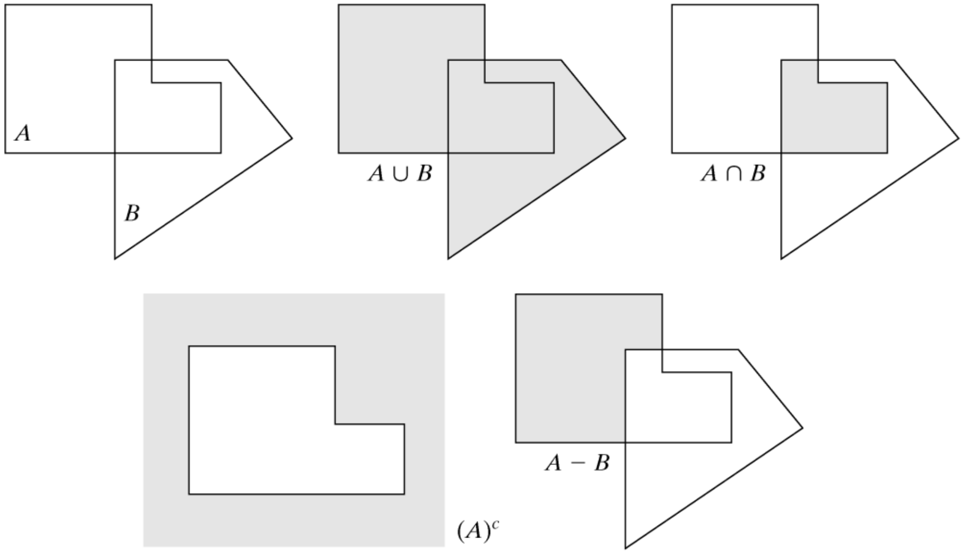
Introduction to Morphological Operators

Image – Set of Pixels

- Basic idea:
  - Object/Region = set of pixels (or coordinates of pixels)

- 0 = background
- 1 = foreground

Object = set of pixels (or coordinates of pixels)



a b c
d e

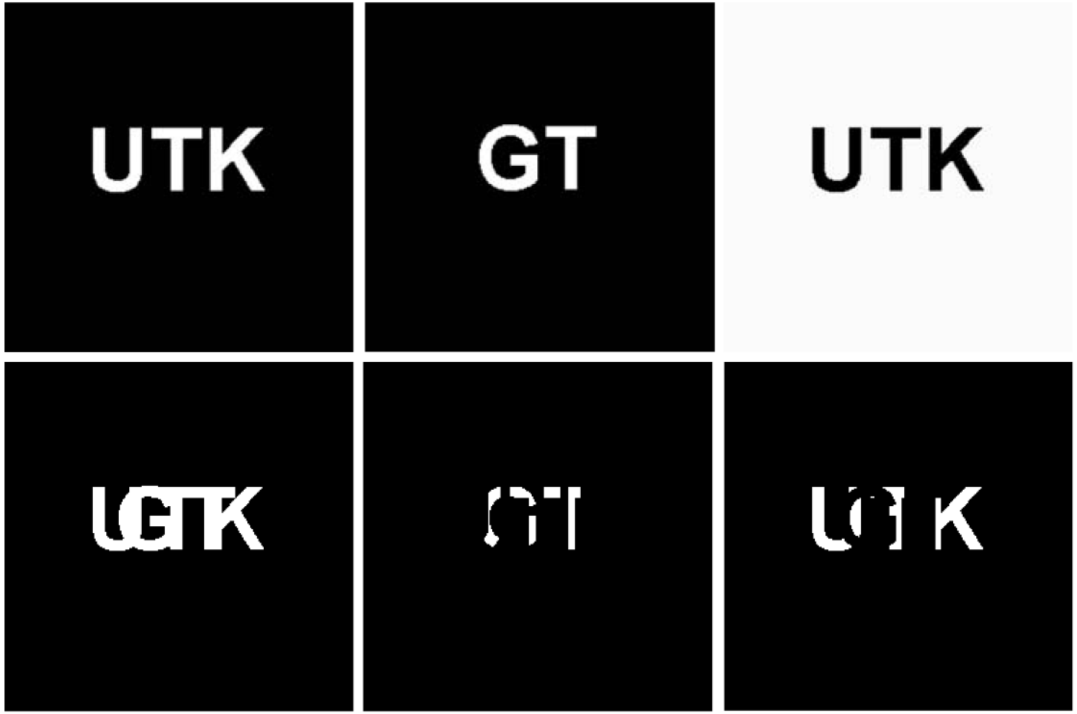
FIGURE 9.1
(a) Two sets A and B. (b) The union of A and B. (c) The intersection of A and B. (d) The complement of A. (e) The difference between A and B.

From: Digital Image Processing, Gonzalez, Woods And Eddins

Basic operations on shapes

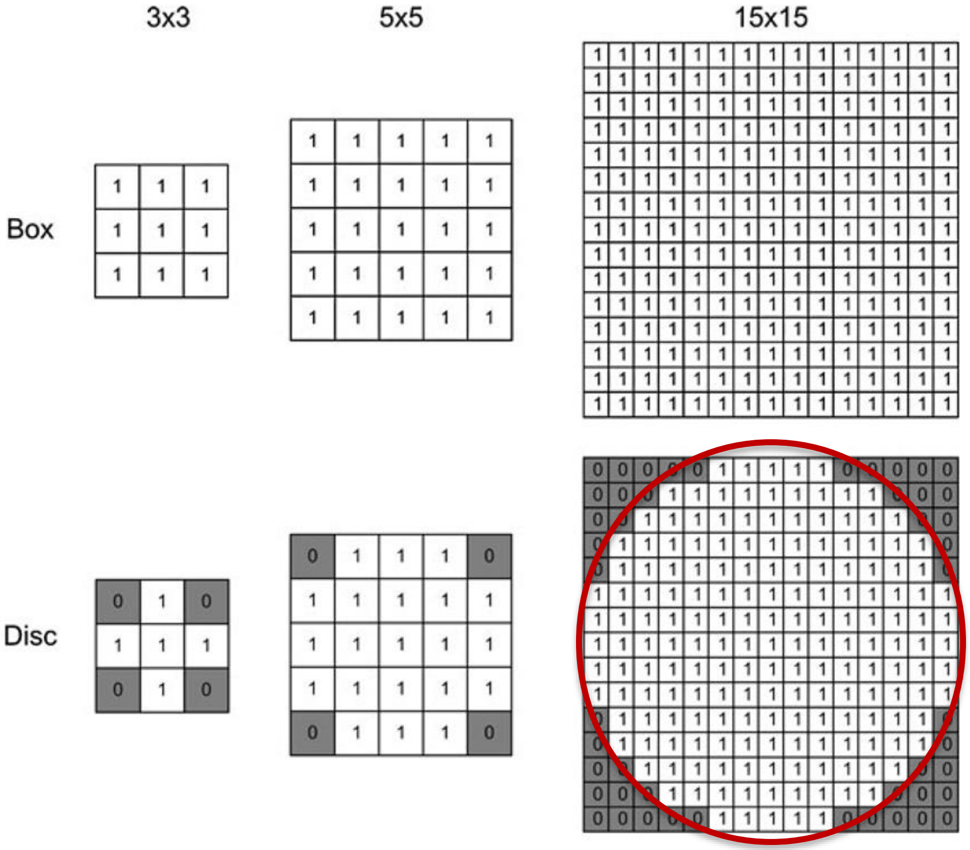
Set Operations on Binary Images

A B



From: Digital Image Processing, Gonzalez,Woods And Eddins

Structuring Element



Structuring Element (Kernel)

- Can have varying sizes
- Have an origin
- Usually, element values are 0,1 and none(!)
  - 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

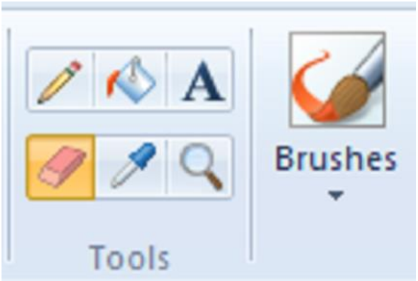
Disc

	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		

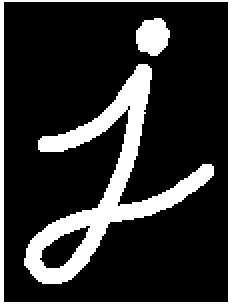
1	1	
1	0	
1		0

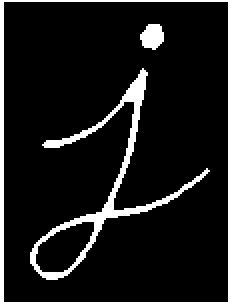
1	1	1
1	10	1
1	1	1



Erosion

Erosion





Erosion : Effect

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) }

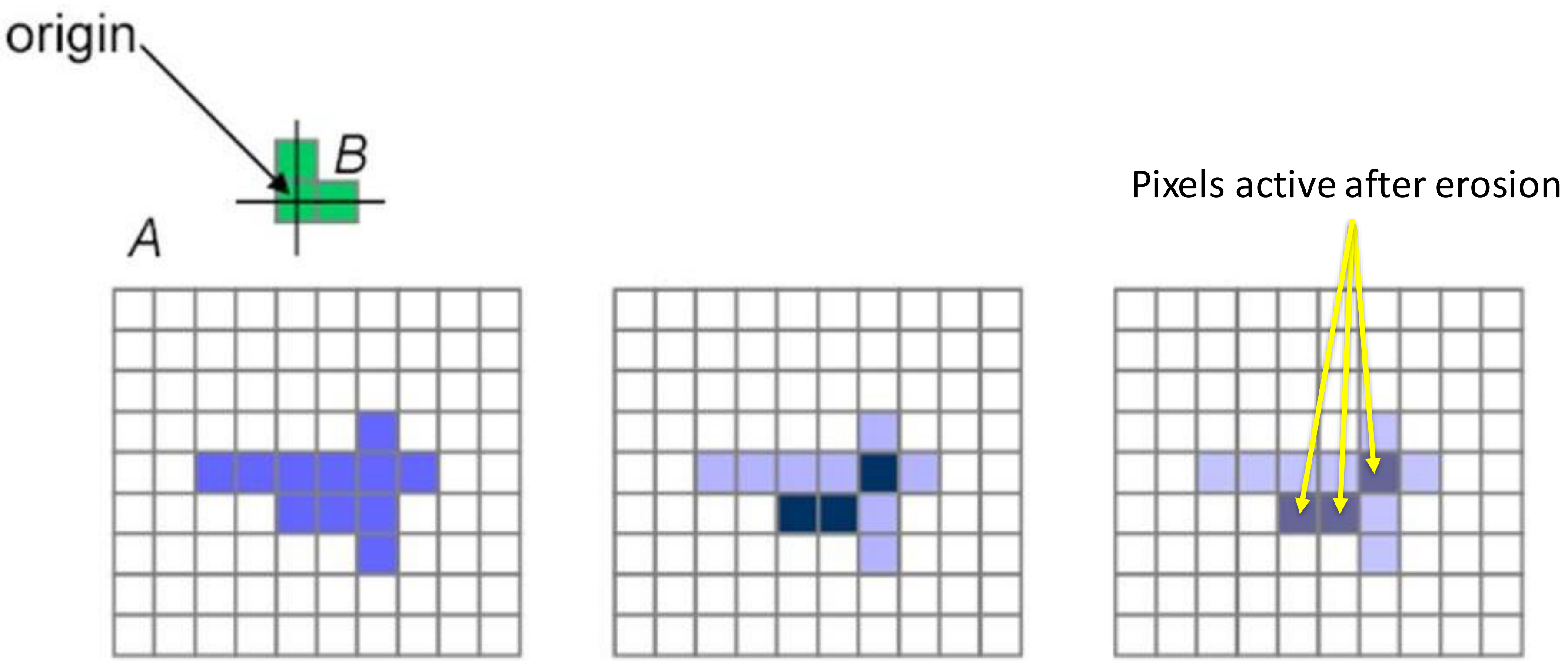
If, for a particular location of Structuring Element (SE) origin, SE lies **fully within the region**, retain the location, else set to 0

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

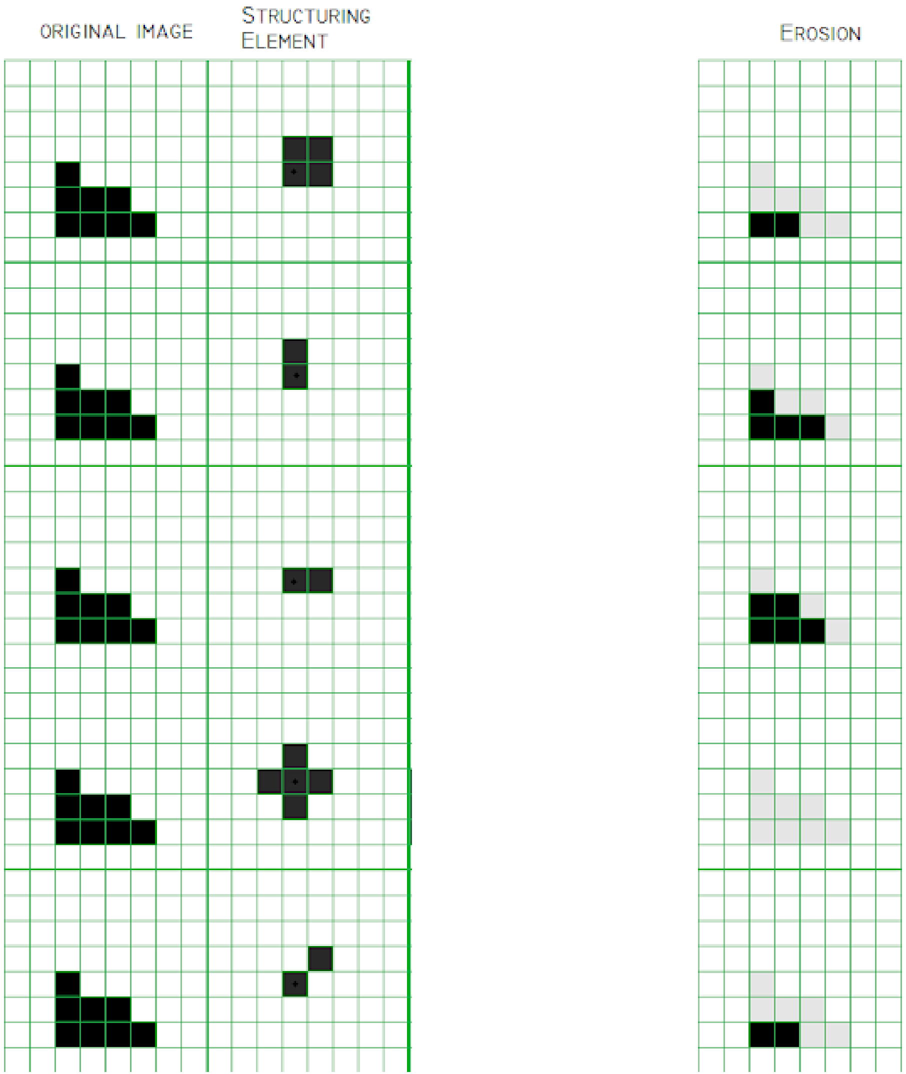
$\Longrightarrow$

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

SEs operate wrt an origin

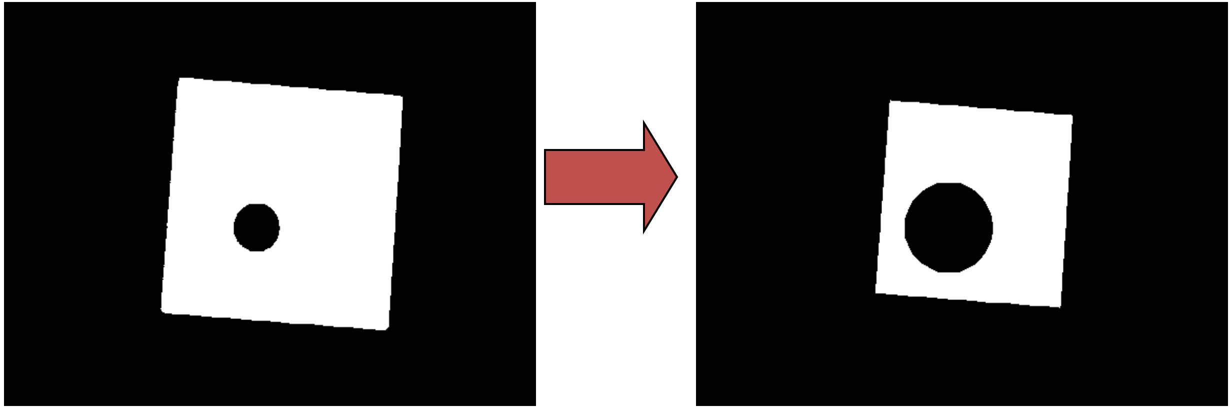


Lucas J van Vliet: ASCI a11: 8



ORIGINAL IMAGE
STRUCTURING ELEMENT
EROSION

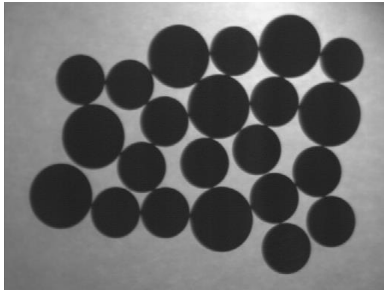
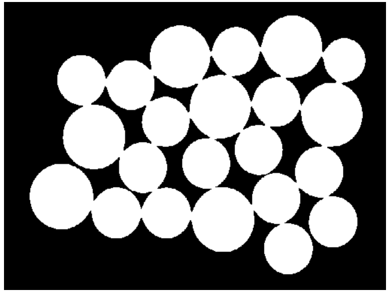
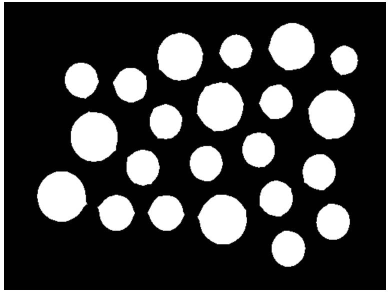
Another example of erosion



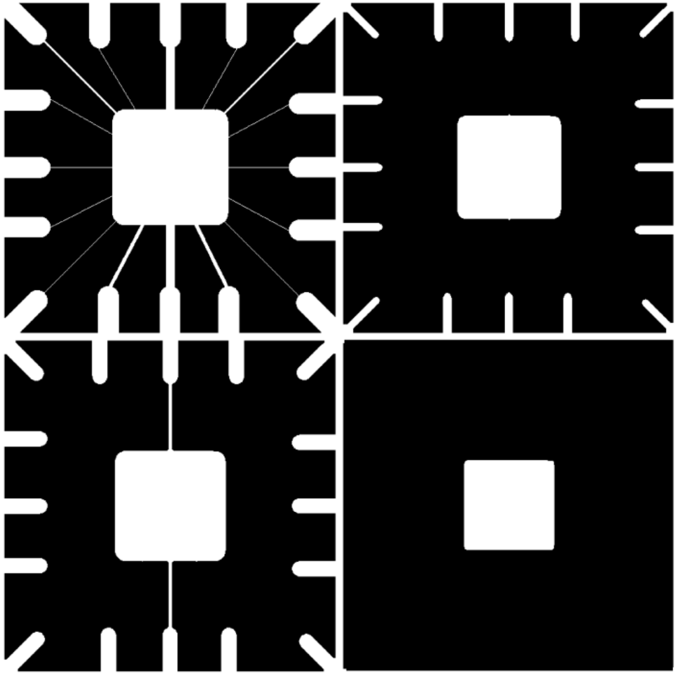
Erosion ➔ Image gets darker

Example: Counting coins

- Difficult because they touch each other!
- Solution: Binarization and Erosion separates them!

Erosion - example



a b
c d

FIGURE 9.8 An illustration of erosion. (a) Original image. (b) Erosion with a disk of radius 10. (c) Erosion with a disk of radius 5. (d) Erosion with a disk of radius 20.

From: Digital Image Processing, Gonzalez,Woods And Eddins

Erosion : Operation (min filter)

1	1	1
1	1	1
1	1	1

Set of coordinate points =

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

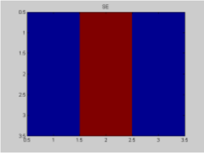
(-1, 0), (0, 0), (1, 0),

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

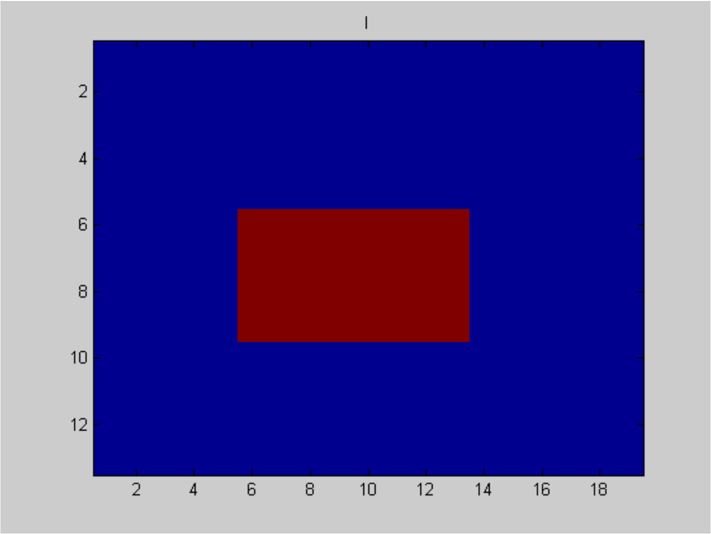
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	0	0	0	0	1	1	1	0	0
0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0
0	0	0	1	1	0	0	0	0	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0
0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0
0	0	0	0	0	0	1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

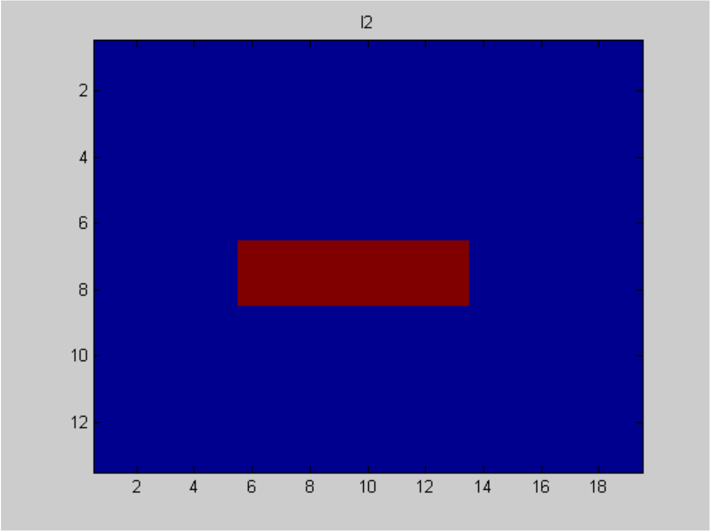
MATLAB code



SE = 3x3



I2



I3=imerode(I2,SE);

Erosion

- Simple application of **pattern matching**

1	1	1
1	1	1
1	1	1

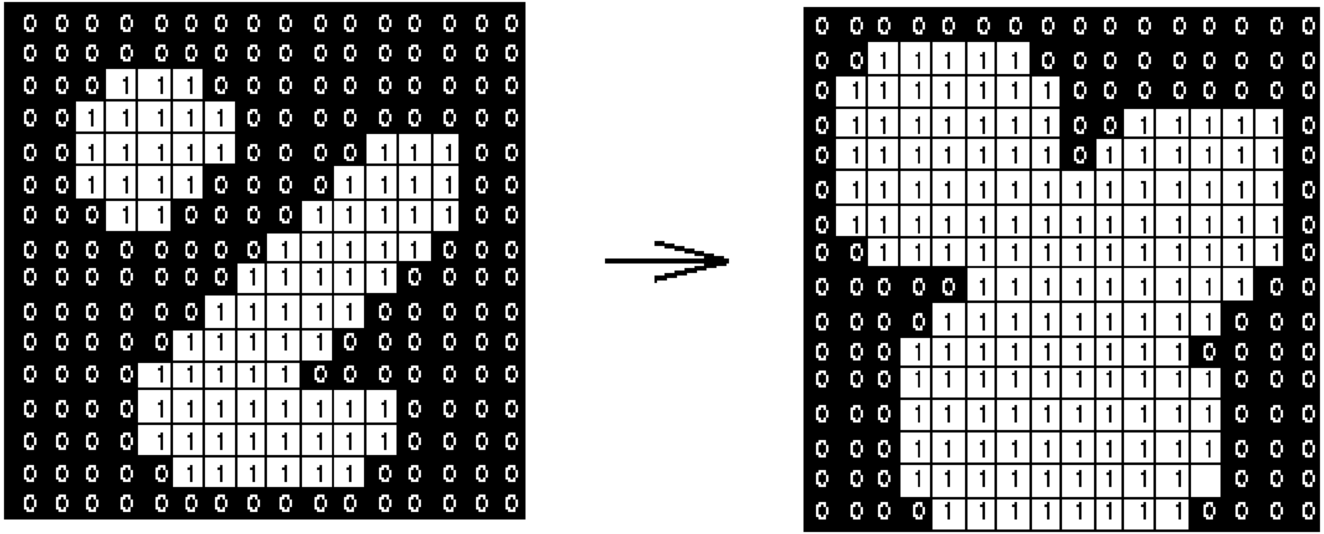
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	0	0	0	0	1	1	1	0	0
0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0
0	0	0	1	1	0	0	0	0	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0
0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0
0	0	0	0	0	0	1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

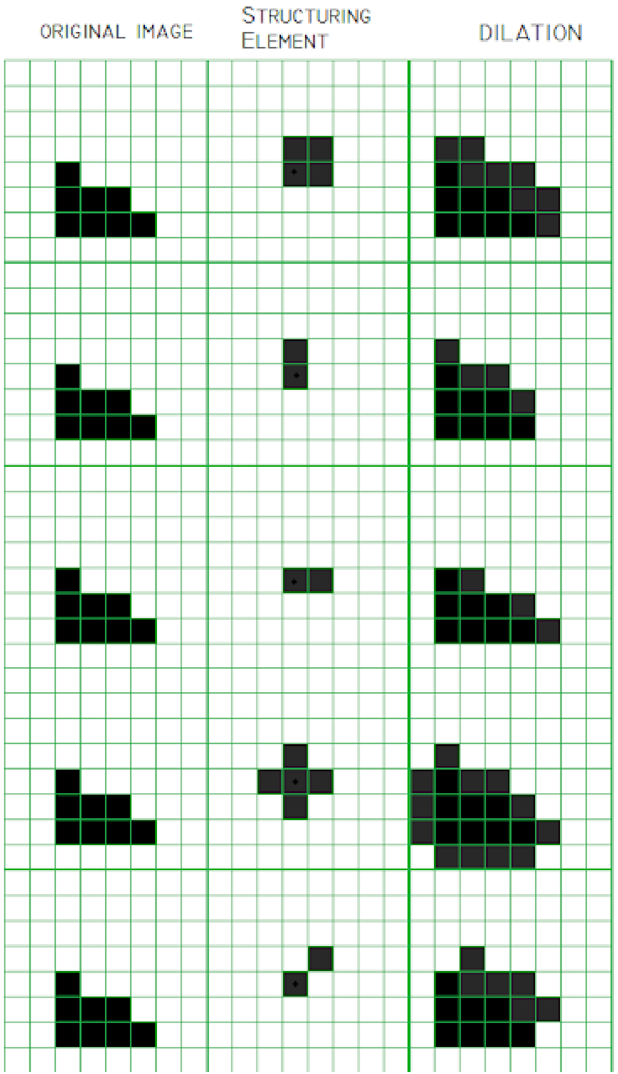
$\Longrightarrow$

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

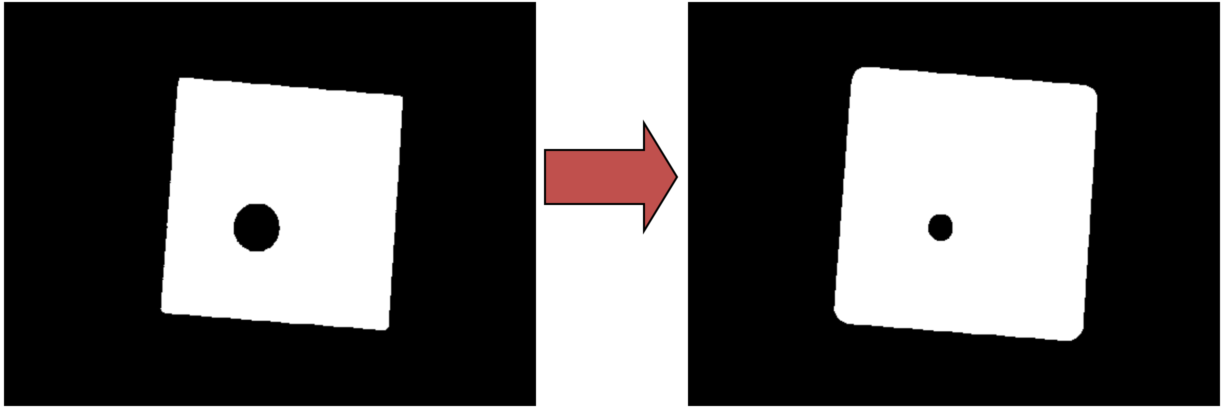
Dilation

1	1	1
1	1	1
1	1	1




ORIGINAL IMAGE
STRUCTURING ELEMENT
DILATION

Dilation Example



- Image gets lighter, more uniform intensity
- NOTE-1: SE = disk
- NOTE-2: Multiple iterations of dilation

Dilation (max filter)

1	1	1
1	1	1
1	1	1

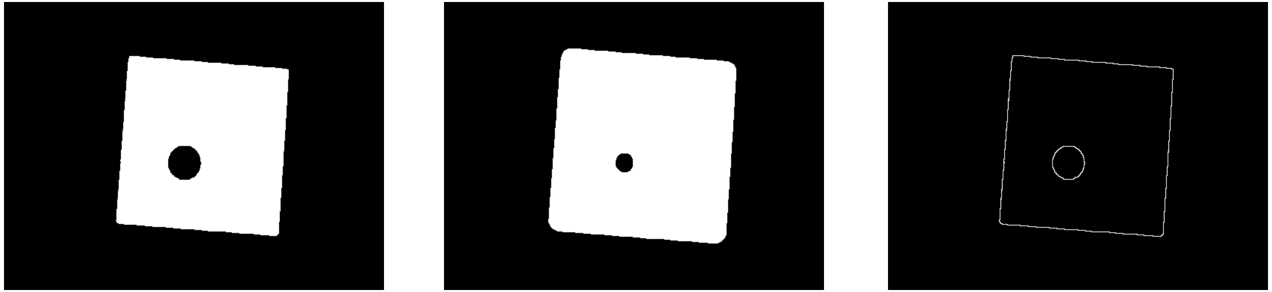
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	0	0	0	0	1	1	1	0	0
0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0
0	0	0	1	1	0	0	0	0	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0
0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0
0	0	0	0	0	0	1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

$\Rightarrow$

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

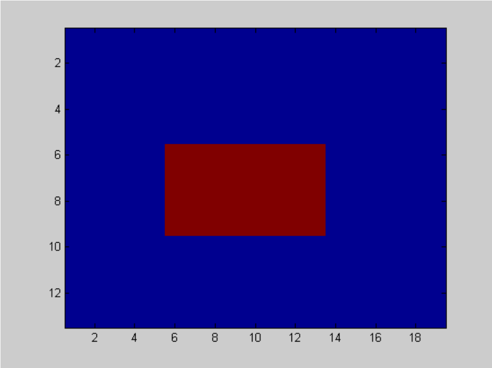
Boundary Detection

1. Dilate input image
2. Subtract input image from dilated image
3. Boundaries remain!

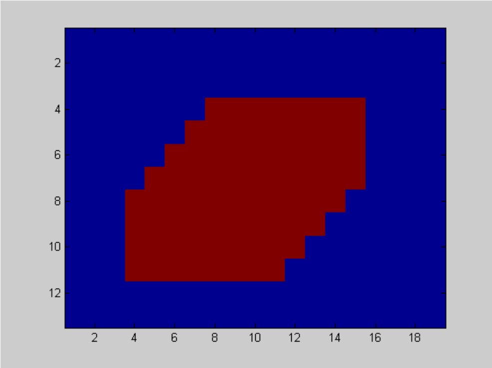


Can use erosion also ..

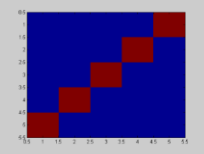
Fig 3: (a) Original Image (linkon.tif) (B) After erosion operation (C) Boundary Extraction with the help of Erosion.



I



I2



SE

```
>> I(6:9,6:13)=1;
>> figure, imagesc(I)
>> I2=imdilate(I,SE);
>> figure, imagesc(I2)
```

Dilation and Erosion

- DILATION: Adds pixels to the boundary of an object
- EROSION: Removes pixels from the boundary of an object
- Number of pixels added or removed depends on size and shape of structuring element

Opening and Closing

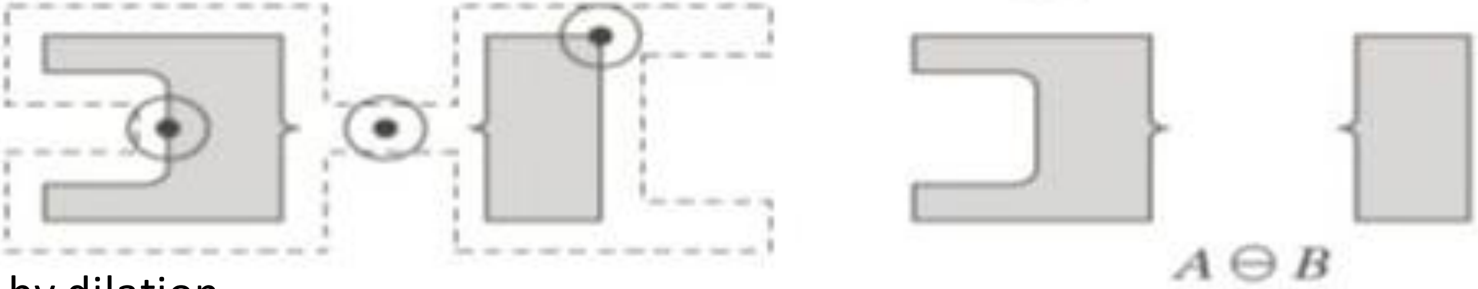
- Important operations
- Derived from the fundamental operations
  - Dilation
  - Erosion

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 can *not* 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

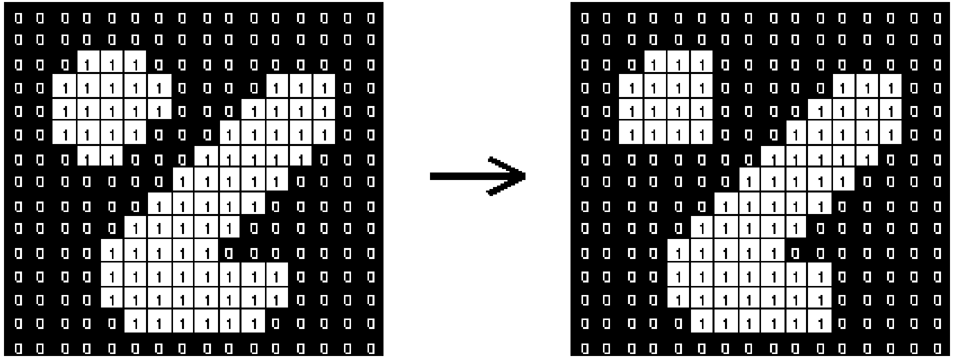
Erosion



Followed by dilation …

Opening

- Structuring element: 3x3 square



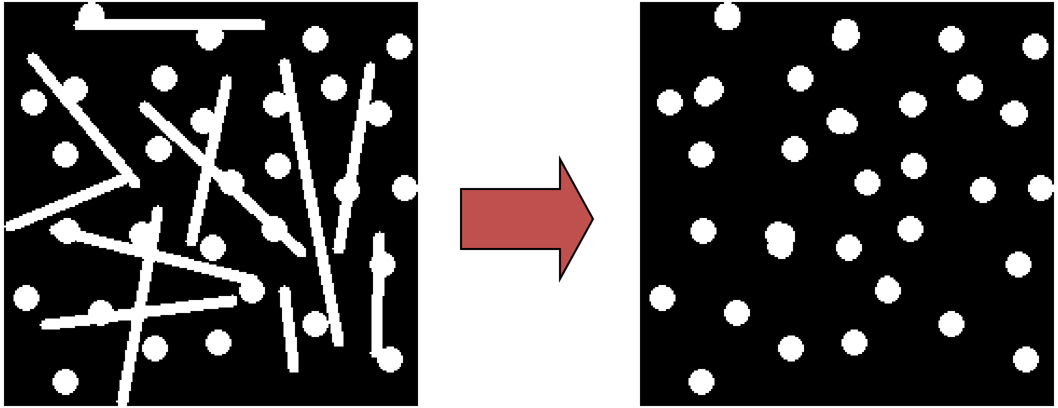
Take the structuring element (SE) and <u>slide it around *inside* (each) foreground region</u>

All foreground pixels which can *not* be reached by the structuring element without lapping over the edge of the foreground object will be eroded away!

All pixels which can be covered by the SE with the SE being entirely within the foreground region will be preserved.

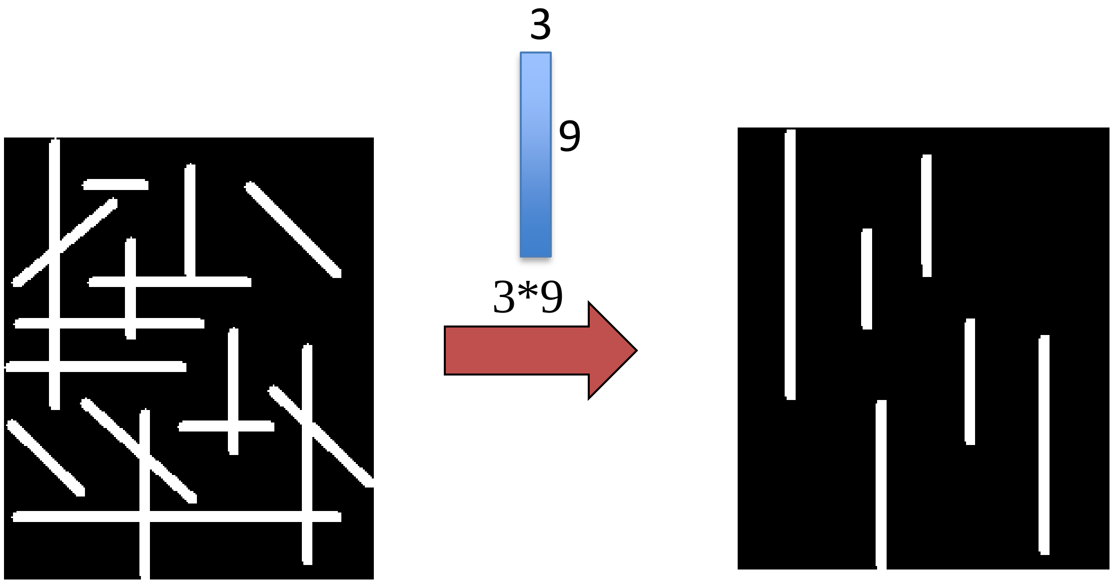
Opening: Example

- Opening with a 11 pixel diameter disc



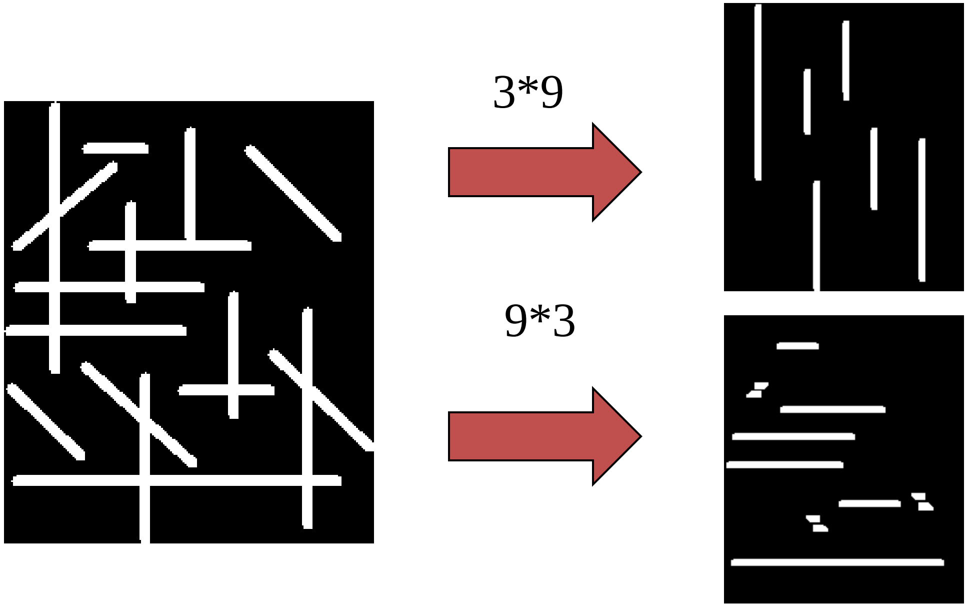
Opening: Another Example

- 3x9 Structuring Element



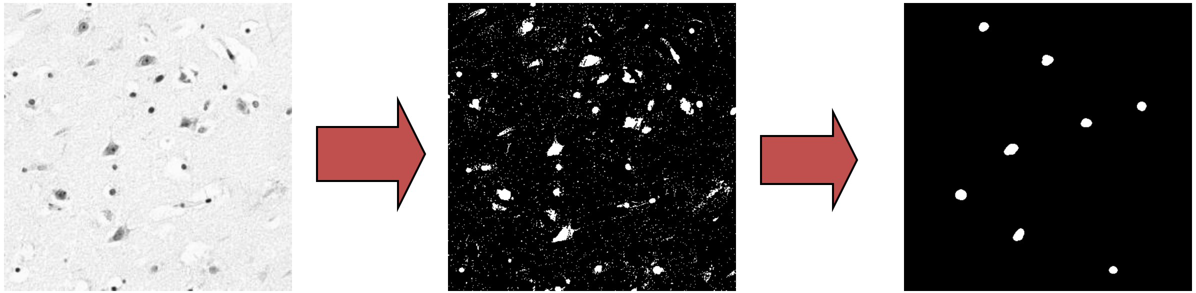
Opening: Another Example

- 3x9 and 9x3 Structuring Element



Use Opening for Separating Blobs

- Use large structuring element that fits into the big blobs
- Structuring Element: 11 pixel disc

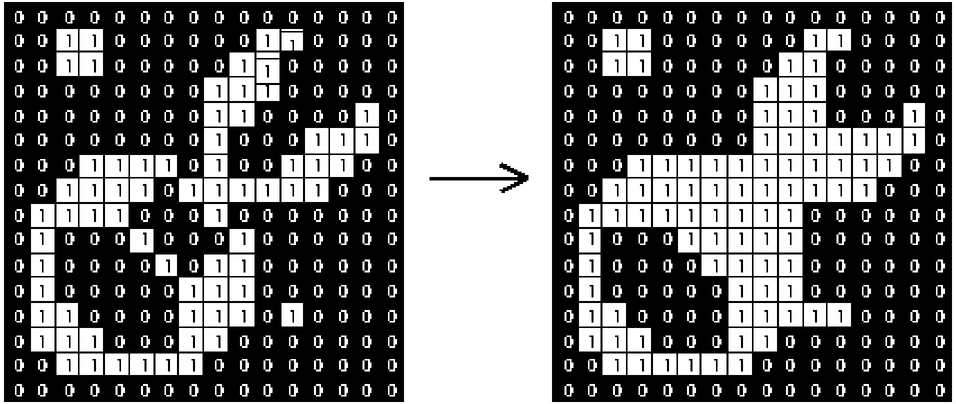


Opening

- Similar to Erosion
  - Spot and noise removal
  - Less destructive
- Erosion next dilation
- the *same structuring element for both operations.*
- Input:
  - Binary Image
  - Structuring Element, containing only 1s!

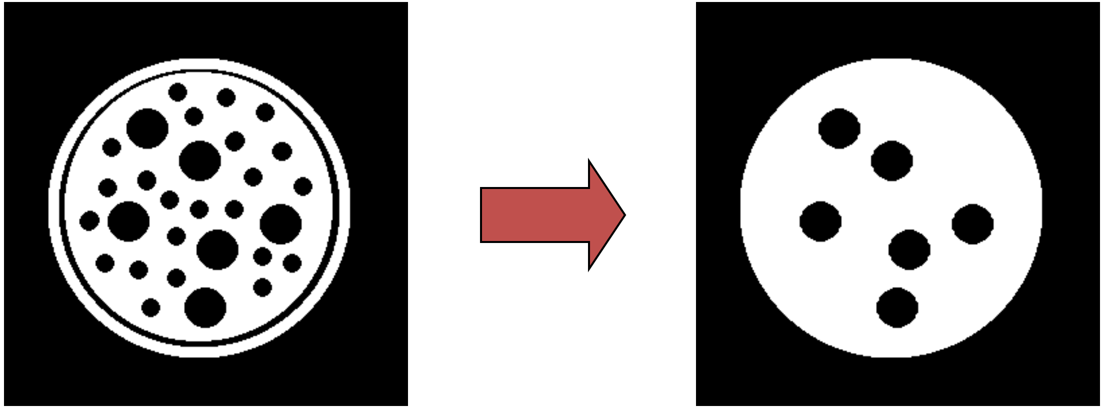
Closing (Dilation then Erosion)

- Structuring element: 3x3 square



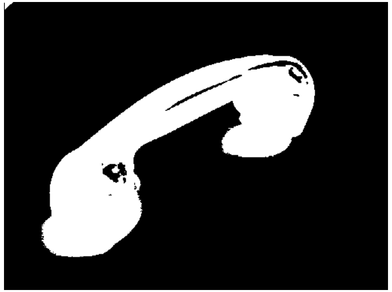
Closing: Example

- Closing operation with a 22 pixel disc
- Closes small holes in the foreground

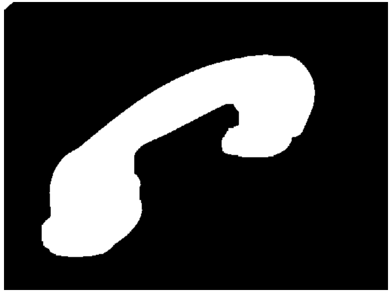


Closing Example 1

1. Threshold
2. Closing with disc of size 20



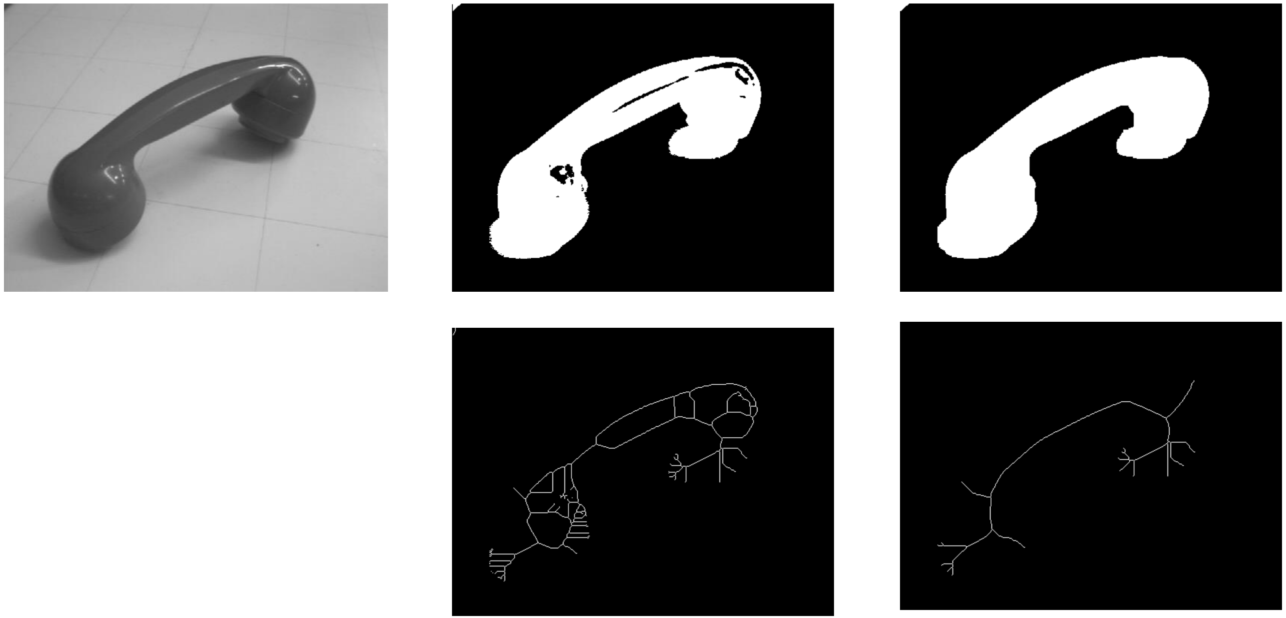
Thresholded



closed

Application of Closing

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



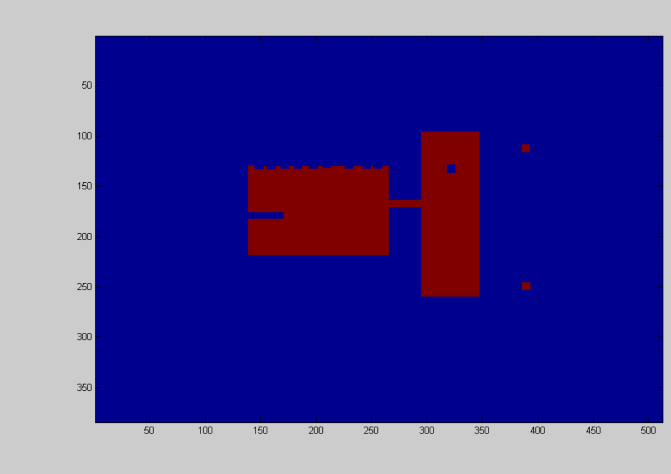
Opening & Closing

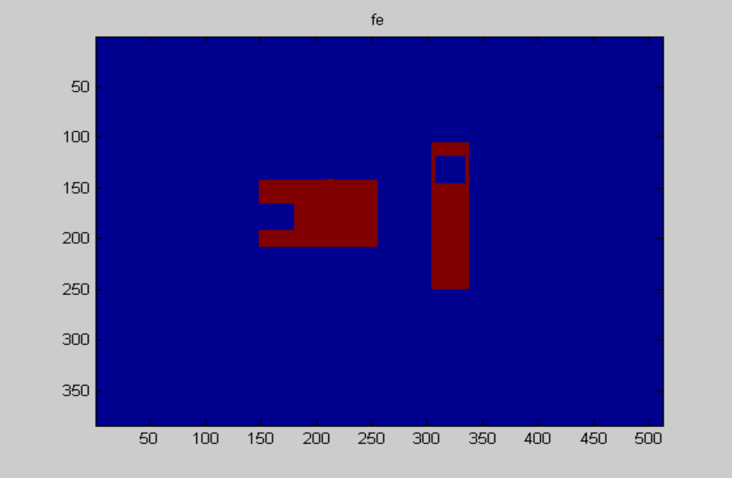
- Opening is the *dual* of closing
- *i.e.* opening the foreground pixels with a particular structuring element
- is equivalent to closing the background pixels with the same element.

• Opening of A by B →A∘B

Erosion of A by B, followed by the dilation of the result by B

Erosion- if any element of structuring element overlaps with background, output is zero



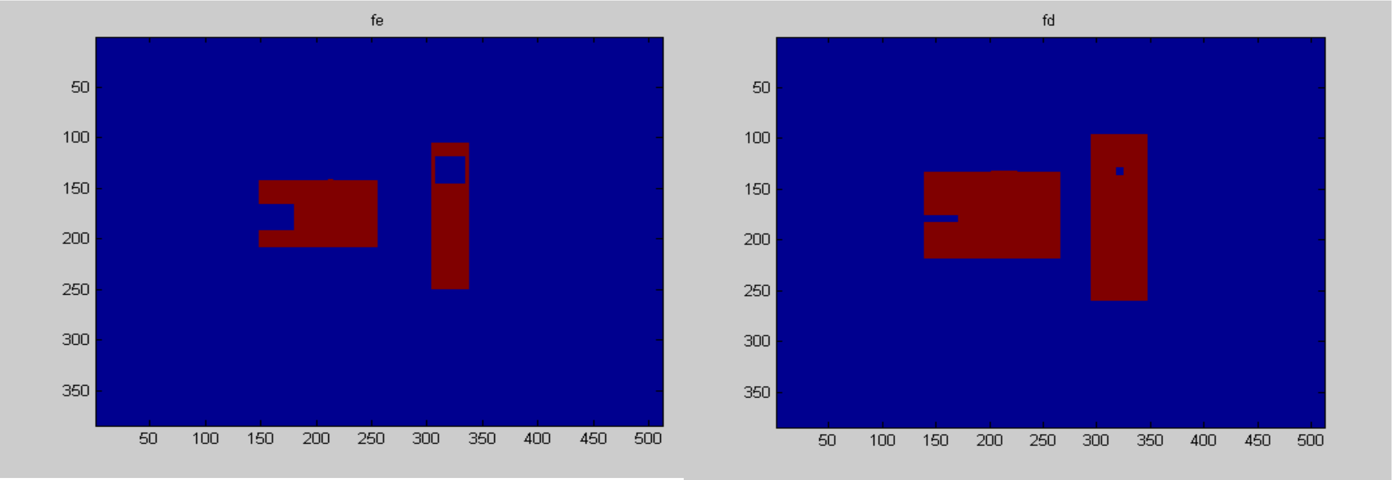


FIRST - EROSION

```
>> se = strel('square', 20);fe = imerode(f,se);figure, imagesc(fe),title('fe')
```

Dilation of Previous Result

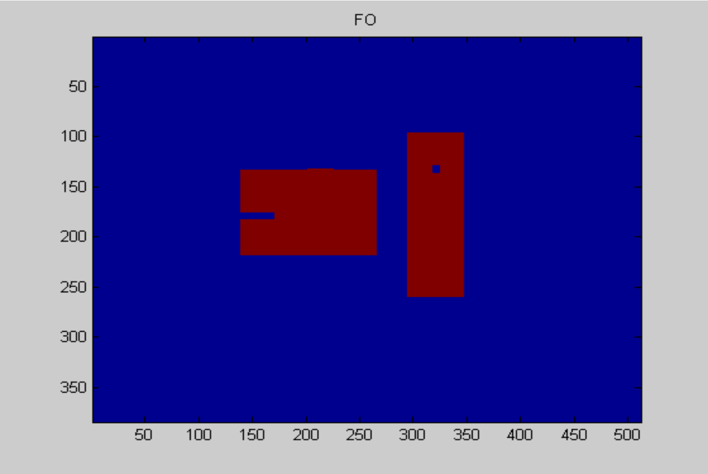
Outputs 1 at center of SE when at least one element of SE overlaps object



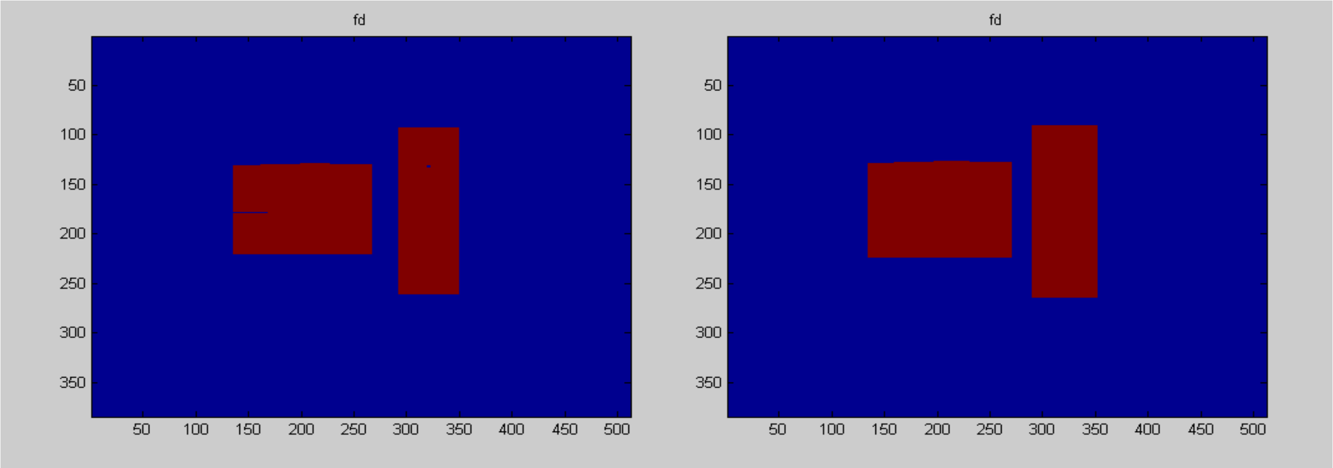
SECOND - DILATION

```
>> se = strel('square', 20);fd = imdilate(fe,se);figure, imagesc(fd),title('fd')
```

FO=imopen(f,se); figure, imagesc(FO),title('FO')



What if we increased size of SE for DILATION operation??



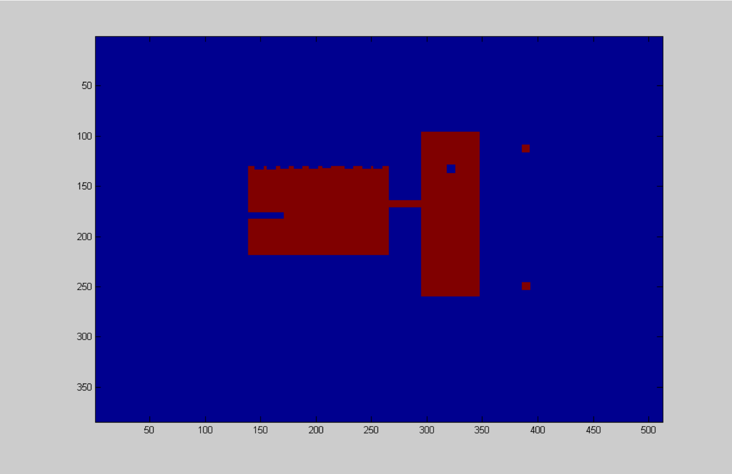
se = 25 se = 30

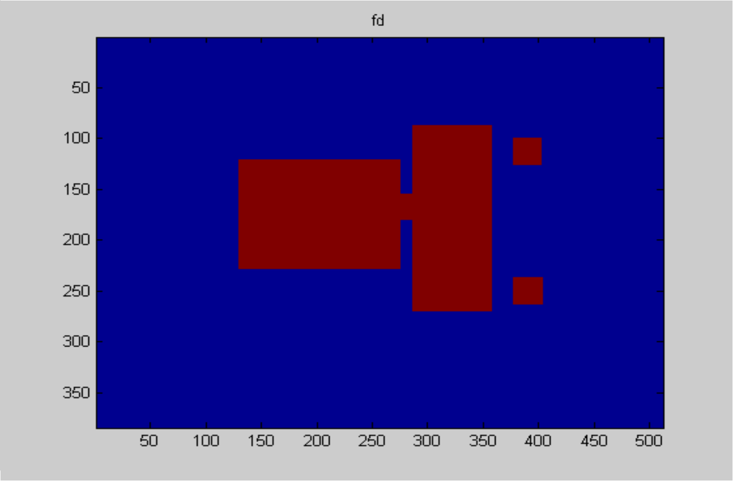
se = strel('square', 25);fd = imdilate(fe,se);figure, imagesc(fd),title('fd')
se = strel('square', 30);fd = imdilate(fe,se);figure, imagesc(fd),title('fd')

Closing of A by B → A• B

Dilation of A by B

Outputs 1 at center of SE when at least one element of SE overlaps object

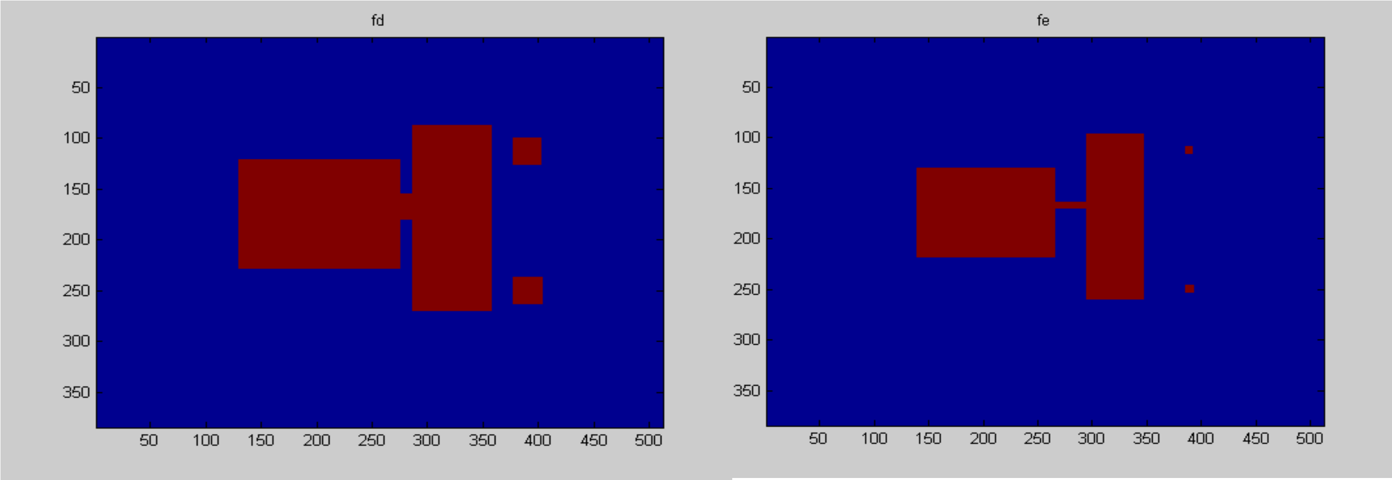


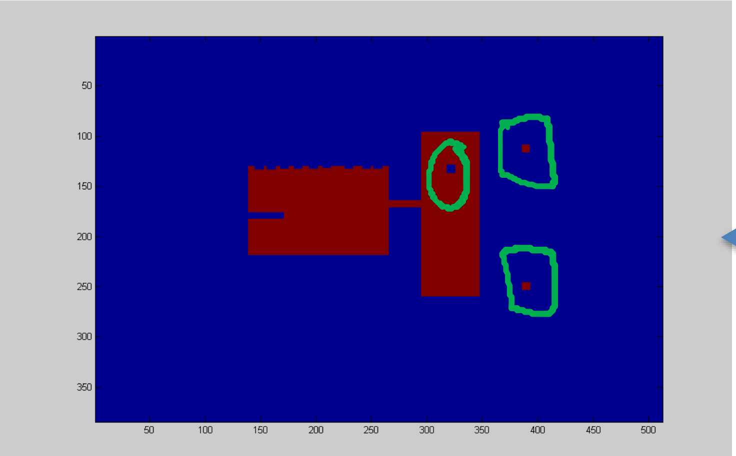



```
se = strel('square', 20);fd = imdilate(f,se);figure, imagesc(fd),title('fd')
```

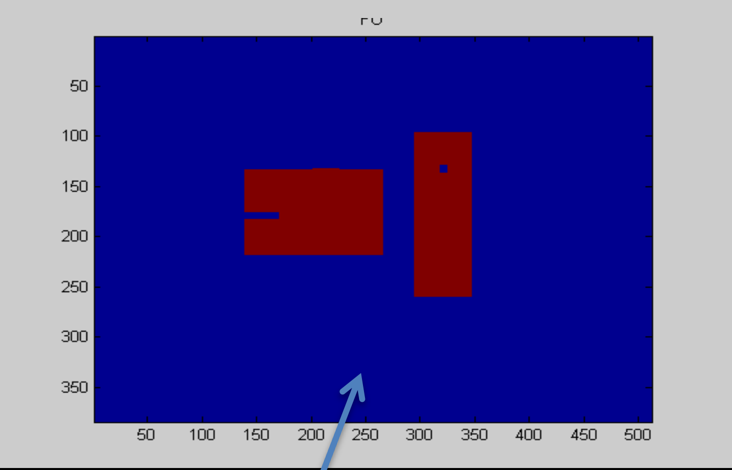
Erosion of the result by B

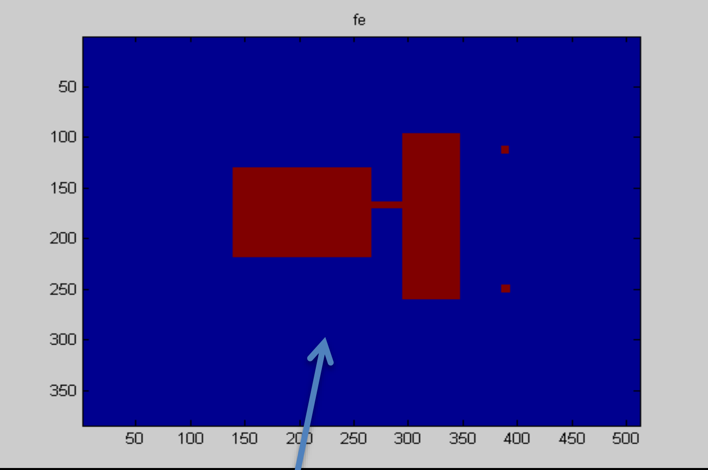
Erosion- if any element of structuring element overlaps with background output is zero





ORIGINAL





OPENING

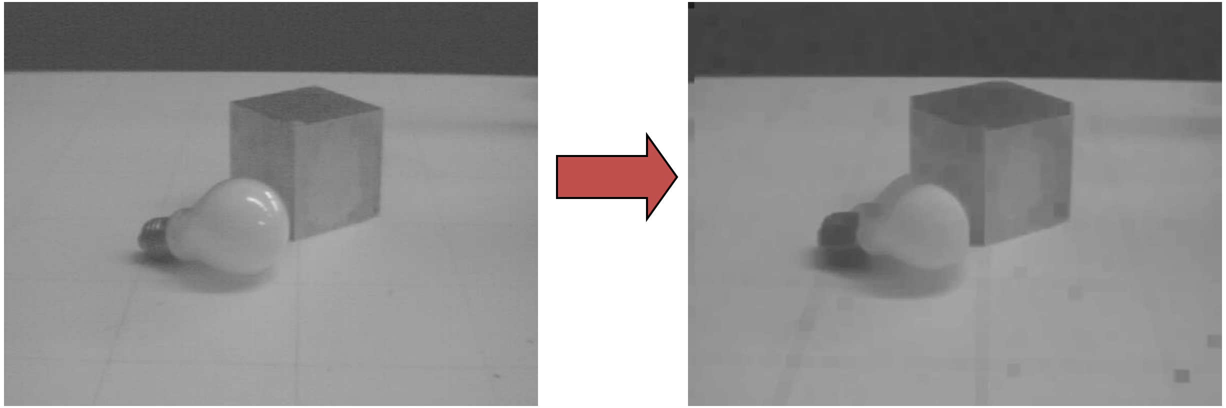
CLOSING

Fingerprint problem

FIGURE 9.11 (a) Noisy fingerprint image. (b) Opening of image. (c) Opening followed by closing. (Original image courtesy of the National Institute of Standards and Technology.)

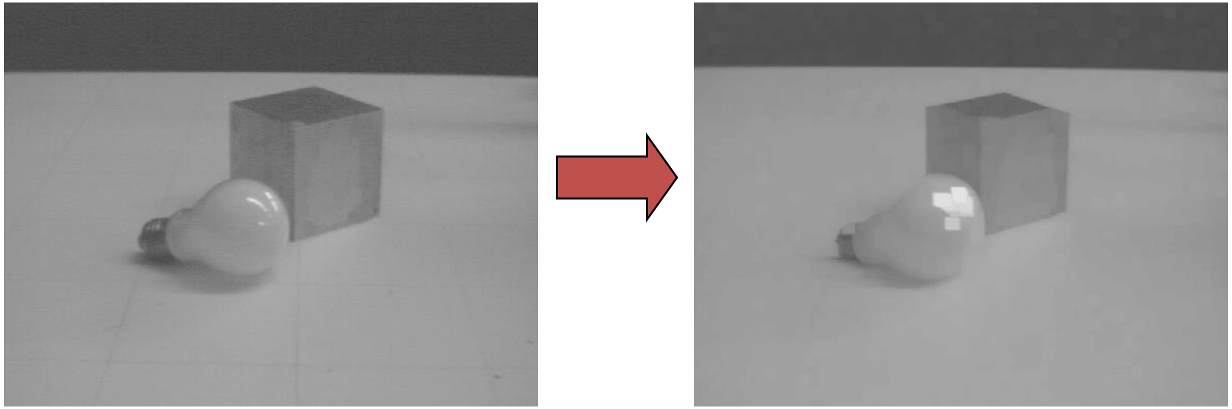
From: Digital Image Processing, Gonzalez, Woods And Eddins

Erosion on Gray Value Images



- min filter
- Images get darker!

Dilation on Gray Value Images



- max filter
- More uniform intensity

References

- G&W, 3rd Ed., 9.1-9.3, 9.6
- https://in.mathworks.com/help/images/morphological-dilation-and-erosion.html
- https://scikit-image.org/docs/dev/auto_examples/applications/plot_morphology.html#sphx-glr-auto-examples-applications-plot-morphology-py