ECEN 642, Fall 2019

Texas A&M University

Electrical and Computer Engineering Department

Dr. Raffaella Righetti

Due: 11/27/2019 (before class)

Assignment #6

Gonzalez and Woods (4th edition), projects: 9.1(a), 9.4, 9.5, 9.7, 9.8(a, b), 9.9, 9.10

Gonzalez and Woods (4th edition), problems: 9.3, 9.7, 9.8, 9.9, 9.13, 9.15, 9.21, 9.22, 9.31

A more detailed book on mathematical morphology (electronic copy) can be found on Evans website: just type ***"Morphological Image Analysis: Principles and Applications"***

Starting from project 9.4 you CAN use the MATLAB function to do erosion, dilation, opening and closing if necessary. For project 9.4(b), use the figure in "HMT_example" and transform the square in the center. Please clearly indicate your processing parameters (if there is any) in your solutions

For the assignments, you will need to use MATLAB. You can access it on campus through the open access lab (OAL) or remotely through the virtual open access lab (VOAL). The link below will guide you into configuring the Horizon client for VOAL remote connection step by step on your PC or mac:

https://tamu.service-now.com/tamu-selfservice/knowledge_detail.do?sysparm_document_key=kb_knowledge,6f7e0c6adbce5f84778ff5961d96199f#

After you connect to the server, you will see the MATLAB icon. If you don't, you can connect to your VOAL desktop (VOAL icon) and start MATLAB from there.

For projects from the 4th edition, photo copies of the project statements are provided. For projects from the 3rd edition, you can access them via the link: http://www.imageprocessingplace.com/DIP-3E/dip3e_student_projects.htm#02-04 the washers. You may assume the following: (1) A "golden" (perfect with respect to the problem) image of an acceptable washer is available; and (2) the imaging and positioning components ultimately used in the system will have an accuracy high enough to allow you to ignore errors due to digitalization and positioning. You are hired as a consultant to help specify the visual inspection part of the system. Propose a solution based on morphological/logical operations.

Projects

MATLAB solutions to the projects marked with an asterisk (*) are in the DIP4E Student Support Package (consult the book website: www.ImageProcessingPlace.com).

9.1* Numerous morphological functions are based on moving the center of a structuring element (SE) over an image I and, at each location (x, y), determining how well the elements of the SE match the pixels of the corresponding neighborhood of I centered at (x, y). This is similar to the mechanics of convolution and correlation discussed in Section 3.4 (see Fig. 3.34). Let **I** be a binary image of size **M**×**N** and **B** an SE of size **m**×**n** (**m** and **n** odd) whose origin is at its canter. The elements of **B** can be: 0, corresponding to the background of **I**; 1, corresponding to the foreground; or any other value (e.g., any integer other than 0 or 1) corresponding to "don't care" values. As in convolution and correlation, **I** must be padded. To accommodate all possible excursions of **B**, pad **I** with **m** rows of **padval** above and below and **n** columns to the left and right. The padding value can be 0 (the default) or 1. The padded image, **Ip**, will be of size (**M**+**2m**)×(**N**+**2n**).

(a) Write a function, **S = morphoMatch4e(I,B,padval, mode)** that finds all matches of **B** in **I**. Output **S** has elements with three possible values: 0, meaning no matches; 0.5, meaning partial matches; and 1 meaning a perfect match. Thus, a value of 1 at coordinates **(x, y)** in **S** means that the center of **B** was at **(x, y)** when **B** and the subimage of **Ip** directly under **B** were identical. In a partial match, at least one element of **B** matches a corresponding element in **Ip**. When **S** is 0 at **(x, y)**, no elements of **B** and the corresponding elements of the subimage were equal. Elements of **B** that have "don't care" values are always forced to match their corresponding elements in **Ip**. If **mode = 'full'**, **S** will be of the same size as **Ip**. If **mode = 'same'** (the default), **S** is cropped to the same size as **I**. If **mode** is included in the input argument, **padval** must be provided also.

You can implement this function in two basic ways. If you do not have the Image Processing Toolbox in your MATLAB installation, use **for** loops. If you do have the toolbox, you may *optionally* write the function using toolbox function **colfilt**, which implements sliding neighborhoods. The first approach is the simplest (but it generally is slower). The second approach is much more difficult, but it is faster and more elegant. We give solutions using both approaches. The solution using **colfilt** is called **morphoMatch4e**. The solution using loops is called **morphoMatchLoops4e**. If you implement only the loops solution, name it **morphoMatch4e** for use in later projects.

(b) Function **morphoMatch4e** is the foundation for most of the functions you will be writing in the following projects, so test it extensively with synthetic images of your choice. In your tests, make sure you use rectangular arrays (i.e., not square) for both **I** and **B**.

9.2 Erosion and dilation.

(a)* Write a function **E = morphoErode4e(I,B,padval)** for performing morphological erosion of binary image **I** by a structuring element **B**. The specifications for **I**, **B**, and **padval**, are the same as in Project 9.1, except that all elements of **B** should be 1. A value of **padval = 1** is used, for example, when eroding the complement of **I**. Because we assume that the background is by default 0, complementing **I** turns the background into 1, so the border has to be padded with 1's in such cases. (*Hint:* Use function **morphoMatch4e** from Project 9.1.)

(b) Write a function, **D = morphoDilate4e(I,B,padval)** for performing morphological dilation. The specifications of the parameters are the same as in (a). (*Hint:* Use the duality property of erosion and dilation.)

(c)* Read the image **UTK.tif**. Using a 3×3 structuring element of 1's, erode this image successively three times and display your result.

(d)* Using the same SE as in (c), dilate the image successively three times, and display your result. Specify a single structuring element that will achieve the same result as in (c) in one pass. Confirm that your SE does what it is supposed to do by computing the difference between your result and the result in (c). Explain why your single SE gave the same result as in (c).

(e)* Obtain the complement, **Ic**, of image **I** in (c), and erode the complement using a 9×9 SE of 1's. Explain why your eroded result looks as it does. (*Hint:* Be sure to use **padval = 1** because you will be eroding the complement.)

(f) Repeat (e), but use **padval = 0**. Explain why your image looks as it does..

9.3 Opening and closing.

(a)* Write a function, **O = morphoOpen4e(I,B)** for performing morphological opening of binary image **I** by structuring element **B**. The specifications for **I** and **B**, are the same as in Project 9.2. (*Hint:* Remember, opening is based on erosion and dilation.)

(b) Write a function, **C = morphoClose4e(I,B)** for performing morphological opening of binary image **I** by structuring element **B**. **I** and **B** are as in (a). (*Hint:* Remember, closing is based on erosion and dilation.)

(c)* Read the image **circuitmask.tif** and use either opening or closing to eliminate all the connections between the center and border pads, except the vertical connections.

(d) Erosion is an alternative to eliminating the connections in (c). Do this, and use your results to explain why you would choose your approach in (c) over erosion. (*Hint:* Displaying the difference image between the results in (c) and (d) will simplify your explanation.)

9.4 Morphological hit-miss transform.

(a) Write a function **H = morphoHitmiss4e(I,B,padval, mode)** for computing the morphological hit-miss transform of a binary image, *I*, using structuring element **B**. The specifications for **I**, **B**, **padval**, and **mode** are the same as in Project 9.1. (*Hint:* Consider using function **morphoMatch4e** from Project 9.1, and keep in mind that the values of **H** have to be 1 for a perfect match and 0 for any partial match, including no match at all.)

(b)* To test your function, construct a binary image close in appearance to the image in Fig. 9.13(a) and a structuring element similar in form to the SE in Fig. 9.13(b). Duplicate the result in 9.13(c) using function **morphoHitmiss4e(I,B)**. **H** should be of the same size as **I**, and be 0's everywhere, except at the location corresponding to the center of object *D* in the figure, where it should be 1.

9.5 Boundary extraction.

(a) Write a function **BD = morphoBoundary4e(I,B)** that uses **B** to extract the boundary of the objects of foreground pixels in binary image **I**. If **B** is not included in the input argument list, it defaults to a 3×3 structuring element of 1's. [*Hint:* Refer to Eq. (9-18) and use the set intersection formulation to compute set differences, as given in Eq. (2-40).]

(b)* Read the image **testpattern512-binary.tif** whose objects are black. Compute the complement of the image so the objects are composed of 1-valued foreground pixels. Use function **morphoBoundary4e** with its default setting and display the result.

(c) Repeat (b), but without complementing the image. As in (b) the result will be an image composed of boundaries. However the two results are quite different. Explain why.

9.6 Object thinning, pruning, and end-point removal. (*Note:* The functions in this project perform a significant number of computations, and are not optimized for speed. Be patient when running them.)

(a)* Write a function, **T = morphoThin4e(I,B,numiter)** for thinning the foreground of binary image **I** using a sequence of **K**, 3×3 structuring elements contained in $3 \times 3 \times K$ array, **B**. If **numiter** is included in the function call, the function stops when this number of complete iterations through all **K** structuring elements is reached. If only **I** is included in the input argument, the structuring elements in Fig. 9.23 are used. If only **I** and **B** are provided, then structuring elements in **B** are used. If **numiter** is not provided, the procedure is allowed to proceed until convergence.

(b)* Test function **morphoThin4e** with the small image in Fig. 9.23(b) using only **I** in the input argument. You should get the same result as in Fig. 9.23(l).

(c)* Read the image **UTK.tif** and apply **morphoThin4e** to it. Observe all the spurs and y-shaped tips in the thinned characters. You will be removing those in part (f) below.

(d)* Write a function **P = morphoPrun4e(I,numthin, numdil)** to prun the foreground of binary image **I** (this typically will be a thinned image). The structuring elements and method used are as described in Fig. 9.27. Parameter **numthin** is the number of times the thinning algorithm is applied to delete end points, and **numdil** is the number of dilations done at the end. Generally, you will want **numdil** to be less that or equal to **numthin**, so add a warning statement alerting the user that unexpected results might ensue if **numdil** > **numthin**. Do not stop the program, just issue a warning.

(e)* Test **morphoPrun4e** with the small image in Fig. 9.27(a) using **numthin = numdil = 3**. You should get the same result as in that figure.

(f) Apply **morphoPrun4e** to the thinned image from (c) to remove the spurs and y-tips in the thinned image almost completely, while keeping the basic structure of the three characters. (*Hint:* Keep **numdil** much smaller than **numthin** and start with a small number for **numthin**.)

(g) Modify function **morphoPrun4e** so that all it does is remove end points from single-pixel branches. Call the modified function **Idel = morphoEndpointsDel4e(Ithin,numiter,mode)**, where **Ithin** is a thinned binary image, and **numiter** is the number of iterations through all structuring elements. If **mode = 'nosingletons'** (the default) all single, isolated points are removed at the end of **numiter** iterations. If **mode = 'leave'** the points are not removed. The latter mode is used, for example, to reduce one-pixel isolated lines and curves to single points.

(h) Apply **morphoEndpointsDel4e** with **mode = 'leave'** to the thinned image from (c) until the image is reduced to three isolated points. Do this interactively by letting **numiter = 20** each time. Display the results after each set of iterations so you can see the image being reduced. Include only the final result in your report, and indicate how many iterations it took to get there.

9.7 Geodesic dilation and erosion.

(a)* Write a function **DG = morphoGeoDilate4e(F,G,B, n)** for performing geodesic dilation based on Eqs. (9-38) and (9-39). The input arguments to this function are as explained in those equations.

(b) Write a function **EG = morphoGeoErode4e(F,G,B,n)** for performing geodesic erosion based on Eqs. (9-40) and (9-41). The input arguments to this function are as explained in those equations.

(c)* Read the image **calculator-binary.tif** and use your results from Project 9.4 and this project to delete the white upper arrow in the bottom, left area of the image. Display the image without the arrow. You will also need image **calculator-binary-upper-arrow.tif**.

(d) Read the image **testpattern512-binary.tif** and use your results from Project 9.4 and this project to delete every object from the image, except the small, black square at the top left. (*Hint:* The size of the small square is 10×10 pixels, and keep in mind that our SEs are designed to work with objects composed of foreground (1-valued) pixels.)

9.8 In this project, you are asked to write functions for morphological reconstructions by dilation and by erosion. These two functions are the basis for Projects 9.9 and 9.10.

(a)* Write a function **[RD,k] = morphoReconDilate4e(F, G,B)** based on Eq. (9-42) for computing the morphological reconstruction by dilation of marker image **F** with respect to mask image **G**, using structuring element **B**. Parameter **k** is the step at which stability was reached.

(b) Write a function **[RE,k]=morphoReconErode4e(F,G,B)** based on Eq. (9-43) for computing the morphological reconstruction by erosion of marker image **F** with respect to mask image **G**, using structuring element **B**. Parameter **k** is the step at which stability was reached.

(c)* Use your result from (a) to implement a function **[C,NC] = morphoConComp4e(I)** that finds all connected components in binary image **I** and labels the elements of each connected component with a different integer value. **C** is the array containing the connected components and **NC** is the number of connected components in **C**. (*Hint:* Take a look at the solution to Problem 9.38.)

9.9 Opening and closing by reconstruction.

(a)* Write a function **[OR,k] = morphoOpenbyRecon4e(MARKER,MASK)** for computing the morphological opening by reconstruction of **MARKER** with respect to **MASK**. This function is a generalization of Eq. (9-44). That equation can be implemented by inputting into this function the erosion of **MARKER** in place of **MARKER** and **MARKER** in place of **MASK**.

(b)* Read the image **numbers.tif** and extract from it the numbers 1 and 4. That is, your output should be a binary image of the same size as the original, but containing only the numbers 1 and 4, in their original positions with respect to the original image. You will need to extract character sizes from the image. If you are using the Image Processing Toolbox, function **impixelinfo** will do the job. If not, you can use an application like Adobe PhotoShop to perform the measurements. (*Hint:* Study Fig. 9.31.)

(c) Write a function **[CR, k] = morphoClosebyRecon4e(MARKER,MASK)** for computing the morphological closing by reconstruction of **MARKER** with respect to **MASK**. This function is a generalization of the equation for $C_R^{(n)}(F)$ in Table 9.1. That equation can be implemented by inputting into this function the dilation of **MARKER** in place of **MARKER** and **MARKER** in place of **MASK**.

9.10 Classification of objects by length. Read the image **elliptical-blobs-small.tif** and use morphological functions of your choice from those you developed in the previous projects to extract the five longest objects from the image. Your final image should contain only those objects.

9.3* Erosion of a set A by structuring element B is a subset of A, provided that the origin of B lies within B. Give an example in which the erosion $A \ominus B$ lies outside, or partially outside, A.

9.7 Dilation of a set A by structuring element B is the set of locations of the origin of B such that A contains at least one (foreground) element of B. Give an example in which the dilation of A by B lies completely outside of A. (*Hint:* Let A and B be disks of different radii.)

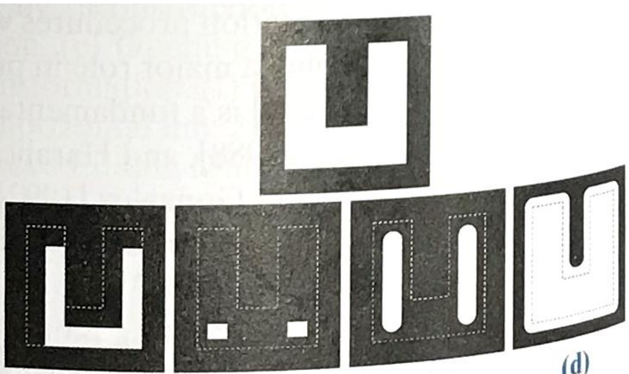
9.8 With reference to the image at the top of the figure shown below, answer the following:

(a)* Give the structuring element and morphological operation(s) that produced image (a). Show the origin of the structuring element. The dashed lines denote the boundary of the original object and are shown for reference; they are not part of the result. (The white elements are foreground pixels.)

(b) Repeat part (a) for the output shown in image (b).

(c)* Repeat part (a) for the output shown in image (c).

(d) Repeat part (a) for the solution shown in figure (d). Note that in image (d) all corners are rounded.



(a)* (b) (c)* (d)

9.9 Let A denote the set shown shaded in the following figure, and refer to the structuring elements shown (the black dots denote the origin). Sketch

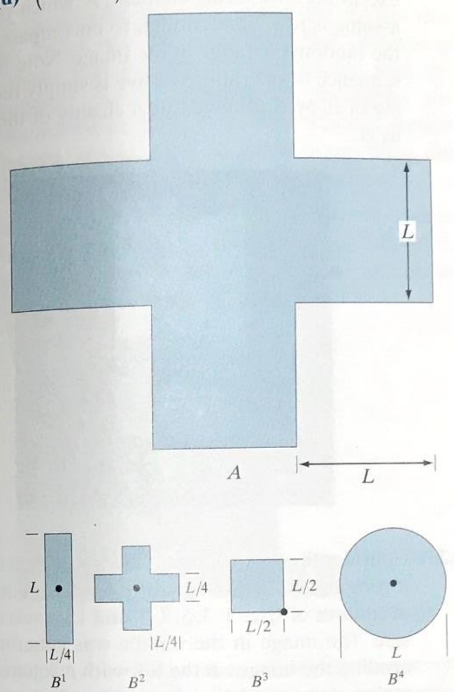
the result of the following operations:

(a)* $(A \ominus B^4) \oplus B^2$.

(b) $(A \ominus B^1) \oplus B^3$.

(c)* $(A \oplus B^1) \oplus B^3$.

(d) $(A \oplus B^3) \ominus B^2$.



9.10 Be specific in answering the following:

(a)* What is the limiting effect of repeatedly dilating a set of foreground pixels in an image? Assume that a trivial (one point) structuring element is not used.

(b) What is the smallest set from which you can start in order for your answer in (a) to hold?

9.11 Be specific in answering the following:

(a) What is the limiting effect of repeatedly eroding a set of foreground pixels in an image? Assume that a trivial (one point) structuring element is not used.

(b) What is the smallest set of foreground pixels from which you can start in order for your answer in (a) to hold?

9.12* An alternative definition of erosion is

$$A \ominus B = \{w \in Z^2 \mid w + b \in A \text{ for every } b \in B\}$$

Show that this definition is equivalent to the definition in Eq. (9-3).

9.13 Do the following:

(a) Show that the definition of erosion given in Problem 9.12 is equivalent to yet another definition of erosion:

$$A \ominus B = \bigcap_{b \in B} (A)_{-b}$$

(If $-b$ is replaced with b, this expression is called the *Minkowsky subtraction* of two sets.)

(b)* Show that the expression in (a) is equivalent to the definition in Eq. (9-3).

9.14* An alternative definition of dilation is

$$A \oplus B = \{w \in Z^2 \mid w = a + b, \text{ for some } a \in A \text{ and } b \in B\}$$

Show that this definition and the definition in Eq. (9-6) are equivalent.

9.15 Do the following:

(a) Show that the definition of dilation given in Problem 9.14 is equivalent to yet another definition of dilation:

$$A \oplus B = \bigcup_{b \in B} (A)_b$$

(This expression is called the *Minkowsky addition* of two sets.)

(b)* Show that the expression in (a) is equivalent also to the definition in Eq. (9-6).

9.16 Prove the validity of the duality expression given in Eq. (9-9).

9.17 Answer the following:

(a)* The curved portions the black border of Fig. 9.8(d) delineate the opening of set A in Fig. 9.8(a), but those curved segments are not part of the boundary of A. Are the black straight-line portions in (d) part of the boundary of A? Explain.

9.21 Show the validity of the following expressions:

(a)* $A \circ B$ is a subset of A. You may assume that Eq. (9-12) is valid. (*Hint:* Start with this equation and Fig. 9.8.)

(b)* If C is a subset of D, then $C \circ B$ is a subset of $D \circ B$. [*Hint:* Start with Eq. (9-12).]

(c) $(A \circ B) \circ B = A \circ B$. [*Hint:* Start with the definition of opening.]

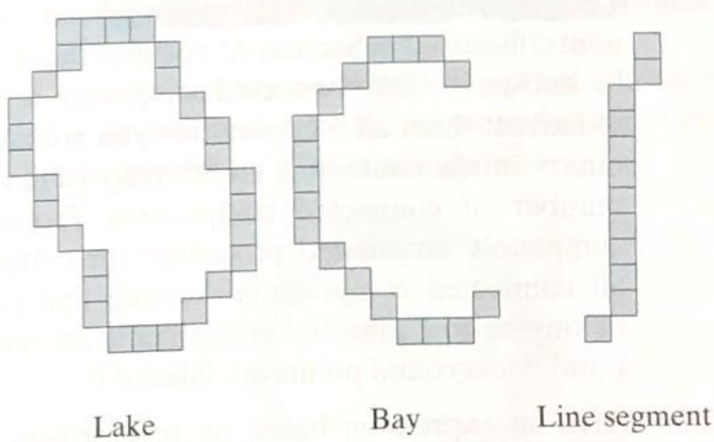
9.22 Show the validity of the following expressions. (*Hint:* Study the solution to Problem 9.21.)

(a) A is a subset of $A \bullet B$.

(b) If C is a subset of D, then $C \bullet B$ is a subset of $D \bullet B$.

(c) $(A \bullet B) \bullet B = A \bullet B$.

9.31* Three curve types (lake, bay, and line segment) useful for differentiating thinned objects in an image are shown in the following figure. Develop a morphological/logical algorithm for differentiating between these shapes. The input to your algorithm would be one of these three curves. The output must be the type of the input. You may assume that the curves are 1 pixel thick and are fully connected. However, they can appear in any orientation.



Figures & formulas

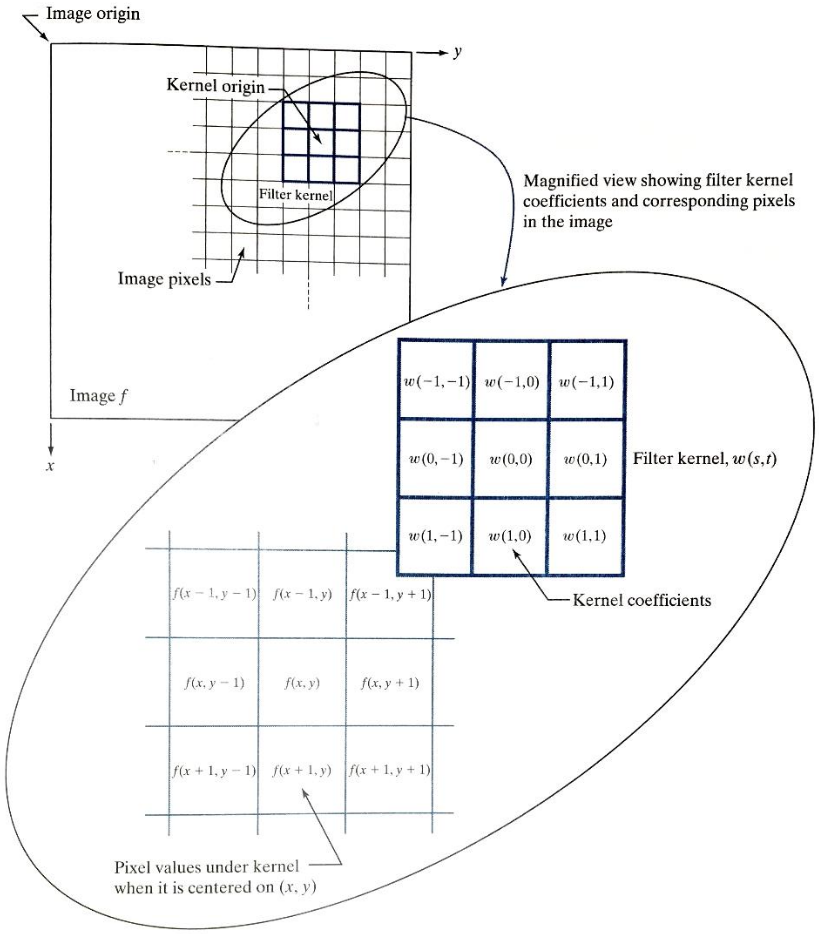
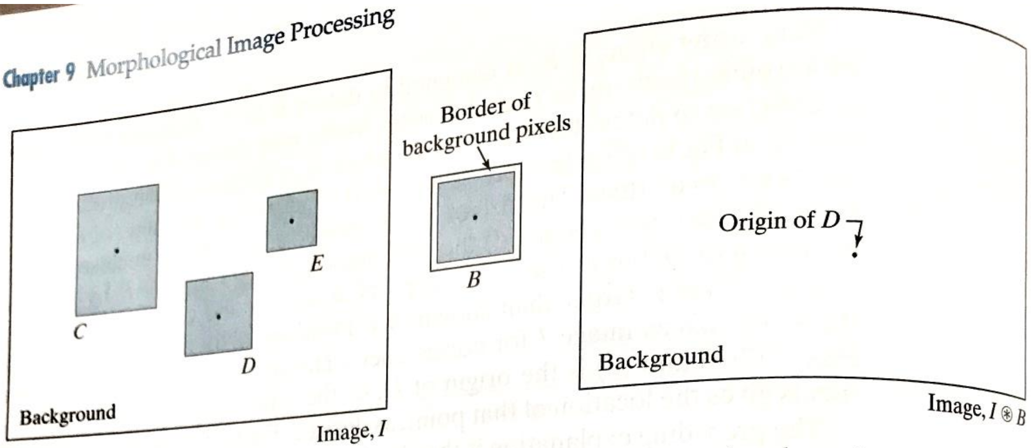


FIGURE 3.34 The mechanics of linear spatial filtering using a 3×3 kernel. The pixels are shown as squares to simplify the graphics. Note that the origin of the image is at the top left, but the origin of the kernel is at its center. Placing the origin at the center of spatially symmetric kernels simplifies writing expressions for linear filtering.

Figure 3.35(a) shows a 1-D function, f, and a kernel, w. The kernel is of size 1×5, so $a = 2$ and $b = 0$ in this case. Figure 3.35(b) shows the starting position used to perform correlation, in which w is positioned so that its center coefficient is coincident with the origin of f.

Zero padding is not the only padding option, as we will discuss in detail later in this chapter.

The first thing we notice is that part of w lies outside f, so the summation is undefined in that area. A solution to this problem is to *pad* function f with enough 0's on either side. In general, if the kernel is of size $1 \times m$, we need $(m-1)/2$ zeros on either side of f in order to handle the beginning and ending configurations of w with respect to f. Figure 3.35(c) shows a properly padded function. In this starting configuration, all coefficients of the kernel overlap valid values.



a b c

FIGURE 9.13 Same solution as in Fig. 9.12, but using Eq. (9-17) with a single structuring element.

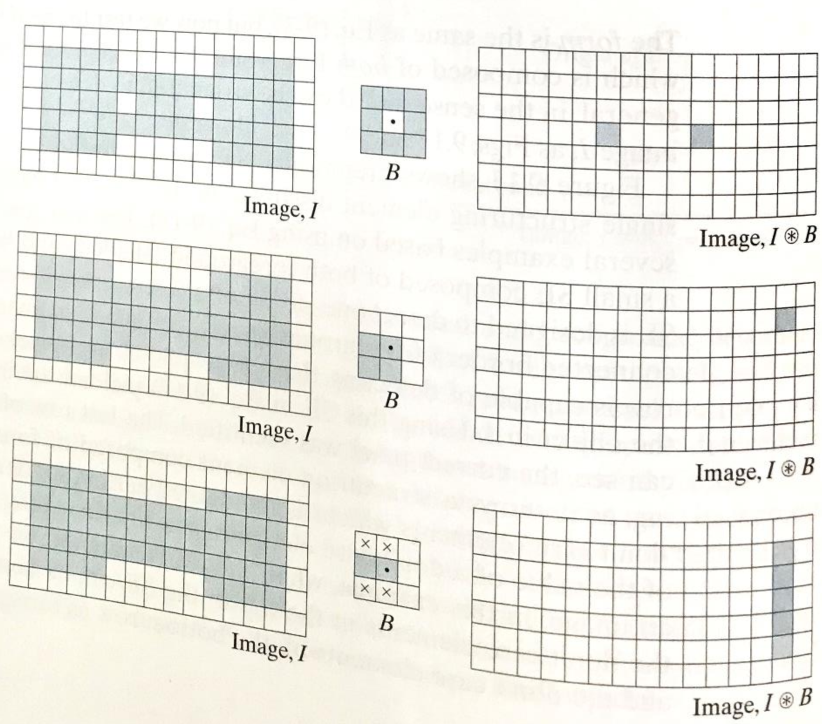
match. When the SE is centered on the bottom, right corner pixel, the role of the don't care elements is reversed, again resulting in a correct match. The other border pixels between the two corners were similarly detected by considering all don't care elements as foreground. Thus, using don't care elements increases the flexibility of structuring elements to perform multiple roles.

9.5 SOME BASIC MORPHOLOGICAL ALGORITHMS

With the preceding discussion as a foundation, we are now ready to consider some practical uses of morphology. When dealing with binary images, one of the principal applications of morphology is in extracting image components that are useful in the

a b c
d e f
g h i

FIGURE 9.14 Three examples of using a single structuring element and Eq. (9-17) to detect specific features. First row: detection of single-pixel holes. Second row: detection of an upper-right corner. Third row: detection of multiple features.



The *complement* of a set A is the set of elements that are not in A:

$$A^c = \{w \mid w \notin A\} \tag{2-39}$$

The *difference* of two sets A and B, denoted $A - B$, is defined as

$$A - B = \{w \mid w \in A, w \notin B\} = A \cap B^c \tag{2-40}$$

This is the set of elements that belong to A, but not to B. We can define A^c in terms of Ω and the set difference operation; that is, $A^c = \Omega - A$. Table 2.1 shows several important set properties and relationships.

Figure 2.35 shows diagrammatically (in so-called *Venn diagrams*) some of the set relationships in Table 2.1. The shaded areas in the various figures correspond to the set operation indicated above or below the figure. Figure 2.35(a) shows the sample set, Ω. As no earlier, this is the set of all possible elements in a given application. Figure 2.35(b) shows that the complement of a set A is the set of all elements in Ω that are not in A, which agrees with our earlier definition. Observe that Figs. 2.35(e) and (g) are identical, which proves the validity of Eq. (2-40) using Venn diagrams. This representation and description of shape. In particular, we consider morphological algorithms for extracting boundaries, connected components, the convex hull, and the skeleton of a region. We also develop several methods (for region filling, thinning, thickening, and pruning) that are used frequently for pre- or post-processing. We make extensive use in this section of "mini-images," designed to clarify the mechanics of each morphological method as we introduce it. These binary images are shown graphically with foreground (1's) shaded and background (0's) in white, as before.

BOUNDARY EXTRACTION

The boundary of a set A of foreground pixels, denoted by $\beta(A)$, can be obtained by first eroding A by a suitable structuring element B, and then performing the set difference between A and its erosion. That is,

$$\beta(A) = A - (A \ominus B) \tag{9-18}$$

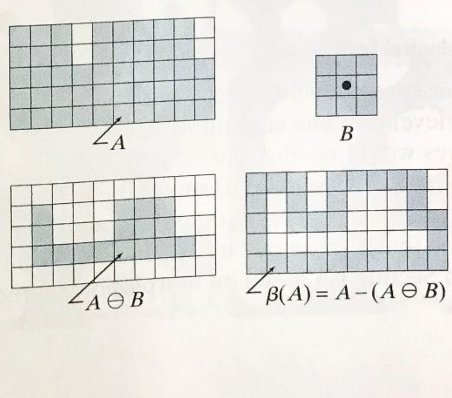
Figure 9.15 illustrates the mechanics of boundary extraction. It shows a simple binary object, a structuring element B, and the result of using Eq. (9-18). The structuring element in Fig. 9.15(b) is among the most frequently used, but it is not unique. For example, using a 5×5 structuring element of 1's would result in a boundary between 2 and 3 pixels thick. It is understood that the image in Fig. 9.15(a) was padded with a border of background elements, and that the results were cropped back to the original size after the morphological operations were completed.

EXAMPLE 9.5: Boundary extraction.

Figure 9.16 further illustrates the use of Eq. (9-18) using a 3×3 structuring element of 1's. As before when working with images, we show foreground pixels (1's) in white and background pixels (0's) in black. The elements of the SE, which are 1's, also are treated as white. Because of the size of the structuring element used, the boundary in Fig. 9.16(b) is one pixel thick.

HOLE FILLING

As mentioned in the discussion of Fig. 9.14, a *hole* may be defined as a background region surrounded by a connected border of foreground pixels. In this section, we develop an algorithm based on set dilation, complementation, and intersection for



a b
c d

FIGURE 9.15 (a) Set, A, of foreground pixels. (b) Structuring element. (c) A eroded by B. (d) Boundary of A.

9.6 MORPHOLOGICAL RECONSTRUCTION

The morphological concepts discussed thus far involve a single image and one or more structuring elements. In this section, we discuss a powerful morphological transformation called *morphological reconstruction* that involves two images and a structuring element. One image, the *marker*, which we denote by F, contains the starting points for reconstruction. The other image, the *mask*, denoted by G, constrains (conditions) the reconstruction. The structuring element is used to define connectivity.† For 2-D applications, connectivity typically is defined as 8-connectivity, which is implied by a structuring element of size 3×3 whose elements are all 1's.

See Section 2.5 regarding connectivity.

GEODESIC DILATION AND EROSION

Central to morphological reconstruction are the concepts of geodesic dilation and geodesic erosion. Let F denote the marker image and G the mask image. We assume in this discussion that both are binary images and that $F \subseteq G$. The *geodesic dilation of size* 1 of the marker image with respect to the mask, denoted by $D_G^{(1)}(F)$, is defined as

$$D_G^{(1)}(F) = (F \oplus B) \cap G \tag{9-38}$$

where, as usual, $\cap$ denotes the set intersection (here $\cap$ may be interpreted as a logical AND because we are dealing with binary quantities). The *geodesic dilation of size n* of F with respect to G is defined as

$$D_G^{(n)}(F) = D_G^{(1)}\left(D_G^{(n-1)}(F)\right) \tag{9-39}$$

where $n \geq 1$ is an integer, and $D_G^{(0)}(F) = F$. In this recursive expression, the set intersection indicated in Eq. (9-38) is performed at each step.‡ Note that the intersection operation guarantees that mask G will limit the growth (dilation) of marker F. Figure 9.28 shows a simple example of a geodesic dilation of size 1. The steps in the figure are a direct implementation of Eq. (9-38). Note that the marker F consists of just one point from the object in G. The idea is to grow (dilate) this point successively, masking of the result at each step by G. Continuing with this process would yield a result whose shape is influenced by the structure of G. In this simple case, the reconstruction would eventually result in an image identical to G (see Fig. 9.30).

The *geodesic erosion of size* 1 of marker F with respect to mask G is defined as

$$E_G^{(1)}(F) = (F \ominus B) \cup G \tag{9-40}$$

† In much of the literature on morphological reconstruction, the structuring element is tacitly assumed to be isotropic and typically is called an *elementary isotropic structuring element*. In the context of this chapter, an example of such an SE is a 3×3 array of 1's with the origin at the center.

‡ Although it is more intuitive to develop morphological reconstruction methods using recursive formulations (as we do here), their practical implementation typically is based on more computationally efficient algorithms (see, for example, Vincent [1993] and Soille [2003]).

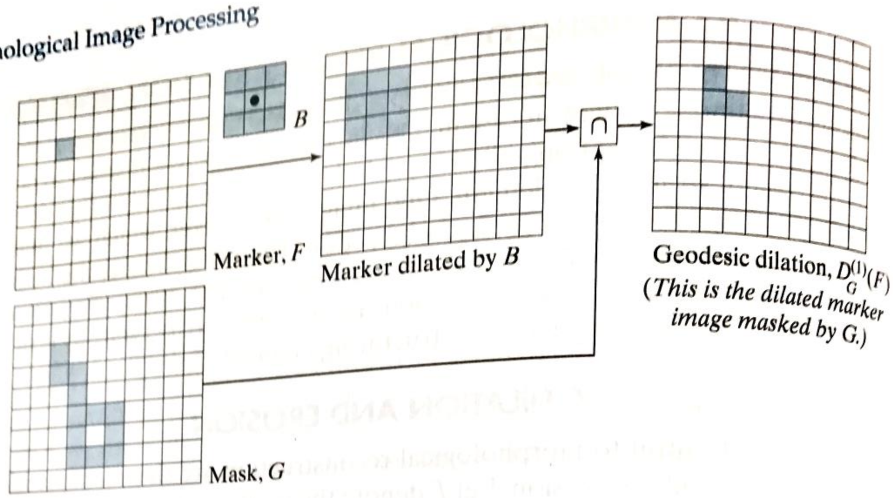


FIGURE 9.28 Illustration of a geodesic dilation of size 1. Note that the marker image contains a point from the object in G. If continued, subsequent dilations and maskings would eventually result in the object contained in G.

where $\cup$ denotes set union (or logical OR operation). The geodesic erosion of size n of F with respect to G is defined as

$$E_G^{(n)}(F) = E_G^{(1)}\left(E_G^{(n-1)}(F)\right) \tag{9-41}$$

where $n \geq 1$ is an integer and $E_G^{(0)}(F) = F$. The set union in Eq. (9-40) is performed at each step, and guarantees that geodesic erosion of an image remains greater than or equal to its mask image. As you might have expected from the forms in Eqs. (9-38) and (9-40), geodesic dilation and erosion are duals with respect to set complementation (see Problem 9.42). Figure 9.29 shows an example of a geodesic erosion of size 1. The steps in the figure are a direct implementation of Eq. (9-40).

Geodesic dilation and erosion converge after a finite number of iterative steps, because propagation or shrinking of the marker image is constrained by the mask.

MORPHOLOGICAL RECONSTRUCTION BY DILATION AND BY EROSION

Based on the preceding concepts, *morphological reconstruction by dilation* of a marker image F with respect to a mask image G, denoted $R_G^D(F)$, is defined as the geodesic dilation of F with respect to G, iterated until stability is achieved; that is,

$$R_G^D(F) = D_G^{(k)}(F) \tag{9-42}$$

with k such that $D_G^{(k)}(F) = D_G^{(k+1)}(F)$.

Figure 9.30 illustrates reconstruction by dilation. Figure 9.30(a) continues the process begun in Fig. 9.28. The next step in reconstruction after obtaining $D_G^{(1)}(F)$ is to dilate this result, then AND it with mask G to yield $D_G^{(2)}(F)$, as Fig. 9.30(b) shows. Dilation of $D_G^{(2)}(F)$ and masking with G then yields $D_G^{(3)}(F)$, and so on. This procedure is repeated until stability is reached. Carrying out this example one more step would give $D_G^{(5)}(F) = D_G^{(6)}(F)$, so the image, morphologically reconstructed by dilation, is given by $R_G^D(F) = D_G^{(5)}(F)$, as indicated in Eq. (9-42). The reconstructed image is identical to the mask, as expected.

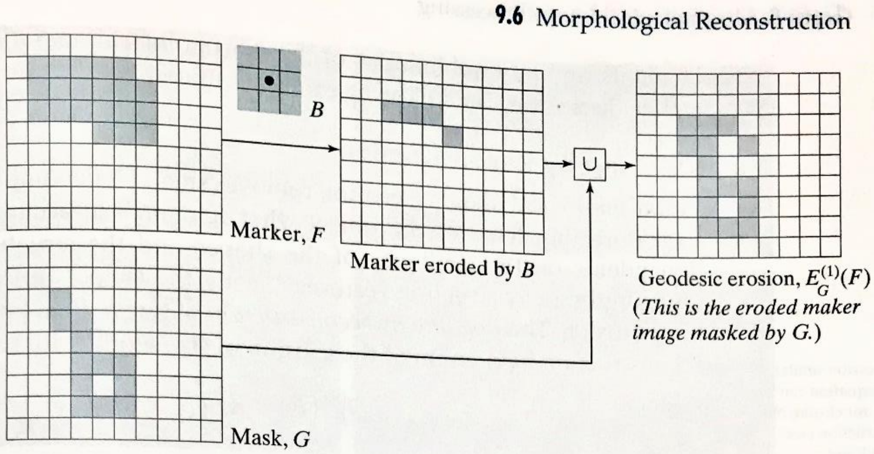


FIGURE 9.29 Illustration of a geodesic erosion of size 1.

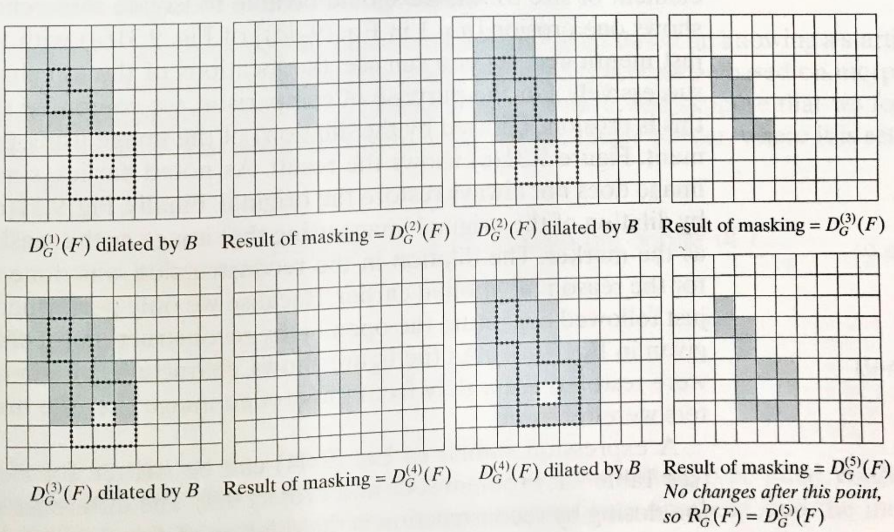
In a similar manner, the *morphological reconstruction by erosion* of a marker image F with respect to a mask image G, denoted $R_G^E(F)$, is defined as the geodesic erosion of F with respect to G, iterated until stability; that is,

$$R_G^E(F) = E_G^{(k)}(F) \tag{9-43}$$

with k such that $E_G^{(k)}(F) = E_G^{(k+1)}(F)$. As an exercise, generate a figure similar to Fig. 9.30 for morphological reconstruction by erosion. Reconstruction by dilation and erosion are duals with respect to set complementation (see Problem 9.43).

SAMPLE APPLICATIONS

Morphological reconstruction has a broad spectrum of practical applications, each determined by the selection of the marker and mask images, by the structuring



a b c d
e f g h

FIGURE 9.30 Illustration of morphological reconstruction by dilation. Sets $D_G^{(1)}(F)$, G, B and F are from Fig. 9.28. The mask (G) is shown dotted for reference.

elements, and by combinations of the morphological operations defined in the preceding discussion. The following examples illustrate the usefulness of these concepts.

Opening by Reconstruction

In morphological opening, erosion removes small objects and then dilation attempts to restore the shape of the objects that remain. The accuracy of this restoration dependents on the similarity of the shapes and the structuring element(s) used. Opening by reconstruction restores exactly the shapes of the objects that remain after erosion. The *opening by reconstruction* of size n of an image F is defined as the reconstruction by dilation of the erosion of size n of F with respect to F; that is,

A expression similar to this equation can be written for closing by reconstruction (see Table 9.1 and Problem 9.45).

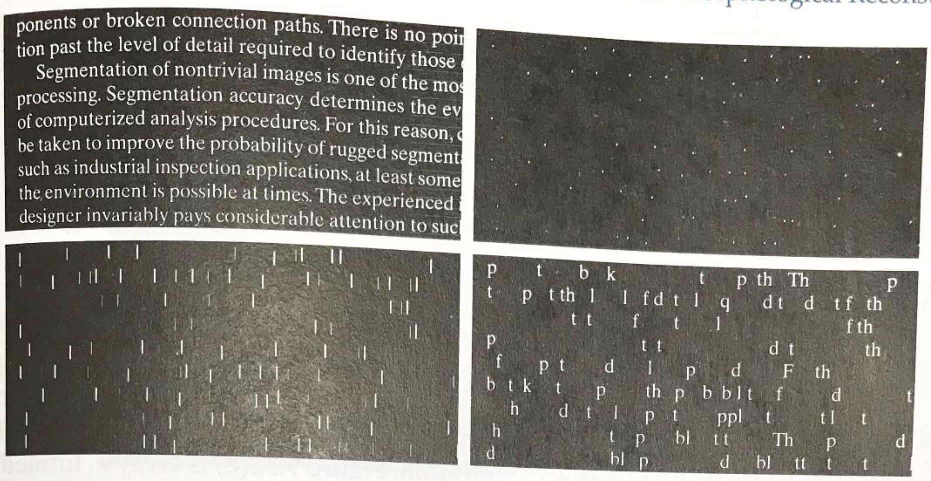
$$O_R^{(n)}(F) = R_F^D(F \ominus nB) \tag{9-44}$$

where $F \ominus nB$ indicates n erosions by B starting with F, as defined in Eq. (9-30). Note that F itself is used as the mask. By comparing this equation with Eq. (9-42), we see that Eq. (9-44) indicates that the opening by reconstruction uses an eroded version of F as the marker in reconstruction by dilation.

As you will see in Fig. 9.31, Eq. (9-44) can lead to some interesting results. Typically, the structuring element, B, used in Eq. (9-44) is designed to extract some feature of interest, based on erosion. However, as mentioned at the beginning of this section, the structuring element used in reconstruction (i.e., in the dilation that is performed to obtain R_F^D) is designed to define connectivity and, for 2-D, that structuring element typically is a 3×3 array of 1's. It is important that you do not confuse this SE with the structuring element, B, used for erosion in Eq. (9-44). Finally, we point out that this equation is most commonly used with $n = 1$.

Figure 9.31 shows an example of opening by reconstruction. We are interested in extracting from Fig. 9.31(a) the characters that contain long, vertical strokes. This objective determines the nature of B in Eq. (9-44). The average height of the tall characters in the figure is 51 pixels. By eroding the image with a thin structuring element of size 51×1, we should be able to isolate these characters. Figure 9.31(b) shows one erosion [$n = 1$ in Eq. (9-44)] of Fig. 9.31(a) with the structuring element just mentioned. As you can see, the locations of the tall characters were extracted successively. For the purpose of comparison, we computed the opening (remember this is erosion followed by the dilation) of the image using the same structuring element. Figure 9.31(c) shows the result. As noted earlier, simply dilating an eroded image does not always restore the original. Finally, Fig. 9.31(d) is the reconstruction by dilation of the original image using that image as the mask and the eroded image as the marker. The dilation in the reconstruction was done using a 3×3 SE of 1's for the reason mentioned earlier. Because we only performed one erosion, the steps just followed constitute the opening by reconstruction (of size 1) of F [i.e., $O_R^{(1)}(F)$] given in Eq. (9-44). As the figure shows, characters containing long vertical strokes were restored accurately from the eroded image (i.e., the marker); all other characters were removed.

A expression similar to Eq. (9-44) can be written for *closing by reconstruction* (see Table 9.1, Problem 9.45, and Project 9.9). The difference is that the marker used for closing by reconstruction is the dilation of F and, instead of R_F^D, we use R_F^E. As



a b
c d

FIGURE 9.31 (a) Text image of size 918×2018 pixels. The approximate average height of the tall characters is 51 pixels. (b) Erosion of (a) with a structuring element of size 51×1 elements (all 1's). (c) Opening of (a) with the same structuring element, shown for comparison. (d) Result of opening by reconstruction.

you saw, opening by reconstruction works with images in which the background is black (0) and the foreground is white (1). Closing by reconstruction works with the opposite scenario. For example, if we were working with the complement of Fig. 9.31(a), the background would be white and the foreground black. To solve the same problem of extracting the tall characters, we would use opening by reconstruction. All the other images in Fig. 9.31 would be identical, except that they would be black on white. The structuring element used would be the *same* in both cases, so the operations of closing by reconstruction would be performed on background pixels.

Automatic Algorithm for Filling Holes

In Section 9.5, we developed an algorithm for filling holes based on knowing a starting point in each hole. Here, we develop a fully automated procedure based on morphological reconstruction. Let $I(x, y)$ denote a binary image, and suppose that we form a marker image F that is 0 everywhere, except at the image border, where it is set to $1 - I$, that is,

$$F(x, y) = \begin{cases} 1 - I(x, y) & \text{if } (x, y) \text{ is on the border of } I \\ 0 & \text{otherwise} \end{cases} \tag{9-45}$$

Then,

$$H = \left[R_{I^c}^{D}(F) \right]^c \tag{9-46}$$

is a binary image equal to I with all holes filled.

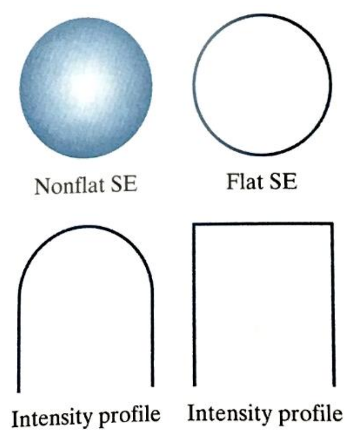
To see how Eqs. (9-45) and (9-46) cause holes in an image to be filled, consider Figs. 9.32(a) and (b), which show an image, I, containing one hole, and the image

TABLE 9.1
(Continued)

Operation	Equation	Comments
Thinning	$A \otimes B = A - (A \circledast B)$ $= A \cap (A \circledast B)^c$ $A \otimes \{B\} =$ $\left(\left(\ldots\left(\left(A \otimes B^1\right) \otimes B^2\right)\ldots\right) \otimes B^n\right)$ $\{B\} = \{B^1, B^2, B^3, \ldots, B^n\}$	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	$A \odot B = A \cup (A \circledast B)$ $A \odot \{B\} =$ $\left(\left(\ldots\left(\left(A \odot B^1\right) \odot B^2\right)\ldots\right) \odot B^n\right)$	Thickens set A using a sequence of structuring elements, as above. Uses (IV) with 0's and 1's reversed.
Skeletons	$S(A) = \bigcup_{k=0}^{K} S_k(A)$ $S_k(A) = (A \ominus kB) - (A \ominus kB) \circ B$ Reconstruction of A: $A = \bigcup_{k=0}^{K} \left(S_k(A) \oplus kB\right)$	Finds the skeleton $S(A)$ of set A. The last equation indicates that A can be reconstructed from its skeleton subsets $S_k(A)$. 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 erosions of A by B. (I)
Pruning	$X_1 = A \otimes \{B\}$ $X_2 = \bigcup_{k=1}^{8} \left(X_1 \circledast B^k\right)$ $X_3 = (X_2 \oplus H) \cap A$ $X_4 = X_1 \cup X_3$	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)
Geodesic dilation–size 1	$D_G^{(1)}(F) = (F \oplus B) \cap G$	F and G are called the *marker* and the *mask* images, respectively. (I)
Geodesic dilation–size n	$D_G^{(n)}(F) = D_G^{(1)}\left(D_G^{(n-1)}(F)\right)$	Same comment as above.
Geodesic erosion–size 1	$E_G^{(1)}(F) = (F \ominus B) \cup G$	Same comment as above.
Geodesic erosion–size n	$E_G^{(n)}(F) = E_G^{(1)}\left(E_G^{(n-1)}(F)\right)$	Same comment as above.
Morphological reconstruction by dilation	$R_G^D(F) = D_G^{(k)}(F)$	With k is such that $D_G^{(k)}(F) = D_G^{(k+1)}(F)$.

TABLE 9.1 *(Continued)*

Operation	Equation	Comments
Morphological reconstruction by erosion	$R_G^E(F) = E_G^{(k)}(F)$	With k such that $E_G^{(k)}(F) = E_G^{(k+1)}(F)$.
Opening by reconstruction	$O_R^{(n)}(F) = R_F^D(F \ominus nB)$	$F \ominus nB$ indicates n successive erosions by B, starting with F. The form of B is application-dependent.
Closing by reconstruction	$C_R^{(n)}(F) = R_F^E(F \oplus nB)$	$F \oplus nB$ indicates n successive dilations by B, starting with F. The form of B is application-dependent.
Hole filling	$H = \left[R_{I^c}^D(F)\right]^c$	H is equal to the input image I, but with all holes filled. See Eq. (9-45) for the definition of marker image F.
Border clearing	$X = I - R_I^D(F)$	X is equal to the input image I, but with all objects that touch (are connected to) the boundary removed. See Eq. (9-47) for the definition of marker image F.



a b
c d

FIGURE 9.36 Nonflat and flat structuring elements, and corresponding horizontal intensity profiles through their centers. All examples in this section are based on flat SEs.

EROSION

Remember, set A can represent (be the union of) multiple disjoint sets of foreground pixels (i.e., objects).

Morphological expressions are written in terms of structuring elements and a set, A, of foreground pixels, or in terms of structuring elements and an image, I, that contains A. We consider the former approach first. With A and B as sets in Z^2, the *erosion* of A by B, denoted $A \ominus B$, is defined as

$$A \ominus B = \left\{ z \,\middle|\, (B)_z \subseteq A \right\} \tag{9-3}$$

where A is a set of foreground pixels, B is a structuring element, and the z's are foreground values (1's). In words, this equation indicates that the erosion of A by B is the set of all points z such that B, translated by z, is contained in A. (Remember, displacement is defined with respect to the *origin* of B.) Equation (9-3) is the formulation that resulted in the *foreground* pixels of the image in Fig. 9.3(c).

As noted, we work with sets of foreground pixels embedded in a set of background pixels to form a complete image, I. Thus, inputs and outputs of our morphological procedures are images, not individual sets. We *could* make this fact explicit by writing Eq. (9-3) as

$$I \ominus B = \left\{ z \,\middle|\, (B)_z \subseteq A \text{ and } A \subseteq I \right\} \cup \left\{ A^c \,\middle|\, A^c \subseteq I \right\} \tag{9-4}$$

where I is a rectangular array of foreground and background pixels. The contents of the first braces say the same thing as Eq. (9-3), with the added clarification that A is a subset of (i.e., is contained in) I. The union with the operation inside the second set of braces "adds" the pixels that are not in subset A (i.e., A^c, which is the set of background pixels) to the result from the first braces, requiring also that the background pixels be part of the rectangle defined by I. In words, all this equation says is that erosion of I by B is the set of all points, z, such that B, translated by z, is contained in A. The equation also makes explicit that A is contained in I, that the result is embedded in a set of background pixels, and that the entire process is of the same size as I.

Of course, we do not use the cumbersome notation of Eq. (9-4), which we show only to emphasize an important point. Instead, we use the notation $A \ominus B$ when a morphological operation uses *only* foreground elements, and $I \ominus B$ when the operation uses foreground *and* background elements. This distinction may seem trivial, but suppose that we want to perform erosion with Eq. (9-3), using the foreground elements of the structuring element in the last column in Fig. 9.2. This structuring element also has background elements, but Eq. (9-3) assumes that B only has foreground elements. In fact, erosion is *defined* only for operations between foreground elements, so writing $I \ominus B$ would be meaningless without the "explanation" embedded in Eq. (9-4). To avoid confusion, we use A in morphological expressions when the operation involves only foreground elements, and I when the operation also involves background and/or "don't-care" elements. We also avoid using standard morphological symbols like $\ominus$ when working with "mixed" SEs. For example, later in Eq. (9-17) we use the symbol $\circledast$ in the expression $I \circledast B = \{z \,|\, (B)_z \subseteq I\}$, which has the same *form* as Eq. (9-3), but instead involves an entire image and the mixed-value SE in the last column of Fig. 9.2. As you will see, using SE's with mixed values adds considerable power to morphological operations.

EXAMPLE 9.1: Using erosion to remove image components.

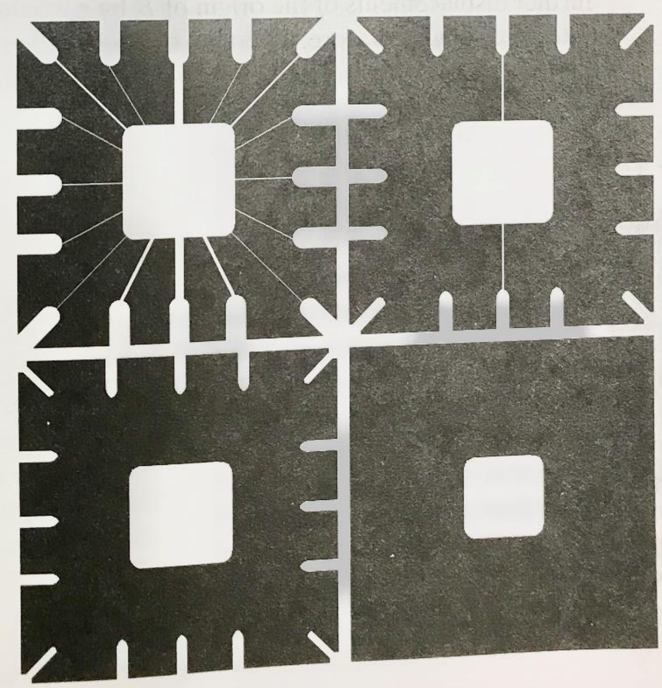
Figure 9.5(a) is a binary image depicting a simple wire-bond mask. As mentioned previously, we generally show the foreground pixels in binary images in white and the background in black. Suppose that we want to remove the lines connecting the center region to the border pads in Fig. 9.5(a). Eroding the image (i.e., eroding the *foreground* pixels of the image) with a square structuring element of size 11×11 whose components are all 1's removed most of the lines, as Fig. 9.5(b) shows. The reason that the two vertical lines in the center were thinned but not removed completely is that their width is greater than 11 pixels. Changing the SE size to 15×15 elements and eroding the original image again did remove all the connecting lines, as Fig. 9.5(c) shows. An alternate approach would have been to erode the image in Fig. 9.5(b) again, using the same 11×11, or smaller, SE. Increasing the size of the structuring element even more would eliminate larger components. For example, the connecting lines and the border pads can be removed with a structuring element of size 45×45 elements applied to the original image, as Fig. 9.5(d) shows.

We see from this example that erosion shrinks or thins objects in a binary image. In fact, we can view erosion as a *morphological filtering* operation in which image details smaller than the structuring element are filtered (removed) from the image. In Fig. 9.5, erosion performed the function of a "line filter." We will return to the concept of morphological filters in Sections 9.4 and 9.8.

DILATION

With A and B as sets in Z^2, the dilation of A by B, denoted as $A \oplus B$, is defined as

$$A \oplus B = \left\{ z \mid (\hat{B})_z \cap A \neq \varnothing \right\} \tag{9-6}$$



a b
c d

FIGURE 9.5 Using erosion to remove image components. (a) A 486×486 binary image of a wire-bond mask in which foreground pixels are shown in white. (b)–(d) Image eroded using square structuring elements of sizes 11×11, 15×15, and 45×45 elements, respectively, all valued 1.

to smooth sections of contours, but, as opposed to opening, it generally fuses narrow breaks and long thin gulfs, eliminates small holes, and fills gaps in the contour.

The *opening* of set A by structuring element B, denoted by $A \circ B$, is defined as

$$A \circ B = (A \ominus B) \oplus B \tag{9-10}$$

Thus, the opening A by B is the erosion of A by B, followed by a dilation of the result by B.

Similarly, the *closing* of set A by structuring element B, denoted $A \bullet B$, is defined as

$$A \bullet B = (A \oplus B) \ominus B \tag{9-11}$$

which says that the closing of A by B is simply the dilation of A by B, followed by erosion of the result by B.

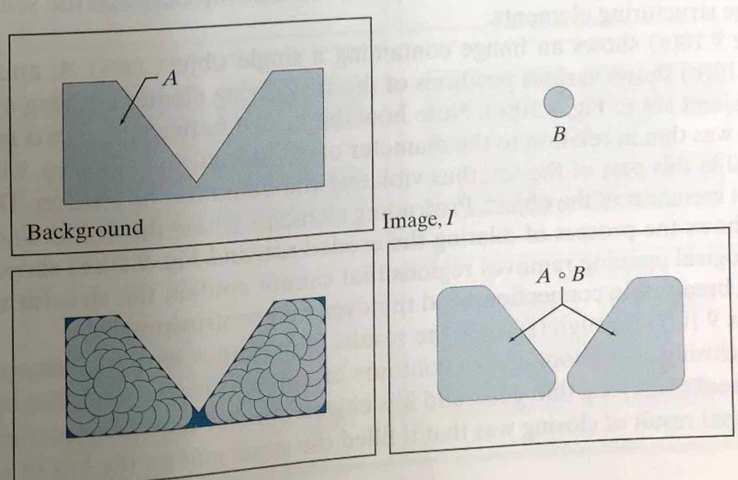
When a circular structuring element is used for opening, the analogy is often made of the shape of the opening being determined by a "rolling ball" reaching as far as it can on the inner boundary of a set. For morphological closing the ball rolls outside, and the shape of the closing is determined by how far the ball can reach into the boundary.

Equation (9-10) has a simple geometrical interpretation: The opening of A by B is the union of all the translations of B so that B fits entirely in A. Figure 9.8(a) shows an image containing a set (object) A and Fig. 9.8(b) is a solid, circular structuring element, B. Figure 9.8(c) shows some of the translations of B such that it is contained within A, and the set shown shaded in Fig. 9.8(d) is the union of all such possible translations. Observe that, in this case, the opening is a set composed of two disjoint subsets, resulting from the fact that B could not fit in the narrow segment in the center of A. As you will see shortly, the ability to eliminate regions narrower than the structuring element is one of the key features of morphological opening.

The interpretation that the opening of A by B is the union of all the translations of B such that B fits entirely within A can be written in equation form as

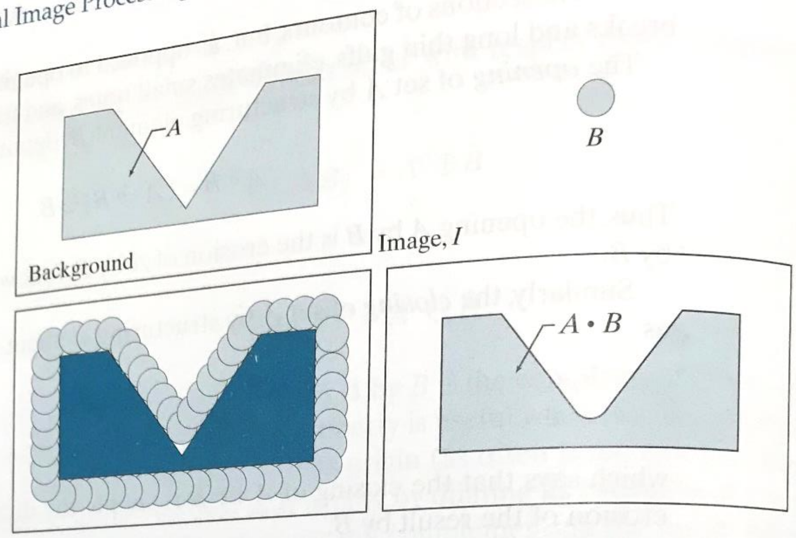
$$A \circ B = \bigcup \left\{ (B)_z \,\middle|\, (B)_z \subseteq A \right\} \tag{9-12}$$

where $\bigcup$ denotes the union of the sets inside the braces.



a b
c d

FIGURE 9.8 (a) Image I, composed of set (object) A and background. (b) Structuring element, B. (c) Translations of B while being contained in A. (A is shown dark for clarity.) (d) Opening of A by B.



a b
c d

FIGURE 9.9 (a) Image I, composed of set (object) A, and background. (b) Structuring element B. (c) Translations of B such that B does not overlap any part of A. (A is shown dark for clarity.) (d) Closing of A by B.

Closing has a similar geometric interpretation, except that now we translate B *outside* A. The closing is then the *complement* of the union of all translations of B that *do not* overlap A. Figure 9.9 illustrates this concept. Note that the boundary of the closing is determined by the furthest points B could reach without going inside any part of A. Based on this interpretation, we can write the closing of A by B as

$$A \bullet B = \left[\bigcup \left\{ (B)_z \,\middle|\, (B)_z \cap A = \varnothing \right\} \right]^c \qquad (9\text{-}13)$$

EXAMPLE 9.3: Morphological opening and closing.

Figure 9.10 shows in more detail the process and properties of opening and closing. Unlike Figs. 9.8 and 9.9, whose main objectives are overall geometrical interpretations, this figure shows the individual processes and also pays more attention to the relationship between the scale of the final results and the size of the structuring elements.

Figure 9.10(a) shows an image containing a single object (set) A, and a disk structuring element. Figure 9.10(b) shows various positions of the structuring element during erosion. This process resulted in the disjoint set in Fig. 9.10(c). Note how the bridge between the two main sections was eliminated. Its width was thin in relation to the diameter of the structuring element, which could not be completely contained in this part of the set, thus violating the definition of erosion. The same was true of the two rightmost members of the object. Protruding elements where the disk did not fit were eliminated. Figure 9.10(d) shows the process of dilating the eroded set, and Fig. 9.10(e) shows the final result of opening. Morphological opening removes regions that cannot contain the structuring element, smoothes object contours, breaks thin connections, and removes thin protrusions.

Figures 9.10(f) through (i) show the results of closing A with the same structuring element. As with opening, closing also smoothes the contours of objects. However, unlike opening, closing tends to join narrow breaks, fills long thin gulfs, and fills objects smaller than the structuring element. In this example, the principal result of closing was that it filled the small gulf on the left of set A.