Digital Image Processing

Ch-3 Intensity Transformations & Spatial Filtering

Introduction to Image Enhancement using Spatial domain

Image Enhancement : Process → Improves quality of image

→ Highlight important details

→ To remove noise → image → more appealing

Methods :

1) Spatial domain : manipulation of pixel values

2) Freq. domain : Modifying the F.T of image

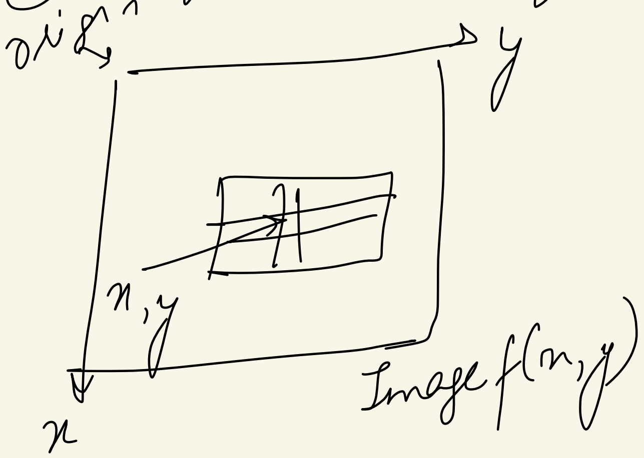
3) Combination method : 1 + 2

Spatial domain : Image plane itself

→ Direct manipulation of pixels

(a) Intensity Transformation

(b) Spatial filtering



$$g(x,y) = T(f(x,y))$$

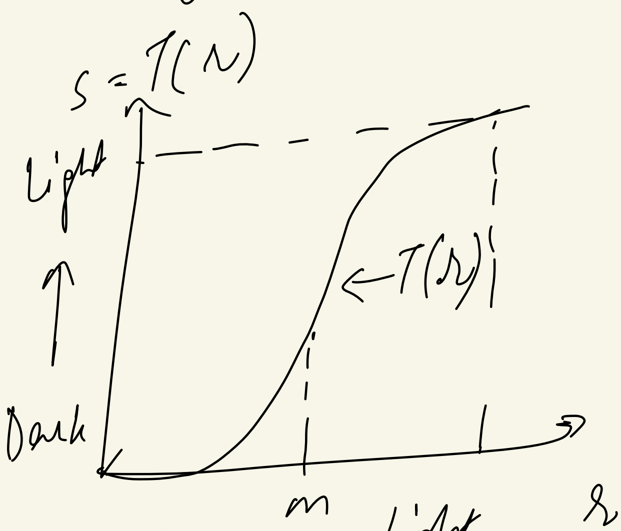
Simplest form → neighbourhood 1x1

T → grey level transformation

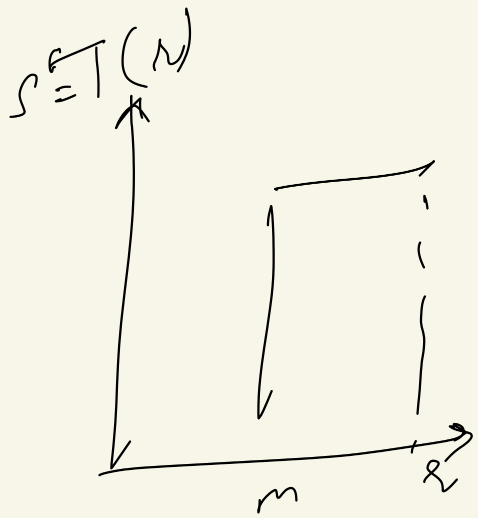
intensity or mapping

$S = T(r)$ ← I/P pixel value

↑ O/P pixel value



Contrast stretching



Point processing

mask → Small 3x3 2D array

Filters / Kernels / Templets / window

Mask processing or Filtering

Basic Intensity transformation fn.

Grey level transformation:

$S = T(r)$

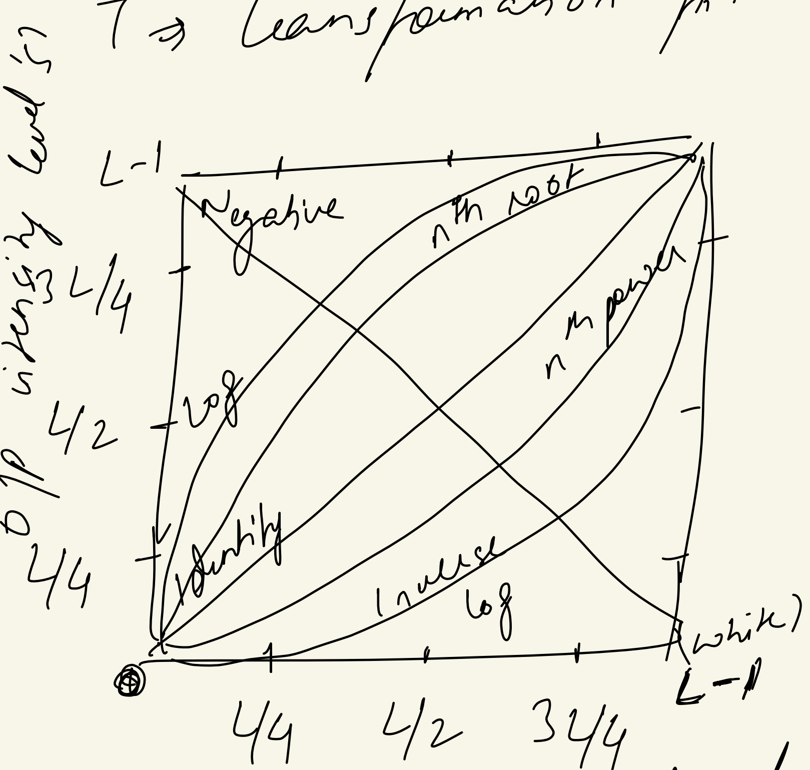
r → value of pixel before preprocessing

S → value of pixel after processing

$T \Rightarrow$ Transformation fn.

$2^{8} = 256$

0 to 255



① Linear (Negative & Identity)

② Logarithmic (log & inverse log)

③ Power law (nth power & nth root)

a) Image negative

Intensity level $[0, L-1]$

$$\boxed{S = L - 1 - r}$$

Eg: $r = 0$

$S = L - 1$

$r = L - 1$

$S = 0$

b) log transformation

$$s = c \log(1 + r)$$ $c \rightarrow$ constant

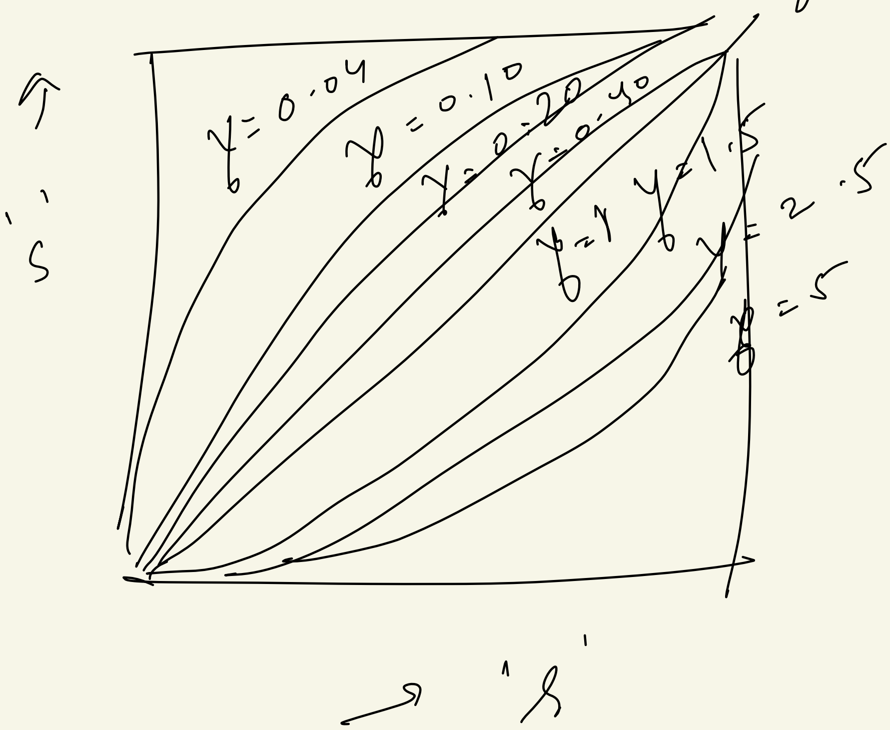
$r \geq 0$

c) power law [gamma correction]

$$s = c r^{\gamma}$$

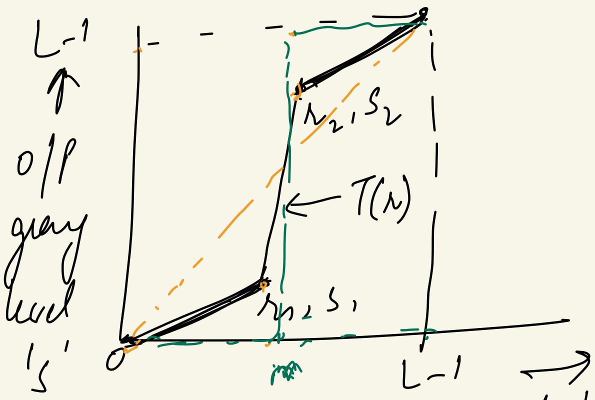
$\gamma > 1 \rightarrow n^{th}$ power

$\gamma < 1 \rightarrow n^{th}$ root



Piecewise-linear Transformation func.

① Contrast Stretching

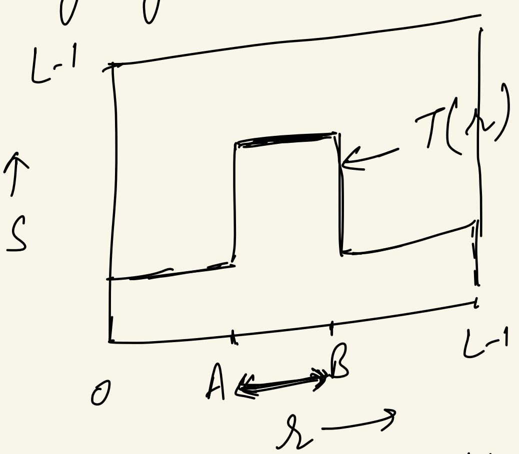


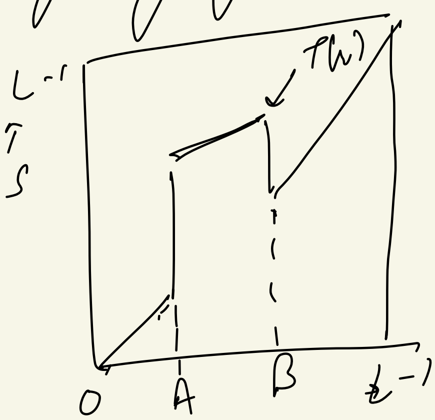
makes
→ Dark portion darker
→ light portion lighter

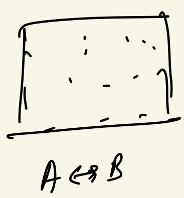
1. $r_1 = s_1$ & $r_2 = s_2$ → linear Transformation
2. $r_1 = r_2$ & $s_1 = 0$, $s_2 = L-1$
↳ Thresholding
3. Intermediate values (r_1, s_1) & (r_2, s_2) ⇒ various degrees of spread in gray levels
4. Generally $r_1 \leq r_2$ & $s_1 \leq s_2$ ⇒ Single valued & monotonically increasing

② Gray level slicing

To Highlight specific range of gray levels

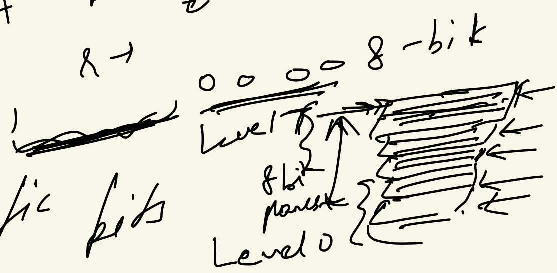




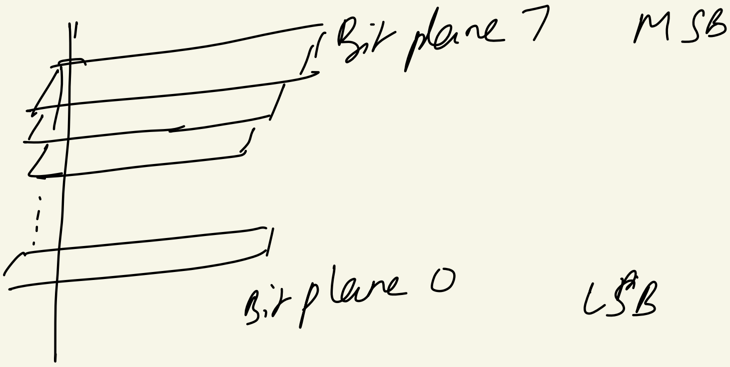


③ Bit plane slicing

It highlights contribution of specific bits



→ 8 bit image
→ Top 4 bits ⇒ majority of visually significant data
→ Useful for analysing relative importance of each bit
→ Image compression



Histogram Equalization : Image Enhancement

→ Graphical representation of data

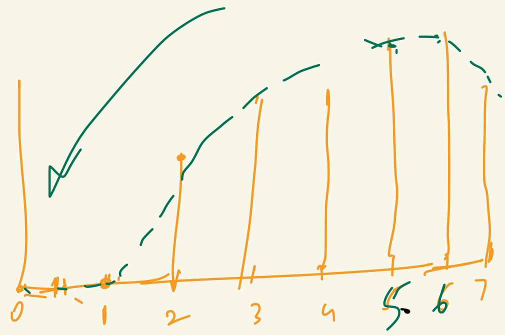
→ Representation : Frequency of occurrence of various gray levels

Eg:

6	6	7	7	6
5	2	2	3	4
3	3	4	4	5
5	7	3	6	2
7	6	5	5	4

5x5

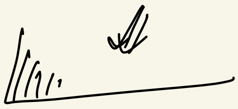
$0 \to 0$
$1 \to 0$
$2 \to 3$
$3 \to 4$
$4 \to 4$
$5 \to 5$
$6 \to 5$
$7 \to 4$



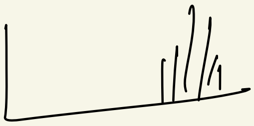
→ Used for manipulating contrast & brightness

→ can control quality of a digital image → Normalizing → Histogram → flat profile

Dark image

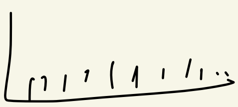


White image



Low contrast

High contrast

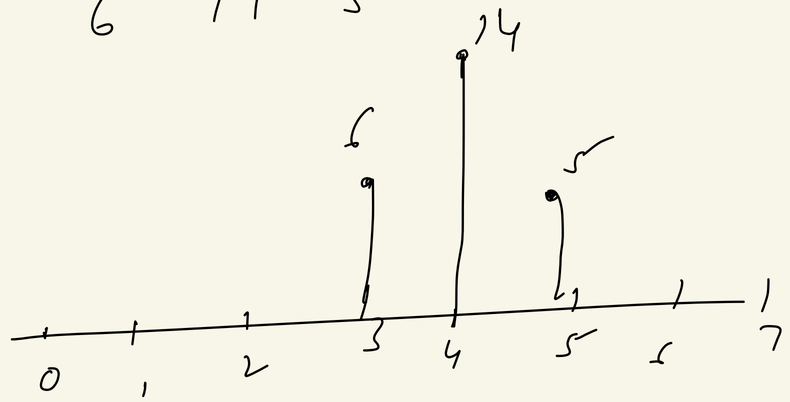


Eg:

$$\begin{bmatrix} 4 & 4 & 4 & 4 & 4 \\ 3 & 4 & 5 & 4 & 3 \\ 3 & 5 & 5 & 5 & 3 \\ 3 & 4 & 5 & 4 & 3 \\ 4 & 4 & 4 & 4 & 4 \end{bmatrix} \xrightarrow{H.E.} \begin{bmatrix} 6 & 6 & 6 & 6 & 6 \\ 2 & 6 & 7 & 6 & 2 \\ 2 & 7 & 7 & 7 & 2 \\ 2 & 6 & 7 & 6 & 2 \\ 6 & 6 & 6 & 6 & 6 \end{bmatrix}$$

I/P image

Gray levels	0	1	2	3	4	5	6	7
no of pixels n_k	0	0	0	6	14	5	0	0



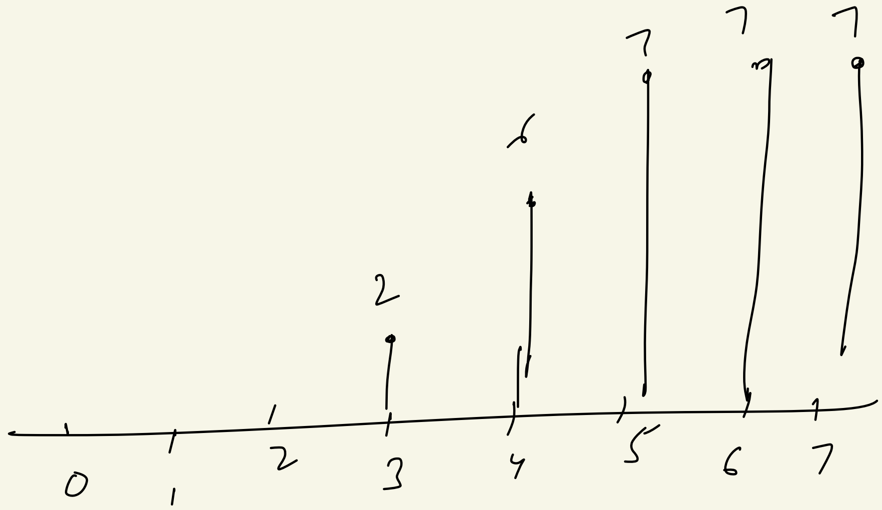
highest gray value = 5

$2^3 = 8 \rightarrow 3$ bits

[0 to 7]

Σn_k = 25

Gray level	no of pixels n_k	PDF n_k/sum	CDF = S_k (Σn_k)	$S_k \times 7$	H E
0	0	0	0	0	0
1	0	0	0+0 → 0	0	0
2	0	0	0	0	0
3	6	6/25 = 0.24	0.24	1.68	approx → 2
4	14	14/25 = 0.56	0.24+0.56 → 0.8	5.6	6
5	5	5/25 = 0.2	1.0	7	7
6	0	0	1.0	7	7
7	0	0	1.0	7	7



2
6
7
7
7
0
1
2
3
4
5
6
7

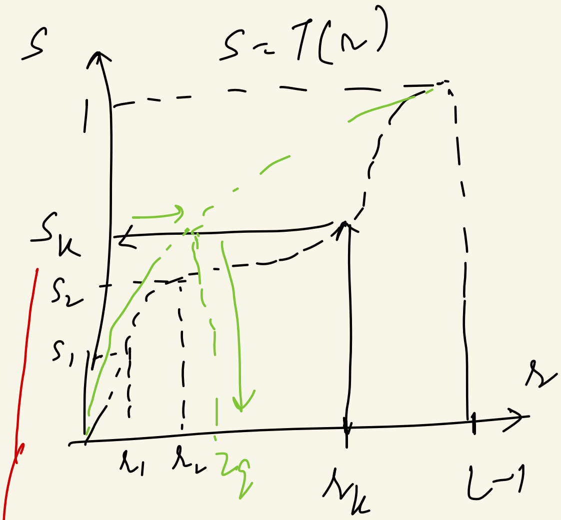
Histogram matching

→ Histogram equalization → not application for some application

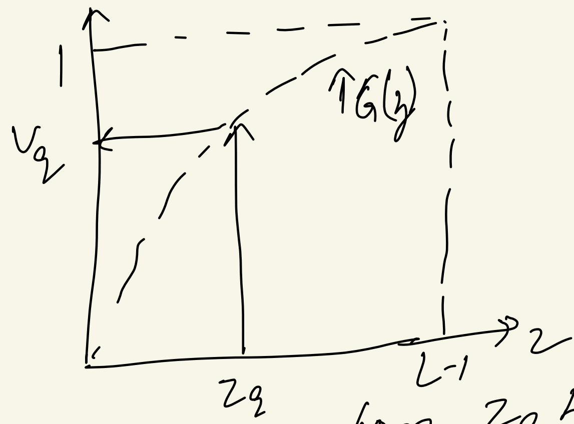
→ Histogram matching or histogram specification

Method: Generate processed image → specified histogram

$H_1 \approx H_2$



a) Mapping from r_k to s_k

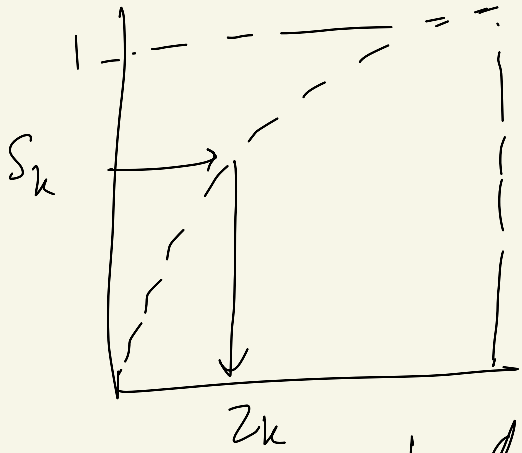


b) Mapping from z_q to v_q

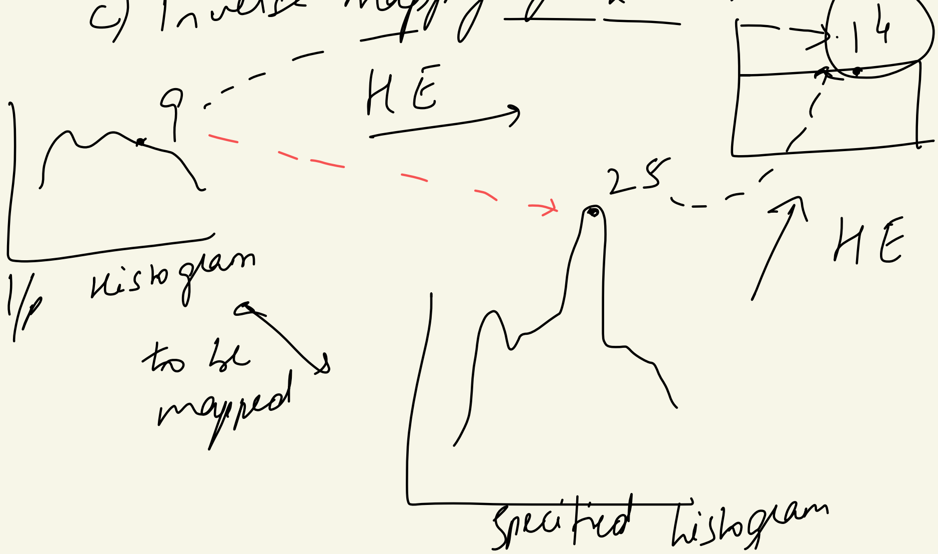
$G(z)$ → Transformation fn. obtained from PDF of r_k

If $v_q \approx s_k$

we can take inverse transformation fn. & map s_k with z_k

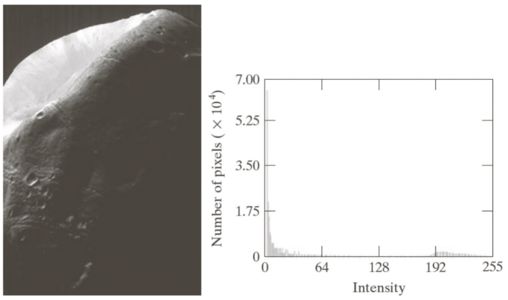


c) Inverse mapping of s_k to z_k



Take inverse & get 25

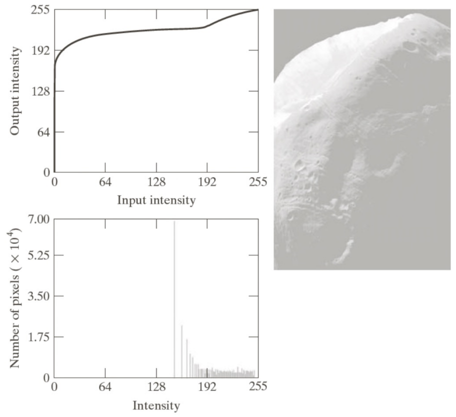
Eg:



a b

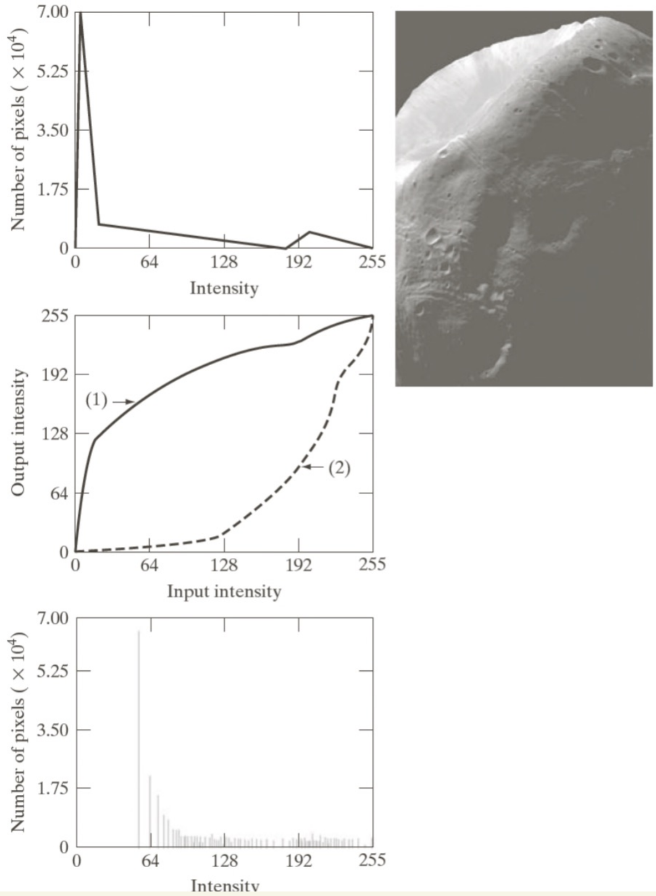
FIGURE 3.23 (a) Image of the Mars moon Phobos taken by NASA's *Mars Global Surveyor.* (b) Histogram. (Original image courtesy of NASA.)

HE



a b
c

FIGURE 3.24 (a) Transformation function for histogram equalization. (b) Histogram-equalized image (note the washed-out appearance). (c) Histogram of (b).



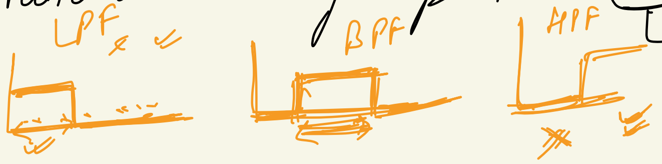
a c
b
d

FIGURE 3.25 (a) Specified histogram. (b) Transformations. (c) Enhanced image using mappings from curve (2). (d) Histogram of (c).

⇒ Histogram matching

Fundamentals of Spatial filtering → passing/rejecting some freq. components

↳ LPF : blurs an image/smoothen to pass low freq. components



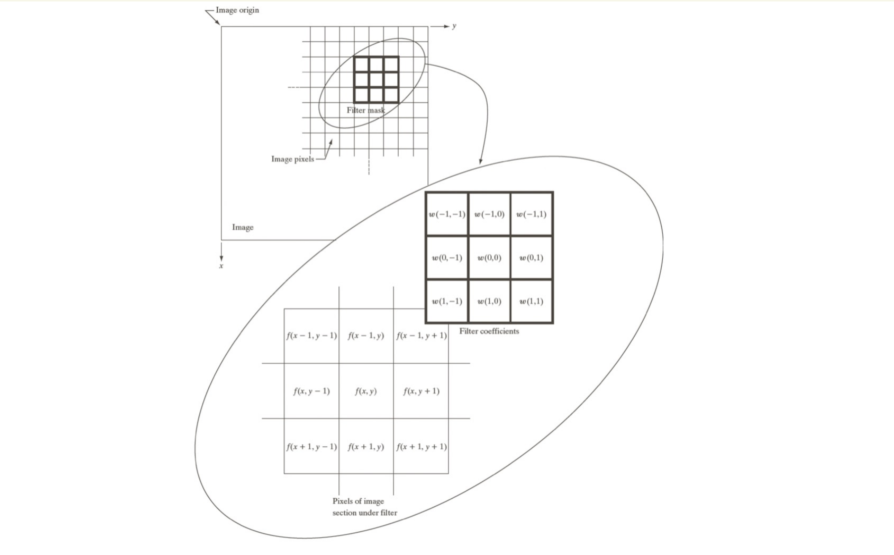


FIGURE 3.28 The mechanics of linear spatial filtering using a 3 × 3 filter mask. The form chosen to denote the coordinates of the filter mask coefficients simplifies writing expressions for linear filtering.

Response of linear filter

$$R = w(-1,-1)f(x-1,y-1) + w(-1,0)f(x-1,y) + \dots w(0,0)f(x,y) + \dots + w(1,0)f(x+1,y) + w(1,1)f(x+1,y+1)$$

↳ (1)

mask $m \times n \Rightarrow m = 2a+1$ & $n = 2b+1$ where a, b are non -ve integers

if $a=1, b=1$

min. size 3×3

Linear filtering of image of size $M \times N$ & mask size $m \times n$

$$g(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t) \quad \text{—— (2)}$$

where $a = \frac{m-1}{2}$ & $b = \frac{n-1}{2}$

eqn. (2) ⇒ Convolution mask or kernel

by simplifying (1)

$$R = w_1 z_1 + w_2 z_2 + \dots w_{mn} z_{mn}$$

w → mask coefficients

z → values of the image gray level

mn → total no of coefficients

$$R = \sum_{i=1}^{mn} w_i z_i \quad \text{— (3)}$$

for 3x3 general mask

$$R = w_1 z_1 + w_2 z_2 + \cdots w_9 z_9$$

$$R = \sum_{i=1}^{9} w_i z_i \quad \text{— (4)}$$

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

sub image

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

mask coeffs.

$f(x,y)$

0 0 0 0 0
0 0 0 0 0
0 0 1 0 0
0 0 0 0 0
0 0 0 0 0

$w =$

1 2 3
4 5 6
7 8 9

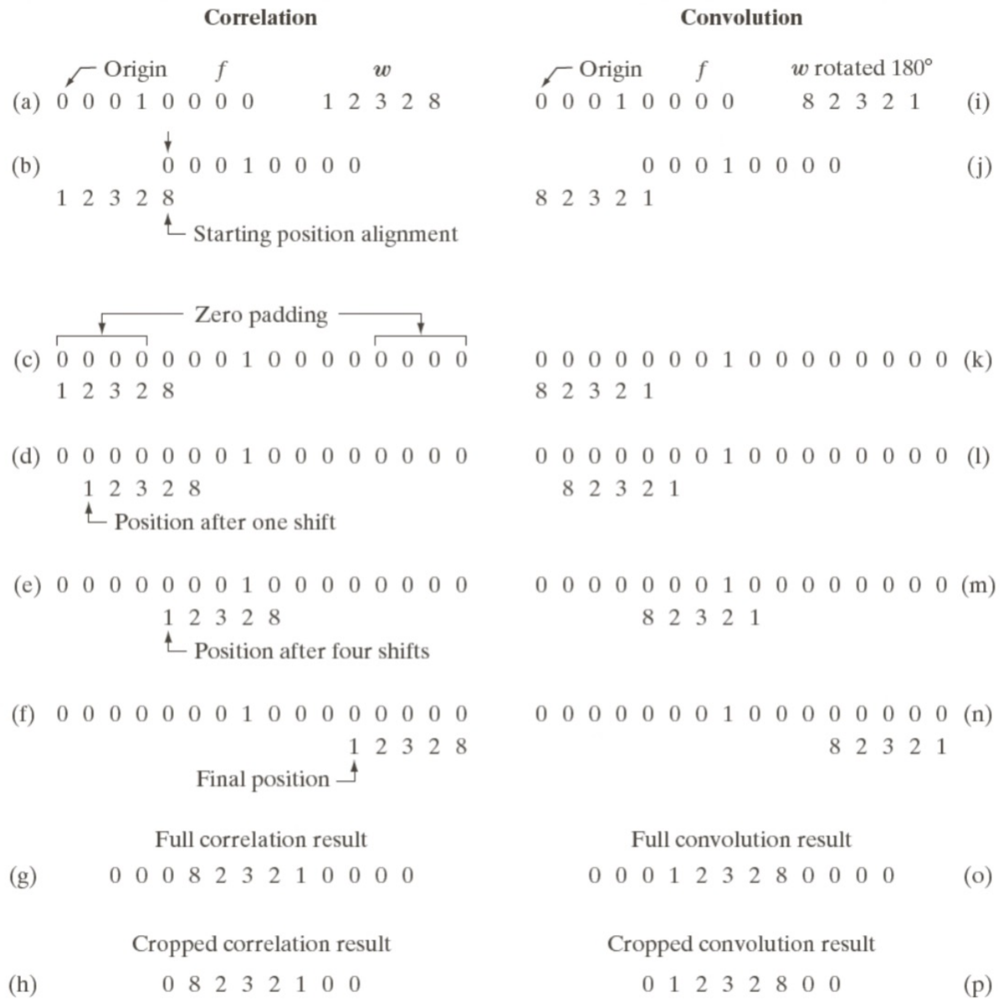


FIGURE 3.29 Illustration of 1-D correlation and convolution of a filter with a discrete unit impulse. Note that correlation and convolution are functions of *displacement*.

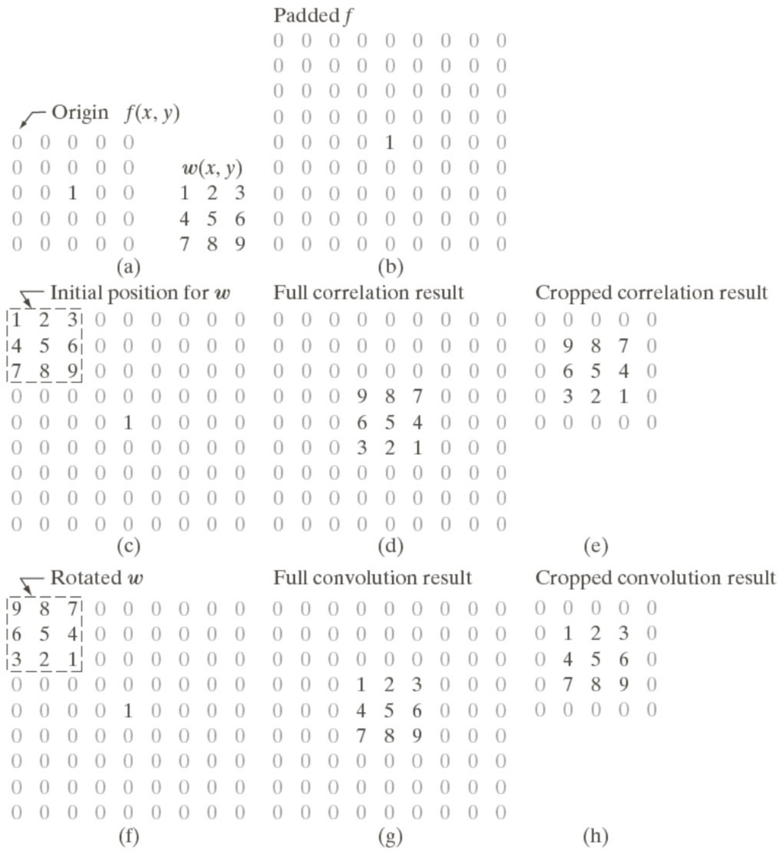


FIGURE 3.30 Correlation (middle row) and convolution (last row) of a 2-D filter with a 2-D discrete, unit impulse. The 0s are shown in gray to simplify visual analysis.

Smoothing Spatial filters :

→ Used for Blurring & Noise reduction
→ Blurring : Removal of small details from an image prior to object extraction
→ Noise reduction : Blurring with a linear or non-linear filter

<u>Smoothing linear filter</u> :

O/P : average of pixels contained in the neighborhood of filter mask

Averaging filters or LPFs

Eg: a) $\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ b) $\frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$

3x3 Smoothing filter mask

→ Replacing each pixel → by avg of gray levels
→ Application → Noise reduction
→ Side effect → Edge blurring

→ Fig a) Standard avg of pixel values

→ m×n mask → $1/mn$

→ Box filters

→ Fig b) → weighted avg

→ Pixel at the centre of mask → more importance

→ This is to reduce blurring during smoothing process

→ General implementation for image → M×N & mask m×n

$$g(x,y) = \frac{\sum_{s=-a}^{a} \sum_{t=-b}^{b} \omega(s,t) f(x+s, y+t)}{\sum_{s=-a}^{a} \sum_{t=-b}^{b} \omega(s,t)}$$

$x = 0, 1, 2 \ldots M-1$ & $y = 0, 1, 2 \ldots N-1$

Order statistics filters

→ non-linear spatial filters

→ Response → ordering [ranking] the pixels in the image

→ Replacing the centre pixel value with value determined by ranking result.

1) Median filter: → Replacing the value of pixel by the median of the gray levels

→ Most popular → excellent noise reduction

→ less blurring

→ Effective for impulse noise → Salt & pepper noise

Eg:

$$\begin{bmatrix} 10 & 20 & 20 \\ 20 & 15 & 20 \\ 20 & 25 & 100 \end{bmatrix} \Rightarrow \begin{bmatrix} 10 & 20 & 20 \\ 20 & 20 & 20 \\ 20 & 25 & 100 \end{bmatrix}$$

10, 15, 20, 20, 20, 20, 20, 25, 100

2) Max filter ⇒ finding the brightest point

$$R = \max \{ z_k \mid k = 1, 2, 3 \ldots 9 \}$$

3) Min filter ⇒ finding the darkest point

$$R = \min \{ z_k \mid k = 1, 2, 3 \ldots 9 \}$$

Eg: max value : 100 (brightest point)

min value : 10 (darkest point)

Sharpening spatial filters:

→ highlight the fine details or to enhance details which might have been blurred

→ Applications ⇒ Electronic printing, medical imaging, industrial inspections & autonomous guidance in military systems

→ Image blurring → pixel averaging

↳ Integration

→ Sharpening → "Spatial differentiation"

→ Image differentiation → enhances edges and noise & deemphasizes areas with slowly varying gray level values.

foundation :

→ first order & second order derivatives

→ Derivatives → defined in terms of differences

→ Definition for first derivative

a) must be zero in flat segments

b) must be non zero at onset of a gray level step or ramp

c) must be non zero along ramp

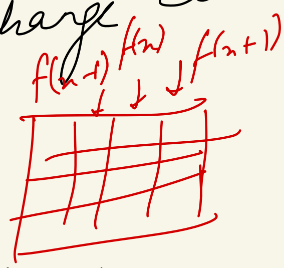
ly for second derivatives :

a) must be zero in flat area.

b) must be non zero at the onset & end of gray level step or ramp

c) must be zero along ramp of constant slope

→ The shortest distance over which change can occur is b/w adjacent pixels



The basic definition

1st order derivative : $\frac{\partial f}{\partial x} = f(x+1) - f(x)$

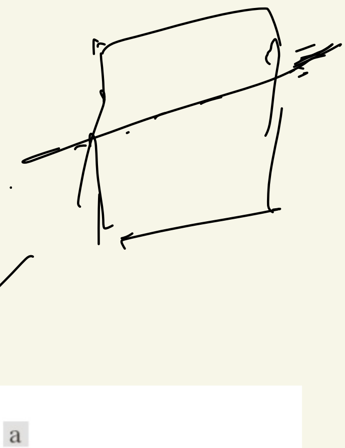
2nd order derivative : $\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$

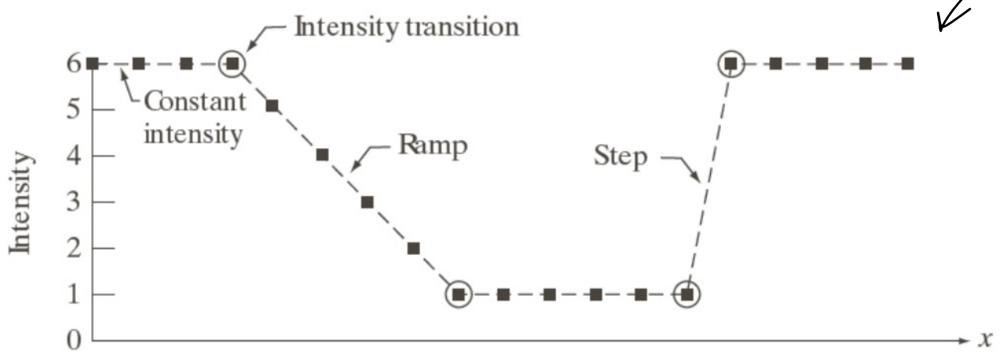
Conditions
For ramp:
1st → thicker edges
2nd → fine details or isolated points

noise

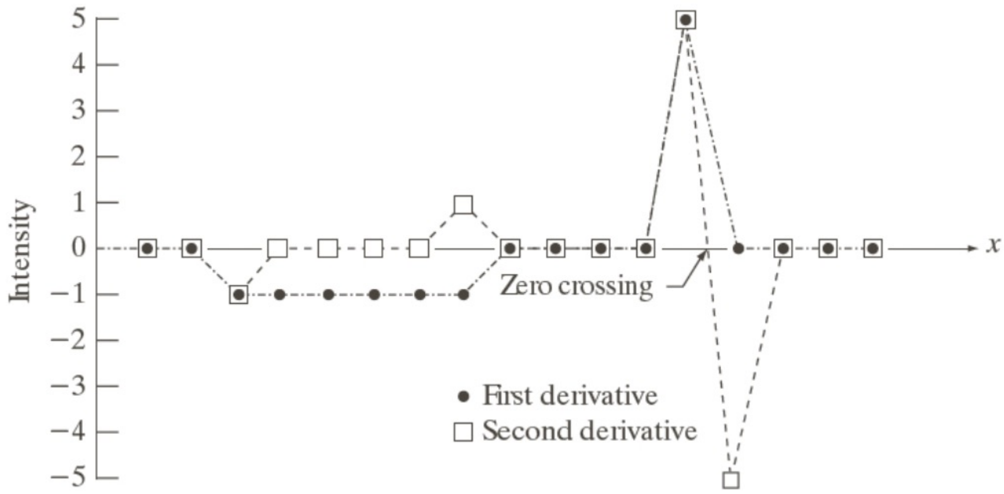
Isolated pt:
2nd → enhances the

Step:
1st → stronger response to gray level step
2nd → double response at step change





Scan line	6	6	6	6	5	4	3	2	1	1	1	1	1	1	6	6	6	6	6 → x
1st derivative		0	0	−1	−1	−1	−1	−1	0	0	0	0	0	5	0	0	0	0	
2nd derivative		0	0	−1	0	0	0	0	1	0	0	0	0	5	−5	0	0	0	



a
b
c

FIGURE 3.36 Illustration of the first and second derivatives of a 1-D digital function representing a section of a horizontal intensity profile from an image. In (a) and (c) data points are joined by dashed lines as a visualization aid.

Use of second order derivatives for enhancement → Laplacian

Laplacian filter

Defn: $$\Delta^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

In x direction

$$\frac{\partial^2 f}{\partial x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y)$$

In y-direction

$$\frac{\partial^2 f}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y)$$

$$\therefore \Delta^2 f = \left[f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) \right] - 4f(x, y)$$

$f(x-1, y-1)$	$f(x, y-1)$	$f(x+1, y-1)$
$f(x-1, y)$	$f(x, y)$	$f(x+1, y)$
$f(x-1, y+1)$	$f(x, y+1)$	$f(x+1, y+1)$

0	1	0
1	-4	1
0	1	0

⇐ Mask

Different Laplacian filters

→ central pixel less highlighted

0	1	0
1	-4	1
0	1	0

→ central pixel less highlighted

1	1	1
1	-8	1
1	1	1

→ more highlighted

0	-1	0
-1	4	-1
0	-1	0

-1	-1	-1
-1	8	-1
-1	-1	-1