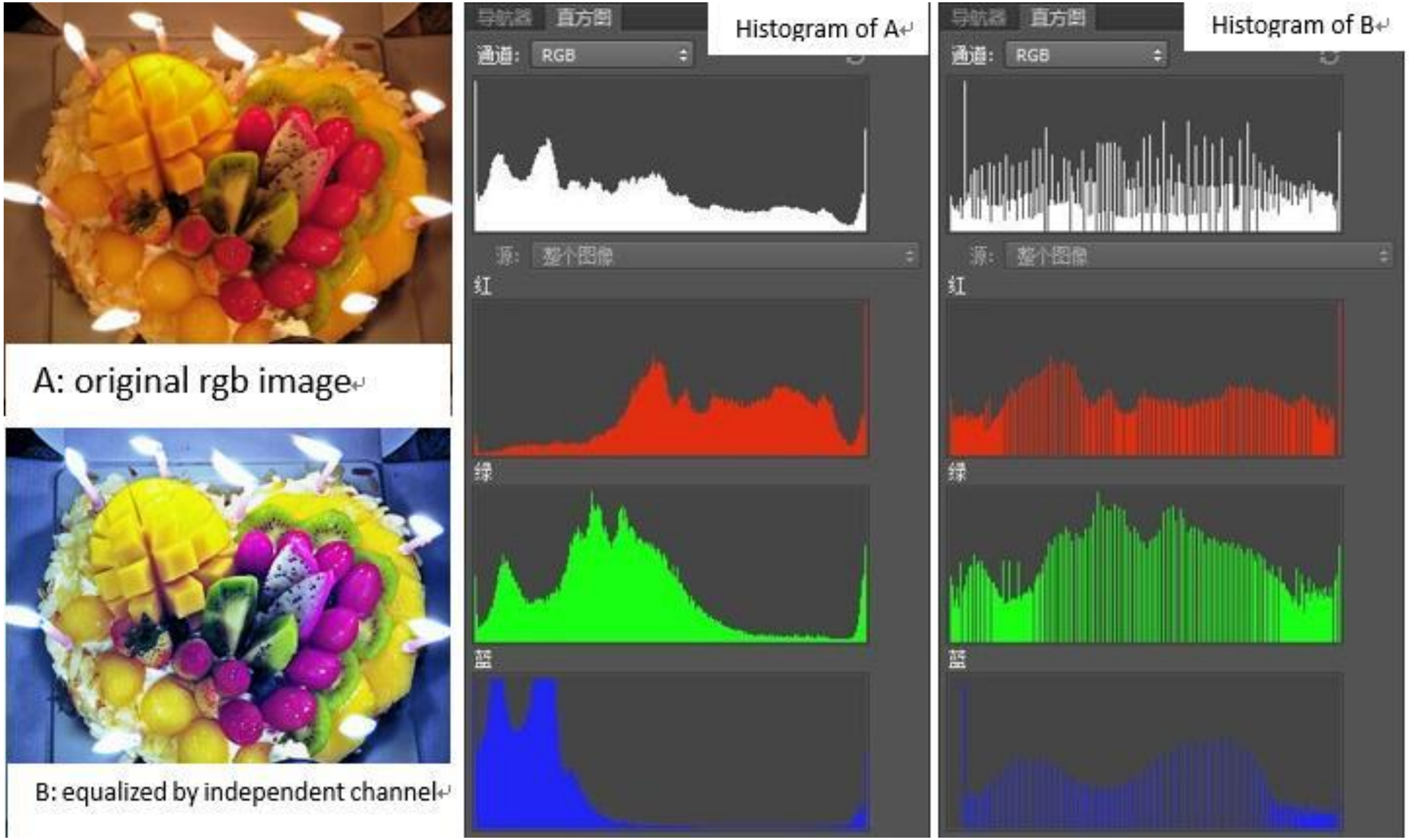
- **E.g. Contrast enhancement in color images:**

- **Basic (popular) approach:**
  - ⇒ human eye - 5 more sensitive to brightness contrast then color contrast
  - ⇒ can achieve good contrast enhancement on brightness component alone!
  - ⇒ typically:

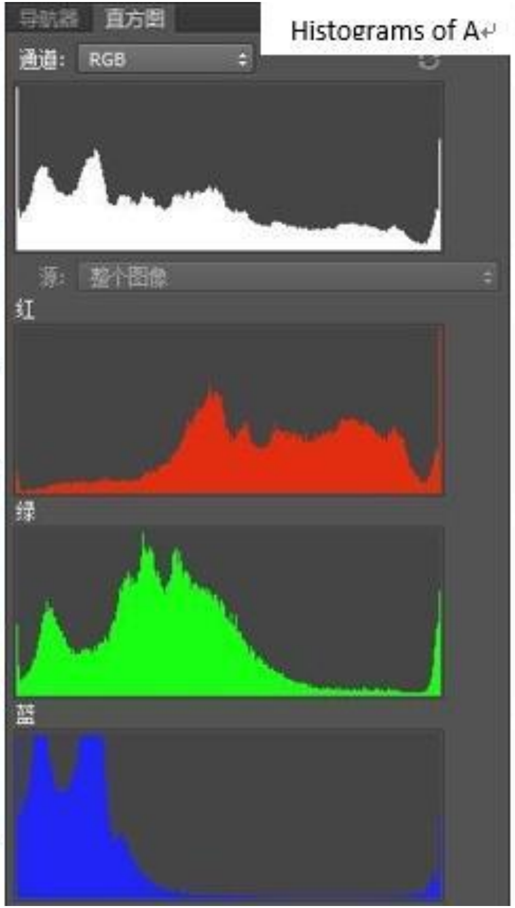
Histogram equalization (per-channel)



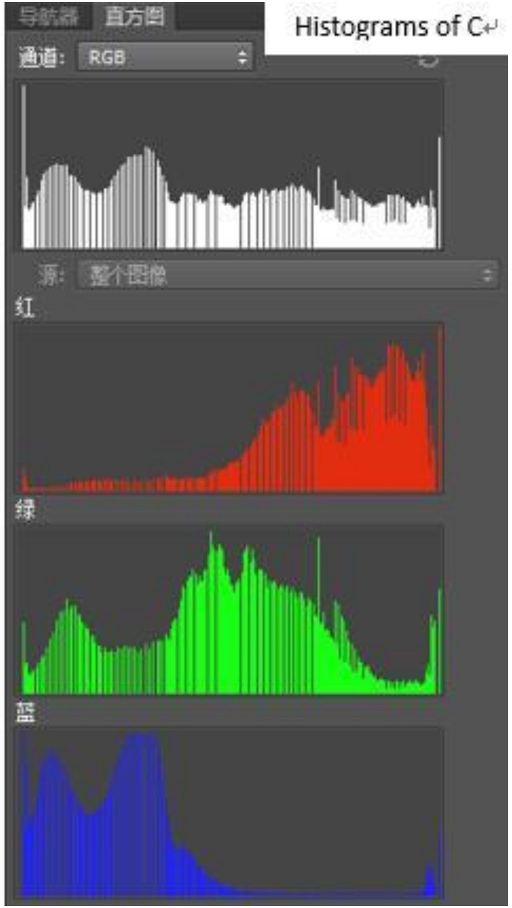
Histogram equalization (ref histogram = average of R,G,B)

A: original rgb image

C: equalized by average histogram

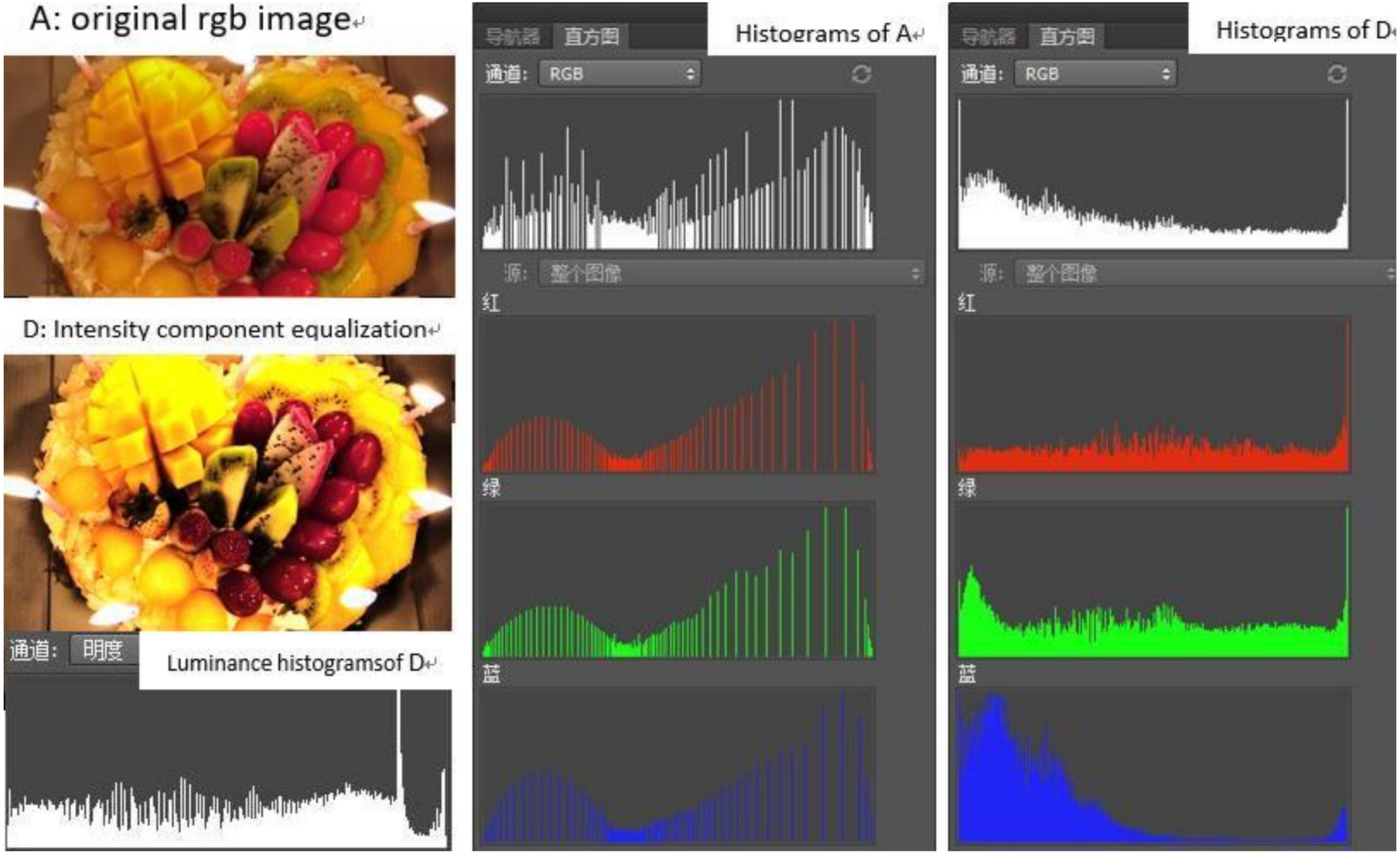


Histograms of A



Histograms of C

Histogram equalization (ref histogram = I of HSI)



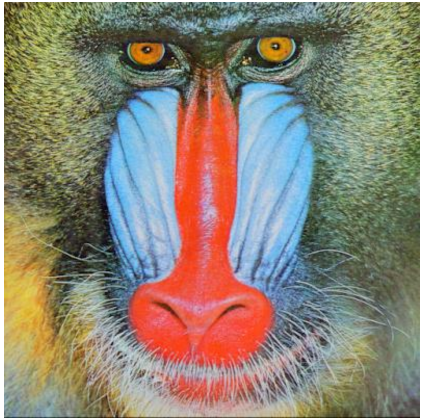
Grayscale quantization

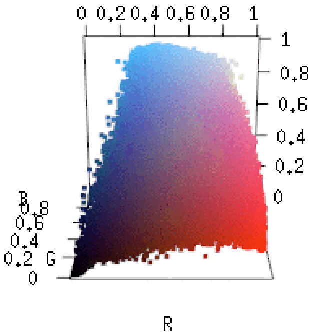
Original
(256 colors)

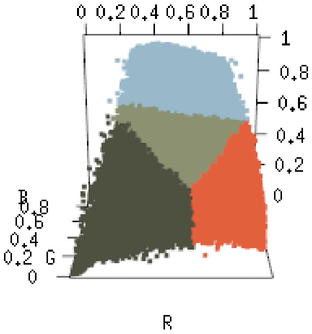
8 colors

4 colors

Color Quantization (k-means on RGB)

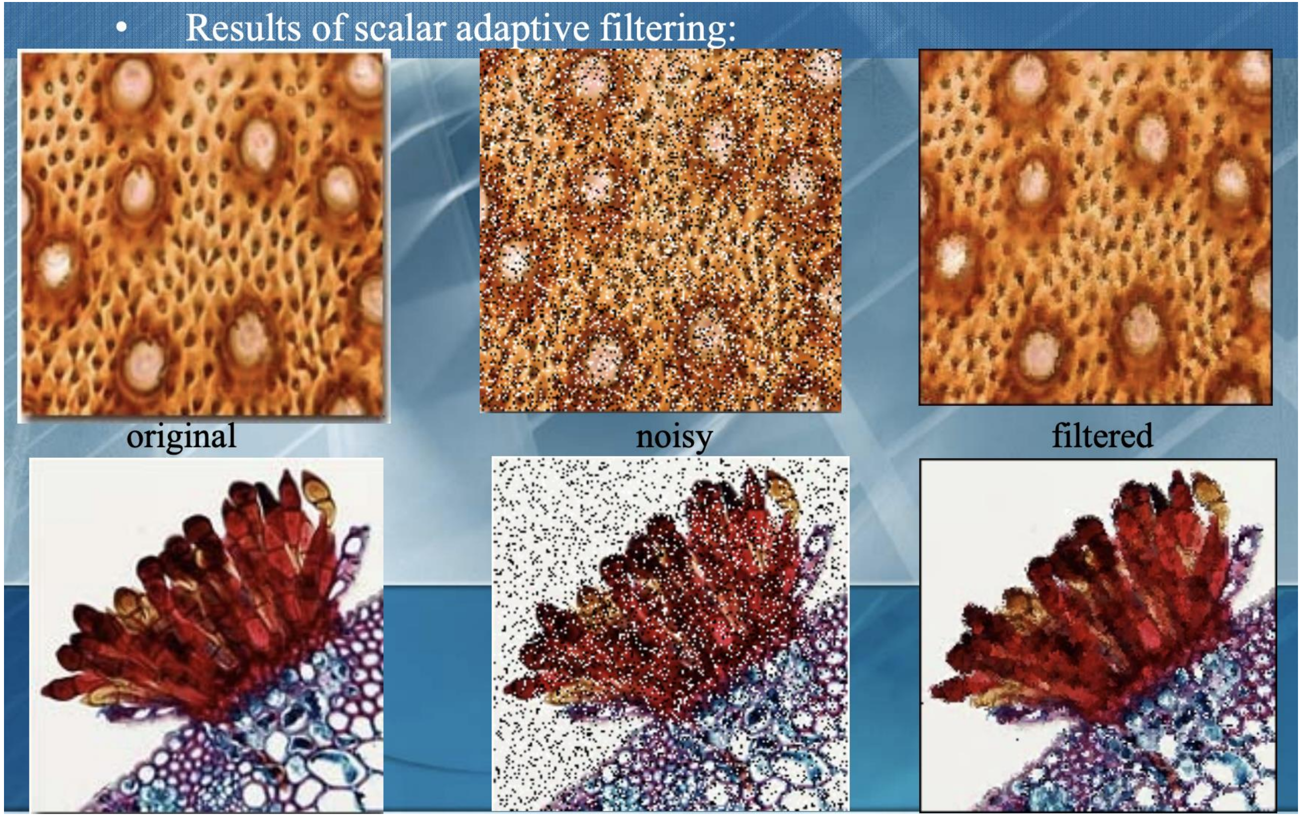






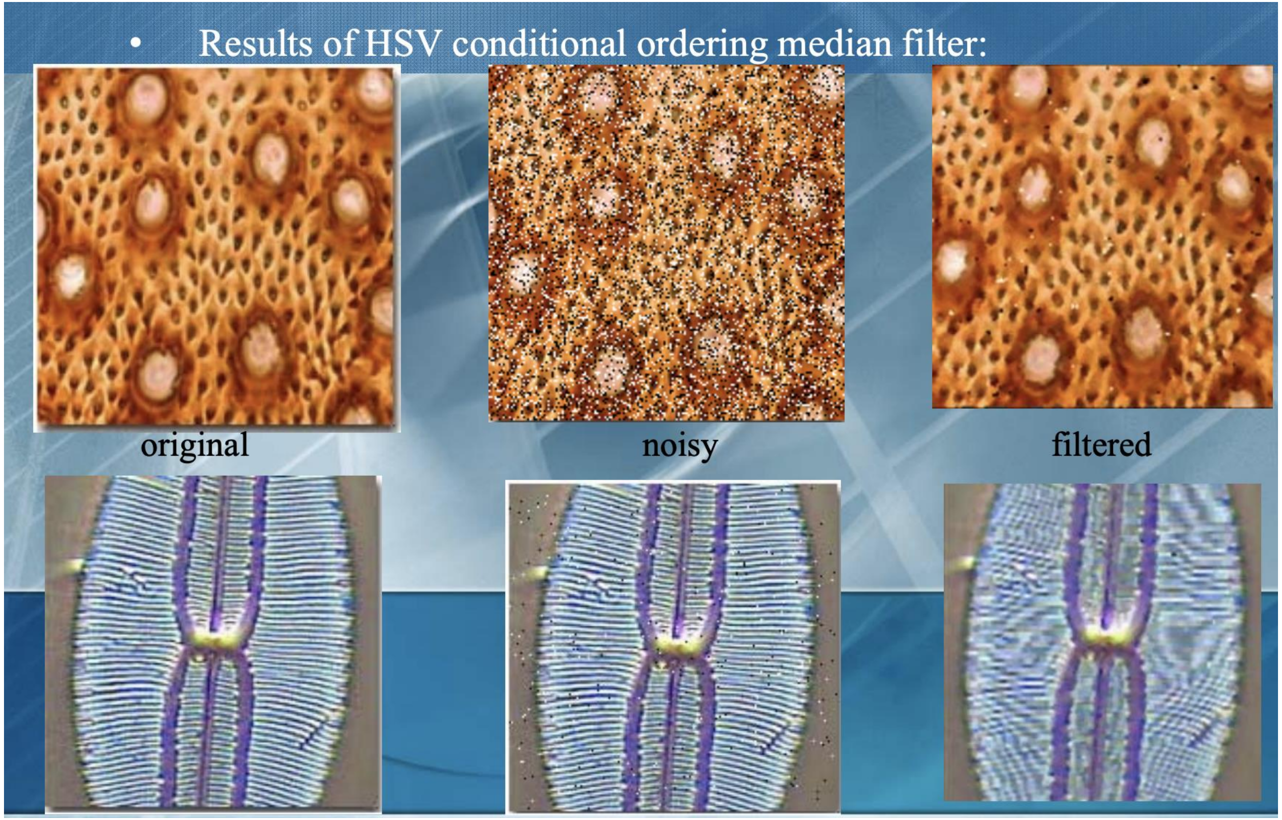
Color Image Filtering

- Median filtering



Color Image Filtering

- Median filtering



Can have various goals (more than grey level image enhancement) ; some typ

1. Image contrast enhancement
2. Color enhancement ⇔ increase of color saturation, illuminant lightin compensation, etc.

... and others....

pointwise operations

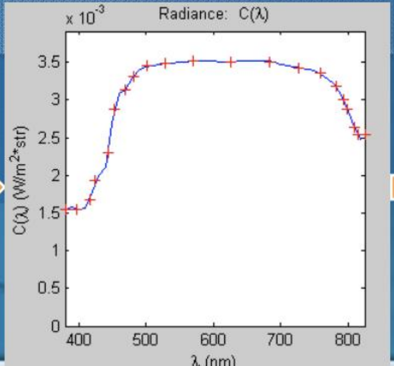
3. Image de-blurring
4. Edge enhancement

... and others...

spatial operations

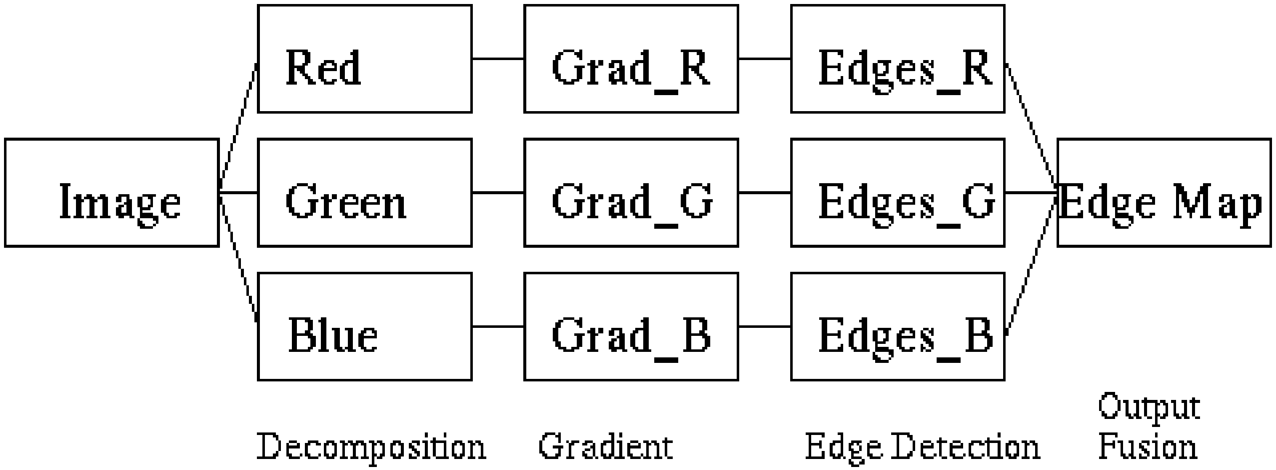
contrast

saturation

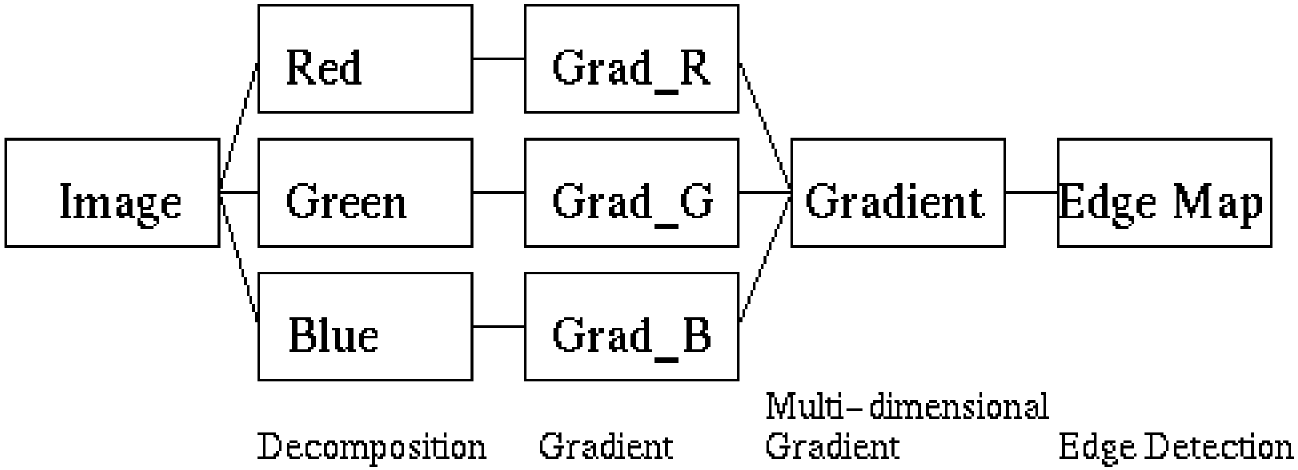


Changing illuminar

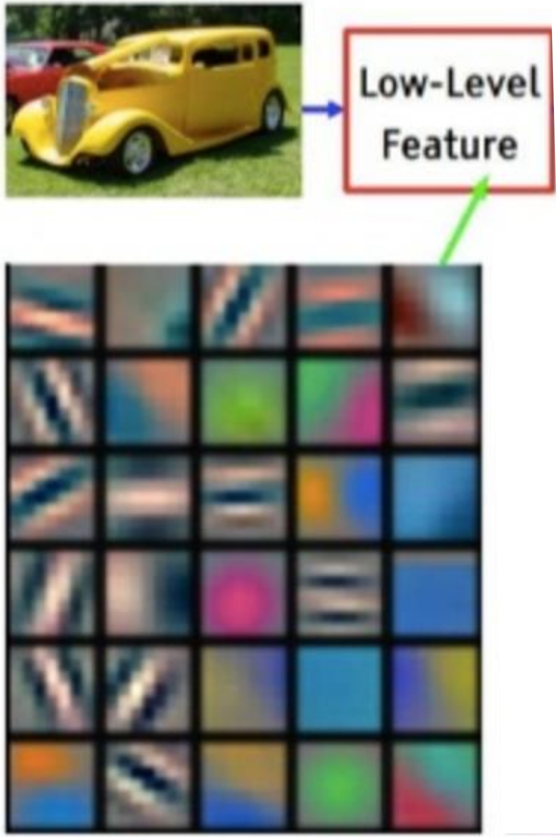
Color Edge Detection



Color Edge Detection



Convolutional Neural Network



26.10.2021

Digital Image Processing (CSE/ECE 478)

Lecture-17: IMAGE SEGMENTATION

Sudipta Banerjee

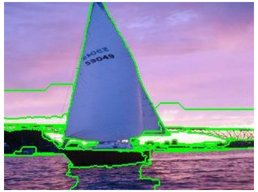
Ravi Kiran

Center for Visual Information Technology (CVIT), IIIT Hyderabad

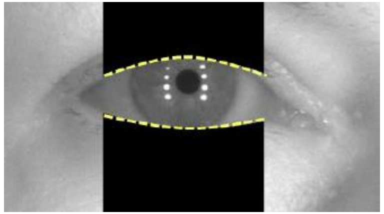
Image Segmentation

Partitioning an image into a collection of connected sets of pixels.

1. into regions, which usually cover the image



2. into linear structures, such as
   - line segments
   - curve segments



3. into 2D shapes, such as
   - circles
   - ellipses
   - ribbons (long, symmetric regions)

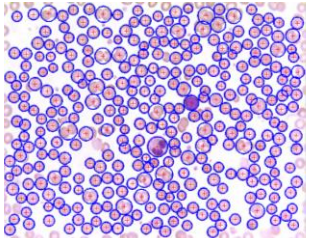


Image Segmentation - Approaches

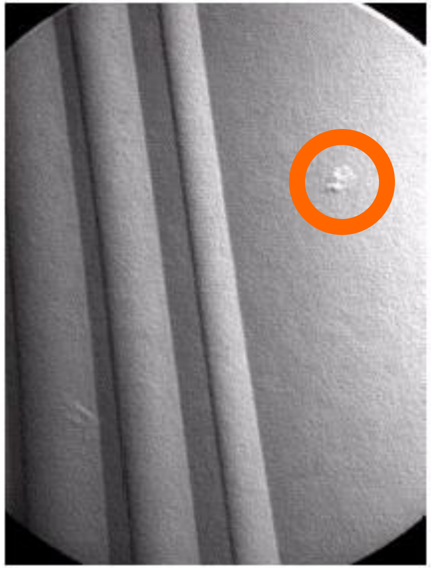
- Edge-based
- Thresholding
- Region-growing
- Morphological Watersheds
- Motion

Edge-based segmentation → Detection of Discontinuities

- Three basic types of grey level discontinuities
  - Points / Corners
  - Edges
  - Lines

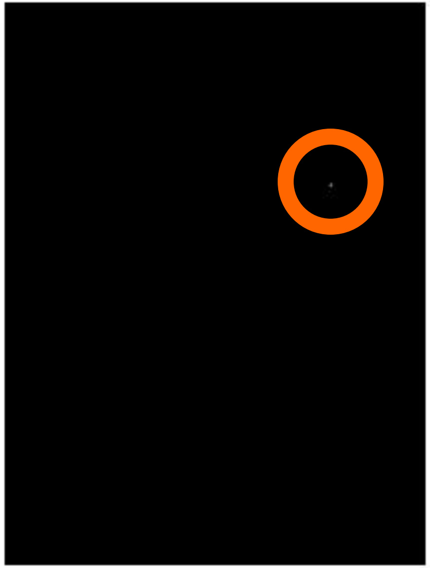
Point Detection (cont…)

−1	−1	−1
−1	8	−1
−1	−1	−1



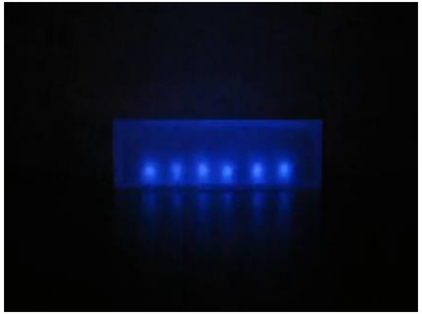
X-ray image of a turbine blade

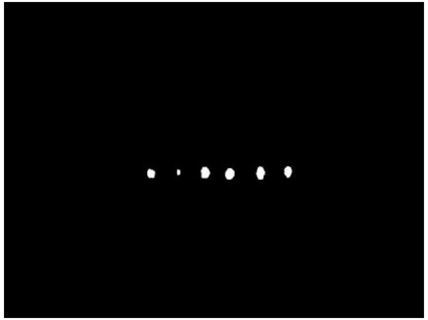
Result of point detection

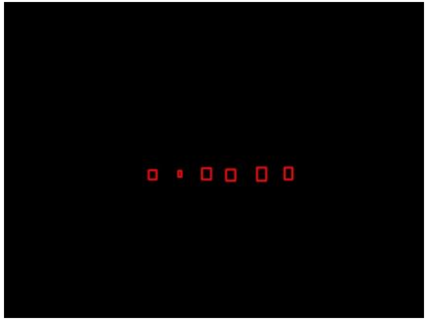


Result of thresholding

Example: Blinking/LED detector







https://stackoverflow.com/questions/10702105/detecting-led-object-status-from-image

Edges & Derivatives

- 1st derivative → edge location
- 2nd derivative →
  - edge location
  - edge direction

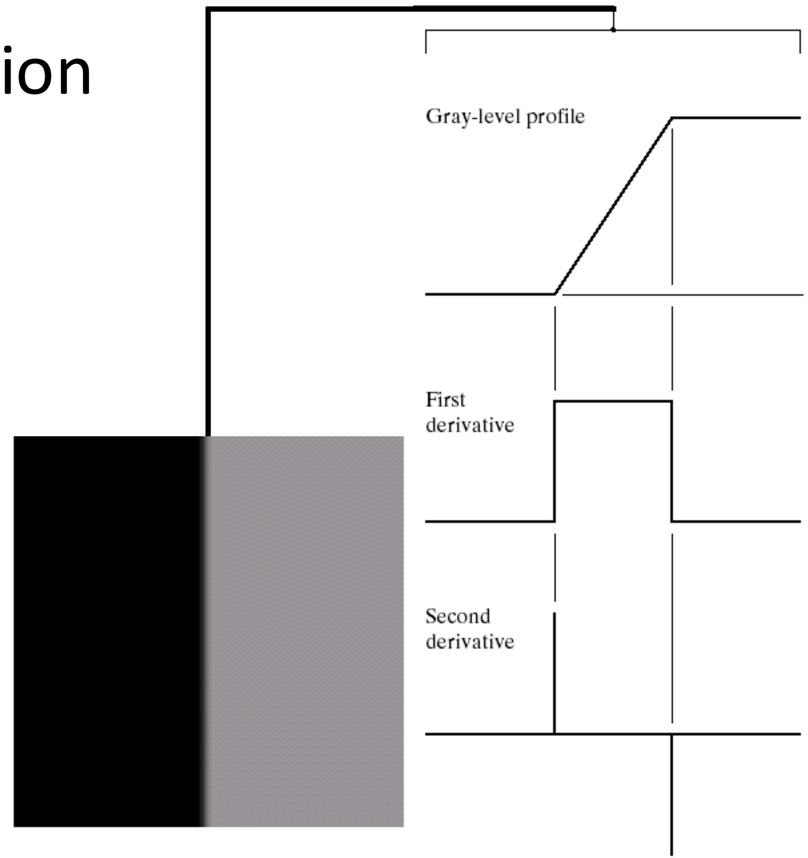


Image Gradient and Edges

$$\frac{f(x+h,y)-f(x-h,y)}{2h} \Rightarrow \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$$

x-derivative

$$\frac{f(x,y+h)-f(x,y-h)}{2h} \Rightarrow \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

y-derivative

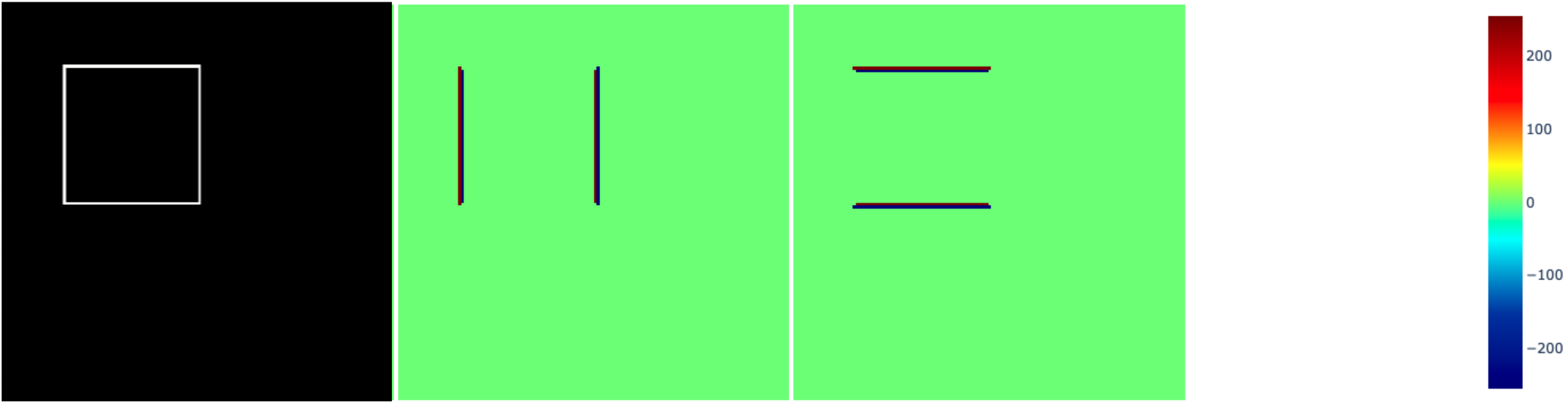


Image x-derivative y-derivative

Prewitt Edge Filter

Dr. Prewitt

https://nihrecord.nih.gov/sites/recordNIH/files/pdf/1984/NIH-Record-1984-03-13.pdf

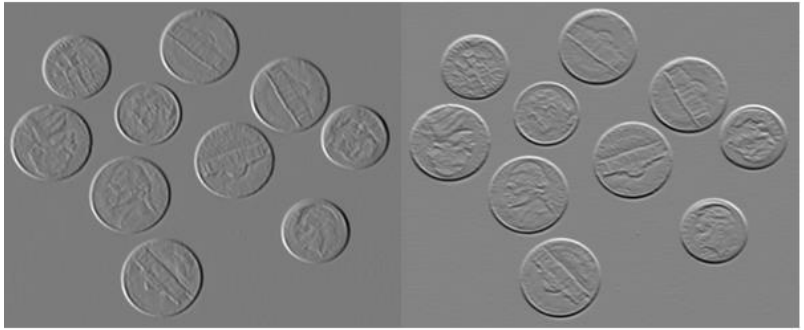
-1	0	+1
-1	0	+1
-1	0	+1

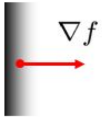
G_x

+1	+1	+1
0	0	0
-1	-1	-1

G_y

Edge is perpendicular to gradient





$\nabla f = \left[\frac{\partial f}{\partial x}, 0\right]$

$\nabla f = \left[0, \frac{\partial f}{\partial y}\right]$

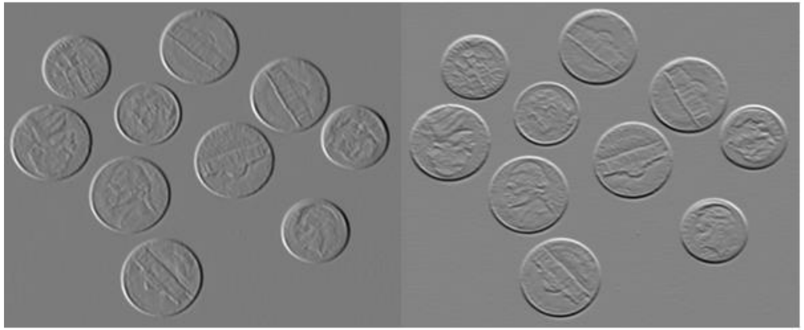
-1	0	+1
-1	0	+1
-1	0	+1

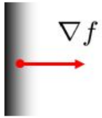
G_x

+1	+1	+1
0	0	0
-1	-1	-1

G_y

Gradient Magnitude and Orientation

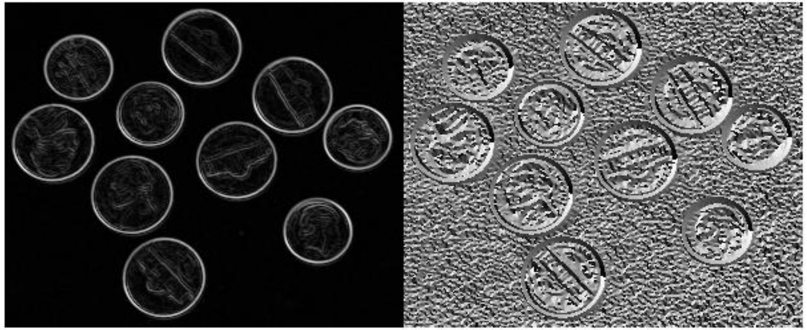


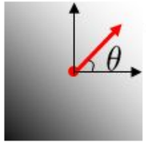


$\nabla f = \left[\frac{\partial f}{\partial x}, 0\right]$

$\nabla f = \left[0, \frac{\partial f}{\partial y}\right]$

$\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$

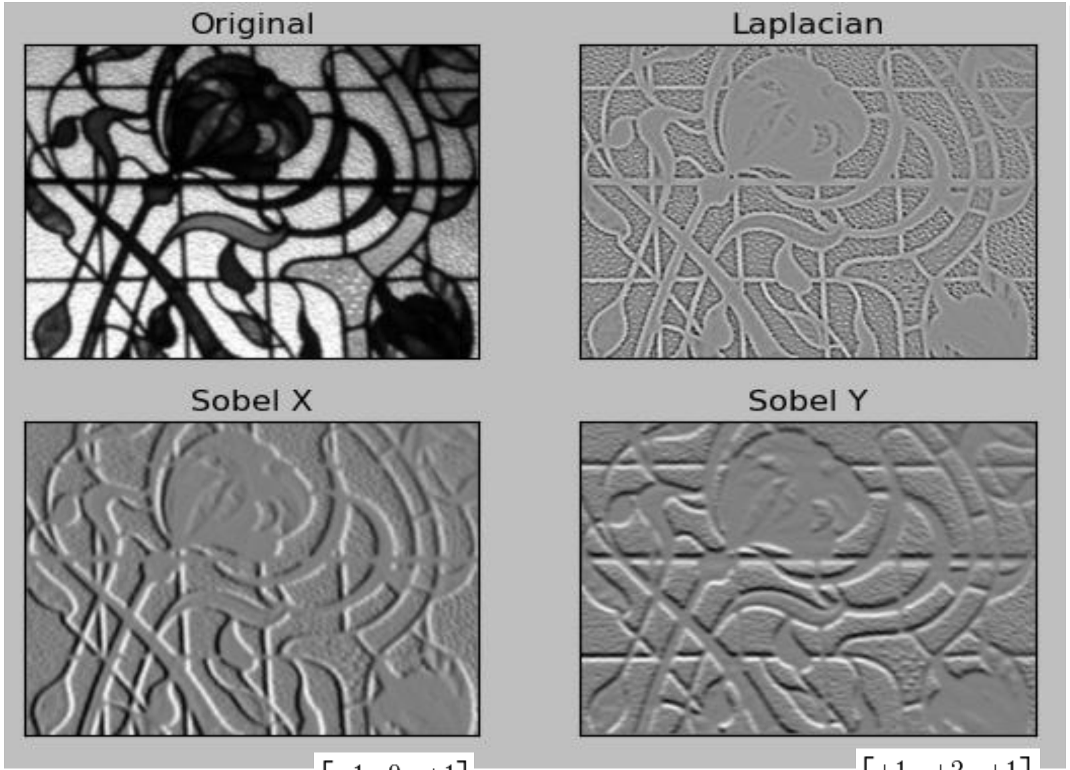




$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$

$\theta = \tan^{-1}\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right)$

Edge Masks – Sobel , Laplacian



0	−1	0
−1	4	−1
0	−1	0

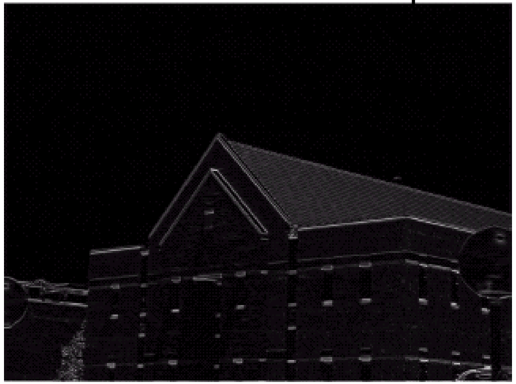
$$\begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}$$

$$\begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

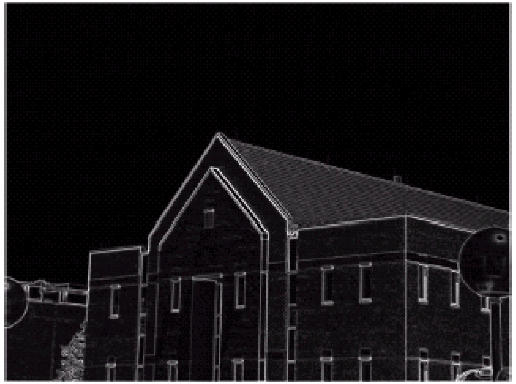
Edge Detection Example With Smoothing

Original Image

Horizontal Gradient Component

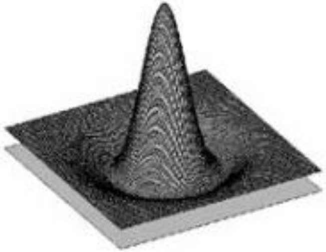


Vertical Gradient Component



Combined Edge Image

Laplacian ~ Difference of Gaussian



DoG = Difference of Gaussians

Cheap approximation - no derivatives needed.

Oriented Line Detection

−1	−1	−1
2	2	2
−1	−1	−1

Horizontal

−1	−1	2
−1	2	−1
2	−1	−1

+45°

−1	2	−1
−1	2	−1
−1	2	−1

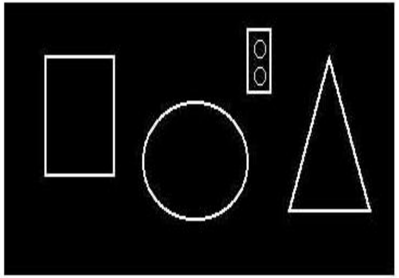
Vertical

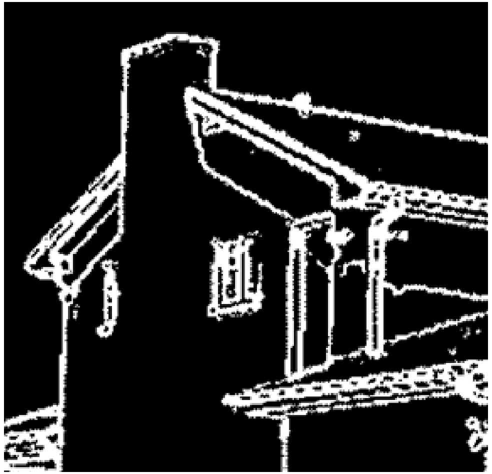
2	−1	−1
−1	2	−1
−1	−1	2

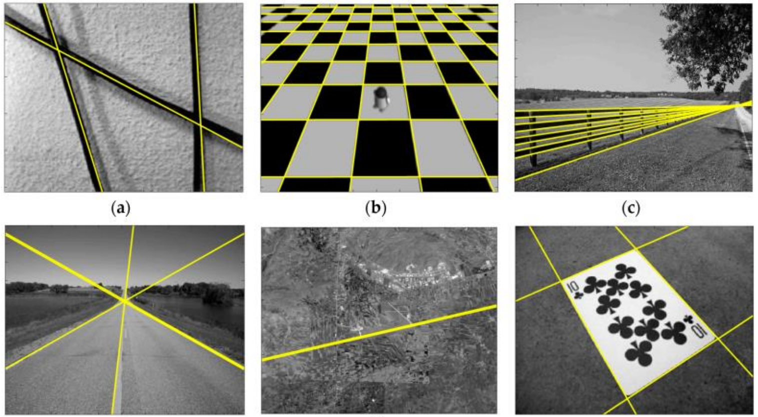
−45°

Hough Transform

- Straight lines
- Circles
- Algebraic curves
- Arbitrary specific shapes in an image

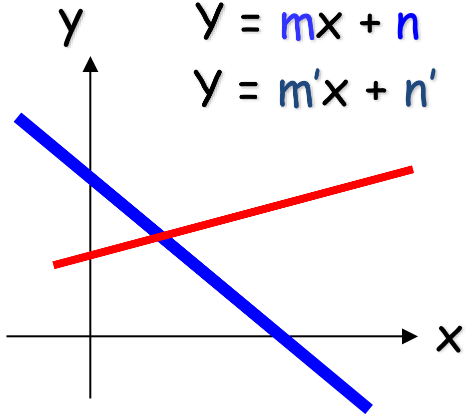






(a) (b) (c)
(d) (e) (f)

Hough Transform for Lines: Image and Parameter Spaces



$y = mx + n$

$y = m'x + n'$

$y_1 = m\,x_1 + n$

$y_2 = m\,x_2 + n$

$y_N = m\,x_N + n$

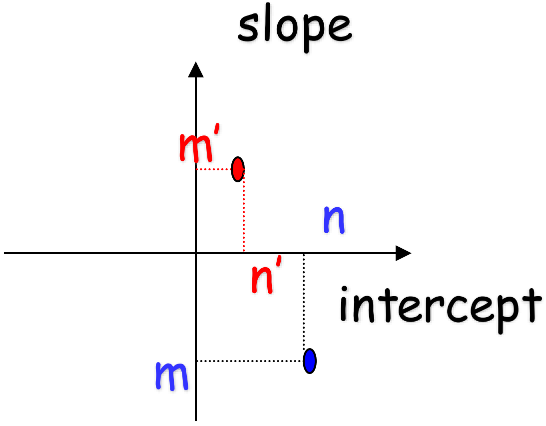


Image Space

Parameter Space

Line in Img. Space ~ Point in Param. Space

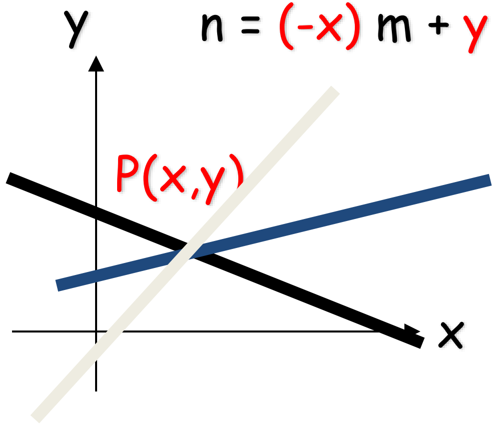
Image Parameter Spaces

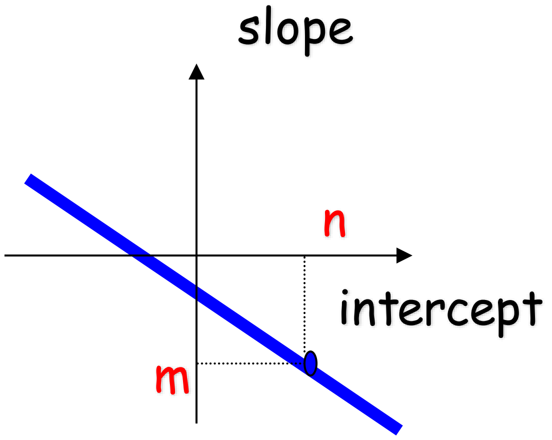
- Image Space
  - Lines
  - Points
  - Collinear points

- Parameter Space
  - Points
  - Lines
  - Intersecting lines

Hough Transform Technique

- Given an edge point, there is an infinite number of lines passing through it (Vary m and n).
  - These lines can be represented as a line in parameter space.

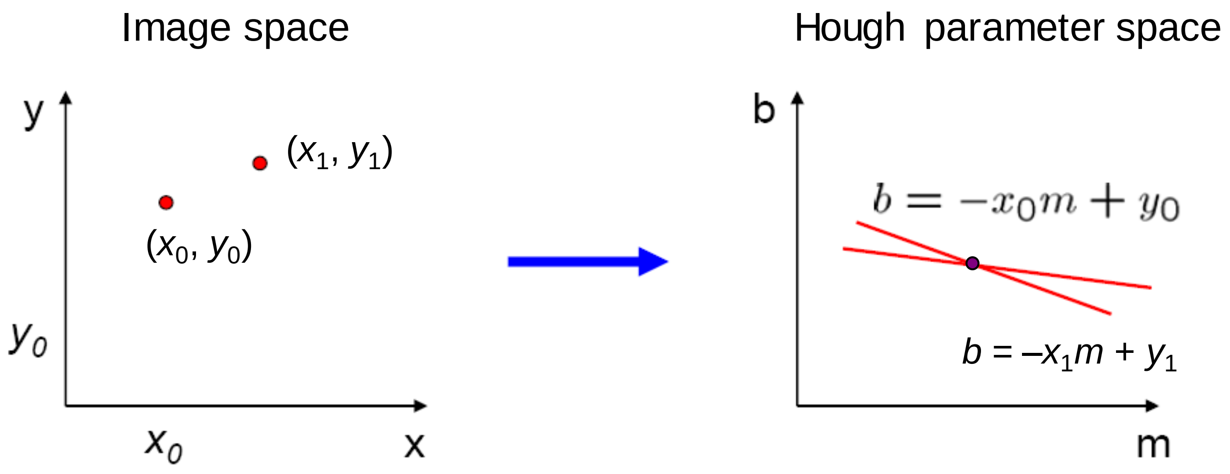


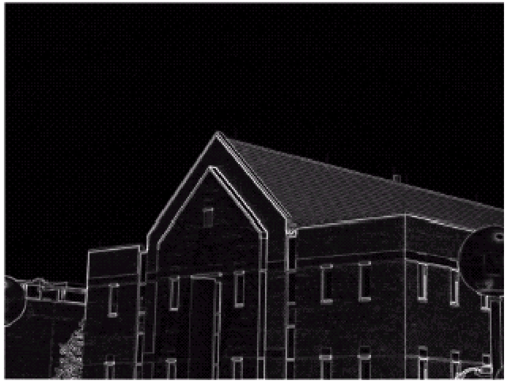


Parameter Space

Parameter space representation

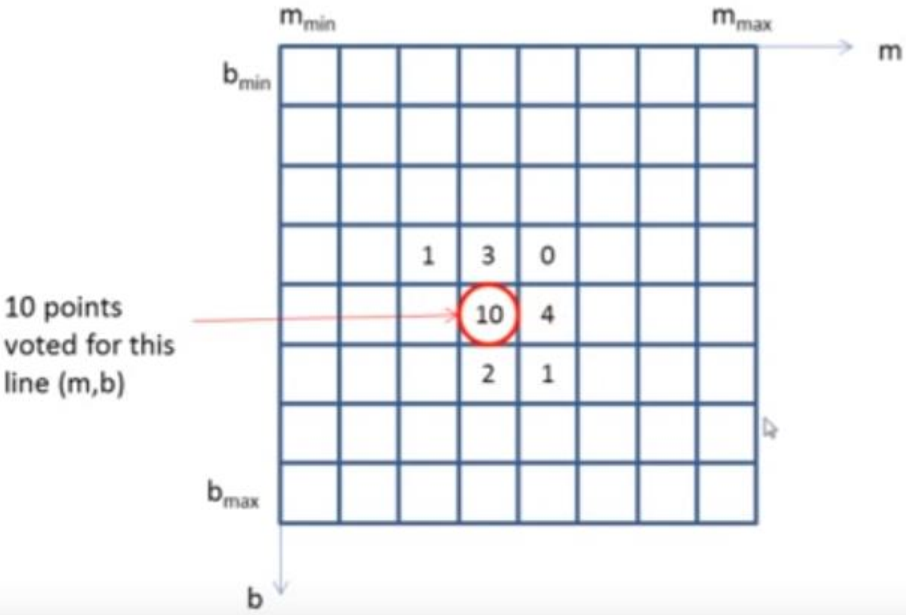
- Where is the line that contains both (x_0, y_0) and (x_1, y_1)?
  - It is the intersection of the lines $b = -x_0m + y_0$ and $b = -x_1m + y_1$





Hough Transform Algorithm

- Initialize an accumulator array A(m,b) to zero
- For each edge element (x,y), increment all cells that satisfy b = -x m + y
- Local maxima in A(m,b) correspond to lines

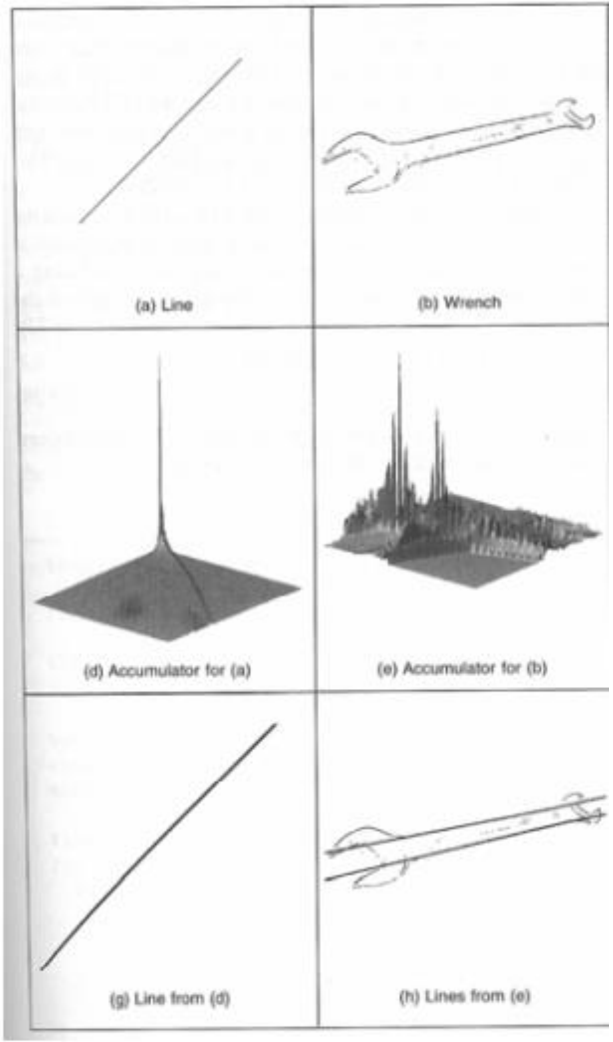


https://www.youtube.com/watch?v=uDB2qGqnQ1g

Thresholded edge images

Visualizing the accumulator space
The height of the peak will be defined by the number of pixels in the line.

Thresholding the accumulator space and superimposing this onto the edge image

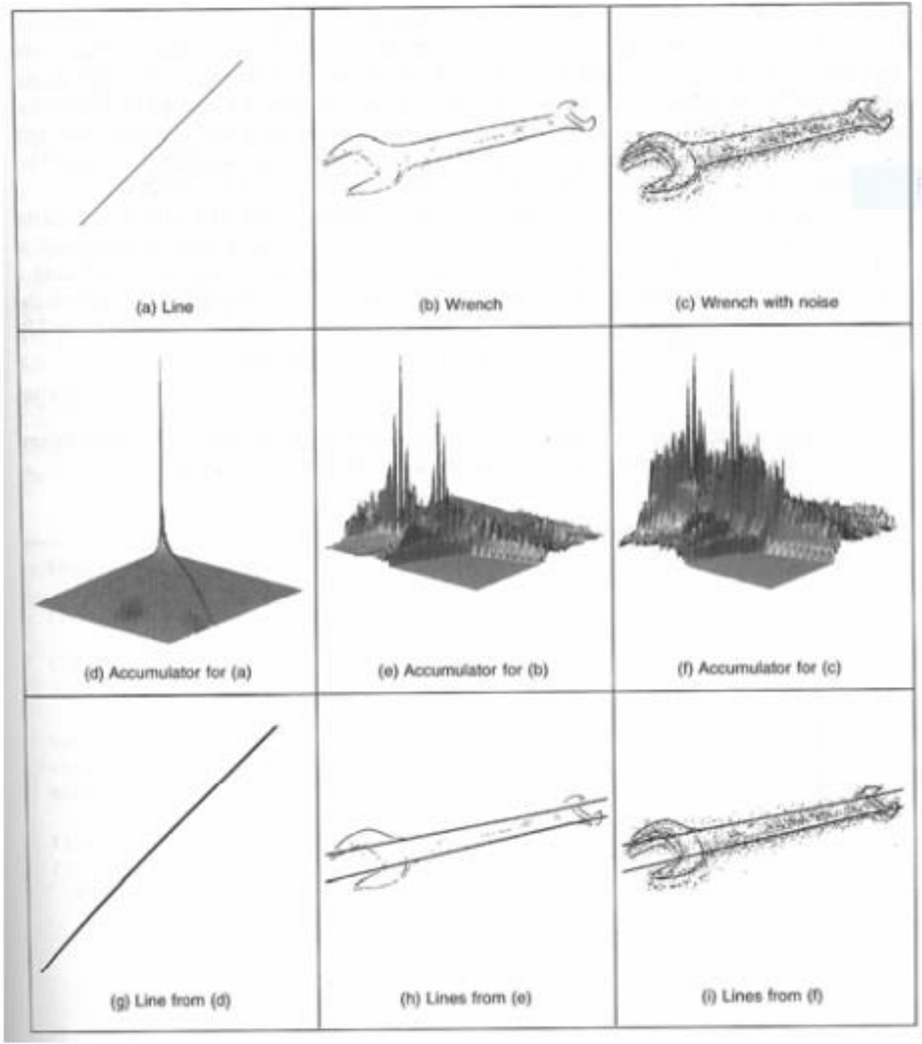


Thresholded edge images

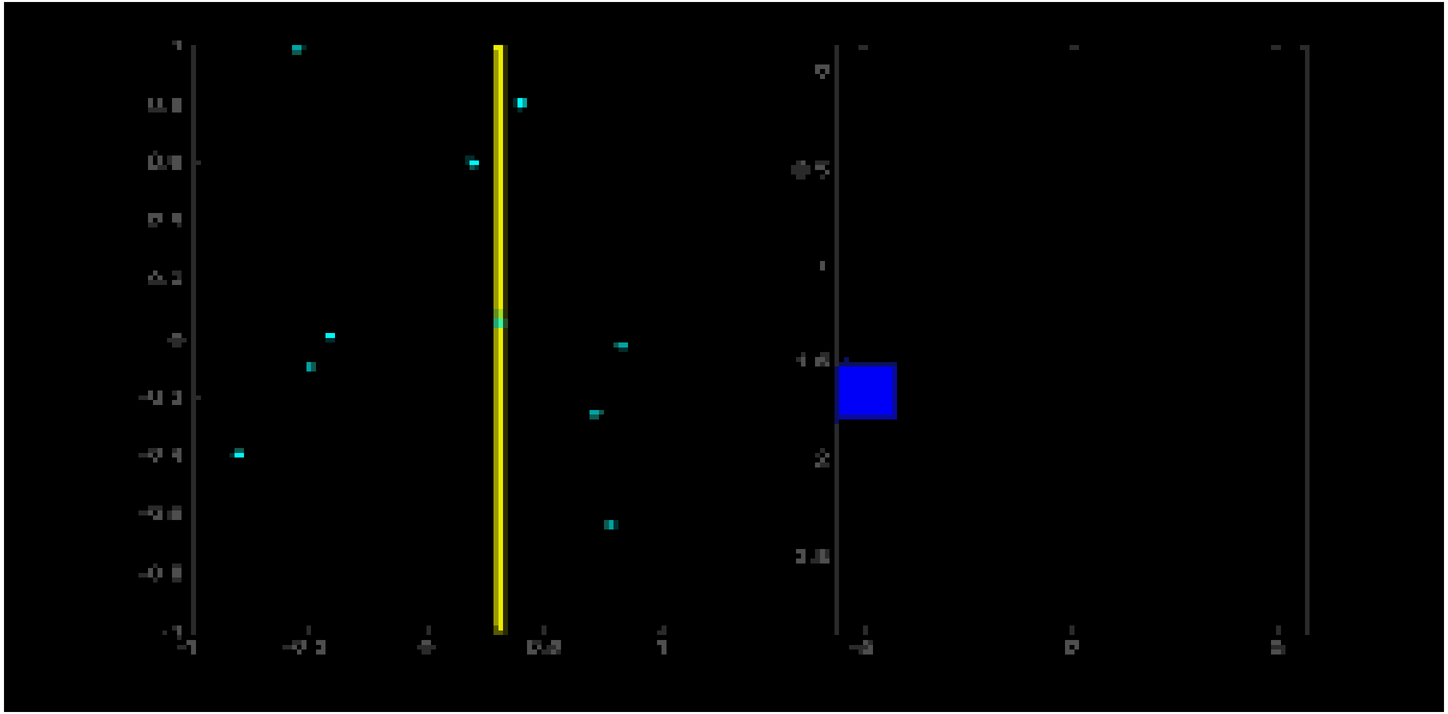
Visualizing the accumulator space

The height of the peak will be defined by the number of pixels in the line.

Thresholding the accumulator space and superimposing this onto the edge image



Animation



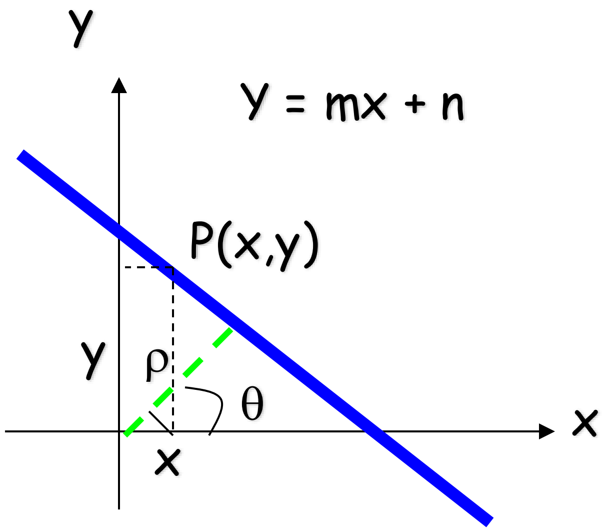
http://homepages.inf.ed.ac.uk/amos/hough.html

Practical Issues with This Hough Parameterization

- The slope of the line is $-\infty < m < \infty$
  - The parameter space is INFINITE
- The representation $y = mx + n$ does not express lines of the form $x = k$

Solution:

- Use the "Normal" equation of a line:



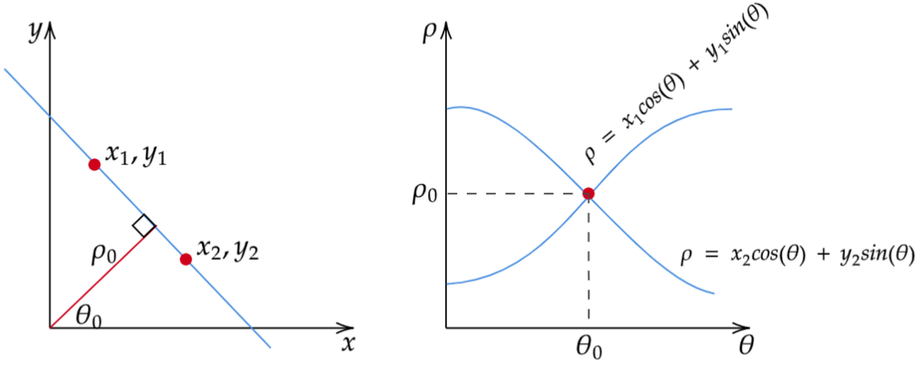
$\rho = x\cos\theta + y\sin\theta$

θ Is the line orientation

ρ Is the distance between the origin and the line

Consequence:

- A Point in Image Space is now represented as a SINUSOID
  - $\rho_0 = x \cos\theta_0 + y \sin\theta_0$



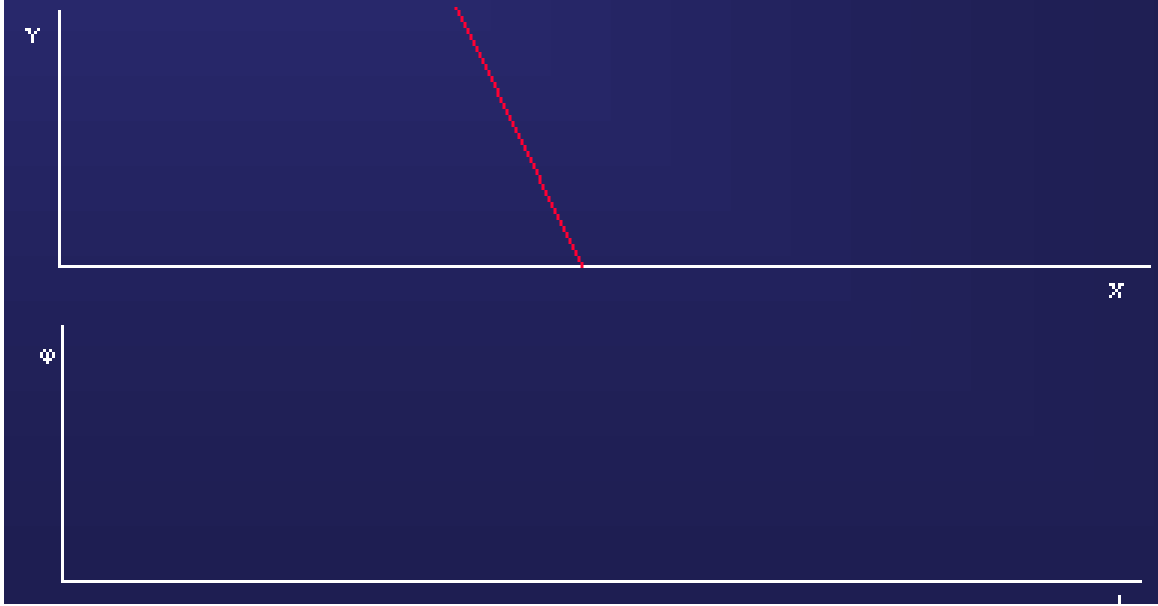
https://miro.medium.com/max/1400/1*Cr73Mte5NNgO16D4moKDQg.png

New Parameter Space for Hough based on trigonometric functions

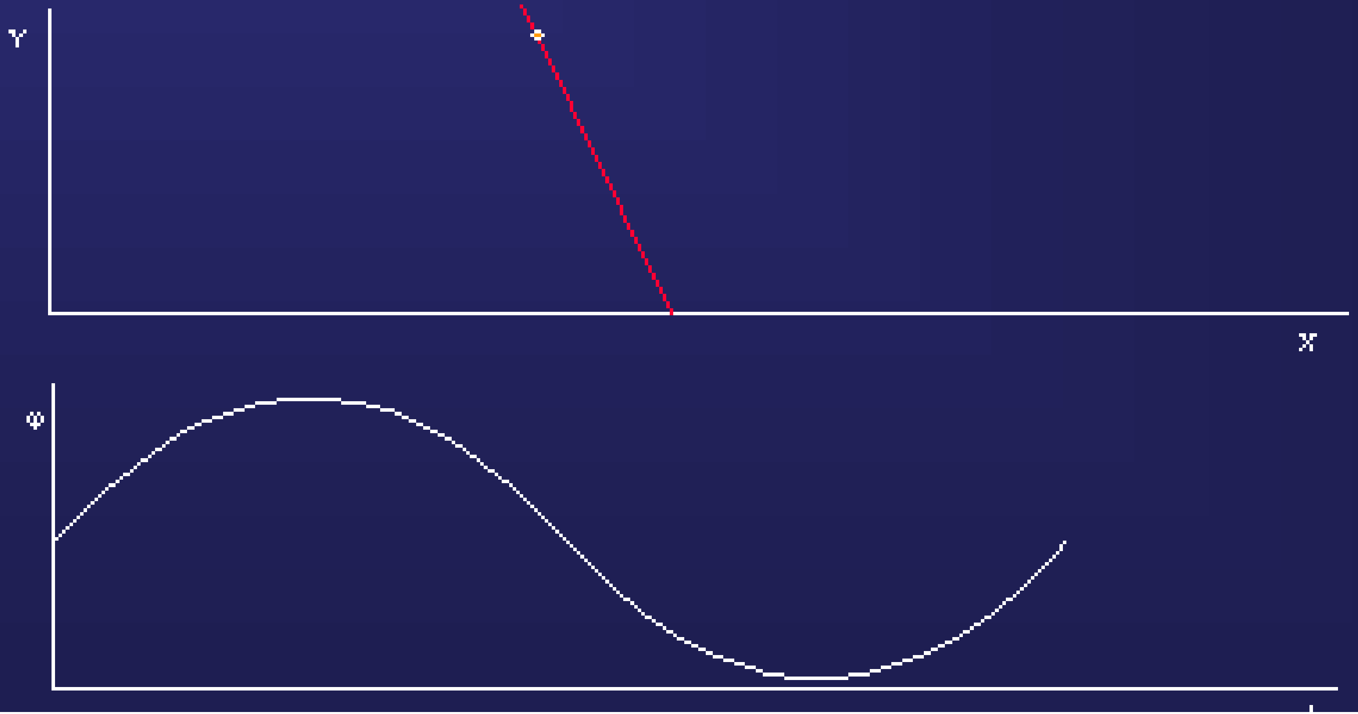
- Use the parameter space (ρ, θ)
- The new space is FINITE
  - $0 < \rho < D$, where D is the image diagonal.
  - $-\pi < \theta < \pi$
- The new space can represent all lines
  - $Y = k$ is represented with $\rho = k$, $\theta=90$
  - $X = k$ is represented with $\rho = k$, $\theta=0$

SHT: Another Viewpoint

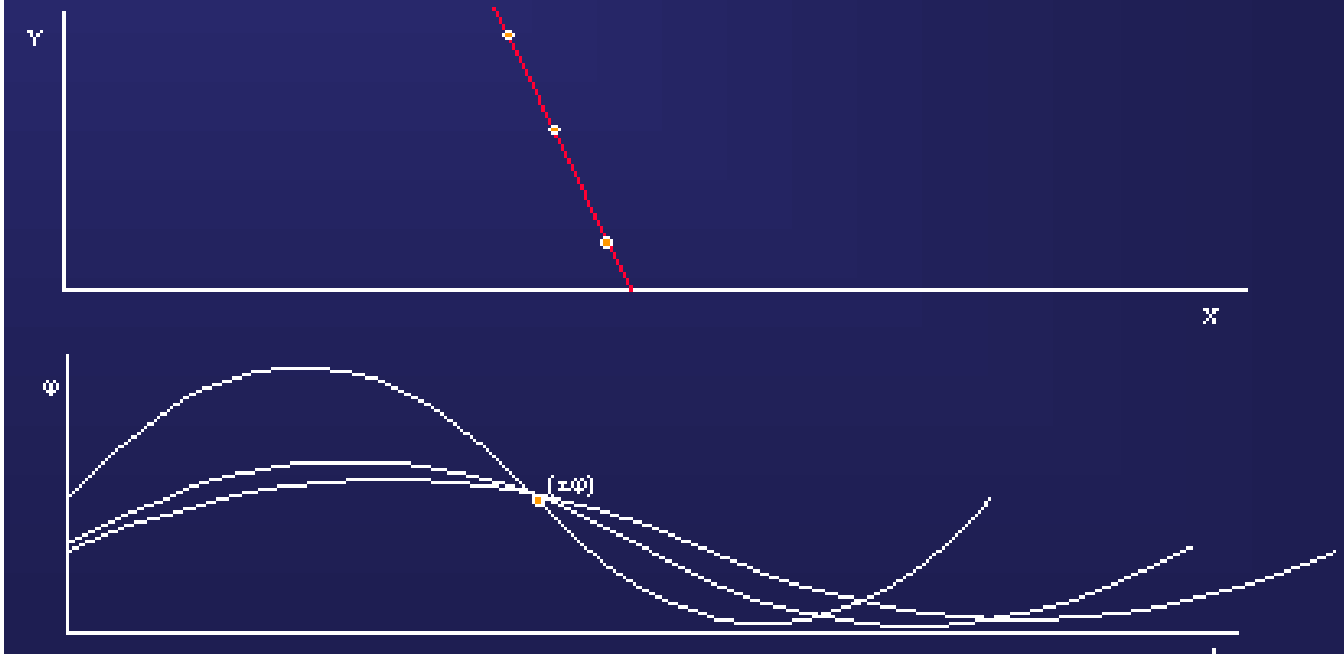
- Since any (r,φ) represents a point in parameter space we may plot all possible (r,φ) points for each (x,y) poi



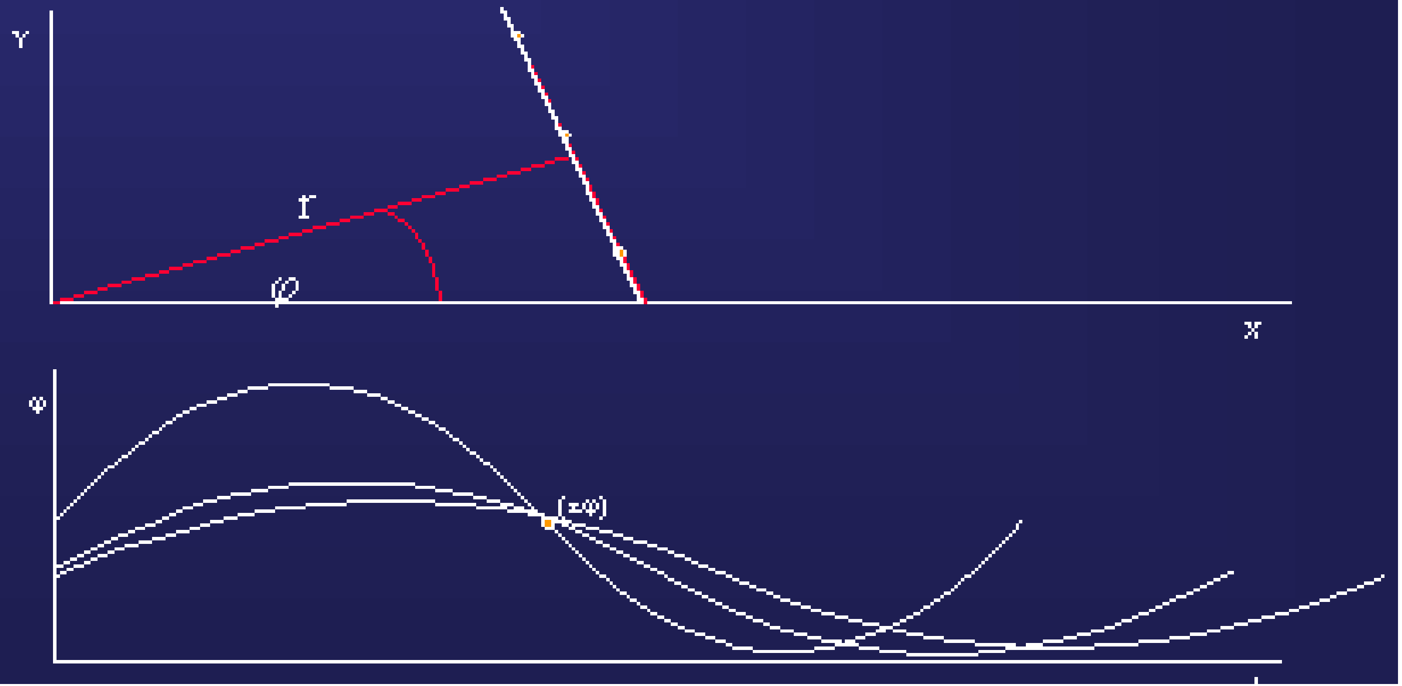
- Taking any particular (x,y) point its set of (r,φ) points is a sinusoid through parameter space



- The line upon which the (x,y) points lie is then given by the (r,φ) pair on which all the sinusoids *agree* (i.e. intersect)



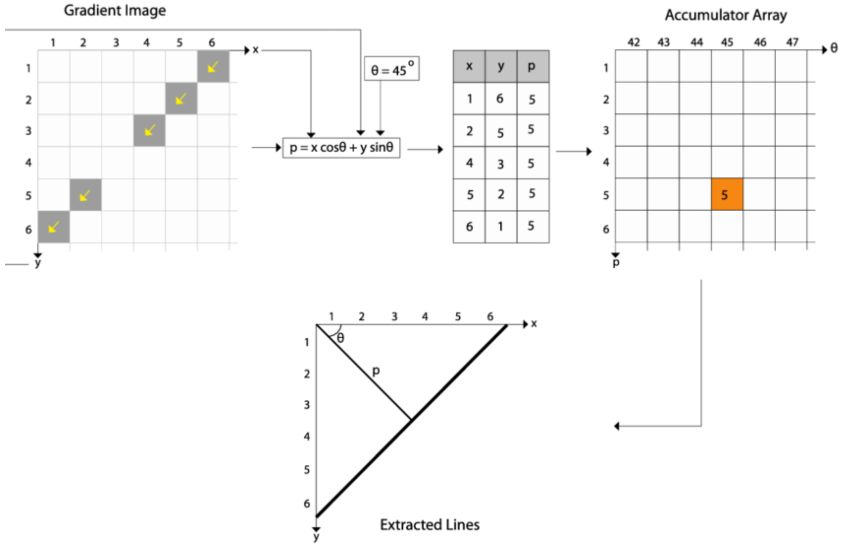
- Finding this intersection we then have the required (r,φ) parameters and therefore the line in the image



Hough Transform Algorithm

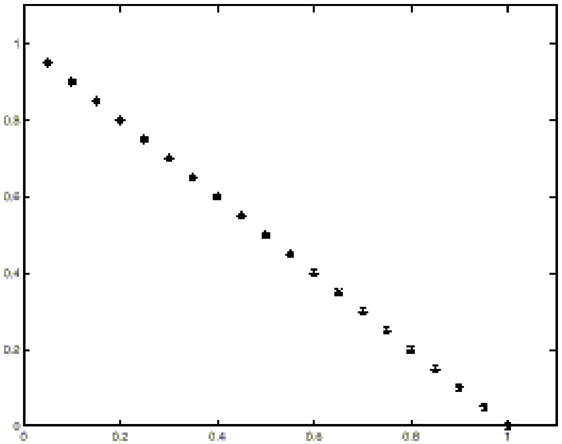
Input is an edge image (E(i,j)=1 for edgels)

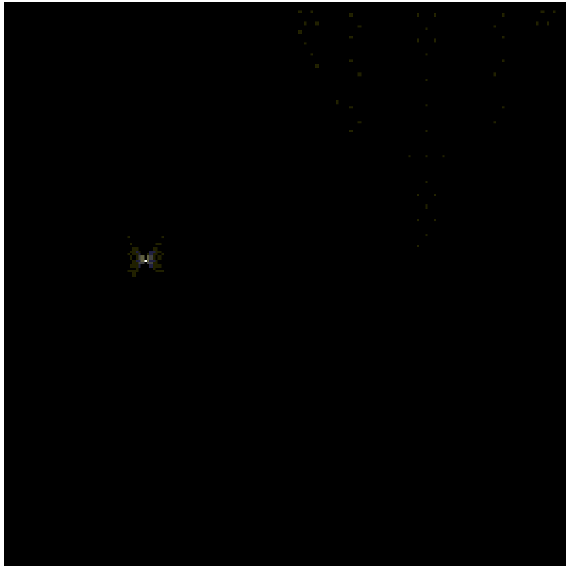
1. Discretize θ and ρ in increments of $d\theta$ and $d\rho$. Let A(R,T) be an array of integer accumulators, initialized to 0.
2. For each pixel E(i,j)=1 and h=1,2,…T do
   1. $\rho = j\cos(h * d\theta) + i\sin(h * d\theta)$
   2. Find closest integer k corresponding to ρ
   3. Increment counter A(h,k) by one
3. Find local maxima in A(R,T)

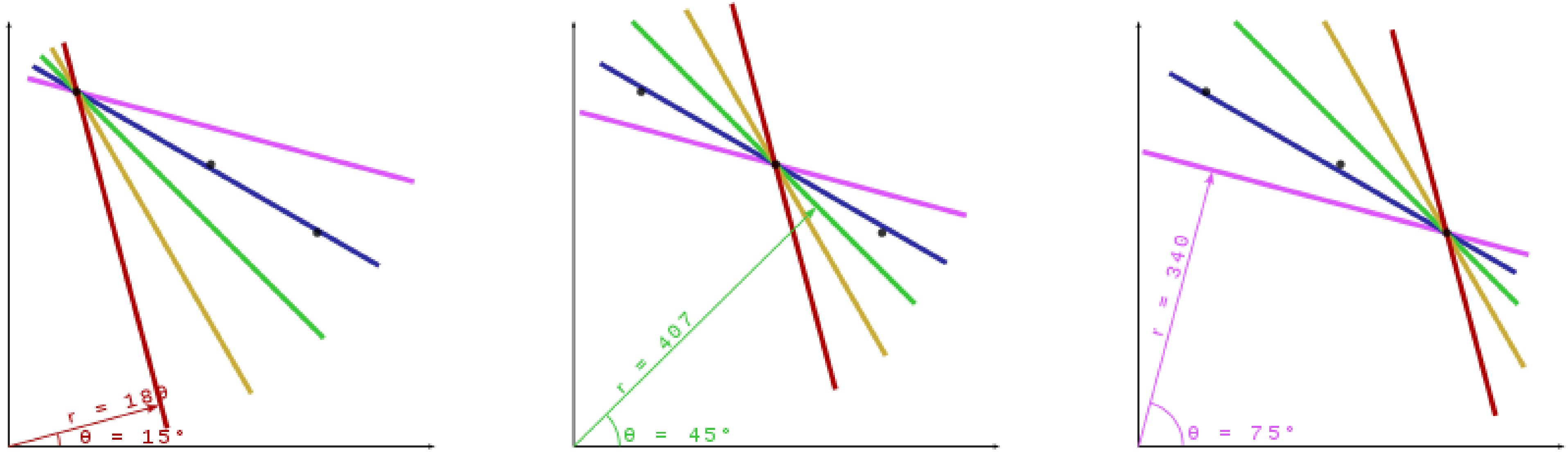


x	y	p
1	6	5
2	5	5
4	3	5
5	2	5
6	1	5

Hough Transform – cont.







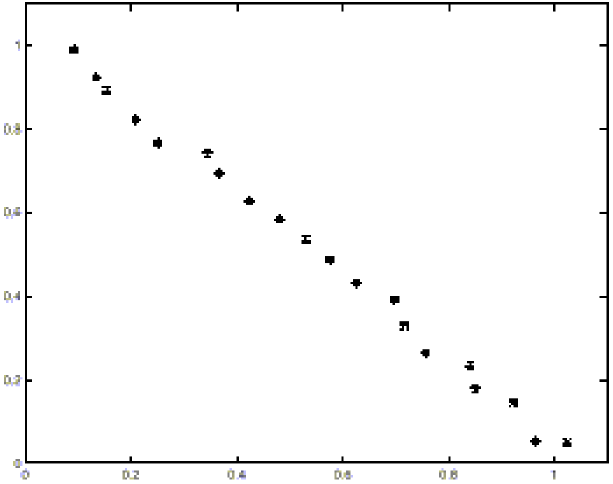
θ	r
15	189.0
30	282.0
45	355.7
60	407.3
75	429.4

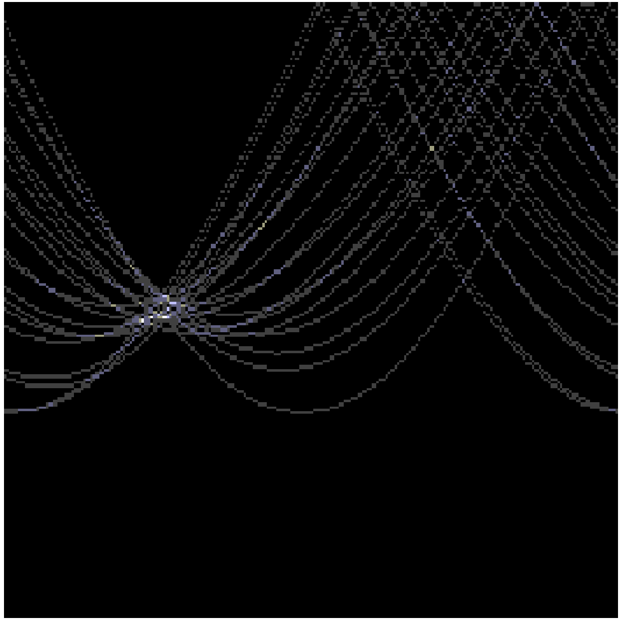
θ	r
15	318.5
30	376.8
45	407.3
60	409.8
75	385.3

θ	r
15	419.0
30	443.6
45	438.4
60	402.9
75	340.1

Wikipedia

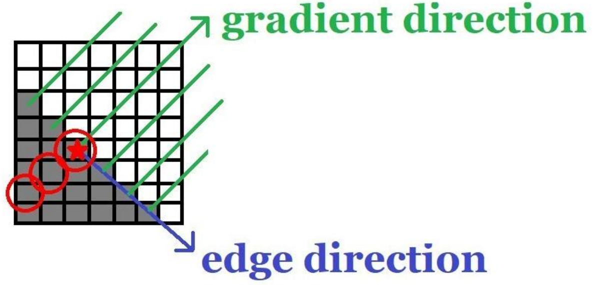
Hough Transform – cont.





Hough Transform Speed Up

- If we know the orientation of the edge – usually available from the edge detection step
  - We fix theta in the parameter space and <u>increment **only one** counter</u>!



Hough Transform Speed Up

- If we know the orientation of the edge – usually available from the edge detection step
  - We fix theta in the parameter space and increment **only one** counter!
  - We can allow for orientation uncertainty by incrementing a *few* counters around the "nominal" counter.

HT for Circles

- Extend HT to other shapes that can be expressed parametrically
- Circle, fixed radius r, centre (a,b)
  - $(x_1-a)^2 + (x_2-b)^2 = r^2$
  - accumulator array must be 3D
  - unless circle radius, r is known
  - re-arrange equation so x_1 is subject and x_2 is the variable
  - for every point on circle edge (x,y) plot range of (x_1,x_2) for a given r

Hough circle Fitting

- Implicit circle equation:
  $(x - a)^2 + (y - b)^2 = r^2$

Optimization: HT for circles

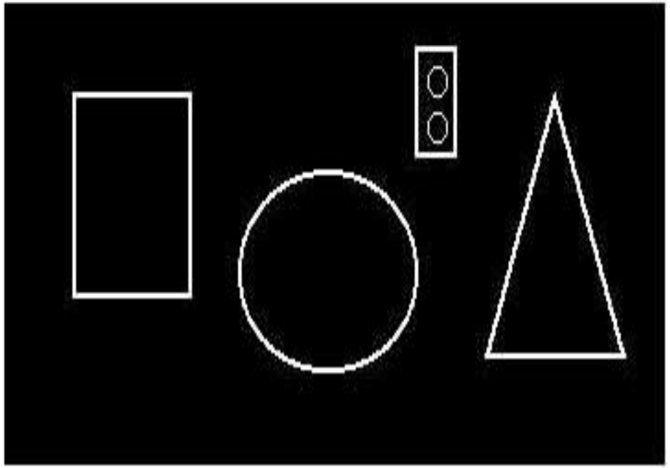
- With edge direction
  - edge directions are quantized into 8 possible directions
  - only 1/8 of circle needs take part in accumulator
- Using edge directions
  - a & b can be evaluated from

$$a = x_1 - R\cos(\psi(\mathbf{x}))$$
$$b = x_2 - R\sin(\psi(\mathbf{x}))$$
$$\psi(\mathbf{x}) \in [\phi(\mathbf{x}) - \Delta\phi\,,\ \phi(\mathbf{x}) + \Delta\phi]$$

  - θ = edge direction in pixel x
  - delta θ = max anticipated edge direction error
- Also, weigh contributions to accumulator A(a) by edge magnitude

Hough Transform Generalizations

- It locates straight lines (SHT) - standard, simple HT
- It locates straight line intervals
- It locates circles
- It locates algebraic curves
- It locates arbitrary specific shapes in an image
  - **But you pay progressively for complexity of shapes by time and memory usage**



Hough Transform – cont.

- More complicated shapes
  - Can be used to find shapes with arbitrary complexity
  - … as long as we can describe the shape with some fixed number of parameters
  - The number of parameters required indicates the dimensionality of the accumulator

Generalized Hough Transform

- Some shapes may not be easily expressed using a small set of parameters
  - In this case, we explicitly list all the points on the shape
  - This variation of Hough transform is known as generalized Hough transform

Hough Transform: Philosophy

- A method for detecting straight lines, shapes and curves in images.
- Main idea:
  - Map a difficult pattern problem into a simple **peak detection** problem