Chapter 9: Morphological Image Processing

(Digital Image Processing – Gonzalez/Woods)

In form and in feature, face and limb,
I grew so like my brother
That folks got taking me for him
And each for one another.

Henry Sambrooke Leigh, Carols of Cockayne, The Twins

Preview

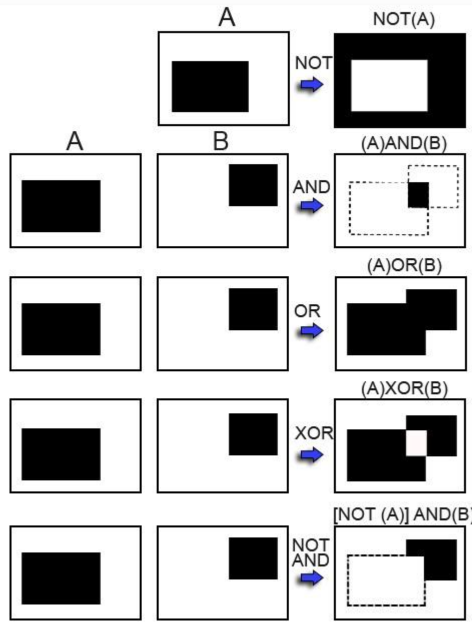
- ***"Morphology "*** – a branch in biology that deals with the form and structure of animals and plants.
- "***Mathematical Morphology***" – as a tool for extracting image components, that are useful in the representation and description of region shape.
- The language of mathematical morphology is – **Set theory**.
- Unified and powerful approach to numerous image processing problems.
- In binary images , the set elements are members of the 2-D integer space – Z^2. where each element *(x,y)* is a coordinate of a black (or white) pixel in the image.

9.1 Basic Concepts in Set Theory

- Subset
  $A \subseteq B$
- Union
  $A \cup B$
- Intersection
  $A \cap B$
  disjoint / mutually exclusive $A \cap B = \emptyset$
- Complement $A^c \equiv \{ w \mid w \notin A \}$
- Difference $A - B \equiv \{ w \mid w \in A, w \notin B \} = A \cap B^c$
- Reflection $\hat{B} \equiv \{ w \mid w = -b, \quad \forall b \in B \}$
- Translation $(A)z \equiv \{ c \mid c = a + z, \quad \forall a \in A \}$

Logic Operations Involving Binary Pixels and Images

- The principal logic operations used in image processing are: **AND, OR, NOT** (**COMPLEMENT**).
- These operations are *functionally complete*.
- Logic operations are preformed on a pixel by pixel basis between corresponding pixels (bitwise).
- Other important logic operations :
  XOR (exclusive OR), NAND (NOT-AND)
- Logic operations are just a private case for a **binary set operations**, such : AND – Intersection , OR – Union, NOT-Complement.



9.2.1 Dilation

- Dilation is used for expanding an element A by using structuring element B
- Dilation of A by B and is defined by the following equation:

$$A \oplus B = \{z | (\hat{B})z \cap A \neq \emptyset\} \quad (9.2 - 1)$$

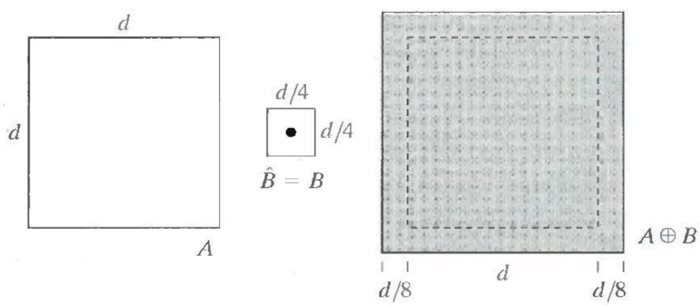
- This equation is based on obtaining the reflection of B about its origin and shifting this reflection by z.
- The dilation of A by B is the set of all displacements z, such that $\hat{B}$ and A overlap by at least one element. Based On this interpretation the equation of (9.2-1) can be rewritten as:

$$A \oplus B = \{z | [(\hat{B})z \cap A] \subset A\} \quad (9.2 - 2)$$

9.2.1 Dilation – Example 1

a b c

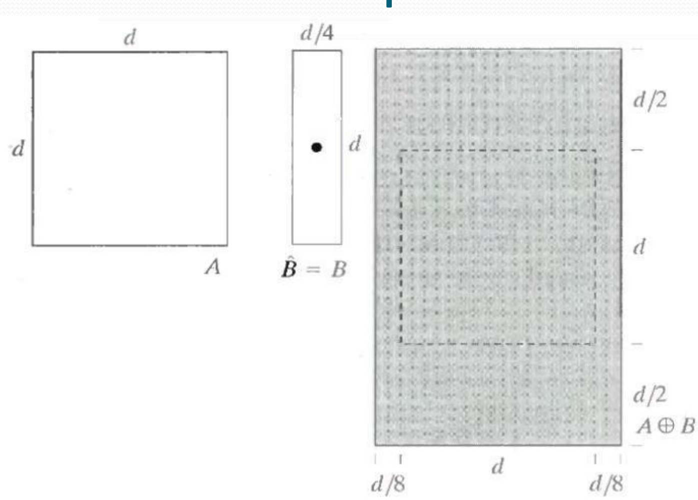
FIGURE 9.4
(a) Set A.
(b) Square structuring element (dot is the center).
(c) Dilation of A by B, shown shaded.



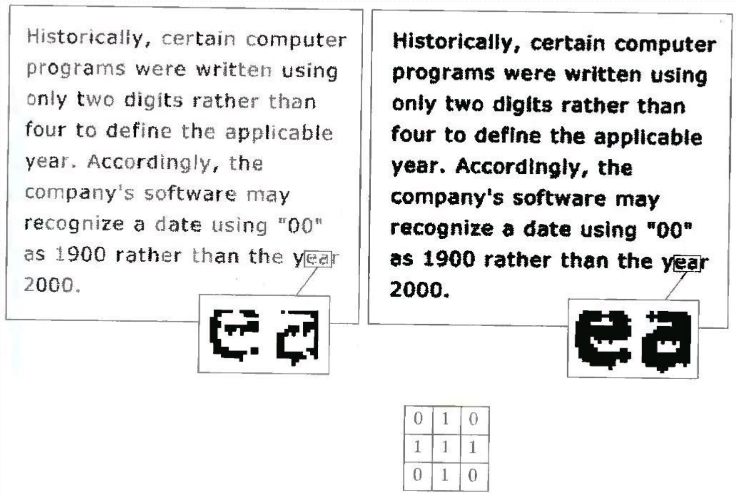
9.2.1 Dilation – Example 2

a d e

(d) Elongated structuring element.
(e) Dilation of A using this element.



9.2.1 Dilation – A More interesting Example



a b c

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.

9.2.2 Erosion

- Erosion is used for shrinking of element A by using element B
- Erosion for Sets A and B in Z^2, is defined by the following equation:

$$A \ominus B = \{z | [(B)z \subseteq A\} \quad (9.2-3)$$

- This equation indicates that the erosion of A by B is the set of all points z such that B, translated by z, is combined in A.

9.2.2 Erosion – Example 1

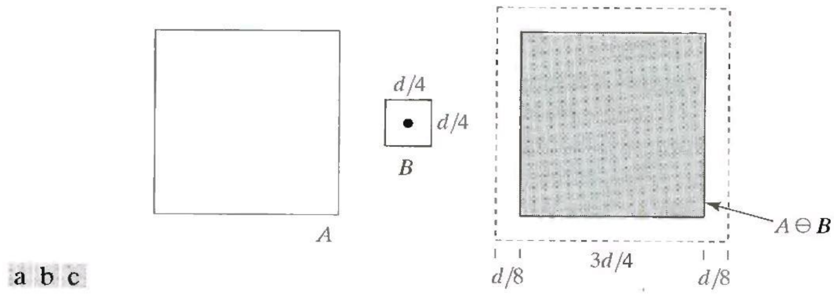


FIGURE 9.6 (a) Set A. (b) Square structuring element. (c) Erosion of A by B, shown shaded

9.2.2 Erosion – Example 2

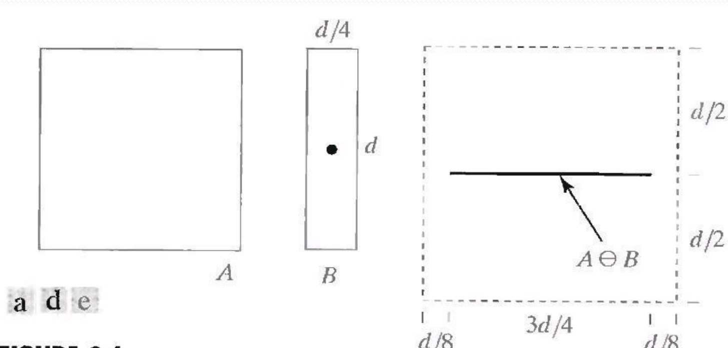


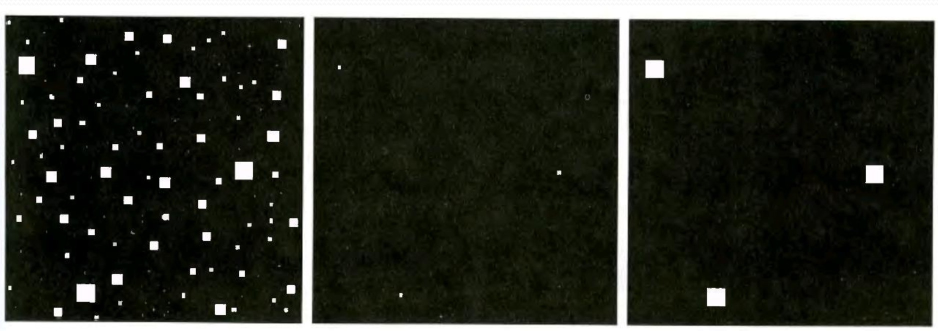
FIGURE 9.6

(a) Set A.(d) Elongated structuring element. (e) Erosion of A using this element.

Duality between dilation and erosion

- Dilation and erosion are duals of each other with respect to set complementation and reflection. That is,
  $(A \ominus B)^c = A^c \oplus \hat{B}$
- One of the simplest uses of erosion is for eliminating irrelevant details (in terms of size) from a binary image.

Erosion and Dilation summary



a b c

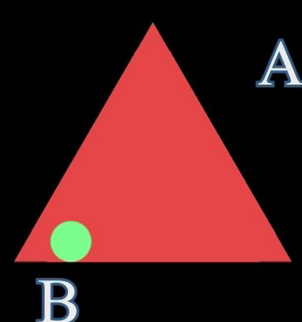
FIGURE 9.7 (a) Image of squares of size 1, 3, 5, 7, 9, and 15 pixels on the side. (b) Erosion of (a) with a square structuring element of 1's, 13 pixels on the side. (c) Dilation of (b) with the same structuring element.

9.3 Opening And Closing

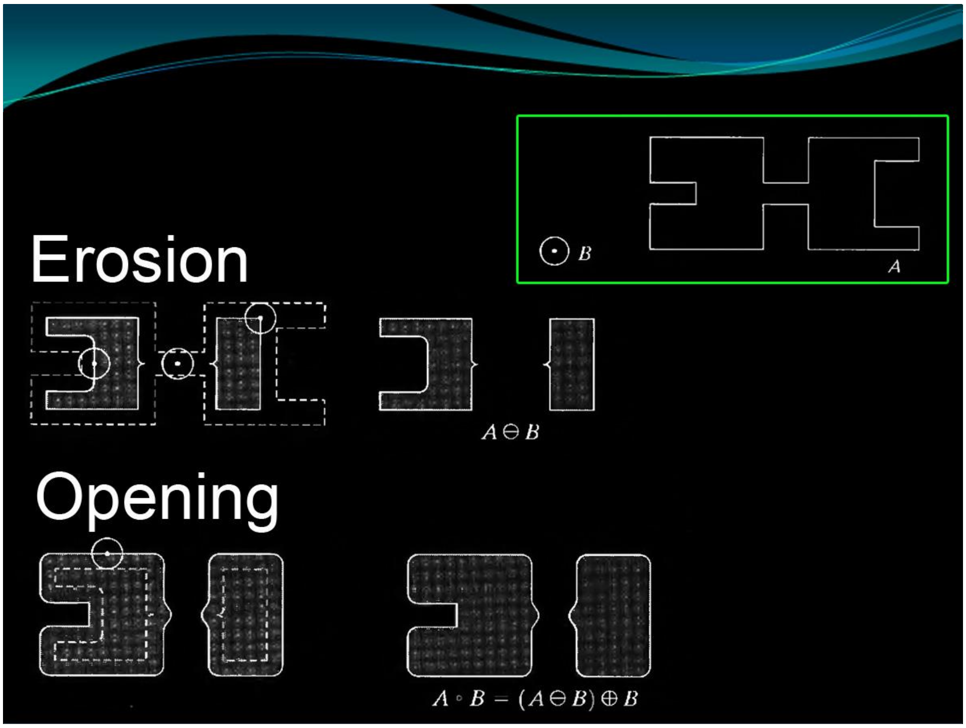
- Opening – smoothes contours , eliminates protrusions
- Closing – smoothes sections of contours, fuses narrow breaks and long thin gulfs, eliminates small holes and fills gaps in contours
- These operations are dual to each other
- These operations are can be applied few times, but has effect only once

9.3 Opening And Closing

- Opening –
  - First – erode A by B, and then dilate the result by B
  - In other words, opening is the unification of all B objects Entirely Contained in A

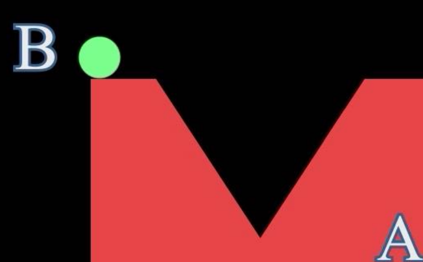


$$A \circ B = (A \ominus B) \oplus B$$



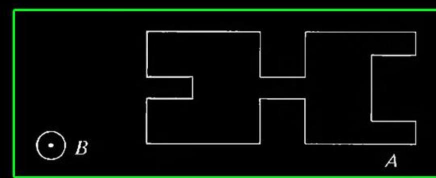
9.3 Opening And Closing

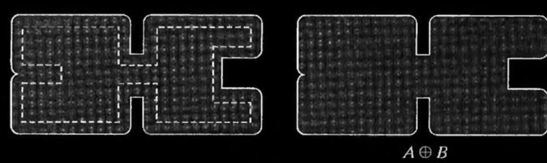
- Closing –
  - First – dilate A by B, and then erode the result by B
  - In other words, closing is the group of points, which the intersection of object B around them with object A – is not empty



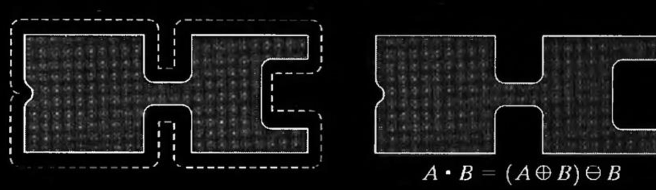
$$A \cdot B = (A \oplus B) \ominus B$$

Dilation

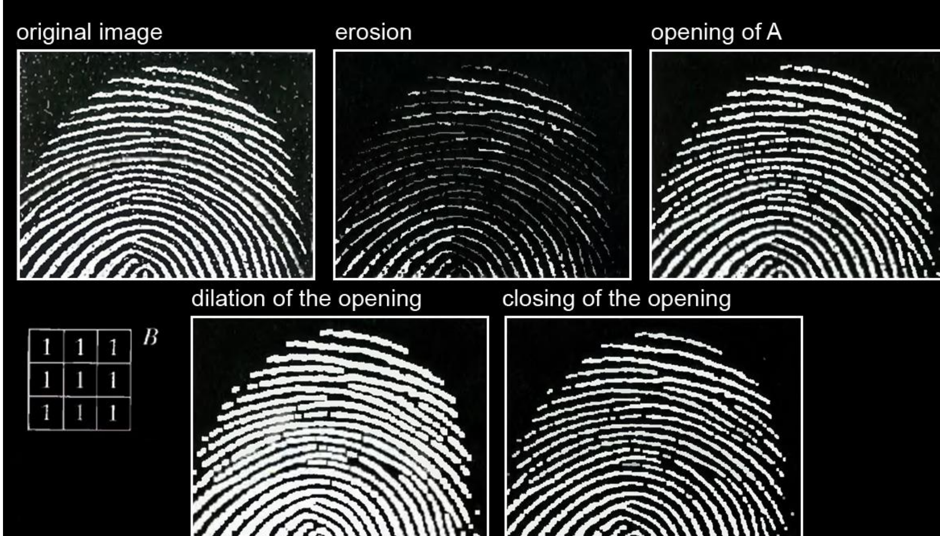




Closing



Use of opening and closing for morphological filtering



9.4 The Hit-or-Miss Transformation

- A basic morphological tool for **shape detection**.
- Let the origin of each shape be located at its center of gravity.
- If we want to find the location of a shape , say – **X** , at (larger) image, say – **A** :
  - Let X be enclosed by a small window, say – **W**.
  - The ***local background*** of X with respect to W is defined as the *set difference* (**W - X**).
  - Apply *erosion* operator of A by X, will get us the set of locations of the origin of X, such that X is completely contained in A.
  - It may be also view geometrically as the set of all locations of the origin of X at which X found a match (**hit**) in A.

9.4 The Hit-or-Miss Transformation

Cont.

- Apply *erosion* operator on the *complement of A* by the *local background* set (W – X).
- Notice, that the set of locations for which X **exactly** fits inside A is the ***intersection*** of these two last operators above.

  This intersection is precisely the location sought.

Formally:

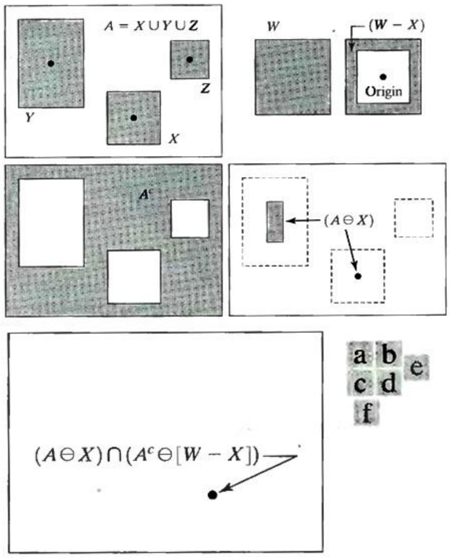
If B denotes the set composed of X and it's background –

$B = (B_1,B_2)$; $B_1 = X$, $B_2 = (W-X)$.

The match (or set of matches) of B in A, denoted $A \circledast B$ is:

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

Hit-or-Miss exp:



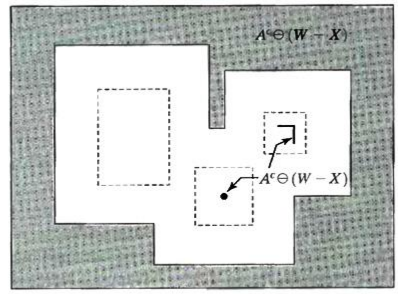


FIGURE 9.12
(a) Set A. (b) A window, W, and the local background of X with respect to W, $(W - X)$. (c) Complement of A. (d) Erosion of A by X. (e) Erosion of A^c by $(W - X)$. (f) Intersection of (d) and (e), showing the location of the origin of X, as desired.

9.4 The Hit-or-Miss Transformation

- The reason for using these kind of structuring element – B = (B1,B2) is based on an assumed definition that,

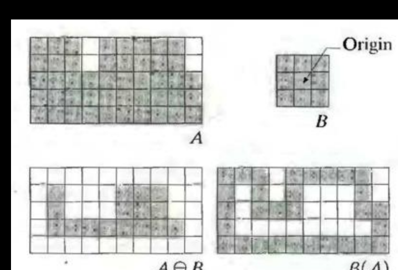
  two or more objects are distinct only if they are disjoint (disconnected) sets.
- In some applications , we may interested in detecting **certain patterns (combinations)** of 1's and 0's. and not for detecting individual objects.
- In this case <u>a background is not required</u>.
  and the *hit-or-miss transform* reduces to <u>simple *erosion*</u>.
- This simplified pattern detection scheme is used in some of the algorithms for – **identifying characters within a text**.

9.5 Basic Morphological Algorithms

- 1 – Boundary Extraction
- 2 – Region Filling
- 3 – Extraction of Connected Components
- 4 – Convex Hull
- 5 – Thinning
- 6 – Thickening
- 7 – Skeletons

9.5.1 Boundary Extraction

- First, erode A by B, then make set difference between A and the erosion
- The thickness of the contour depends on the size of constructing object – B

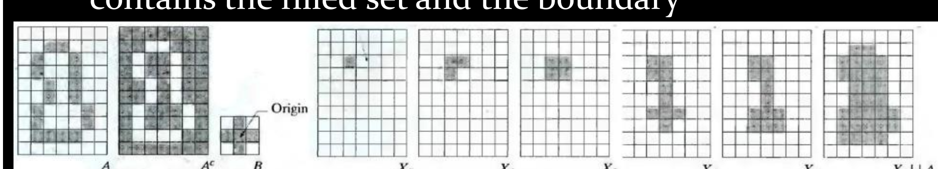


$$\beta(A) = A - (A \ominus B)$$

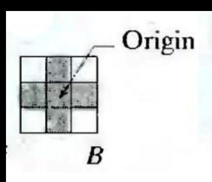
9.5.1 Boundary Extraction

9.5.2 Region Filling

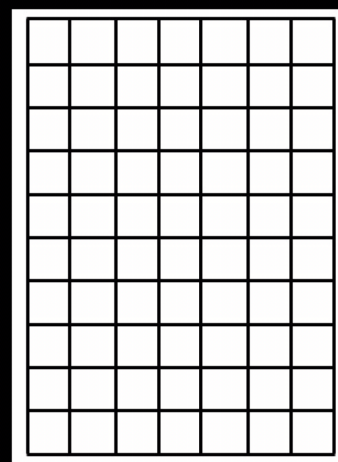
- This algorithm is based on a set of dilations, complementation and intersections
- p is the point inside the boundary, with the value of 1
- $X_{(k)} = (X_{(k-1)}$ xor B) conjunction with complemented A
- The process stops when $X_{(k)} = X_{(k-1)}$
- The result that given by union of A and $X_{(k)}$, is a set contains the filled set and the boundary



9.5.2 Region Filling



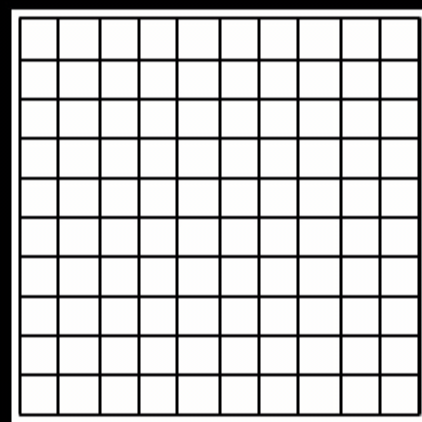
$X_k = (X_{k-1} \oplus B) \cap A^c$



9.5.3 Extraction of Connected Components

- This algorithm extracts a component by selecting a point on a binary object A
- Works similar to region filling, but this time we use in the conjunction the object A, instead of it's complement

9.5.3 Extraction of Connected Components



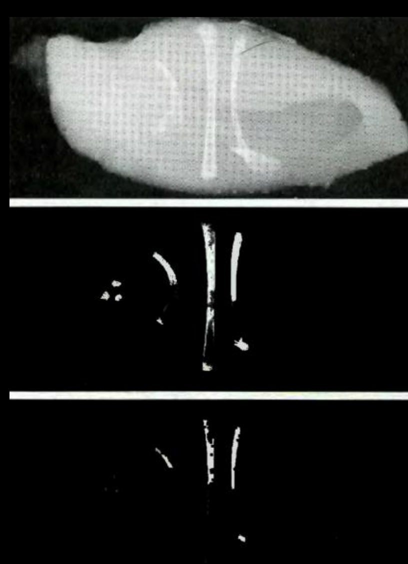
$$X_k = (X_{k-1} \oplus B) \cap A$$

This shows automated inspection of chicken-breast, that contains bone fragment

The original image is thresholded

We can get by using this algorithm the number of pixels in each of the connected components

Now we could know if this food contains big enough bones and prevent hazards

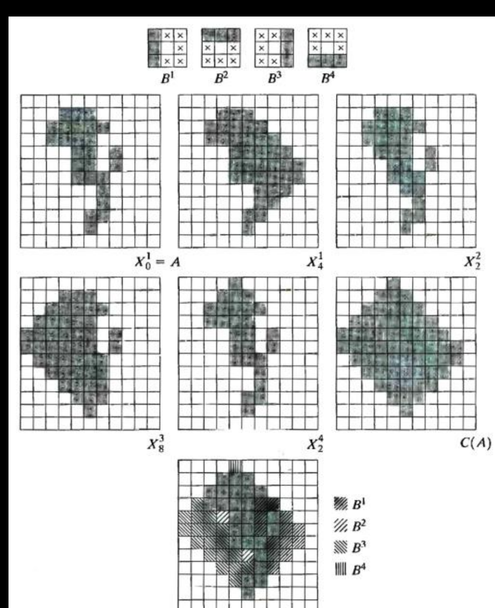


Connected component	No. of pixels in connected comp
01	11
02	9
03	9
04	39
05	133
06	1
07	1
08	743
09	7
10	11
11	11
12	9
13	9
14	674
15	85

9.5.4 Convex Hull

- A is said to be convex if a straight line segment joining any two points in A lies entirely within A
- The convex hull H of set S is the smallest convex set containing S
- Convex deficiency is the set difference H-S
- Useful for object description
- This algorithm iteratively applying the hit-or-miss transform to A with the first of B element, unions it with A, and repeated with second element of B

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



9.5.5 Thinning

- The thinning of a set A by a structuring element B, can be defined by terms of the hit-and-miss transform:

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

- A more useful expression for thinning A symmetrically is based on a sequence of structuring elements:

  $\{B\}=\{B^1, B^2, B^3, \ldots, B^n\}$

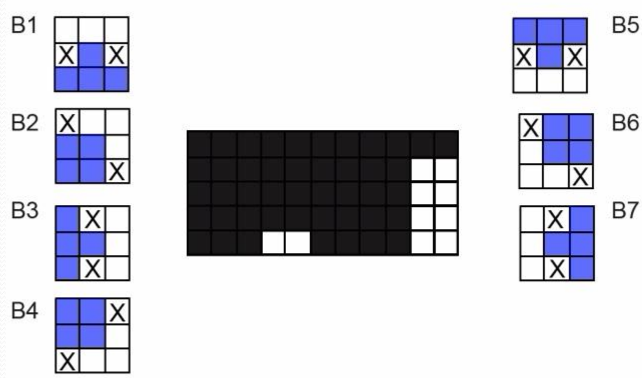
- Where B^i is a rotated version of B^{i-1}. Using this concept we define thinning by a sequence of structuring elements: $A \otimes \{B\} = ((\ldots((A \otimes B^1) \otimes B^2)\ldots) \otimes B^n)$

9.5.5 Thinning cont

- The process is to thin by one pass with B^1 , then thin the result with one pass with B^2, and so on until A is thinned with one pass with B^n.
- The entire process is repeated until no further changes occur.
- Each pass is preformed using the equation:

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

9.5.5 Thinning example



9.5.6 Thickening

- Thickening is a morphological dual of thinning.
- Definition of thickening $A \odot B = A \cup (A \circledast B)$.
- As in thinning, thickening can be defined as a sequential operation:

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

- the structuring elements used for thickening have the same form as in thinning, but with all 1's and 0's interchanged.

9.5.6 Thickening - cont

- A separate algorithm for thickening is often used in practice, Instead the usual procedure is to thin the background of the set in question and then complement the result.
- In other words, to thicken a set A, we form $C=A^c$, thin C and than form C^c.
- depending on the nature of A, this procedure may result in some disconnected points. Therefore thickening by this procedure usually require a simple post-processing step to remove disconnected points.

9.5.6 Thickening example preview

- We will notice in the next example 9.22(c) that the thinned background forms a boundary for the thickening process, this feature does not occur in the direct implementation of thickening
- This is one of the reasons for using background thinning to accomplish thickening.

9.5.6 Thickening example

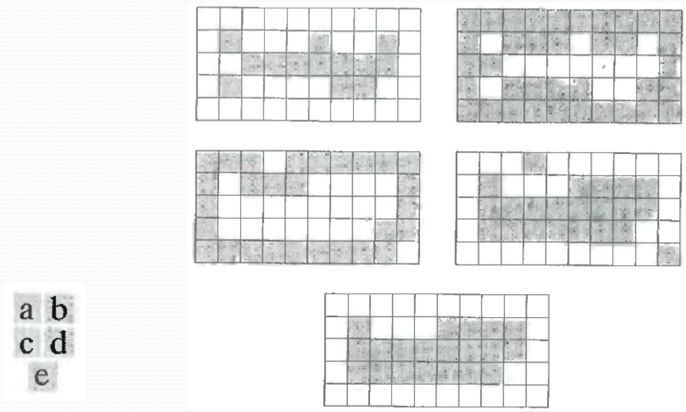


FIGURE 9.22 (a) Set *A*. (b) Complement of *A*. (c) Result of thinning the complement of *A*. (d) Thickened set obtained by complementing (c). (e) Final result, with no disconnected points.

9.5.7 Skeleton

- The notion of a skeleton S(A) of a set A is intuitively defined, we deduce from this figure that:

  a) If z is a point of S(A) and (D)z is the largest disk centered in z and contained in A (one cannot find a larger disk that fulfils this terms) – this disk is called "maximum disk".

  b) The disk (D)z touches the boundary of A at two or more different places.

9.5.7 Skeleton

- The skeleton of A is defined by terms of erosions and openings:

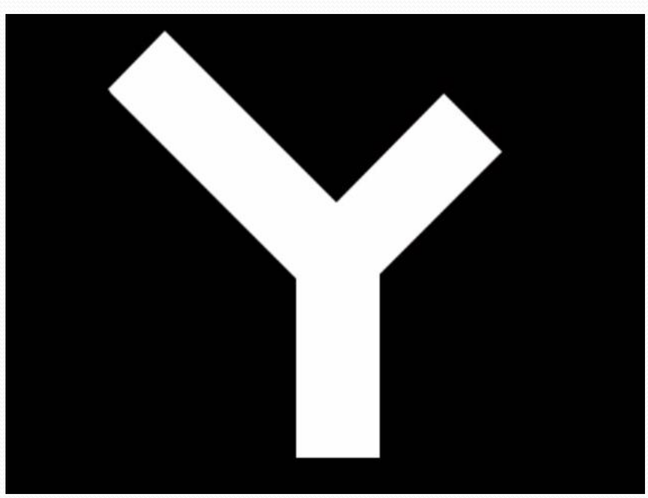
$$S(A) = \bigcup_{k=0}^{K} S_k(A)$$

- with $S_k(A) = (A \ominus kB) - (A \ominus kB) \circ B$
- Where B is the structuring element and $(A \ominus kB)$ indicates k successive erosions of A:

$$(A \ominus kB) = (\ldots((A \ominus B) \ominus B) \ominus \ldots) \ominus B$$

  - k times, and K is the last iterative step before A erodes to an empty set, in other words: $K = \max\{k | (A \ominus kB) \neq \emptyset\}$
- in conclusion S(A) can be obtained as the union of skeleton subsets Sk(A).

9.5.7 Skeleton Example



9.5.7 Skeleton

- A can be also reconstructed from subsets $S_k(A)$ by using the equation:

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

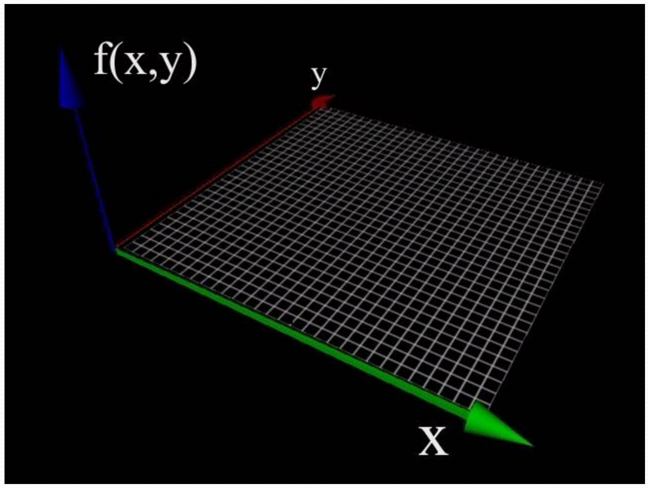
- Where $(S_k(A) \oplus kB)$ denotes k successive dilations of $S_k(A)$ that is:

$$(S_k(A) \oplus kB) = ((\ldots((S_k(A) \oplus B) \oplus B) \oplus \ldots) \oplus B$$

9.6 Gray-Scale Images

- In gray scale images on the contrary to binary images we deal with digital image functions of the form f(x,y) as an input image and b(x,y) as a structuring element.
- (x,y) are integers from Z*Z that represent a coordinates in the image.
- f(x,y) and b(x,y) are functions that assign gray level value to each distinct pair of coordinates.
- For example the domain of gray values can be 0-255, whereas 0 – is black, 255- is white.

9.6 Gray-Scale Images



9.6.1 Dilation – Gray-Scale

- Equation for gray-scale dilation is:

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

- D_f and D_b are domains of f and b.
- The condition that (s-x),(t-y) need to be in the domain of f and x,y in the domain of b, is analogous to the condition in the binary definition of dilation, where the two sets need to overlap by at least one element.

9.6.1 Dilation – Gray-Scale (cont)

- We will illustrate the previous equation in terms of 1-D. and we will receive an equation for 1 variable:

  $$(f \oplus b)(s) = \max\{f(s-x) + b(x) | (s-x) \in D_f \text{ and } x \in D_b\}$$
- The requirements the (s-x) is in the domain of f and x is in the domain of b imply that f and b overlap by at least one element.
- Unlike the binary case, f, rather than the structuring element b is shifted.
- Conceptually f sliding by b is really not different than b sliding by f.

9.6.1 Dilation – Gray-Scale (cont)

- The general effects of performing dilation on a gray scale image is twofold:

1. If all the values of the structuring elements are positive than the output image tends to be brighter than the input.
2. Dark details either are reduced or elimanted, depending on how their values and shape relate to the structuring element used for dilation

9.6.1 Dilation – Gray-Scale example

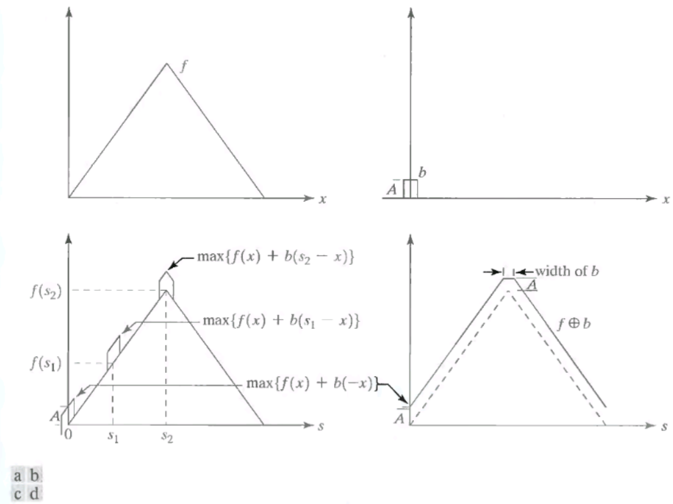


FIGURE 9.27 (a) A simple function. (b) Structuring element of height A. (c) Result of dilation for various positions of sliding b past f. (d) Complete result of dilation (shown solid).

9.6.2 Erosion – Gray-Scale

- Gray-scale erosion is defined as:

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

- The condition that (s+x),(t+y) have to be in the domain of f, and x,y have to be in the domain of b, is completely analogous to the condition in the binary definition of erosion, where the structuring element has to be completely combined by the set being eroded.
- The same as in erosion we illustrate with 1-D function

$$(f \ominus b)(s) = \min\{f(s+x) - b(x) | (s+x) \in D_f \; and \; x \in D_b\}$$

9.6.2 Erosion– Gray-Scale example 1

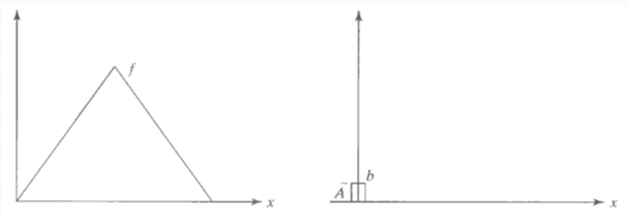
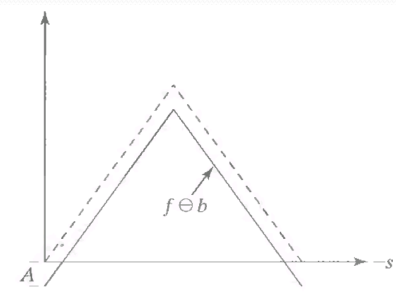


FIGURE 9.28 Erosion of the function shown in Fig. 9.27(a) by the structuring element shown in Fig. 9.27(b).

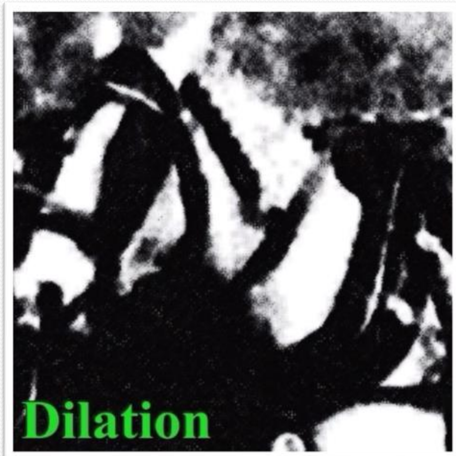


9.6.2 Erosion– Gray-Scale (cont)

- General effect of performing an erosion in grayscale images:
  1. If all elements of the structuring element are positive, the output image tends to be darker than the input image.
  2. The effect of bright details in the input image that are smaller in area than the structuring element is reduced, with the degree of reduction being determined by the grayscale values surrounding by the bright detail and by shape and amplitude values of the structuring element itself.
- Similar to binary image grayscale erosion and dilation are duals with respect to function complementation and reflection.

9.6.2 Dilation & Erosion– Gray-Scale

Over Applying the Filter



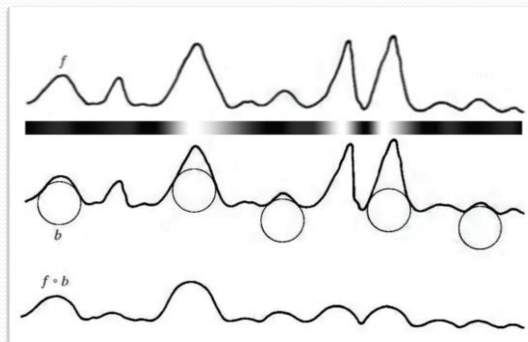
Filter Demonstration

9.6.3 Opening And Closing

- Similar to the binary algorithms
- Opening – $f \circ b = (f \ominus b) \oplus b.$
- Closing – $f \bullet b = (f \oplus b) \ominus b.$
- In the opening of a gray-scale image, we remove small light details, while relatively undisturbed overall gray levels and larger bright features
- In the closing of a gray-scale image, we remove small dark details, while relatively undisturbed overall gray levels and larger dark features

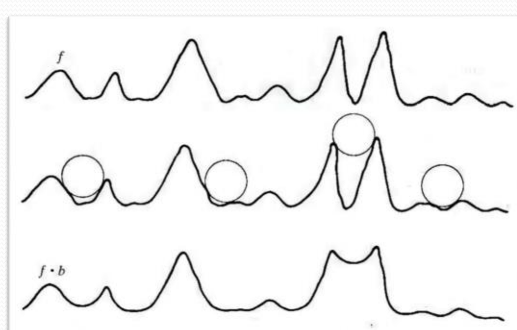
9.6.3 Opening And Closing

- Opening a G-S picture is describable as pushing object B under the scan-line graph, while traversing the graph according the curvature of B

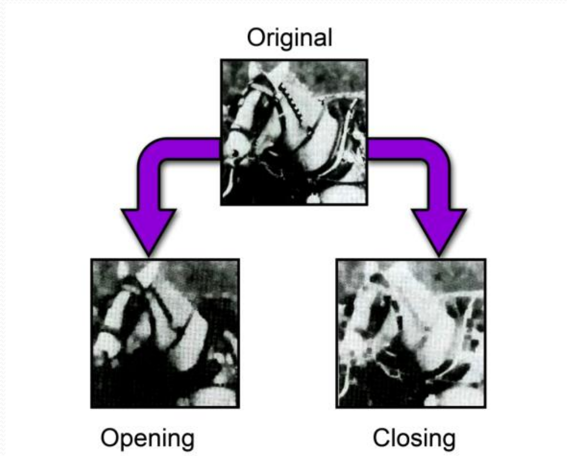


9.6.3 Opening And Closing

- Closing a G-S picture is describable as pushing object B on top of the scan-line graph, while traversing the graph according the curvature of B
- The peaks are usually remains in their original form



9.6.3 Opening And Closing



9.6.4 Some Applications of Gray-Scale Morphology

- **Morphological smoothing**
  - Perform *opening* followed by a *closing*
  - The net result of these two operations is to **remove or attenuate both bright and dark artifacts or noise**.
- **Morphological gradient**
  - *Dilation* and *erosion* are use to compute the *morphological gradient* of an image, denoted g:

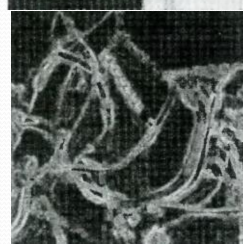
$$g = (f \oplus b) - (f \ominus b)$$

  - It uses to **highlights sharp gray-level transitions** in the input image.
  - Obtained using symmetrical structuring elements tend to depend less on edge directionality.

9.6.4 Some Applications of Gray-Scale Morphology

- **Morphological smoothing**

- **Morphological gradient**



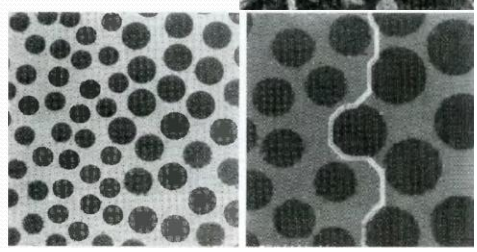
9.6.4 Some Applications of Gray-Scale Morphology

- **Top-hat transformation**
  - Denoted h, is defined as: $h = f - (f \circ b)$
  - Cylindrical or parallelepiped *structuring element function* with a flat top.
  - Useful for **enhancing detail in the presence of shading**.
- **Textural segmentation**
  - The objective is **to find the boundary between different image regions based on their textural content**.
  - *Close* the input image by using successively larger *structuring elements*.
  - Then, single *opening* is preformed ,and finally a simple *threshold* that yields the boundary between the textural regions.

9.6.4 Some Applications of Gray-Scale Morphology

- **Top-hat transformation**

- **Textural segmentation**



9.6.4 Some Applications of Gray-Scale Morphology

- **Granulometry**
  - *Granulometry* is a field that deals principally with **determining the size distribution of particles in an image.**
  - Because the particles are lighter than the background, we can use a morphological approach to determine size distribution. To construct at the end a ***histogram*** of it.
  - Based on the idea that *opening* operations of particular size have the most effect on regions of the input image that contain particles of similar size.
  - This type of processing is **useful for describing regions with a predominant particle-like character.**

9.6.4 Some Applications of Gray-Scale Morphology

- **Granulometry**

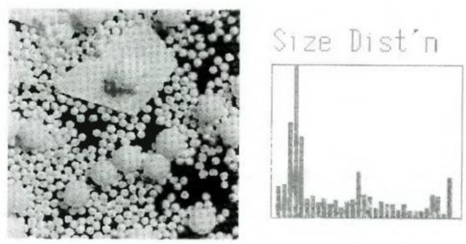


FIGURE 9.36 (a) Original image consisting of overlapping particles; (b) size distribution. (Courtesy of Mr. A. Morris, Leica Cambridge, Ltd.)

The End!