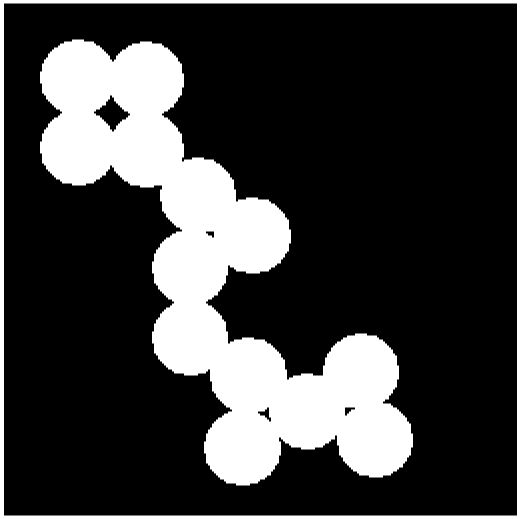
Binary Image Processing

Binary Images

- "Binary" means
  - 0 or 1 values only
  - Also called "logical" type (true/false)
- Obtained from
  - Thresholding gray level images
  - Result of feature detectors
- Often want to count or measure shape of 2D binary image regions
- Typical applications
  - Objects on a conveyor belt
  - Characters and text, maps
  - Chromosomes
  - Fingerprints
  - Circuit boards
  - Overhead aerial images



Outline

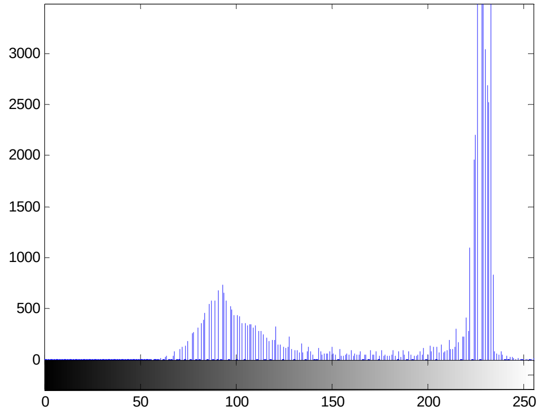
- Thresholding
- Logic operations
- Connected components
- Morphological operators
- Region properties, moments
- Lab – example to find fiducial target

Thresholding

- Convert gray scale image to binary (0s and 1s)
- Simplifies processing and computation of features
- Can use a single threshold value (global) or a local value (adaptive)

Thresholding in Matlab:

```
B = I > t;
```



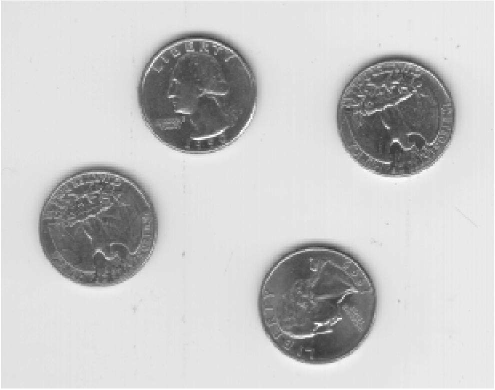
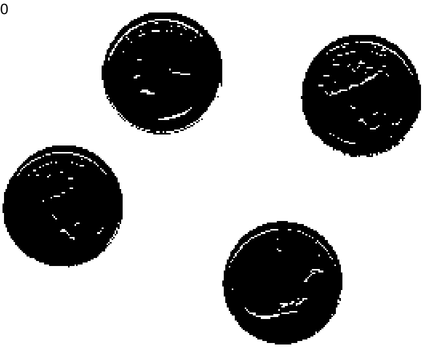


Image "eight.tif"

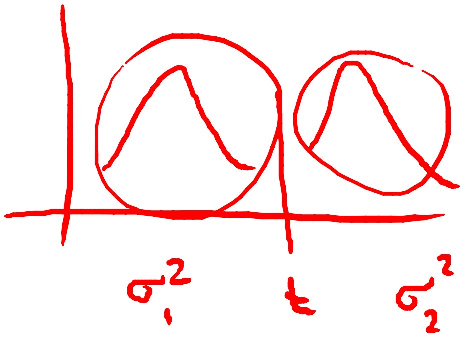


Global Threshold Computation

- Assume histogram is bimodal
- Choose threshold value to separate modes, and minimize classification error
- Otsu algorithm: minimize the within-group variances

VARIANCE: $\sigma^2 = \frac{1}{N}\sum (x_i - \bar{x})^2$ $\bar{x} = \frac{1}{N}\sum x_i$

$$\sigma_w^2 = N_1\sigma_1^2 + N_2\sigma_2^2$$



Matlab Examples

- Images
  - cameraman.tif, eight.tif, coins.png
- Functions
  - `t = graythresh(I)  % Otsu algorithm`
  - `BW = im2bw(I,t);   % performs thresholding`

Logic Operations

- Recall truth tables for logic operators

AND

x	y	z
0	0	0
0	1	0
1	0	0
1	1	1

OR

x	y	z
0	0	0
0	1	1
1	0	1
1	1	1

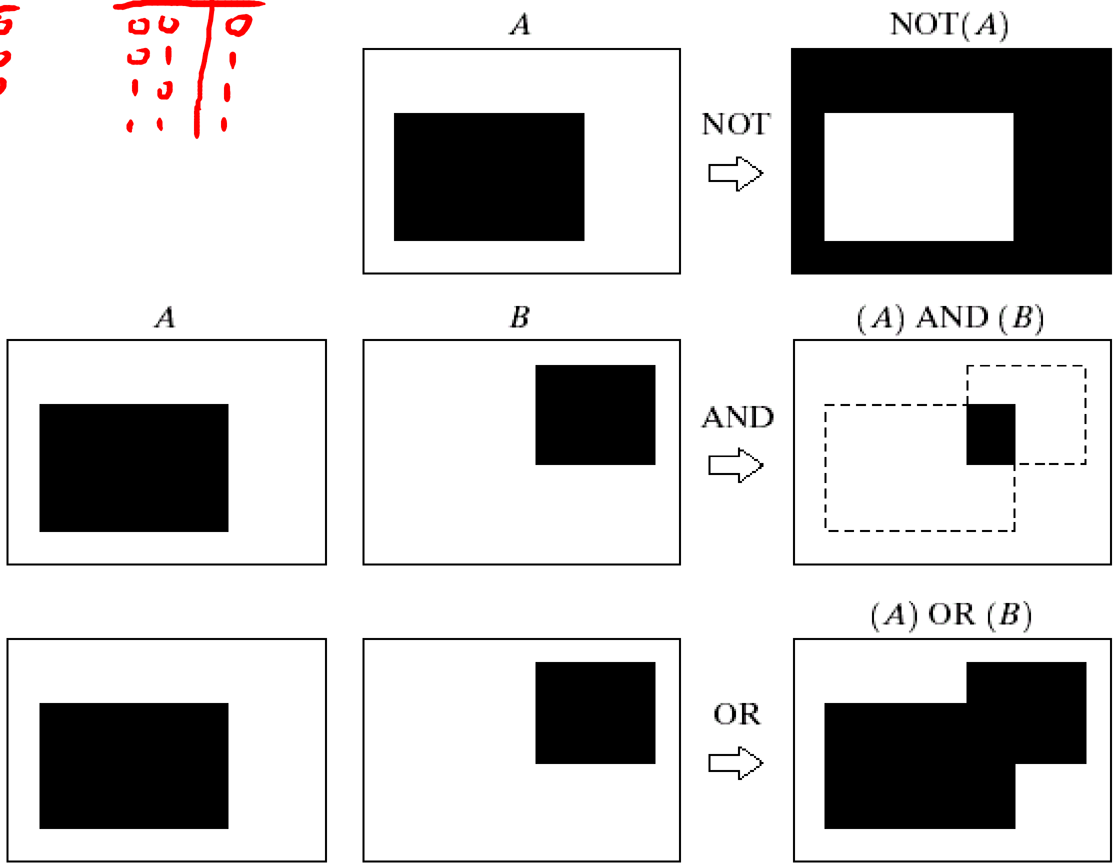
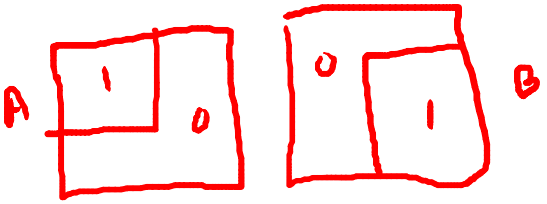


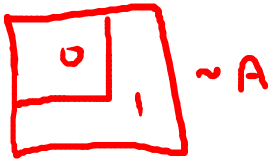
FIGURE 9.3 Some logic operations between binary images. Black represents binary 1s and white binary 0s in this example.

Also XOR, NAND, NOR

Matlab

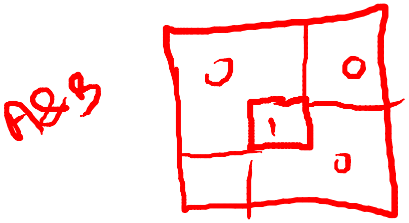


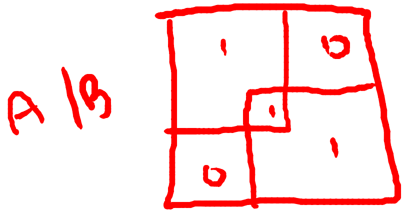
- Create binary images
  - A=false(10,10), A(1:6,1:6)=true;
  - B=false(10,10), B(4:10,4:10)=true;
- NOT
  - ~A
  - imcomplement(A)   % works on gray level too



Does a photo negative: 255-A

- AND
  - A&B;
- OR
  - C = A|B;





- Can also do bitwise AND, OR, etc of unsigned integers
  - A=uint8(zeros(10,10)), A(1:6,1:6)=30;
  - B=uint8(zeros(10,10)), B(4:10,4:10)=120;
  - bitand(A,B);

30   0 0 0 1   1 1 1 0

128 64 32 16   8 4 2 1

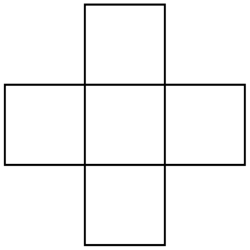
120   0 1 1 1   1 0 0 0

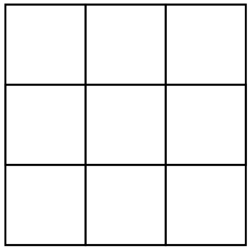
0 0 0 1 1 0 0 0

$= 24_{10}$

Connected Components

- Define adjacency
  - 4-adjacent
  - 8-adjacent

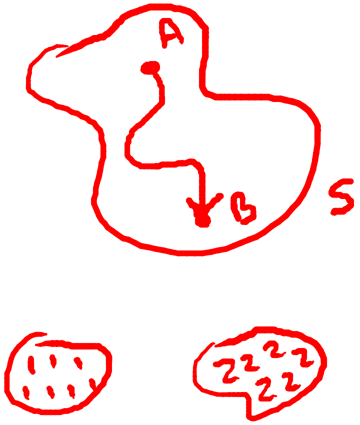




- Two pixels are connected in S if there is a path between them consisting entirely of pixels in S

- S is a (4- or 8-) connected component ("blob") if there exists a path between every pair of pixels

- "Labeling" is the process of assigning the same label number to every pixel in a connected component



Example

- Hand label simple binary image

	1	1				
	1	1				
			1	1		
	1		1	1		
	1		1			1

Binary image

	1	1				
	1	1				
			2	2		
	3		2	2		
	3		2			4

Labeled image (4-connected)

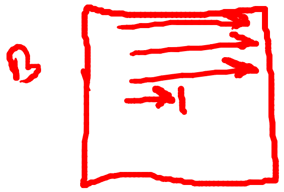
	1	1				
	1	1				
			1	1		
	2		1	1		
	2		1			3

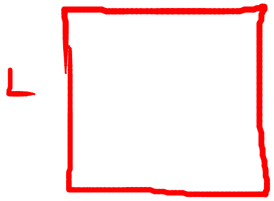
Labeled image (8-connected)

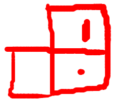
A Fast Labeling Algorithm

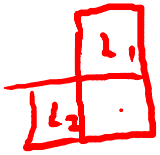
- One pass through image to assign temporary labels and record equivalences
- Second pass to replace temporary labels with equivalence labels
- Data:
  - B(r,c) is the input binary image
  - L(r,c) is the output labeled image

```
for c=1 to MAXCOL       {
        for r=1 to MAXROW       {
                if B(r,c) == 0 then
                        L(r,c) = 0;       % if pixel not white, assign no label
                else       {
                    IF   L(r-1,c)==0 && L(r,c-1)==0
                            L(r,c)= NUMLABEL++;

                    ELSE   L(r-1,c)≠0 && L(r,c-1)==0
                            L(r,c) = L(r-1,c)

                    ELSE   L(r,c-1) ≠0 && L(r-1,c) ==0
                            L(r,c) = L(r,c-1)

                    ELSE   L(r,c-1) ≠0 && L(r-1,c) ≠0
                            L(r,c) = $L_1$
                            RECORD   $L_1$,$L_2$ EQUIV
                END
        END
END
```









EQUIV	TEMP
1	1,2
2	3
3	4,5,6

Example

	1	1		1	
	1	1		1	
		1	1	1	
		1	1	1	

Binary image

	1	1		2	
	1	1		2	
		1	1	2	
		1	1	2	

Temporary labels after 1st pass

	1	1		1	
	1	1		1	
		1	1	1	
		1	1	1	

Final (equivalence) labels after 2nd pass

EQUIV	TEMP
1	1, 2

Matlab Example

- Labeling connected components (white blobs)
  - `im2bw`
    - threshold to convert to binary image
  - `bwlabel`
    - do connected component labeling
    - generate an image of labels
  - `label2rgb`
    - for visualization
    - converts each label to a random color

- If we want to find black blobs
  - `imcomplement`
    - Flip black and white regions
  - then repeat steps

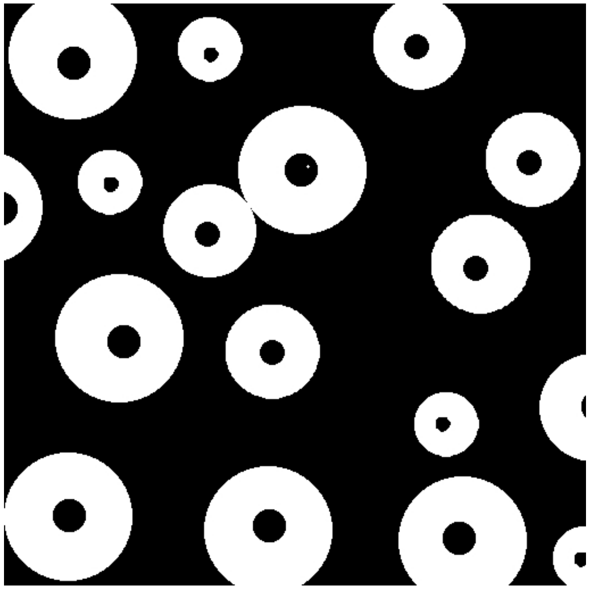


Image Fig9-16(a).jpg

```
>> I = imread('Fig9.16(a).jpg');
>> imshow(I,[])
>> whos
  Name       Size             Bytes  Class    Attributes

  I        512x512           262144  uint8

>> BW = im2bw(I, graythresh(I));
>> figure, imshow(BW)
>> [L,num] = bwlabel(BW);
>> figure, imshow(L,[])
>> num

num =

    17

>> RGB = label2rgb(L);
>> figure, imshow(RGB)
>> BW = imcomplement(BW);
>> [L,num] = bwlabel(BW);
>> RGB = label2rgb(L);
>> figure, imshow(RGB)
```

Binary Image Morphology

- Operations on binary images:
  - dilation and erosion
  - opening and closing
- Uses include
  - shrinking and expanding regions
  - eliminating small regions or holes
- Operations are performed with a “structuring element” S
  - a small binary image
  - like a filter mask

Dilation

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

S

- Defined as

$$B \oplus S = \bigcup_{b \in B} S_b$$

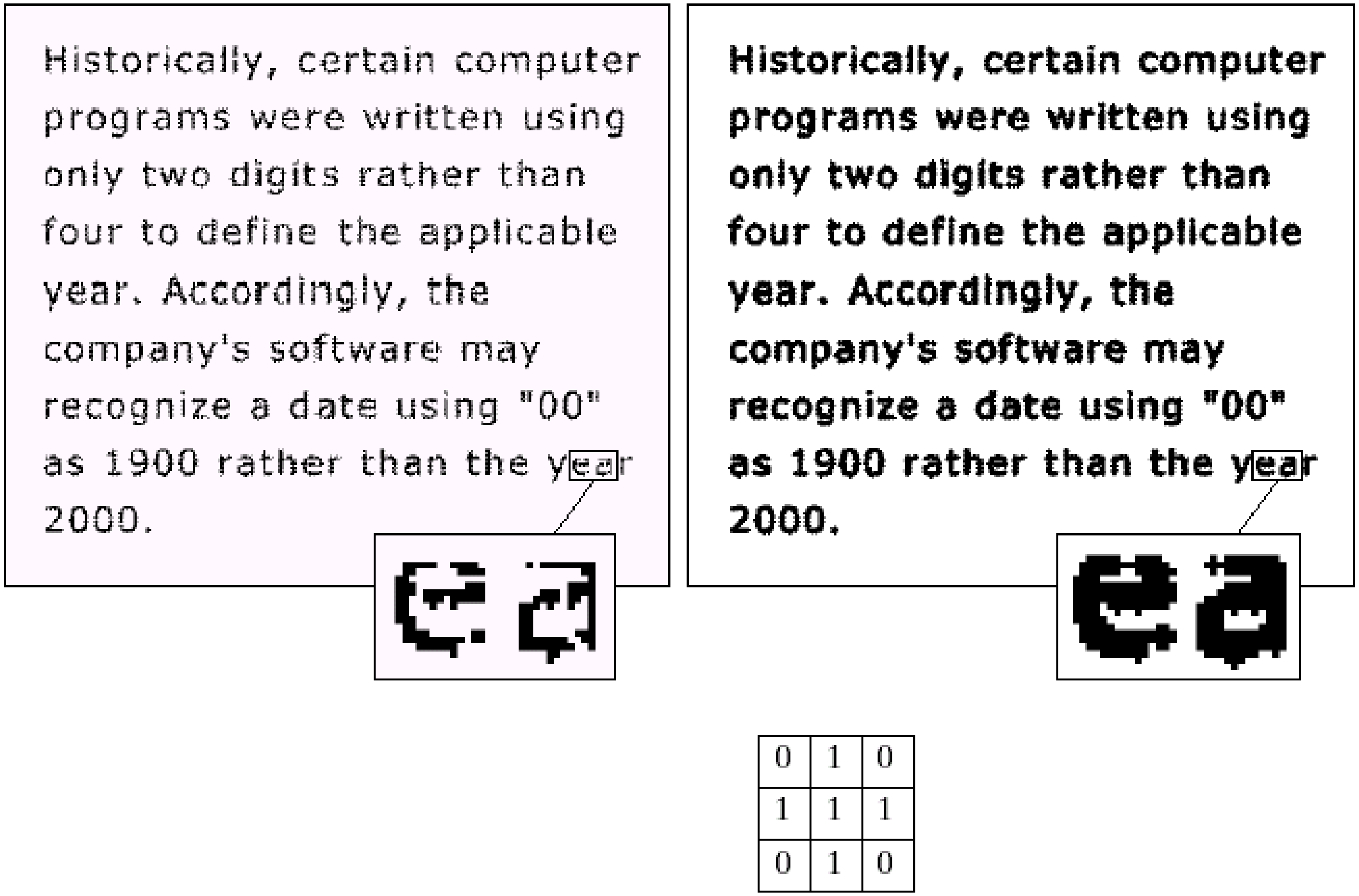
- where
  - S_b is the structuring element S, shifted to b
- Procedure
  - Sweep S over B
  - Everywhere the origin of S touches a 1, OR S with B
- Expands regions

		1		1	
		1	1		

B

	1	1	1	1	1
	1	1	1	1	1
	1	1	1	1	1
	1	1	1	1	

$B \oplus S$



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.

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

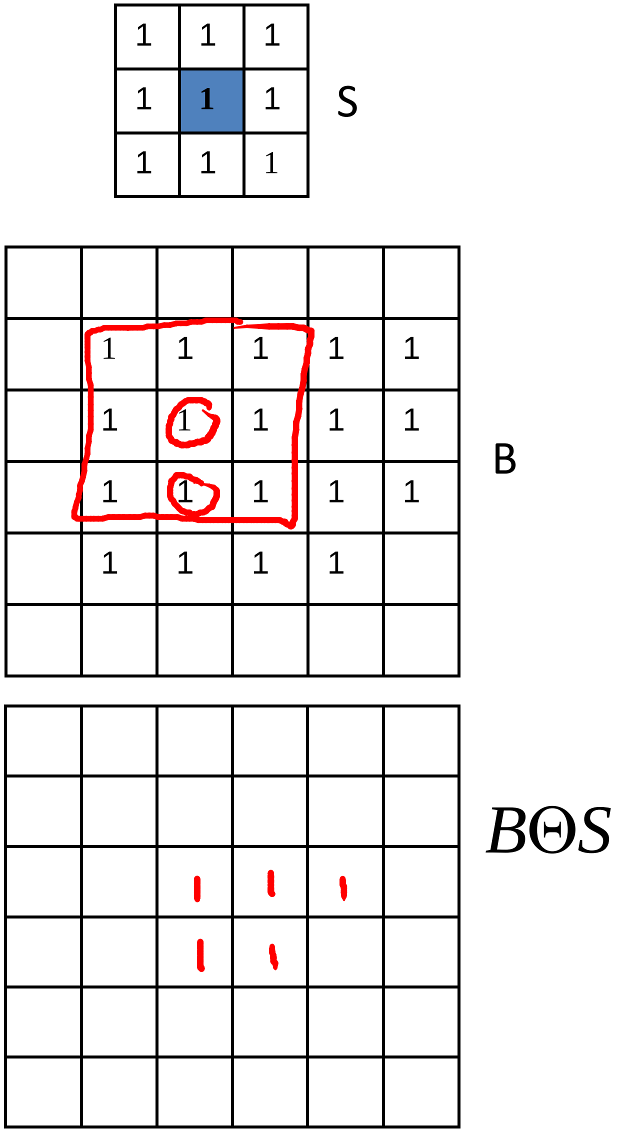
FIGURE 9.5
(a) Sample text of poor resolution with broken characters (magnified view). (b) Structuring element. (c) Dilation of (a) by (b). Broken segments were joined.

Erosion

- Defined as

$$B \Theta S = \{ b \mid b + s \in B, \forall s \in S \}$$

- Procedure
  - Sweep S over B
  - Everywhere S is completely contained in B, output a 1 at the origin of S
- Shrinks regions



S

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

B

	1	1	1	1	1
	1	1	1	1	1
	1	1	1	1	1
	1	1	1	1	

$B \Theta S$

		1	1	1	
		1	1		

Matlab

- Create a structuring element
  - `strel(‘disk’, 5)     % disk shape, radius=5`
- Dilation
  - `imdilate(I,S)`
- Erosion
  - `imerode(I,S)`

Openings and Closings

- Opening
  - Erosion followed by dilation
  - Eliminate small regions and projections

$$B \circ S = \left(B \Theta S\right) \oplus S$$

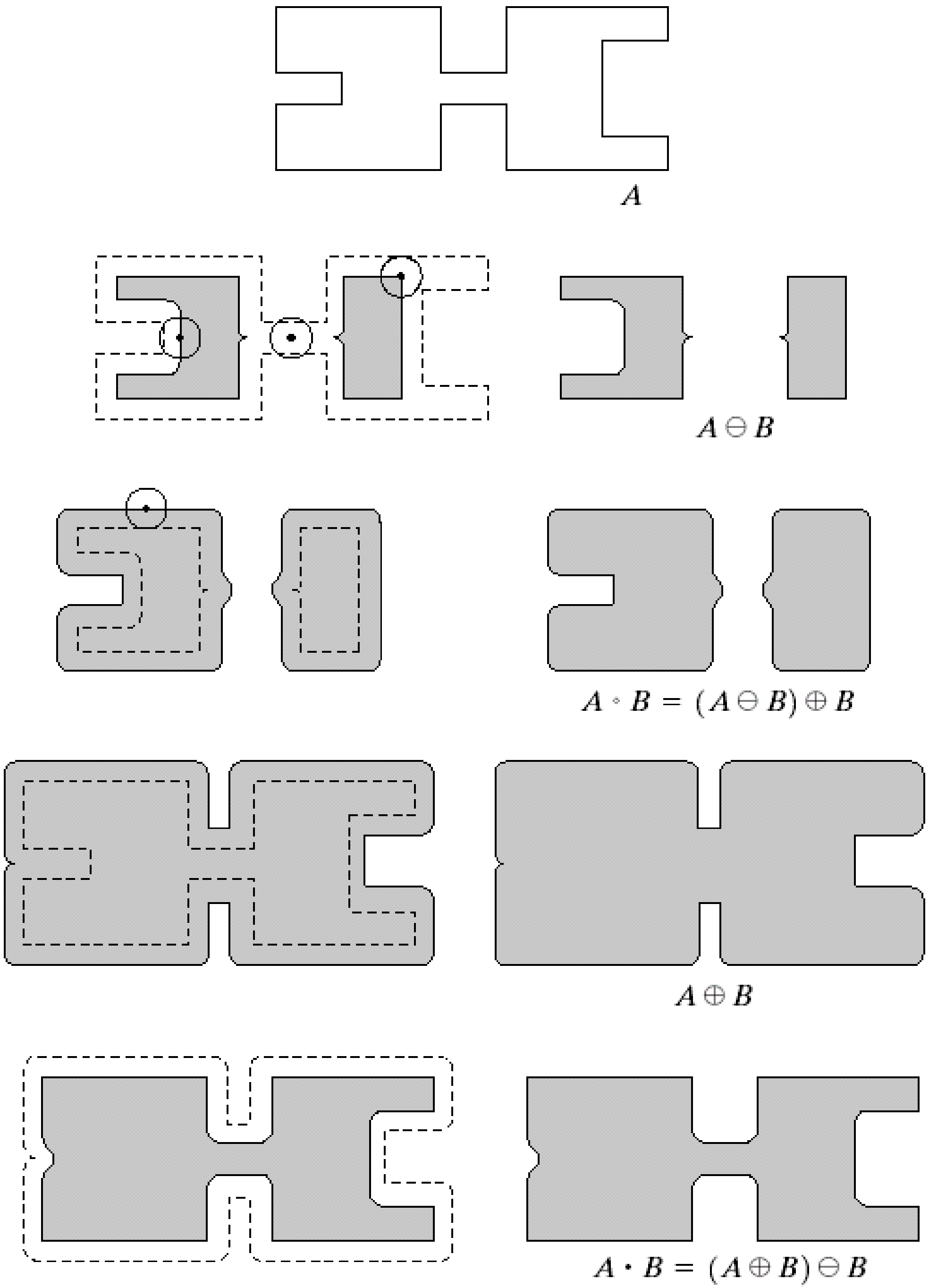
- Closing
  - Dilation followed by erosion
  - Fill in small holes and gaps

$$B \bullet S = \left(B \oplus S\right) \Theta S$$

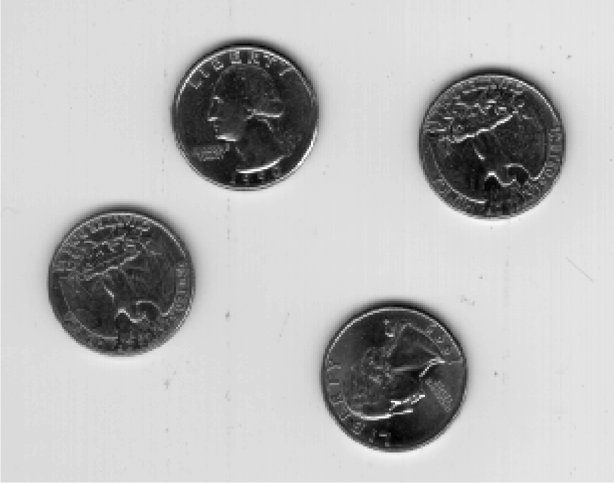
- Matlab `imopen, imclose`

a
b c
d e
f g
h i

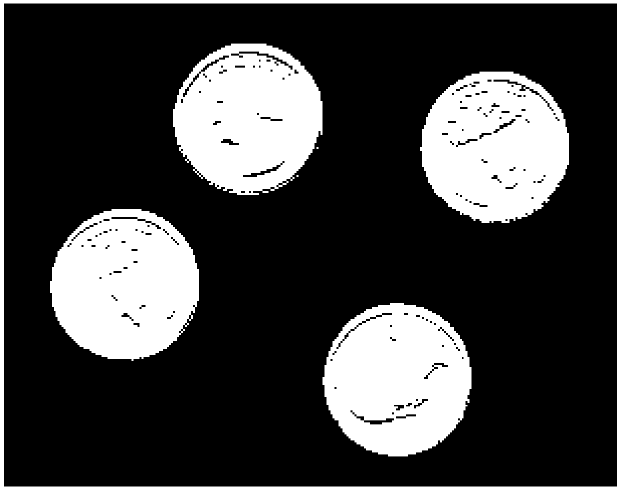
FIGURE 9.10 Morphological opening and closing. The structuring element is the small circle shown in various positions in (b). The dark dot is the center of the structuring element.


A
$A \ominus B$
$A \circ B = (A \ominus B) \oplus B$
$A \oplus B$
$A \bullet B = (A \oplus B) \ominus B$

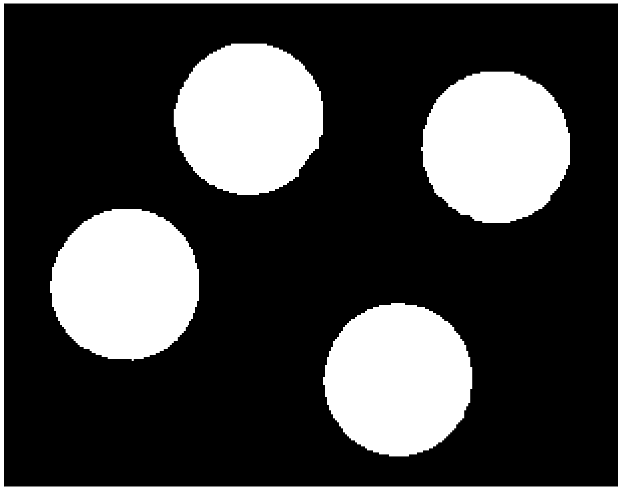
Closing Example



Matlab image “eight.tif”



threshold, complement



close with disk of radius=5

Region Properties

- Basic features

$$\text{Area} \quad A = \sum_{(r,c)\in R} 1 \qquad \text{Centroid} \quad \bar{r} = \frac{1}{A}\sum_{(r,c)\in R} r, \quad \bar{c} = \frac{1}{A}\sum_{(r,c)\in R} c$$

- Higher order moments

$$\mu_{rr} = \frac{1}{A}\sum_{(r,c)\in R}(r-\bar{r})^2, \quad \mu_{cc} = \frac{1}{A}\sum_{(r,c)\in R}(c-\bar{c})^2, \quad \mu_{rc} = \frac{1}{A}\sum_{(r,c)\in R}(r-\bar{r})(c-\bar{c})$$

- Principal axes
  - computed from second order moments

- Matlab function `regionprops(L)`

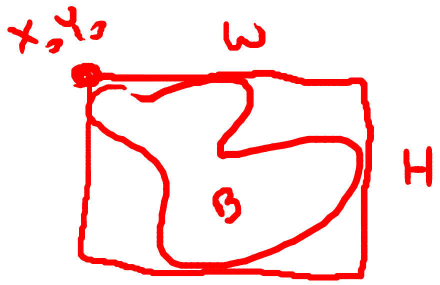
Matlab Structures

- A structure is an array with named "fields"
- Access using: *structurename.fieldname*
- See Matlab help for more info
- Example

TO ACCESS

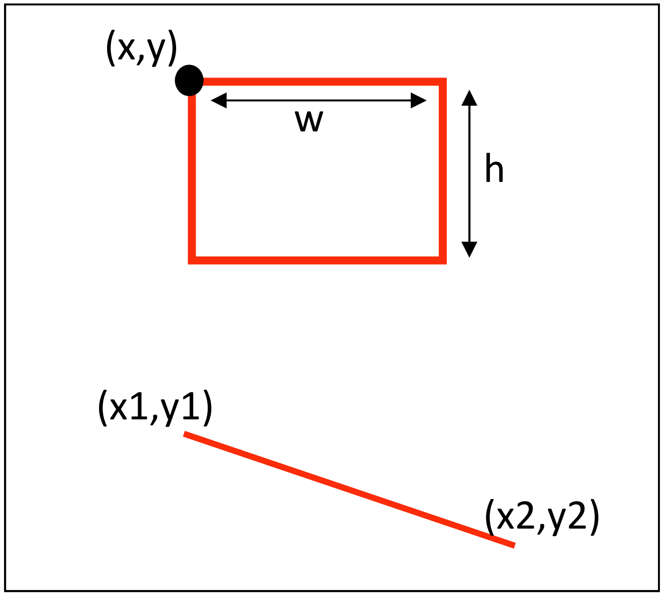
BLOBS(2). AREA

```
>> blobs = regionprops(L);
>> blobs(1)

ans =

           Area: 2058  X      Y
       Centroid: [15.7216 179.8717]
    BoundingBox: [0.5000 133.5000 34 93]

                  X0       Y0
```



Matlab Graphics

```
rectangle('Position', [x y w h]), ‘EdgeColor’, ‘r’);
```



```
line([x1 x2], [y1 y2], ‘Color’, ‘r’);
```

Matlab Example

- Draw bounding box around largest blob
- Draw cross hair on its centroid

```
>> I = imread('Fig9.16(a).jpg');
>> BW = im2bw(I, graythresh(I));
>> L = bwlabel(BW);
>> blobs = regionprops(L);
>> areas = cat(1, blobs(:).Area);  % concatentate along dimension 1
>> [value, index] = max(areas);
>>
>> value
value =
      14240

>> index
index =
    6

>> imshow(I)
>> rectangle('Position', blobs(index).BoundingBox, 'EdgeColor', 'w');

>> rectangle('Position', [x0-1 y0-1 3 3], 'EdgeColor', 'k');
```