The output value y is then given by

$$y = \sum_{n=1}^{8} a_n$$

6.2.6 Connected Shrink Operator

The connected shrink operator is a recursive operator having a symbolic data domain. It is similar in certain respects to the connectivity number operator and the region-shrinking operator. Instead of labeling all border pixels with the background symbol g, the connected shrink operator labels only those border pixels that can be deleted from a connected region without disconnecting the region. Since it is applied recursively, pixels that are interior during one position of the scan may appear as border pixels at another position of the scan and eventually may be deleted by this operator. After one complete scan of the image, the set of pixels that get labeled as background is a strong function of the way in which the image is scanned with the operator. For example, as illustrated in Fig. 6.13, a top-down, left-right scan will delete all edge pixels that are not right-boundary edge pixels.

The theoretical basis of the connected shrink operator was explored by Rosenfeld and Pfaltz (1966), Rosenfeld (1970), and Stefanelli and Rosenfeld (1971). Basically a pixel's label is changed to g, (for background) if upon deleting it from the region it belongs to, the region remains connected. The operator definition given here is based on Yokoi, Toriwaki, and Fukumura (1975). The operator uses an 8-connected neighborhood and can be defined for deleting either 4-deletable or 8-deletable pixels. It requires two primitive functions: a function h that can determine whether the three-pixel corner of a neighborhood is connected and a function g that basically counts the number of arguments having certain values.

In the 4-connectivity mode, the four-argument primitive function h is defined by using

$$h(b,c,d,e) = \begin{cases} 1 & \text{if } b = c \text{ and } (d \neq b \text{ or } e \neq b) \\ 0 & \text{otherwise} \end{cases}$$

In the 8-connectivity mode, the four-argument primitive function h is defined by

$$h(b,c,d,e) = \begin{cases} 1 & \text{if } c \neq b \text{ and } (d = b \text{ or } e = b) \\ 0 & \text{otherwise.} \end{cases}$$

```
  **
 ****
******                 ***
 ** **                 * *
  * **                 * *
  * **                 * *
  * **                   *

   (a)                 (b)
input image     connected shrink
                  output image
```

Figure 6.13 Connected shrink operator applied in a top-down, left-right scan using 4-connectivity.

The five-argument primitive function f is defined by

$$f(a_1,a_2,a_3,a_4,x) = \begin{cases} g & \text{if exactly one of } a_1,a_2,a_3,a_4 = 1 \\ x & \text{otherwise} \end{cases}$$

Using the indexing convention of Fig. 6.11, we define the connected shrink operator by letting

$$a_1 = h(x_0,x_1,x_6,x_2)$$

$$a_2 = h(x_0,x_2,x_7,x_3)$$

$$a_3 = h(x_0,x_3,x_8,x_4)$$

$$a_4 = h(x_0,x_4,x_5,x_1)$$

The output symbol y is defined by $y = f(a_1,a_2,a_3,a_4,x_0)$. Figure 6.14 further illustrates the connected shrink operator.

The earliest discussion of connectivity in digital pictures can be found in Rosenfeld (1971). Rutovitz (1966) preceded Rosenfeld in the use of crossing numbers but did not use connectivity in his development. Related algorithms and discussion of connectivity can be found in Levialdi (1972), who introduced a parallel or nonrecursive shrinking algorithm for the purpose of counting the number of components in a binary image. This iterative algorithm does not employ the 1-deletability of the Yokoi, Toriwaki, and Fukumura method; it uses a 2×2 window rather than a 3×3 window in the shrinking process but requires the detection of an isolated element

Original Image

g	g	g	g	g	g	g	g	
g	1	1	g	g	g	1	1	g
g	1	1	1	g	g	1	1	g
g	1	1	1	1	1	1	1	g
g	1	1	1	g	g	1	1	g
g	1	1	g	g	g	1	1	g
g	g	g	g	g	g	g	g	

g	g	g
g	1	1
g	1	1

$a_1 = 1$ $\quad a_3 = 0$

$a_2 = 0$ $\quad a_4 = 0$

$f(1,0,0,0,1) = g$

Result

g	g	g	g	g	g	g	g
g	g	g	1	g	g	g	g
g	g	1	1	1	1	1	g
g	g	g	1	g	g	g	g
g	g	g	g	g	g	g	g

Figure 6.14 Application of the connected shrink operator to an image using 4-connectivity.

during the iterative process so that it may be counted before it is made to disappear by the process. A three-dimensional extension to this nonrecursive algorithm can be found in Arcelli and Levialdi (1972). Lobregt, Verbeck, and Groen (1980) discuss a recursive operator for three-dimensional shrinking.

6.2.7 Pair Relationship Operator

The pair relationship operator is nonrecursive and has a symbolic data domain. It is a general operator that labels a pixel on the basis of whether it stands in the specified relationship with a neighborhood pixel. An example of a pair relationship operator is one that relabels with a specified label all border pixels that are next to an interior pixel and either can relabel all other pixels with another specified label or can leave their labels alone. Formally, a pair relationship operator marks a pixel with the specified label p if the pixel has a specified label l and neighbors enough pixels having a specified label m. All other pixels it either marks with another specified label or leaves unmodified.

The pair relationship operator employs two primitive functions. The two-argument function h can recognize whether its first argument has the value of its second argument. It is defined by

label=m

m = 1 mean edge in Yokoi (p.273)

primitive function ①

$$h(a,m) = \begin{cases} 1 & \text{if } a = m \\ 0 & \text{otherwise} \end{cases}$$

如果 a這個 pixel 是 m這個類型的 pixel h 就 output a 為 1

primitive function ②

For the $\boxed{\text{4-connected mode,}}$ the output value y is defined by

Note: HW7 做改 m, θ=1

x_0 的 neighboring pixels 中不到 θ個為 label m

$$y = \begin{cases} q & \text{if } \sum_{n=1}^{4} h(x_n, m) < \overset{\theta=1}{\theta} \text{ or } x_0 \neq 1 \\ p & \text{if } \sum_{n=1}^{4} h(x_n, m) \geq \theta \text{ and } x_0 = 1 \end{cases}$$

如果 x_0 是 background

如果 x_0 是 foreground pixel ($x_0=1$) 且 x_0 的 4-neighbors 其中有 ≥θ個 都是 m 這個類型的 → y 就把 x_0 這樣的 pixel 標記為一個特定的 label p

where q can be either a specified output label or the label x_8.

For the $\boxed{\text{8-connected mode,}}$ the output y is defined by

x_7	x_2	x_6
x_3	x_0	x_1
x_8	x_4	x_5

$$y = \begin{cases} q & \text{if } \sum_{n=1}^{8} h(x_n, m) < \theta \text{ or } x_0 \neq 1 \\ p & \text{if } \sum_{n=1}^{8} h(x_n, m) \geq \theta \text{ and } x_0 = 1 \end{cases}$$

where q can be either a specified output or the label x_8.

(label=1)

△ HW7: 如果 x_0 是 edge, 且 x_0 的 4-neighbors 中有 ≥1 個 pixel 也是 edge → x_0 標成 p (否則標成 q)

6.2.8 Thinning Operator

The thinning operator discussed here is defined as a composition of three operators: the ① mark-interior/border-pixel operator, the ② pair relationship operator, and the ③ marked-pixel connected shrink operator. It works by marking all border pixels that are next to interior pixels and then deleting (or shrinking) any marked pixel that is

Thinning Operator

$\xrightarrow{\text{successively apply}}$ no interior pixel
remain center lines

→ composition of 3 operators

1. **mark interior / border pixel**

   → examine original symbolic image → produce : interior / border image

2. **pair relationship operator**

   → examine interior / border image → produce : pixel marked if :
   → original image → 1. is border pixel
   & 2. is next to interior pixels

3. **marked pixel connected shrink operator**

   input : original symbolic image
   marked pixel image

   → like connected shrink operator
   but shrink pixels : deletable
   &
   marked

center lines

```
******
******
*******
*  ****
*   ***
*     *
```

(a)
input image

```
 ****
**  *
*   **
*    **
*     *
```

(b)
thinned output image

Figure 6.15 Result of one application of the thinning operator using the 4-connectivity deletability condition.

deletable. The result of successively applying the thinning operator on a symbolic image is that all regions are symmetrically shrunk down until no interior pixels are left. What remains is their center lines, as shown in Fig. 6.15. This operator has the nice property that the center line is connected in exactly the same geometric and topologic way the original figure is connected. For other similar operators that thin without changing geometry or topology, see Rosenfeld and Davis (1976), Stefanelli and Rosenfeld (1971), or Arcelli and Sanniti di Baja (1978). To implement the operator as the composition of three operators, the mark-interior/border-pixel operator examines the original symbolic image to produce an interior/border image. The interior/border image is examined by the pair relationship operator, which produces an image whose pixels are marked if on the original image they were border pixels and were next to interior pixels. The marked-pixel image and the original symbolic image constitute the input to the marked-pixel connected shrink operator, which is exactly like the connected shrink operator except it shrinks only pixels that are deletable and marked.

The first discussions of thinning appeared in Hilditch (1969) and Deutsch (1969). These initial insights were later expanded by Fraser (1970), Stefenelli and Rosenfeld (1971), Deutsch (1972), Rosenfeld (1975), and Rosenfeld and Davis (1976). A brief comparison of thinning techniques can be found in Tamura (1978), who suggests that a smooth 8-connected thinning results if 8-deletable pixels are removed from thinning 4-connected curves. Tamura also notes that the thinning of Rosenfeld and Davis (1976) is very sensitive to contour noise when used in the 4-connected mode.

6.2.9 Distance Transformation Operator

The distance transformation operator can be implemented as either a recursive or a nonrecursive operator. It requires a binary image whose border pixels are labeled 0 and whose interior pixels are labeled i. The purpose of the distance transformation operator is to produce a numeric image whose pixels are labeled with the distance between each of them and their closest border pixel. The distance between two pixels can be defined by the length of the shortest 4-connected path (city-block distance) or 8-connected path (max or chessboard distance) between them.