

Algorithm 11.3

Combined contour tracing and region labeling. Given a binary image I , the method COMBINED-CONTOURLABELING() returns a set of contours and an array containing region labels for all pixels in the image. When a new point on either an outer or inner contour is found, then an ordered list of the contour's points is constructed by calling the method TRACECONTOUR() (line 20 and line 27). TRACECONTOUR() itself is described in Alg. 11.4.

```

1: COMBINEDCONTOURLABELING ( $I$ )
    $I$ : binary image
   Returns a set of contours and a label map (labeled image).
2: Create an empty set of contours:  $\mathcal{C} \leftarrow \{\}$ 
3: Create a label map  $LM$  of the same size as  $I$  and initialize:
4: for all  $(u, v)$  do
5:      $LM(u, v) \leftarrow 0$  ▷ label map  $LM$ 
6:  $R \leftarrow 0$  ▷ region counter  $R$ 
7: Scan the image from left to right and top to bottom:
8: for  $v \leftarrow 0 \dots N-1$  do
9:      $L_k \leftarrow 0$  ▷ current label  $L_k$ 
10:    for  $u \leftarrow 0 \dots M-1$  do
11:        if  $I(u, v)$  is a foreground pixel then
12:            if  $(L_k \neq 0)$  then ▷ continue existing region
13:                 $LM(u, v) \leftarrow L_k$ 
14:            else
15:                 $L_k \leftarrow LM(u, v)$ 
16:                if  $(L_k = 0)$  then ▷ hit new outer contour
17:                     $R \leftarrow R + 1$ 
18:                     $L_k \leftarrow R$ 
19:                     $\mathbf{x}_S \leftarrow (u, v)$ 
20:                     $\mathbf{c}_{\text{outer}} \leftarrow \text{TRACECONTOUR}(\mathbf{x}_S, 0, L_k, I, LM)$ 
21:                     $\mathcal{C} \leftarrow \mathcal{C} \cup \{\mathbf{c}_{\text{outer}}\}$  ▷ collect new contour
22:                     $LM(u, v) \leftarrow L_k$ 
23:                else ▷  $I(u, v)$  is a background pixel
24:                    if  $(L_k \neq 0)$  then
25:                        if  $(LM(u, v) = 0)$  then ▷ hit new inner contour
26:                             $\mathbf{x}_S \leftarrow (u-1, v)$ 
27:                             $\mathbf{c}_{\text{inner}} \leftarrow \text{TRACECONTOUR}(\mathbf{x}_S, 1, L_k, I, LM)$ 
28:                             $\mathcal{C} \leftarrow \mathcal{C} \cup \{\mathbf{c}_{\text{inner}}\}$  ▷ collect new contour
29:                             $L_k \leftarrow 0$ 
30: return  $(\mathcal{C}, LM)$ . ▷ return the set of contours and the label map

```

continued in Alg. 11.4 ▷▷

(defined on p. 203). The Java container class `ArrayList` (templated on the type `Node`) is used as a dynamic data structure for storing the point sequences of the outer and inner contours.

- The method `traceContour()` (see p. 538) traverses an outer or inner contour, beginning from the starting point \mathbf{x}_S (\mathbf{x}_S , \mathbf{y}_S). It calls the method `findNextPoint()`, to determine the next contour point \mathbf{x}_T (\mathbf{x}_T , \mathbf{y}_T) following \mathbf{x}_S :
 - In the case that no following point is found, then $\mathbf{x}_S = \mathbf{x}_T$ and the region (contour) consists of a single isolated pixel. The method `traceContour()` is finished.
 - In the other case the remaining contour points are found by repeatedly calling `findNextPoint()`, and for every successive pair of points the *current* point \mathbf{x}_c (\mathbf{x}_C , \mathbf{y}_C) and the *previous* point \mathbf{x}_p