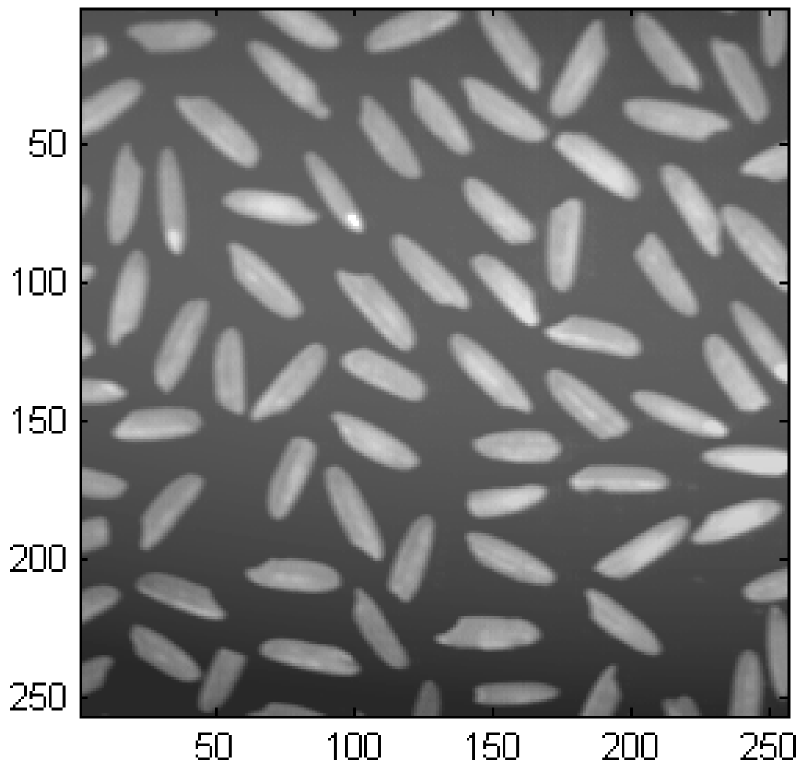
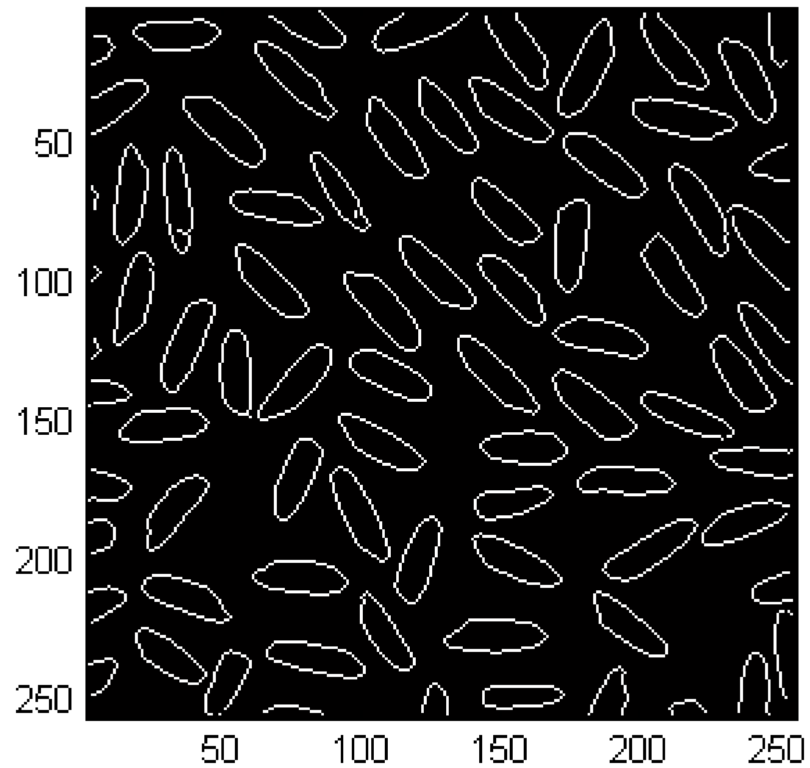
Line Detection

Using Edge Representation…

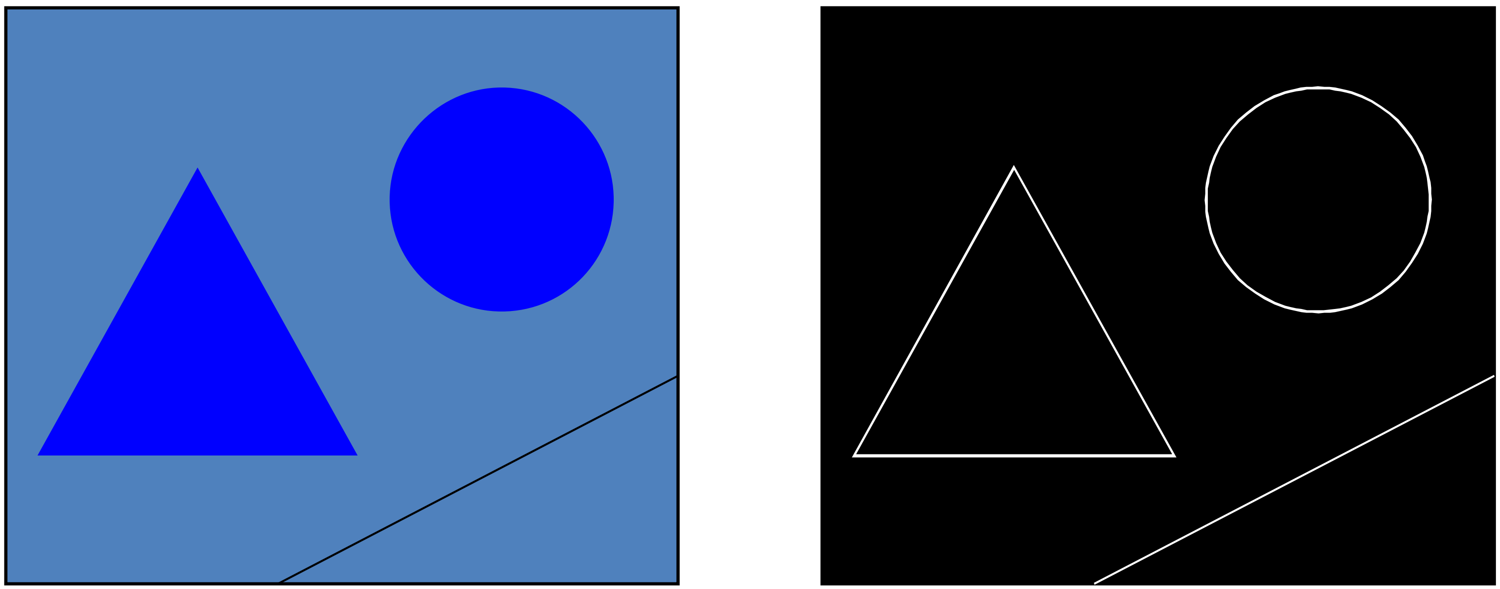
- Stereo matching problem
- Input:
  - Two images with disparity
  - Camera Calibration information
- Computation
  - Find corresponding features in two images
- Output
  - Disparity in corresponding features is related to depth
- Edges and corners help in finding correspondences

Finding Shapes from Edges





Finding Shapes from Edges



Edge Representation for Shape Analysis

- What about noisy edges?

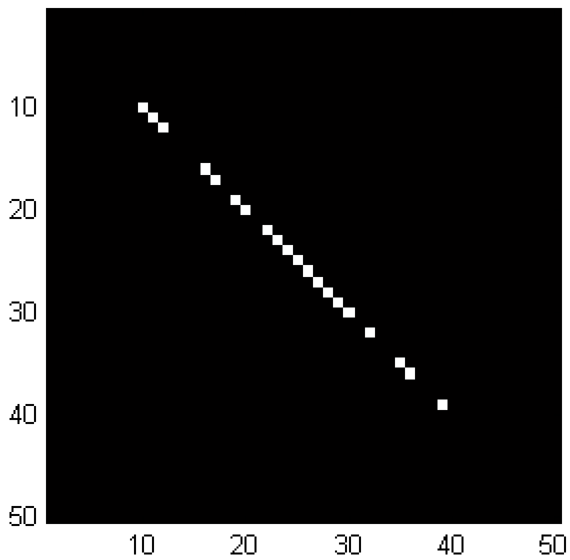
 

- Problem Def.: Find straight lines…

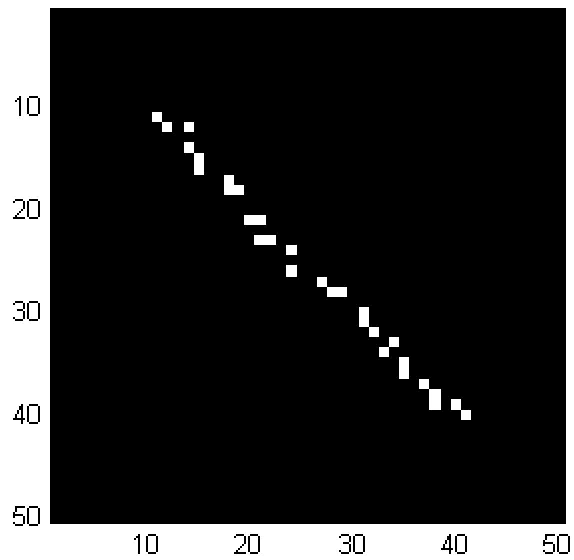
Images from: http://www.cogs.susx.ac.uk/users/davidy/teachvision/vision4.html

Problems in Finding Lines

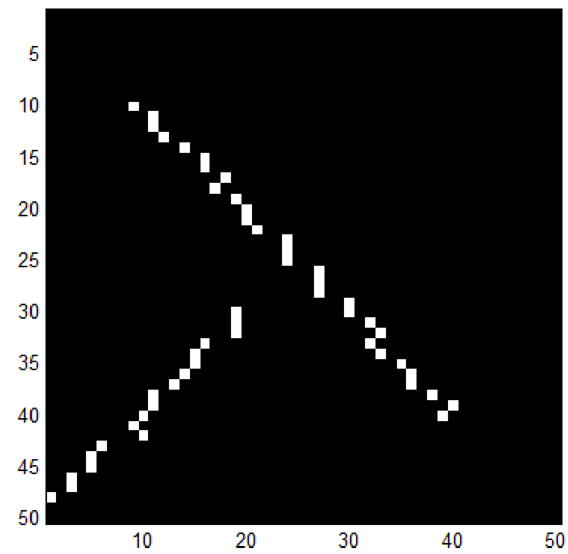
Missing Data



Noisy Data



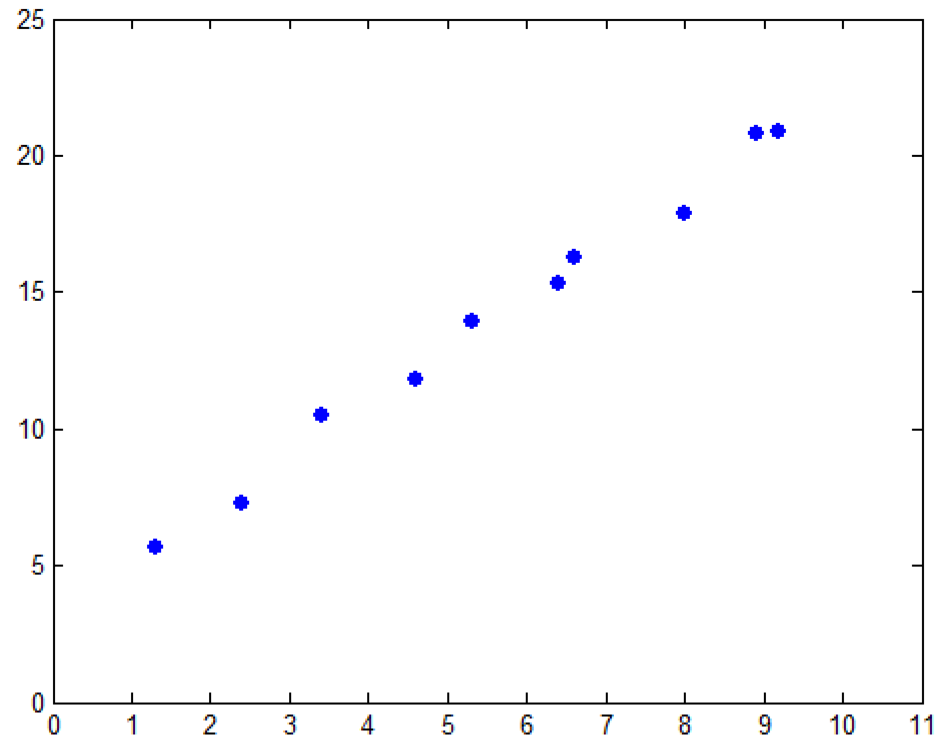
Multiple Lines



Parameter Optimization: Least Squared Error Solutions

- Fitting a line to a set of data points…

x	y
1.3	5.7
2.4	7.3
3.4	10.5
4.6	11.8
5.3	13.9
6.6	16.3
6.4	15.3
8.0	17.9
8.9	20.8
9.2	20.9



- Equation of best fit line ?

Line Fitting: Least Squared Error Solution

- Step 1: Identify the model
  - Equation of line: $y = mx + c$
- Step 2: Set up an error term which will give the goodness of every point with respect to the (unknown) model
  - Error induced by i^{th} point:
  - $$e_i = mx_i + c - y_i$$
  - Error for whole data: $E = \Sigma_i e_i^2$
  - $$E = \Sigma_i (mx_i + c - y_i)^2$$
- Step 3: Differentiate Error w.r.t. parameters, put equal to zero and solve for minimum point

Line Fitting: Least Squared Error Solution

$$E = \sum_i (mx_i + c - y_i)^2$$

$$\frac{\partial E}{\partial m} = \sum_i (mx_i + c - y_i)x_i = 0$$

$$\frac{\partial E}{\partial c} = \sum_i (mx_i + c - y_i) = 0$$

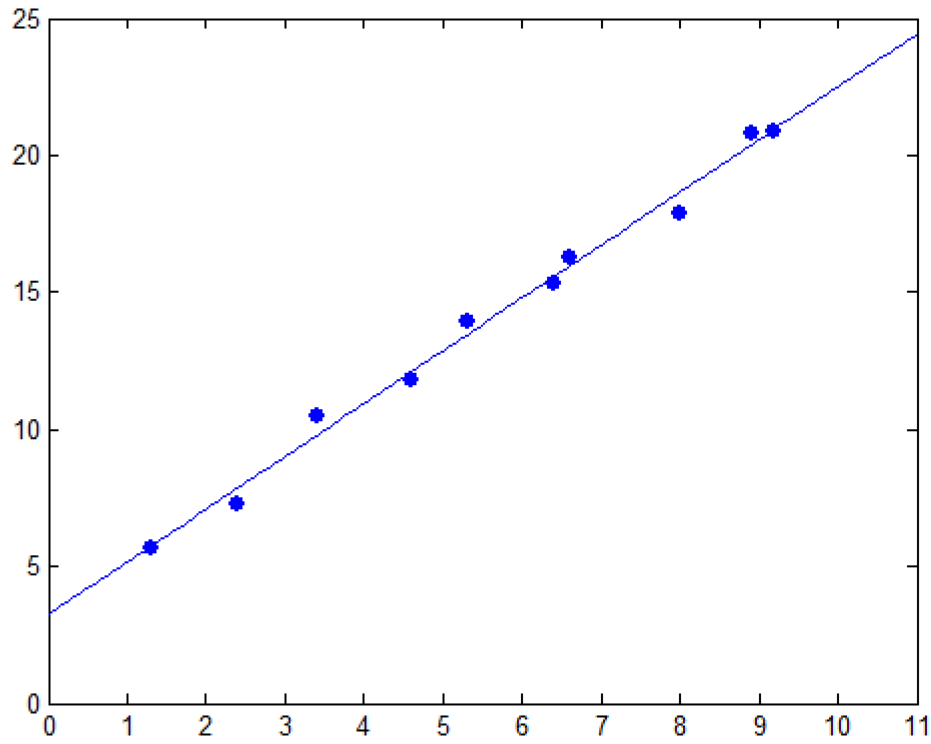
$$\begin{bmatrix} \sum_i x_i^2 & \sum_i x_i \\ \sum_i x_i & \sum_i 1 \end{bmatrix} \begin{bmatrix} m \\ c \end{bmatrix} = \begin{bmatrix} \sum_i x_i y_i \\ \sum_i y_i \end{bmatrix}$$

x	y
1.3	5.7
2.4	7.3
3.4	10.5
4.6	11.8
5.3	13.9
6.6	16.3
6.4	15.3
8.0	17.9
8.9	20.8
9.2	20.9

$$\begin{pmatrix} 380.63 & 56.1 \\ 56.1 & 10 \end{pmatrix} \begin{pmatrix} m \\ c \end{pmatrix} = \begin{pmatrix} 914.68 \\ 140.4 \end{pmatrix}$$

Solution: m = 1.9274 c = 3.227

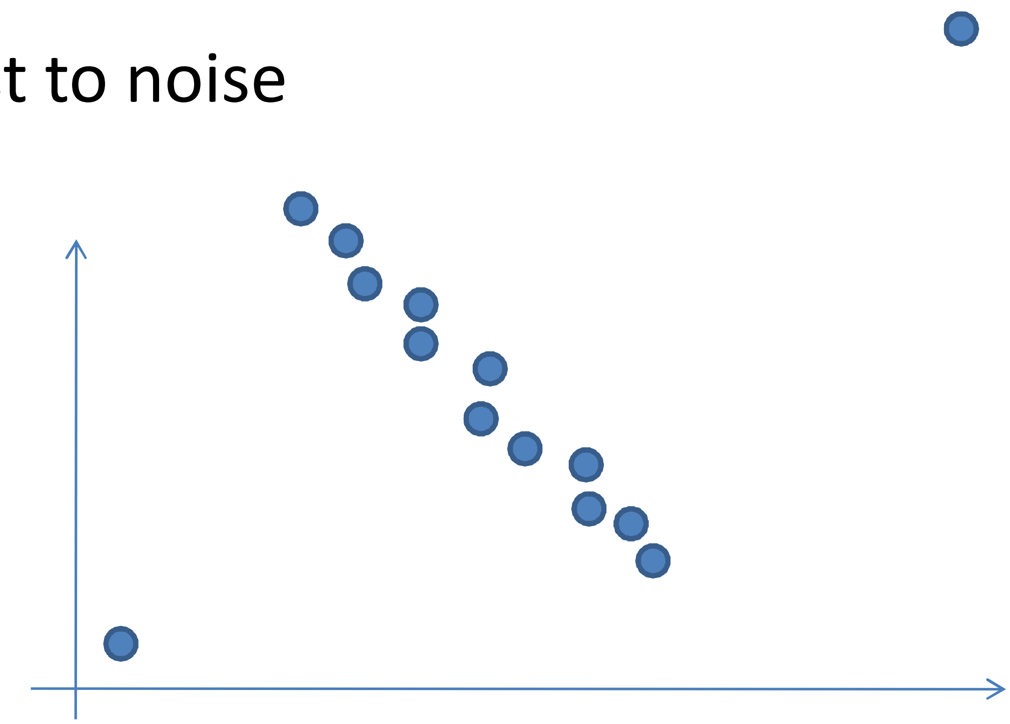
Line Fitting: Least Squared Error Solution



Least Squared Error Solution

- Disadvantages?
  - Multiple Lines…
  - Not robust to noise

- Example



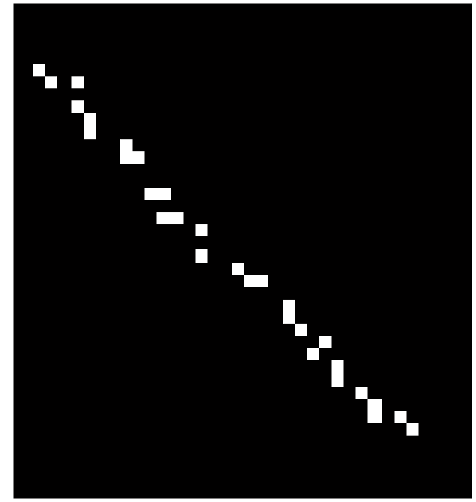
Finding Lines

- Problem Definition:
- Given a binary image, find all *significant* lines
- Line: $y = \boldsymbol{m}x + \boldsymbol{c}$
- Estimate m,c parameters of all significant lines in presence of noise

Hough Transform

- Method to find any type of shape that can be represented in **parametric** form
- E.g. lines, circles, parabolas, ellipses…
- Generalized Hough Transform
  - For arbitrary shapes

Hough Transform for Lines

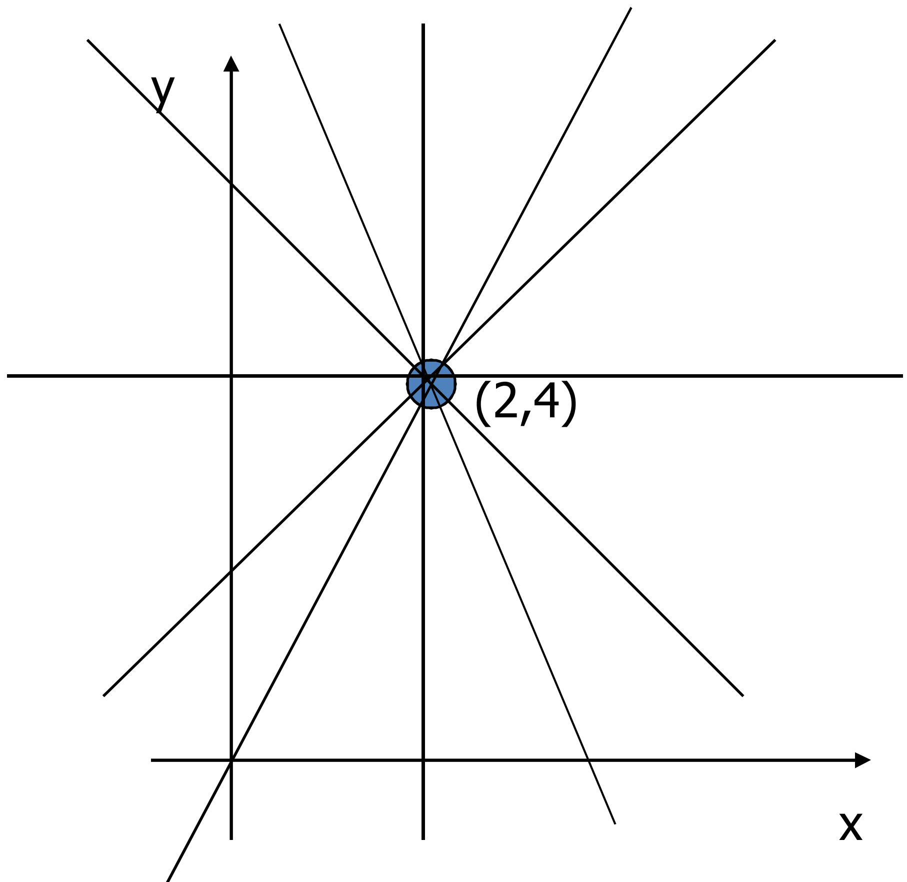


- General Idea:
  - Search for the best possible **m** and **c** parameters given the data
- Consider all possible lines in the image
- What is the most significant line?
- **A line passing through most of the points**
- A line that passes through 1 point gets one vote
- Find the line that gets most votes

Hough Transform for Lines

- **Aim**: Create a mechanism for **voting**
  - **A line should get as many votes as the points it passes through**
- Equation of line is **y=mx+c**
  - **m** is slope, **c** is intercept
- Consider only one point **(x,y)**
  - For example (2,4)
- How many lines can pass through this point?

Hough Transform for Lines



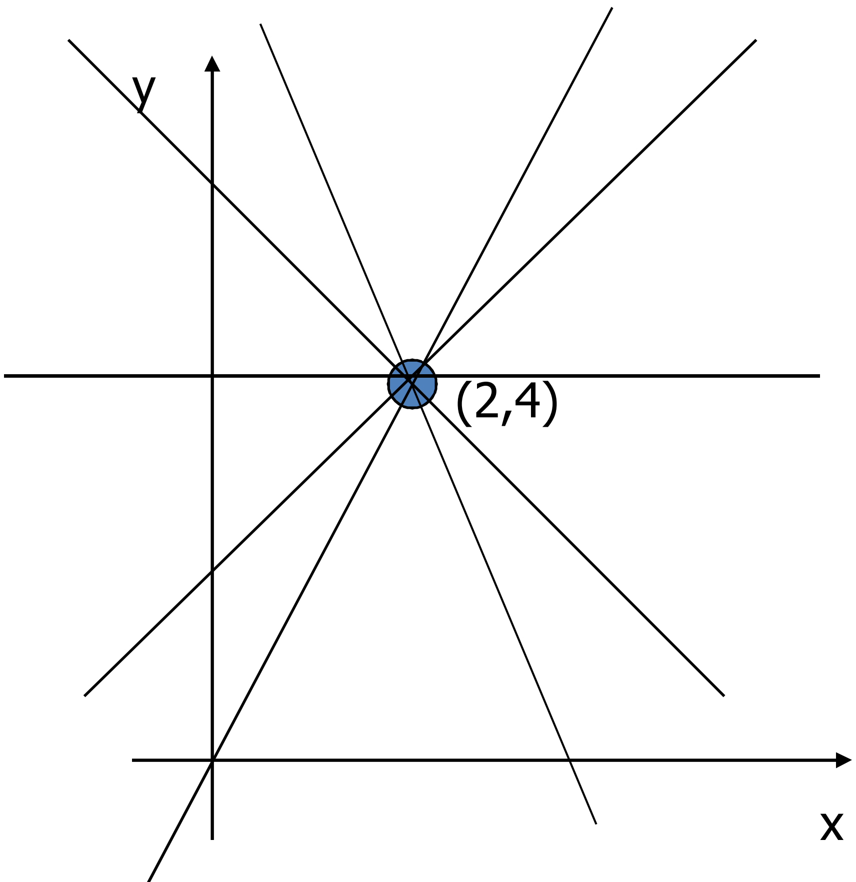
$x=2$
$y=4$
$y=x+2$
$y=-x+6$
$y=2x$
$y=-2x+8$

And so on… (infinite lines)

Hough Transform for Lines

- Can we write the general expression for all the lines passing through (2,4)?
- All those lines will have a specific relationship between **m** and **c**
- Any arbitrary combination of **m** and **c** will not pass through the given point; only certain combinations will work

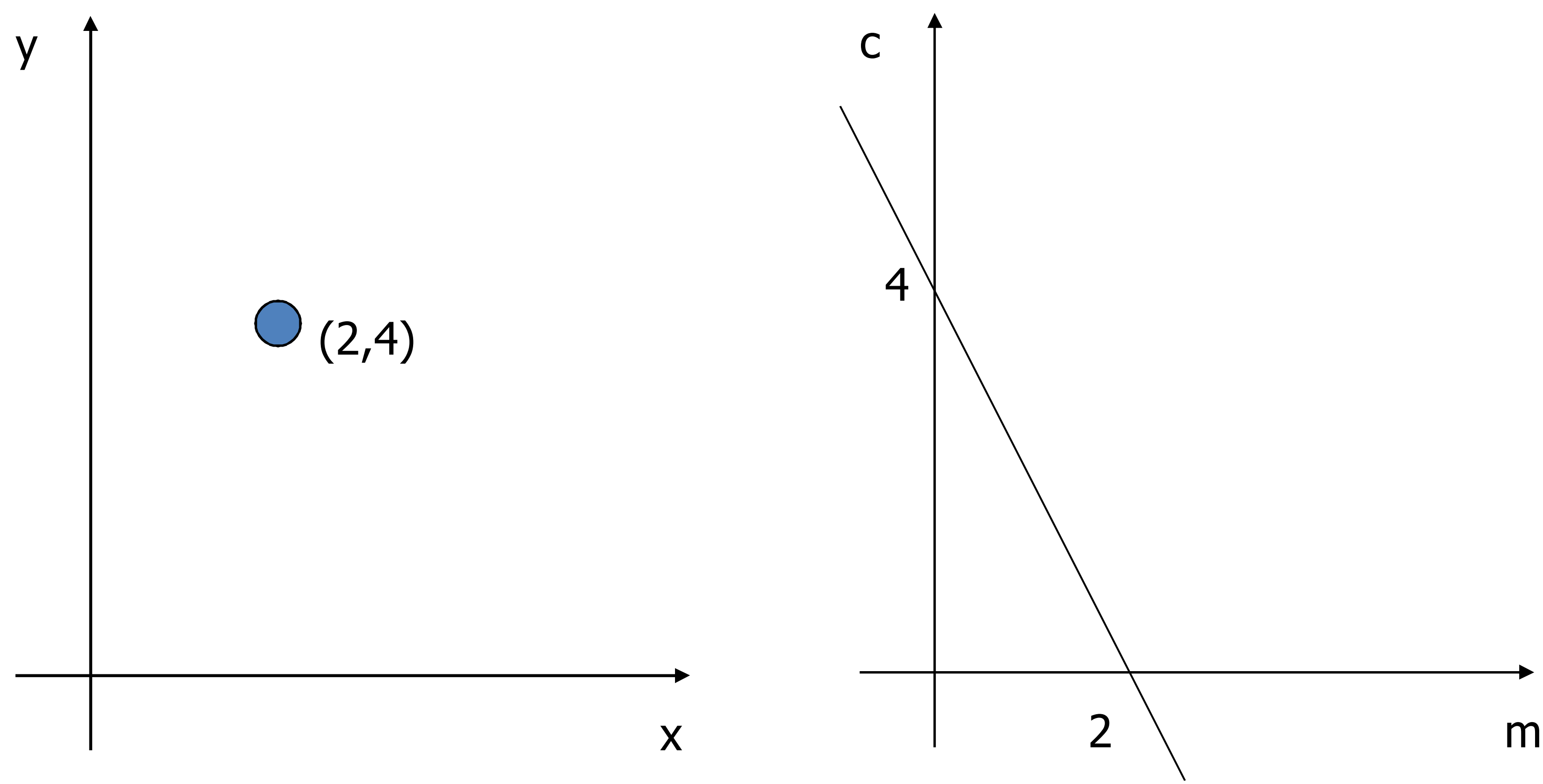
Hough Transform for Lines



y=4	m=0, c = 4
y=x+2	m=1, c = 2
y=-x+6	m=-1, c = 6
y=2x	m=2, c = 0
y=-2x+8	m=-2, c = 8

Plot the m, c points in m-c space

Hough Transform for Lines



Hough Transform for Lines

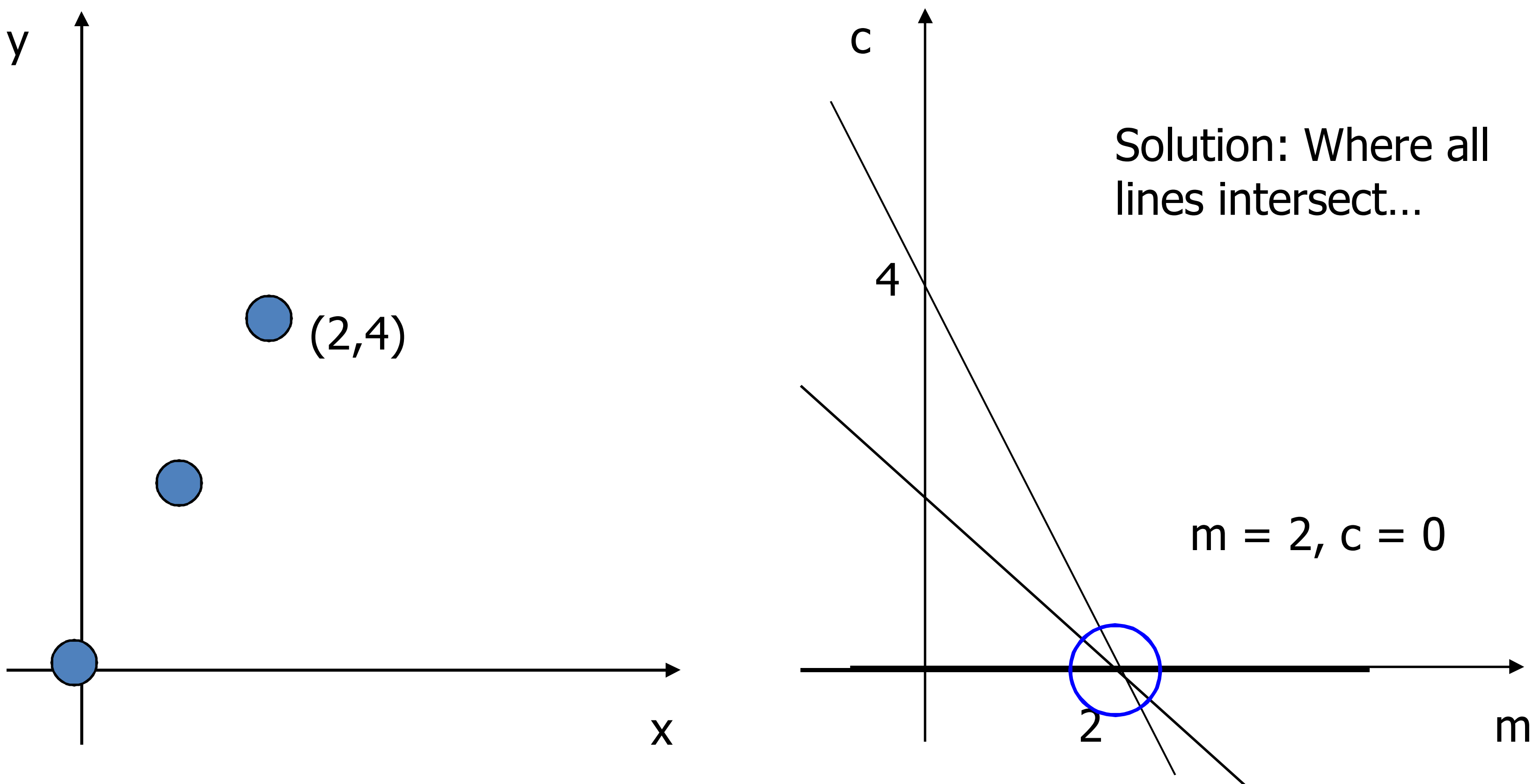
- Equation of line is **y=mx+c**
- We are given **(x,y)** [e.g. (2,4)]
- **(m,c)** are the unknowns
- Can be rewritten as **c = (-x)m +** y
- Consider **(x,y)** space: **y=mx+c** represents a line
- Consider transformed space **(m,c)**, then **c=(-x)m + y** is a line in this space
- **(-x)** is **gradient**, **y** is the **intercept**

Interpretation

- Line in **(m,c)** space represents all possible lines that could pass through a **single point (x,y)**
- **Point** in **(x,y)** space is a **line** in **(m,c)** space
- **Point** in **(m,c)** space is a …
- **Line** in **(x,y)** space

- **Aim: In x-y space, find a line passing through most of the points**
- i.e., What is the equal statement in (m-c) space?
- In m-c space, find a point from which most of the lines are passing through? (i.e., the point in which most of the lines interest)

Finding Lines using Hough Transform



Hough Transform for Lines

- Initialize **Accumulator** array, **A**, of two dimensions
  (m, c)
- For each point **(x,y)** in image, increment cells along line
  c = -xm+y by 1
- Find **maximum** point in accumulator array for solution

Algorithm

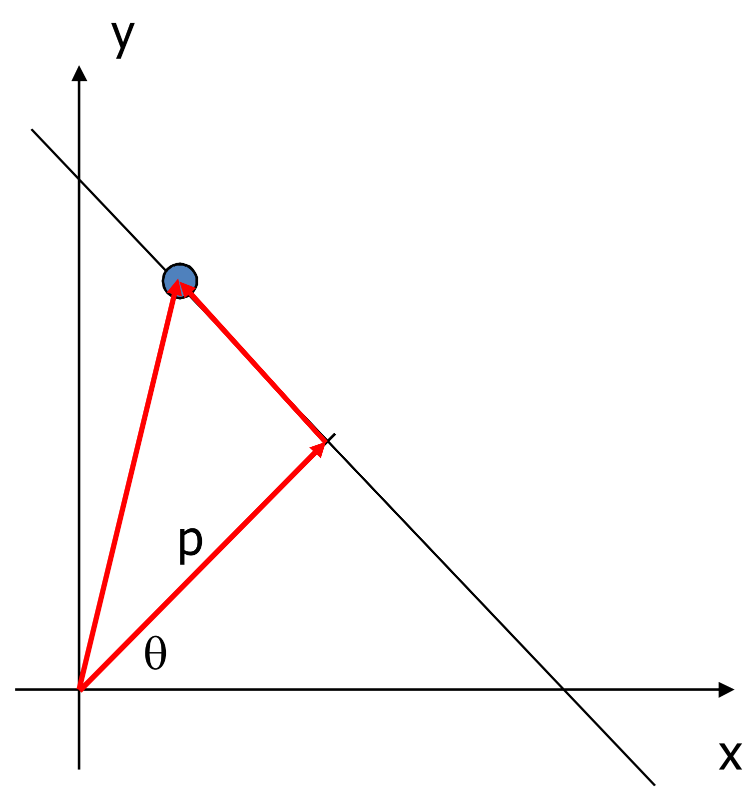
1. Quantize parameter space
   $\mathbf{A}[c_{min}, \ldots , c_{max}, m_{min}, \ldots , m_{max}]$
2. For each edge point (x,y)
   For ($m = m_{min}, m \leq m_{max}, m$++)
      $c = (-x)m + y$
      $\mathbf{A}[c,m] = \mathbf{A}[c,m] + 1$;
3. Find local maxima in **A**

Hough Transform for Lines

- Problems with this procedure?
- What about the range of slope?
- m spans -∞ to ∞
- Solution?
- Use alternate parameterization of line

Alternate Line parameterization

- $p = x \cos\theta + y \sin\theta$
- p is the perpendicular to the line
- θ is the angle p makes with the x-axis

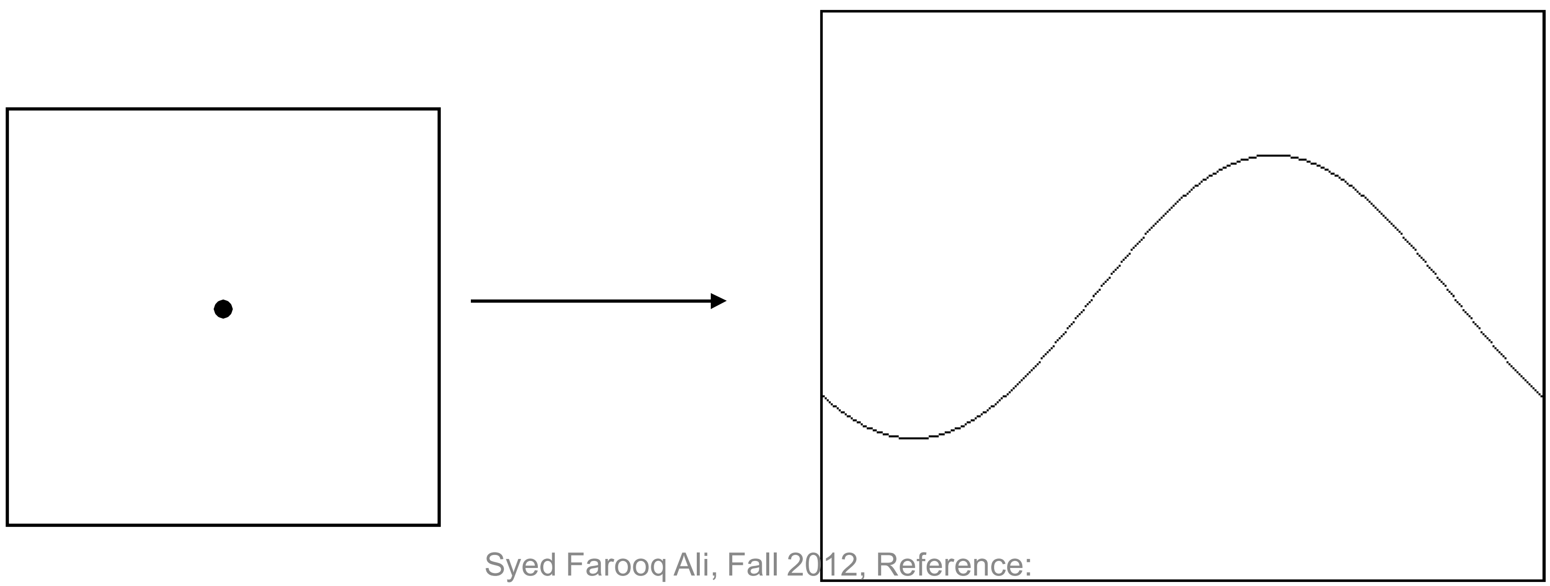


Algorithm (polar form)

1. Quantize parameter space
   A [θ_{min}, … , θ_{max}, p_{min}, … , p_{max}]
2. For each edge point (x,y)
   For ($\theta = \theta_{min}$, $\theta \leq \theta_{max}$, θ++)
      $p = x \cos\theta + y \sin\theta$
      A [θ, p] = **A** [θ, p] + 1;
3. Find local maxima in **A**

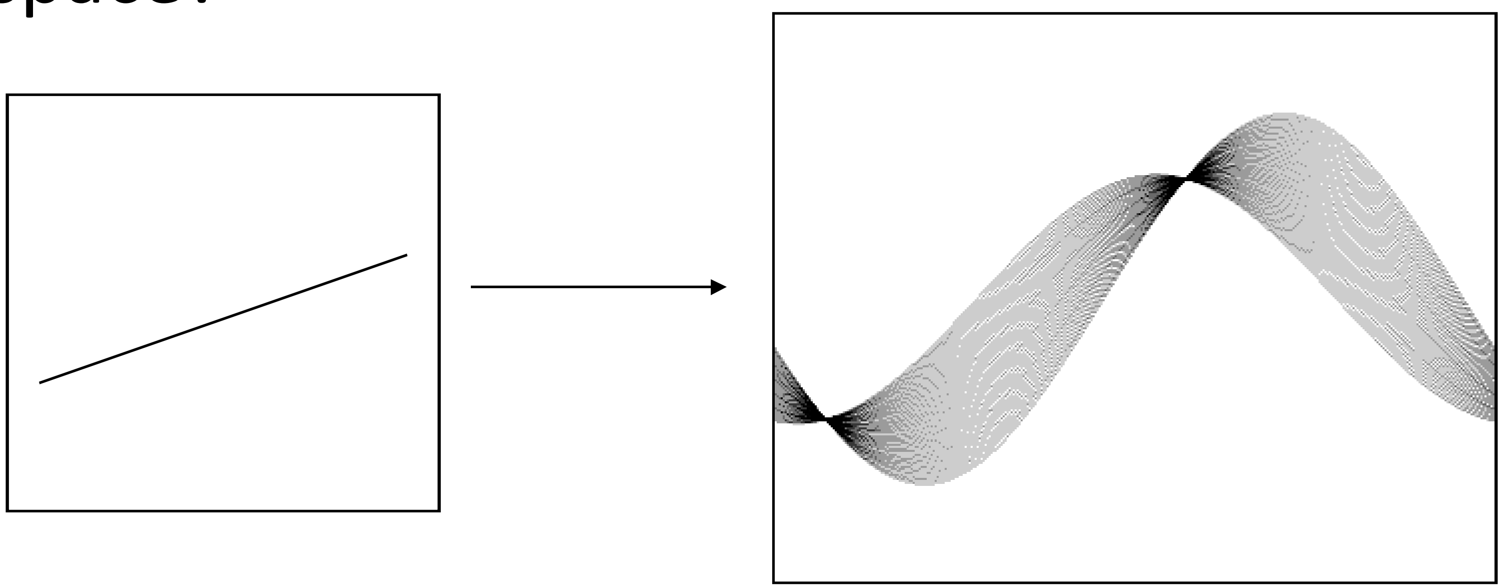
HT for Lines (polar form)

- Point is **(x,y)** space represents _______ in the parameter space **(p, θ)**?
- Answer: Sinusoid curve

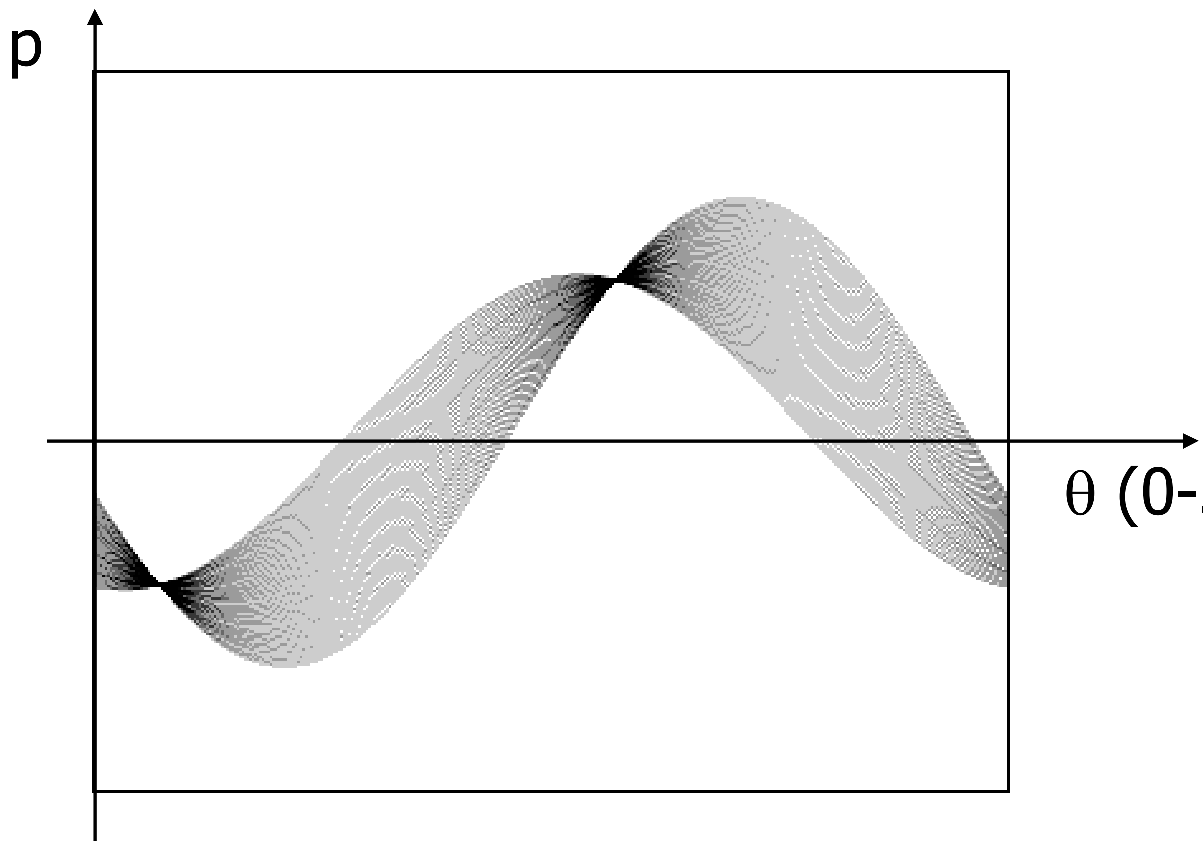


HT for Lines (polar form)

- Line in **(x,y)** space represents ________ in **(p, θ)** space?



HT for Lines (polar form)



Each line appears as two peaks

Additional advantage of Polar Form

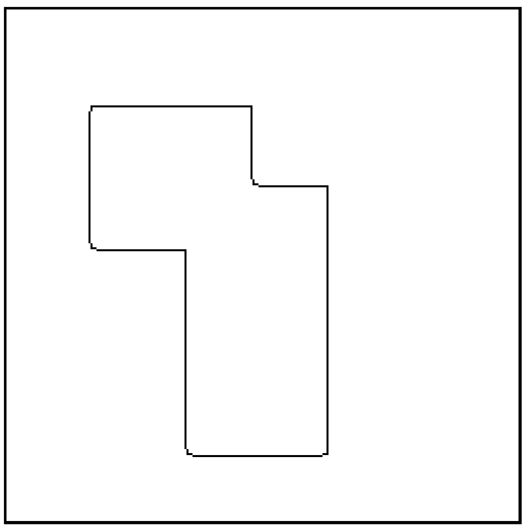
- Line which passes through **(x, y)** was assumed to have all possible values of θ
- Gradient direction?
- θ can be computed from **gradient direction**

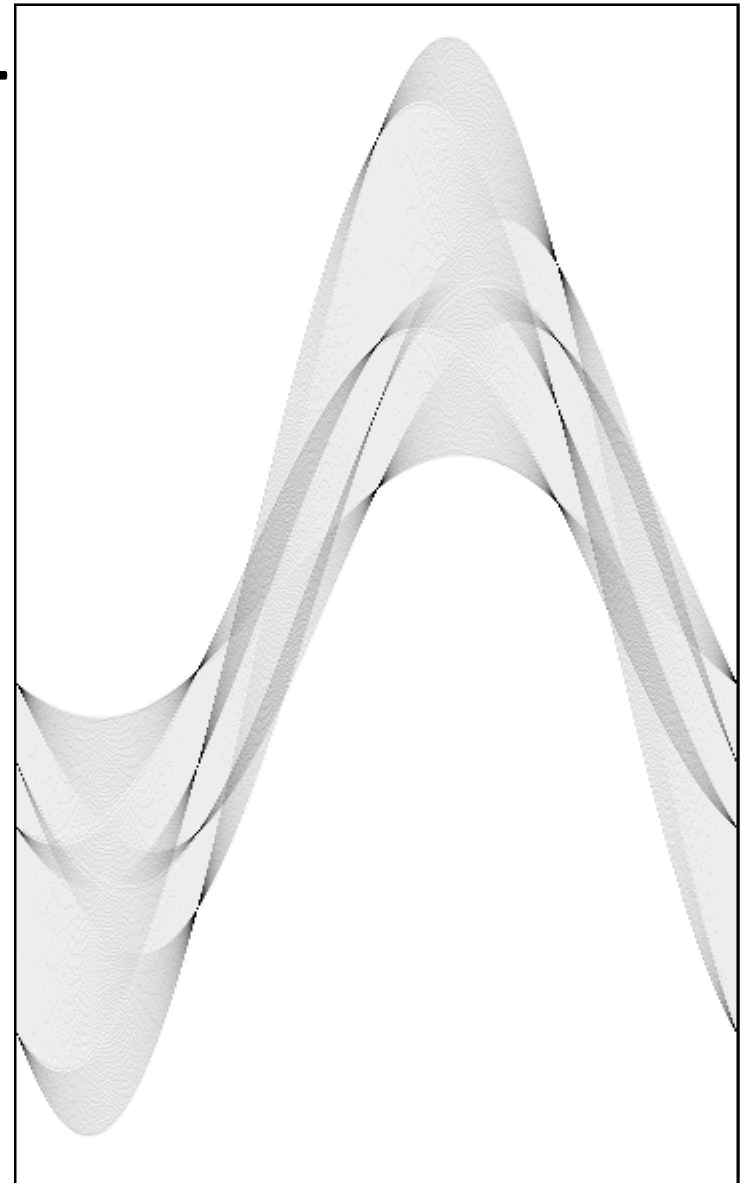
Algorithm (polar form/improved)

1. Quantize parameter space
   A [θ_{min}, … , θ_{max}, p_{min}, … , p_{max}]
2. For each edge point (x,y)
   Compute θ from gradient direction
   $p = x \cos\theta + y \sin\theta$
   A [θ, p] = **A** [θ, p] + 1;
3. Find local maxima in **A**

HT for L

- What about multiple lines in an image?

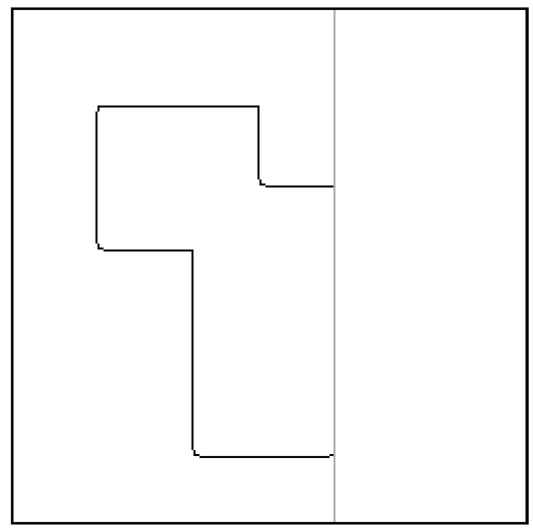




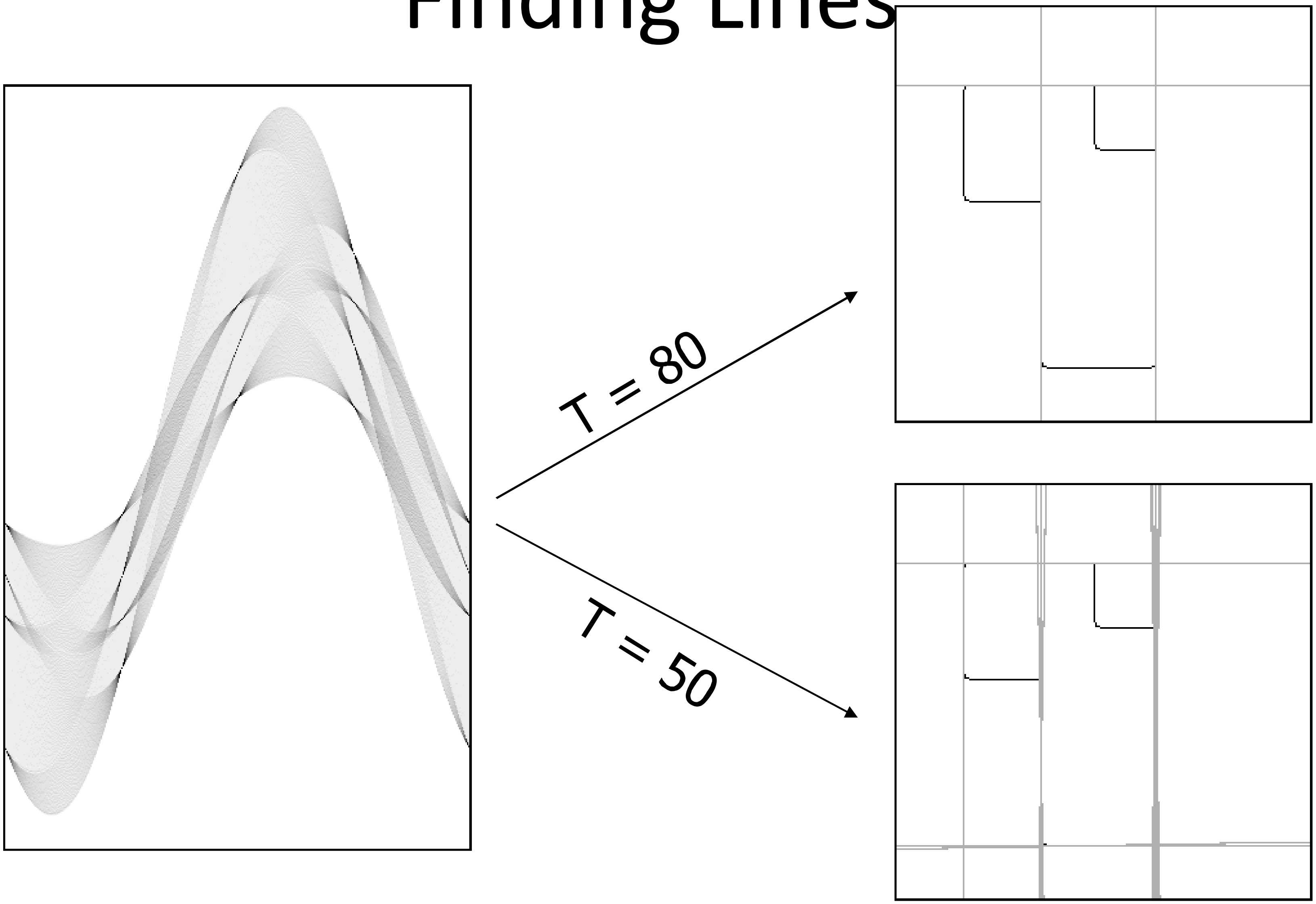
Finding Lines

- Detect peaks in the accumulator array
- Threshold – or more complicated peak finding function

T = 120



Finding Lines



Hough Transform for Circles

- Equation…
- Centered at (x_0, y_0) with radius r

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

- Three unknowns… x_0 y_0 r
- Three dimensional parameter space
- Conceptually… ?

How to simplify this algorithm

- Use of gradient direction, θ

$$x_0 = x - r\cos\theta$$
$$y_0 = y - r\sin\theta$$

- Algorithm?...

Hough transform

- Can be applied to any parametric representation $f(\mathbf{x}, \mathbf{a})=0$
- Initialize accumulator array, A to zeros…
- A is |a| dimensional
- For each pixel **x**, and each **a** such that $f(\mathbf{x},\mathbf{a})=0$, A[**a**] = A[**a**]+1
- Local maxima of A corresponds to curves *f* in image.

Finding more than one curve

- Parameter space will have multiple maxima
- Threshold
- Or use better methods to find maximum points

Hough Transform

- Given parametric representation of a curve
  - LINE: $p = x \cos\theta + y \sin\theta$
  - CIRCLE: $x_0 = x - r \cos\theta$
    $y_0 = y - r \sin\theta$
  - ELLIPSE: $x_0 = x - a \cos\theta$
    $y_0 = y - b \sin\theta$
  - GENERAL: $f(\mathbf{x}, \mathbf{a}) = 0$

Hough Transform

- Initialize **A** (accumulator array) to all zeros
- **A** is |**a**| dimensional
- For each pixel **x**, and each **a** such that f(**x,a**)=0, A[**a**] = A[**a**]+1
- Local maxima of A corresponds to curves *f* in image.

Generalized Hough Transform

To find arbitrary shapes in images

Shapes which do not have an *easy* parametric representation

Centroid and Area

- The average location of all pixels in a region *R*

$$\bar{r} = \frac{1}{A} \sum_{(r,c)\in R} r$$

$$\bar{c} = \frac{1}{A} \sum_{(r,c)\in R} c$$

- where A is the area of region *R*

$$A = \sum_{(r,c)\in R} 1$$

Example

	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	1	1	1	1	0	0	0	0
3	0	0	0	1	0	0	0	0	1	0	0	0
4	0	0	1	0	0	0	0	0	0	1	0	0
5	0	0	1	0	0	0	0	0	0	1	0	0
6	0	0	0	1	0	0	0	0	1	0	0	0
7	0	0	0	0	1	1	1	1	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0

A = 16

r = 1/16 * (2+2+2+2+3+4+5+6 +7+7+7+7+6+5+4+3)

= 4.5

c = 1/16 * (5+6+7+8+9+10+10+ 9+8+7+6+5+4+3+3+4)

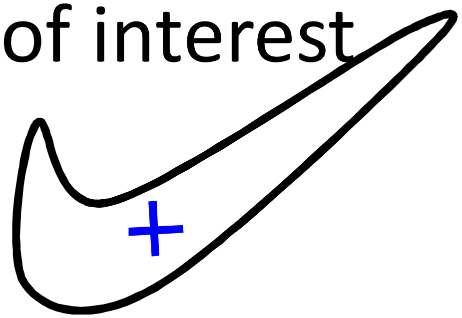
= 6.5

Generalized Hough Transform

- Training
  - A representation of shape of interest is built in the form of an R-Table
- Detection
  - Using R-Table, a given shape is matched to the shape of interest

GHT - Training

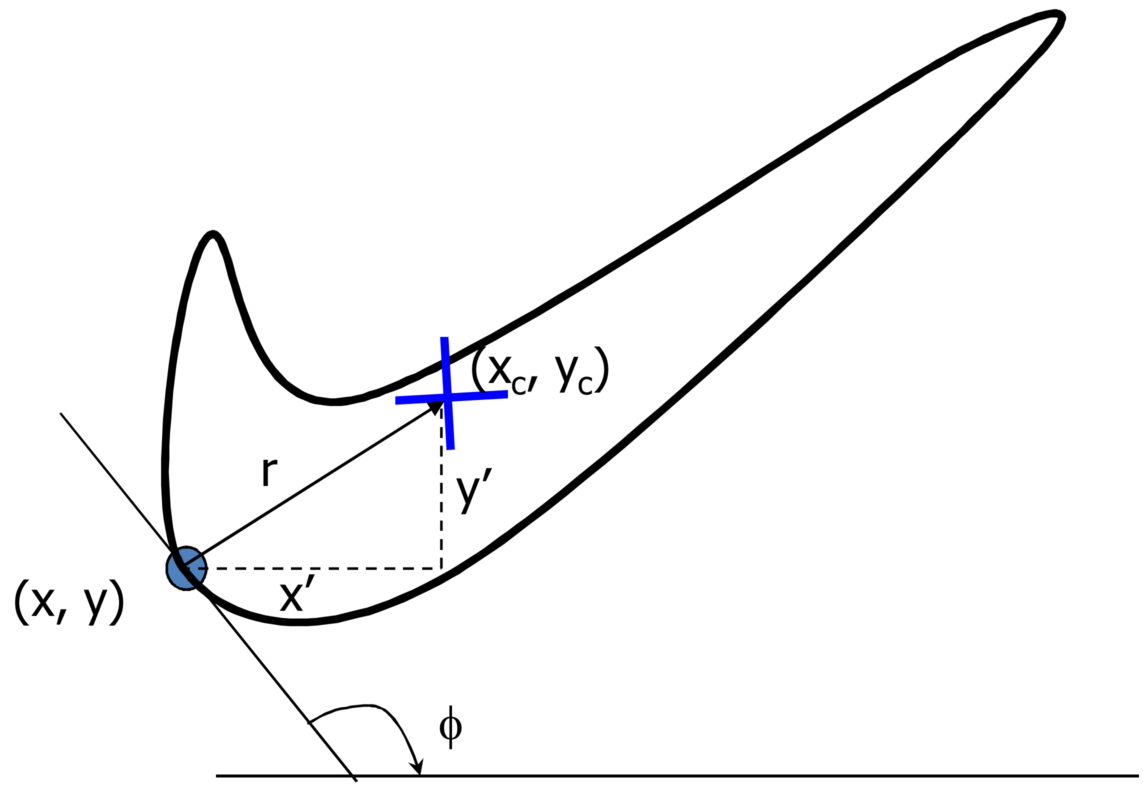
- Given the shape of interest



- Find Centroid (x_c, y_c) of shape

GHT - Training

- Find r = (x', y') for each edge point
- $x_c = x + x'$
- $y_c = y + y'$



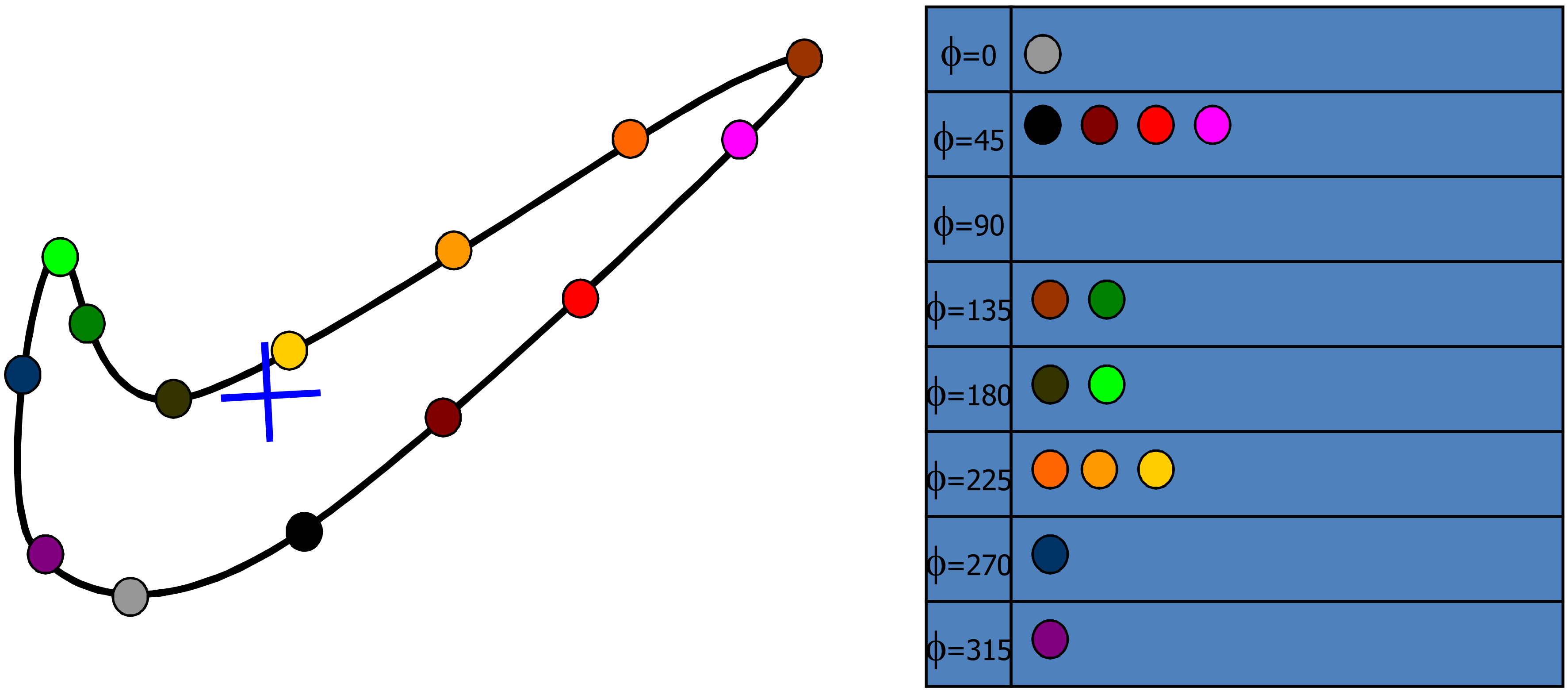
- ϕ is the angle tangent at (x,y) makes with x-axis

GHT - Training

- R-Table is indexed by ϕ

ϕ_1	$r_1^1, r_2^1, r_3^1, \ldots, r_{m_1}^1$
ϕ_2	$r_1^2, r_2^2, r_3^2, \ldots, r_{m_2}^2$
ϕ_3	$r_1^3, r_2^3, r_3^3, \ldots, r_{m_3}^3$
$\bullet$	$\bullet$
$\bullet$ $\bullet$	$\bullet$ $\bullet$
ϕ_n	$r_1^n, r_2^n, r_3^n, \ldots, r_{m_n}^n$

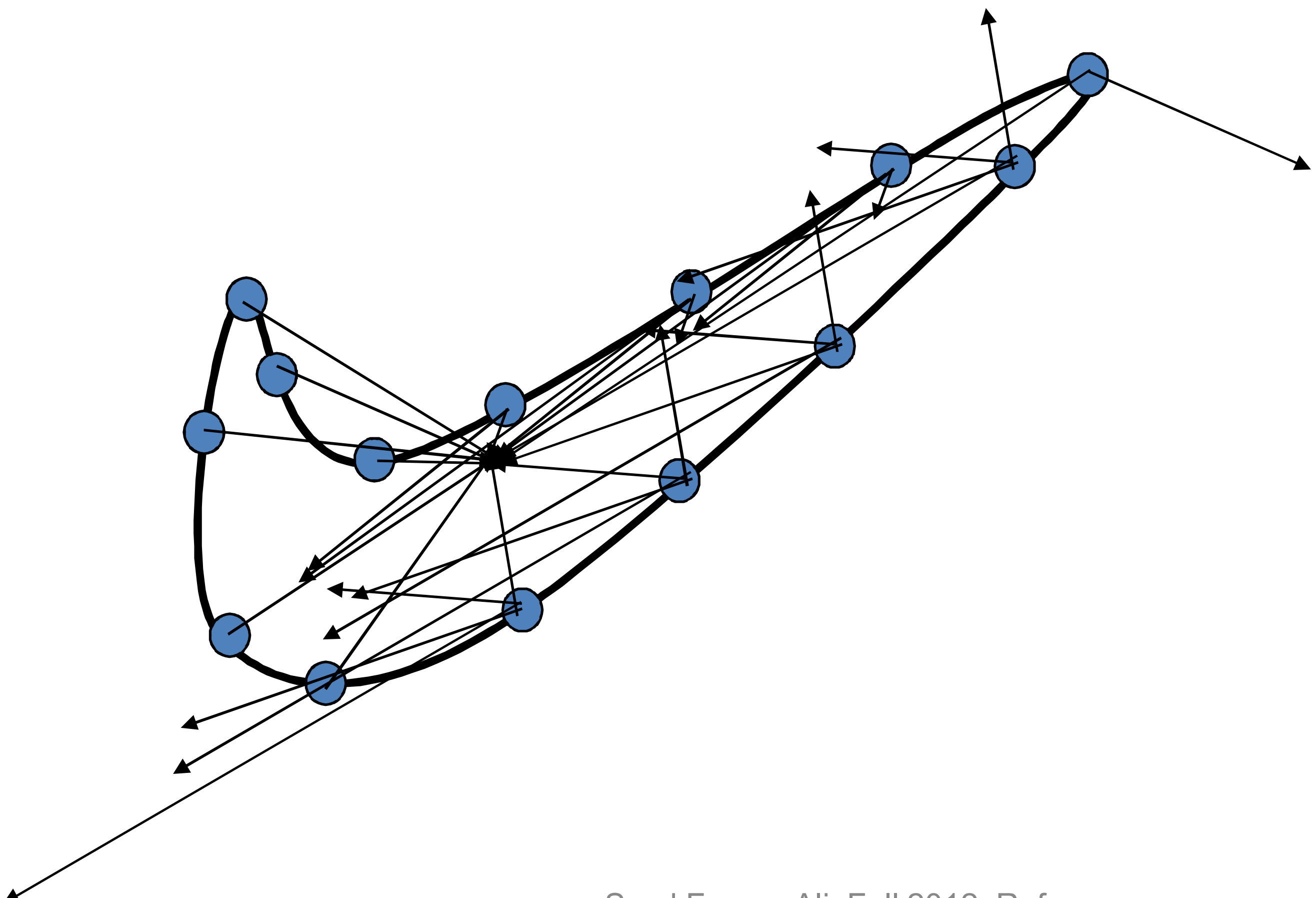
Example - Training

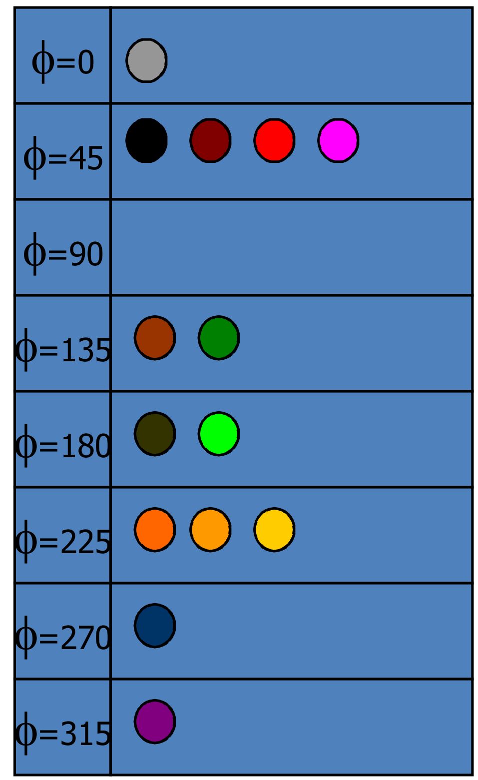


Detection

- Go to each (x,y) in image
- Find φ
- For corresponding entry in R Table
- Find all possible locations of centriods
- xc = x + x’
- yc = y + y’
- Increment centroid accumulator by 1

Detection





GHT - Algorithm

- Quantize parameter space

  $P\ [x_{cmin}, \ldots, x_{cmax}, y_{cmin}, \ldots, y_{cmax}]$

- For each edge point (x,y)
  - Compute φ from gradient direction
  - For each table entry in row φ

    xc = x + x'

    yc = y + y'

  - P [xc, yc] = P [xc, yc] + 1;
- Find local maxima in P

GHT - Questions

- Uniqueness of R-Table?
- Invariance to translation?
- Invariance to rotation?
- Invariance to scaling?

Rotation and Scaling Invariance (Not in current course)

- Rotation invariance…

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

  $y'' = -x' \sin\theta + y' \cos\theta$

- Rotation + Scaling invariance

  $x'' = s_x (x' \cos\theta + y' \sin\theta)$

  $y'' = s_y (-x' \sin\theta + y' \cos\theta)$

- Substitute in

  $x_c = x + x'$

  $y_c = y + y'$

- To get

  $x_c = x + s_x (x' \cos\theta + y' \sin\theta)$

  $y_c = y + s_y (-x' \sin\theta + y' \cos\theta)$

- Substitute these values of (x_c, y_c) in GHT algorithm