CSCI435 / CSCI935 – Computer Vision

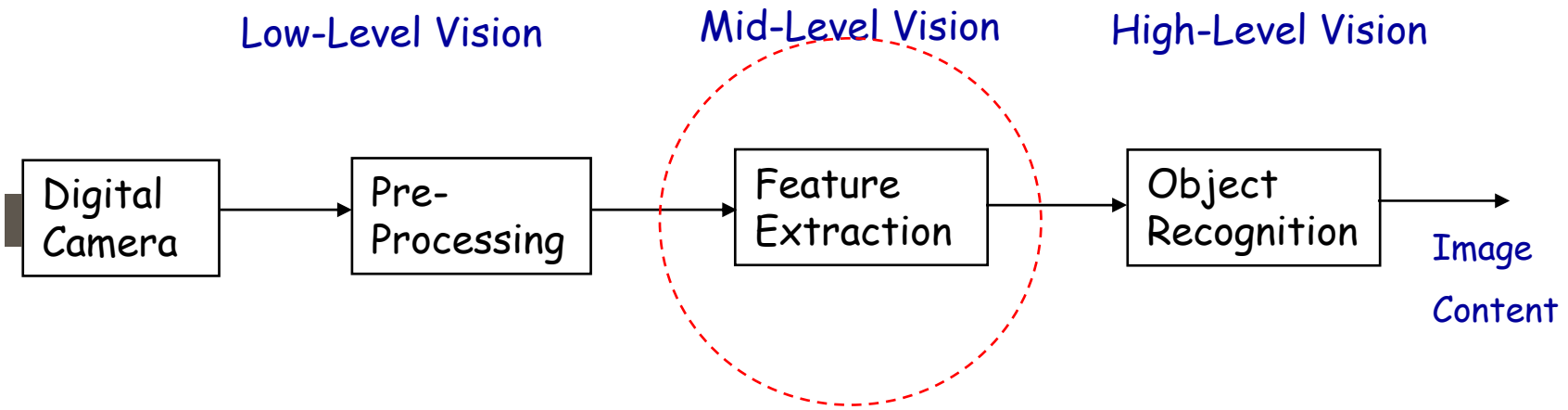
Lecture 6

Shape Detection

Lecturer: Dr. Igor Kharitonenko
Room 3.108
ph: 4221 4825
igor@uow.edu.au

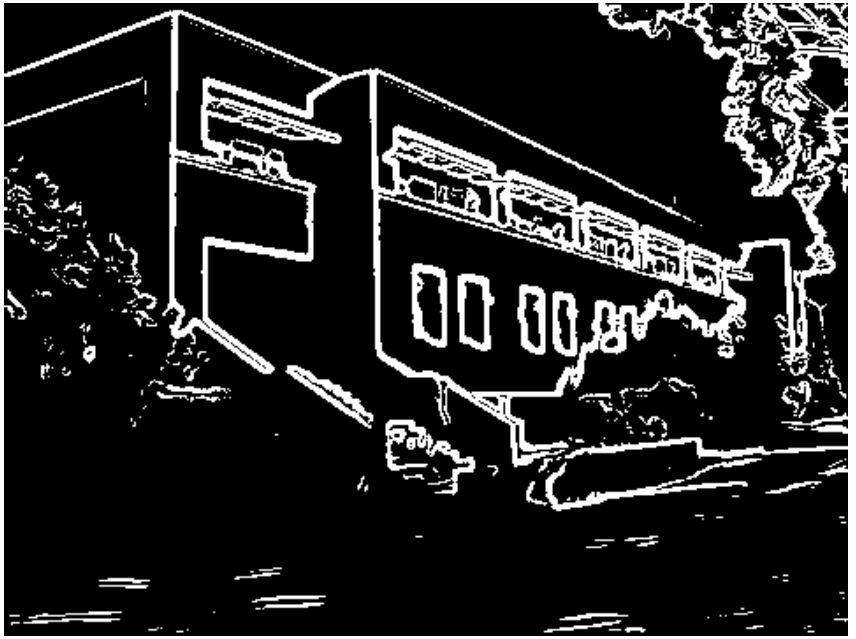
Machine Vision Concept (review)

Machine Vision is a multistage process where each previous stage affects performance of all following stages

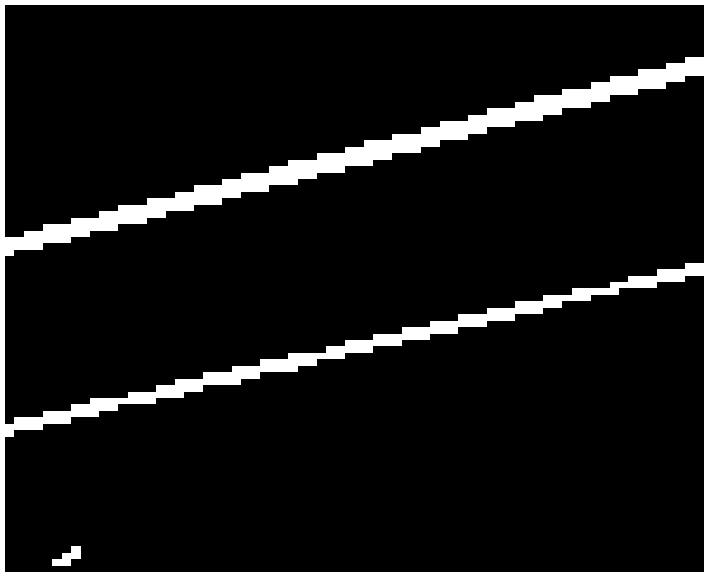


Intermediate-Level Vision

- To recognise objects their structural elements need to be extracted and analysed
- Typical structural elements are lines and basic geometric shapes



Line Detection

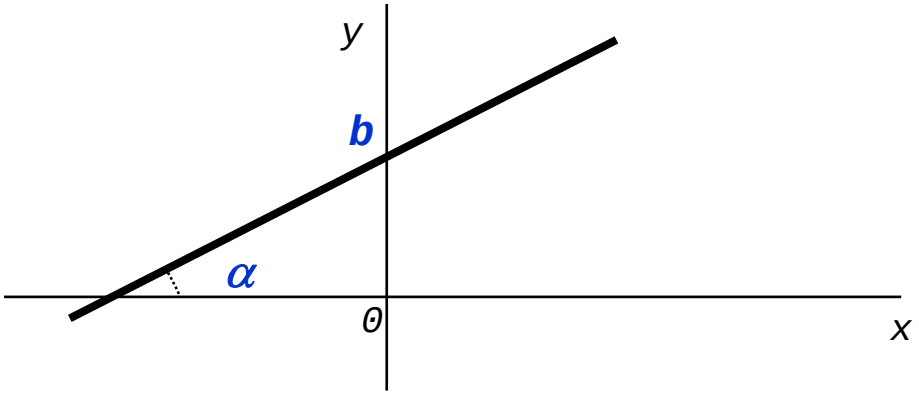


- A straight line is an abstract geometric object that is mathematically described by the equation:

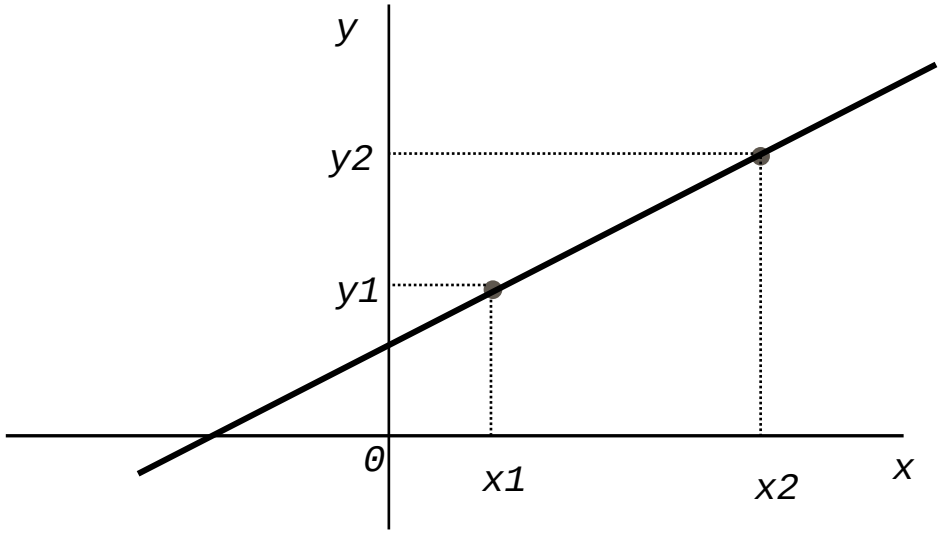
$$y = a*x + b$$

where *a* and *b* are parameters describing the line

$$a = tan\alpha$$



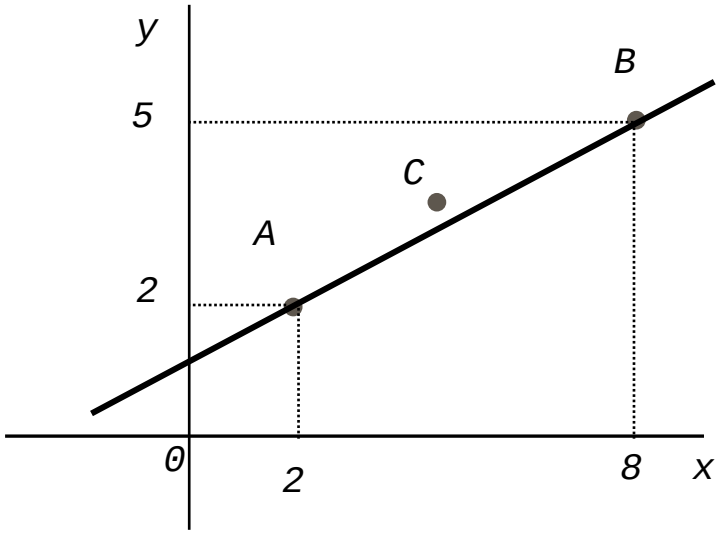
Quiz



- How many lines can go through two points?
- Given two points, how to find the equation of the straight line? (a and b)

1. `a = (y2 - y1)/(x2 - x1)`
2. as `y1 = a*x1 + b,` therefore `b = y1 - a*x1`

Quiz



- Given two points A(2, 2) and B(8, 5). How to check if the point C(5, 4) belongs to the line AB?

1. Derive the line equation using A and B

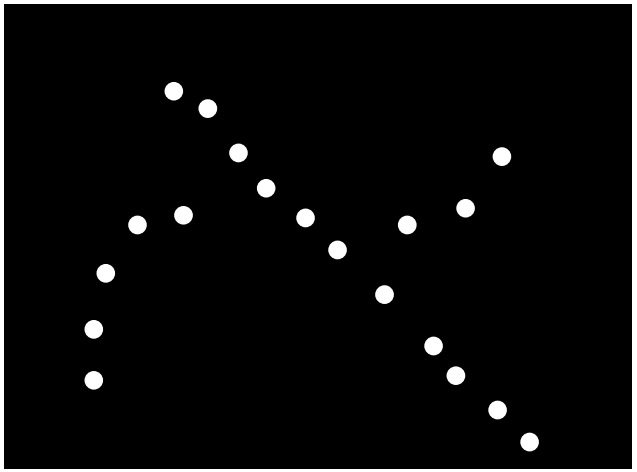
   $a = (5 - 2)/(8 - 2) = 0.5$, $b = 2 - 0.5*2 = 1$

   $y = 0.5*x + 1$

2. Check if the coordinates of C(5, 4) satisfy the equation

   $y = 0.5*5 + 1 = 3.5 \neq 4$

Problem



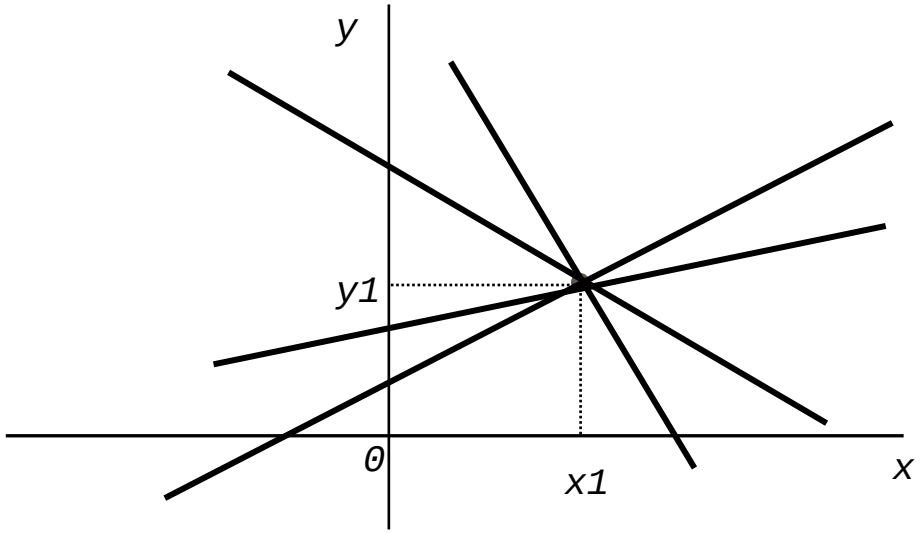
- Suppose you need to identify straight lines formed by N points in the following binary image.

What could be a possible solution?

Find all lines determined by every pair of points. Group together those lines, which have similar parameters. Calculate average parameters to represent each group by one averaged line.

N points, 2 points per line => `N!/(2* (N-2)!) = N(N-1)/2` lines

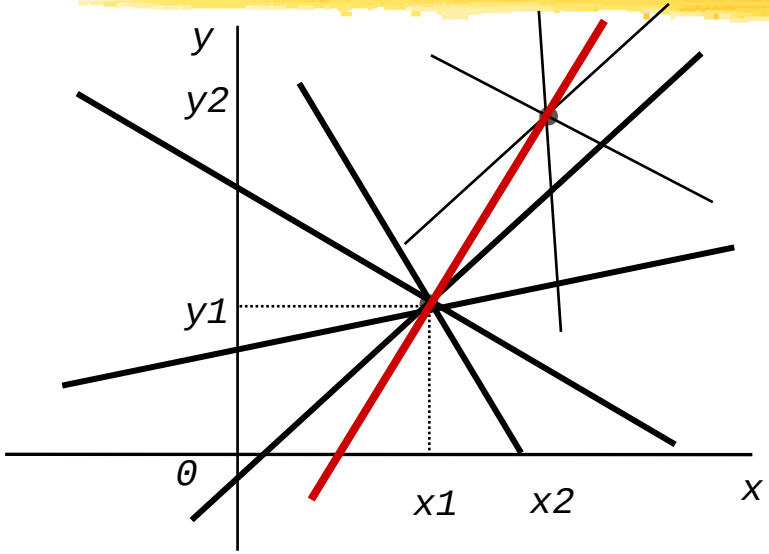
Quiz

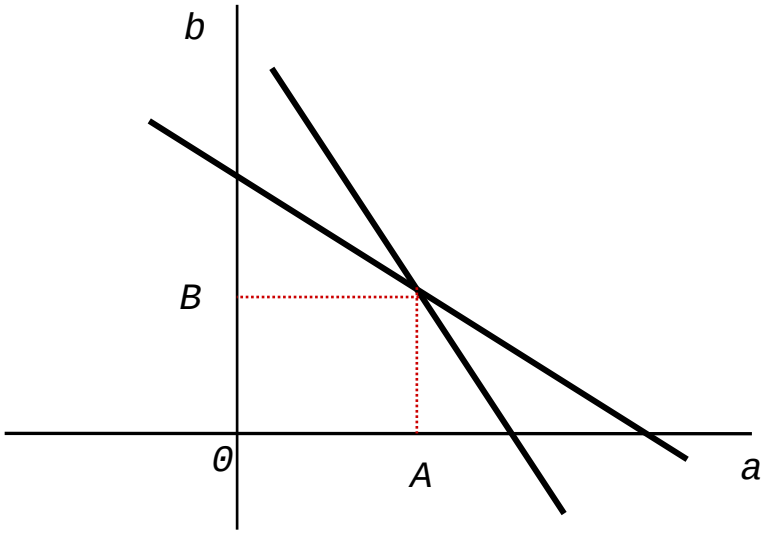


- How many lines can go through one point?
- Given one points, how to find all possible equation?

Since $y_1 = a*x_1 + b$,
thus $b = y_1 - a*x_1$ defines an infinite number of suitable pairs $<a, b>$

Hough Transform



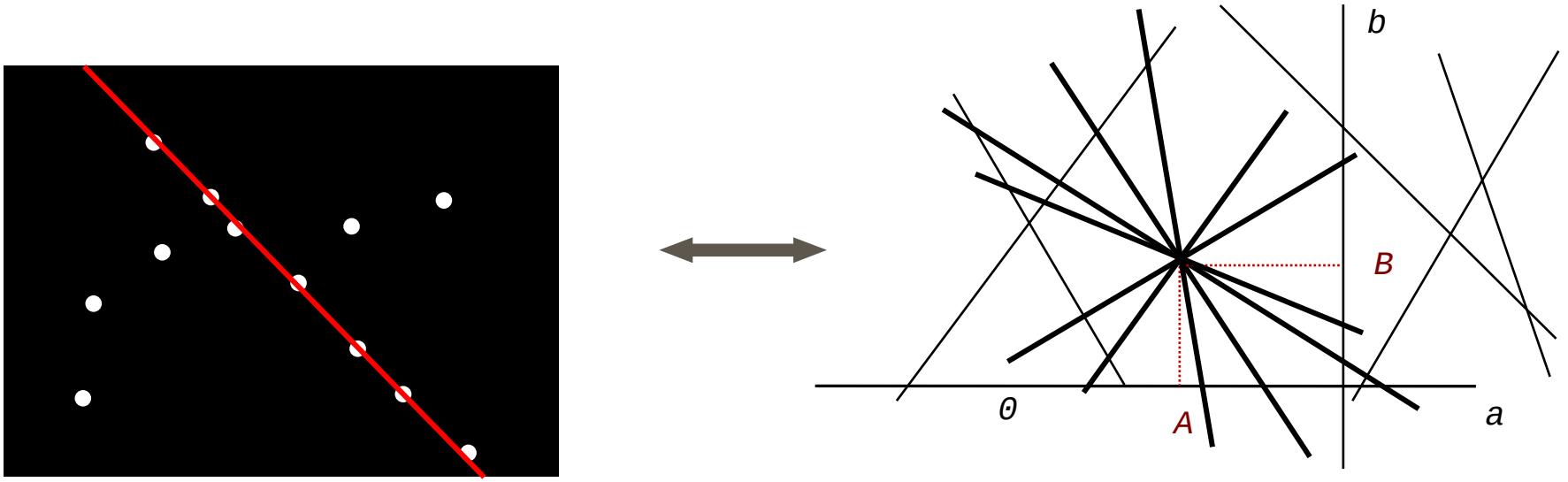


$y_1 = a*x_1 + b$ $\quad\Longleftrightarrow\quad$ $b = -x_1*a + y_1$

$y_2 = a*x_2 + b$ $\qquad\qquad$ $b = -x_2*a + y_2$

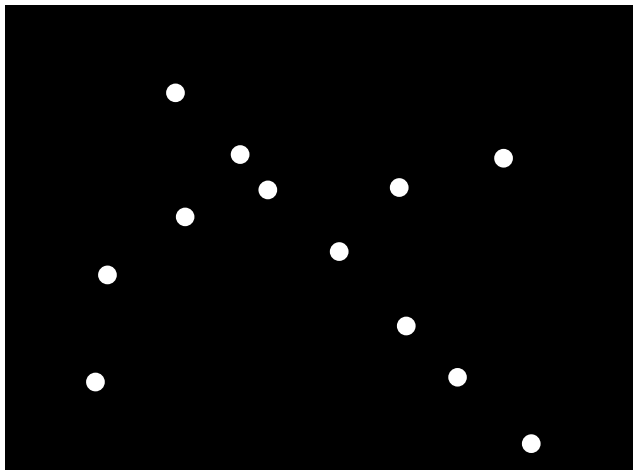
$y = A*x + B$

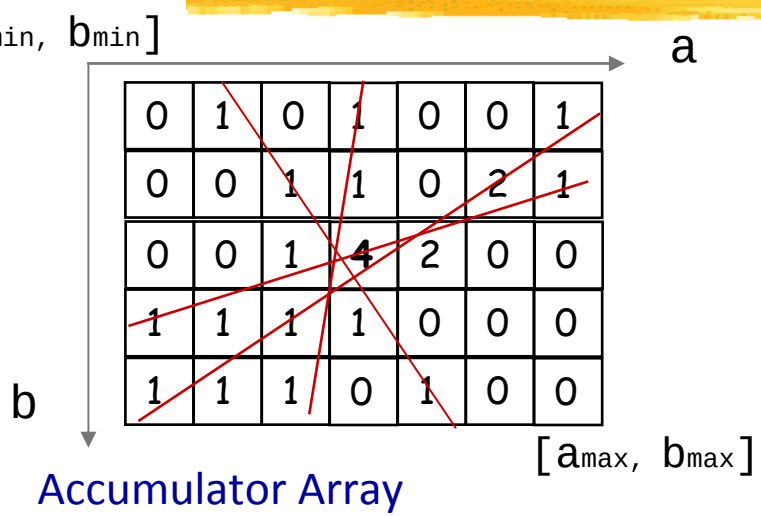
Hough Transform



- The concept is simple, but how it can be converted into a feasible algorithm?

Hough Transform





Accumulator Array

- The array elements are initialised with `0`
- For each line in (a, b) space the obtained values a and b are rounded to nearest cells
- If a line crosses a cell, its accumulated value is incremented
- The accuracy of the algorithm depends on the array size

What values should be used for $a = tan\alpha$ considering vertical lines?

The Problem of Vertical Lines

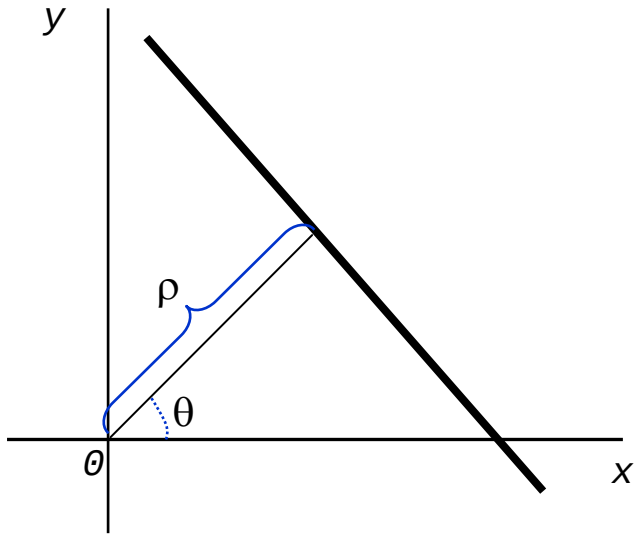
- Parameter a of vertical lines goes to infinity
- A possible solution is to use the Polar Coordinate System instead of Cartesian Coordinate System

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

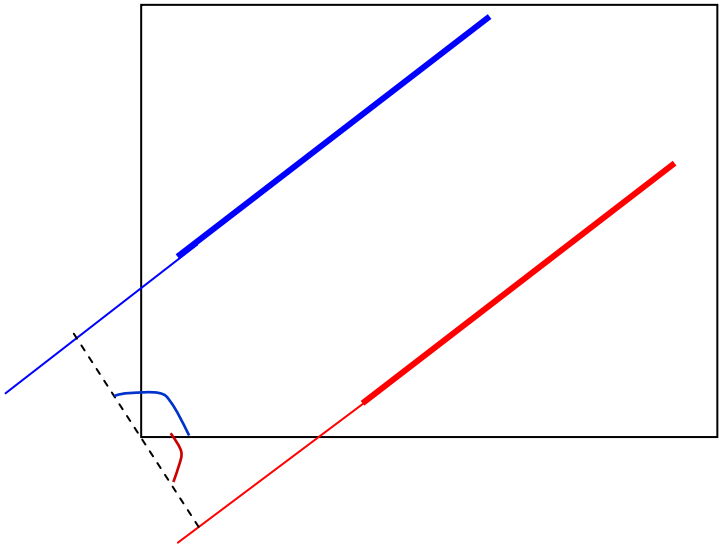
Both parameters are bounded:

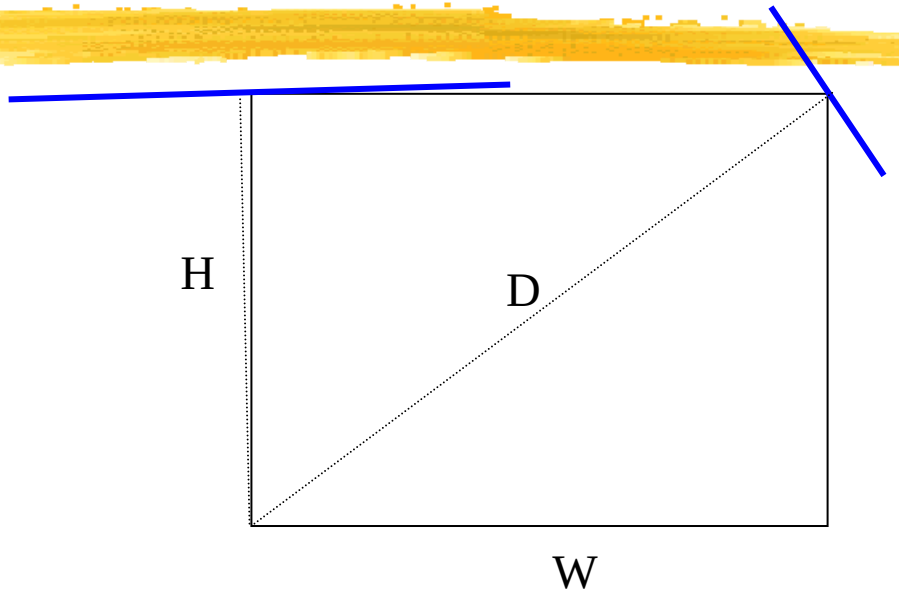
ρ cannot be greater than the diagonal of the image

θ is in the range $[0...360^{o}]$



The Parameter Range





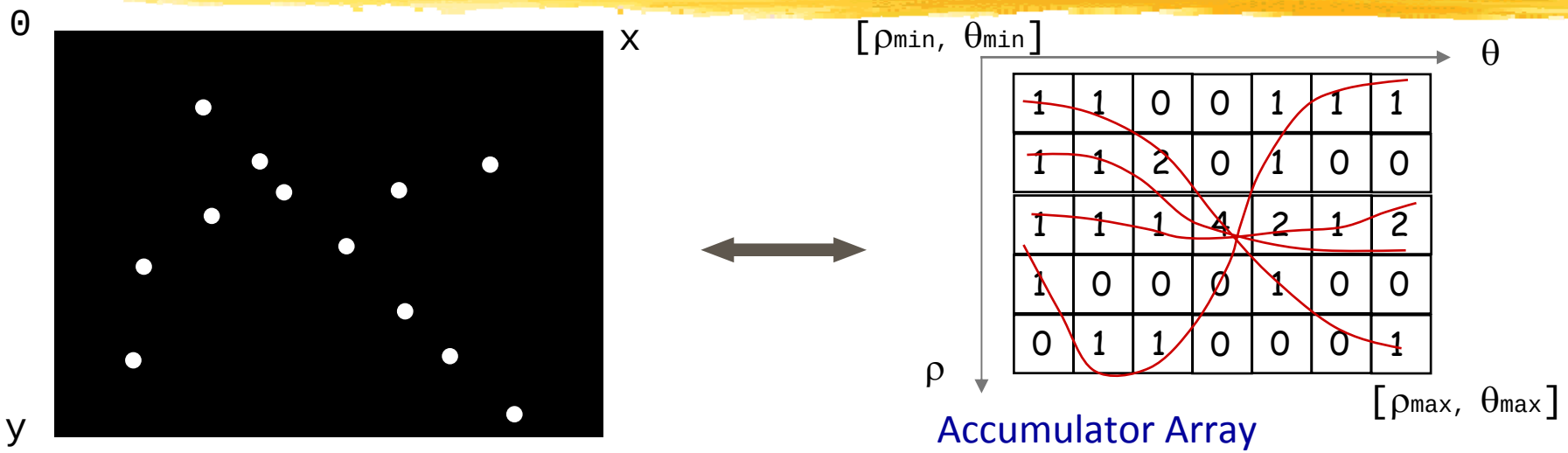
Red line: $\theta = -45^o$ $p = 10$
Blue line: $\theta = 135^o$ $p = 10$
The lines are parallel and p is positive

If the range for θ is limited to [-90 …90] , the only way to parameterize the blue line is
Blue line: $\theta = -45^o$ $p = -10$

Thus, if θ is limited to **[–90…+90]** , the range for p is: **[-H … +D]**

where $D = \sqrt{H^2 + W^2}$

Hough Transform



For each point x_i, y_i in the image space:

1. For all values of θ calculate ρ using the equation

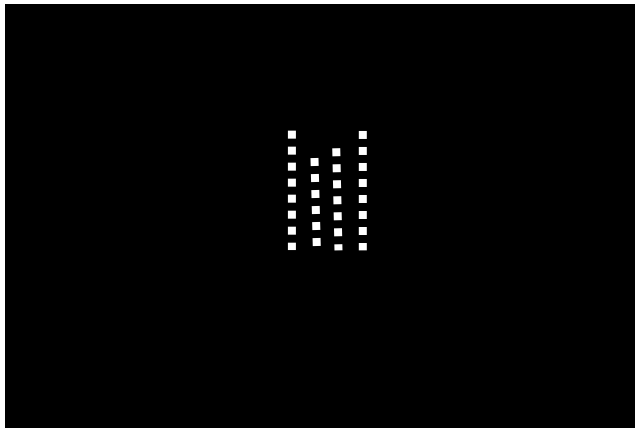
   $\rho = x_i * \cos\theta + y_i * \sin\theta$

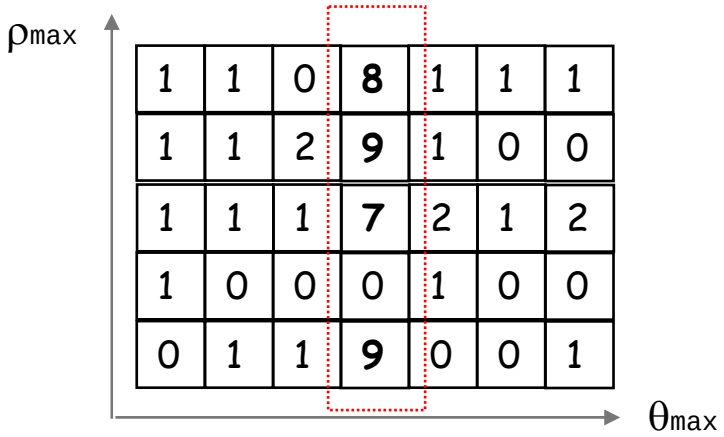
2. Increment the counter corresponding to the cell with coordinates θ, ρ

The peaks in the accumulator array correspond to the parameters of detected lines

Quiz

1. How parallel lines can be detected?





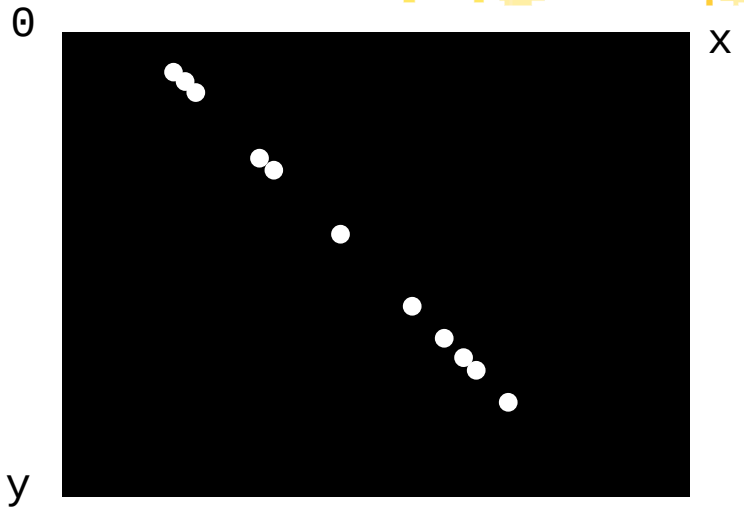
1	1	0	**8**	1	1	1
1	1	2	**9**	1	0	0
1	1	1	**7**	2	1	2
1	0	0	0	1	0	0
0	1	1	**9**	0	0	1

2. Given an image with the resolution 200x400 pixels. What size of the accumulator array is required to provide an angular resolution of 2^o ?

$$200 + \sqrt{200^2 + 400^2} \times 180/2 = 647 \times 90$$

the range for p is: [-H … +D] *the range for θ is [-90..90]*

Line Localisation



- Hough Transform may produce exactly the same result for different groups of points, which spread across a straight line described by the same parameters
- Connectivity analysis may be needed to identify continues lines. Segments shorter than a certain minimum length can be ignored

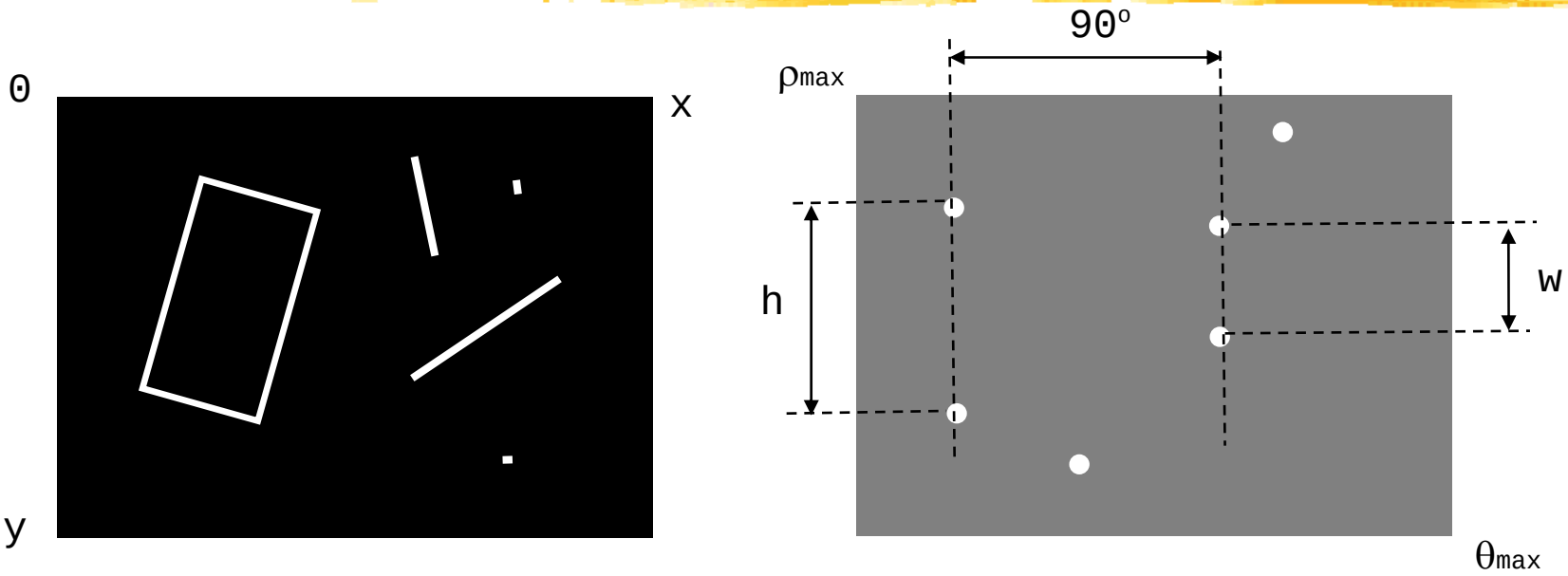
Line Thickness

- Only abstract geometrical lines have no thickness defined
- If an image line has thickness of several pixels, Hough Transform can find several lines produced by different combinations of points

  This problem can be eliminated by

  - parameter averaging of close lines
  - thinning of image lines by morphological erosion

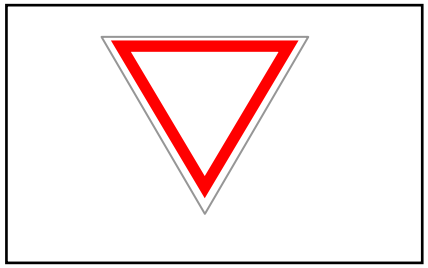
Quiz



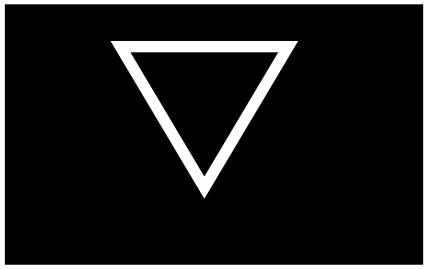
- An image can contain rectangular objects of a certain size h×w, segments of straight lines and random points generated by noise.

How would you deduce from the Hough Transform that a rectangular object is present in the image?

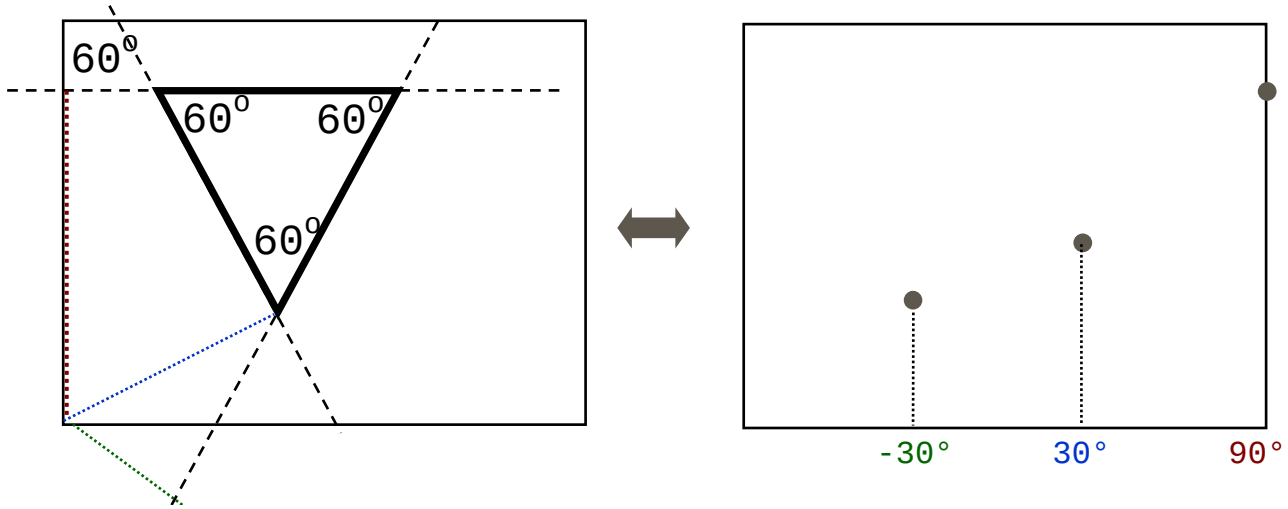
Road Sign Detection



How would you deduce from the Hough Transform that a triangular road sigh is present in the image?



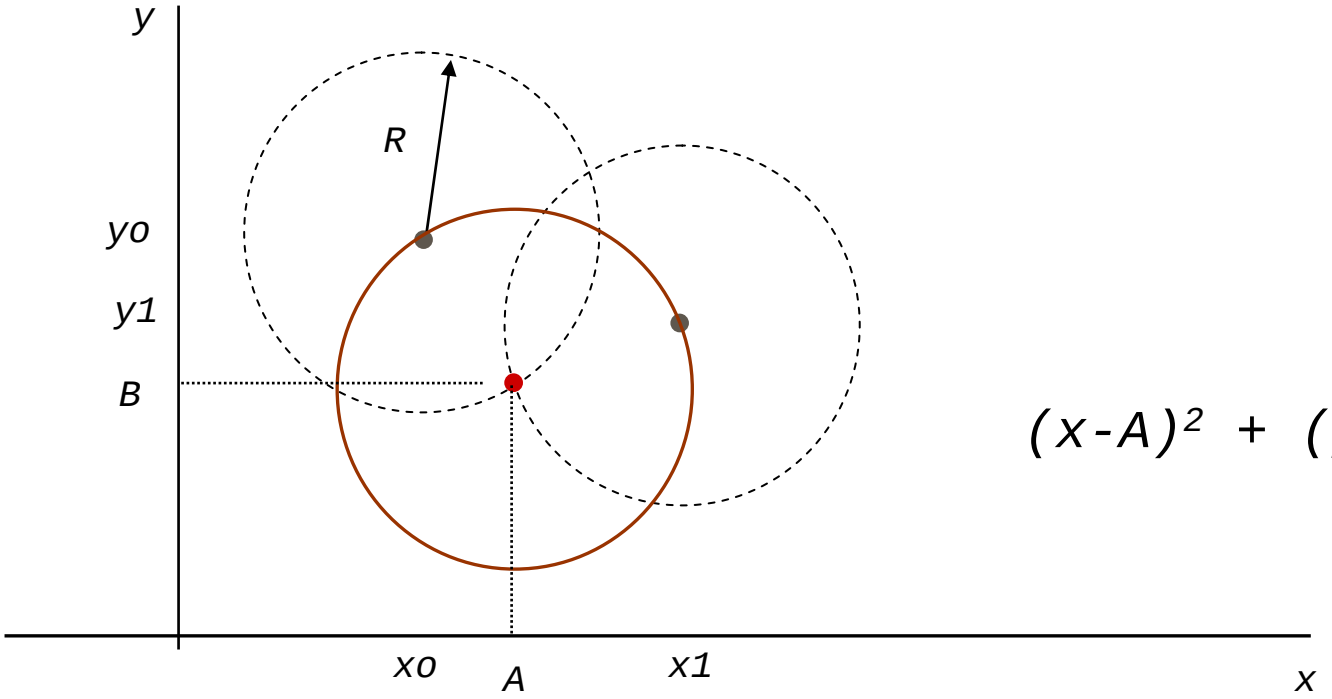
Use colour information to extract candidates



Use relations between angles and the property that the top edge of the triangle is horizontal

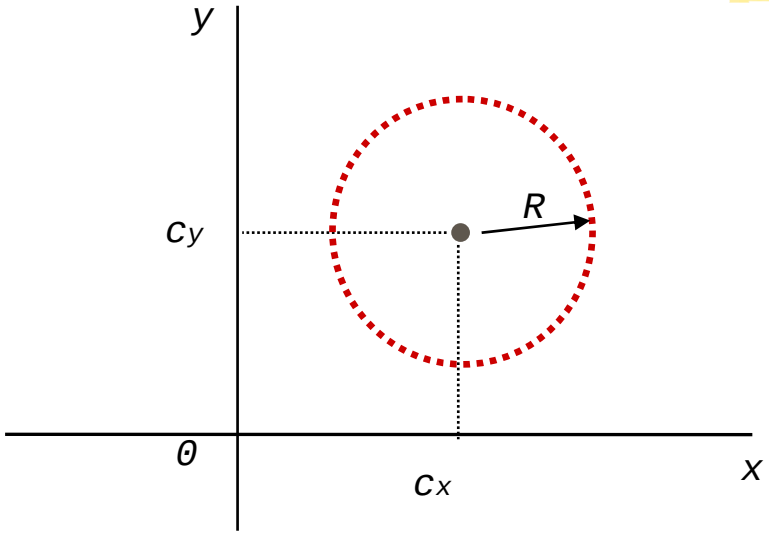
Quiz

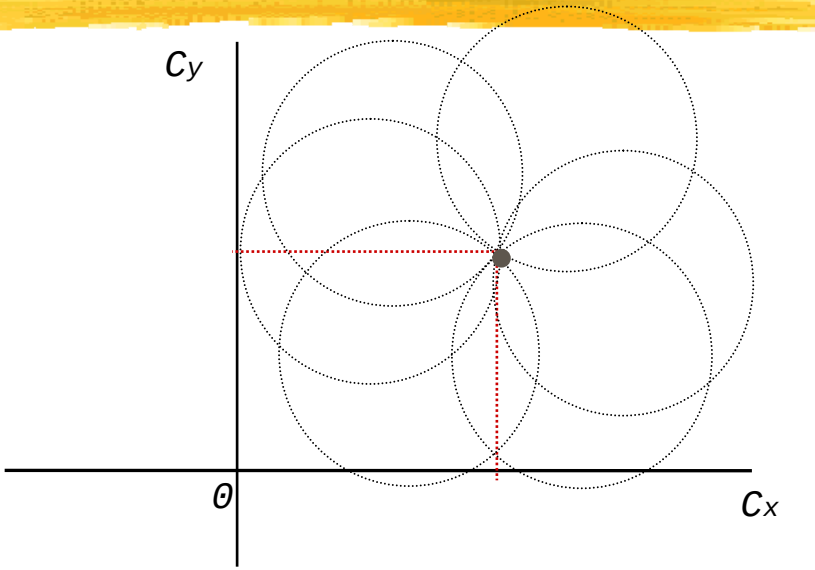
Given two points `(xo, yo)` and `(x1, y1)`. How to draw a circle of radius R that passes through these points?



$$(x-A)^2 + (y-B)^2 = R^2$$

Circle Detection





for each point (x, y)

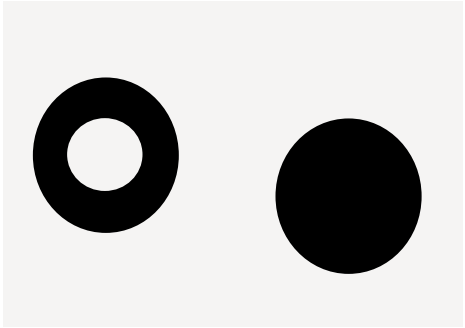
$$(x-c_x)^2 + (y-c_y)^2 = R^2 \quad \Longleftrightarrow \quad (c_x-x)^2 + (c_y-y)^2 = R^2$$

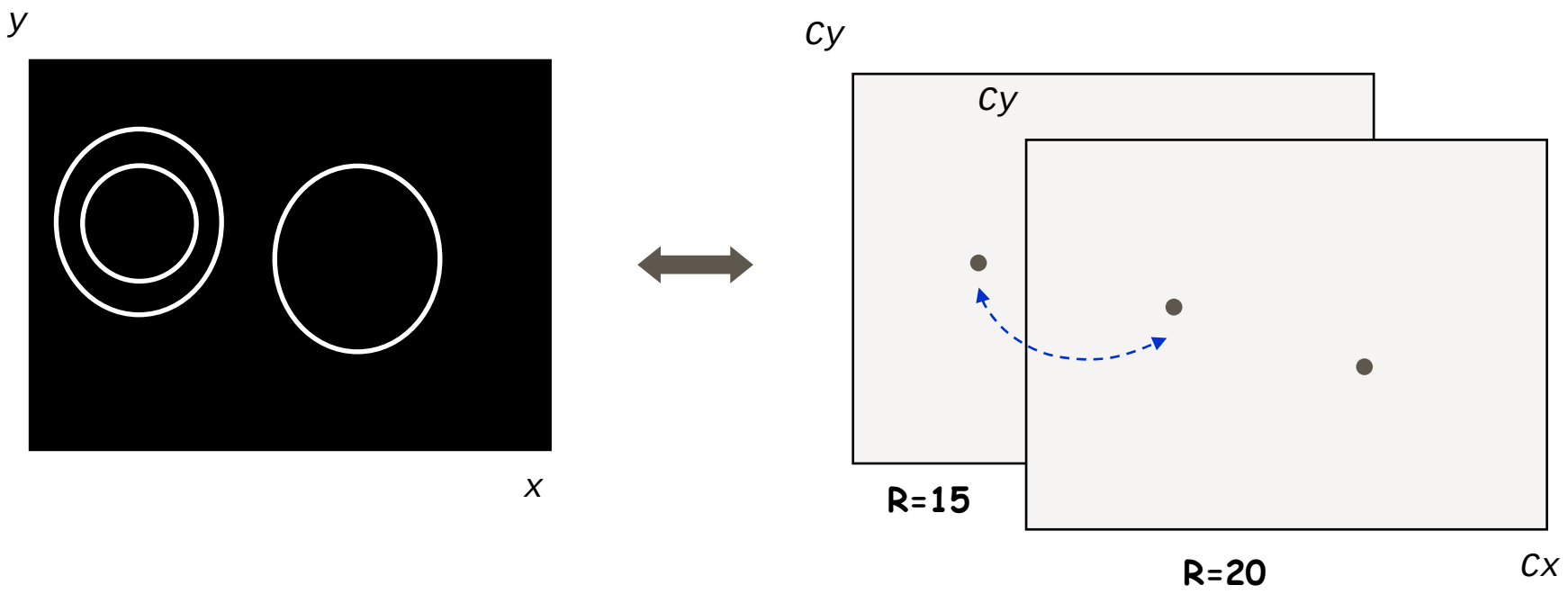
Circle Detection

An automatic assembly line must detect the presence and positions of ruber washers and gaskets

Internal radius: 15 pix

External radius: 20 pix





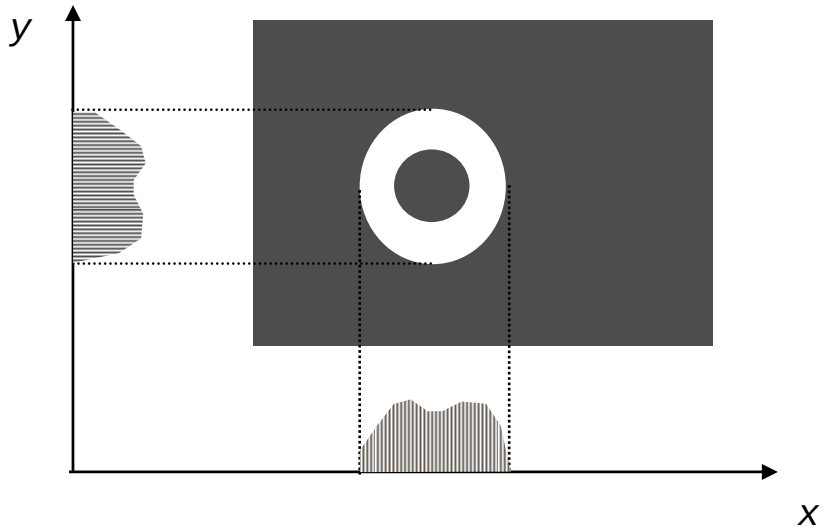
Lateral Histograms

- There is Generalised Haugh Transform that may be used for detection of arbitrary shapes. However, its computational complexity is very high
- Assuming that there is no overlap between objects, a computationally efficient method based on Lateral Histograms may be used for shape detection

An image is projected on two or more axes

Computational complexity for an N×N image is $2N^2$

If several objects are present, the histograms may be misinterpreted



Conclusions

- Hough Transform can be used to locate straight lines, circles and simple polygons
- Computational complexity and memory requirements for this algorithm are high
- Connectivity analysis may be needed to discard irrelevant points and short segments
- If ambiguities can be resolved, Literal Histogram Techniques are a low complexity alternative for shape detection and location

Suggested Reading

- D Forsyth, *Computer Vision. A Modern Approach*

  Chapter 16: Fitting

  - 16.1 The Hough Transform

- G. Bradski, A. Kaehler, *Learning OpenCV*

  Chapter 6 Image Transforms

  - Hough Transforms