Machine Vision
Jain, Kasturi, Schunck 1995

a line end are erased. In Figure 2.22 you can see how, on each iteration, the outside layer of a 1-valued region is peeled off in this manner. When no changes are made on an iteration, the process is complete and the image is thinned.

2.5.12 Expanding and Shrinking

A component of an image can be systematically expanded or contracted. When a component is allowed to change such that some background pixels are converted to 1, the operation is called expanding. If object pixels are systematically deleted or converted to 0, then it is called shrinking. A simple implementation of expanding and shrinking may be:

Expanding: Change a pixel from 0 to 1 if any neighbors of the pixel are 1.

Shrinking: Change a pixel from 1 to 0 if any neighbors of the pixel are 0.

Thus, shrinking may be considered as expanding the background. Example of these operations are given in Figure 2.23.

It is interesting that simple operations like expanding and shrinking can be used to do some very useful and seemingly complex operations on images. Let us denote:

$S^{(k)}$: S expanded k times
$S^{(-k)}$: S shrunk k times

It can be shown that the following properties hold:

$$\begin{aligned}(S^m)^{-n} &\neq (S^{-n})^m \\ &\neq S^{(m-n)}\end{aligned}$$

$$\begin{aligned}S &\subset (S^k)^{-k} \\ S &\supset (S^{-k})^k\end{aligned}$$

Expanding followed by shrinking can be used for filling undesirable holes, and shrinking followed by expanding can be used for removing isolated noise pixels (see Figure 2.24). Expanding and shrinking can be used to determine isolated components and clusters. In morphological image processing and dilation and erosion operations, generalized forms of expanding and shrinking are used extensively to do many tasks.

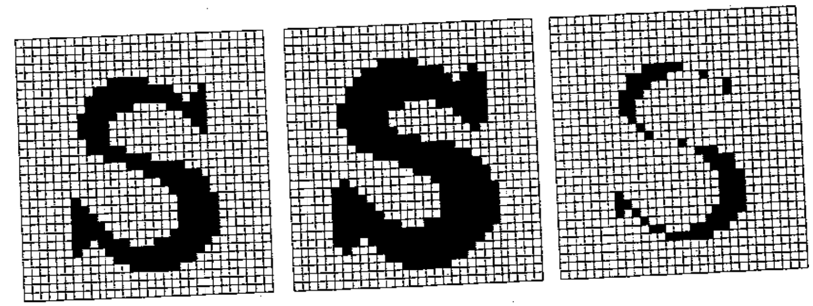


Figure 2.23: An example of expanding and shrinking operations on the letter "s." *Left:* The original image. *Middle:* Expanded image. *Right:* Shrunken image.

2.6 Morphological Operators

Mathematical morphology gets its name from the study of shape. This approach exploits the fact that in many machine vision applications, it is natural and easy to think in terms of shapes when designing algorithms. A morphological approach facilitates shape-based, or iconic, thinking. The fundamental unit of pictorial information in the morphological approach is the binary image.

The *intersection* of any two binary images A and B, written $A \cap B$, is the binary image which is 1 at all pixels p which are 1 in both A and B. Thus,

$$A \cap B = \{p | p \in A \text{ and } p \in B\}. \tag{2.44}$$

The *union* of A and B, written $A \cup B$, is the binary image which is 1 at all pixels p which are 1 in A or 1 in B (or 1 in both). Symbolically,

$$A \cup B = \{p | p \in A \text{ or } p \in B\}. \tag{2.45}$$

Let Ω be a universal binary image (all 1) and A a binary image. The *complement* of A is the binary image which interchanges the 1s and 0s in A. Thus,

$$\overline{A} = \{p | p \in \Omega \text{ and } p \notin A\}. \tag{2.46}$$

The *vector sum* of two pixels p and q with indices $[i,j]$ and $[k,l]$ is the pixel $p+q$ with indices $[i+k, j+l]$. The *vector difference* $p-q$ is the pixel with indices $[i-k, j-l]$.

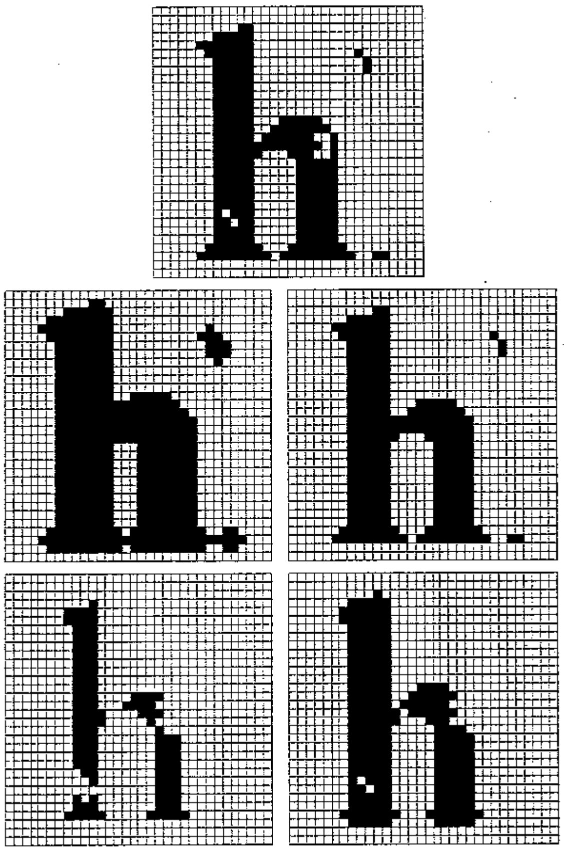


Figure 2.24: Sequences of expanding and shrinking the letter "h." *Top:* The original noisy image. *Middle:* Expanding followed by shrinking. *Bottom:* Shrinking followed by expanding. Note that expanding followed by shrinking effectively filled the holes but did not eliminate the noise. Conversely, shrinking followed by expanding eliminated the noise but did not fill the holes.

If A is a binary image and p is a pixel, then the *translation* of A by p is an image given by

$$A_p = \{a + p | a \in A\}. \tag{2.47}$$

Dilation

Translation of a binary image A by a pixel p shifts the origin of A to p. If $A_{b_1}, A_{b_2}, \ldots, A_{b_n}$ are translations of the binary image A by the 1 pixels of the binary image $B = \{b_1, b_2, \ldots, b_n\}$, then the union of the translations of A by the 1 pixels of B is called the *dilation* of A by B and is given by

$$A \oplus B = \bigcup_{b_i \in B} A_{b_i} = \bigcup_{a_i \in A} B_{a_i} \tag{2.48}$$

Dilation has both associative and commutative properties. Thus, in a sequence of dilation steps the order of performing operations is not important. This fact allows breaking a complex shape into several simpler shapes which can be recombined as a sequence of dilations.

Erosion

The opposite of dilation is *erosion*. The erosion of a binary image A by a binary image B is 1 at a pixel p if and only if every 1 pixel in the translation of B to p is also 1 in A. Erosion is given by

$$A \ominus B = \{p | B_p \subseteq A\}. \tag{2.49}$$

Often the binary image B is a regular shape which is used as a probe on image A and is referred to as a *structuring element*. Erosion plays a very important role in many applications. Erosion of an image by a structuring element results in an image that gives all locations where the structuring element is contained in the image.

Figures 2.25 through 2.28 illustrate the dilation and erosion operations with a simple binary object and an upside-down "T-shaped" structuring element. Figure 2.26 shows examples of translations of the structuring element to 1 pixels of the original figure where the entire structuring element does *not* fit entirely inside the original object. In this case, during a dilation, every pixel in the structuring element will be present in the final dilated image, including the pixel not contained in the original object (shown as a lightly

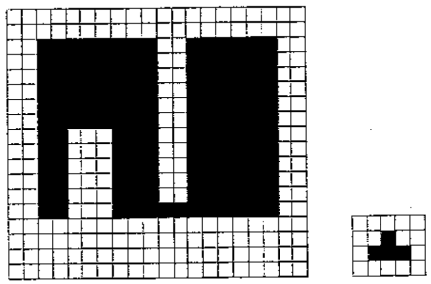


Figure 2.25: The original test image A (left) and structuring element B (right). Note that the origin of the structuring element is darker than the other pixels in B.

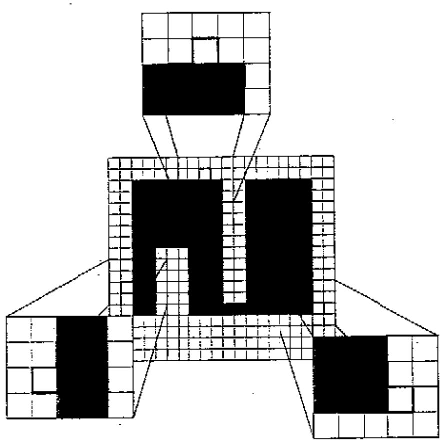
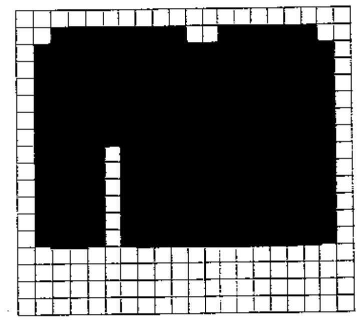
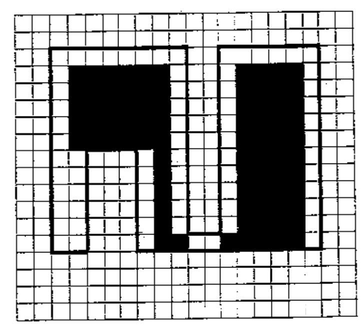


Figure 2.26: Translations of the structuring element B to 1 pixels in A where the entire structuring element is *not* contained within A. During a dilation operation, every pixel in the structuring element will be present in the final image. During an erosion operation, the pixel at the origin of the structuring element will be deleted.



$$A \oplus B = \bigcup_{b_i \in B} A_{b_i}$$

Figure 2.27: The dilation of A by B. The boundary of the original figure A is shown as a bold line.



$$A \ominus B = \{p | B_p \subseteq A\}$$

Figure 2.28: The erosion of A by B. The boundary of the original figure A is shown as a bold line.

shaded pixel). But during an erosion operation the pixel at the origin of the structuring element will be removed because the entire structuring element is not within the object. Conversely, in the case where the entire structuring element *does* fit within the original object, there will be no change to the final dilated or eroded image (i.e., no pixels will be added or deleted at that point).

Dilation and erosion exhibit a dual nature that is geometric rather than logical and involves a geometric complement as well as a logical complement. The geometric complement of a binary image is called its *reflection*. The reflection of a binary image B is that binary image B' which is symmetric with B about the origin, that is

$$B' = \{-p \mid p \in B\}. \tag{2.50}$$

The geometric duality of dilation and erosion is expressed by the relationships

$$\overline{A \oplus B} = \overline{A} \ominus B' \tag{2.51}$$

and

$$\overline{A \ominus B} = \overline{A} \oplus B'. \tag{2.52}$$

Geometric duality contrasts with logical duality:

$$\overline{A \cup B} = \overline{A} \cap \overline{B} \tag{2.53}$$

and

$$\overline{A \cap B} = \overline{A} \cup \overline{B}, \tag{2.54}$$

also called deMorgan's law. The duality of dilation and erosion are illustrated in Figures 2.29 through 2.31.

Erosion and dilation are often used in filtering images. If the nature of noise is known, then a suitable structuring element can be used and a sequence of erosion and dilation operations can be applied for removing the noise. Such filters affect the shape of the objects in the image.

The basic operations of mathematical morphology can be combined into complex sequences. For example, an erosion followed by a dilation with the same structuring element (probe) will remove all of the pixels in regions which are too small to contain the probe, and it will leave the rest. This sequence

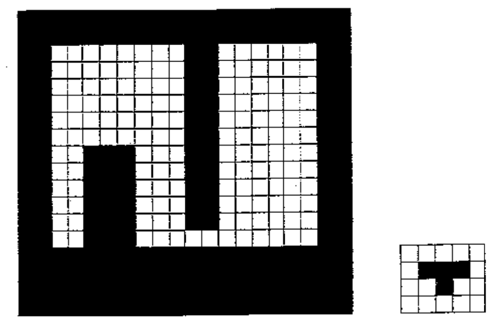
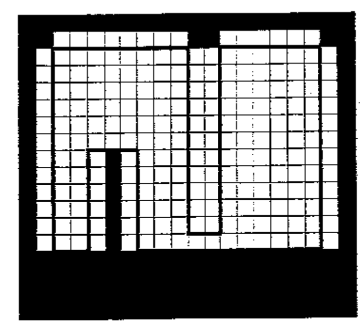


Figure 2.29: The complement of A and the reflection of B. Note that the origin of the reflected structuring element is still the darker pixel.



$\overline{A} \ominus B'$

Figure 2.30: The dual of dilation: the result of eroding the background of A with the reflection of B. The original boundary is shown as a bold line.

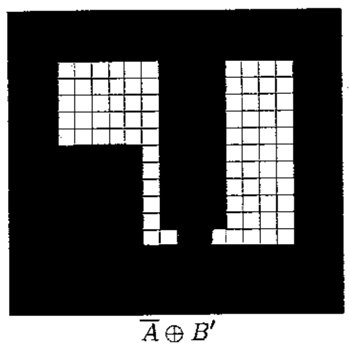


Figure 2.31: The dual of erosion: the result of dilating the background of A with the reflection of B. The original boundary is shown as a bold line.

is called *opening*. As an example, if a disc-shaped probe image is used, then all of the convex or isolated regions of pixels smaller than the disc will be eliminated. This forms a filter which suppresses positive spatial details. The remaining pixels show where the structuring element is contained in the foreground. The difference of this result and the original image would show those regions which were too small for the probe, and these could be the features of interest, depending on the application.

The opposite sequence, a dilation followed by an erosion, will fill in holes and concavities smaller than the probe. This is referred to as *closing.* These operations are illustrated in Figures 2.32 and 2.33 with the same T-shaped structuring element. Again, what is removed may be just as important as what remains. Such filters can be used to suppress spatial features or discriminate against objects based upon their size. The structuring element used does not have to be compact or regular, and can be any pattern of pixels. In this way, features made up of distributed pixels can be detected.

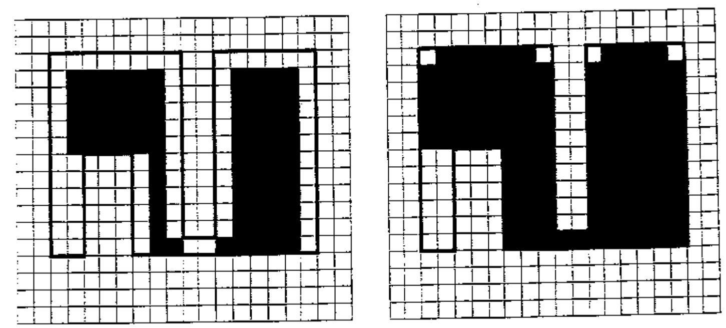


Figure 2.32: Opening operation. *Left:* Initial erosion. *Right:* Succeeding dilation. The boundary of the original figure A is shown as a bold line.

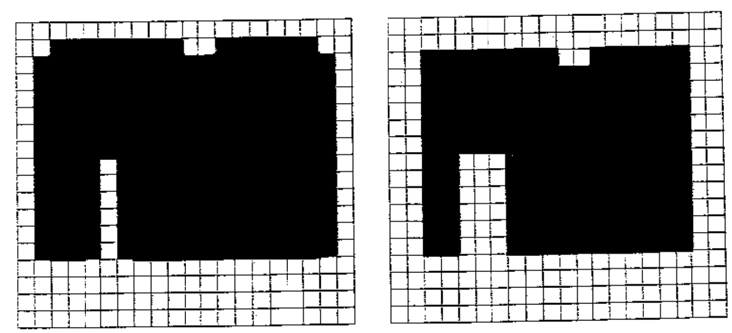


Figure 2.33: Closing operation. *Left:* Initial dilation. *Right:* Succeeding erosion. The boundary of the original figure A is shown as a bold line.

2.7 Optical Character Recognition

Morphological operations can be used for optical character recognition when the characters are relatively noise-free and are of the same font and size. First, extract the character to be recognized from one of the images in which the character appears. Next, use expanding or closing to fill holes and cavities. Then shrink the character image to remove unwanted regions and to reduce the size of the character so that it will fit comfortably inside an instance of the character. This processed image is the model for the character.

To recognize instances of the character in other images, use that character model as a probe and perform erosion. The images may have to be cleaned (holes filled and unwanted clutter removed) before performing erosion. After erosion, compute connected components, apply the size filter to discard regions that are too small, and compute the position of each region that passes through the size filter. This provides the position of each recognized instance of the character model in the image. Good character models obtained after cleaning, filling, and shrinking will match most instances of the character, including instances in a slightly different font and size. However, omnifont recognition has been a very challenging problem for researchers. Optical character recognition can be implemented in real time with special-purpose hardware.

Further Reading

Several aspects of binary vision are given in the books by Rosenfeld and Kak [206] and Horn [109]. Morphological image processing is discussed in the books [68, 103]. An example of the use of morphological operations in a practical application is provided in the paper by Mitchell and Gillies on reading hand-printed zip codes [168]. For a tutorial on optical character recognition and document analysis see O'Gorman and Kasturi [188].

Exercises

2.1 We saw in this chapter that the zeroth-, first-, and second-order moments for a binary region provide important information about the image's size, location, and