INTENSITY TRANSFORMATION AND SPATIAL FILTERING

Lecture 3

Image Domains

- Spatial domain
  - Refers to the image plane itself
  - Image processing methods are based and directly applied to image pixels
- Transform domain
  - Transforming an image into a transform domain, doing the processing there and obtaining the results back into the spatial domain

Spatial domain

- Two principle categories of spatial processing:
  - Intensity transformation
  - Spatial filtering
- Intensity transformation operate on single pixels of an image
  - Contrast manipulation
  - Image thresholding

Spatial domain

- Spatial filtering deals with operations :
  - Image sharpening -> working in a neighbourhood of every pixel in an image
- "Classical" techniques of intensity transformations and spatial filtering
- Fuzzy techniques -> incorporate imprecise, knowledge based information in the formulation of intensity transformation and spatial filtering

Basics of intensity transformation

- Spatial domain techniques are applied on pixels
- Frequency domain are performed on Fourier transform of an image

- Some application requires spatial domain techniques other rely on frequency domain approaches.

Transformation

$$G(j,k) = T\left[F(j,k)\right]$$

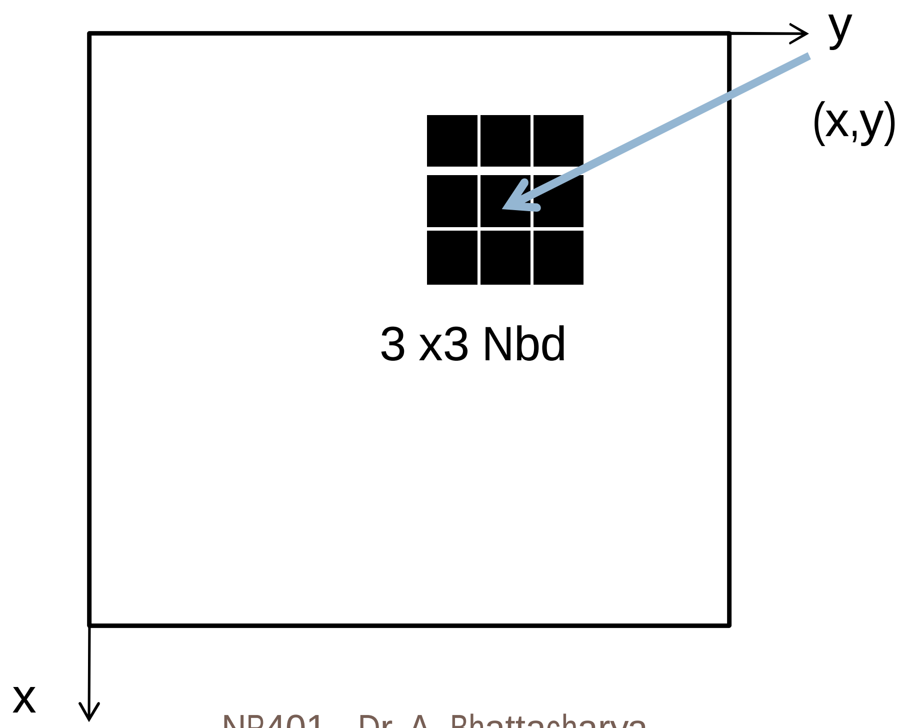
$F(j,k)$ = Input image

$G(j,k)$ = Output image

T = Operator defined over a neighbourhood

Transformation

- The operator can be applied to a single image or to a set of images over pixel-by-pixel



Transformation

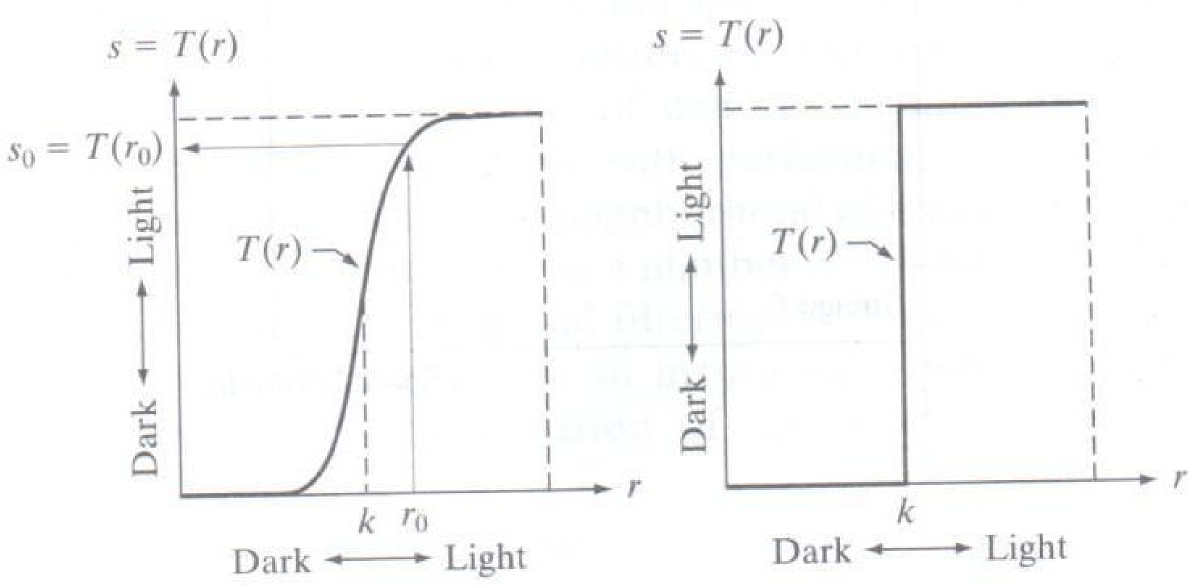
- Process of small window operation are called spatial filtering
- The type of operation performed in the neighbourhood determines the filtering process
- The smallest possible neighnourhood is 1x1
  - G depends on the value of F at a single point (x,y)
  - Intensity transformation function

Transformation

$$s = T(r)$$

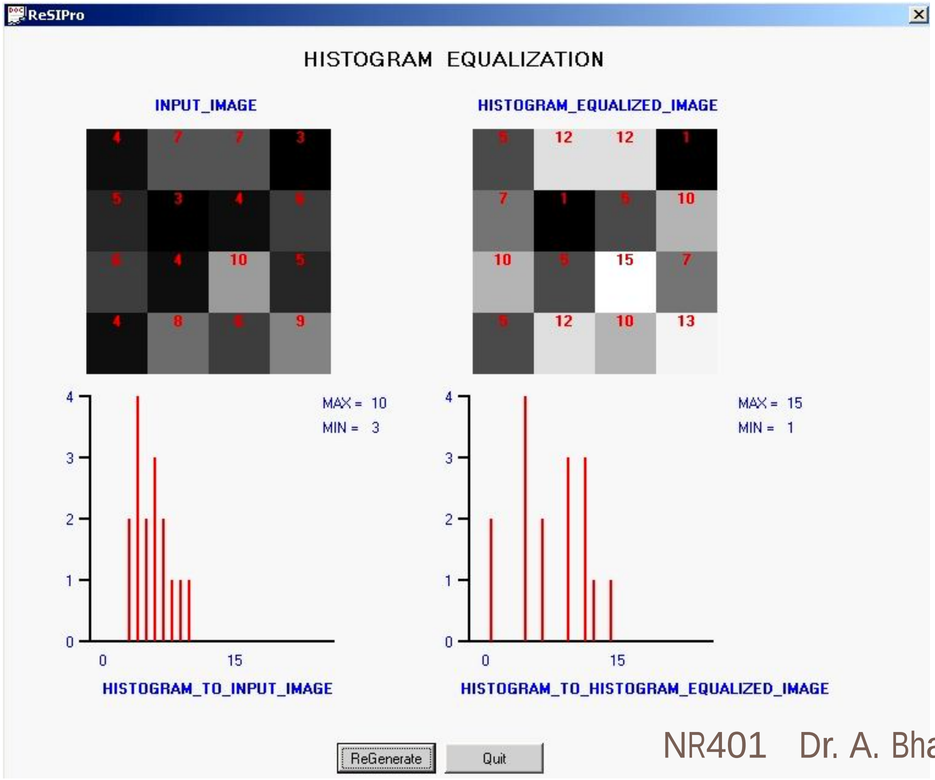
Intensity of G = s

Intensity of F = r



Transformation

- Intensity transformation are among the simplest of all image processing techniques.



Linear Contrast Enhancement

- Linear Contrast Enhancement
  - Identify minimum and maximum gray levels in the input
  - Specify minimum and maximum gray levels in the output image
  - Compute the gray level mapping based on the line

$$y = m.x + c$$

Logarithmic Contrast Stretch

- Logarithmic Contrast Stretch
  - Identify minimum and maximum gray levels in the input
  - Apply the transformation **$y = k.\log(1+x)$**
  - k is user-specified parameter.
  - The resulting floating point values of y are scaled linearly to the desired output range as before

Exponential Contrast Stretch

- Exponential Contrast Stretch
  - Identify the input minimum and maximum gray levels
  - Apply the transformation $y = kx^r$
  - The parameter r controls the rate at which x^r rises
  - The resulting values of **y** are rescaled to the desired output range using linear stretching

Contrast

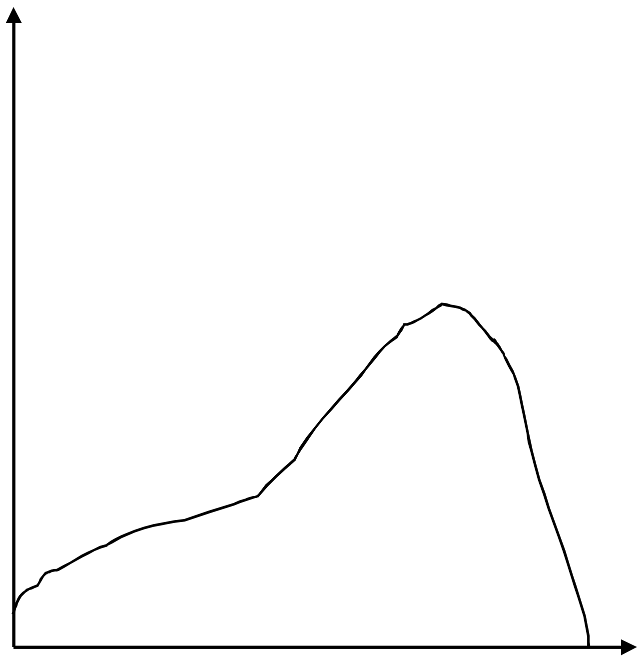
- Contrast is the *difference* in the intensity of the object of interest compared to the background (rest of the image)

- The perceptual contrast is a function of the logarithm of the difference in the object and background intensities
  - This means that in the darker regions, small changes in intensity can be noticed, but in brighter regions, the difference has to be much more

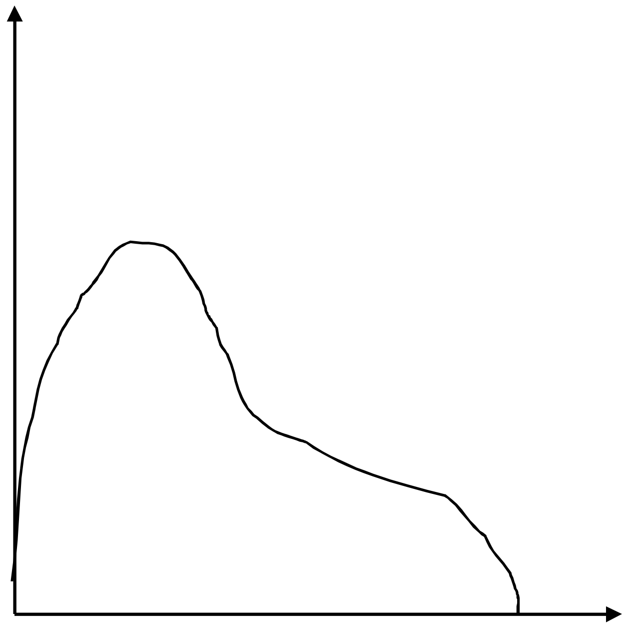
Histogram

- Given a 1-D histogram (computed for a black/white image or for one band in a multispectral image), it conveys information about the quality of the image.
- Positively skewed histogram – darkish image
- Negatively skewed histogram – lightish image

Histogram



Negatively skewed histogram



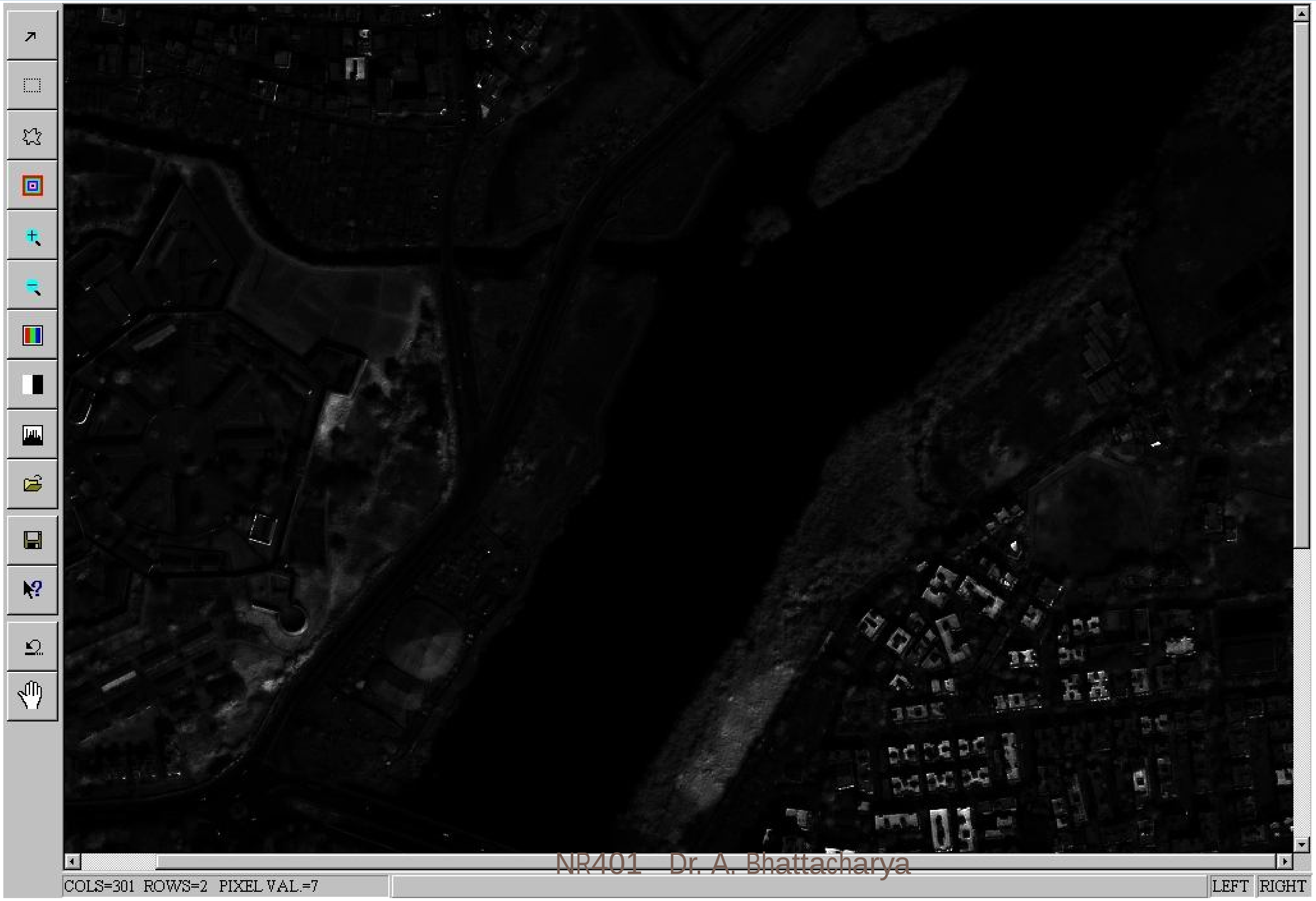
Positively skewed histogram

Example image

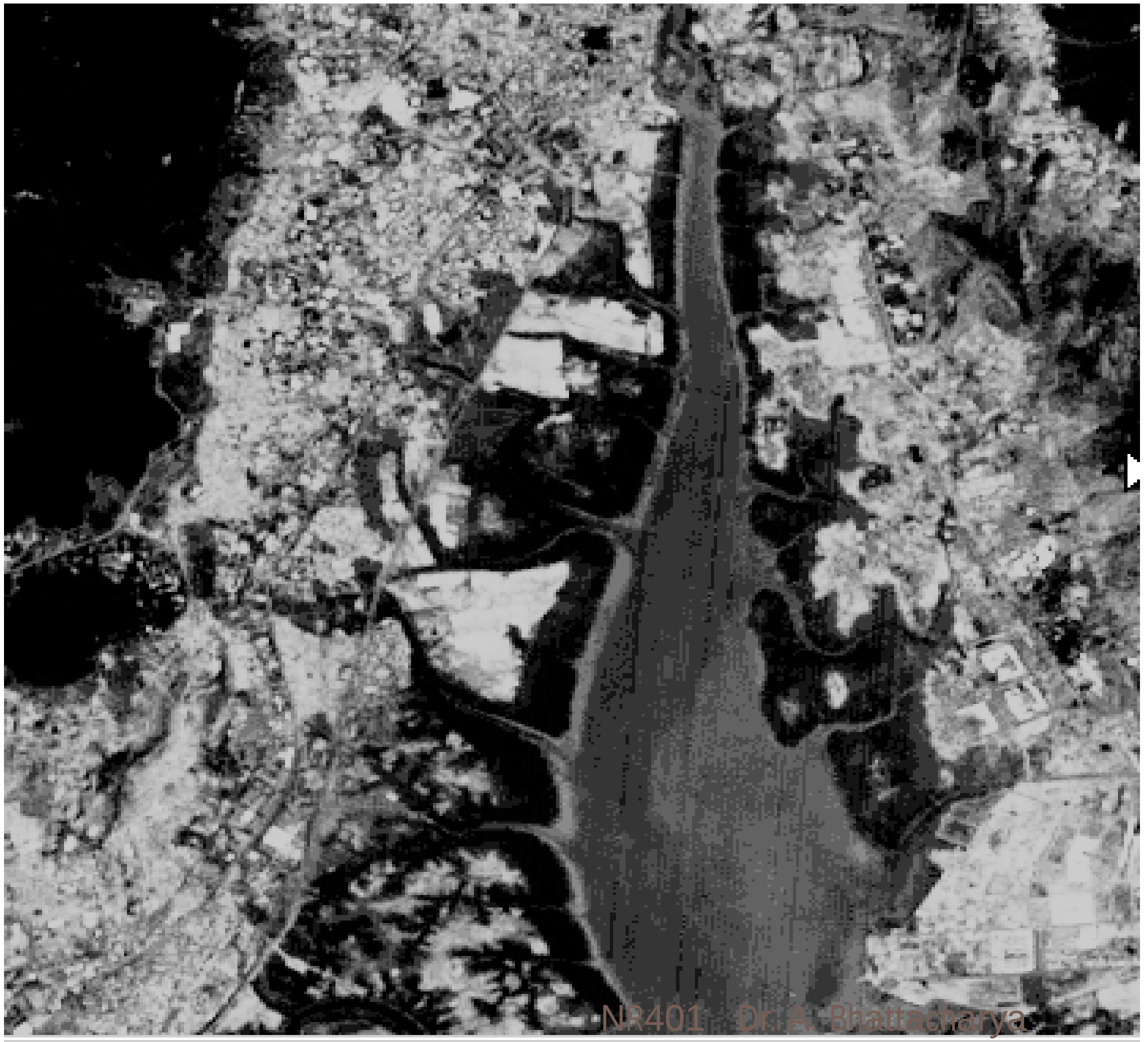
After Logarithmic stretch

Example image

After exponential stretch



Example image 2



After Logarithmic stretch

Histogram Equalization

- When the image contains very few similar valued gray levels, then the ability to interpret it is hampered. It is desirable that the dynamic range of the display device is better utilized.
- One way to achieve this is by transforming the image such that all gray levels have equal likelihood of occurrence.

Principle of Histogram Equalization

Given an imperfect histogram, and an ideal histogram that has equal population of all gray levels, map the input histogram to approximate the "equalized" histogram.

Translation of Theory into Practice

- Essentially, the enhanced image which has an equalized histogram has (ideally) equal number of pixels at each gray level. In practice, we can only achieve an approximation of it.
- For an equalized histogram, the cumulative histogram is known given the size of the image and the number of gray levels.

How to equalize the histogram?

- For a gray level, corresponding to its cumulative frequency, find the nearest gray level that matches the ideal cumulative frequency
- **Therefore image enhancement by histogram equalization is achieved by the mapping of gray levels is based on the actual cumulative histogram of the image and the desired cumulative histogram**

Local histogram processing

- In the global sense :
  - Pixels are modified by a transformation function based on the intensity distribution of the entire region
- In the local sense :
  - It is sometime necessary to enhance the details over small areas in an image

Histogram statistics

- Statistics obtained directly from an image histogram can be used for image enhancement.
  - nth moment of r about its mean
  - Second moment

- Global mean and variance computed over the image are useful for gross adjustments in overall intensity
- Local mean and variance over a neighbour of a pixels -> S_{xy}

Histogram

- The normalized version of f(n) may be defined as

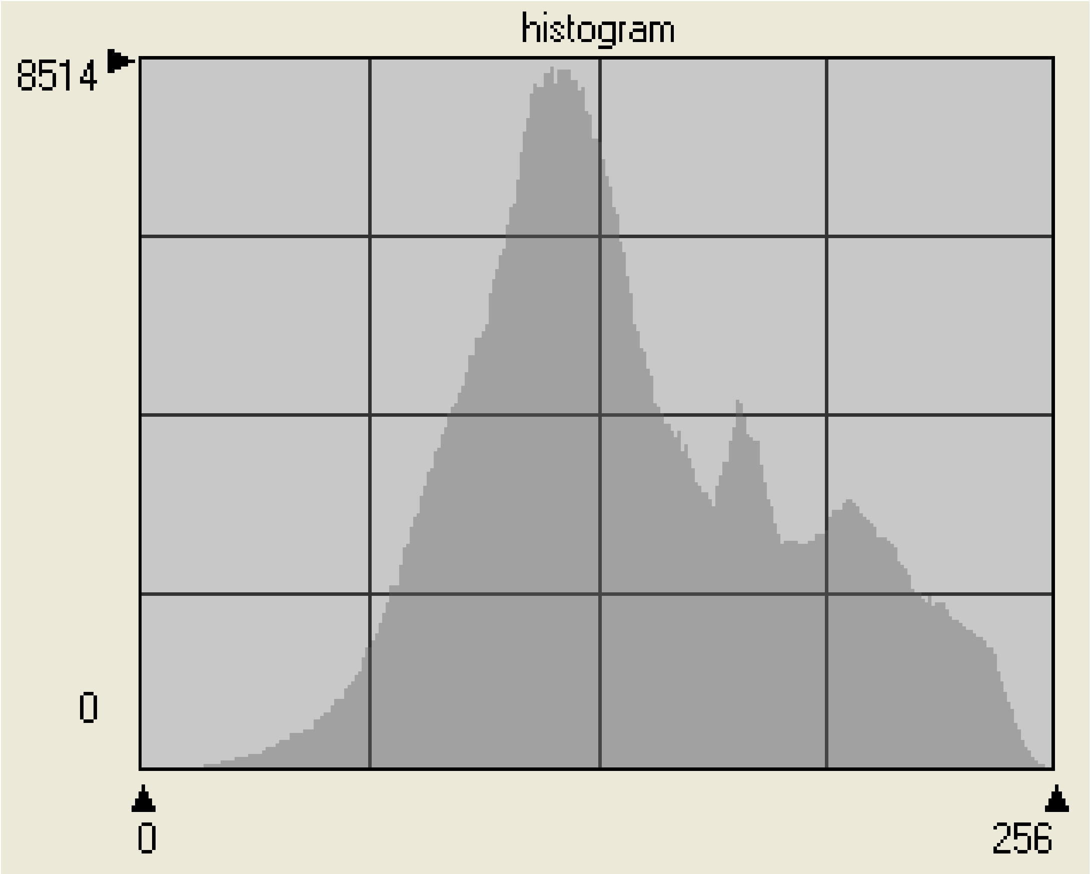
$$p(n) = f(n) / (M.N)$$

  - p(n) ➔ probability of the occurrence of gray level n in the image (in relative freq. sense)
  - $$\Sigma_n \; p(n) = 1$$

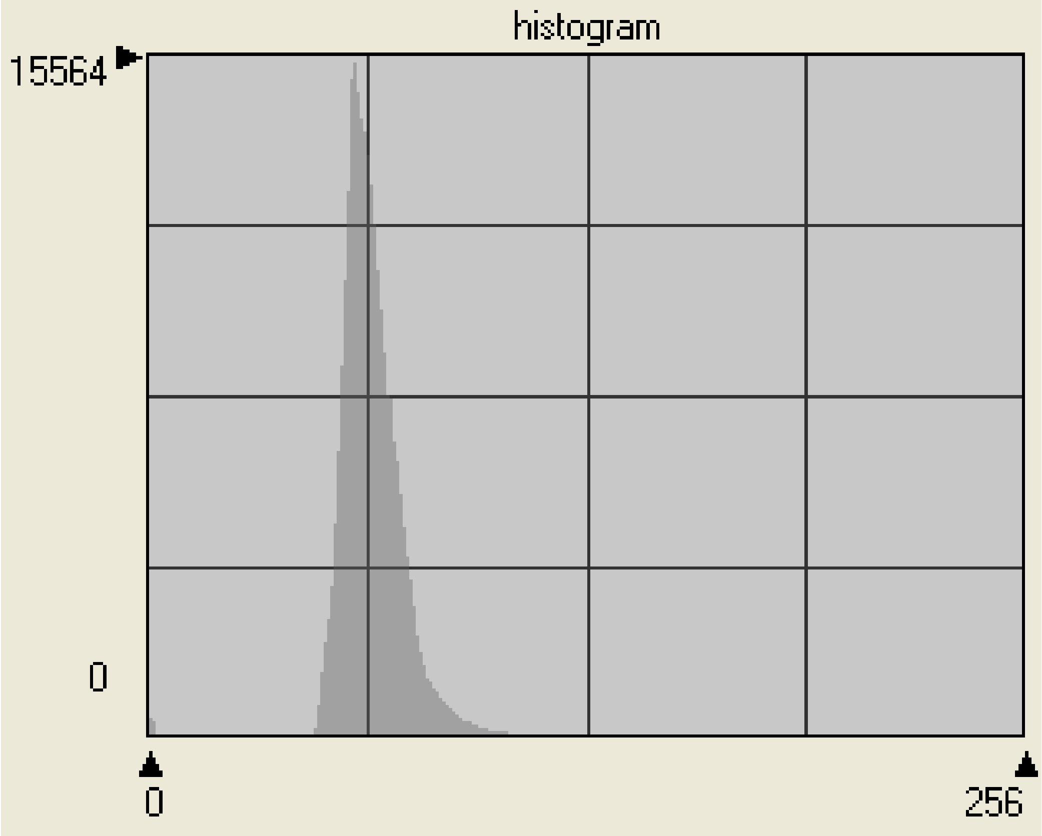
$$\textbf{MIN} = \min_n \{f(n) \mid f(n) \neq 0\}$$

$$\textbf{MAX} = \max_n \{f(n) \mid f(n) \neq 0\}$$

Histogram of image with good contrast



Histogram of Low Contrast Image



Information from Histogram

- The *information* conveyed by the occurrence of an event whose probability of occurrence is p(n) is given by

$$I(n) = \ln\{1/p(n)\} = -\ln\{p(n)\}$$

- This implies that *if the probability of occurrence of an event is low, then its occurrence conveys significant amount of information*

- *If the probability of an event is high, the information conveyed by its occurrence is low*

Average Information – Entropy

- Average information conveyed by a set of events with probabilities p(i), i=1,2,…, is given by

$$H = - \Sigma_n p(n) \log \{p(n)\}$$

- H is called *entropy* and is extensively used in image processing operations
- H is highest when all probabilities are equally likely.

$$H_{max} = -\Sigma_n k.\ln(k), \text{ where } k = p(n) \text{ for all } n$$

- H is zero when p(j)=1 for some j, and p(k) = 0 for all $k \neq j$

Role of Entropy

- Indicator whether very few gray levels are *actually present*, or wide range of levels in sufficient numbers
-
- Entropy is also used for threshold selection
- e.g., separating image into object of interest and background

Histogram

A Bimodal Histogram

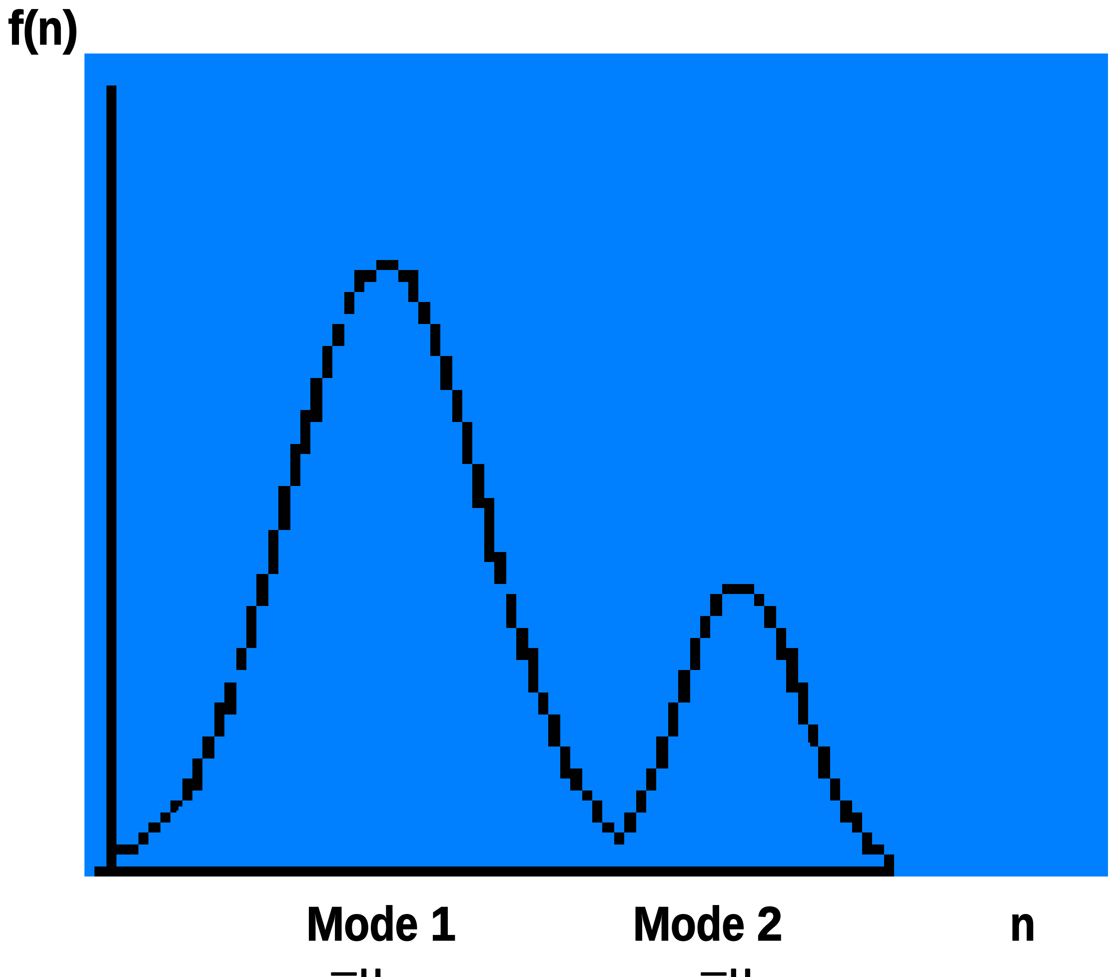


Image Statistics from Histogram

- **MIN** gray level MIN = n: $\min_n f(n) \neq 0$
- **Max** gray level MAX = n: $\max_n f(n) \neq 0$
- **Mean** gray level

$$\mu = \Sigma_n n.f(n) / (M.N)$$

- **Variance**

$$\sigma^2 = \Sigma_n f(n)[n-\mu]^2 / (M.N)$$

- **Median**

$$Med = k: \Sigma^k_{n=0} f(n) = (M.N)/2$$

Image Statistics from Histogram

- **Skewness**

$$Sk = \frac{1}{(MN-1)\sigma^3}\sum (n-\mu)^3 f(n)$$

Skewness is positive if the histogram is skewed to the left of the mean, i.e., it has a long tail towards the higher gray levels

Skewness is negative if the histogram is skewed to the right of the mean

Image Statistics from Histogram

- **Kurtosis** $$Ku = \frac{1}{(MN-1)\sigma^4}\sum (n-\mu)^4 f(n) - 3$$

- For Gaussian distributions, Kurtosis = 3. Therefore the excess kurtosis is defined by subtracting 3 from the above equation. Positive kurtosis in this case indicates a sharply peaked distribution, and negative kurtosis denotes a flat distribution, with *uniform* distribution being the limiting case.

Pixel and Neighborhood

A	B	C
D	X	E
F	G	H

- Pixel under consideration X
- Neighbors of X are A, C, F,H, B,D,E,G
- Size of neighborhood = 3x3
- Neighborhoods of size mxn m and n are odd; Unique pixel at the centre of the neighborhood

Point Operations v/s Neighborhood Operations

- Point operations do not alter the sharpness or resolution of the image
- Gray level associated with a pixel is manipulated independent of the gray levels associated with neighbors
- Pixel operations cannot deal with noise in the image, nor highlight local features like object boundaries

Neighborhood Operations

- Simple averaging

A B C

D X E

F G H

- g(X) = (1/9)[f(A) + f(B) + f(C) + f(D) + f(X) + f(E) + f(F) + f(G) + f(H)]
- The output gray level is the average of the gray levels of all the pixels in the 3x3 neighborhood

Example

15	17	16		15	17	16
18	***17***	15		18	***37***	15
17	14	16		17	14	16
Case 1				Case 2		

- **In case 1, after averaging, the central element 17 is replaced by the local average 16 – negligible change**
- **In case 2, after averaging, the central element 37 is replaced by 18 – significant change**
- **Averaging is a powerful tool to deal with random noise**

Neighborhood Operations - Procedure

- The procedure involves applying the computational step at every pixel, considering its value and the values at the neighboring pixels
- Then the neighborhood is shifted by one pixel to the right and the centre pixel of the new neighborhood is in focus
- This process continues from left to right, top to bottom

Mathematical form for averaging

- In general, we can write

$$g(X) = \frac{\sum_{i=1}^{K} f(A_i)}{|N(X)|}$$

where K is the number of neighbors A_i. A_5 refers to X, the central pixel for a 3x3 neighborhood.

- It is obvious that all neighbors are given equal weightage during the averaging process

General form for averaging

- In case different weights are preferred for different neighbors, then we can write

$$g(X) = \frac{\sum_{i=1}^{K} w_i f(A_i)}{\sum_{i=1}^{K} w_i}$$

- For simple averaging over a 3x3 neighborhood, w_i = (1/9), i=1,2,…,9
- We can alter, for example, the weights for 4-neighbors and 8-neighbors. In such a case, w_i is not a constant for all values of i.

Concept of Convolution

- **Convolution is a weighted summation of inputs to produce an output; weights do not change anytime during the processing of the entire data**
- **If the input shifts in time or position, the output also shifts in time or position; *character* of the processing operation will not change**
- **The weights with which the pixels in the image are modified are represented by the term *filter***

Filter Mask

- The filter can be compactly represented using the weights or multiplying coefficients:
- e.g., *3x3 averaging filter*

$$\begin{matrix} 0.111 & 0.111 & 0.111 \\ 0.111 & 0.111 & 0.111 \\ 0.111 & 0.111 & 0.111 \end{matrix} \quad \text{or} \quad (1/9)\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

- This implies that the pixels in the image are multiplied with corresponding filter coefficients and the products are added

Reduced neighborhood influence

0.05	0.15	0.05
0.15	0.20	0.15
0.05	0.15	0.05

- Central pixel is given 20% weight, 4-neighbors 15% weight. Diagonal neighbors given 5% weight.
- Note that the weights are all positive, and sum to unity

Discrete Convolution

- $g_{i,j} = \sum_{k=-w}^{w} \sum_{l=-w}^{w} h_{k,l}\, f_{i-k,\, j-l}$
- The filter coefficients are mirror-reflected around the central element, and then the filter is slid on the input image
- The filter moves from top left to bottom right, moving one position at a time
- For each position of the filter, an output value is computed