M30242 Graphics and Computer Vision

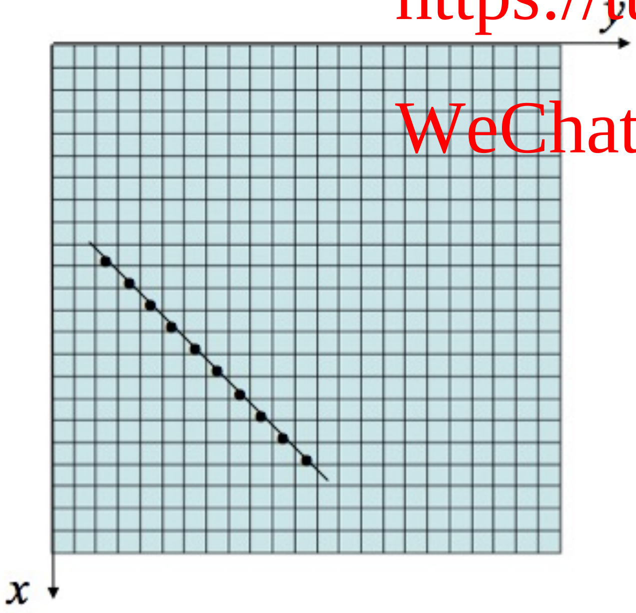
Lecture 6 Hough Transform

Overview

- Hough transform in polar coordinates
- Hough transform for circle detection
- Intro to Generalised Hough Transform (GHT)

Recap: Line Detection

- Aim: Find the equation for a line in an image by finding its slope a and intercept b so that we can write it as $y = ax + b$



$y = ax + b$ $\quad a = ?$ $\quad b = ?$

Hough Transform Algorithm

Algorithm(slope-intercept representation)

1. Discretise the parameter space [a_{min}, …, a_{max}] [b_{min}, …, b_{max}]

2. For each image pixel (x,y) {

    For ($a=a_{min}$, $a<=a_{max}$, a++) {

        $b=-xa+y$ //compute b from a

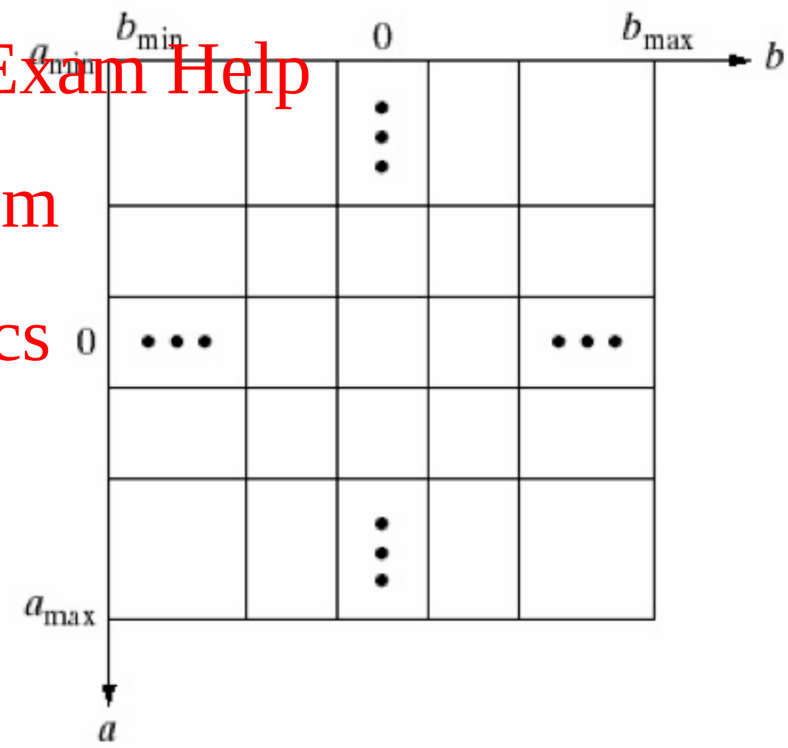
        If ($b>=b_{min}$ *and* $b<=b_{max}$) p[a][b]++

    }

  }

3. Find local maxima in P[a][b]

P[a][b]



Questions from Last Lecture

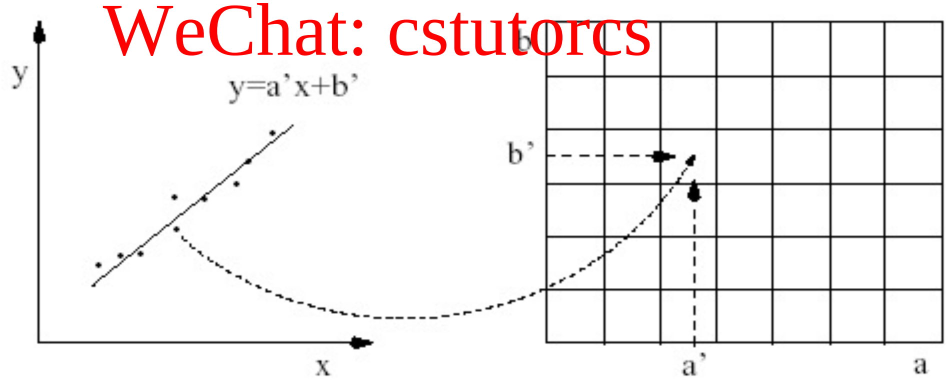
- The way of dealing with the ranges of parameters: a: -∞ to +∞ and b: -∞ to +∞
  - limit the ranges to finite values, e.g., from -200 to +200
- **Are these ranges good choices**?

- The resolution of discretisation, e.g.,
  - a: -200, -195, -190,….+185, 190, 195, 200
  - b: -200, -195, -190,….+185, 190, 195, 200

- Is the division a good choice?
- **What implication it may have on the performance of HT?**

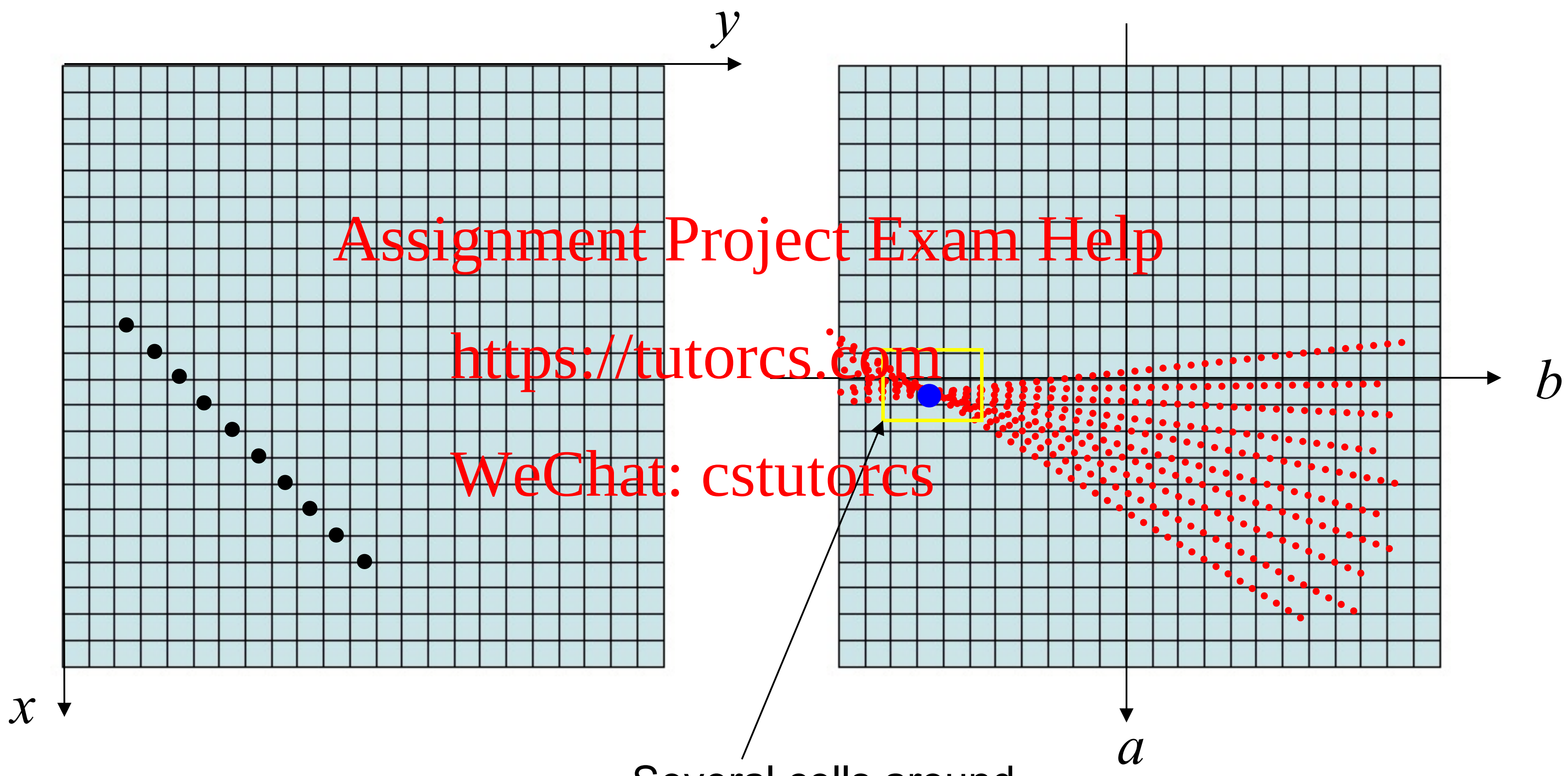
Effects of Discretisation

- Effects of discretisation/quantisation:
  - The parameters of a line can be estimated more accurately using a finer quantisation of the parameter space.
  - Finer quantisation increases space and time requirements.
  - For noise tolerance, however, a coarser quantization is better.



$y=ax+b$

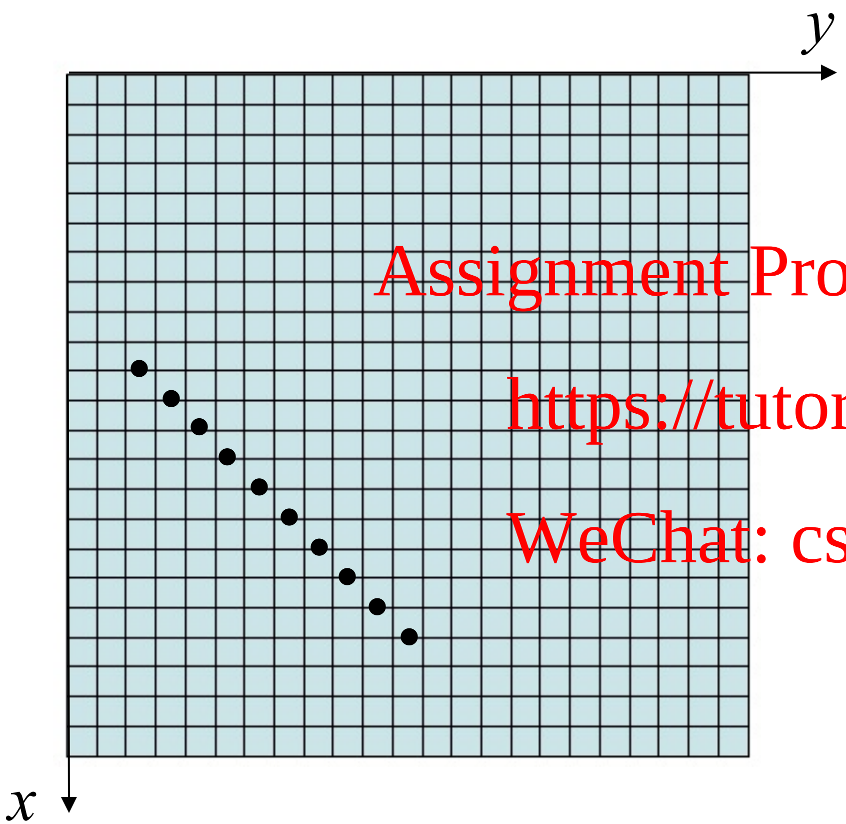
$a_1=1$
$b_1=-8$ $\Rightarrow$ $y=x-8$

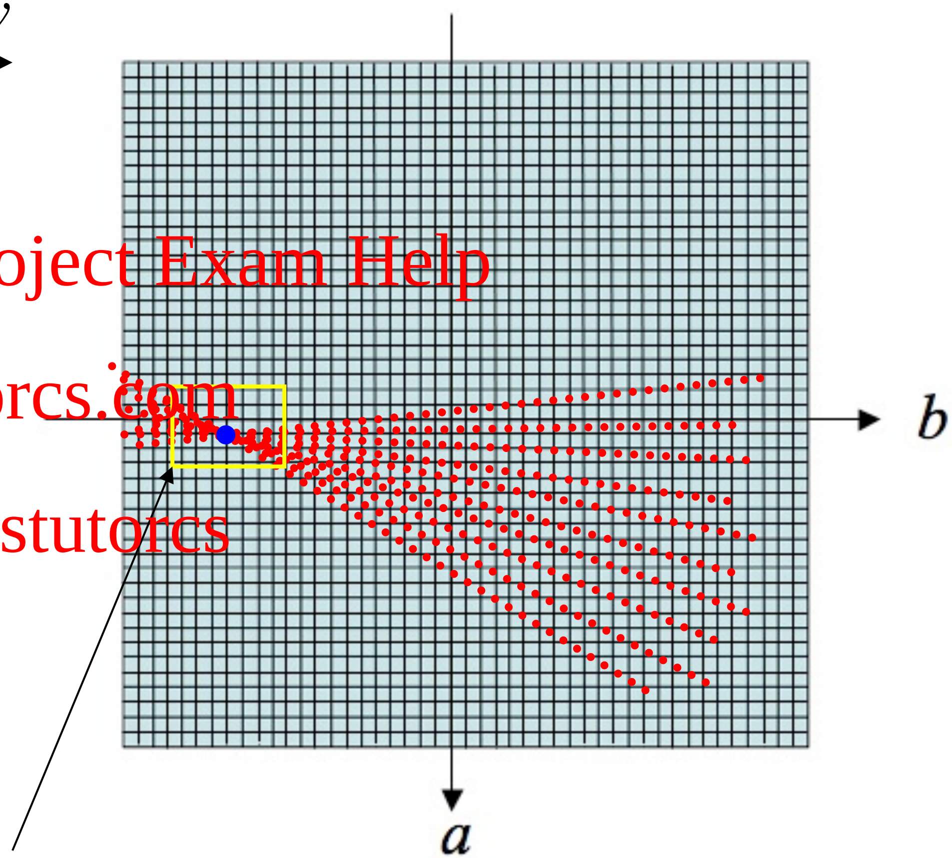


Several cells around here may have high number of votes

$y=ax+b$

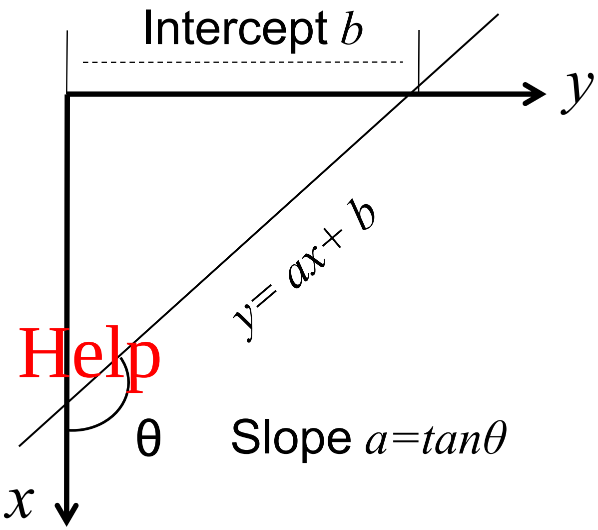
$a_1=1$
$b_1=-8$ $\Rightarrow$ $y=x-8$



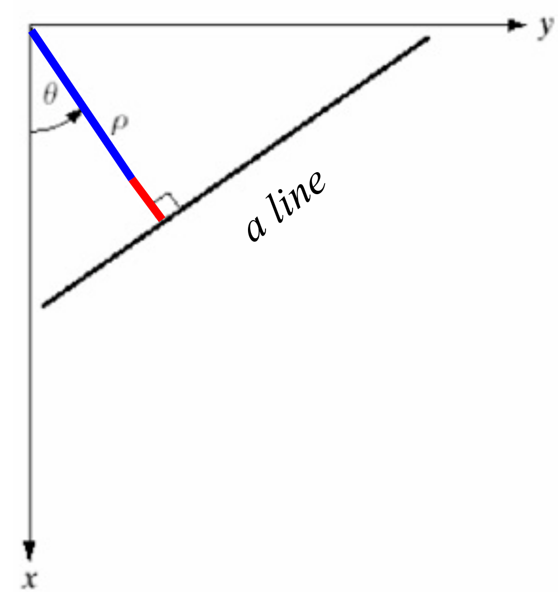


Increase the resolution of quantisation can improve the accuracy, but…

Polar Representation of Lines



- Problem with slope-intercept equation:
  - The slope and intercept can become very large or even infinite (e.g., horizontal lines).
- The problem can be overcome by using the line equation in polar coordinate system – the polar representation.
- In the polar coordinate system, a line is defined by a polar angle θ and a polar radius ρ.



Line Eq. in Polar Coord.

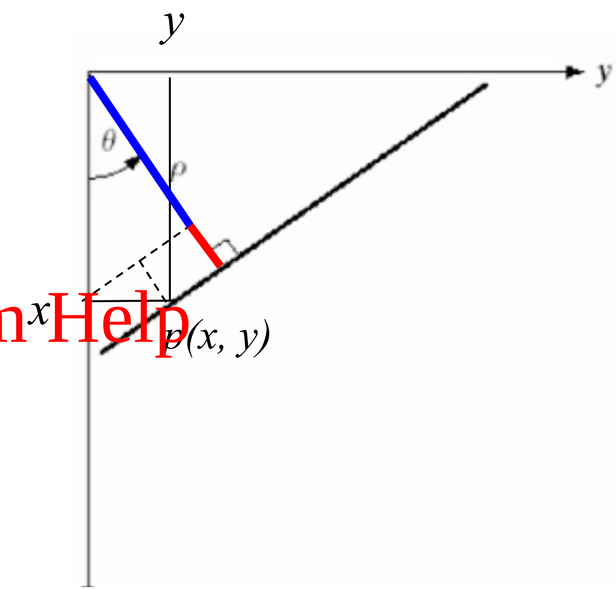
- Polar equation of lines: for any point on the line, the following equation holds:

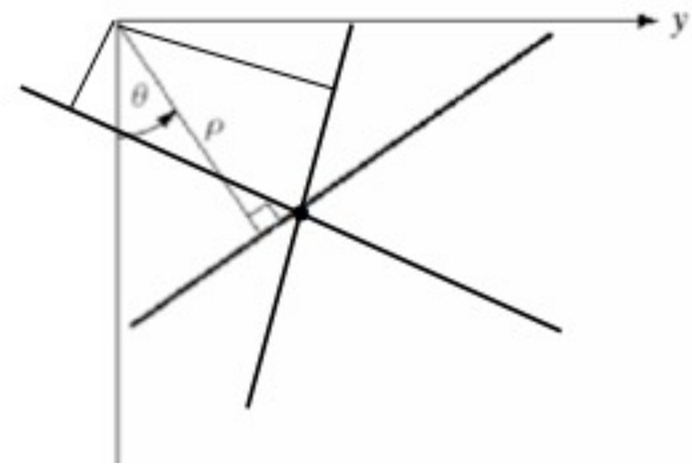
$$\rho = x \cos\theta + y \sin\theta$$

- Examples
  - Horizontal line: $\theta=0$, $\rho=$ *some value*
  - Vertical line: $\theta=90^o$, $\rho=$ *some value*
- Now the ranges of parameters become

θ: [-π/2, π/2) or [0, π)

ρ: [0, ∞)



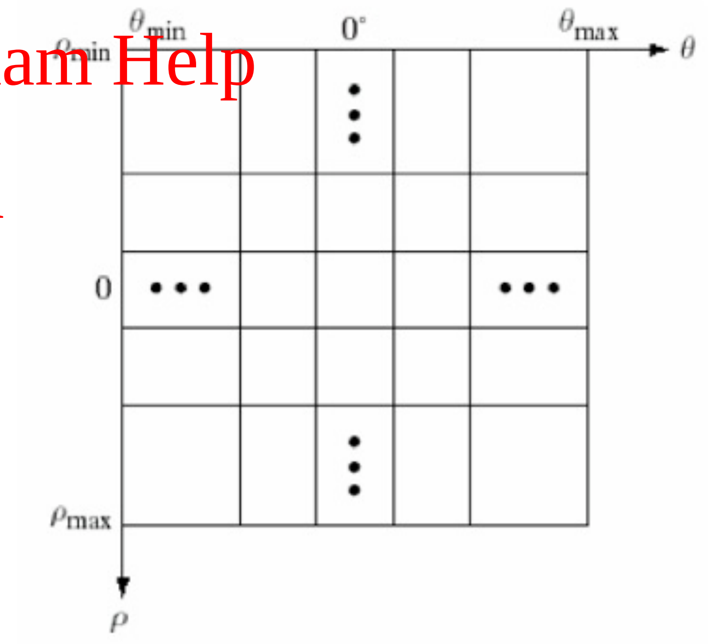


Lines that pass a pixel

HT in Polar Coordintes

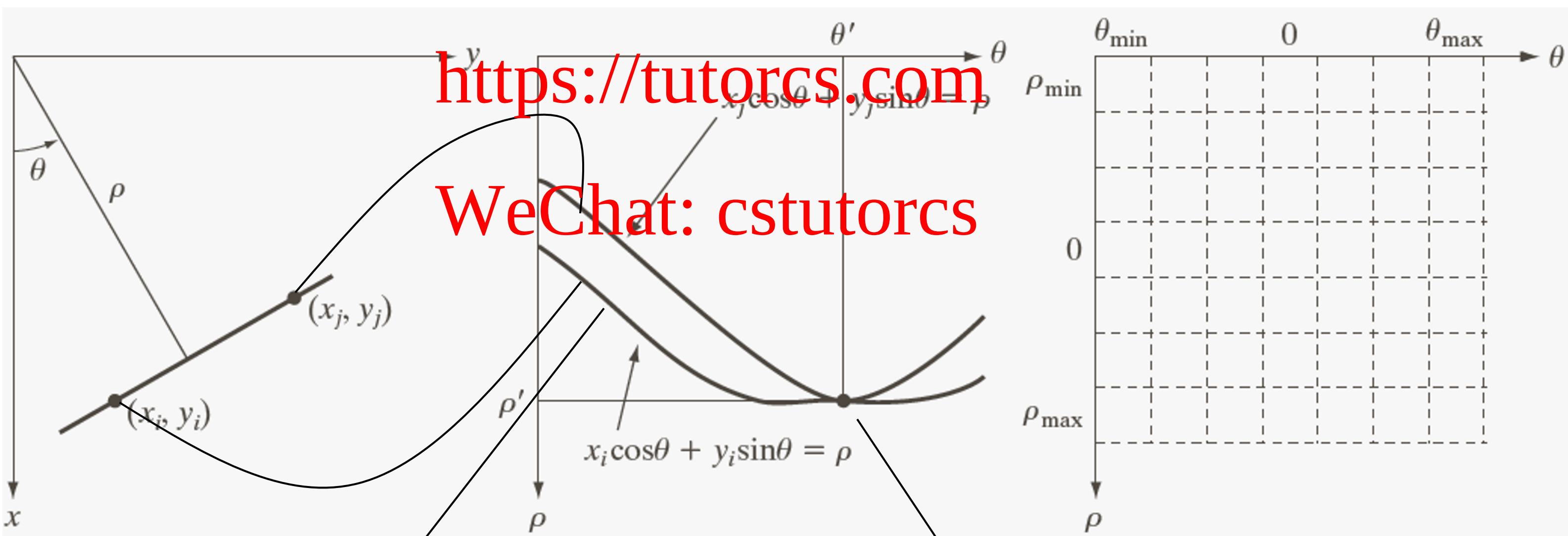
Algorithm(polar representation)

1. Quautise the parameter space $[\rho_{min}, \ldots, \rho_{max}]$
   $[\theta_{min}, \ldots, \theta_{max}]$
2. For each edge point (x,y) {
   For ($\theta=\theta_{min}$, $\theta<=\theta_{max}$, θ++){
   $\rho= x \cos\theta + y \sin\theta$ //compute ρ
   If ($\rho>=\rho_{min}$ *and* $\rho<=\rho_{max}$) $p[\rho][\theta]$++
   }
   }
3. Find local maxima in P[ρ][θ]



Visual Interpretation of HT

- For a given image pixel (x,y), a line passing through the pixel is represented as a ρ-θ pair (a point in the parameter ρ-θ space).
- If plotted, the ρ_i-θ_i pairs of all the lines passing through a pixel will form a sinusoidal curve in ρ-θ space.

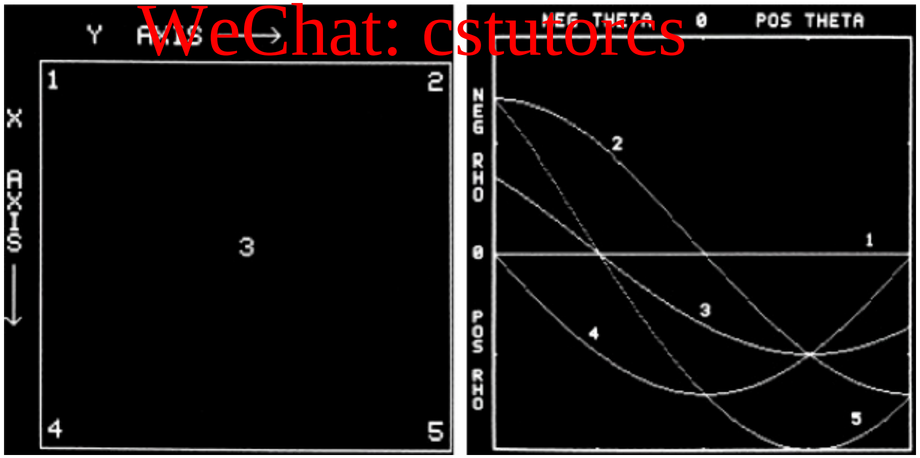


This sinusoidal curve represents all the lines passing through image point (x_i, y_i)

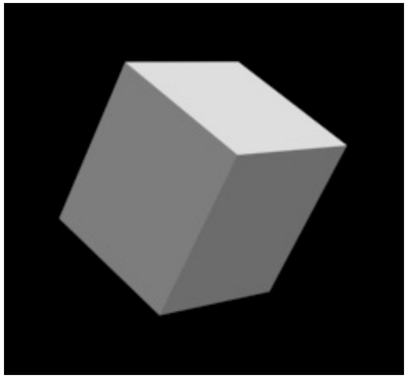
The intersection point of all the sinusoidal curves are the parameters of (ρ-θ pair) of the line being detected.

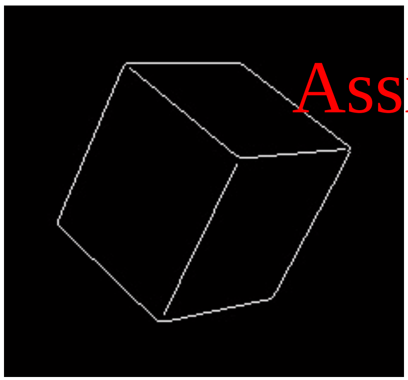
Cont'd

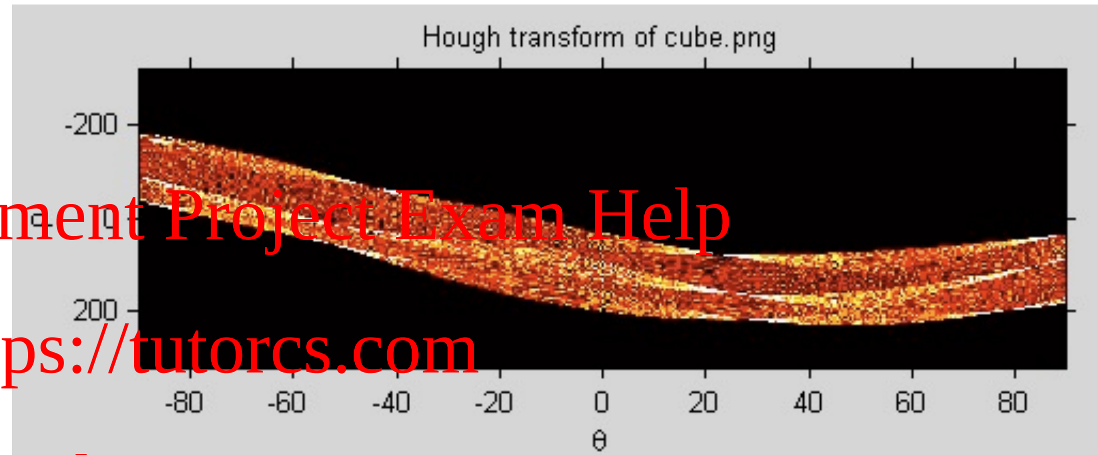
- Each point in x-y space are transformed into a sinusoidal curve in ρ-θ space
- Because the line being detected passes though all the image points and have the same ρ and θ pair, all the sinusoidal curves will intersect at that pair of ρ-θ

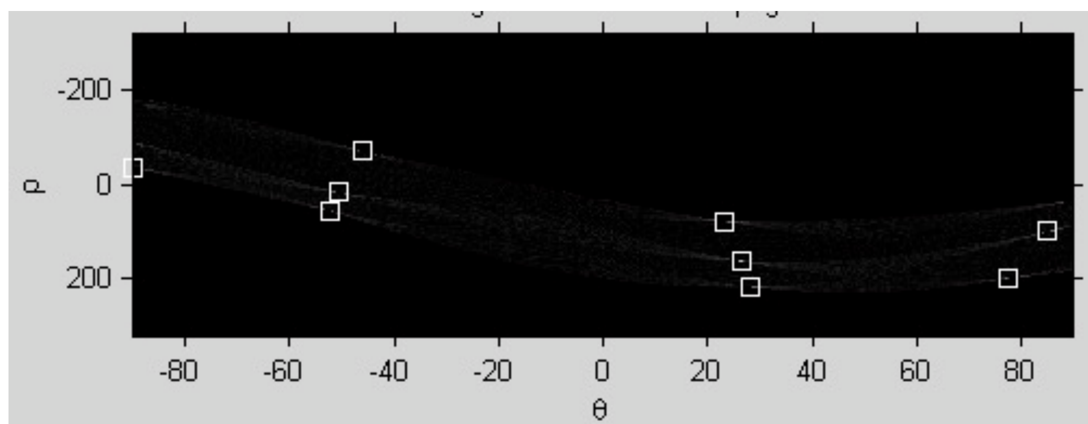


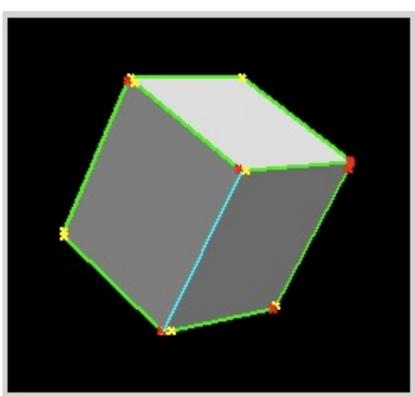
Example






Hough transform of cube.png
-200
200
-80
-60
-40
-20
0
20
40
60
80
θ


-200
0
200
ρ
-80
-60
-40
-20
0
20
40
60
80
θ



HT for Circle Detection

- The idea of Hough Transform can be extended for detecting other shapes, e.g., circles, ellipses or even general 2D shapes.

- The equation of a circle is:

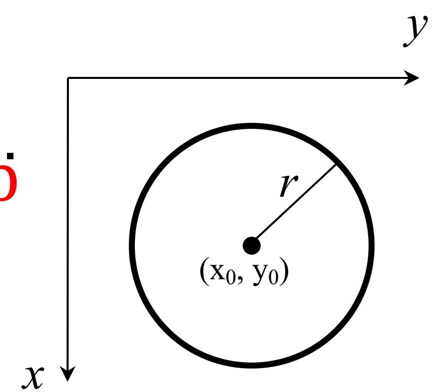
$$(x - x_0)^2 + (y - y_0)^2 = r^2$$

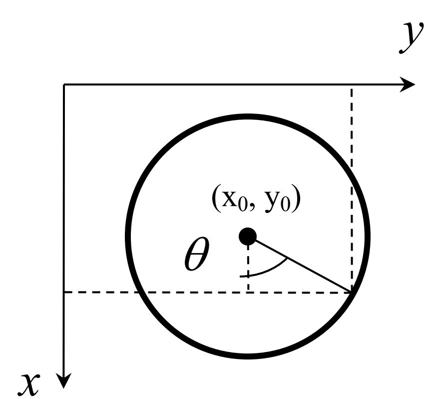
Or in parametric form:

$$x = x_0 + r\cos\boldsymbol{\theta}$$

$$y = y_0 + r\sin\boldsymbol{\theta}$$

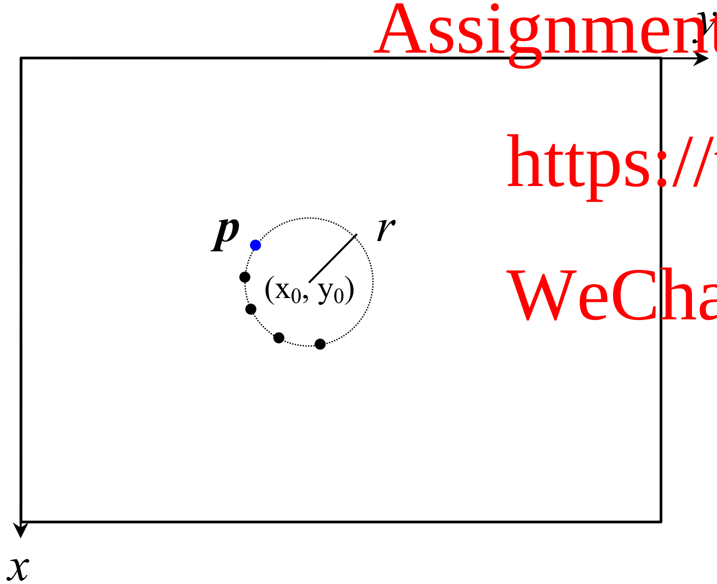
- To determine a circle, we need to know three parameters:
  - x_0 and y_0 determine its location,
  - r determines its size.





Circles of Fixed Sizes

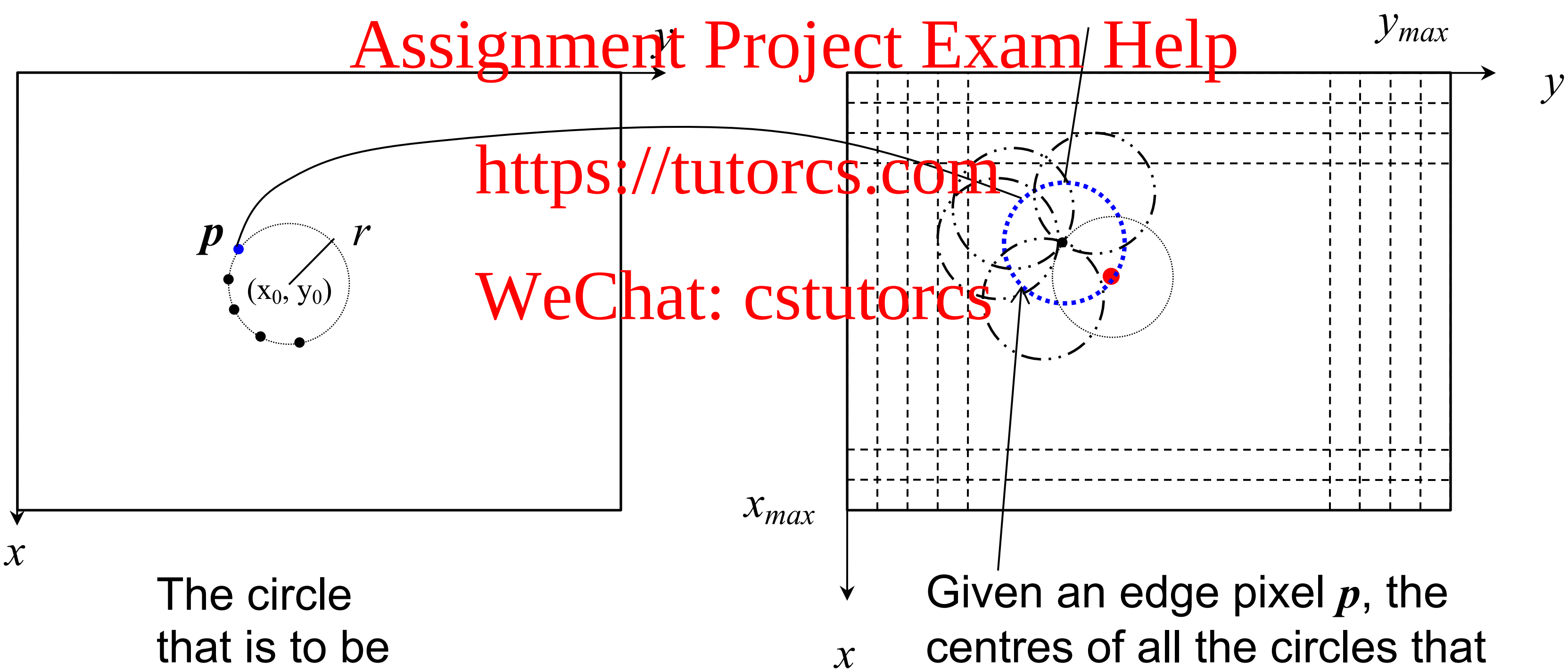
- Detection of a circle of fixed/known size r is to determine its location, i.e., find its centre (x_0, y_0).



The circle that is to be detected

Given an evidence pixel, P (x_i, y_i), on the circumference of the circle, the locus of the centres of all the circles that pass through the pixel is a circle (the blue circle in the parameter space).

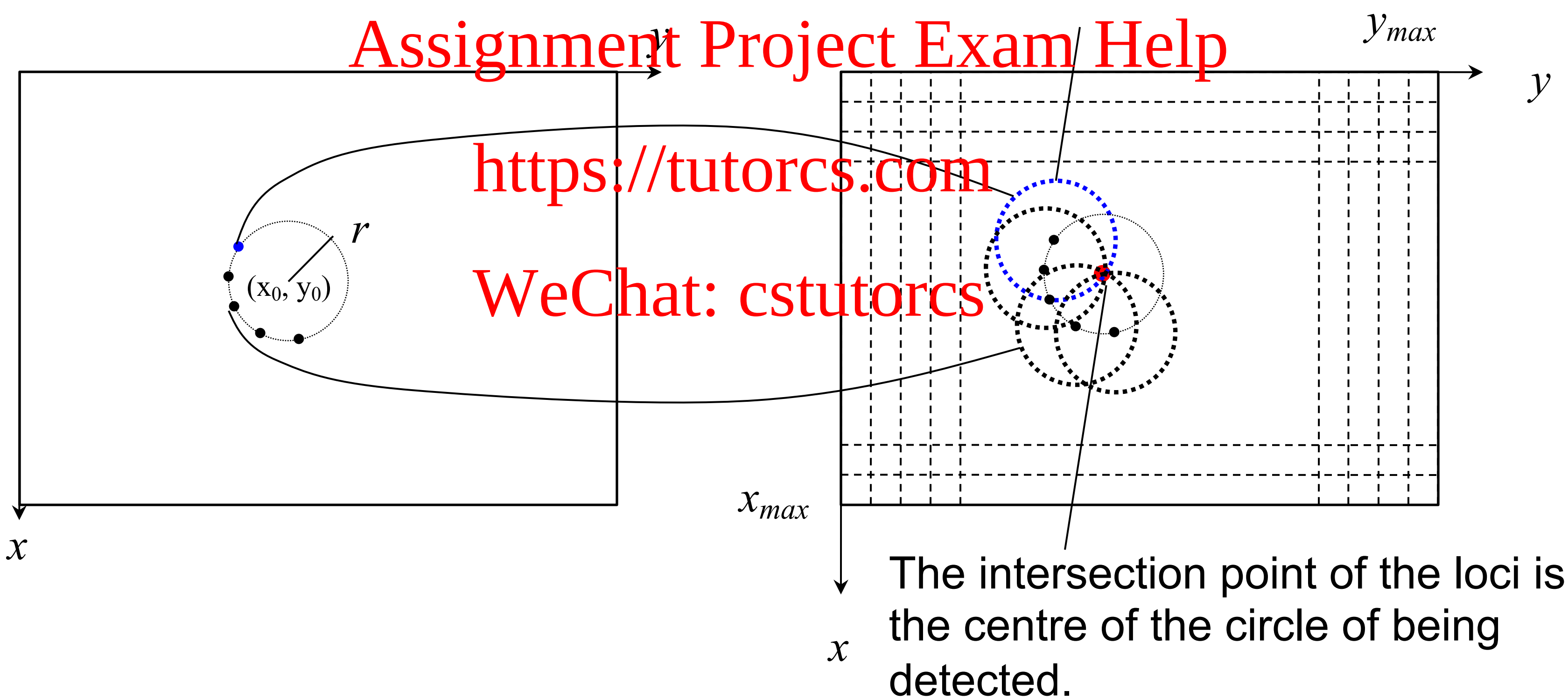
$$x = x_0 + r\cos\boldsymbol{\theta}$$
$$y = y_0 + r\sin\boldsymbol{\theta}$$



The circle that is to be detected

Given an edge pixel ***p***, the centres of all the circles that pass ***p*** form a circle (blue dotted circle).

$$x = x_0 + r\cos\boldsymbol{\theta}$$
$$y = y_0 + r\sin\boldsymbol{\theta}$$



Detection Procedure

- Procedure
  - Discretise parameter space x_c-y_c. This space consists of all the possible locations of the centre – the entire image. [1,…, x_{max}], [1, . . ., y_{max}]
  - Also, discretise $\boldsymbol{\theta}$: [0, …, $\boldsymbol{\theta}_k$, …, 2π]
  - For each edge pixel with coorindates (x_i, y_i), create a circle of the known r centred at (x_i, y_i) and calculate the edge points of the created circle using

$$x_e = x_i + r\cos\boldsymbol{\theta}_k$$

$$y_e = y_i + r\sin\boldsymbol{\theta}_k$$

  - Use (x_e, y_e) to vote in parameter space x_c-y_c. (find the bin for x_e, y_e in or [1,…, x_{max}], [1, . . ., y_{max}]
  - Find the bin having the maximum number of votes.

GHT

- The idea of Hough transform can be generalised for detection of arbitrary 2D shapes (i.e., shapes having no simple analytical form).

- This is the so called Generalised Hough Transform (GHT)

- Given any fixed 2D shape, GHT can detect its position in an image.

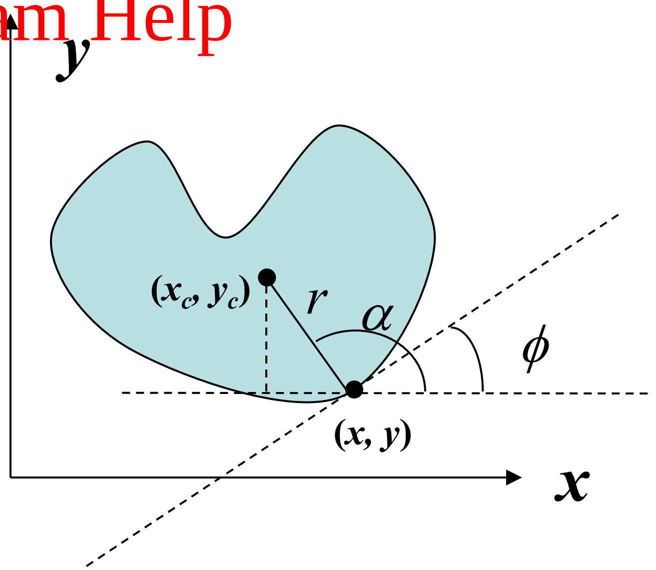
Define General Shapes

Consider the case of shapes that have fixed orientation and size.

- For general shapes, we define their shapes by specifying their boundary points (do you have any other way?).
- If we choose, arbitrarily, a centre at $C(x_c, y_c)$, then the relationship between the centre and a point on the boundary can be expressed as

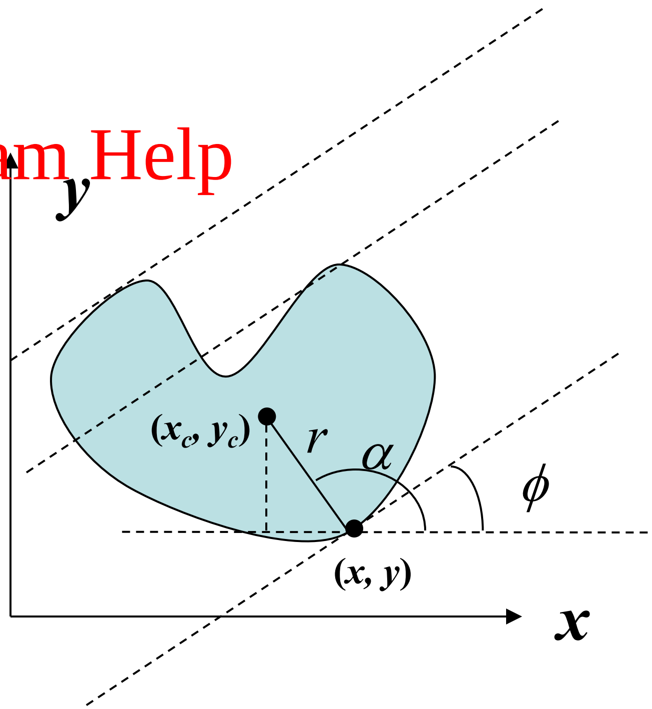
$$x = x_c + r\,cos\alpha$$
$$y = y_c + r\,sin\alpha$$

  – Where (x, y) are coordinates of a point on the boundary.



Define General Shapes

- With the centre $C(x_c, y_c)$ being chosen and fixed, for any given boundary point (x, y), we can calculate the distance from the centre to the boundary point (x, y) – the radius, r, of the shape at that boundary point.
- We can also compute the radius angle α, and the angle of tangent, ϕ, at that point with respect to the horizontal line. The ranges for α and ϕ are $[0, 2\pi)$ and $[0, \pi)$, respectively.
- **Notice** that, *for a given shape there will be several points on the boundary that have the same angle of tangent,* ϕ (but with different α).



R-Table

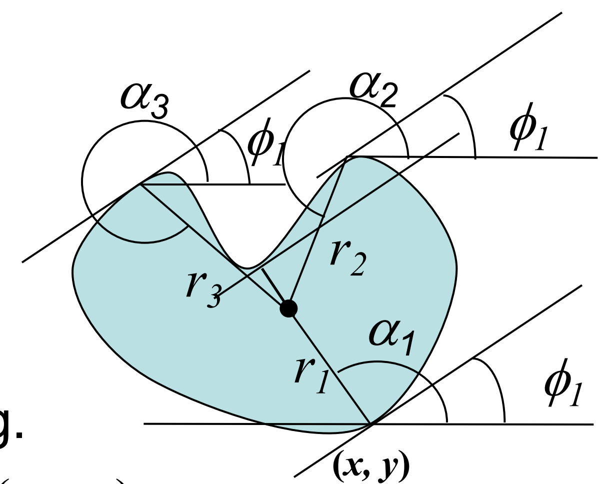
- An arbitrary 2D shape can be defined as a table of r and α pairs indexed by ϕ - the **R-table**.
- To construct the R-table for a shape,
  - Discretise the range of angle ϕ, $[0, \pi)$, into intervals
  - For each discrete angle ϕ_i, find out **all** the edge points that have the same ϕ_i and calculate r_j and α_j for those points.
  - At the end of this process, one would get

ϕ_1: $(r_1, \alpha_1), (r_2, \alpha_2), (r_3, \alpha_3),\ldots$

ϕ_2: $(r_5, \alpha_5), (r_{13}, \alpha_{13}),\ldots$

$\ldots$

ϕ_n: $(r_4, \alpha_4), (r_{91}, \alpha_{91}),\ldots$



- This process is called template R-table building.
- Note: ϕ_i $[0, \pi)$ is independent of the choice of $C(x_c, y_c)$, but r_j and α_j $[0, 2\pi)$ do.

Detection Algorithm

Algorithm (given the template R-table)

1. Quautise the parameter space $[x_{cmin}, \ldots, x_{cmax}]$, $[y_{cmin}, \ldots, y_{cmax}]$
2. For each edge point (x,y) {

    Compute ϕ (from gradient direction) at (x,y)

    Retrieve all $(r_i,\ \alpha_i)$ pairs that have ϕ as their index

    For each $(r_i,\ \alpha_i)$ pair {

        compute centre position (x_c, y_c)

$$x_c = x - r_i \cos\boldsymbol{\alpha}_i$$

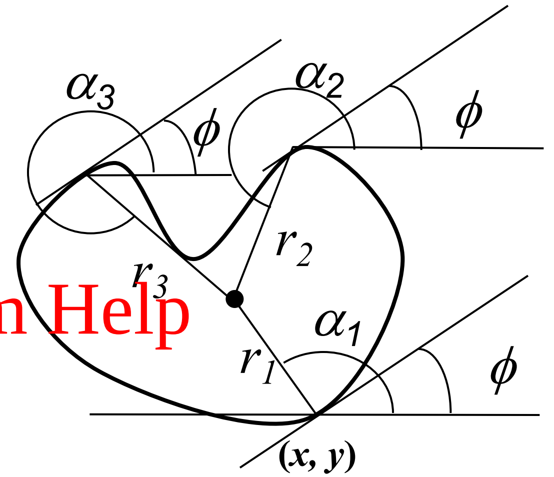
$$y_c = y - r_i \sin\boldsymbol{\alpha}_i$$

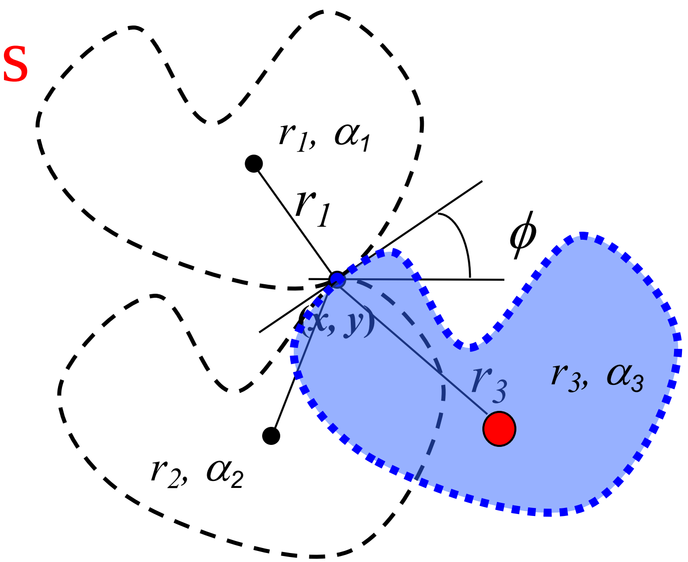
        $p[x_c][y_c]$ ++

    }

  }

3. Find local maxima in $p[x_c][y_c]$



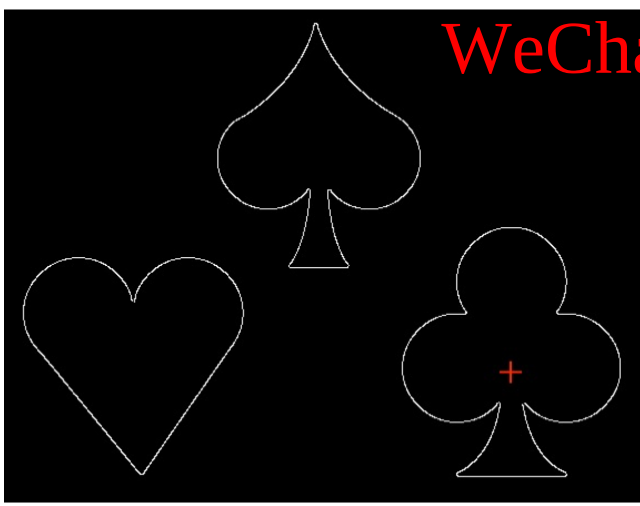


- If the local maxima of the bins is greater than some threshold, then it indicates the presence of the shape.
- Its position is given by the indices of the bin $p[x_c][y_c]$ (i.e., x_c and y_c)

Example

The image to be detected

The shape (template) to be detected



After edge detection

Votes for the centres in parameter space

Further Readings

- Shapiro, L.G., Stockman, G.C., Computer Vision, Prentice-Hall, 2001, ISBN 0-13-030796-3
  - Section 10.3.4 for Hough transform