Introduction to Digital Image Processing with MATLAB® Asia Edition
McAndrew‧Wang‧Tseng

Chapter 10:

Mathematical Morphology

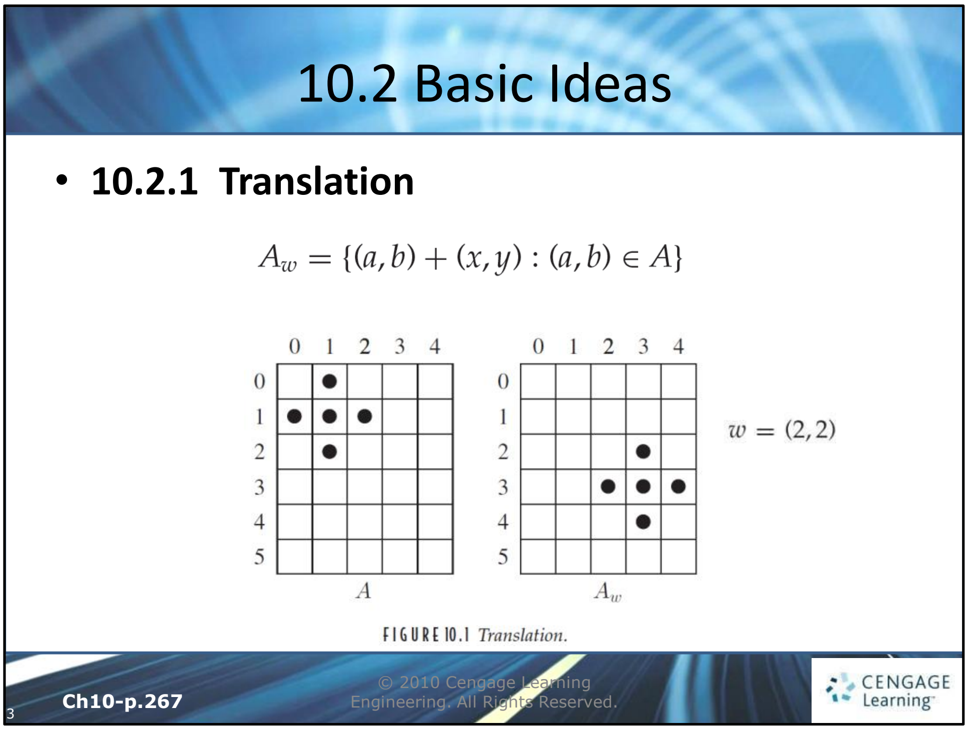
CENGAGE Learning

10.1 Introduction

- *Morphology* is a branch of image processing that is particularly useful for analyzing **shapes** in images
- We will develop basic morphological tools for investigation of **binary images** and then show how to extend these tools to **grayscale images**

Ch10-p.267

CENGAGE Learning



10.2 Basic Ideas

- **10.2.1 Translation**

$$A_w = \{(a,b) + (x,y) : (a,b) \in A\}$$

	0	1	2	3	4
0		●			
1	●	●	●		
2		●			
3					
4					
5					

A

	0	1	2	3	4
0					
1					
2				●	
3			●	●	●
4				●	
5					

A_w

$w = (2,2)$

FIGURE 10.1 *Translation.*

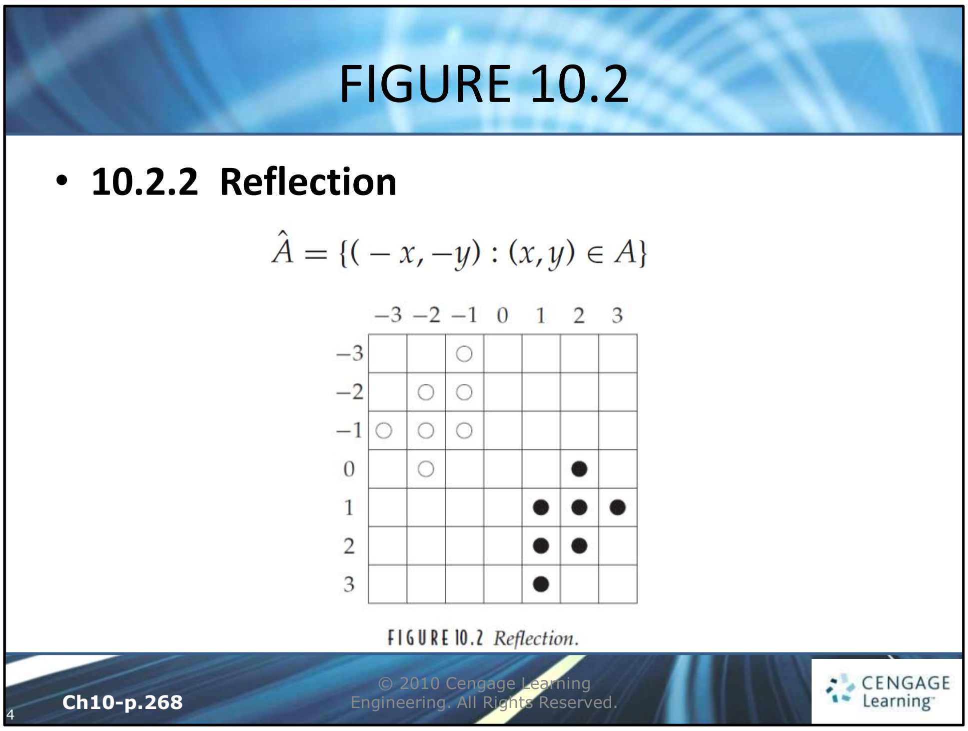


FIGURE 10.2

- **10.2.2 Reflection**

$$\hat{A} = \{(-x,-y) : (x,y) \in A\}$$

	−3	−2	−1	0	1	2	3
−3			○				
−2		○	○				
−1	○	○	○				
0		○				●	
1					●	●	●
2					●	●	
3					●		

FIGURE 10.2 *Reflection.*

10.3 Dilation and Erosion

- **10.3.1 Dilation**

$$A \oplus B = \bigcup_{x \in B} A_x$$

✓ A and B are sets of pixels

$$A \oplus B = \{(x, y) + (u, v) : (x, y) \in A, (u, v) \in B\}$$

✓ Also known as **Minkowski addition**

FIGURE 10.3

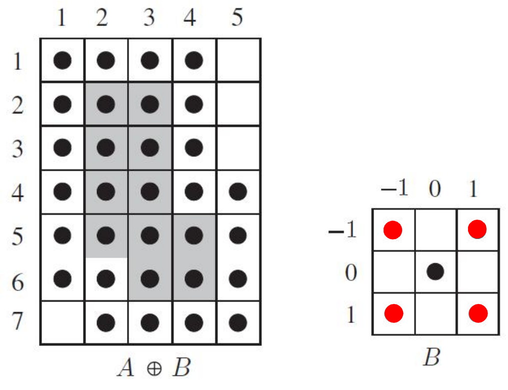


FIGURE 10.3 *Dilation.*

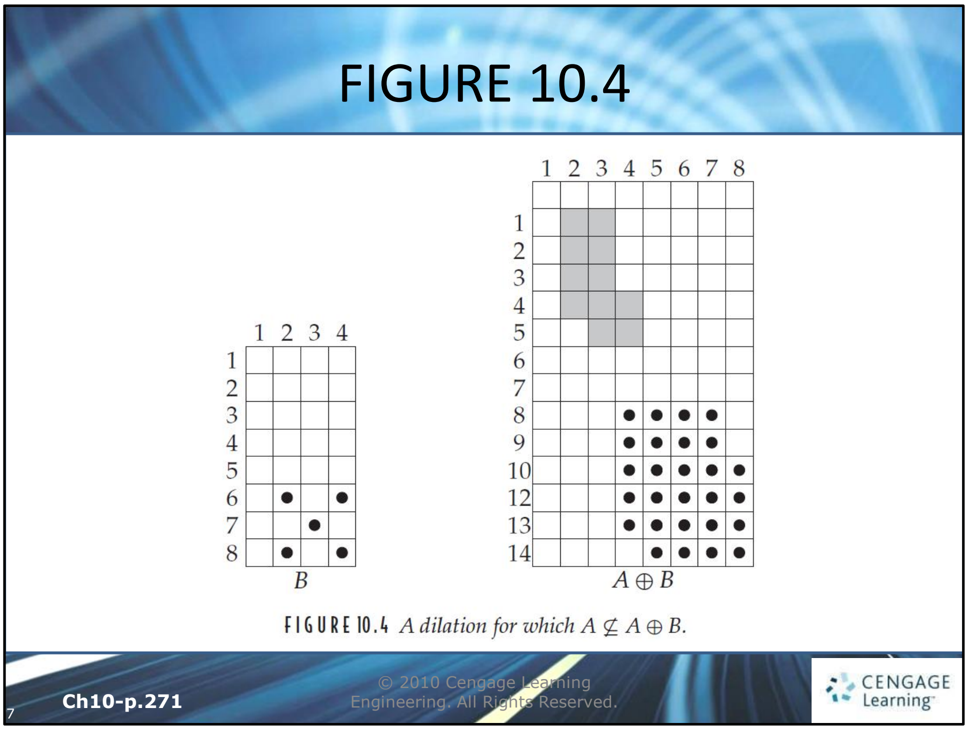


FIGURE 10.4

	1	2	3	4
1				
2				
3				
4				
5				
6		●		●
7			●	
8		●		●

B

	1	2	3	4	5	6	7	8
1		▨	▨					
2		▨	▨					
3		▨	▨					
4		▨	▨	▨				
5			▨	▨				
6								
7								
8				●	●	●	●	
9				●	●	●	●	
10				●	●	●	●	●
12				●	●	●	●	●
13				●	●	●	●	●
14					●	●	●	●

$A \oplus B$

FIGURE 10.4 *A dilation for which* $A \not\subseteq A \oplus B$.

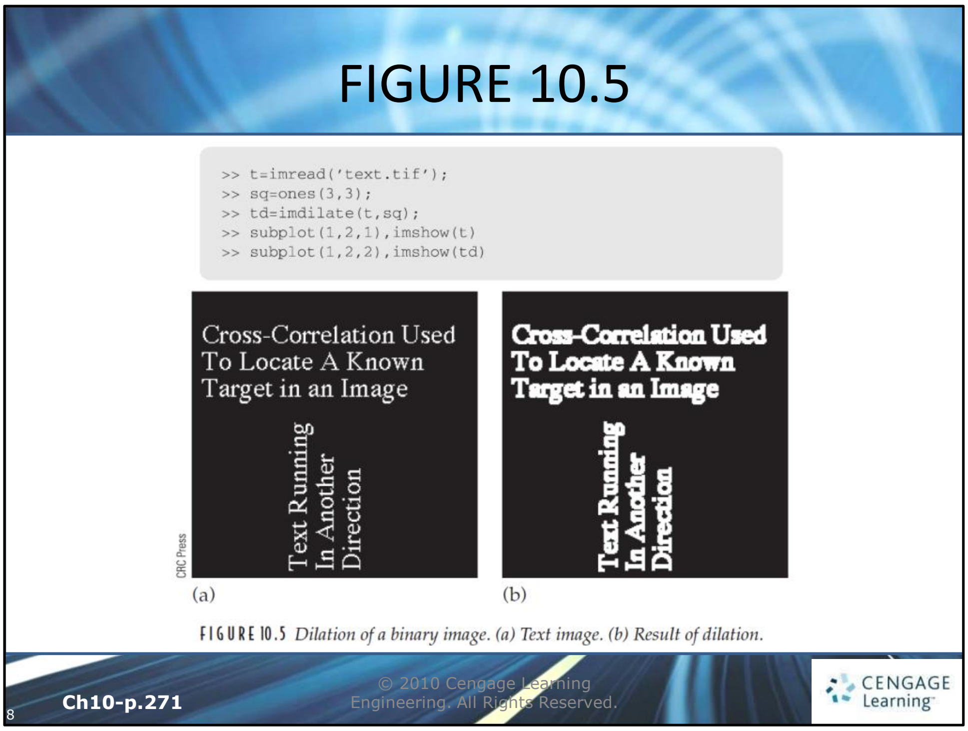


FIGURE 10.5

```
>> t=imread('text.tif');
>> sq=ones(3,3);
>> td=imdilate(t,sq);
>> subplot(1,2,1),imshow(t)
>> subplot(1,2,2),imshow(td)
```

FIGURE 10.5 *Dilation of a binary image. (a) Text image. (b) Result of dilation.*

10.3 Dilation and Erosion

- **10.3.1 Erosion**

$$A \ominus B = \{w : B_w \subseteq A\}$$

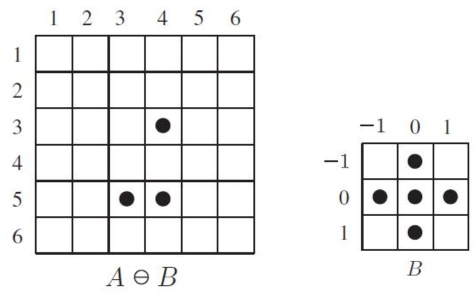


FIGURE 10.6 *Erosion with a cross-shaped structuring element.*

Ch10-p.272

CENGAGE Learning

FIGURE 10.7

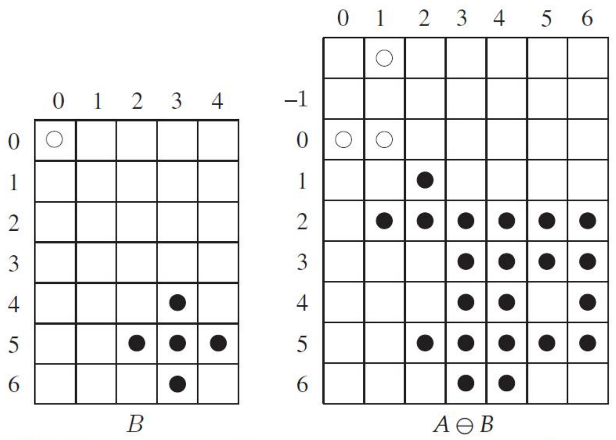


FIGURE 10.7 *Erosion with a structuring element not containing the origin.*

Ch10-p.274

CENGAGE Learning

FIGURE 10.8

```
>> c=imread(’circbw.tif’);
>> ce=imerode(c,sq);
>> subplot(1,2,1),imshow(c)
>> subplot(1,2,2),imshow(ce)
```

(a) (b)

FIGURE 10.8 *Erosion of a binary image. (a) Original image. (b) Result of erosion.*

Ch10-p.272

10.3 Dilation and Erosion

- **RELATIONSHIP BETWEEN EROSION AND DILATION**

$$\overline{A \ominus B} = \overline{A} \oplus \hat{B}$$

$$\overline{A \oplus B} = \overline{A} \ominus \hat{B}$$

```
>> lhs=~imerode(t,sq);
```

```
>> rhs=imdilate(~t,sq);
```

```
>> all(lhs(:)==rhs(:))
```

Ch10-p.274

10.3 Dilation and Erosion

- **10.3.3 An Application: Boundary Detection**
  - ✓ If A is an image and B a small structuring element

(i)	$A - (A \ominus B)$	internal boundary
(ii)	$(A \oplus B) - A$	external boundary
(iii)	$(A \oplus B) - (A \ominus B)$	morphological gradient

FIGURE 10.9

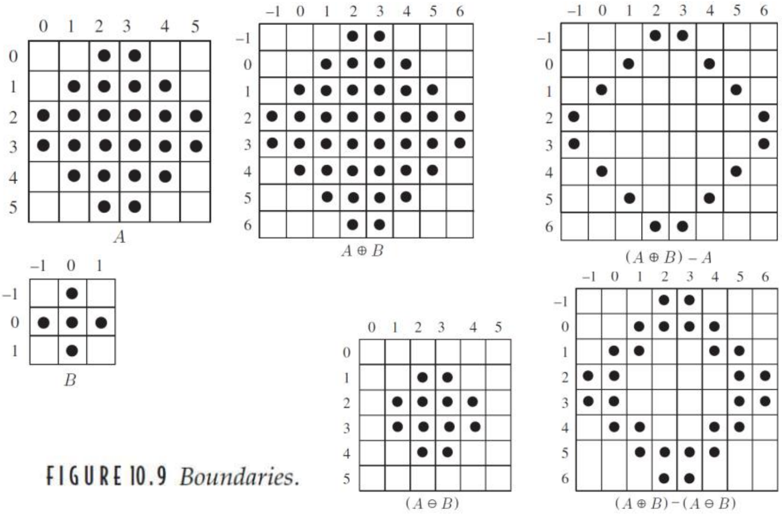


FIGURE 10.9 *Boundaries.*

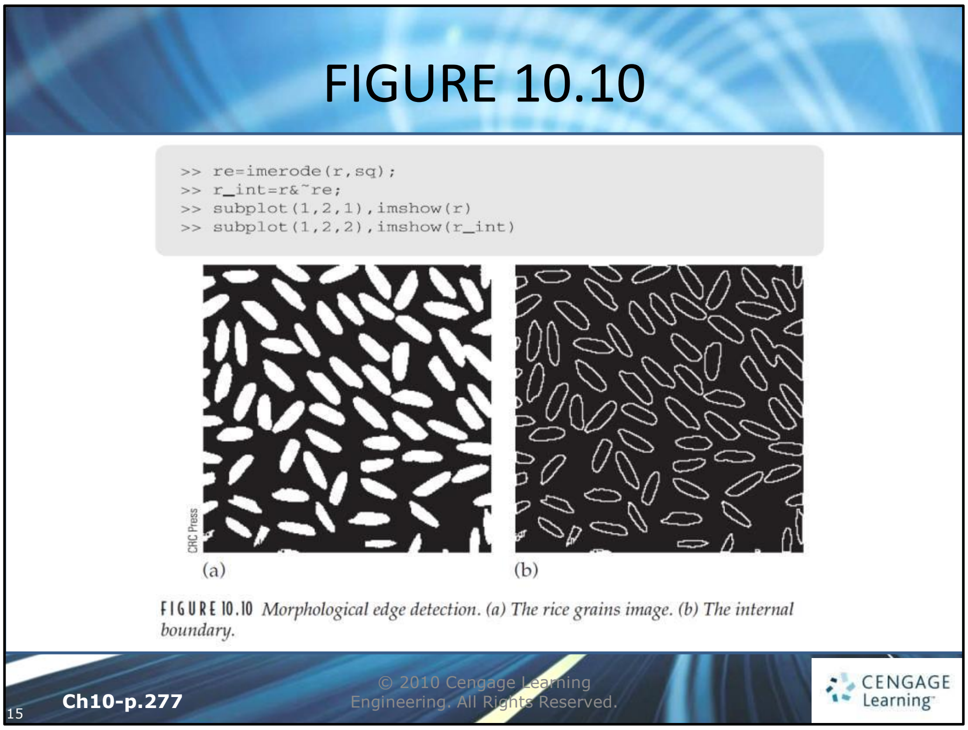


FIGURE 10.10

```
>> re=imerode(r,sq);
>> r_int=r&~re;
>> subplot(1,2,1),imshow(r)
>> subplot(1,2,2),imshow(r_int)
```

CRC Press

FIGURE 10.10 *Morphological edge detection. (a) The rice grains image. (b) The internal boundary.*

Ch10-p.277

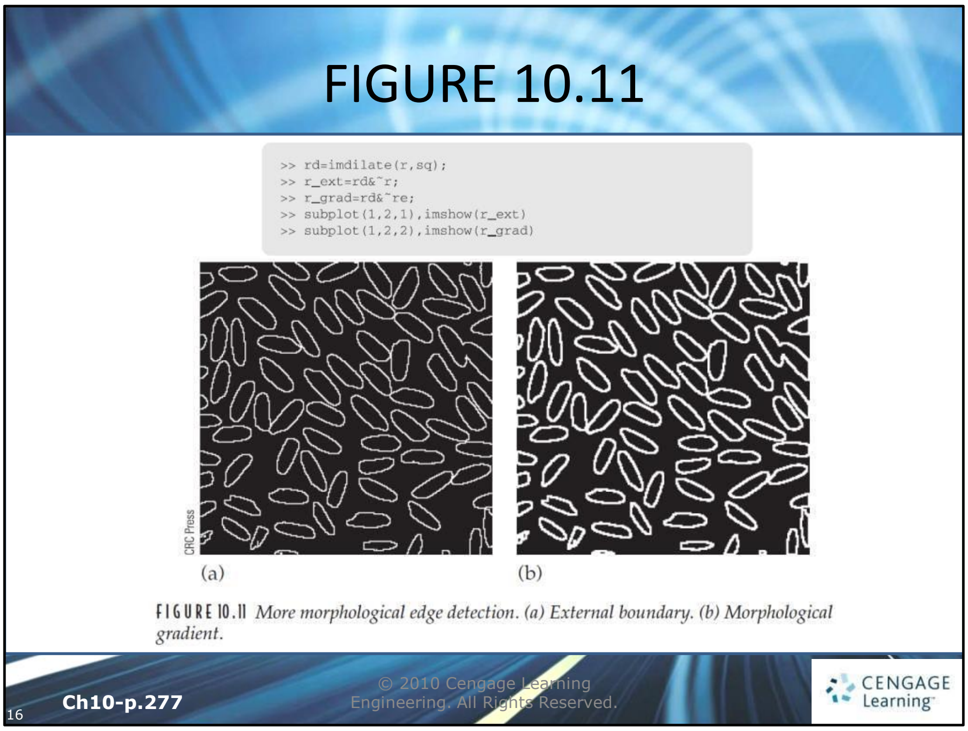


FIGURE 10.11

```
>> rd=imdilate(r,sq);
>> r_ext=rd&~r;
>> r_grad=rd&~re;
>> subplot(1,2,1),imshow(r_ext)
>> subplot(1,2,2),imshow(r_grad)
```

CRC Press

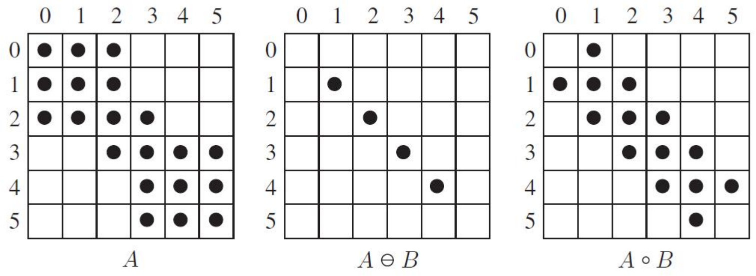
FIGURE 10.11 *More morphological edge detection. (a) External boundary. (b) Morphological gradient.*

Ch10-p.277

10.4 Opening and Closing

- **10.4.1 Opening** (function: `imopen`)

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



A

	0	1	2	3	4	5
0	●	●	●			
1	●	●	●			
2	●	●	●	●		
3			●	●	●	●
4				●	●	●
5				●	●	●

$A \ominus B$

	0	1	2	3	4	5
0						
1		●				
2			●			
3				●		
4					●	
5						

$A \circ B$

	0	1	2	3	4	5
0		●				
1	●	●	●			
2		●	●	●		
3			●	●	●	
4				●	●	●
5					●	

FIGURE 10.12 *Opening.*

10.4 Opening and Closing

✓ $(A \circ B) \subseteq A$. Note that this is not the case with erosion. As we have seen, an erosion may not necessarily be a subset

✓ $(A \circ B) \circ B = A \circ B$. That is, an opening can never be done more than once (**idempotence, 멱등법칙**)

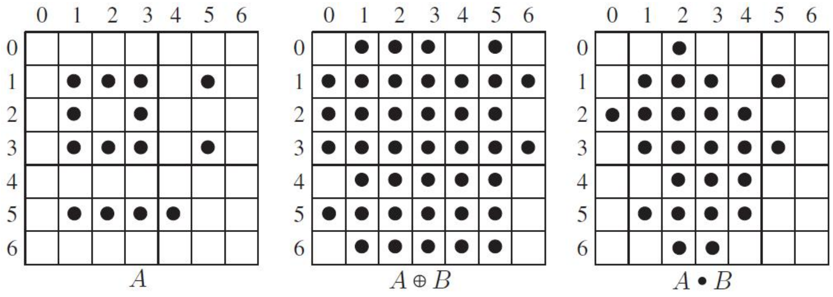
✓ If $A \subseteq C$, then $(A \circ B) \subseteq (C \circ B)$

✓ Opening tends to smooth an image, to break narrow joins, and to remove thin protrusions (돌출부)

10.4 Opening and Closing

- **10.4.2 Closing** (function: `imclose`)

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



A

	0	1	2	3	4	5	6
0							
1		●	●	●		●	
2		●		●			
3		●	●	●		●	
4							
5		●	●	●	●		
6							

$A \oplus B$

	0	1	2	3	4	5	6
0		●	●	●		●	
1	●	●	●	●	●	●	●
2	●	●	●	●	●	●	
3	●	●	●	●	●	●	●
4		●	●	●	●	●	
5	●	●	●	●	●	●	
6		●	●	●	●	●	

$A \bullet B$

	0	1	2	3	4	5	6
0			●				
1		●	●	●		●	
2	●	●	●	●	●		
3		●	●	●	●	●	
4			●	●	●		
5		●	●	●	●		
6			●	●			

FIGURE 10.13 *Closing.*

Ch10-p.279

10.4 Opening and Closing

- ✓ $A \subseteq (A \bullet B)$
- ✓ $(A \bullet B) \bullet B = A \bullet B$; that is, closing, like opening, is **idempotent**
- ✓ If $A \subseteq C$, then $(A \bullet B) \subseteq (C \bullet B)$
- ✓ Closing also tends to smooth an image, but it fuses narrow breaks and thin gulfs and eliminates small holes

Ch10-p.279

FIGURE 10.14

CRC Press

FIGURE 10.14 *An example of closing.*

10.4 Opening and Closing

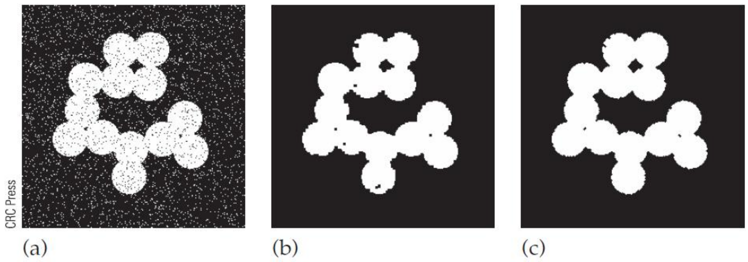
- **AN APPLICATION: NOISE REMOVAL**
  - ✓ **Morphological filtering**

```
>> c=imread(’circles.tif’);
>> x=rand(size(c));
>> d1=find(x<=0.05);
>> d2=find(x>=0.95);
>> c(d1)=0;
>> c(d2)=1;
>> imshow(c)
```

```
>> cf1=imclose(imopen(c,sq),sq);
>> figure,imshow(cf1)
>> cf2=imclose(imopen(c,cr),cr);
>> figure,imshow(cf2)
```

FIGURE 10.15

$$((A \circ B) \bullet B)$$



(a) (b) (c)

FIGURE 10.15 *A noisy binary image and results after morphological filtering with different structuring elements. (a) Binary noise. (b) Using the square kernel. (c) Using the cross kernel.*

10.4 Opening and Closing

- **RELATIONSHIP BETWEEN OPENING AND CLOSING**

$$\overline{A \bullet B} = \overline{A} \circ \hat{B}$$

$$\overline{A \circ B} = \overline{A} \bullet \hat{B}$$

✓ see Haralick and Shapiro [11] for a formal proof

10.5 The Hit-or-Miss Transform

- ✓ B is the square structuring element

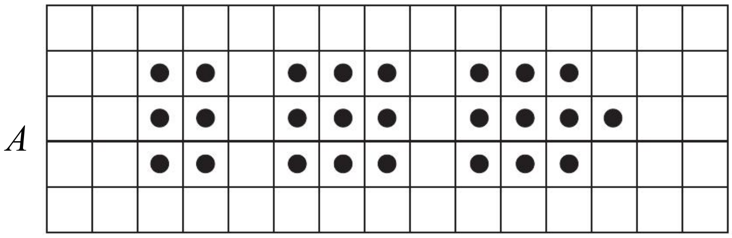


FIGURE 10.16 *An image A containing a shape to be found.*

FIGURE 10.17 & 18

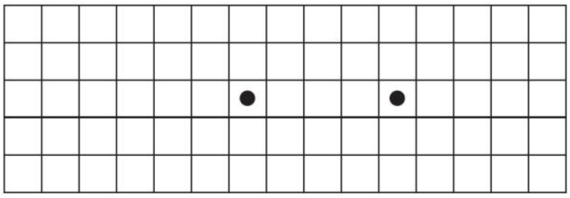


FIGURE 10.17 *The erosion* $A \ominus B$.

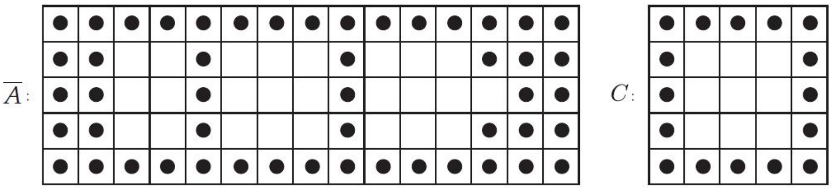


FIGURE 10.18 *The complement and the second structuring element.*

FIGURE 10.19

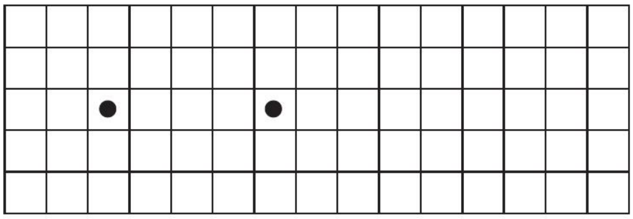


FIGURE 10.19 *The erosion* $\overline{A} \ominus C$.

10.5 The Hit-or-Miss Transform

$$A \circledast B = (A \ominus B_1) \cap (\overline{A} \ominus B_2)$$

```
>> b1=ones(1,6);
>> b2=[1 1 1 1 1 1 1 1;1 0 0 0 0 0 0 1; 1 1 1 1 1 1 1 1];
>> tb1=imerode(t,b1);
>> tb2=imerode(~t,b2);
>> hit_or_miss=tb1&tb2;
>> [x,y]=find(hit_or_miss==1)
```

```
>> tb1=imerode(t,b1);
```

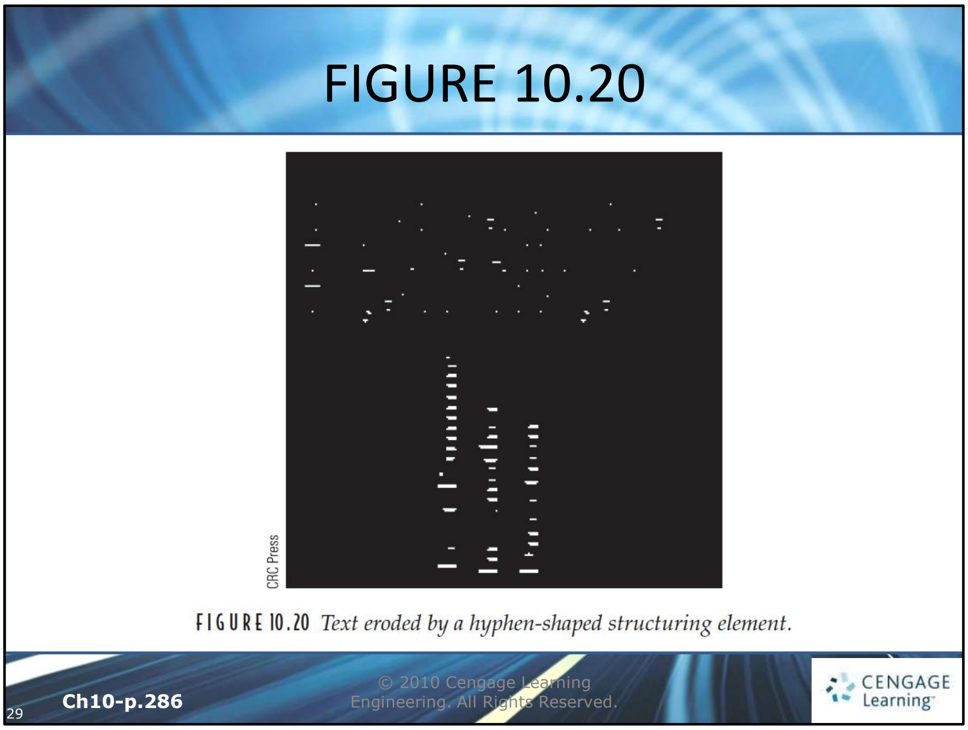
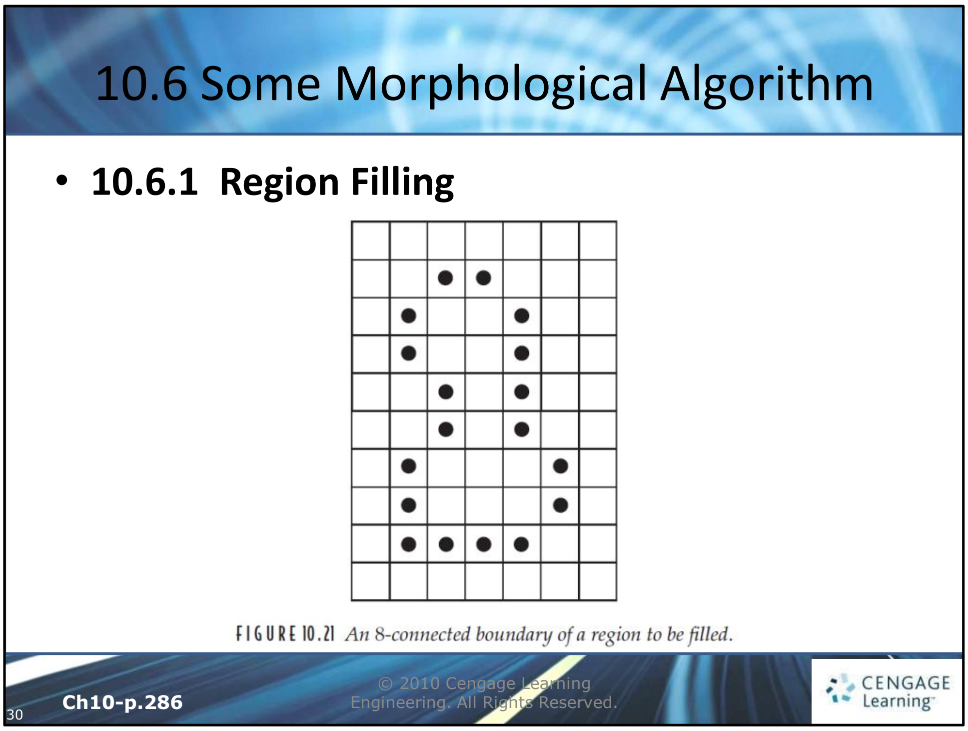


FIGURE 10.20

CRC Press

FIGURE 10.20 *Text eroded by a hyphen-shaped structuring element.*



10.6 Some Morphological Algorithm

- **10.6.1 Region Filling**

		●	●			
	●			●		
	●			●		
		●		●		
		●		●		
	●				●	
	●				●	
	●	●	●	●		

FIGURE 10.21 *An 8-connected boundary of a region to be filled.*

10.6 Some Morphological Algorithm

- Given a pixel p within the region, we wish to fill up the entire region
- we start with p and **dilate** as many times as necessary with the cross-shaped structuring element B

$$\{p\} = X_0, X_1, X_2, \ . \ . \ . \ , X_k = X_{k+1},$$

$$X_n = (X_{n-1} \oplus B) \cap \overline{A}$$

FIGURE 10.22

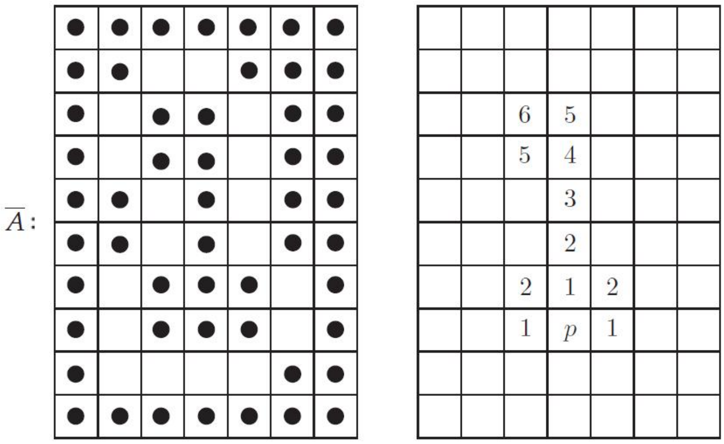
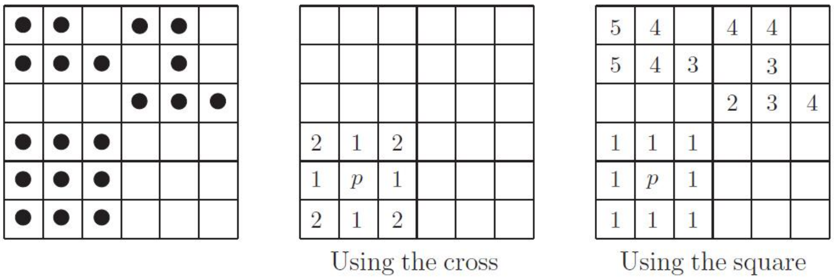


FIGURE 10.22 *The process of filling a region.*

10.6 Some Morphological Algorithm

- **10.6.2 Connected Components**



●	●		●	●	
●	●	●		●	
			●	●	●
●	●	●			
●	●	●			
●	●	●			

2	1	2			
1	p	1			
2	1	2			

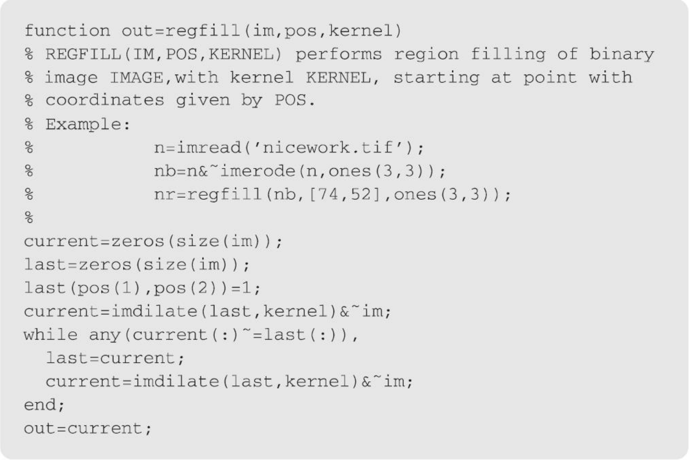
Using the cross

5	4		4	4	
5	4	3		3	
			2	3	4
1	1	1			
1	p	1			
1	1	1			

Using the square

FIGURE 10.23 *Filling connected components.*

FIGURE 10.24



```
function out=regfill(im,pos,kernel)
% REGFILL(IM,POS,KERNEL) performs region filling of binary
% image IMAGE,with kernel KERNEL, starting at point with
% coordinates given by POS.
% Example:
%          n=imread('nicework.tif');
%          nb=n&~imerode(n,ones(3,3));
%          nr=regfill(nb,[74,52],ones(3,3));
%
current=zeros(size(im));
last=zeros(size(im));
last(pos(1),pos(2))=1;
current=imdilate(last,kernel)&~im;
while any(current(:)~=last(:)),
  last=current;
  current=imdilate(last,kernel)&~im;
end;
out=current;
```

FIGURE 10.24 *A simple program for filling regions.*

FIGURE 10.25

```
>> n=imread('nicework.tif');
>> imshow(n),pixval on
>> nb=n&~imerode(n,sq);
>> figure,imshow(nb)
>> nf=regfill(nb,[74,52],sq);
>> figure,imshow(nf)
```

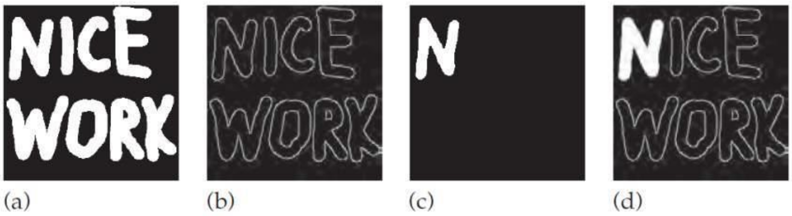


FIGURE 10.25 *Region filling.*

Ch10-p.289

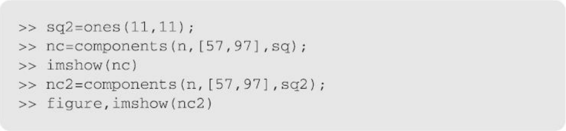
FIGURE 10.26

```
function out=components(im,pos,kernel)
% COMPONENTS(IM,POS,KERNEL) produces the connected component
% of binary image IMAGE which nicludes the point with coordinates given
% by POS, using kernel KERNEL.
%
% Example:
%           n=imread('nicework.tif');
%           nc=components(nb,[74,52],ones(3,3));
%
current=zeros(size(im));
last=zeros(size(im));
last(pos(1),pos(2))=1;
current=imdilate(last,kernel)&im;
while any(current(:)~=last(:)),
  last=current;
  current=imdilate(last,kernel)&im;
end;
out=current;
```

FIGURE 10.26 *A simple program for connected components.*

Ch10-p.290

FIGURE 10.27



```
>> sq2=ones(11,11);
>> nc=components(n,[57,97],sq);
>> imshow(nc)
>> nc2=components(n,[57,97],sq2);
>> figure,imshow(nc2)
```

(a) (b)

FIGURE 10.27 *Connected components.*

10.6 Some Morphological Algorithm

- **10.6.3 Skeletonization**

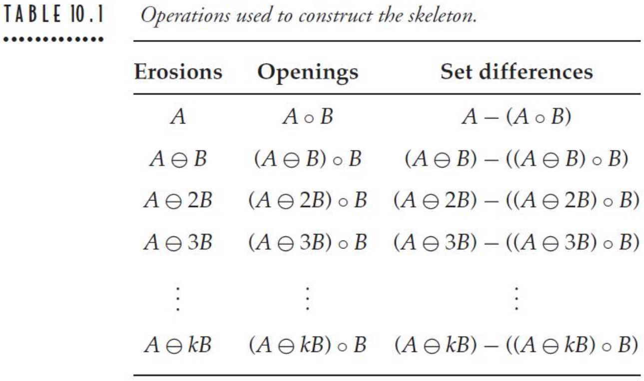


TABLE 10.1 *Operations used to construct the skeleton.*

Erosions	Openings	Set differences
A	$A \circ B$	$A - (A \circ B)$
$A \ominus B$	$(A \ominus B) \circ B$	$(A \ominus B) - ((A \ominus B) \circ B)$
$A \ominus 2B$	$(A \ominus 2B) \circ B$	$(A \ominus 2B) - ((A \ominus 2B) \circ B)$
$A \ominus 3B$	$(A \ominus 3B) \circ B$	$(A \ominus 3B) - ((A \ominus 3B) \circ B)$
$\vdots$	$\vdots$	$\vdots$
$A \ominus kB$	$(A \ominus kB) \circ B$	$(A \ominus kB) - ((A \ominus kB) \circ B)$

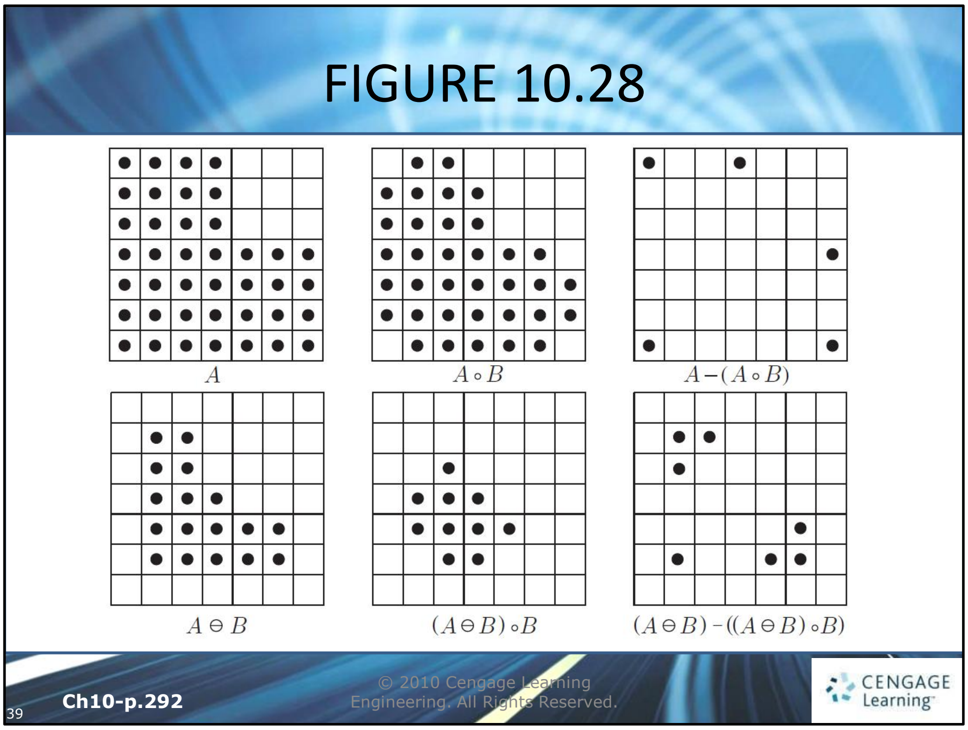


FIGURE 10.28

A

$A \circ B$

$A-(A \circ B)$

$A \ominus B$

$(A \ominus B) \circ B$

$(A \ominus B)-((A \ominus B) \circ B)$

Ch10-p.292

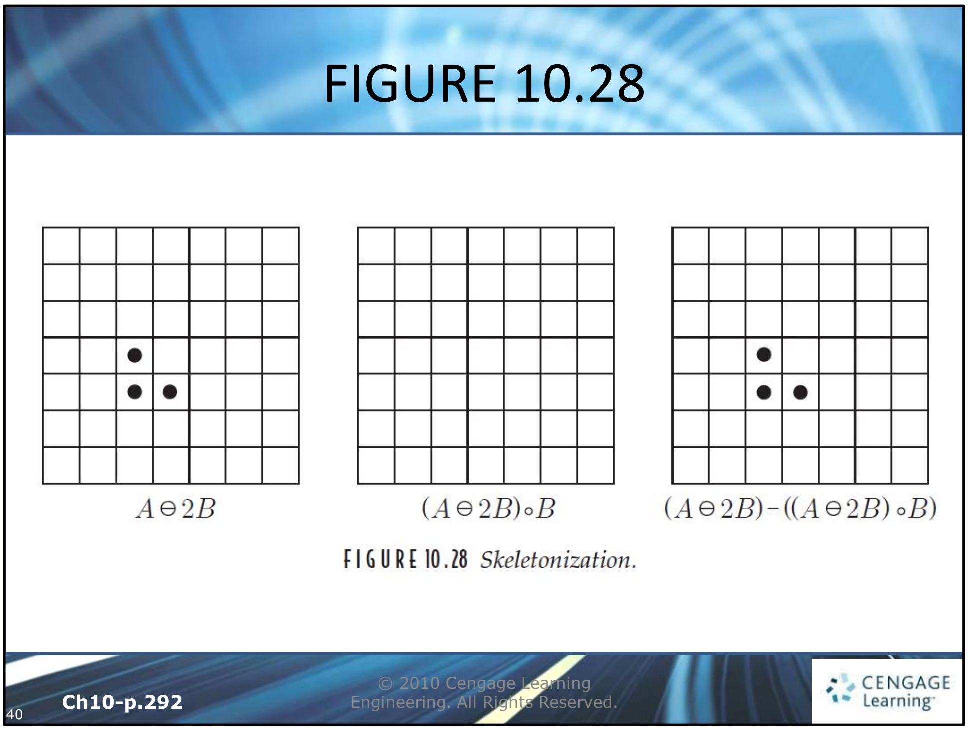


FIGURE 10.28

$A \ominus 2B$

$(A \ominus 2B) \circ B$

$(A \ominus 2B)-((A \ominus 2B) \circ B)$

FIGURE 10.28 *Skeletonization.*

Ch10-p.292

FIGURE 10.29

This method of skeletonization is called **Lantuéjoul's method**. For details see Serra [33]

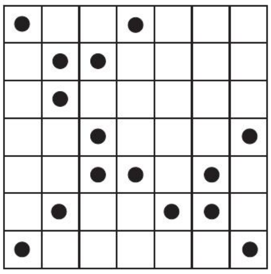
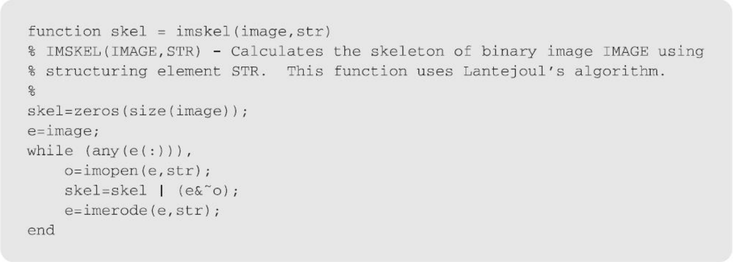


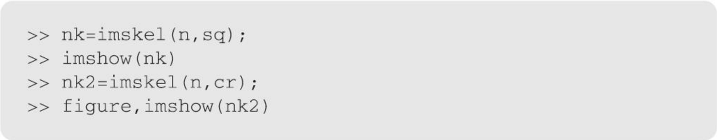
FIGURE 10.29 *The final skeleton.*

FIGURE 10.30



```
function skel = imskel(image,str)
% IMSKEL(IMAGE,STR) - Calculates the skeleton of binary image IMAGE using
% structuring element STR.  This function uses Lantejoul's algorithm.
%
skel=zeros(size(image));
e=image;
while (any(e(:))),
    o=imopen(e,str);
    skel=skel | (e&~o);
    e=imerode(e,str);
end
```

FIGURE 10.30 *A simple program for computing skeletons.*

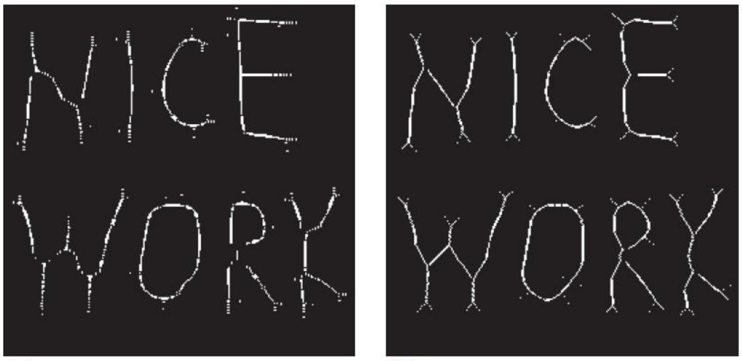


```
>> nk=imskel(n,sq);
>> imshow(nk)
>> nk2=imskel(n,cr);
>> figure,imshow(nk2)
```

FIGURE 10.31

Square-structuring element

Cross-structuring element



(a)

(b)

FIGURE 10.31 *Skeletonization of a binary image.*

10.7 A Note on MATLAB's bwmorph Function

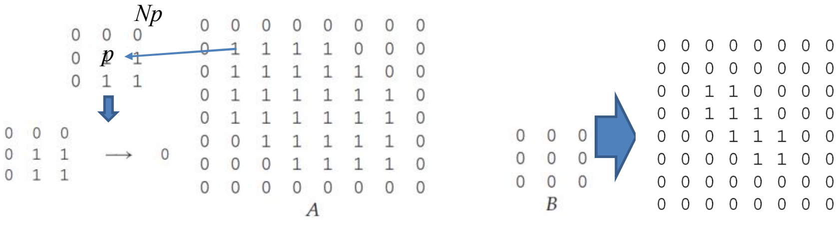
- Based on **lookup tables** (ch11)
  - ✓ Consider the 3 × 3 neighborhood of a pixel
  - ✓ Since each pixel in the neighborhood can have only two values, there are $2^9 = 512$ different possible neighborhoods
  - ✓ Define a morphological operation to be a function that maps these neighborhoods to the values 0 and 1

10.7 A Note on MATLAB's bwmorph Function

- ✓ Each possible neighborhood state can be associated with a numeric value from 0 (all pixels have value 0) to 511 (all pixels have value 1)
- ✓ The lookup table is then a binary vector of length 512. Its kth element is the value of the function for state k
- Many other operations can be defined by this method (see the help file for `bwmorph`)

10.8 Grayscale Morphology

- **Erosion**
  - ✓ 1. Find the 3 × 3 neighborhood N_p of p
  - ✓ 2. Compute the matrix $N_p - B$
  - ✓ 3. Find the minimum of that result



$$\begin{matrix} 0 & 0 & 0 \\ 0 & p & 1 \\ 0 & 1 & 1 \end{matrix}$$

$$\begin{matrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{matrix} \longrightarrow 0$$

$$A = \begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix} \quad B = \begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{matrix}$$

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

FIGURE 10.32 *An example for erosion.*

10.8 Grayscale Morphology

- **Dilation**
  - ✓ 1. Find the 3 × 3 neighborhood N_p of p
  - ✓ 2. Compute the matrix $N_p + B$
  - ✓ 3. Find the maximum of that result
- **Summary**

$$(A \ominus B)(x,y) = \min\{A(x+s, y+t) - B(s,t), (s,t) \in D_B\},$$
$$(A \oplus B)(x,y) = \max\{A(x+s, y+t) + B(s,t), (s,t) \in D_B\}.$$

FIGURE 10.33

$$\begin{array}{cc|ccccc} & & & & y & & \\ & & 1 & 2 & 3 & 4 & 5 \\ \hline & 1 & 10 & 20 & 20 & 20 & 30 \\ & 2 & 20 & 30 & 30 & 40 & 50 \\ x & 3 & 20 & 30 & 30 & 50 & 60 \\ & 4 & 20 & 40 & 50 & 50 & 60 \\ & 5 & 30 & 50 & 60 & 60 & 70 \end{array} \qquad \begin{array}{cc|ccc} & & & t & \\ & & -1 & 0 & 1 \\ \hline & -1 & 1 & 2 & 3 \\ s & 0 & 4 & 5 & 6 \\ & 1 & 7 & 8 & 9 \end{array}$$

A B

FIGURE 10.33 *An example for grayscale erosion and dilation.*

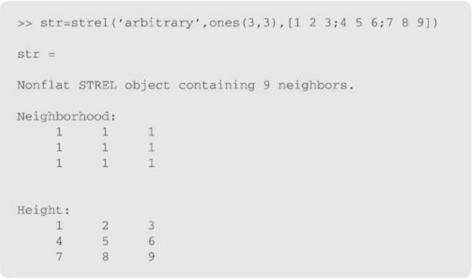
$$A \ominus B = \begin{matrix} 5 & 6 & 14 & 15 & 16 \\ 8 & 9 & 17 & 18 & 19 \\ 12 & 13 & 25 & 26 & 39 \\ 15 & 16 & 28 & 29 & 46 \\ 18 & 19 & 39 & 48 & 49 \end{matrix}$$

$$A \oplus B = \begin{matrix} 39 & 39 & 49 & 59 & 58 \\ 39 & 39 & 59 & 69 & 68 \\ 49 & 59 & 59 & 69 & 68 \\ 59 & 69 & 69 & 79 & 78 \\ 56 & 66 & 66 & 76 & 75 \end{matrix}$$

10.8 Grayscale Morphology

- The `arbitrary` parameter of `strel` allows us to create a structuring element containing any values we like

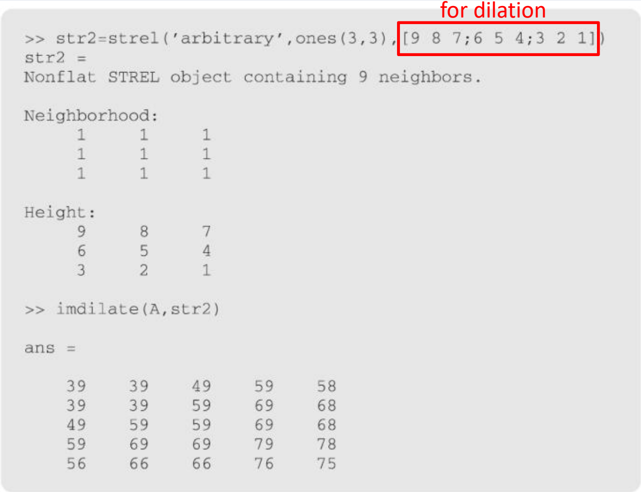
- `ones(3,3)` provides the neighborhood; the second matrix provides the values



```
>> str=strel('arbitrary',ones(3,3),[1 2 3;4 5 6;7 8 9])

str =

Nonflat STREL object containing 9 neighbors.

Neighborhood:
     1     1     1
     1     1     1
     1     1     1


Height:
     1     2     3
     4     5     6
     7     8     9
```

Ch10-p.299

10.8 Grayscale Morphology



for dilation

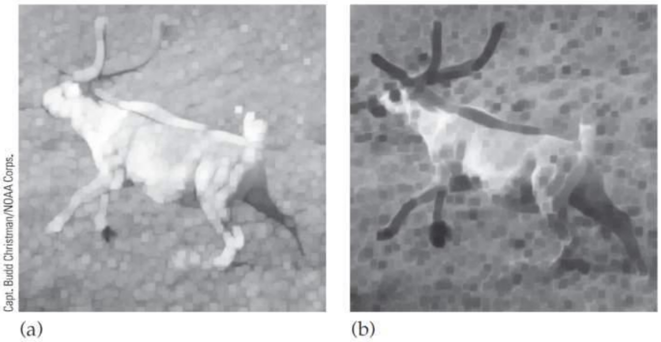
```
>> str2=strel('arbitrary',ones(3,3),[9 8 7;6 5 4;3 2 1])
str2 =
Nonflat STREL object containing 9 neighbors.

Neighborhood:
     1     1     1
     1     1     1
     1     1     1

Height:
     9     8     7
     6     5     4
     3     2     1

>> imdilate(A,str2)

ans =

    39    39    49    59    58
    39    39    59    69    68
    49    59    59    69    68
    59    69    69    79    78
    56    66    66    76    75
```

Ch10-p.300

FIGURE 10.34

```
>> c=imread('caribou.tif');
>> str=strel('square',5)
```

```
>> cd=imdilate(c,str);
>> ce=imerode(c,str);
>> imshow(cd),figure,imshow(ce)
```



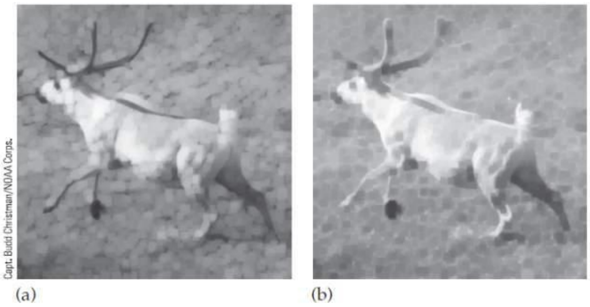
Capt. Budd Christman/NOAA Corps.

FIGURE 10.34 *Morphology. (a) Dilation. (b) Erosion.*

FIGURE 10.35

- **OPENING AND CLOSING**

```
>> co=imopen(c,str);
>> cc=imclose(c,str);
>> imshow(co),figure,imshow(cc)
```



Capt. Budd Christman/NOAA Corps.

FIGURE 10.35 *Grayscale opening and closing. (a) Opening. (b) Closing.*

10.9 Applications of Grayscale Morphology

- **10.9.1 Edge Detection**
  - ✓ morphological gradient

$$(A \oplus B) - (A \ominus B)$$

```
>> str1=strel('square',3);
>> str2=strel('square',5);
>> ce1=imerode(c,str1);
>> ce2=imerode(c,str2);
>> cd1=imdilate(c,str1);
>> cd2=imdilate(c,str2);
>> cg1=imsubtract(cd1,ce1);
>> cg2=imsubtract(cd2,ce2);
>> imshow(cg1),figure,imshow(cg2)
```

FIGURE 10.36

Capt. Budd Christman/NOAA Corps.

(a) (b)

FIGURE 10.36 *Use of the morphological gradient.*

FIGURE 10.37

- **Noise Removal**

```
>> cn=imnoise(c,'salt & pepper');
>> cf=imclose(imopen(cn,str),str);
>> imshow(cn),figure,imshow(cf)
```

Capt. Budd Christman/NOAA Corps.

(a) (b)

FIGURE 10.37 *Use of morphological filtering to remove noise.*

Ch10-p.303