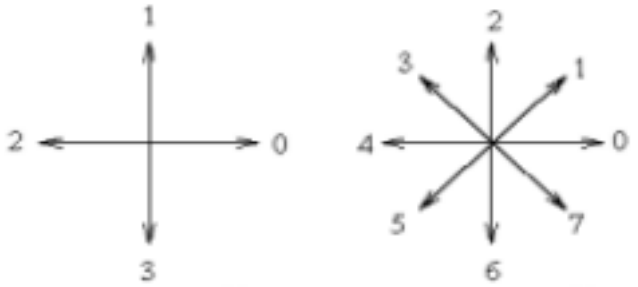
BORDER TRACING

- It is used to extract the borders of the objects from an image.
- The first step is to select an initial border point from the object of a binary image, followed by a search of its 4-neighbouring or 8-neighbouring pixels, and finally

output the next border point.

- Variable is used to record the search direction. 2

- For 4 neighborhood, integers from 0 to 3 are used to record the different directions of the neighborhood of the pixel.
- For 8 neighborhood, integers from 0 to 7 are used to record the different directions of the neighborhood of the pixel.


1
2
0
3
2
3
1
4
0
5
7
6

Algorithm 4.4: Border tracing detection in 8-neighbourhood

For the given binary image $f(i, j)$: $0 \le i \le m-1; 0 \le j \le n-1$;

$k = 0$ denotes the number of the borders;

$q_k = \phi$; //the set of the pixel points on the border of the object k;

$si_k = 0$; $sj_k = 0$; // the search beginning point of the k-th region;

do while (true)

{ For $i = si_k$ to $m - 1$ // find a starting border pixel of a new region;

For $j = sj_k$ to $n - 1$

if ($f(i, j) = 1$) and ($(i, j) \notin q_s, s = 0, \ldots, k - 1$) then

{$p(k, 0) = (i, j)$; // the first border pixel of the k-th region;

$si_{k+1} = i$; $sj_{k+1} = j + 1$ // the search of next region will begin from this pixel;

exit;

}

else halt;

End-For

End-For;

Initialize the search direction variable dir = 7;

$s = 0$; // the current border pixel;

Repeat // search border pixels;

{ If (dir is odd) then dir = (dir + 6) / mod 8

else dir = (dir + 7) / mod 8; // the beginning direction of search;

while (dir < 8) and (the corresponding neighbouring pixel is not a border pixel)

do {dir = (dir + 1) / mod 8;}

$s = s + 1$;

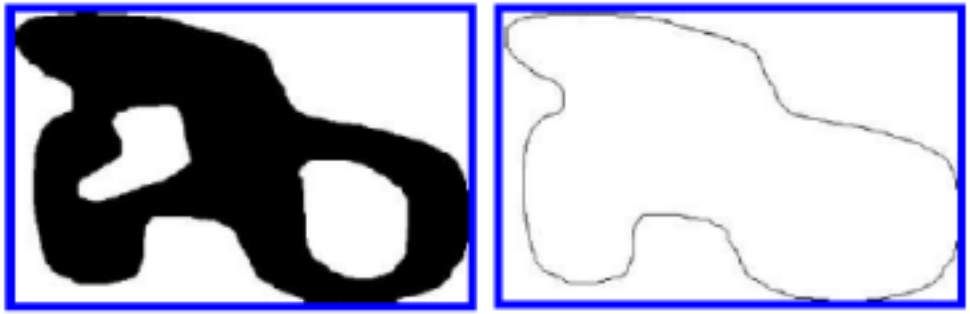
$p(k, s)$ = the corresponding neighbouring pixel; $q_k = q_k \cup \{p(k, s)\}$

} until $p(k, s) = p(k, 1)$ and $p(k, s - 1) = p(k, 0)$;

$k = k + 1$;

}
End-Algorithm

Note that when $p(k, l) = p(k, 0)$, the border is closed. Figure 4.9 depicts the resulting edge image detected by edge tracing method.



(a) Original image

(b) The result edge image detected by edge tracing method

FIGURE 4.9 Edge detection by the edge-tracing method. (a) Original image, and (b) the resulting edge image detected by edge-tracing method.

- When applying this algorithm it is assumed that the image with regions is either binary or those regions have been previously labeled.
- Algorithm's steps:

Step 1:- Search the image from top left until a pixel of a new region is found; this pixel $P_0$is the starting pixel of the region border. Define a variable dir which stores the direction of the previous move along the border from the previous border element to the current border element. Assign

- dir = 0 if the border is detected in 4-connectivity.
- dir = 7 if the border is detected in 8-connectivity.

- Step 2:-

  For 4 neighborhood

✔ dir=(dir + 3) mod 4

For 8 neighborhood

✔dir=(dir + 7) mod 8 if dir is even

✔dir=(dir + 6) mod 8 if dir is odd

- Step 3:-

So look for pixel in dir direction. if it has intensity 0/1 it is next border point, p1.

If not dir=(dir+1) mod 8 i.e. keep changing the dir in anticlockwise direction, till you get a point with intensity 0/1, i.e. p1.

Now with p1 and dir=whatever was last dir, continue same set of steps (from step 2 onwards)