MV_HW4 (Re-done)

Square Tracing Algorithm.

Theory: Lets assume a grid with some black pixels and a background of white pixels. Find a black pixel and consider it as a starting point. If you are on *black* pixel turn left , if you are on *white* pixel turn right. If you come back to starting position stop the procedure. All the black pixels you came across is your boundary of the pattern.

Algorithm:
Input: A square grid T with black pixels P
Output: A list of Boundary points B.

Begin
- Initialize B as empty
- Find the starting point.
- Insert that starting point in B.
- Set the current pixel, p, to be the starting pixel, s.
- Turn left.
- Update p.
  While p is not the starting pixel do
  - If current pixel is black
    - Insert that pixel in B and turn left.
    - Update p
  - Else
    - Turn right.
    - Update p

Disadvantages: This algorithm wont work for a lot of patterns i.e family of 8 connected pattern.
This algorithm stops when the tracing point comes back to its starting position.
The example of this is shown in the figure.

Ooops!!
the algorithm has visited the
start pixel again and will
therefore terminate
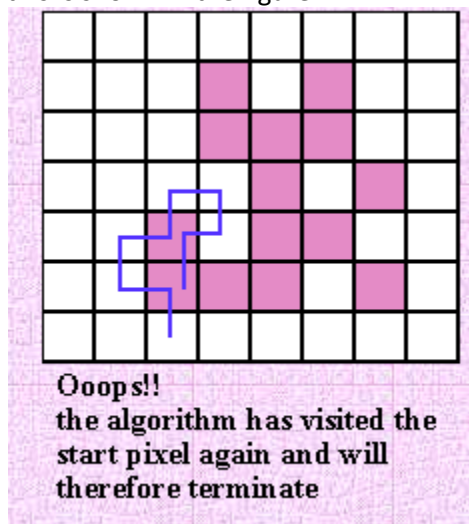
In above figure the algorithm terminates without visiting each and every node.

Reference :
[1]http://www.imageprocessingplace.com/downloads_V3/root_downloads/tutorials/contour_tracing_Abeer_George_Ghuneim/square.html

2) Moore Neighbor Tracing:

Theory : As its name suggest it checks for the 8 neighbor pixels in clock wise direction around a black pixel(P) and if found any adds it in the list of Boundary pixels.
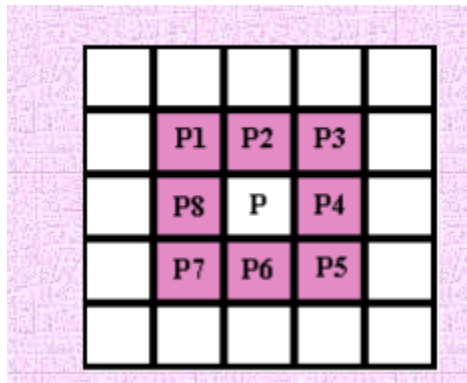
ALGORITHM :

Input: A square grid T with black pixels P
Output: A list of Boundary points B.

Begin

- Initialize B to empty
- Find starting point.
- Insert that point in B
- Set boundary point p to s.(p=s)
- Now backtrack in clockwise direction from boundary point p
- Set c to be next clockwise pixel.

  While c is not starting point
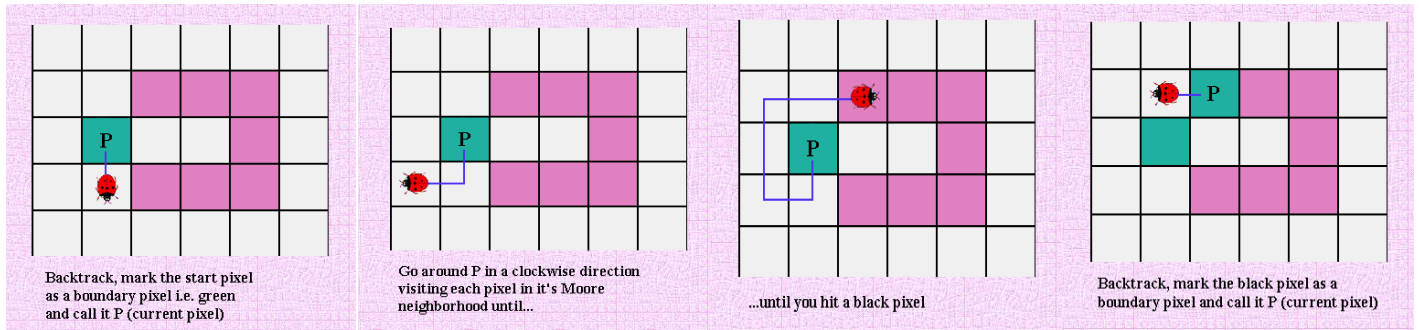   If c is black
          - Insert that pixel in B.
          - Update p such that p=c.
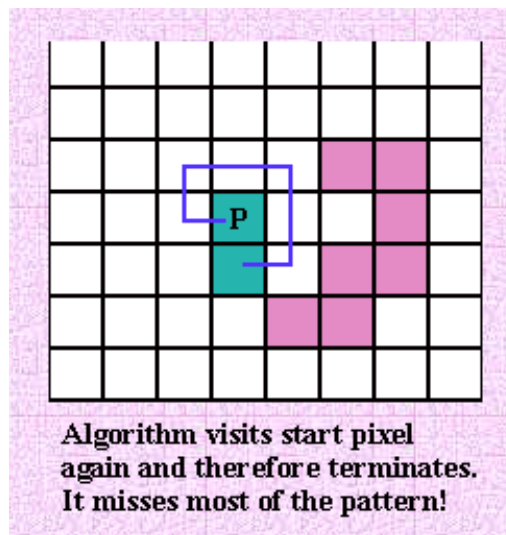   - Backtrack in clockwise direction.
      Else
          - Backtrack until you find a next black pixel.

Demonstration of this algorithm:

Backtrack, mark the start pixel as a boundary pixel i.e. green and call it P (current pixel)

Go around P in a clockwise direction visiting each pixel in it's Moore neighborhood until...

...until you hit a black pixel

Backtrack, mark the black pixel as a boundary pixel and call it P (current pixel)

Disadvantage:

Algorithm visits start pixel again and therefore terminates. It misses most of the pattern!

This algorithm stops when; searching for the next pixel reaches the starting point and so misses to trace the remaining boundary.

Reference:
[1]http://www.imageprocessingplace.com/downloads_V3/root_downloads/tutorials/contour_tracing_Abeer_George_Ghuneim/moore.html