Digital Image Processing
Module 4- Part 3
Morphological Image Processing

- **Morphological Image Processing:** Preliminaries, Erosion and Dilation, Opening and Closing, The Hit-or-Miss Transforms, Some Basic Morphological Algorithms.
- [Text: Chapter 9: Sections 9.1 to 9.5]

What are Morphological Operations?

Morphological operations come from the word "morphing" in Biology which means "changing a shape".

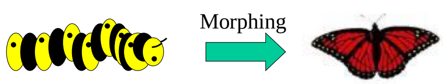


Image morphological operations are used to manipulate object shapes such as thinning, thickening, and filling.

Binary morphological operations are derived from set operations.

Preliminaries – set theory

- A be a set in **Z²**.
- a = (a_1, a_2) is an element of A. $a \in A$
- a is not an element of A $a \notin A$
- Null (empty) set: $\varnothing$

Set theory (cont.)

- Explicit expression of a set

① $A = \{a_1, a_2, ..., a_n\}$

② $A = \{element \mid \text{condition for set elements}\}$

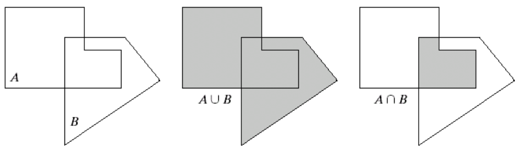
  - Example:

$C = \{w \mid w = -d, \text{ for } d \in D\}$

Set operations

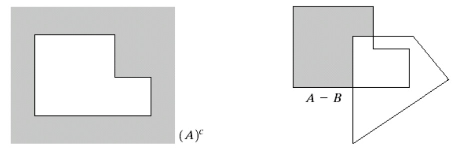
- A is a subset of B: every element of A is an element of another set B $A \subseteq B$
- Union $C = A \cup B$
- Intersection $C = A \cap B$
- Mutually exclusive $A \cap B = \varnothing$

Graphical examples



Graphical examples (cont.)

$A^c = \{w \mid w \notin A\}$ $\quad$ $A - B = \{w \mid w \in A, w \notin B\}$

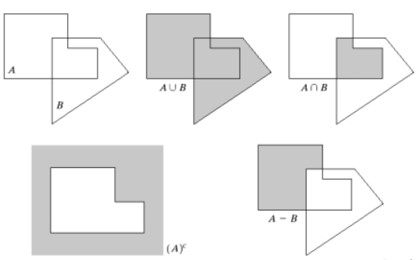


A^c= point which does not belong to set A,

Basic Set Operations

Concept of a set in binary image morphology:

Each set may represent one object. Each pixel (x,y) has its status: belong to a set or not belong to a set.



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*.

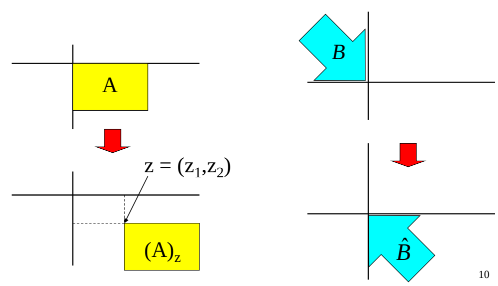
9

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2[nd] Edition.

Translation and Reflection Operations

Translation $\quad$ Reflection

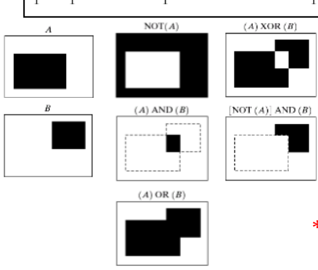
$(A)_z = \{c \mid c = a + z, \text{ for } a \in A\}$ $\quad$ $\hat{B} = \{w \mid w = -b, \text{ for } b \in B\}$



10

Logical Operations*

p	q	p AND q (also $p \cdot q$)	p OR q (also $p + q$)	NOT (p) (also $\bar{p}$)
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

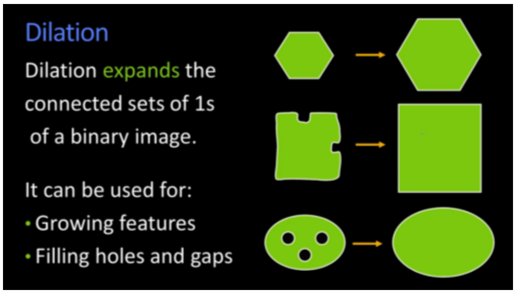


*For binary images only

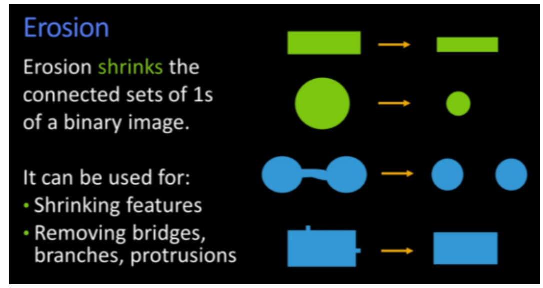
11

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2[nd] Edition.

12



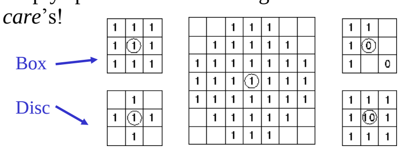
13



14

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!



Examples of stucturing elements

Dilation & Erosion

- Basic operations
- Are dual to each other:
  - Erosion shrinks foreground, enlarges Background
  - Dilation enlarges foreground, shrinks background

11-Nov-19 16

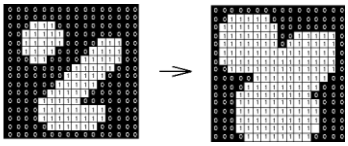
DILATION

11/11/2019 17

11/11/2019 18

Example: Dilation

- **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} }

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!!**

Example for Dilation

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

Example for Dilation

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

Example for Dilation

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

Example for Dilation

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

Example for 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				

Example for 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			

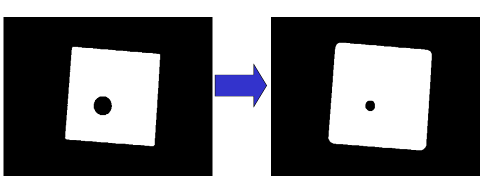
Example for Dilation

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		

Example for 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	

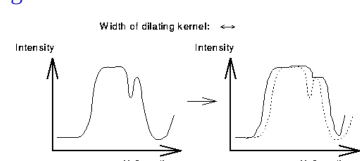
Another Dilation Example



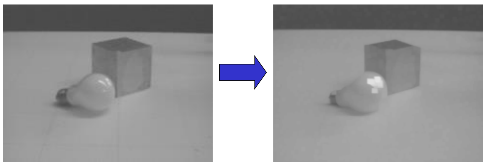
- Image get lighter, more uniform intensity

Dilation on Gray Value Images

- View gray value images as a stack of binary images!



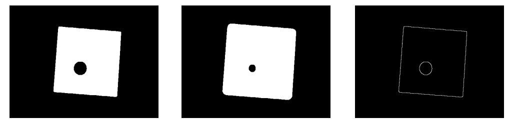
Dilation on Gray Value Images



- More uniform intensity

Edge Detection

- Edge Detection
1. Dilate input image
2. Subtract input image from dilated image
3. Edges remain!

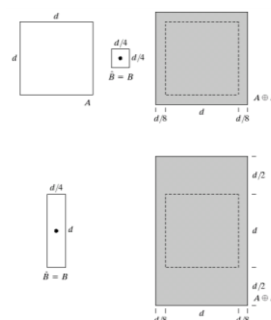


DILATION

- EXPANDS THE CONNECTED SET OF 1'S OF A BINARY IMAGE

Dilation Operations

$$A \oplus B = \{ z | (\hat{B})_z \cap A \neq \phi \}$$

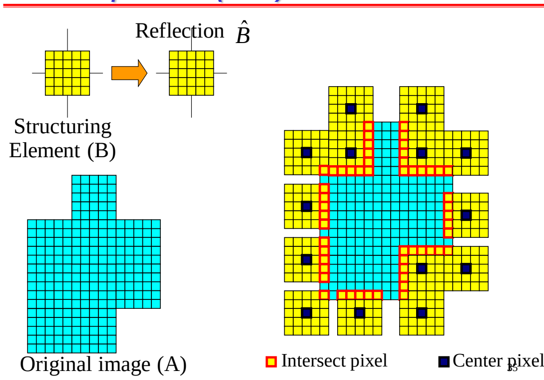


ϕ = Empty set
Dilate means "extend"

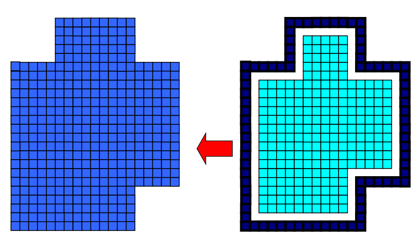
A = Object to be dilated
B = Structuring element

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Dilation Operations (cont.)



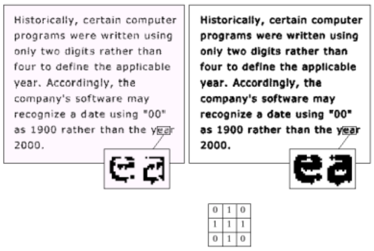
Dilation Operations (cont.)



Result of Dilation

Boundary of the "center pixels" where $(\hat{B})_z$ intersects A

Example: Application of Dilation



a c
b

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.

"Repair" broken characters

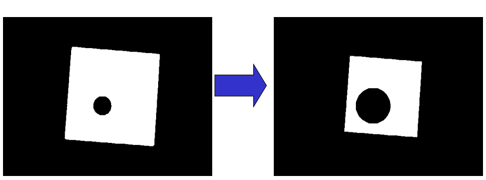
(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

EROSION

Erosion

- **Erosion** is the set of all points in the image, where the structuring element "fits into".
- Consider each foreground pixel in the input image
  - If the structuring element fits in, write a "1" at the origin of the structuring element!
- Simple application of **pattern matching**
- **Input:**
  - **Binary Image (Gray value)**
  - **Structuring Element, containing only 1s!**

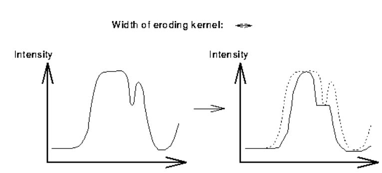
Another example of erosion



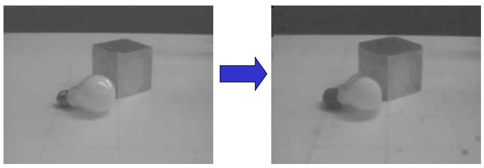
- White = 0, black = 1, dual property, image as a result of erosion gets darker

Erosion on Gray Value Images

- View gray value images as a stack of binary images!



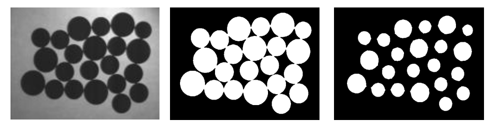
Erosion on Gray Value Images



- Images get darker!

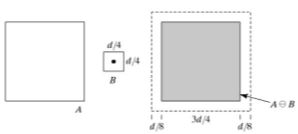
Counting Coins

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

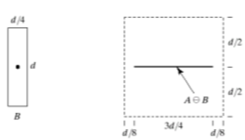


Erosion Operation

$$A \ominus B = \{z | (B)_z \subseteq A\}$$



Erosion means “trim”

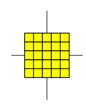


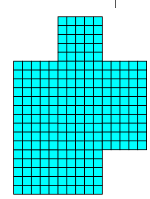
A = Object to be eroded
B = Structuring element

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

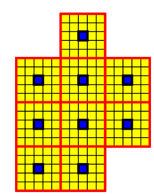
Erosion Operations (cont.)

Structuring Element (B)



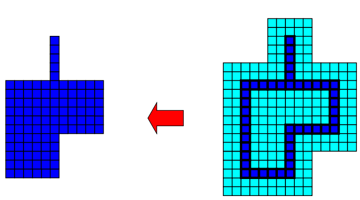


Original image (A)



Intersect pixel Center pixel

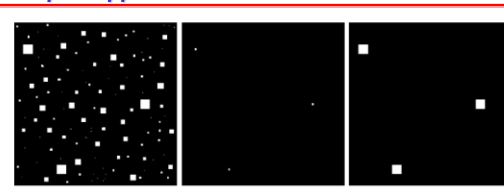
Erosion Operations (cont.)



Result of Erosion

Boundary of the “center pixels” where B is inside A

Example: Application of Dilation and Erosion



a b c

FIGURE 9.7 (a) Image of squares of size 1, 3, 5, 7, 9, and 15 pixels on the side. (b) Erosion of (a) with a square structuring element of 1's, 13 pixels on the side. (c) Dilation of (b) with the same structuring element.

Remove small objects such as noise

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Duality Between Dilation and Erosion

$$(A \ominus B)^c = A^c \oplus \hat{B}$$

where c = complement

Proof:

$$\begin{aligned}(A \ominus B)^c &= \{z | (B)_z \subseteq A\}^c \\ &= \{z | (B)_z \cap A^c = \phi\}^c \\ &= \{z | (B)_z \cap A^c \neq \phi\} \\ &= A^c \oplus \hat{B}\end{aligned}$$

Opening & Closing

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

OPENING

Opening

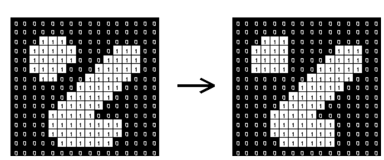
- Similar to Erosion
  - Spot and noise removal
  - Less destructive
- Erosion next dilation
- the s*ame structuring element for both operations.*
- Input:
  - Binary Image
  - Structuring Element, containing only 1s!

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!

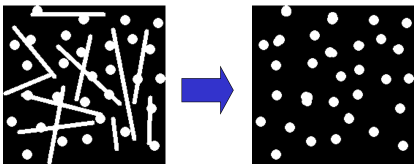
Opening

- Structuring element: 3x3 square



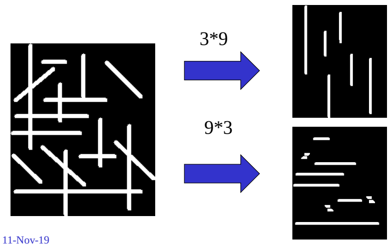
Opening Example

- Opening with a 11 pixel diameter disc



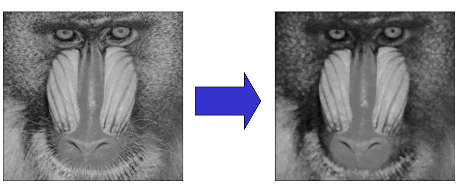
Opening Example

- 3x9 and 9x3 Structuring Element



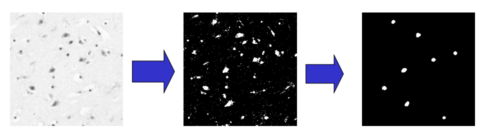
Opening on Gray Value Images

- 5x5 square structuring element



Use Opening for Separating Blobs

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



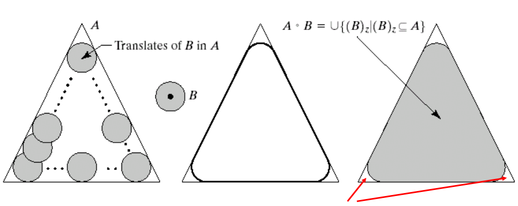
Opening Operation

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

or

$$A \circ B = \cup\{(B)_z \mid (B)_z \subseteq A\}$$

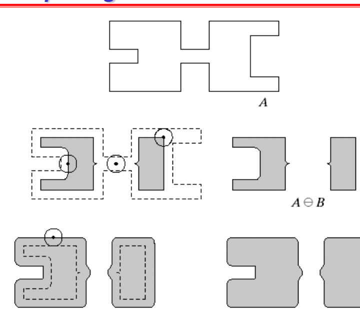
= Combination of all parts of A that can completely contain B



Opening eliminates narrow and small details and corners.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Example of Opening



(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

CLOSING

Closing

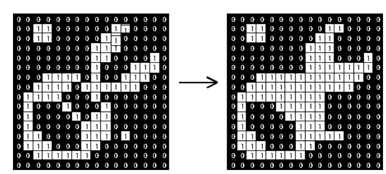
- Similar to Dilation
  - Removal of holes
  - Tends to enlarge regions, shrink background
- Closing is defined as a Dilatation, followed by an Erosion *using the same structuring element for both operations.*
- Dilation next erosion!
- Input:
  - Binary Image
  - Structuring Element, containing only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

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 can *not* be reached by the structuring element without lapping over the edge of the foreground object will be turned into a foreground.
- Opening is **idempotent:** Repeated application has no further effects!

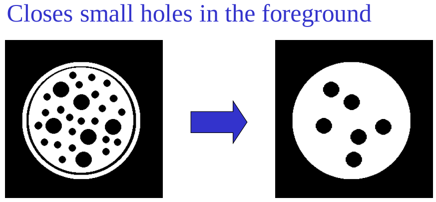
Closing

- Structuring element: 3x3 square



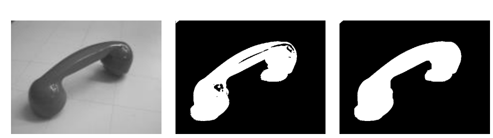
Closing Example

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



Closing Example 1

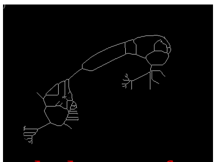
1. Threshold
2. Closing with disc of size 20



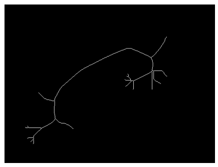
Thresholded closed

Closing Example 2

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



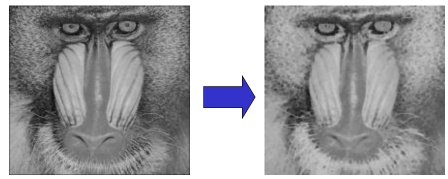
skeleton of Thresholded



skeleton of Thresholded and next closed

Closing Gray Value Images

- 5x5 square structuring element

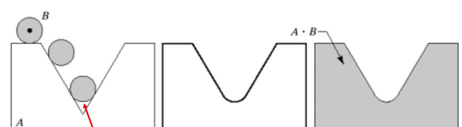


Opening & Closing

- 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.

Closing Operation

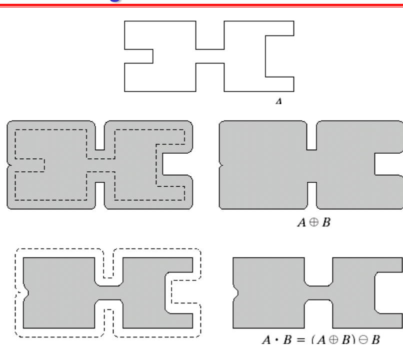
$$A \bullet B = (A \oplus B) \ominus B$$



Closing fills narrow gaps and notches

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Example of Closing



(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Duality Between Opening and Closing

$$(A \bullet B)^c = (A^c \circ \hat{B})$$

Properties Opening

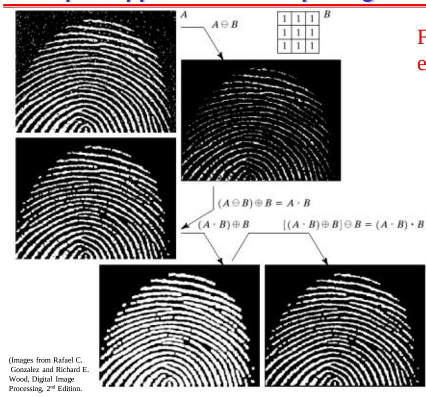
1. $A \circ B \subseteq A$
2. If $C \subset D$ then $C \circ B \subset D \circ B$
3. $(A \circ B) \circ B = A \circ B$

Properties Closing

1. $A \subseteq A \bullet B$
2. If $C \subset D$ then $C \bullet B \subset D \bullet B$
3. $(A \bullet B) \bullet B = A \bullet B$

Idem potent property: can't change any more

Example: Application of Morphological Operations



Finger print enhancement

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

HIT and MISS

Hit-and-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!!

Hit-and-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-and-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 element is set to the foreground color.

Corner Detection with Hit-and-miss Transform

- Structuring Elements representing four corners

	1	
0	1	1
0	0	

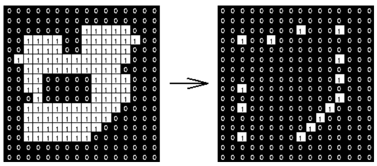
	1	
1	1	0
	0	0

	0	0
1	1	0
	1	

0	0	
0	1	1
	1	

Corner Detection with Hit-and-miss Transform

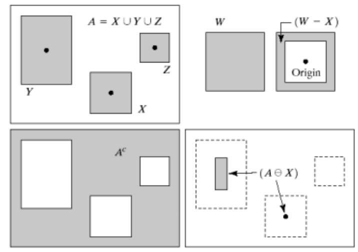
- Apply each Structuring Element
- Use OR operation to combine the four results



Hit-or-Miss Transformation

$$A \circledast X = (A \ominus X) \cap [A^c \ominus (W - X)]$$

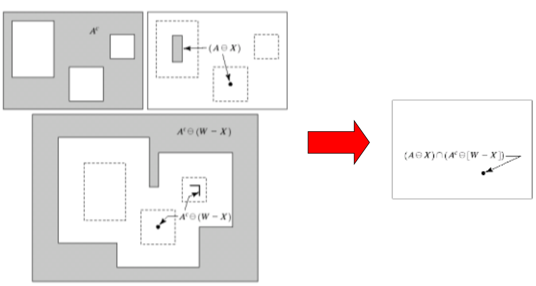
where X = shape to be detected
W = window that can contain X



(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.)

Hit-or-Miss Transformation (cont.)

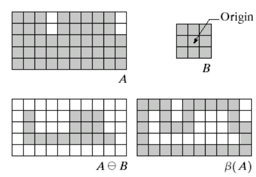
$$A \circledast B = (A \ominus B) \cap [A^c \ominus (W - X)]$$



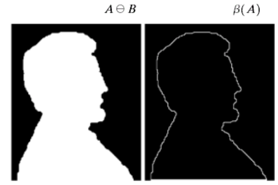
(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.)

Boundary Extraction

$$\beta(A) = A - (A \ominus B)$$



Original image



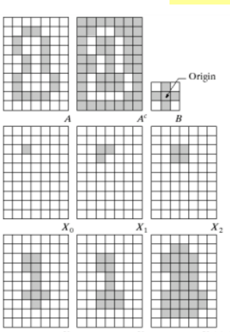
Boundary

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.)

Region Filling

$$X_k = (X_{k-1} \oplus B) \cap A^c$$

where X_0 = seed pixel p



Original image

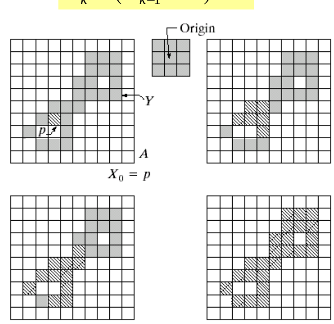
Results of region filling

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.)

Extraction of Connected Components

$$X_k = (X_{k-1} \oplus B) \cap A$$

where X_0 = seed pixel p



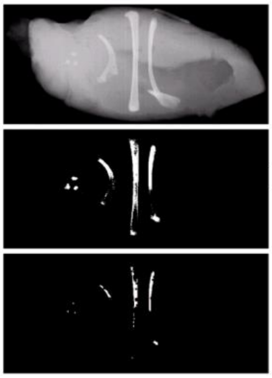
(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.)

Example: Extraction of Connected Components

X-ray image of bones

Thresholded image

Connected components



Connected component	No. of pixels in connected comp
01	11
02	9
03	9
04	39
05	133
06	1
07	1
08	743
09	7
10	11
11	11
12	9
13	9
14	674
15	85

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.)

Some Basic Morphological Algorithms (3)

- **Extraction of Connected Components**

Central to many automated image analysis applications.

$$X_k = (X_{k-1} \oplus B) \cap A$$

B : structuring element

until $X_k = X_{k-1}$

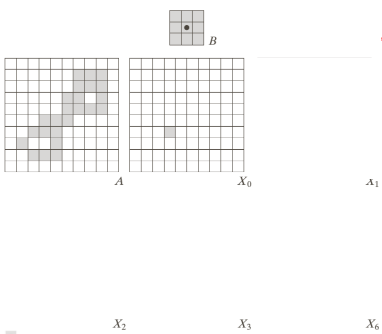


FIGURE 9.17 Extracting connected components. (a) Structuring element. (b) Array containing a set with one connected component. (c) Initial array containing a 1 in the region of the connected component. (d)–(g) Various steps in the iteration of Eq. (9.5-3).

Connected component	No. of pixels in connected comp
01	11
02	9
03	9
04	39
05	133
06	1
07	1
08	743
09	7
10	11
11	11
12	9
13	9
14	674
15	85

a
b
c d

FIGURE 9.18 (a) X-ray image of chicken filet with bone fragments. (b) Thresholded image. (c) Image eroded with a 5 × 5 structuring element of 1s. (d) Number of pixels in the connected components of (c). (Image courtesy of NTB Elektronische Geraete GmbH, Diepholz, Germany, www.ntbxray.com.)

Some Basic Morphological Algorithms (4)

- **Convex Hull**

A set A is said to be ***convex*** if the straight line segment joining any two points in A lies entirely within A.

The ***convex hull*** H or of an arbitrary set S is the smallest convex set containing S.

Some Basic Morphological Algorithms (4)

- **Convex Hull**

Let B^i, $i = 1, 2, 3, 4$, represent the four structuring elements.

The procedure consists of implementing the equation:

$$X_k^i = (X_{k-1} \circledast B^i) \cup A$$

$$i = 1, 2, 3, 4 \text{ and } k = 1, 2, 3, \ldots$$

with $X_0^i = A$.

When the procedure converges, or $X_k^i = X_{k-1}^i$, let $D^i = X_k^i$, the convex hull of A is

$$C(A) = \bigcup_{i=1}^{4} D^i$$

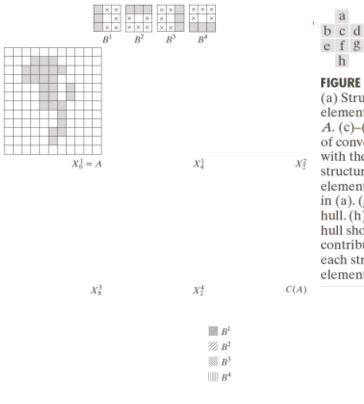


FIGURE 9.19 (a) Structuring elements. (b) Set A. (c)–(f) Results of convergence with the structuring elements shown in (a). (g) Convex hull. (h) Convex hull showing the contribution of each structuring element.

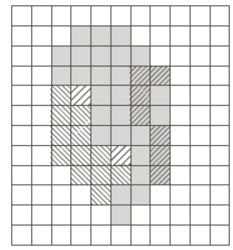
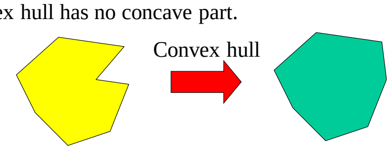


FIGURE 9.20 Result of limiting growth of the convex hull algorithm to the maximum dimensions of the original set of points along the vertical and horizontal directions.

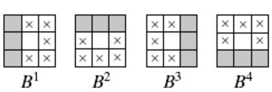
Convex Hull

Convex hull has no concave part.



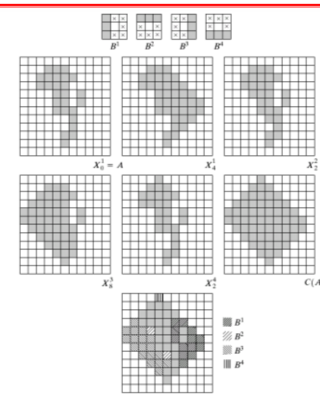
Algorithm: $C(A) = \bigcup_{i=1}^{4} D^i$ where $D^i = X^i_{conv}$

$X^i_k = (X_{k-1} \circledast B^i) \cup A, \quad i = 1,2,3,4$



(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.)

Example: Convex Hull



(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.)

Basic 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.

Definition of Thinning

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

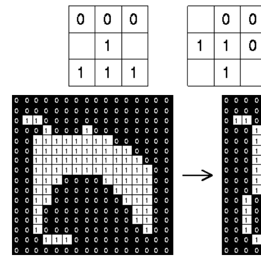
$$\text{thin}(I,K) = I - \text{HitAndMiss}(I,K)$$

with 0-1=0!!

- 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*!

Example Thinning



We use two Hit-and-miss Transforms

Basic THICKENING

Thickening

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

Definition Thickening

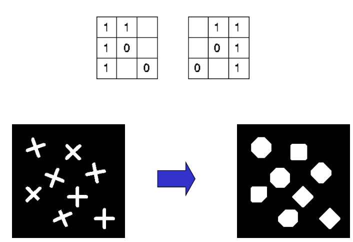
- Let ***K*** be a kernel and ***I*** be an image

$$\text{thicken}(I,K) = I + \text{HitAndMiss}(I,K)$$

with 1+1=1

- 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*!

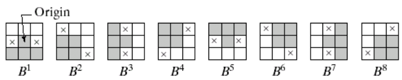
Example Thickening



Thinning

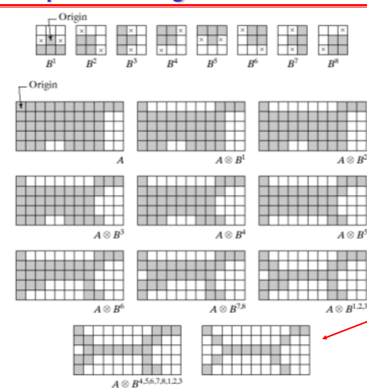
$$A \otimes B = A - (A \circledast B)$$
$$= A \cap (A \circledast B)^c$$

$$A \otimes \{B\} = ((\ldots((A \otimes B^1) \otimes B^2)\ldots) \otimes B^n)$$



(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Example: Thinning



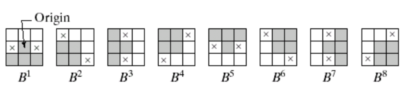
Make an object thinner.

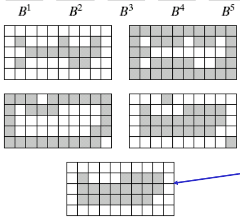
(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Thickening

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

$$A \odot \{B\} = ((\ldots((A \odot B^1) \odot B^2)\ldots) \odot B^n)$$

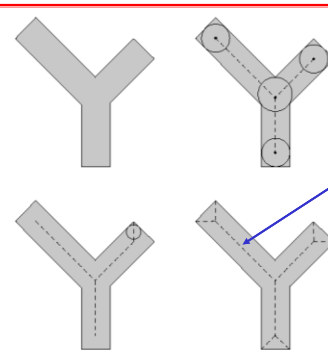




Make an object thicker

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Skeletons



Dot lines are skeletons of this structure

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Skeletons (cont.)

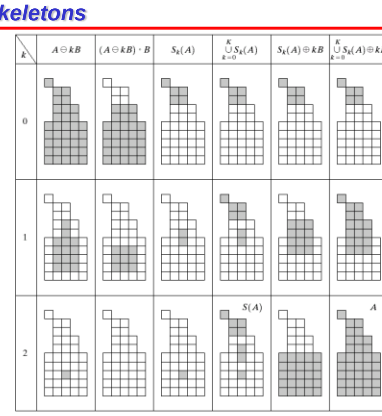
$$S(A) = \bigcup_{k=0}^{K} S_k(A)$$

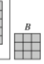
with $S_k(A) = (A \ominus kB) - (A \ominus kB) \circ B$

where $(A \ominus kB) = (\underbrace{\ldots(A \ominus B) \ominus B) \ominus \ldots) \ominus B}_{k \text{ times}}$

and $K = \max\{k \mid (A \ominus kB) \neq \phi\}$

Skeletons





(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

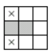
Pruning

$X_1 = A \otimes \{B\}$ = thinning

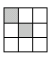
$X_2 = \bigcup_{k=1}^{8} (X_1 \circledast B^k)$ = finding end points

$X_3 = (X_2 \oplus H) \cap A$ = dilation at end points

$X_4 = X_1 \cup X_3$ = Pruned result



B^1, B^2, B^3, B^4 (rotated 90°)



B^5, B^6, B^7, B^8 (rotated 90°)

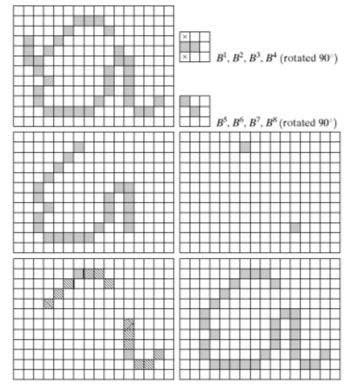
(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Example: Pruning

Original image

After Thinning 3 times

Dilation of end points



End points

Pruned result

Summary of Binary Morphological Operations

TABLE 9.2 Summary of morphological operations and their properties.

Operation	Equation	Comments (The Roman numerals refer to the structuring elements shown in Fig. 9.26).
Translation	$(A)_z = \{w \mid w = a + z, \text{ for } a \in A\}$	Translates the origin of A to point z.
Reflection	$\hat{B} = \{w \mid w = -b, \text{ for } b \in B\}$	Reflects all elements of B about the origin of this set.
Complement	$A^c = \{w \mid w \notin A\}$	Set of points not in A.
Difference	$A - B = \{w \mid w \in A, w \notin B\} = A \cap B^c$	Set of points that belong to A but not to B.
Dilation	$A \oplus B = \{z \mid (\hat{B})_z \cap A \neq \varnothing\}$	"Expands" the boundary of A. (I)
Erosion	$A \ominus B = \{z \mid (B)_z \subseteq A\}$	"Contracts" the boundary of A. (I)
Opening	$A \circ B = (A \ominus B) \oplus B$	Smoothes contours, breaks narrow isthmuses, and eliminates small islands and sharp peaks. (I)
Closing	$A \bullet B = (A \oplus B) \ominus B$	Smoothes contours, fuses narrow breaks and long thin gulfs, and eliminates small holes. (I)

(Tables from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Summary of Binary Morphological Operations (cont.)

Hit-or-miss transform	$A \circledast B = (A \ominus B_1) \cap (A^c \ominus B_2)$ $= (A \ominus B_1) - (A \oplus \hat{B}_2)$	The set of points (coordinates) at which, simultaneously, B_1 found a match ("hit") in A and B_2 found a match in A^c.
Boundary extraction	$\beta(A) = A - (A \ominus B)$	Set of points on the boundary of set A. (I)
Region filling	$X_k = (X_{k-1} \oplus B) \cap A^c$; $X_0 = p$ and $k = 1, 2, 3, \ldots$	Fills a region in A, given a point p in the region. (II)
Connected components	$X_k = (X_{k-1} \oplus B) \cap A$; $X_0 = p$ and $k = 1, 2, 3, \ldots$	Finds a connected component Y in A, given a point p in Y. (I)
Convex hull	$X_k^i = (X_{k-1}^i \circledast B^i) \cup A$; $i = 1, 2, 3, 4$; $k = 1, 2, 3, \ldots$; $X_0^i = A$; and $D^i = X_{\text{conv}}^i$.	Finds the convex hull $C(A)$ of set A, where "conv" indicates convergence in the sense that $X_k^i = X_{k-1}^i$. (III)

(Tables from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Summary of Binary Morphological Operations (cont.)

TABLE 9.2 Summary of morphological results and their properties. *(continued)*

Operation	Equation	Comments (The Roman numerals refer to the structuring elements shown in Fig. 9.26).
Thinning	$A \otimes B = A - (A \circledast B)$ $= A \cap (A \circledast B)^c$ $A \otimes \{B\} = ((\ldots((A \otimes B^1) \otimes B^2) \ldots) \otimes B^n)$ $\{B\} = \{B^1, B^2, B^3, \ldots, B^n\}$	Thins set A. The first two equations give the basic definition of thinning. The last two equations denote thinning by a sequence of structuring elements. This method is normally used in practice. (IV)
Thickening	$A \odot B = A \cup (A \circledast B)$ $A \odot \{B\} = ((\ldots(A \odot B^1) \odot B^2 \ldots) \odot B^n)$	Thickens set A. (See preceding comments on sequences of structuring elements.) Uses IV with 0's and 1's reversed.

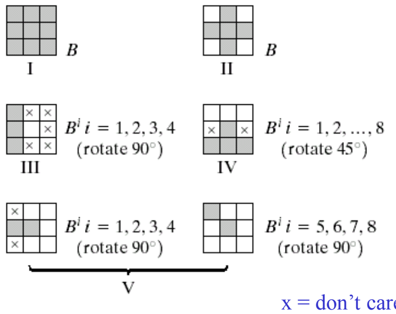
(Tables from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Summary of Binary Morphological Operations (cont.)

Skeletons	$S(A) = \bigcup_{k=0} S_k(A)$ $S_k(A) = \bigcup_{k=0}^{K} \{(A \ominus kB) - [(A \ominus kB) \circ B]\}$ Reconstruction of A: $A = \bigcup_{k=0}^{K} (S_k(A) \oplus kB)$	Finds the skeleton $S(A)$ of set A. The last equation indicates that A can be reconstructed from its skeleton subsets $S_k(A)$. In all three equations, K is the value of the iterative step after which the set A erodes to the empty set. The notation $(A \ominus kB)$ denotes the kth iteration of successive erosion of A by B. (I)
Pruning	$X_1 = A \otimes \{B\}$ $X_2 = \bigcup_{k=1}^{8} (X_1 \circledast B^k)$ $X_3 = (X_2 \oplus H) \cap A$ $X_4 = X_1 \cup X_3$	X_4 is the result of pruning set A. The number of times that the first equation is applied to obtain X_1 must be specified. Structuring elements V are used for the first two equations. In the third equation H denotes structuring element I.

(Tables from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Basic Types of Structuring Elements



x = don't care

115

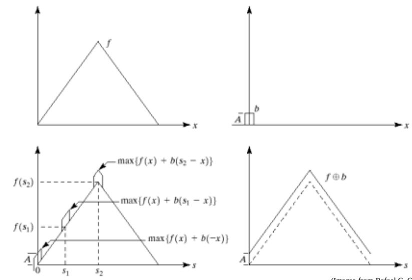
(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Gray-Scale Dilation

1-D Case $f \oplus b = \max\{f(s-x)+b(x) \mid (s-x)\in D_f \text{ and } x\in D_b\}$

2-D Case

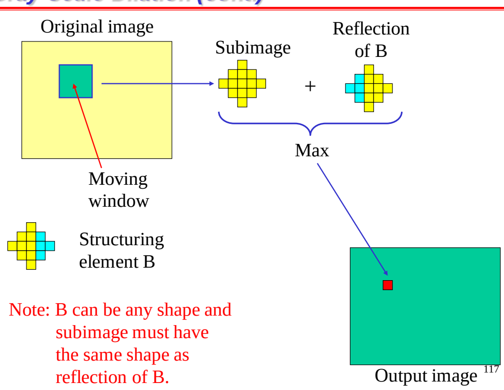
$$f \oplus b = \max\{f(s-x,t-y)+b(x,y) \mid (s-x),(t-y)\in D_f;(x,y)\in D_b\}$$



116

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Gray-Scale Dilation (cont.)



Note: B can be any shape and subimage must have the same shape as reflection of B.

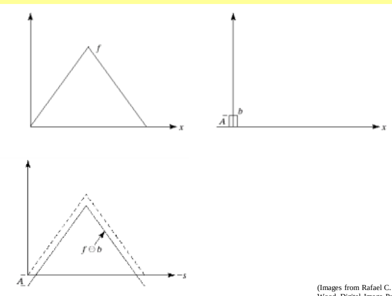
117

Gray-Scale Erosion

1-D Case $f \ominus b = \min\{f(s+x)-b(x) \mid (s+x)\in D_f \text{ and } x\in D_b\}$

2-D Case

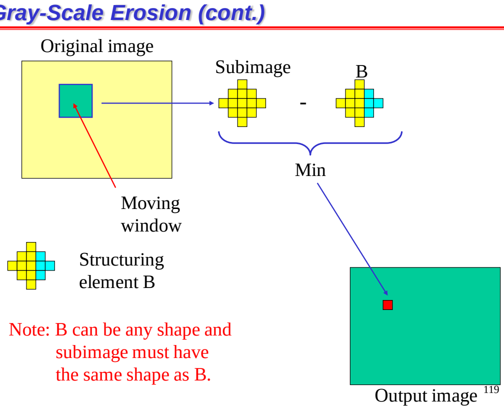
$$f \ominus b = \min\{f(s+x,t+y)-b(x,y) \mid (s+x),(t+y)\in D_f;(x,y)\in D_b\}$$



118

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

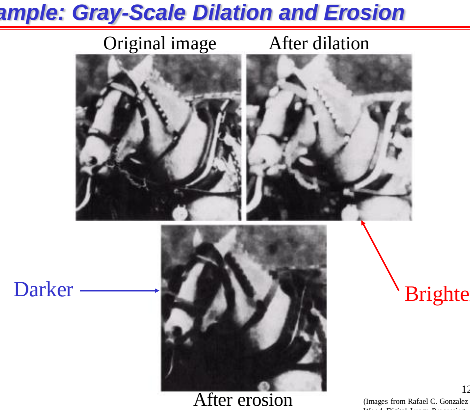
Gray-Scale Erosion (cont.)



Note: B can be any shape and subimage must have the same shape as B.

119

Example: Gray-Scale Dilation and Erosion

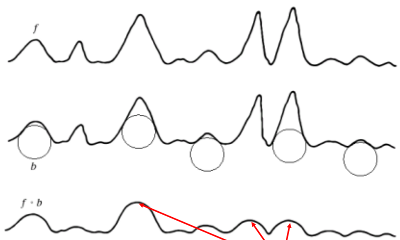


120

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Gray-Scale Opening

$$f \circ b = (f \ominus b) \oplus b$$

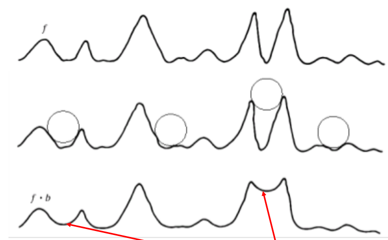


Opening cuts peaks

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Gray-Scale Closing

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



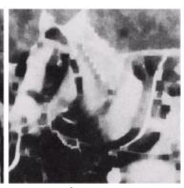
Closing fills valleys

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Example: Gray-Scale Opening and Closing


Original image


After opening


After closing

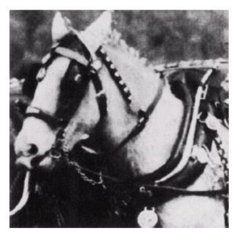
Reduce white objects

Reduce dark objects

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Gray-scale Morphological Smoothing

Smoothing: Perform opening followed by closing


Original image

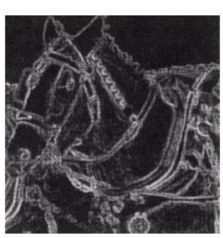

After smoothing

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Morphological Gradient

$$g = (f \oplus b) - (f \ominus b)$$


Original image

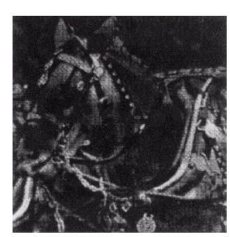

Morphological Gradient

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Top-Hat Transformation

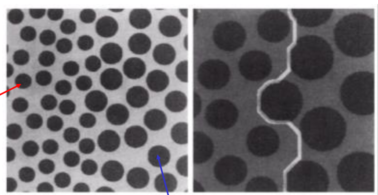
$$h = f - (f \circ b)$$


Original image


Results of top-hat transform

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Example: Texture Segmentation Application



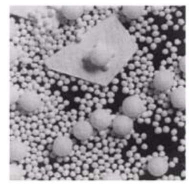
Original image   Segmented result

Algorithm:
1. Perform closing on the image by using successively larger structuring elements until small blobs are vanished.
2. Perform opening to join large blobs together
3. Perform intensity thresholding

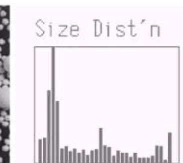
(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Example: Granulometry

Objective: to count the number of particles of each size
Method:
1. Perform opening using structuring elements of increasing size
2. Compute the difference between the original image and the result after each opening operation
3. The differenced image obtained in Step 2 are normalized and used to construct the size-distribution graph.



Original image



Size distribution graph

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Morphological Watershads

FIGURE 10.44 (a) Original image. (b) Topographic view. (c)–(d) Two stages of flooding.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Morphological Watershads

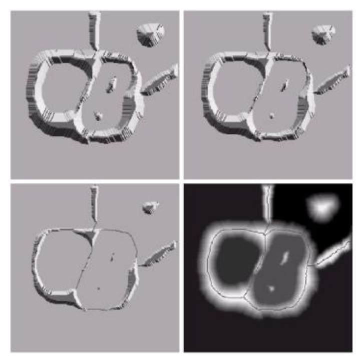


FIGURE 10.44 *(Continued)* (e) Result of further flooding. (f) Beginning of merging of water from two catchment basins (a short dam was built between them). (g) Longer dams. (h) Final watershed (segmentation) lines. (Courtesy of Dr. S. Beucher, CMM/Ecole des Mines de Paris.)

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Morphological Watershads

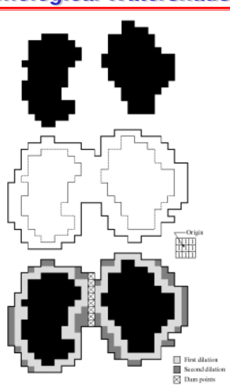
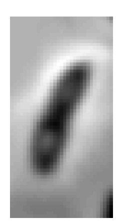


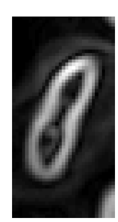
FIGURE 10.45 (a) Two partially flooded catchment basins at stage $n - 1$ of flooding. (b) Flooding at stage n, showing that water has spilled between basins (for clarity, water is shown in white rather than black). (c) Structuring element used for dilation. (d) Result of dilation and dam construction.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

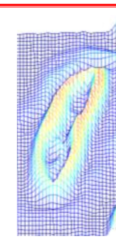
Gradient Image



Original image



$|\nabla P|$



Surface of $|\nabla P|$

$|\nabla P|$ at edges look like mountain ridges.

Morphological Watershads

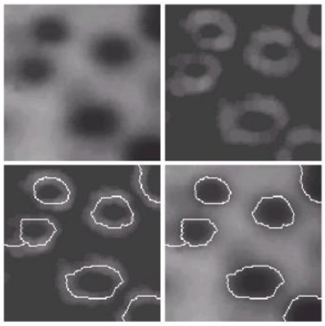


FIGURE 10.46 (a) Image of blobs. (b) Image gradient. (c) Watershed lines. (d) Watershed lines superimposed on original image. (Courtesy of Dr. S. Beucher, CMM/Ecole des Mines de Paris.)

133

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Morphological Watershads

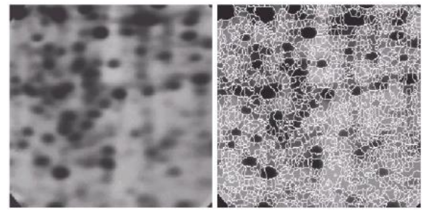


FIGURE 10.47 (a) Electrophoresis image. (b) Result of applying the watershed segmentation algorithm to the gradient image. Oversegmentation is evident. (Courtesy of Dr. S. Beucher, CMM/Ecole des Mines de Paris.)

134

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Morphological Watershads

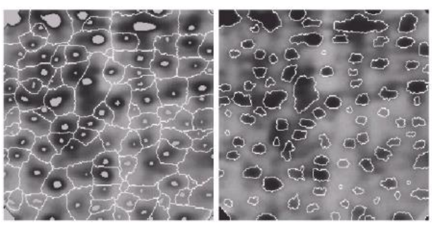


FIGURE 10.48 (a) Image showing internal markers (light gray regions) and external markers (watershed lines). (b) Result of segmentation. Note the improvement over Fig. 10.47(b). (Courtesy of Dr. S. Beucher, CMM/Ecole des Mines de Paris.)

135

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Convex Hull

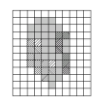


FIGURE 9.20 Result of limiting growth of convex hull algorithm to the maximum dimensions of the original set of points along the vertical and horizontal directions.

136

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.