

# Assignment 5

Harshith S  
0415617305

## Edge Linking and Line deletion

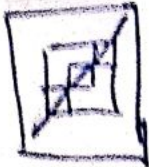
### Edge Linking

- Start with edge pixels and corresponding  $M(x,y)$   $M(s,t)$
- for each edge pixel  $(x,y)$  make a window  $W_x$  around that pixel
- for each  $(s,t) \in W_x$ , "link"

$$(x,y) \text{ to } (s,t) \text{ if } |M(x,y) - M(s,t)| \leq \epsilon_1$$

$$|x(x,y) - x(s,t)| \leq \epsilon_2$$

to trace out long edges



### Boundary following

We have edge pixels around a closed contour, want to link / order them in a clockwise direction

Moore's boundary following algorithm 1 = edge pixel  
0 = non edge

1) let starting point to be the uppermost, leftmost point labeled '1'

let  $c_0$  be the left neighbour of  $b_0$

2) Examine the 8 neighbours of  $b_0$ , starting at  $c_0$  and going clockwise let  $b_1$  be first 1 and  $c_1$  be the preceding 0  
let  $b = b_1, c = c_1$

3) Repeat process of updating  $b$  and  $c$

4) Continue until  $b = b_0$  and next boundary point found is  $b_1$

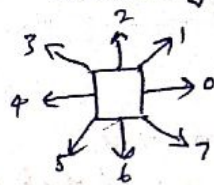
5) the ordered list of  $b_i$  is the boundary.



000566644231



Once we have such boundary can describe it with chain code



to be able to match shapes at different orientations we can

- 1) order the chain code so it always starts with the minimum mag. integer
- 2) just encode the differences & the directions

00510060067268

bw trace boundaries

Polygonal fitting to a set of ordered points

let  $P$  be a sequence of ordered distinct points (eg ordered edges after bon. follow)

Specify two starting points  $A, B$

If the curve is open,  $A, B$  are natural endpoints

If the curve is closed,  $A, B$  are the left and rightmost points specify a threshold  $T$ , (Pixel distance)

Create two stacks

Final  
B

In process  
B A

If closed curve  
(A is first curve)

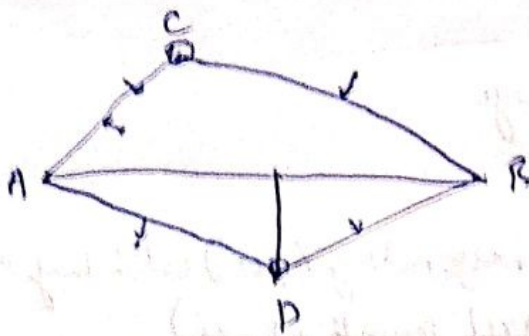
① compute line connecting the last vertices of final and in process

② compute the distances from this line to all points b/w these vertices (counter clock wise)

Select  $V_{max}$  with  $D_{max}$

③ if  $D_{max} > T$  put  $V_{max}$  at the end of "in process" and go to step 1

④ otherwise remove last vertex from in process and make it the last vertex of final.



Final  
B  
BC  
BCADB

Initial  
BAC  
BA  
=

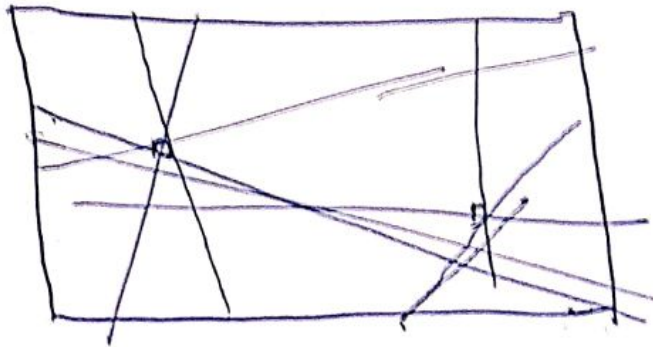
⑤ In process is not empty go to step 1

⑥ Otherwise - the vertices in final are the ordered vertices of polygon

fitting straight lines in image

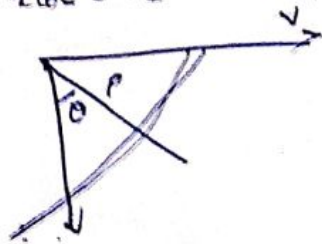
easy to do with previous algo - if one has with multiple lines

the hough transform



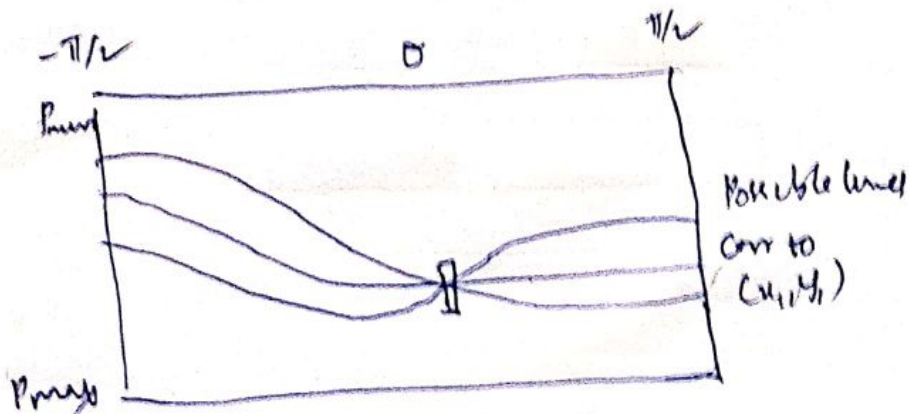
each line  
 $y = mx + b$

$$x \cos \theta + y \sin \theta = \rho$$



$\theta = 0$  horizontal  
 $\theta = \pi/2$  vertical

a single edge point  $(x, y)$  could belong to many possible lines in the  $(\rho, \theta)$  space





### Basic Idea

- ① Detect edge points  $\rightarrow$  binary image
- ② subdivide of  $(x, y)$  plane
- ③ for each point increment corresponding  $(x, y)$  cells by 1
- ④ look for  $(x, y)$  cells with large pixel count (Peak)
- ⑤ select highest peak
- ⑥ map the corr.  $(x, y)$ 's into lines on the  $(x, y)$  plane  
(or get line segments that correspond to votes)