

<pre> 1: TRACECONTOUR(<math>\mathbf{x}_S, d_S, L_c, I, LM</math>)    <math>\mathbf{x}_S</math>: start position, <math>d_S</math>: initial search direction,    <math>L_c</math>: label for this contour    <math>I</math>: original image, <math>LM</math>: label map.    Traces and returns the contour starting at <math>\mathbf{x}_S</math>. 2: (<math>\mathbf{x}_T, d_{next}</math>) <math>\leftarrow</math> FINDNEXTPOINT(<math>\mathbf{x}_S, d_S, I, LM</math>) 3: <math>\mathbf{c} \leftarrow [\mathbf{x}_T]</math> <math>\triangleright</math> create a contour starting with <math>\mathbf{x}_T</math> 4: <math>\mathbf{x}_p \leftarrow \mathbf{x}_S</math> <math>\triangleright</math> previous position <math>\mathbf{x}_p = (u_p, v_p)</math> 5: <math>\mathbf{x}_c \leftarrow \mathbf{x}_T</math> <math>\triangleright</math> current position <math>\mathbf{x}_c = (u_c, v_c)</math> 6: <math>done \leftarrow (\mathbf{x}_S \equiv \mathbf{x}_T)</math> <math>\triangleright</math> isolated pixel? 7: <b>while</b> (<math>\neg done</math>) <b>do</b> 8:   <math>LM(u_c, v_c) \leftarrow L_c</math> 9:   <math>d_{search} \leftarrow (d_{next} + 6) \bmod 8</math> 10:  (<math>\mathbf{x}_n, d_{next}</math>) <math>\leftarrow</math> FINDNEXTPOINT(<math>\mathbf{x}_c, d_{search}, I, LM</math>) 11:  <math>\mathbf{x}_p \leftarrow \mathbf{x}_c</math> 12:  <math>\mathbf{x}_c \leftarrow \mathbf{x}_n</math> 13:  <math>done \leftarrow (\mathbf{x}_p \equiv \mathbf{x}_S \wedge \mathbf{x}_c \equiv \mathbf{x}_T)</math> <math>\triangleright</math> back at start point? 14:  <b>if</b> (<math>\neg done</math>) <b>then</b> 15:    APPEND(<math>\mathbf{c}, \mathbf{x}_n</math>) <math>\triangleright</math> add point <math>\mathbf{x}_n</math> to contour <math>\mathbf{c}</math> 16:  <b>return</b> <math>\mathbf{c}</math>. <math>\triangleright</math> return this contour </pre>																												
<pre> 17: FINDNEXTPOINT(<math>\mathbf{x}_c, d, I, LM</math>)    <math>\mathbf{x}_c</math>: start point, <math>d</math>: search direction,    <math>I</math>: original image, <math>LM</math>: label map. 18: <b>for</b> <math>i \leftarrow 0 \dots 6</math> <b>do</b> <math>\triangleright</math> search in 7 directions 19:  <math>\mathbf{x}' \leftarrow \mathbf{x}_c + \text{DELTA}(d)</math> <math>\triangleright \mathbf{x}' = (u', v')</math> 20:  <b>if</b> <math>I(u', v')</math> is a background pixel <b>then</b> 21:    <math>LM(u', v') \leftarrow -1</math> <math>\triangleright</math> mark background as visited (-1) 22:    <math>d \leftarrow (d + 1) \bmod 8</math> 23:  <b>else</b> <math>\triangleright</math> found a nonbackground pixel at <math>\mathbf{x}'</math> 24:    <b>return</b> (<math>\mathbf{x}', d</math>) 25:  <b>return</b> (<math>\mathbf{x}_c, d</math>). <math>\triangleright</math> found no next point, return start point </pre>																												
<pre> 26: DELTA(<math>d</math>) = (<math>\Delta x, \Delta y</math>), with </pre> <table border="1" style="margin-left: 100px;"> <tr> <td><math>d</math></td> <td>0</td> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td>5</td> <td>6</td> <td>7</td> </tr> <tr> <td><math>\Delta x</math></td> <td>1</td> <td>1</td> <td>0</td> <td>-1</td> <td>-1</td> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td><math>\Delta y</math></td> <td>0</td> <td>1</td> <td>1</td> <td>1</td> <td>0</td> <td>-1</td> <td>-1</td> <td>-1</td> </tr> </table>	$d$	0	1	2	3	4	5	6	7	$\Delta x$	1	1	0	-1	-1	0	1	1	$\Delta y$	0	1	1	1	0	-1	-1	-1	
$d$	0	1	2	3	4	5	6	7																				
$\Delta x$	1	1	0	-1	-1	0	1	1																				
$\Delta y$	0	1	1	1	0	-1	-1	-1																				

( $\mathbf{x}_p, \mathbf{y}_p$ ) are recorded. Only when *both* points correspond to the original starting points on the contour,  $\mathbf{x}_p = \mathbf{x}_S$  and  $\mathbf{x}_c = \mathbf{x}_T$ , we know that the contour has been completely traversed.

- The method `findNextPoint()` (see p. 539) determines which point on the contour follows the current point  $\mathbf{x}_c$  (pt) by searching in the *direction*  $d$  (dir), depending upon the position of the previous contour point. Starting in the first search direction, up to seven neighboring pixels (all neighbors except the previous contour point) are searched in clockwise direction until the next contour point is found. At the same time, all background pixels in the *label map*  $LM$  (labelArray) are marked with the value  $-1$  to prevent them from being searched again. If no valid contour point is found among the

## 11.2 REGION CONTOURS

### Algorithm 11.4

Combined contour finding and region labeling (continued from Alg. 11.3). Starting from  $\mathbf{x}_S$ , the procedure TRACECONTOUR traces along the contour in the direction  $d_S = 0$  for outer contours or  $d_S = 1$  for inner contours. During this process, all contour points as well as neighboring background points are marked in the label array  $LM$ . Given a point  $\mathbf{x}_c$ , TRACECONTOUR uses FINDNEXTPOINT() to determine the next point along the contour (line 10). The function DELTA() returns the next coordinate in the sequence, taking into account the search direction  $d$ .