# *Chapter-3: Image Manipulations*

(1-Resizing, 2-Point operations and 3-Histogram Transformation)

Dr. Mohamed Waleed Fakhr

2023

# 1-Image Resizing

- Shrinking (down-sampling, under-sampling, decimation)
- Enlarging (up-sampling, over-sampling, interpolation)

# Zooming and shrinking digital images

original

resampling

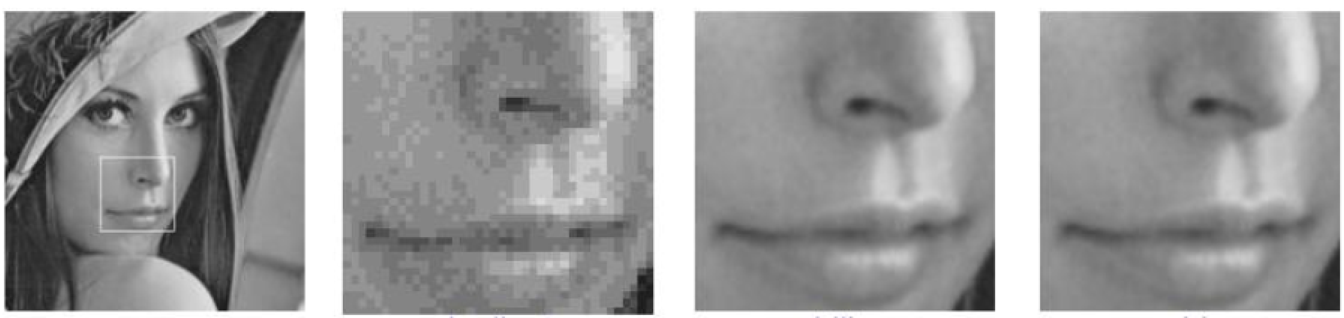
shrinking

zooming

# Zooming and shrinking digital images

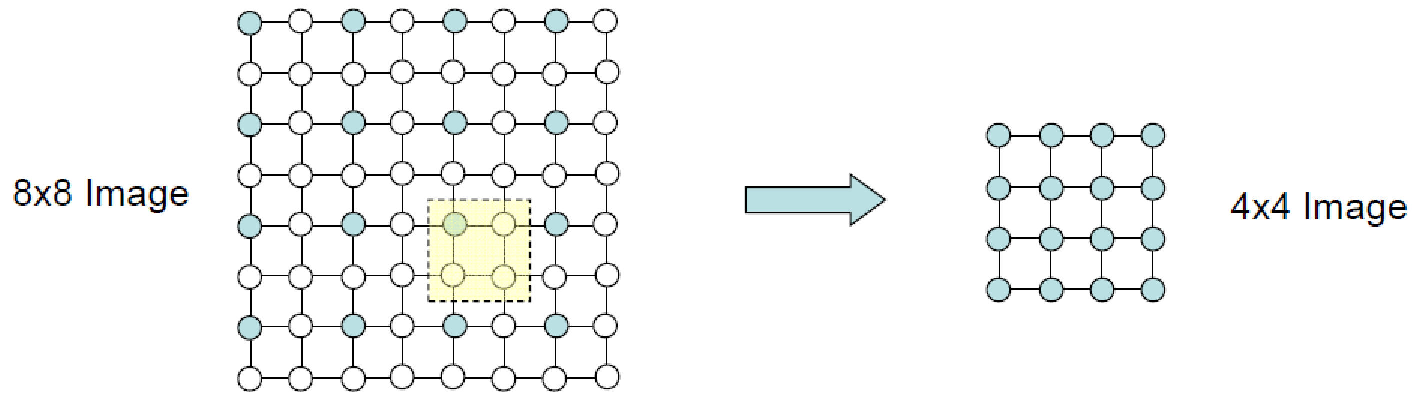
- Shrinking : undersampling
- Zooming : oversampling
  - The creation of new pixel locations
  - The assignment of gray levels to those new locations. □
- Methods □
  - Duplication □
  - Bilinear interpolation □
  - Cubic interpolation

4X zooming



duplicate bilinear cubic

# Down Sampling by a Factor of Two




- Without Pre-filtering (simple approach)

$$f_d(m,n) = f(2m,2n)$$

- Averaging Filter

$$f_d(m,n) = [f(2m,2n) + f(2m,2n+1) + f(2m+1,2n) + f(2m+1,2n+1)]/4$$

# Problem of Simple Approach

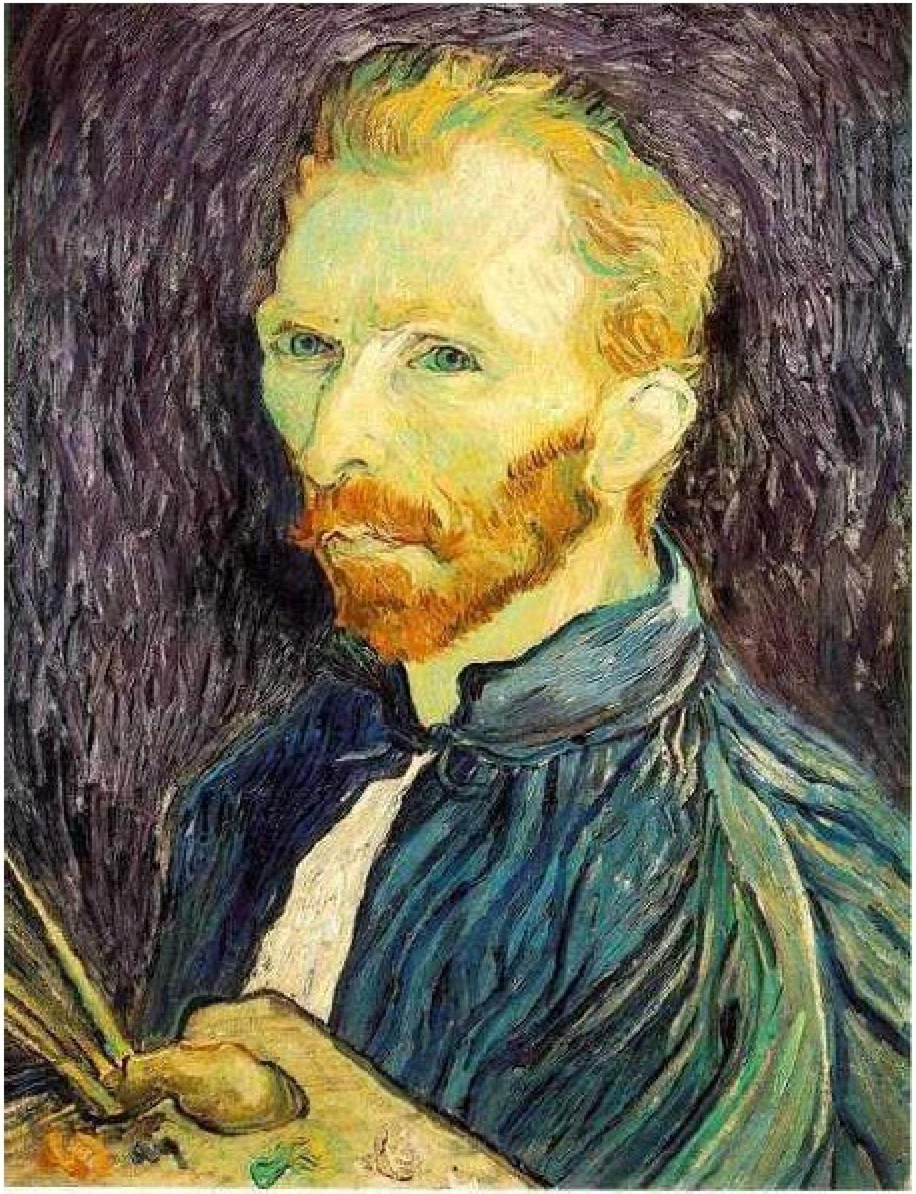
- Aliasing if the effective sampling rate is below the Nyquist sample rate = 2 * highest frequency in the original continuous signal
- We need to prefilter the signal before down-sampling
- Ideally the prefilter should be a low-pass filter with a cut-off frequency half of the new sampling rate.
  - In digital frequency of the original sampled image, the cutoff frequency is 1/2
- In practice, we may use simple averaging filter

# Image Scaling

- Need to resample images

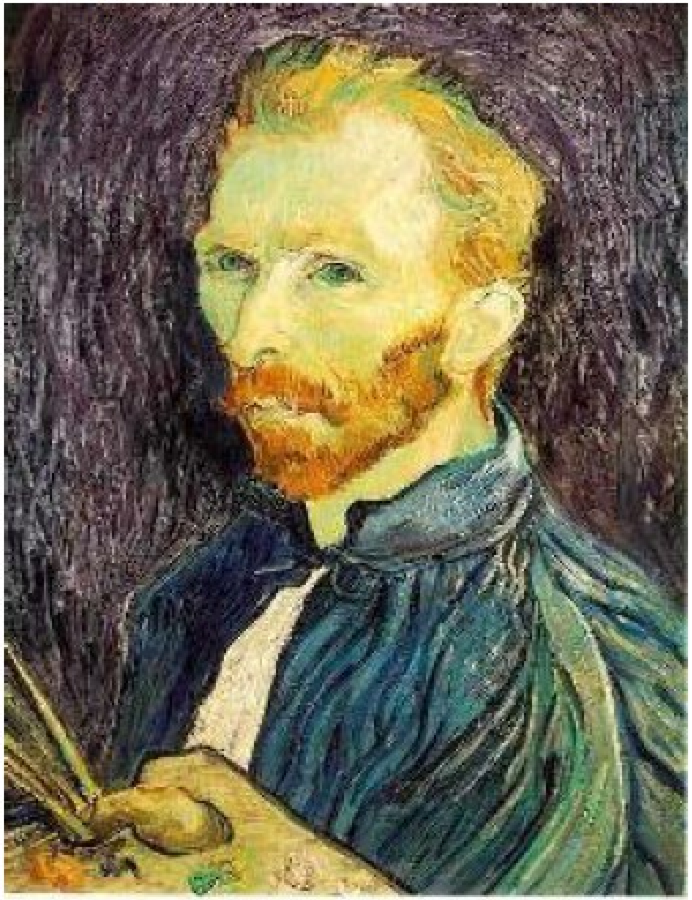
This image is too big to fit on the screen. How can we reduce it?

How to generate a half-sized version?

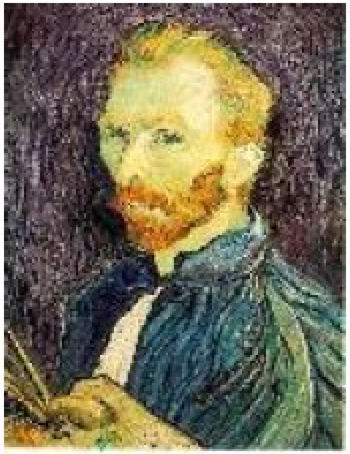


# Image sub-sampling

- sub sample without filtering, what is wrong ?



Throw away every other row and column to create a 1/2 size image
- called *image sub-sampling*

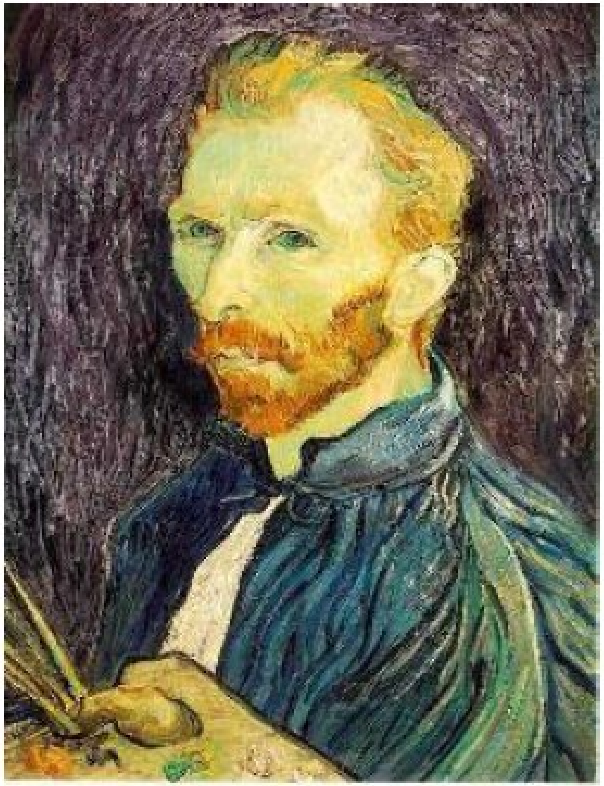


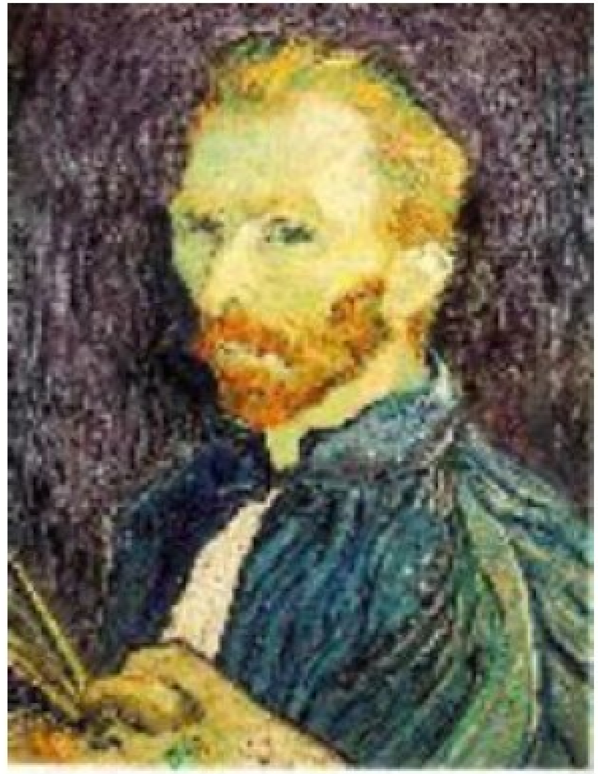
1/4

1/8

# Image sub-sampling

- Aliasing...





1/2

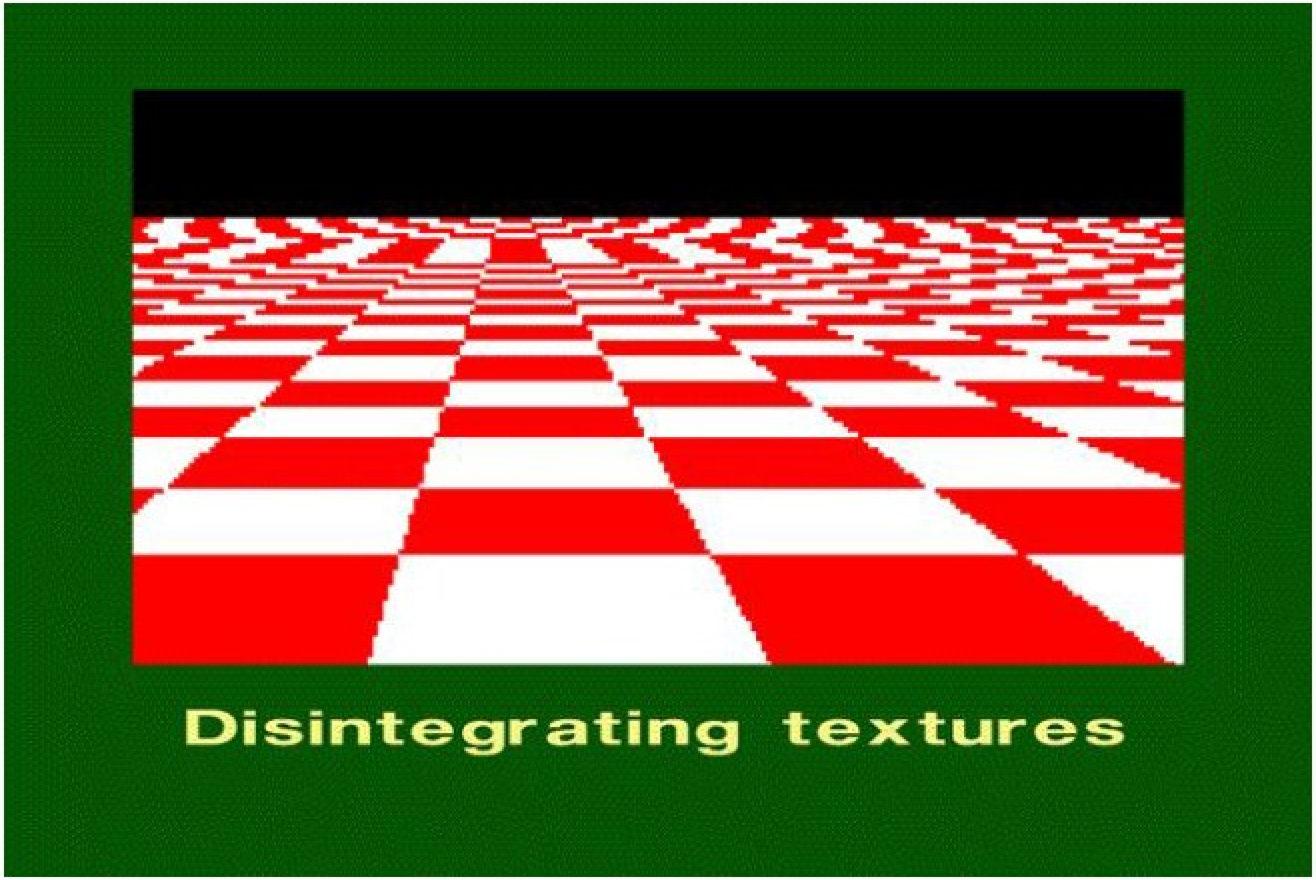
1/4 (2x zoom)

1/8 (4x zoom)

Why does this look so crufty?

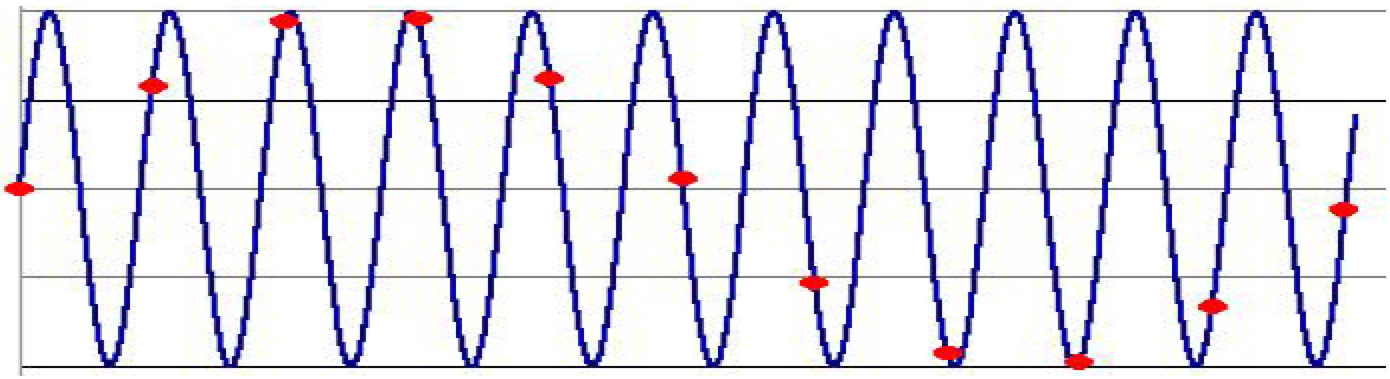
# Even worse for synthetic images

- Aliasing effect



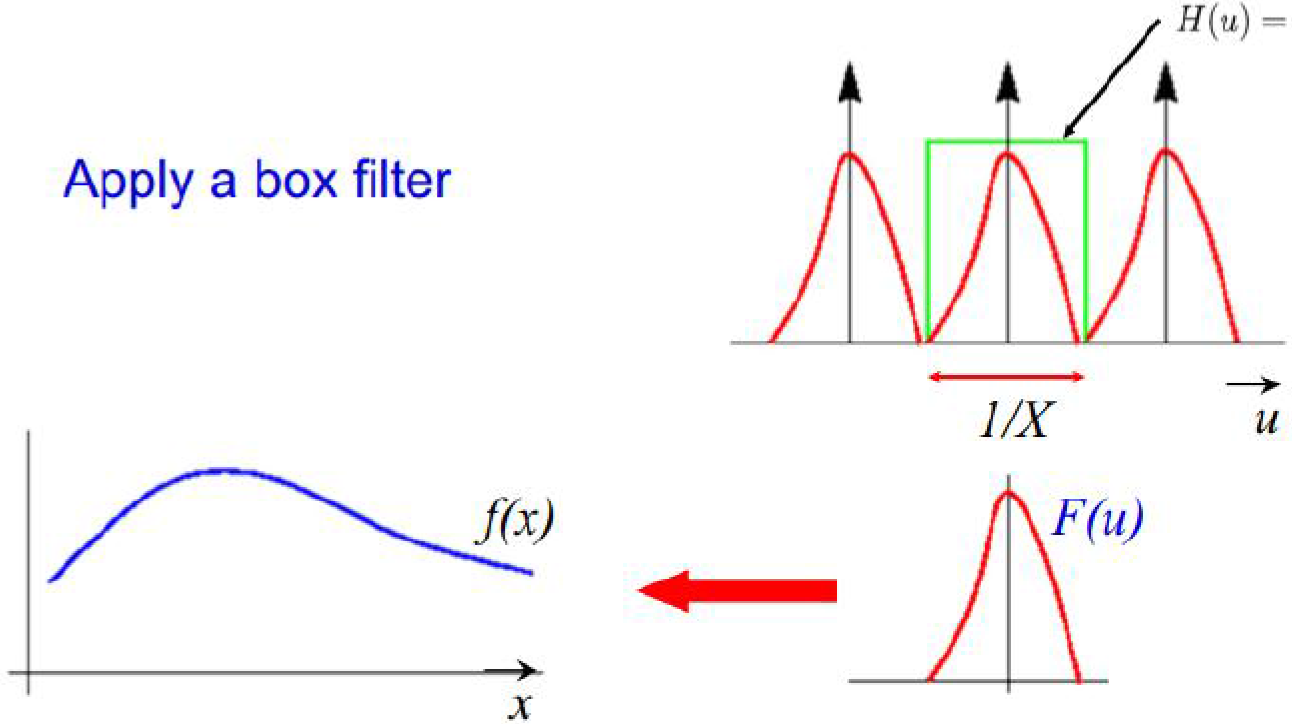
# Sampling and the Nyquist rate

- **Aliasing** can arise when you sample a continuous signal or image
  - occurs when your sampling rate is not high enough to capture the amount of detail in your image
  - Can give you the wrong signal/image—an *alias*
  - formally, the image contains structure at different scales
    - called “frequencies” in the Fourier domain
  - the sampling rate must be high enough to capture the highest frequency in the image
- To avoid aliasing:
  - sampling rate > 2 * max frequency in the image
  - This minimum sampling rate is called the **Nyquist rate**



# Reconstruction

- Recon via convolving with Sinc function (low pass filtering in freq domain)




$F(u) = F_s(u)H(u)$

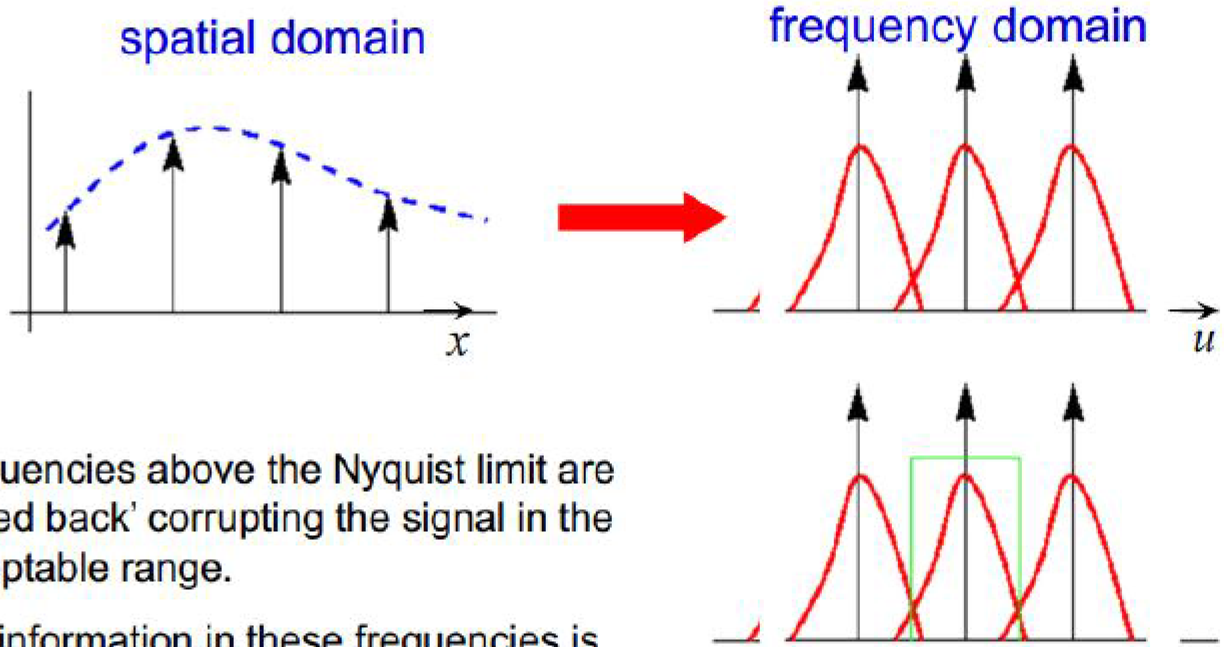
$$f(x) = \sum_{n=-\infty}^{\infty} f(nX)\delta(x - nX) * \text{sinc}\frac{\pi x}{X}$$

$$= \sum_{n=-\infty}^{\infty} f(nX)\text{sinc}\frac{\pi}{X}(x - nX)$$

# Aliasing from under-sampled reconstruction

- Aliasing effect:

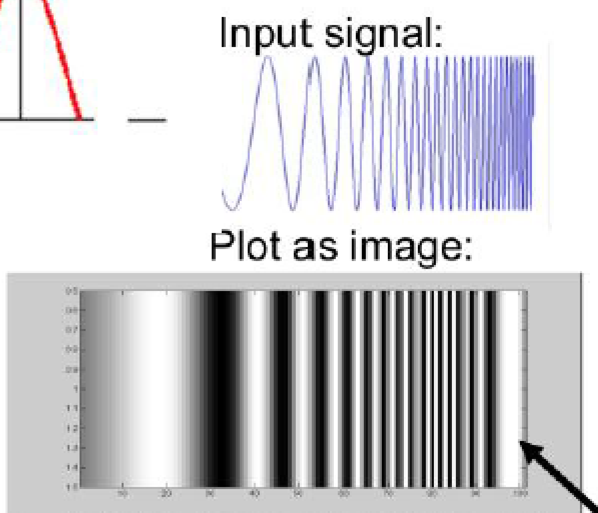
  if sampling frequency is reduced …




Frequencies above the Nyquist limit are 'folded back' corrupting the signal in the acceptable range.

The information in these frequencies is not correctly reconstructed.

- Example:

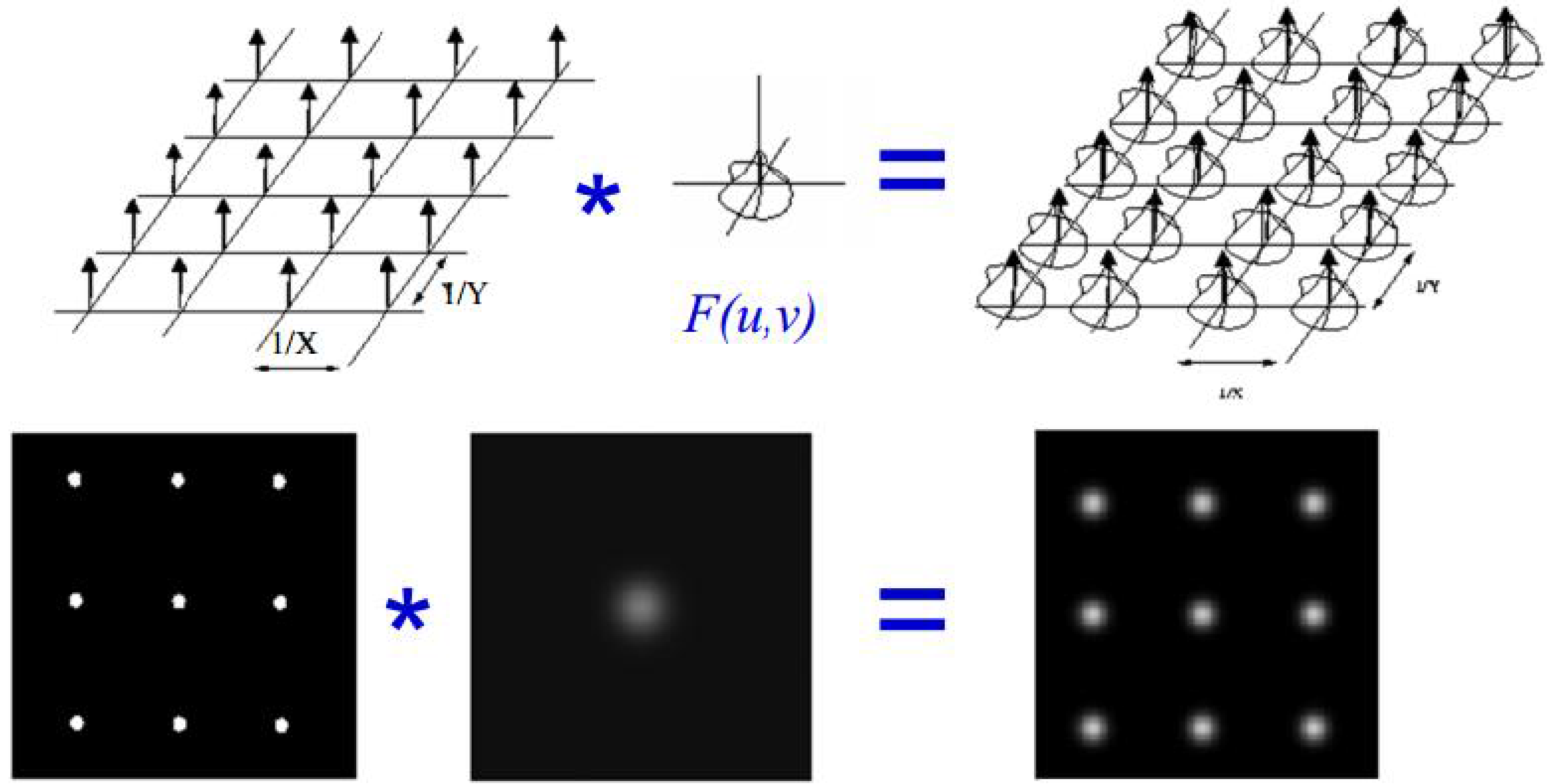



x = 0:.05:5; imagesc(sin((2.^x).*x))

# Sampling in 2D

## ❑ Aliasing in 2D

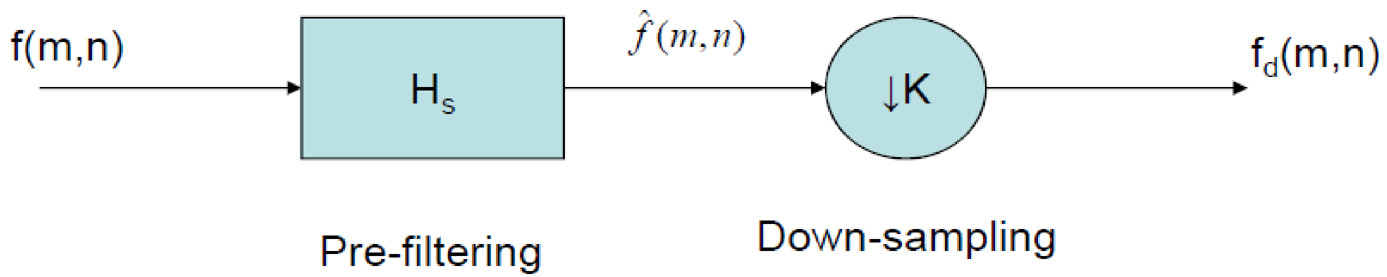
frequency domain




$$H(u,v) = rect(uX)rect(vY)$$

$$f(x,y) = \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} f(nX, mY) \text{sinc}\frac{\pi}{X}(x - nX) \text{sinc}\frac{\pi}{Y}(y - nY)$$

# Down Sampling by a Factor of K




$$f_d(m,n) = \hat{f}(Km, Kn)$$

For factor of K down sampling, the prefilter should be low pass filter with cutoff at fs/(2K), if fs is the original sampling frequency

In terms of digital frequency, the cutoff should be 1/(2K)

# Example: Image Down-Sample

Without prefiltering

With prefiltering (no aliasing, but blurring!)

# Down-Sampling Using Matlab

- Without prefiltering
  - If f(,) is an MxN image, down-sampling by a factor of K can be done simply by

```
>> g=f(1:K:M,1:K:N)
```

- With prefiltering
  - First convolve the image with a desired filter
    - Low pass filter with digital cutoff frequency 1/(2K)
      - In matlab, 1/2 is normalized to 1
  - Then subsample

```
>> h=fir1(N, 1/K)
   %design a lowpass filter with cutoff at 1/K.
>> fp=conv2(f,h)
>> g=fp(1:K:M,1:K:N)
```

```
clearvars,
K = 4;
%I = imread('coins.png');
I = imread('person.bmp');
I = rgb2gray(I);
SS=size(I);
M1=SS(1);
M2=SS(2);
J1=I(1:K:M1,1:K:M2);
N = max(M1,M2);
%LPF first:
FF = fir1(N,1/K);
```

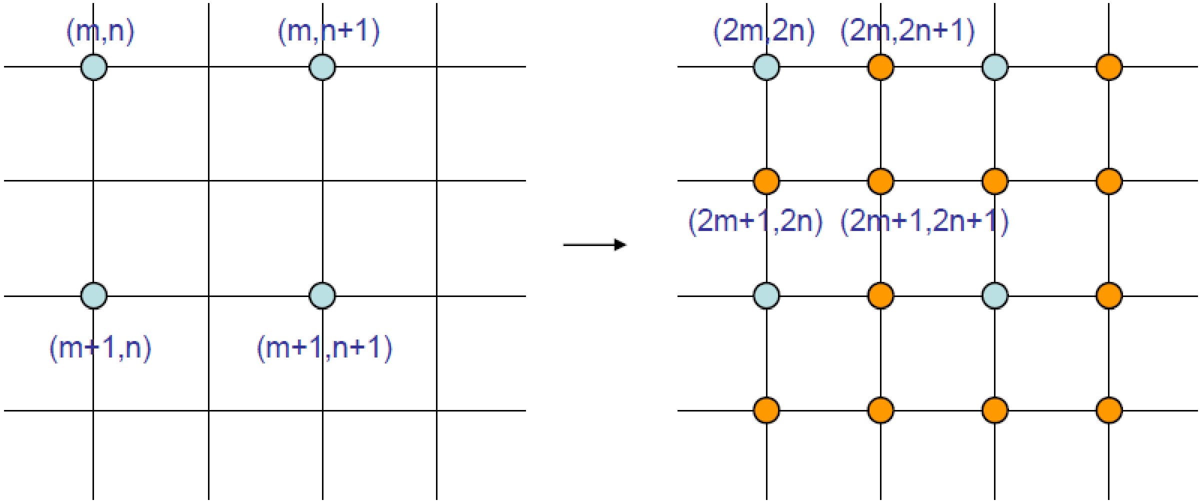
- II = conv2(I , FF, 'same');
- J2=II(1:K:M1,1:K:M2);
- figure();
- subplot(2,2,1), imshow(I, []), title('ogiginal')
- subplot(2,2,2), imshow(II, []), title('filtered')
- subplot(2,2,3), imshow(J1, []), title('downsampled-no-filter')
- subplot(2,2,4), imshow(J2, []), title('downsampled-w-filter')

- Repeat for a colored version?
- Instead of using conv2 we use imfilter

# Image Up-Sampling

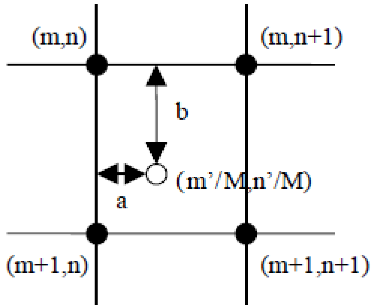
- Produce a larger image from a smaller one
  - Eg. 512x512 -> 1024x1024
  - More generally we may up-sample by an arbitrary factor L
- Questions:
  - How should we generate a larger image?

# Example: Factor of 2 Up-Sampling




Green samples are retained in the interpolated image;
Orange samples are estimated from surrounding green samples.

# Nearest Neighbor Interpolation (pixel replication)

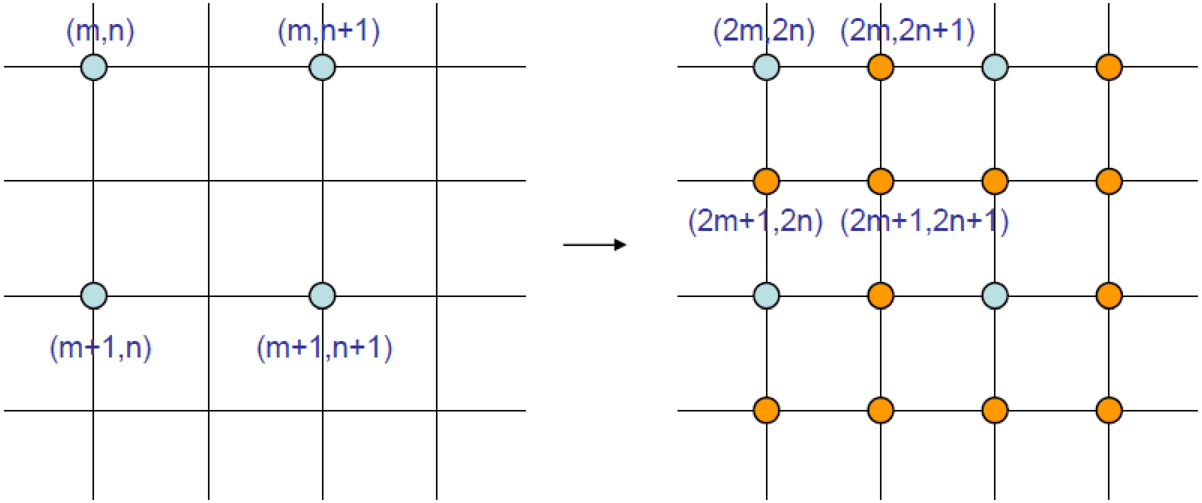



O[m',n'] (the resized image) takes the value of the sample nearest to (m'/M,n'/M) in I[m,n] (the original image):

$$O[m',n'] = I[(\text{int})\,(m+0.5),\,(\text{int})\,(n+0.5)]\,,\; m = m'/M,\; n = n'/M.$$

Also known as pixel replication: each original pixel is replaced by MxM pixels of the sample value
Equivalent to using the sample-and-hold interpolation filter.

# Special Case: M=2




Nearest Neighbor:
O[2m,2n]=I[m,n]
O[2m,2n+1]= I[m,n]
O[2m+1,2n]= I[m,n]
O[2m+1,2n+1]= I[m,n]

# Image Interpolation

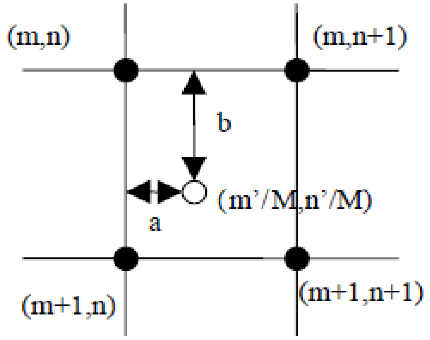
- Nearest neighbour interpolation
  - Simple but produces undesired artefacts
- Bilinear Interpolation (1st degree in both x and y thus Bi-linear)

$$\boxed{v(x, y) = ax + by + cxy + d}$$

- **Bicubic Interpolation** (3rd degree polynomial in both x and y thus is Bi-cubic)

$$\boxed{v(x, y) = \sum_{i=0}^{3} \sum_{j=0}^{3} a_{ij} x^i y^j}$$

# Bilinear Interpolation

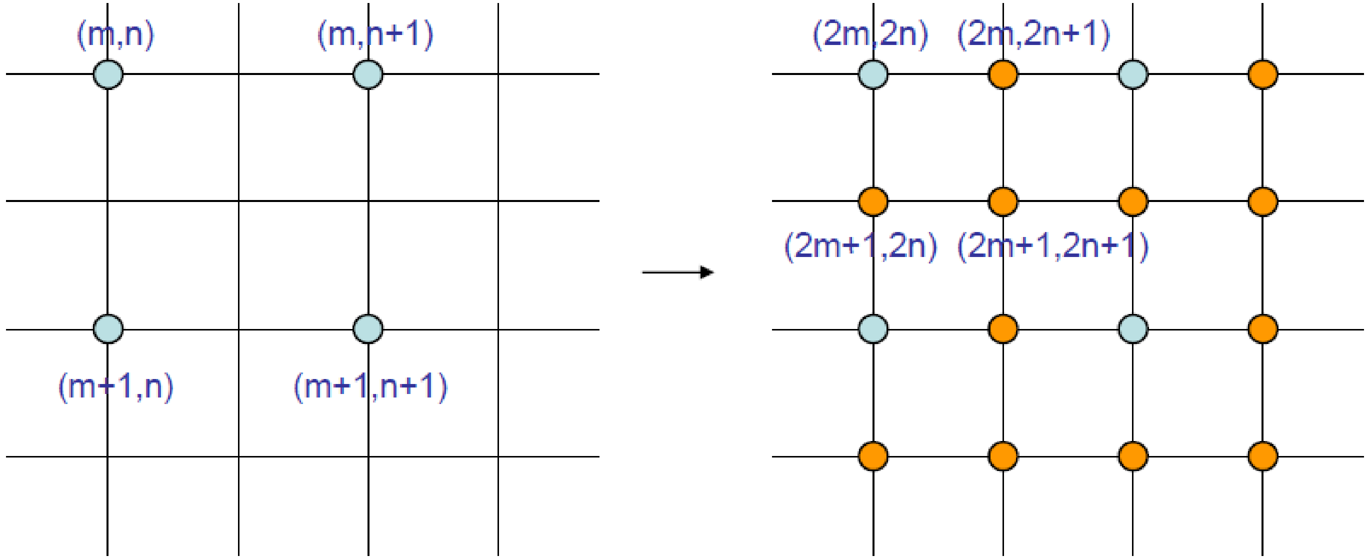



- O(m',n') takes a weighted average of 4 samples nearest to (m'/M,n'/M) in I(m,n).

- Direct interpolation: each new sample takes 4 multiplications:
  O[m',n']=(1-a)*(1-b)*I[m,n]+a*(1-b)*I[m,n+1]+(1-a)*b*I[m+1,n]+a*b*I[m+1,n+1]

- Separable interpolation:
  i) interpolate along each row y: F[m,n']=(1-a)*I[m,n]+a*I[m,n+1]
  ii) interpolate along each column x': O[m',n']=(1-b)*F[m',n]+b*F[m'+1,n]

# Special Case: M=2




Bilinear Interpolation:

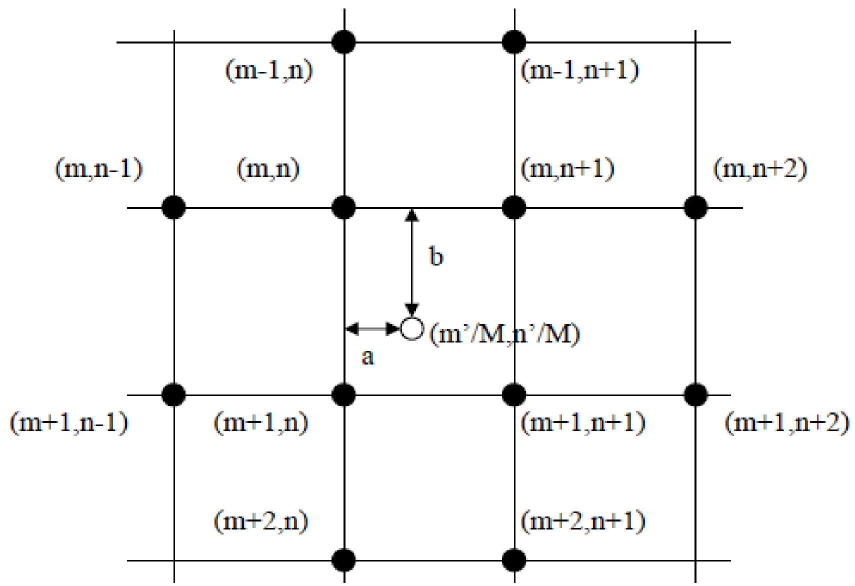
O[2m,2n]=I[m,n]

O[2m,2n+1]=(I[m,n]+I[m,n+1])/2

O[2m+1,2n]=(I[m,n]+I[m+1,n])/2

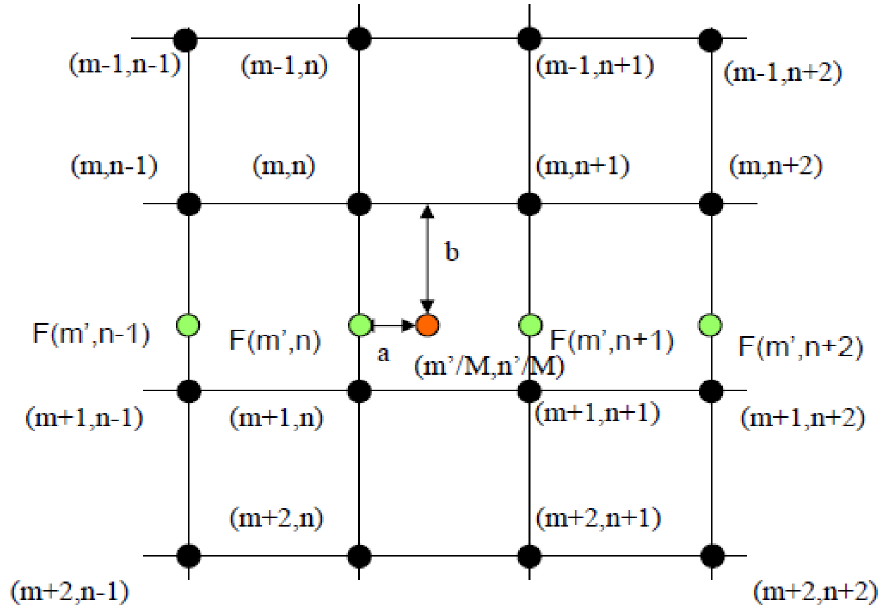
O[2m+1,2n+1]=(I[m,n]+I[m,n+1]+I[m+1,n]+I[m+1,n+1])/4

# Bicubic Interpolation




- O(m',n') is interpolated from 16 samples nearest to (m'/M,n'/M) in I(m,n).
- Direct interpolation: each new sample takes 16 multiplications
- Separable interpolation:
  - i) interpolate along each row y: I[m,n]->F[m,n'] (from 4 samples)
  - ii) interpolate along each column x': F[m,n']-> O[m',n'] (from 4 samples)

# Interpolation Formula



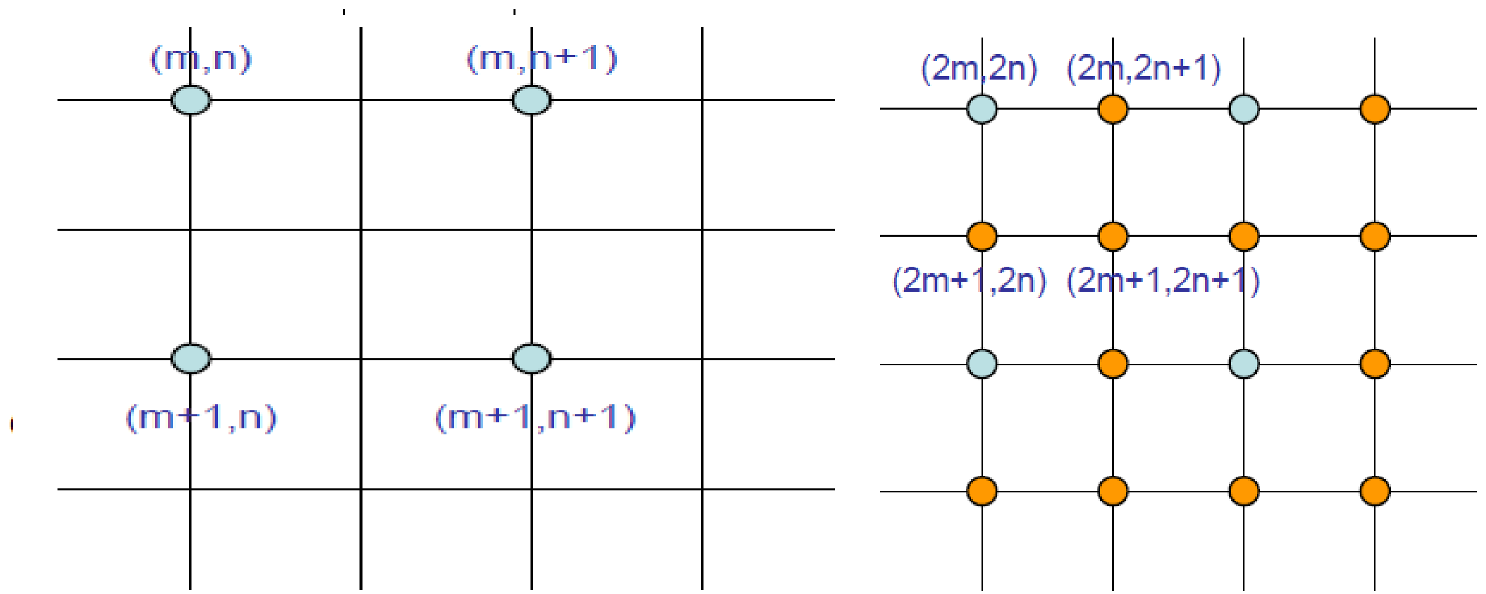

$$F[m',n] = -b(1-b)^2 I[m-1,n] + (1-2b^2+b^3) I[m,n] + b(1+b-b^2) I[m+1,n] - b^2(1-b) I[m+2,n],$$

$$\textit{where} \quad m = (\text{int})\frac{m'}{M}, b = \frac{m'}{M} - m$$

$$O[m',n'] = -a(1-a)^2 F[m',n-1] + (1-2a^2+a^3) F[m',n] + a(1+a-a^2) F[m',n+1] - a^2(1-a) F[m',n+2],$$

$$\textit{where} \quad n = (\text{int})\frac{n'}{M}, a = \frac{n'}{M} - n$$

# Special Case: M=2




Bicubic interpolation in Horizontal direction

F[2m,2n]=I[m,n]
F[2m,2n+1]= -(1/8)I[m,n-1]+(5/8)I[m,n]+(5/8)I[m,n+1]-(1/8)I(m,n+2)

Same operation then repeats in vertical direction

O(2m,2n+1) as a function of the F(x,y)

# Matlab for Image Resizing

```
[img]=imread('fruit.jpg','jpg');
%downsampling without prefiltering
img1=imresize(img,0.5,'nearest');
%upsampling with different filters:
img2rep=imresize(img1,2,'nearest');
img2lin=imresize(img1,2,'bilinear');
img2cubic=imresize(img1,2,'bicubic');

%down sampling with filtering
img1=imresize(img,0.5,'bilinear',11);
%upsampling with different filters
img2rep=imresize(img1,2,'nearest');
img2lin=imresize(img1,2,'bilinear');
img2cubic=imresize(img1,2,'bicubic');
```

# Deep Learning Perspective

- Image super resolution using GAN and Diffusion networks
- Encoder-decoder convolutional networks (e.g. FCN) can be used to down/up sample images

# Deep Network Interpolation

# for Continuous Imagery Effect Transition

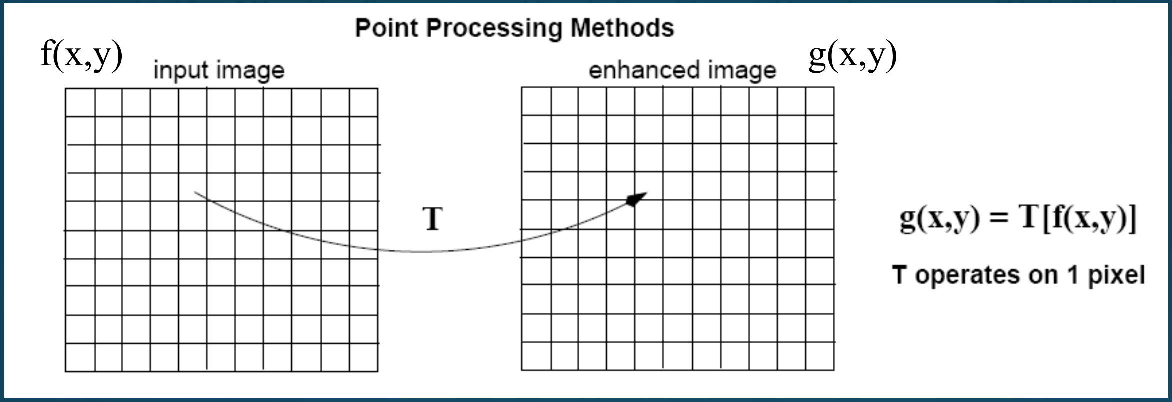
Xintao Wang [1] Ke Yu [1] Chao Dong [2] Xiaoou Tang [1] Chen Change Loy [3]
[1] CUHK - SenseTime Joint Lab, The Chinese University of Hong Kong
[2] Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences
[3] Nanyang Technological University, Singapore

- DNI
- Understanding DNI
- Applications

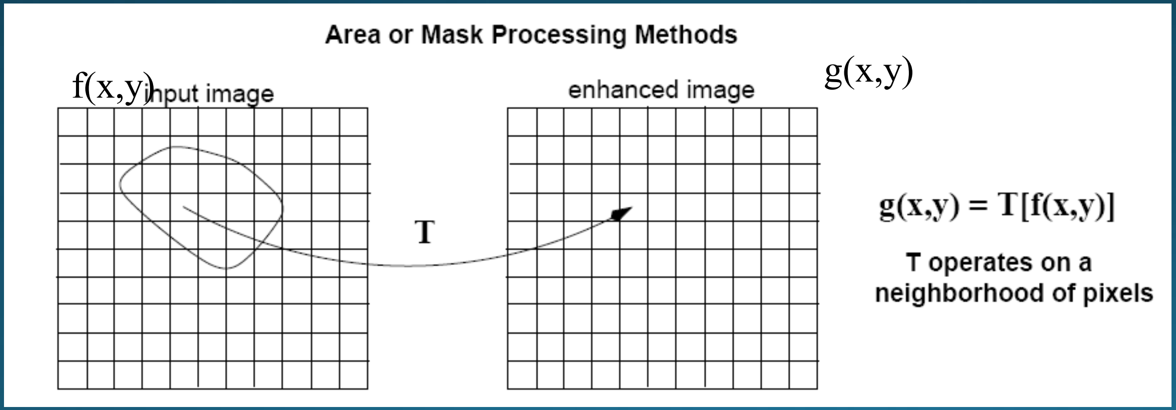
# II- Intensity Transformations

# Spatial Domain Methods

Point Processing

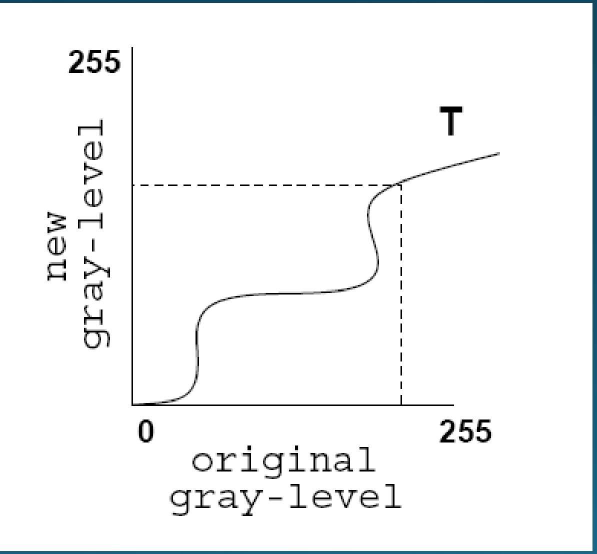



Area/Mask Processing

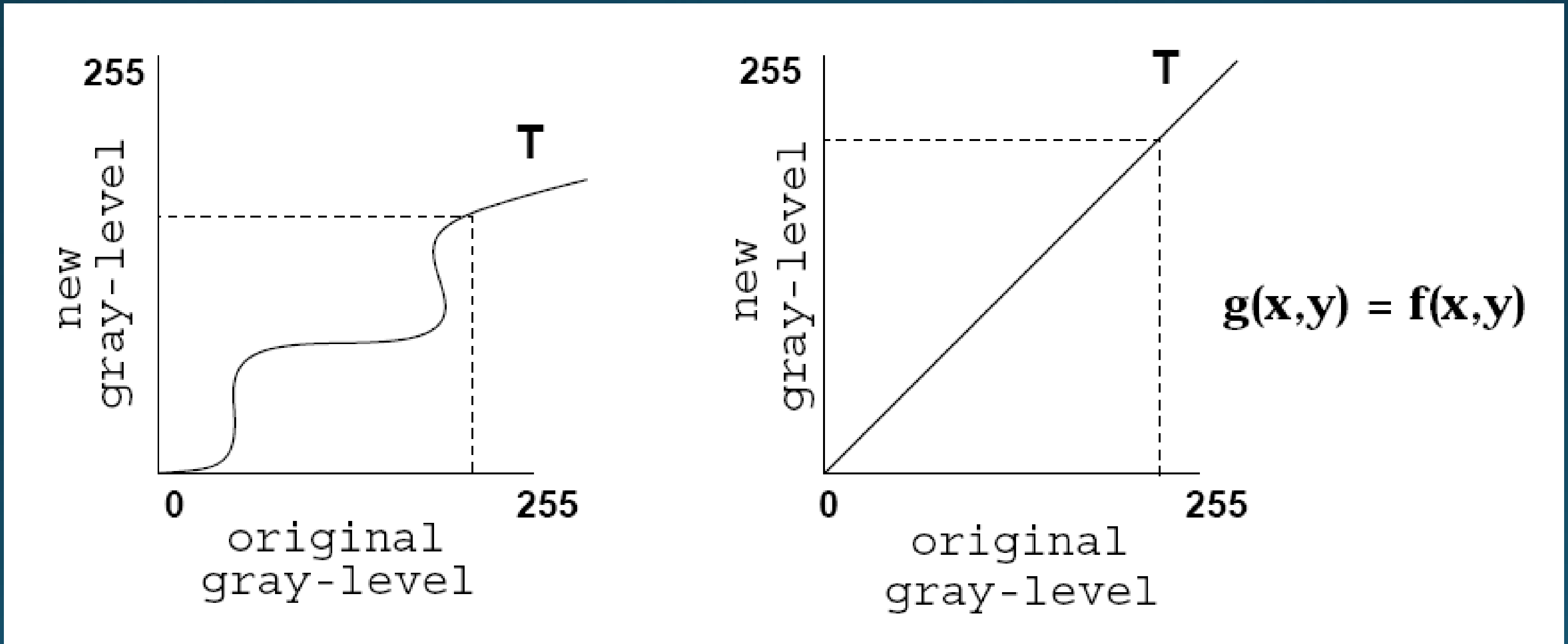


# Point Processing Transformations

- Convert a given pixel value to a new pixel value based on some predefined function.

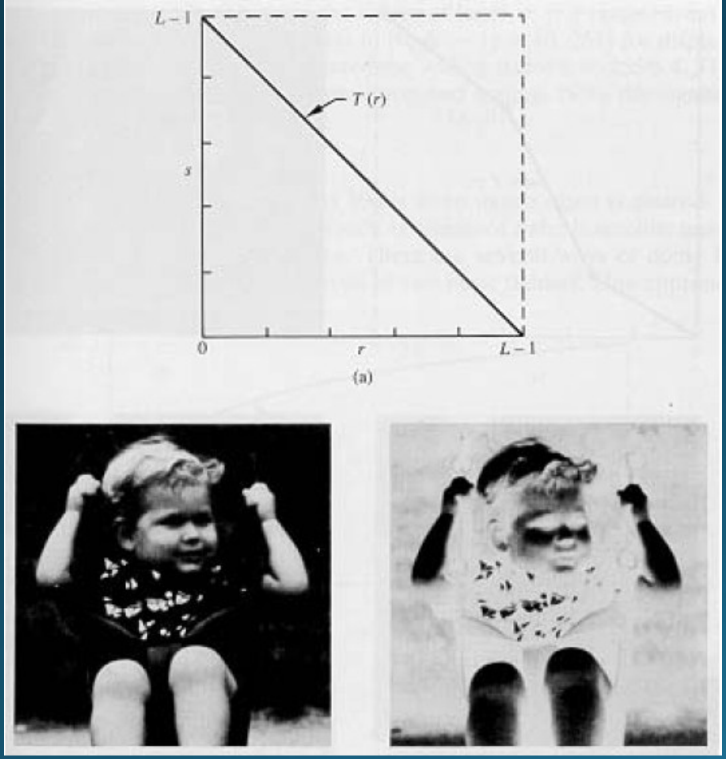


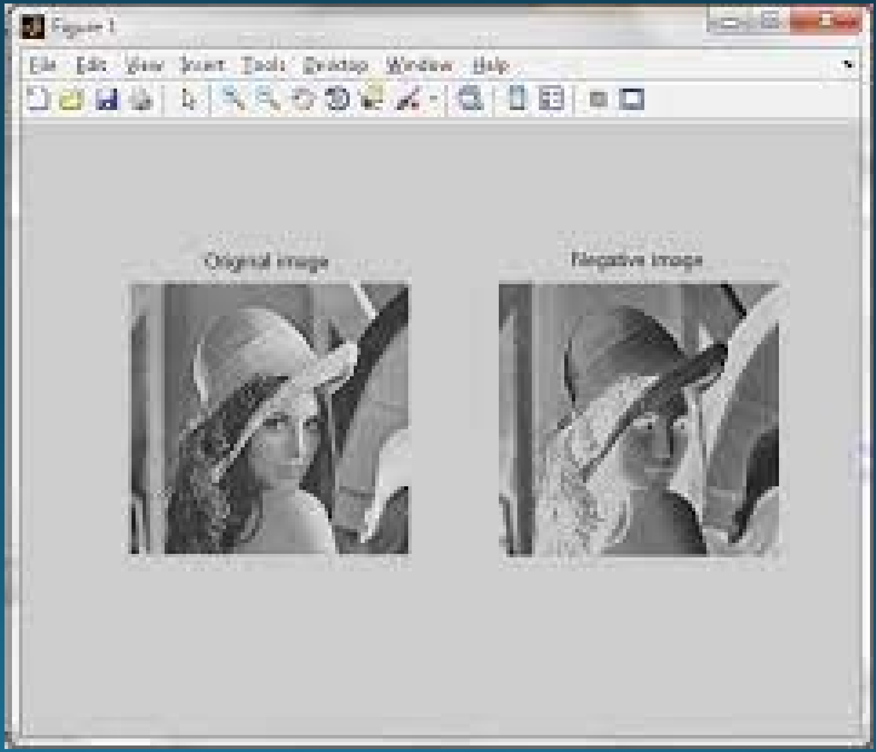
# Identity Transformation



# Negative Image

- O(r,c) = 255-I(r,c)

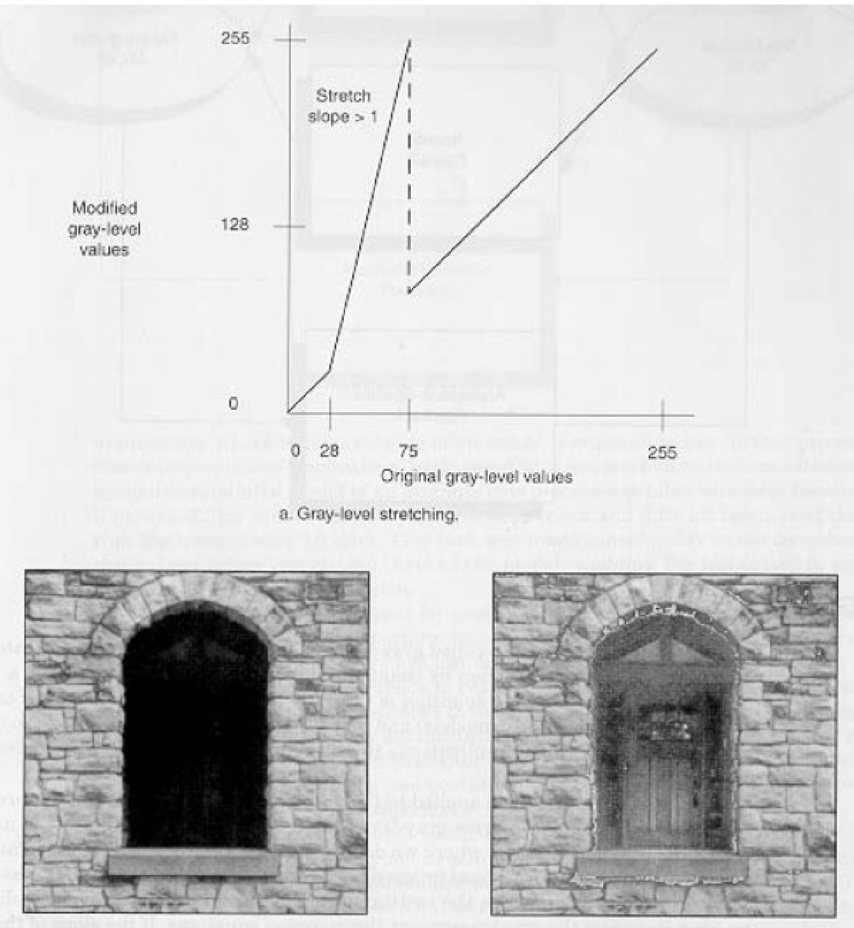


# Contrast Stretching or Compression

- Stretch gray-level ranges where we desire more information (slope > 1).

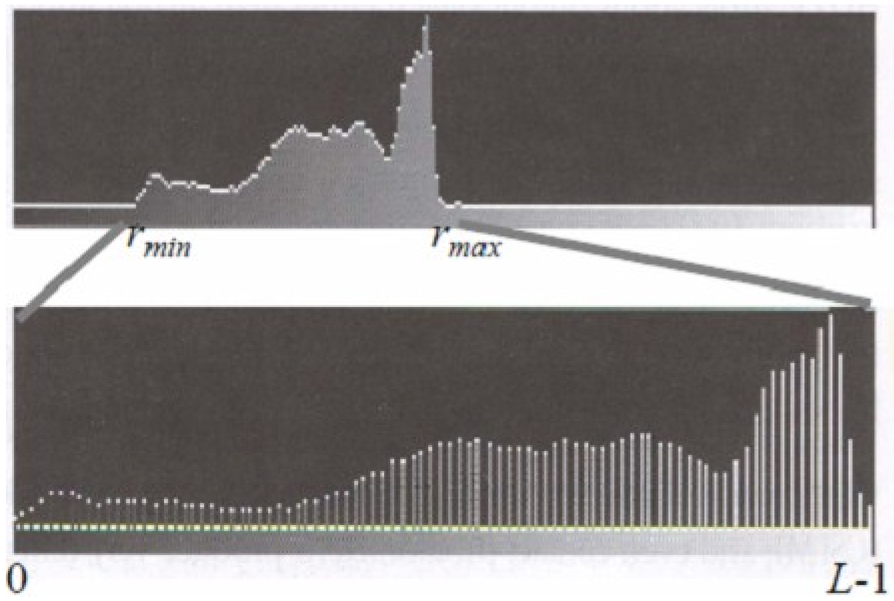
- Compress gray-level ranges that are of little interest (0 < slope < 1).



a. Gray-level stretching.

# Contrast Stretching

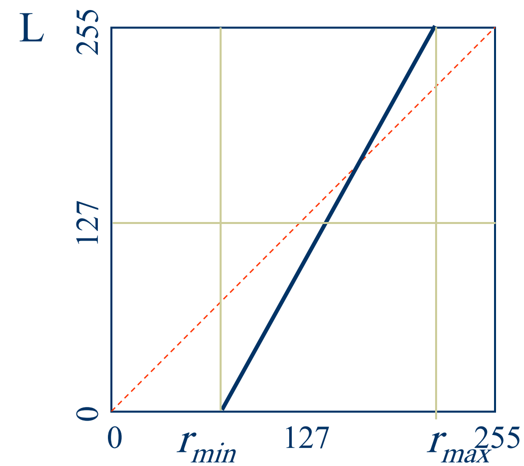
Improve the contrast in an image by `stretching' the range of intensity values it contains to span a desired range of values, *e.g.* the the full range of pixel values



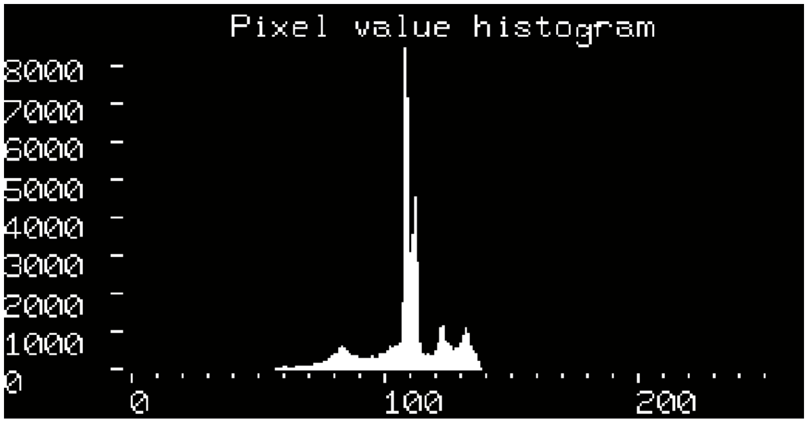
# Contrast Stretching

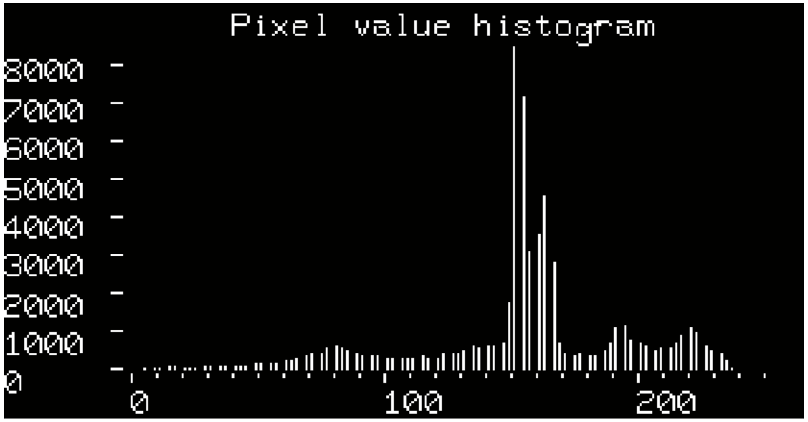
If $r_{max}$ and $r_{min}$ are the maximum and minimum gray level of the input image and $L$ is the total gray levels of output image, the transformation function for contrast stretch will be

$$s = T(r) = (r - r_{\min})\left[\frac{L-1}{r_{\max} - r_{\min}}\right]$$



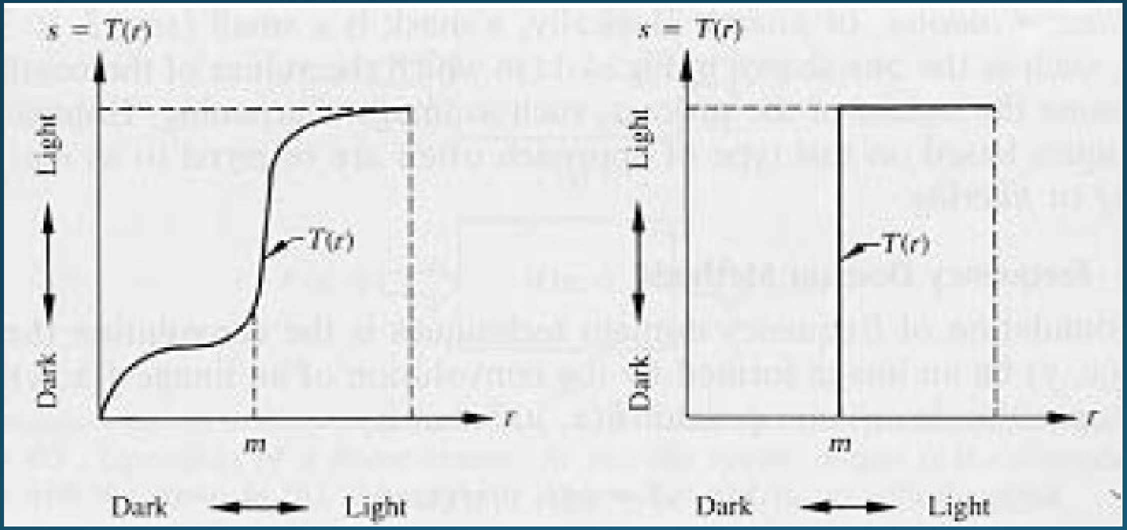
# Contrast Stretching



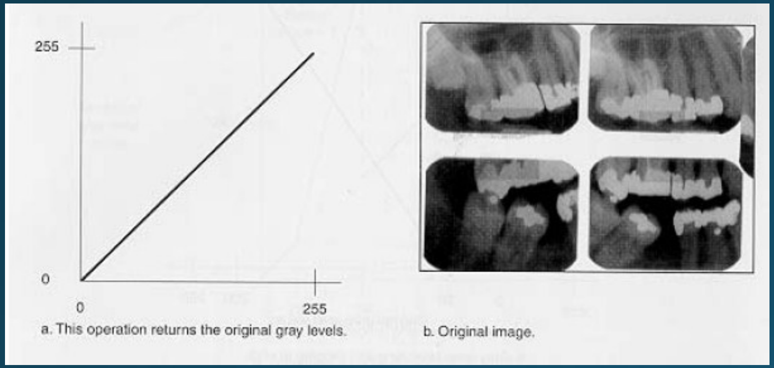
# Thresholding

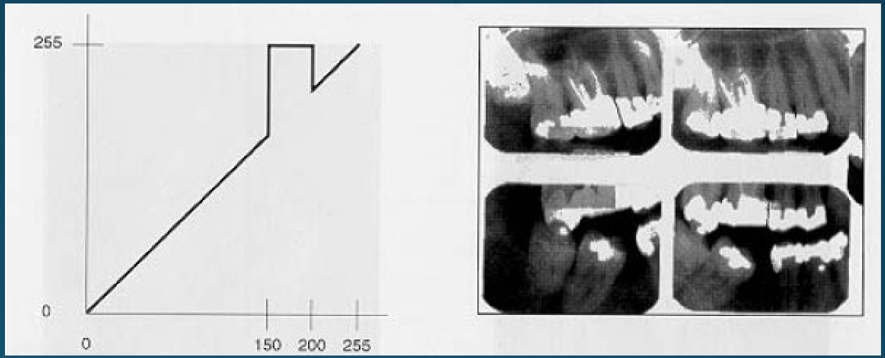
- Special case of contrast compression

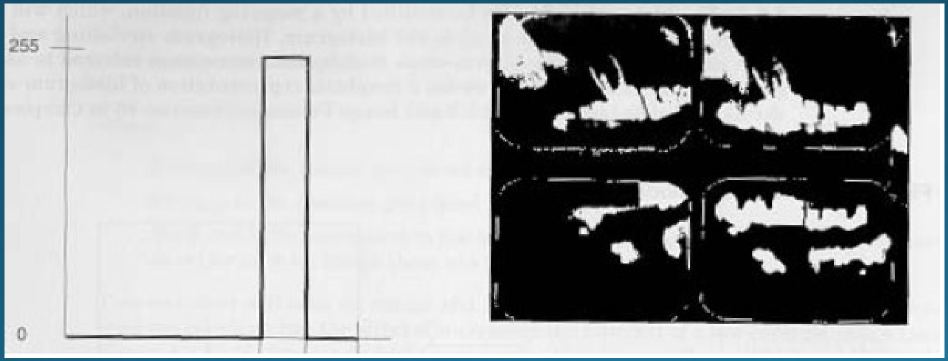


# Intensity Level Slicing

- Highlight specific ranges of gray-levels only.



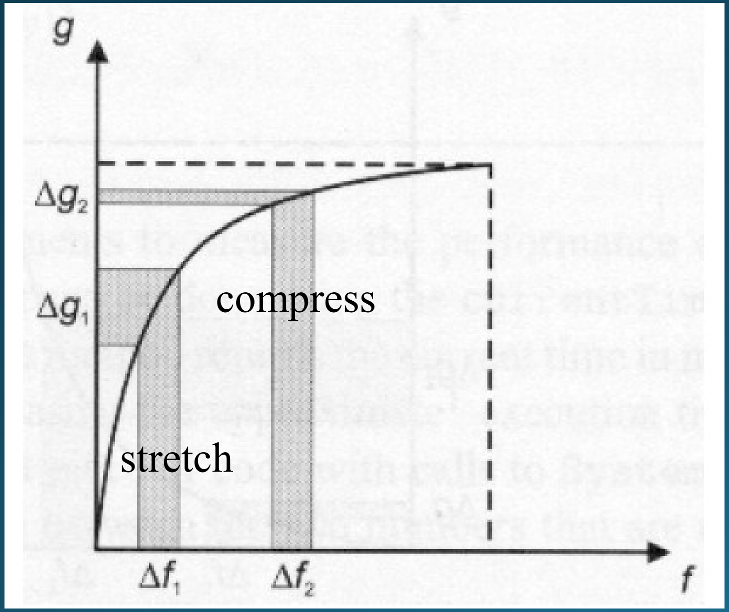



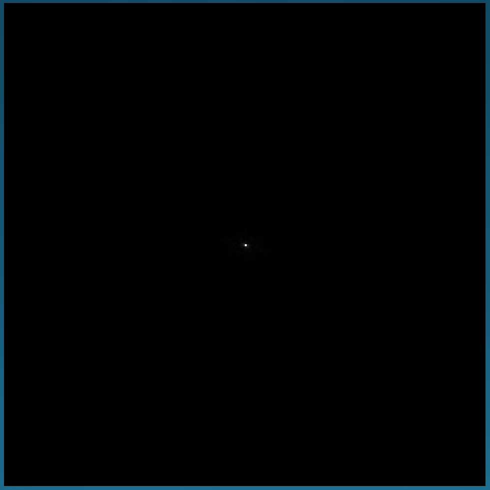

Same as double thresholding!

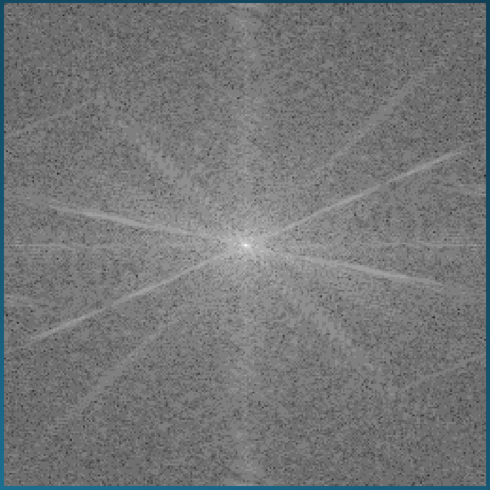
# Logarithmic transformation

- Enhance details in the darker regions of an image at the expense of detail in brighter regions.

$$s = T(r) = c\log(1+|r|)$$

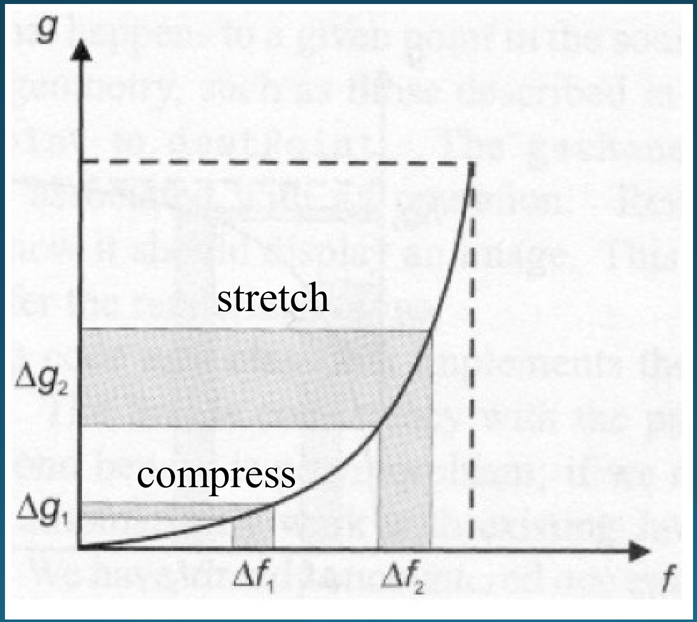




# Exponential transformation

- Reverse effect of that obtained using logarithmic mapping.



# Histogram Based Image Intensity Transformation

# Histogram of a Grayscale Image

- Let $I$ be a 1-band (grayscale) image.
- $I(r,c)$ is an 8-bit integer between 0 and 255.
- Histogram, $h_I$, of $I$:
  - a 256-element array, $h_I$
  - $h_I(g)$ = number of pixels in $I$ that have value $g$.

    *for $g = 0, 1, 2, 3, \ldots, 255$*

# Histogram of a Grayscale Image

- Histogram of a digital image with gray levels in the range *[0,L-1]* is a discrete function

$$\boxed{h(r_k) = n_k}$$

Where

- $r_k = k^{th}$ gray level
- $n_k$ = number of pixels in the image having gray level $r_k$
- $h(r_k)$ = histogram of an image having $r_k$ gray levels

# Normalized Histogram

- Dividing each of histogram at gray level $r_k$ by the total number of pixels in the image, $n$

$$p(r_k) = n_k / n \quad for\ k = 0, 1, \ldots$$

- $p(r_k)$ gives an estimate of the probability of occurrence of gray level $r_k$
- The sum of all components of a normalized histogram is equal to 1

# Image Histograms

- An image histogram is a plot of the gray-level frequencies (i.e., the number of pixels in the image that have that gray level).

| | | | | | |
|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 2 | 0 |
| 1 | 0 | 7 | 7 | 7 | 0 |
| 0 | 7 | 0 | 0 | 7 | 0 |
| 1 | 0 | 0 | 7 | 2 | 0 |
| 0 | 0 | 7 | 1 | 0 | 1 |
| 1 | 0 | 7 | 7 | 7 | 0 |

frequencies

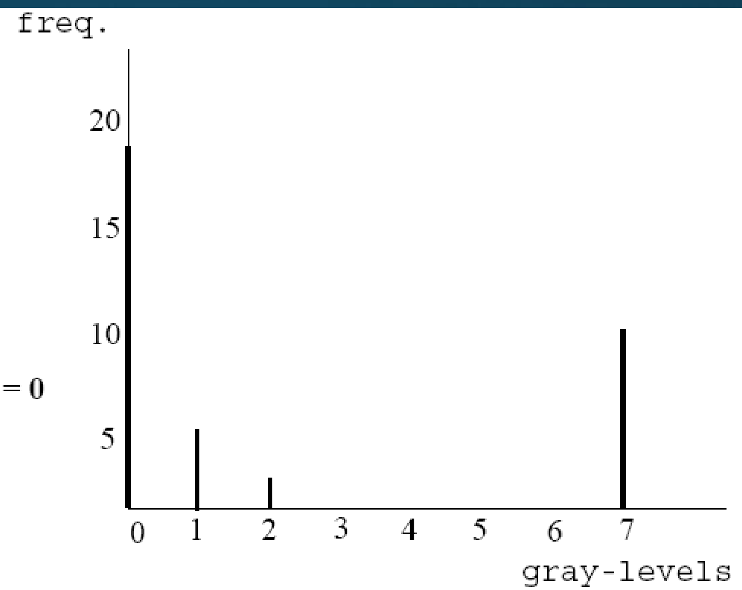
$f(0) = 18$

$f(1) = 6$

$f(2) = 2$

$f(3) = f(4) = f(5) = f(6) = 0$

$f(7) = 10$



# Image Normalized Histograms

- Divide frequencies by total number of pixels to represent as probabilities.

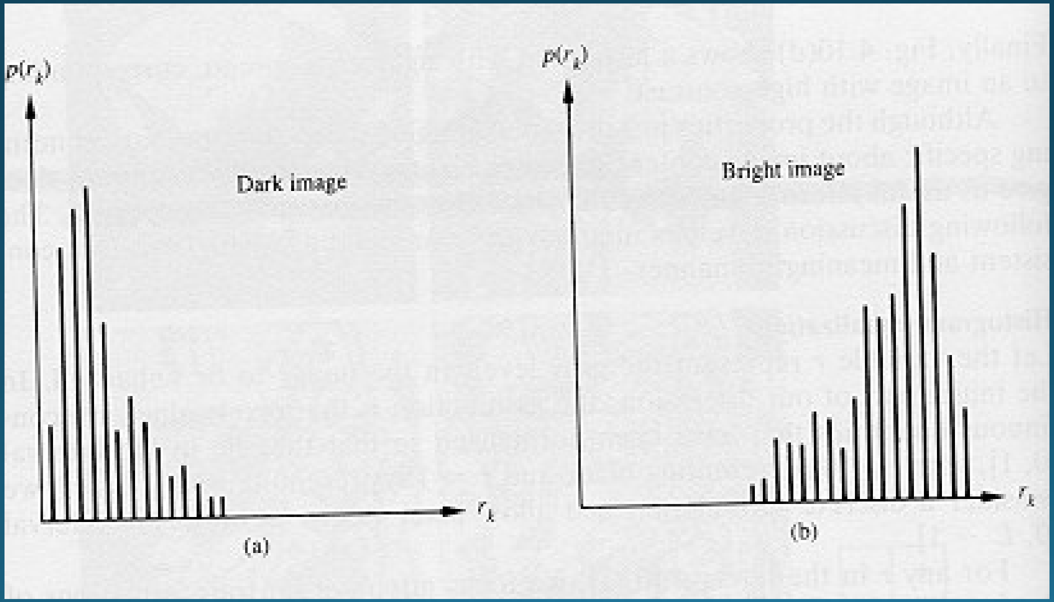
$$P(0) = \frac{f(0)}{36} = \frac{1}{2} \qquad P(1) = \frac{f(1)}{36} = \frac{1}{6}$$

$$P(2) = \frac{f(2)}{36} = \frac{1}{18} \qquad P(3) = P(4) = P(5) = P(6) = 0$$

$$P(7) = \frac{f(7)}{36} = \frac{5}{18} \qquad p_k = n_k / N$$

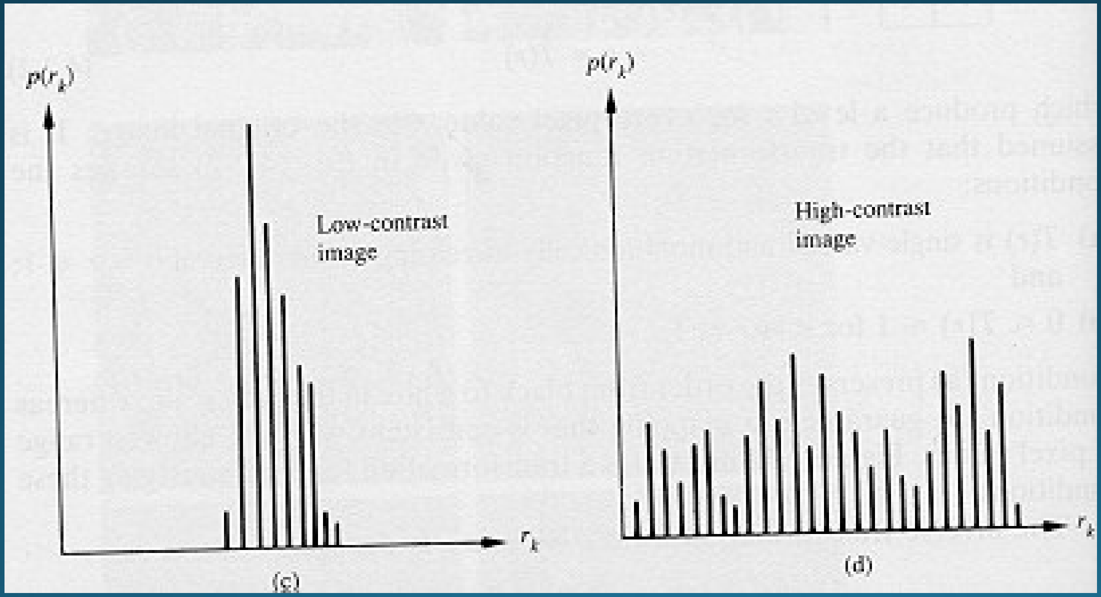
# Properties of Image Histograms

- Histograms clustered at the low end correspond to **dark** images.
- Histograms clustered at the high end correspond to **bright** images.

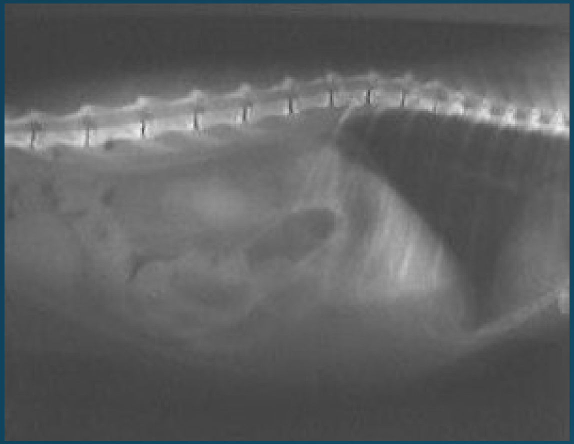
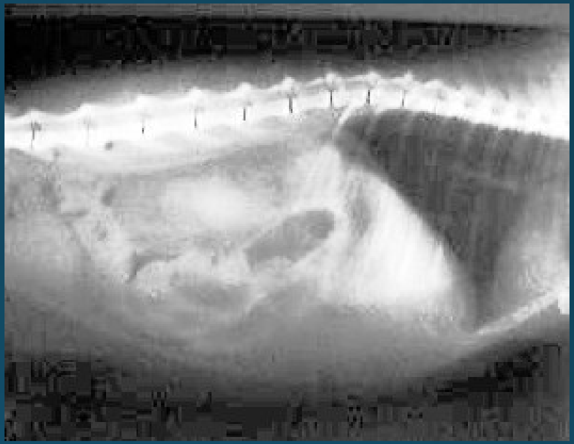
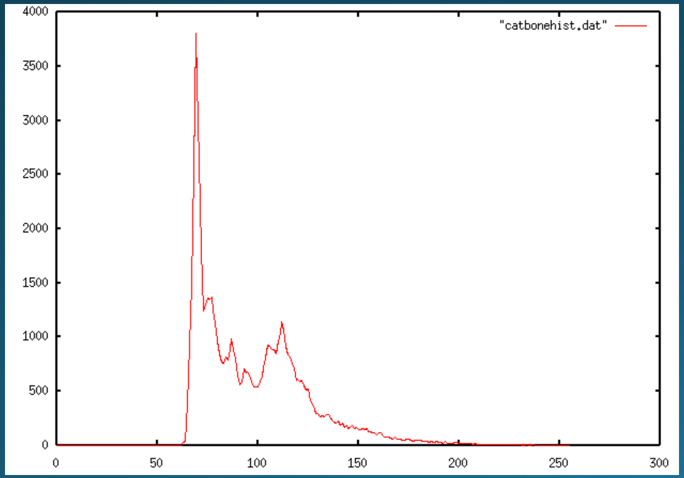
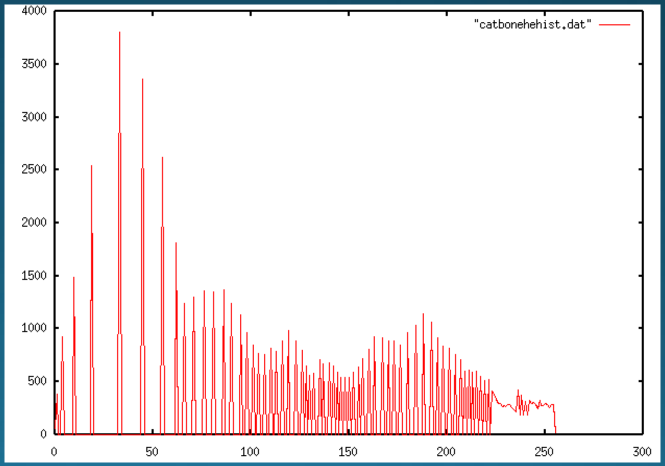


# Properties of Image Histograms (cont'd)

- Histograms with small spread correspond to **low contrast** images (i.e., mostly dark, mostly bright, or mostly gray).
- Histograms with wide spread correspond to **high contrast** images.

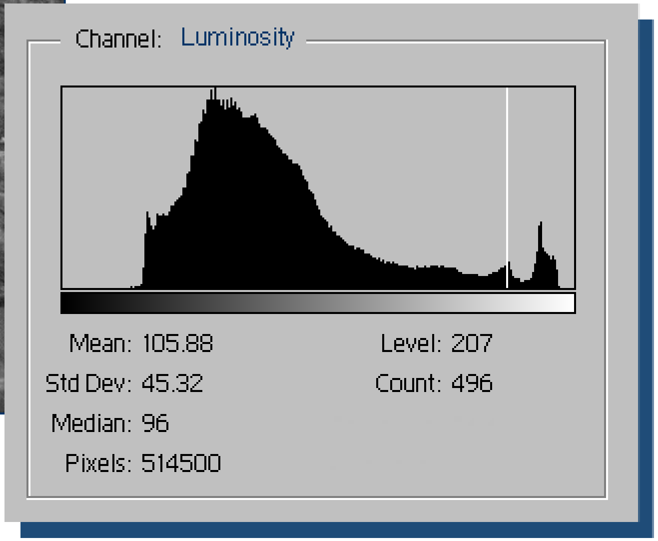


# Properties of Image Histograms (cont'd)

| Low contrast | High contrast |
| --- | --- |
|  |  |
|  |  |

# Histogram of a Grayscale Image

$h_I(g) =$ the number of pixels in $I$ with graylevel $g$.

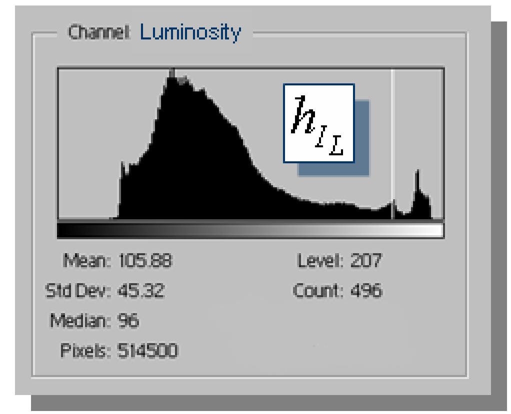


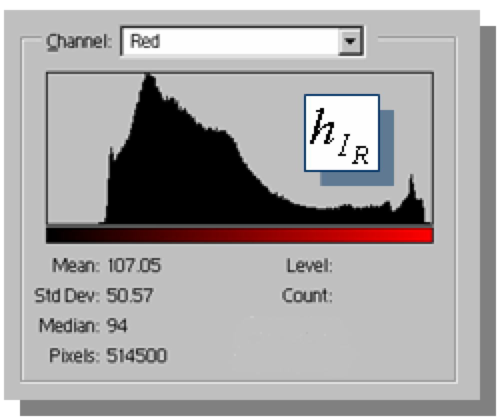
# Histogram of a Color Image

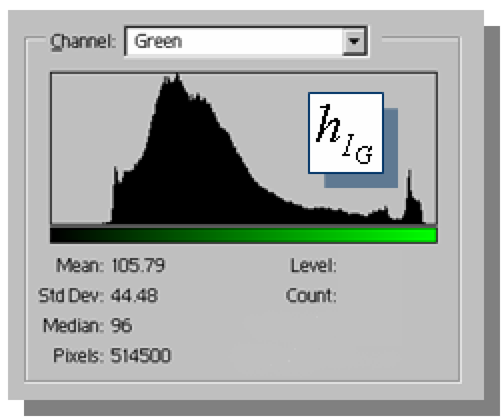
- If I is a 3-band image
- then I($r$,$c$,$b$) is an integer between 0 and 255.
- I has 3 histograms:
  - $h_R(g)$ = # of pixels in I(:,:,1) with intensity value $g$
  - $h_G(g)$ = # of pixels in I(:,:,2) with intensity value $g$
  - $h_B(g)$ = # of pixels in I(:,:,3) with intensity value $g$

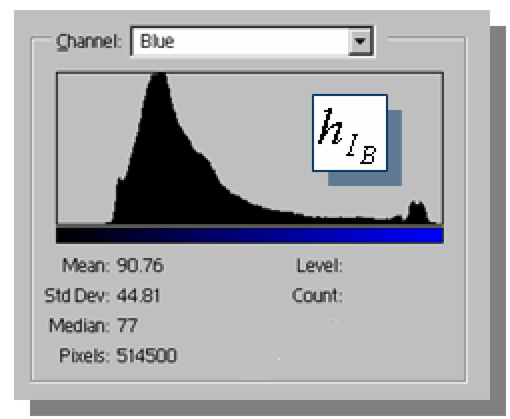
# Histogram of a Color Image

There is one histogram per color band R, G, & B. Luminosity histogram is from 1 band = (R+G+B)/3

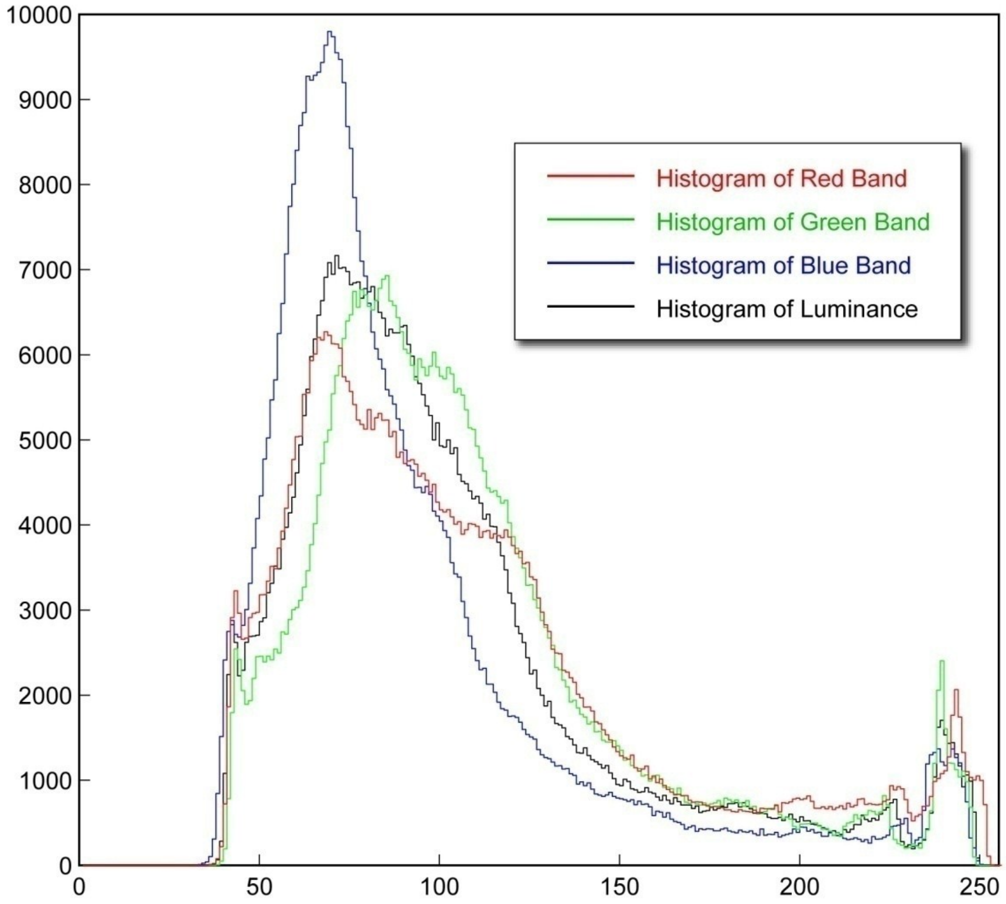


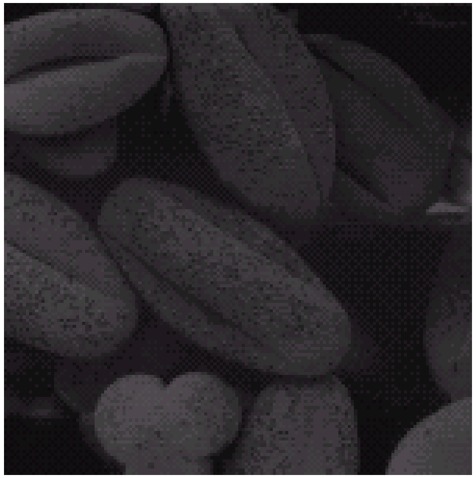



# Histogram of a Color Image



# Histogram of a Color Image

# Histogram: Example

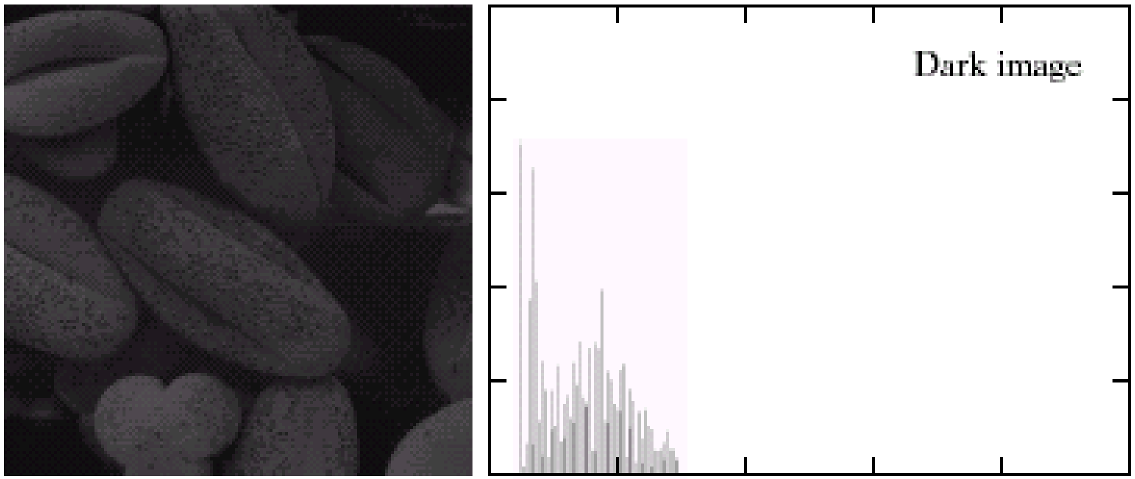


Dark Image

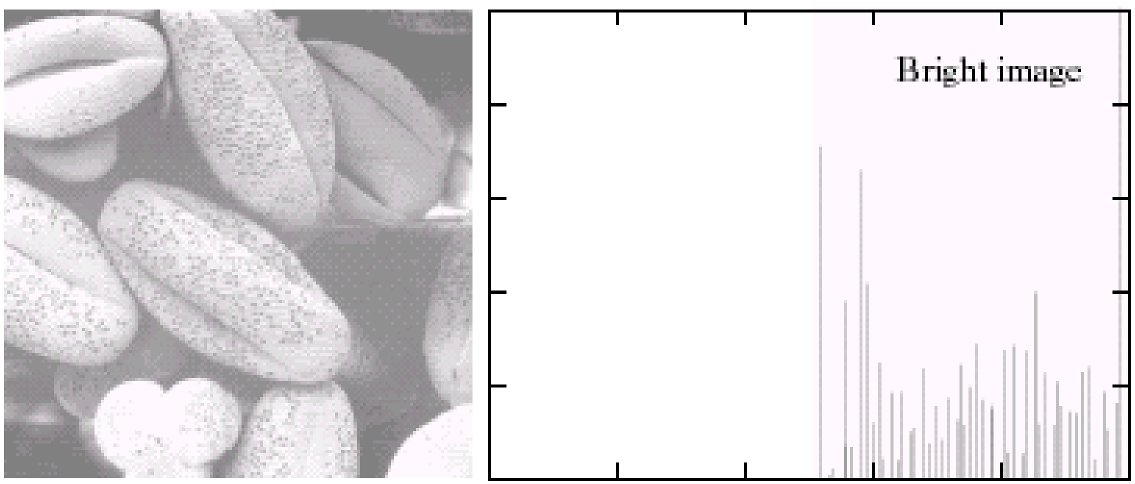
How would the histograms of these images look like?

Bright Image

# Histogram: Example

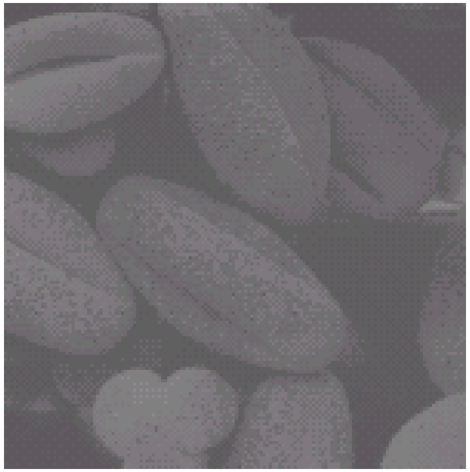



**Dark image**
Components of histogram are concentrated on the low side of the gray scale




**Bright image**
Components of histogram are concentrated on the high side of the gray scale

# Histogram: Example

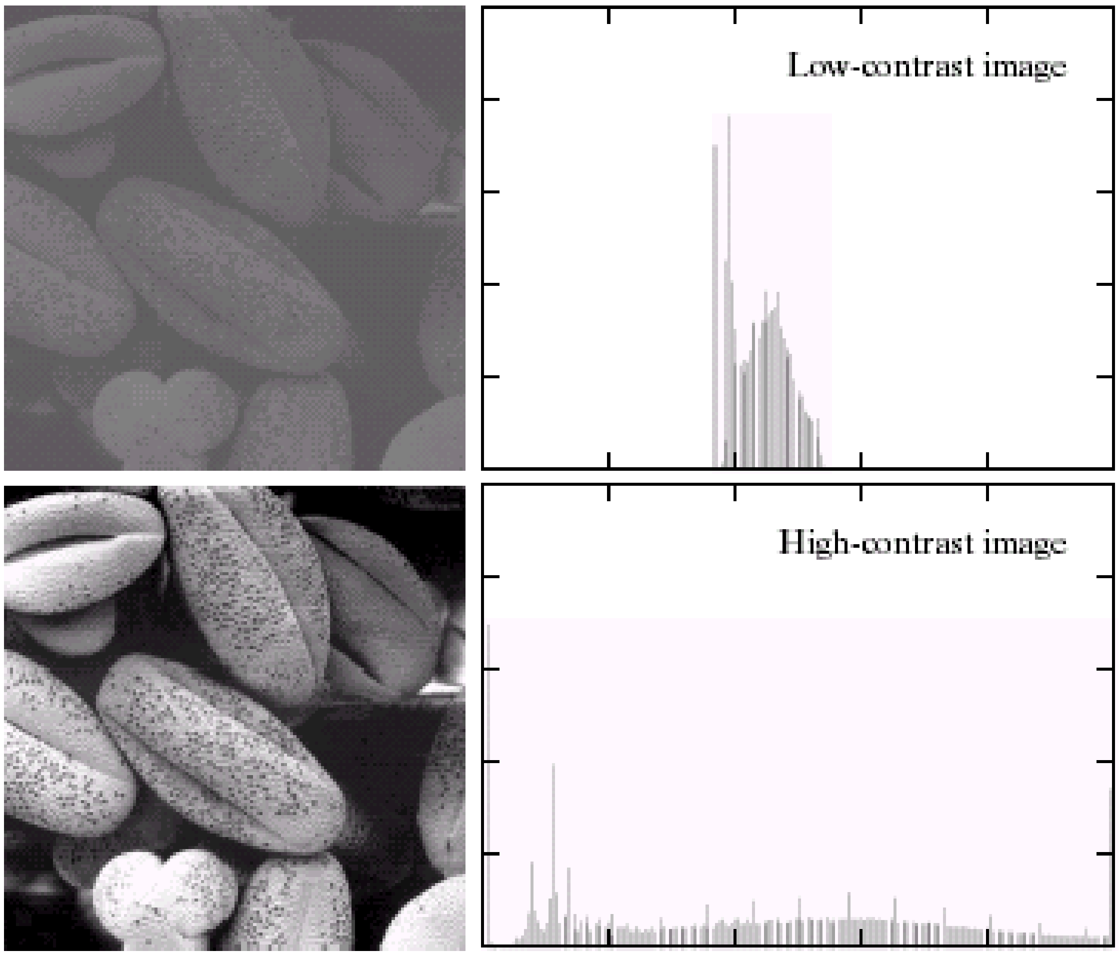


Low Contrast Image

How would the histograms of these images look like?

High Contrast Image

# Histogram: Example




**Low contrast image**

Histogram is narrow and centered toward the middle of the gray scale

**High contrast image**

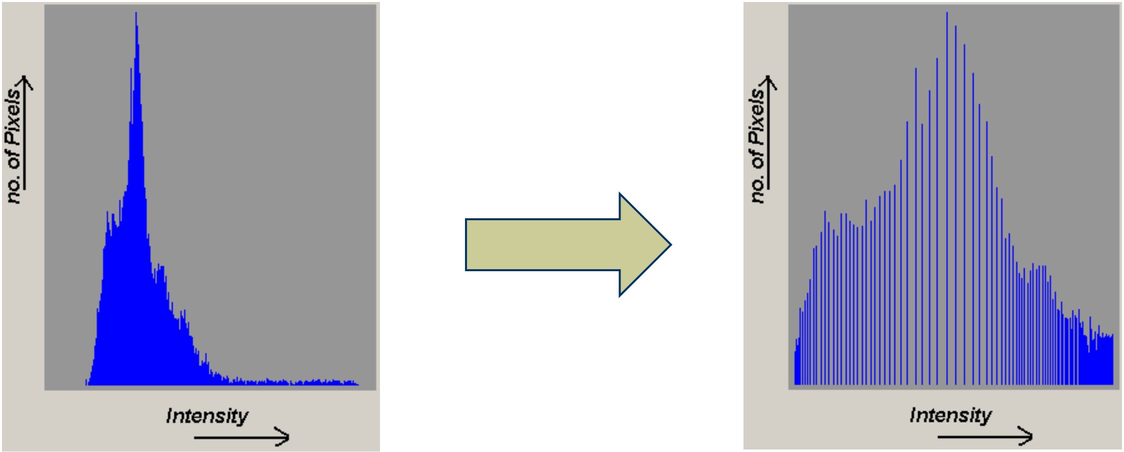
Histogram covers broad range of the gray scale and the distribution of pixels is not too far from uniform with very few vertical lines being much higher than the others

# Histogram Equalization

- A fully automatic gray-level stretching technique.

# Histogram Equalization

Histogram equalization re-assigns the intensity values of pixels in the input image such that the output image contains a uniform distribution of intensities



# The Probability Density of an Image

Let $A = \sum_{g=0}^{255} h_I(g)$

Note that since $h_I(g)$ is the number of pixels in $I$ with value $g$,

$A$ is the number of pixels in $I$. That is if $I$ is $R$ rows by $C$ columns then $A = R \times C$.

Then,

$$p_I(g) = \frac{1}{A} h_I(g)$$

This is the probability that an arbitrary pixel from $I$ has value $g$.

# The Probability Density of an Image

- $p(g)$ is the fraction of pixels in an image that have intensity value $g$.
- $p(g)$ is the probability that a pixel randomly selected from the given image has intensity value $g$.
- Whereas the sum of the histogram $h(g)$ over all $g$ from $0$ to 255 is equal to the number of pixels in the image, the sum of $p(g)$ over all $g$ is 1.
- $p$ is the normalized histogram of the image

# The Cumulative Distribution Function of an Image (CDF)

Let $\mathbf{q} = I(r,c)$ be the value of a randomly selected pixel from $I$. Let $g$ be a specific gray level. The probability that $q \le g$ is given by

$$P_I(g) = \sum_{\gamma=0}^{g} p_I(\gamma) = \frac{1}{A}\sum_{\gamma=0}^{g} h_I(\gamma) = \frac{\sum_{\gamma=0}^{g} h_I(\gamma)}{\sum_{\gamma=0}^{255} h_I(\gamma)},$$

where $h_I(\gamma)$ is the histogram of image $I$.

This is the probability that any given pixel from $I$ has value less than or equal to g.

# The Cumulative Distribution Function of an Image (CDF)

Let **q** = $I(r,c)$ be the value of a randomly selected pixel from $I$. Let $g$ be a specific gray level. The probability that q ≤ g is given by

$$P_I(g) = \sum_{\gamma=0}^{g} p_I(\gamma) = \frac{1}{A}\sum_{\gamma=0}^{g} h_I(\gamma) = \frac{\sum_{\gamma=0}^{g} h_I(\gamma)}{\sum_{\gamma=0}^{255} h_I(\gamma)},$$

where $h_I(\gamma)$ is the histogram of image $I$.

Also called CDF for "Cumulative Distribution Function".

This is the probability that any given pixel from $I$ has value less than or equal to g.

# The Cumulative Distribution Function of an Image (CDF)

- $P(g)$ is the fraction of pixels in an image that have intensity values less than or equal to $g$.
- $P(g)$ is the probability that a pixel randomly selected from the given band has an intensity value less than or equal to $g$.
- $P(g)$ is the cumulative (or running) sum of $p(g)$ from 0 through $g$ inclusive.
- $P(0) = p(0)$ and $P(255) = 1$;

# Histogram Equalization

Task: remap image $I$ so that its histogram is as close to constant as possible

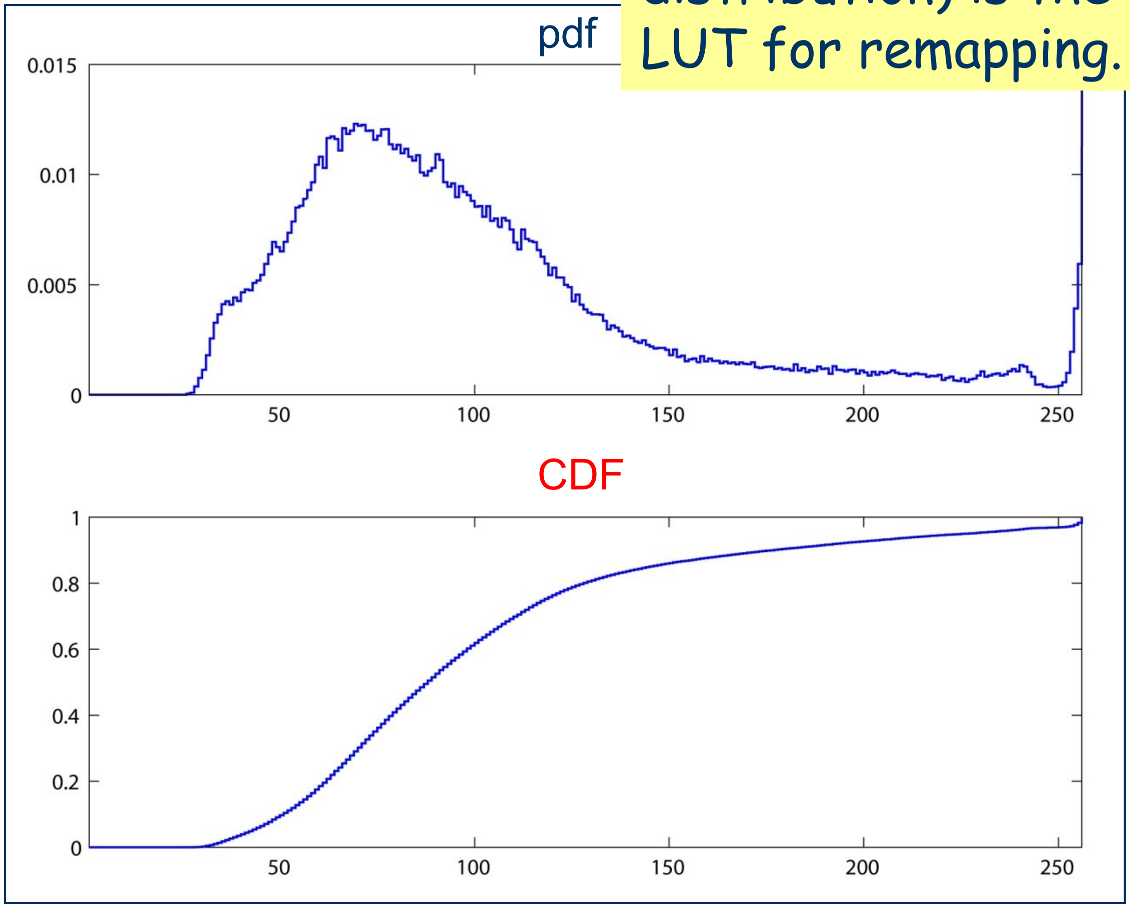
Let $P_I(\gamma)$

be the cumulative (probability) distribution function of $I$.

The CDF itself is used as the LUT.

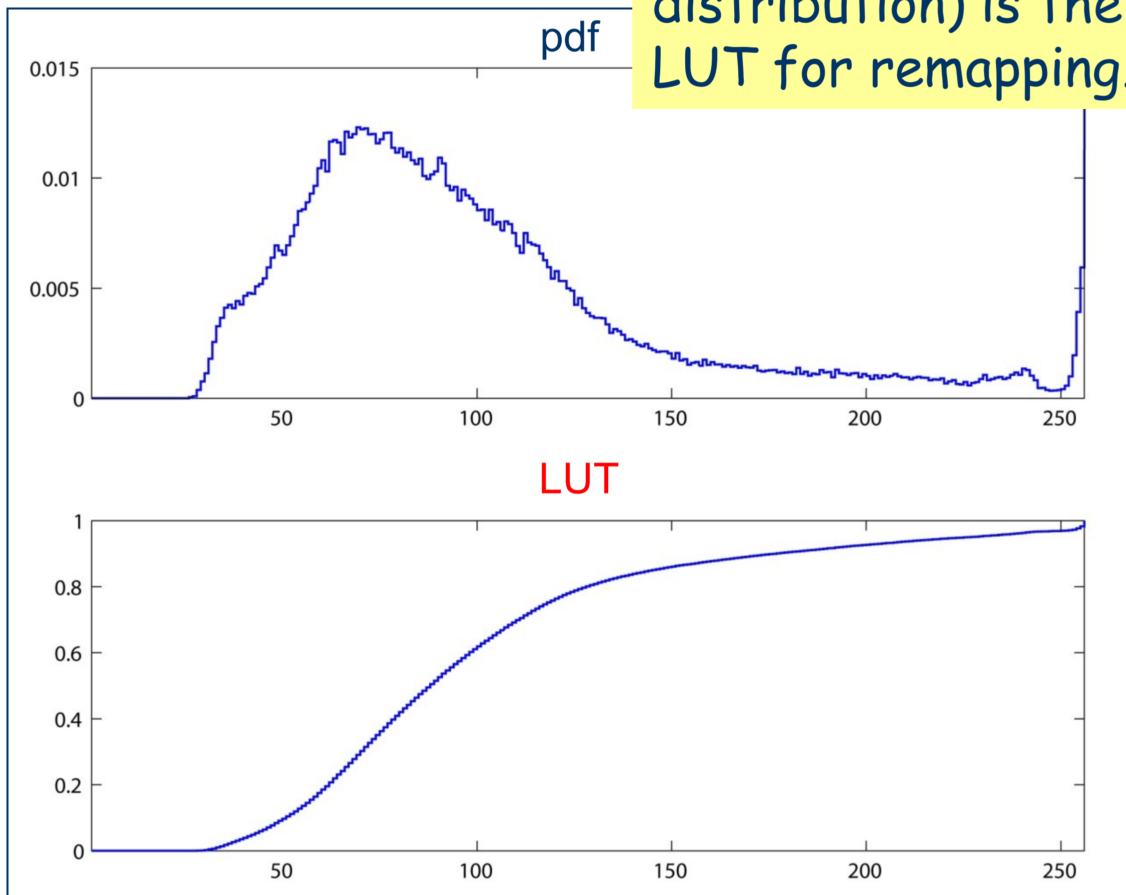
# Histogram Equalization

The CDF (cumulative distribution) is the LUT for remapping.



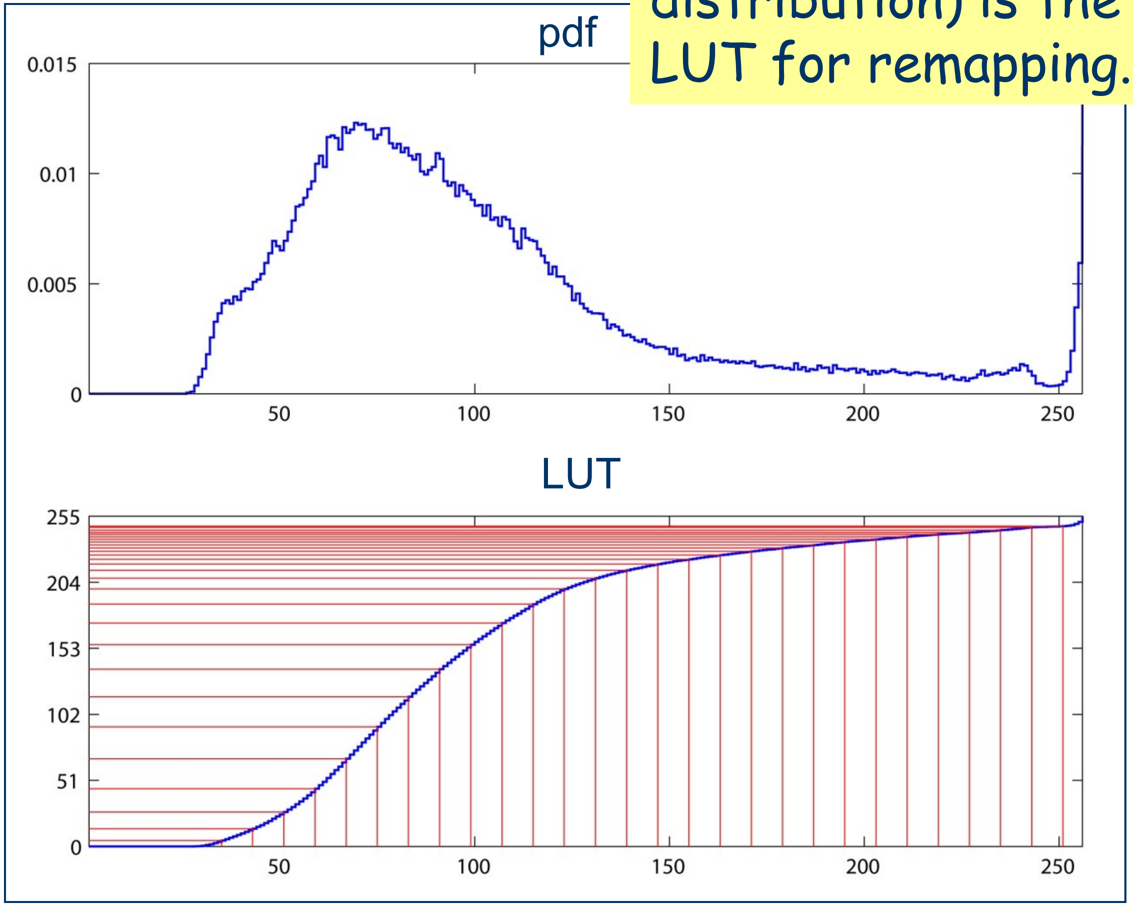
# Histogram Equalization

The CDF (cumulative distribution) is the LUT for remapping.

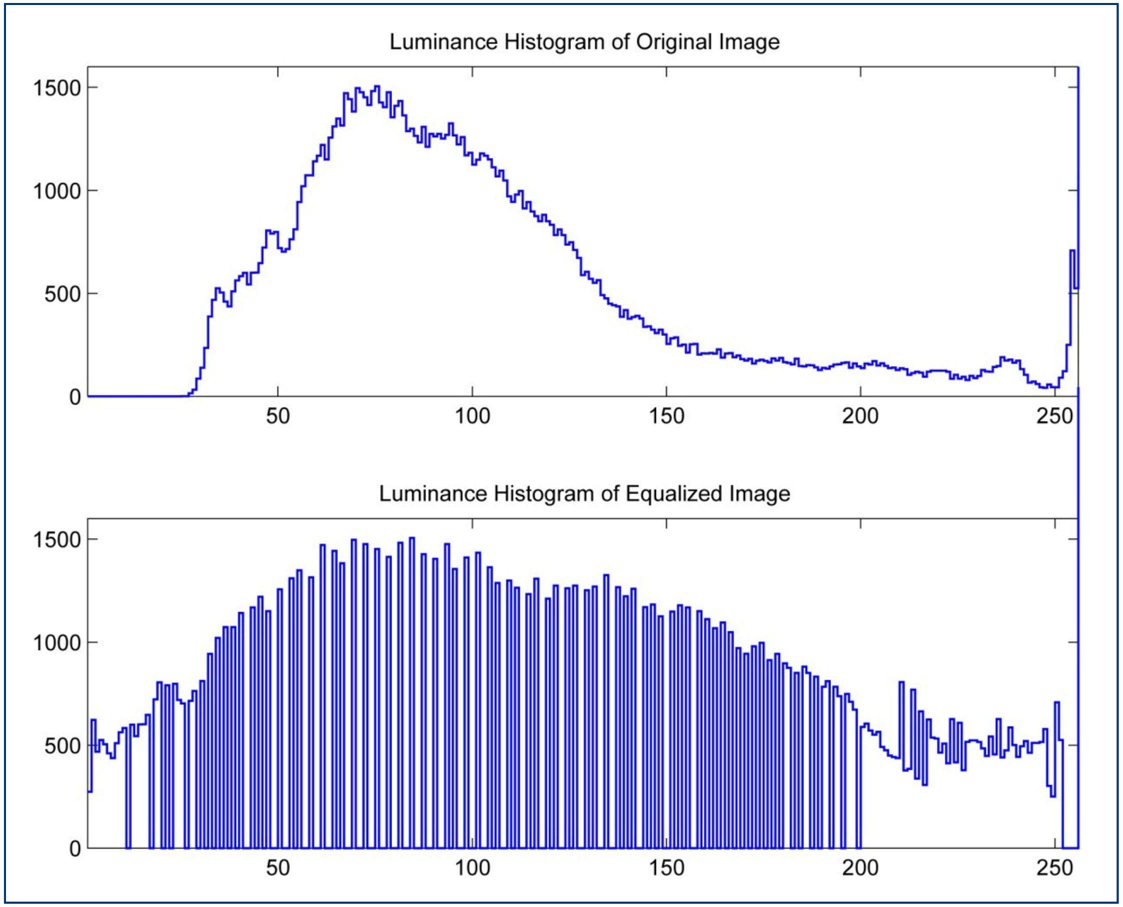


# Histogram Equalization

The CDF (cumulative distribution) is the LUT for remapping.

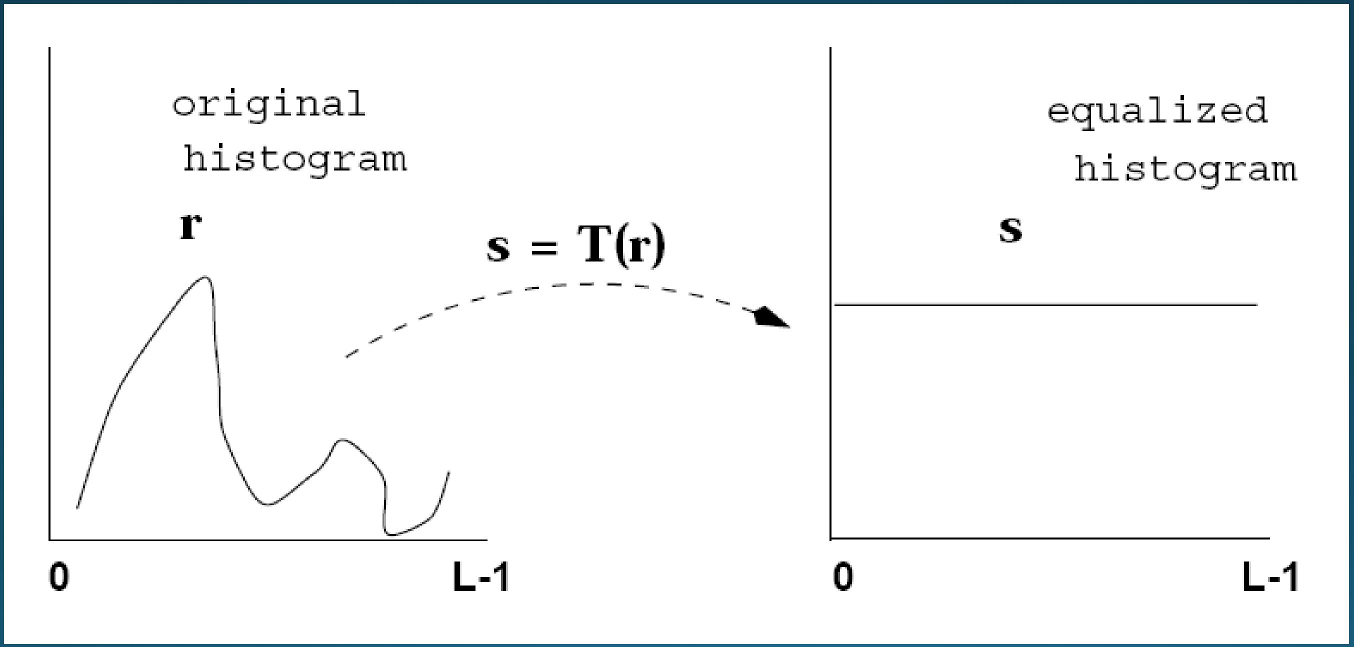


# Histogram Equalization



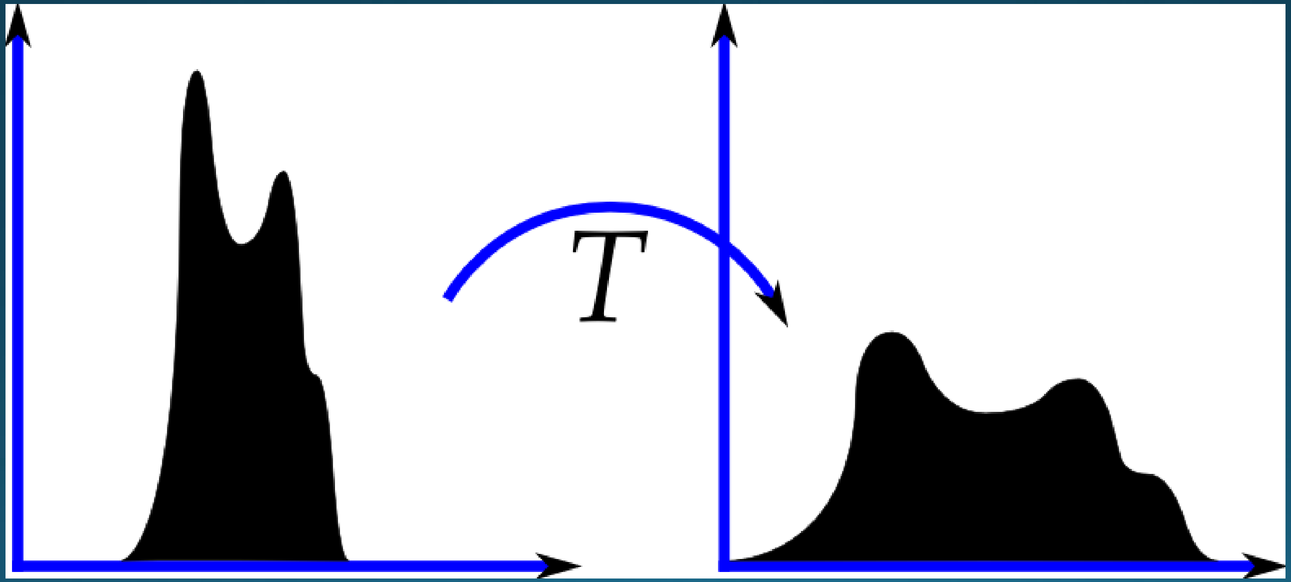
# Histogram Equalization

- The main idea is to redistribute the gray-level values uniformly.



# Histogram Equalization (cont'd)

- In practice, the equalized histogram might not be completely flat.

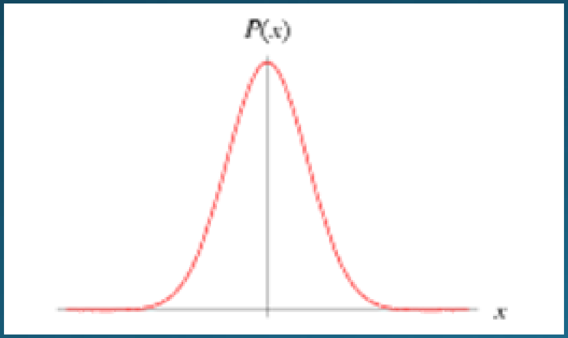


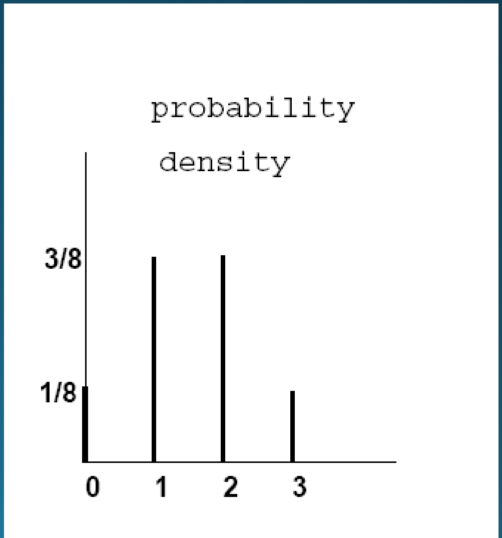
# Probability density function

- The *probability density function* (pdf) is a real-valued function $\mathbf{f_X(x)}$ describing the density of probability at each point in the sample space.

- In the discrete case, this is just a histogram!

Gaussian



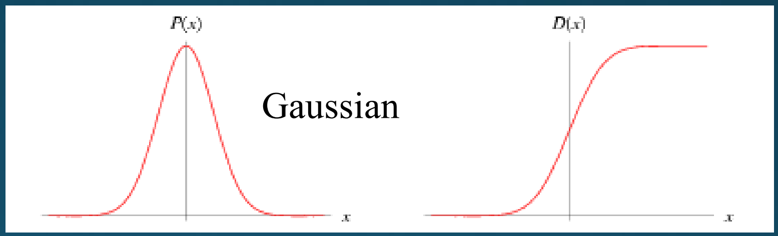



# Probability distribution function (CDF)

- The integral of $\mathbf{f_X(x)}$ defines the *probability distribution function* $\mathbf{F_X(x)}$ (i.e., cumulative probability→ cumulative distribution function "CDF")

$$F_X(x) = P(X \le x) = \int_{-\infty}^{x} f_X(a)da$$

$f_X(x)$ $\quad$ $F_X(x)$




non-decreasing

- In the discrete case, simply take the sum:

$$F_X(x) = P(X \le x) = \sum_{k=0}^{x} P(X = k)$$

# Probability distribution function (cont'd)

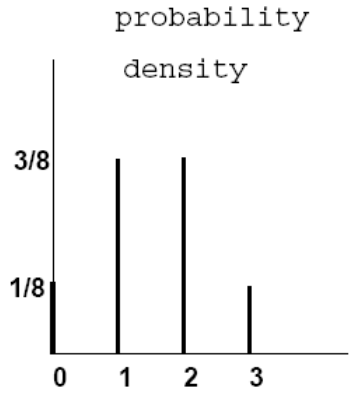
$$F_X(0) = P(X \leq 0) = P(X = 0) = 1/8$$

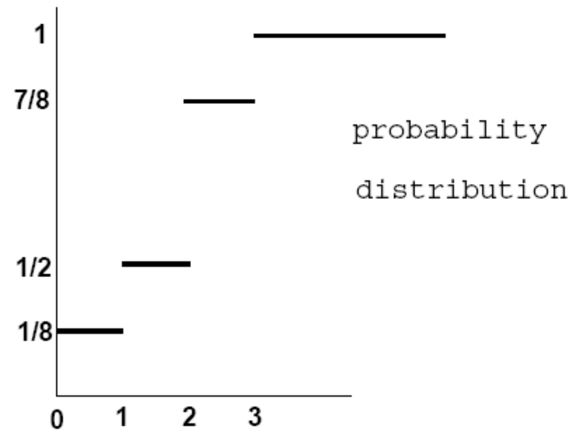
$$F_X(1) = P(X \leq 1) = P(X = 0) + P(X = 1) = 1/2$$

$$F_X(2) = P(X \leq 2) = P(X = 0) + P(X = 1) + P(X = 2) = 7/8$$

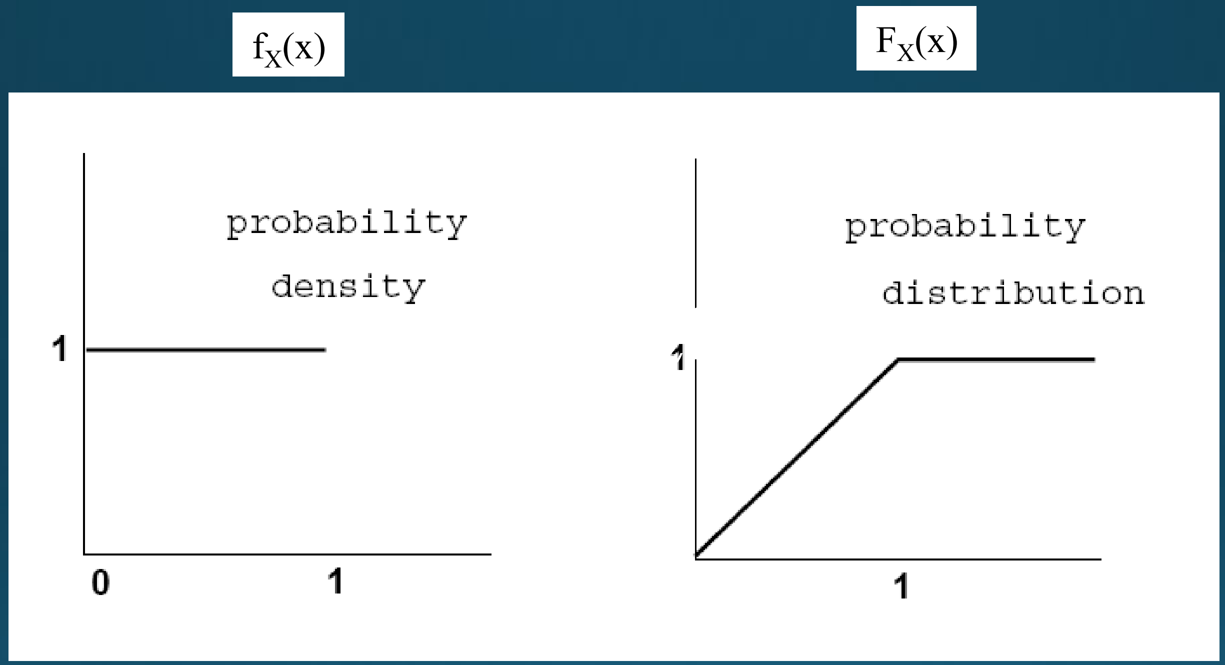
$$F_X(3) = P(X \leq 3) = P(X = 0) + P(X = 1) + P(X = 2) + P(X = 3) = 1$$

- If $X$ is continuous: $F_X(x) = P(X \leq x) = \int_{-\infty}^{x} f_x(a)da$



# Uniform Distribution



# Random Variable Transformation

- Suppose Y=T(X)
  - e.g., Y=X+1

- If we know **$f_X$(x)**, can we find **$f_Y$(y)**?

- Yes - it can be shown that:

$$f_Y(y) = [f_X(x)\ \frac{dX}{dY}] \text{ evaluated at } x = T^{-1}(y)$$

# Transformations of r.v. - Example

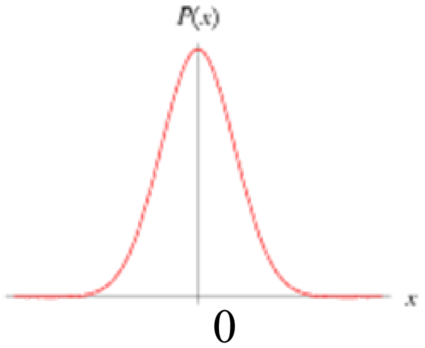
$$f_X(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$$

μ=0, σ=1

$$Y = X + 1 \qquad (Y = T(X))$$

$$X = Y - 1 \qquad (X = T^{-1}(Y))$$

$$\frac{dX}{dY} = 1$$




μ=1, σ=1

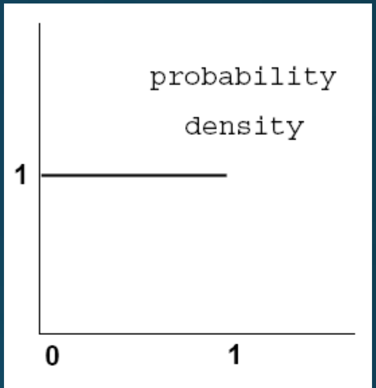
$$f_Y(Y) = [f_X(x)\ 1]\ (\text{evaluated at } x = y - 1) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(y-1)^2}{2}}$$

$$f_Y(y) = [f_X(x)\ \frac{dX}{dY}]\ \text{evaluated at } x = T^{-1}(y)$$

# *Special transformation!*

$$\text{If } Y = T(X) = F_X(x) = \int_{-\infty}^{x} f_X(a)\,da$$

$$\text{then, } f_Y(y) = f_U(u) \quad (f_U(u)\text{: uniform prob. density})$$



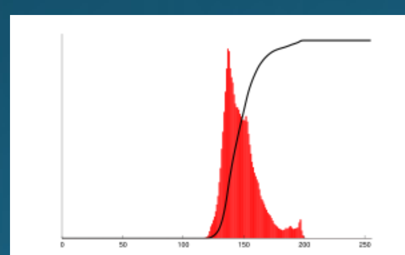

Proof: $$f_Y(y) = [f_X(x)\frac{dX}{dY}]_{x=T^{-1}(y)} = [f_X(x)\frac{1}{f_X(x)}]_{x=T^{-1}(y)} = 1$$

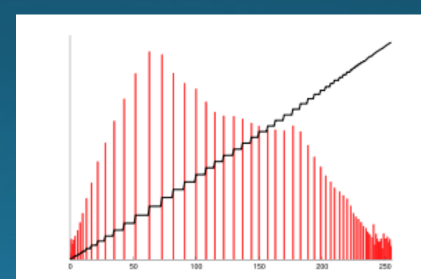
# Histogram Equalization (cont'd)

$T(r)$ can be obtained by discretizing $Y = T(X) = \int_{-\infty}^{x} f_X(a)da$

$$s_k = T(r_k) = \sum_{j=0}^{k} P_r(r_j) = \sum_{j=0}^{k} \frac{n_j}{n}$$

then, de-normalize:
$s_k$ x (L-1)





# So, Histogram equalization

1. T($r$) is single-valued and monotonically increasing in the interval [0,1], which preserves the order from black to white in the gray scale.
2. $0 \le T(r) \le 1$ for $0 \le r \le 1$, which guarantees the mapping is consistent with the allowed range of pixel values.

Where T(r) is:

$$s_k = \frac{L-1}{MN} \sum_{j=0}^{k} n_j$$

# Example:

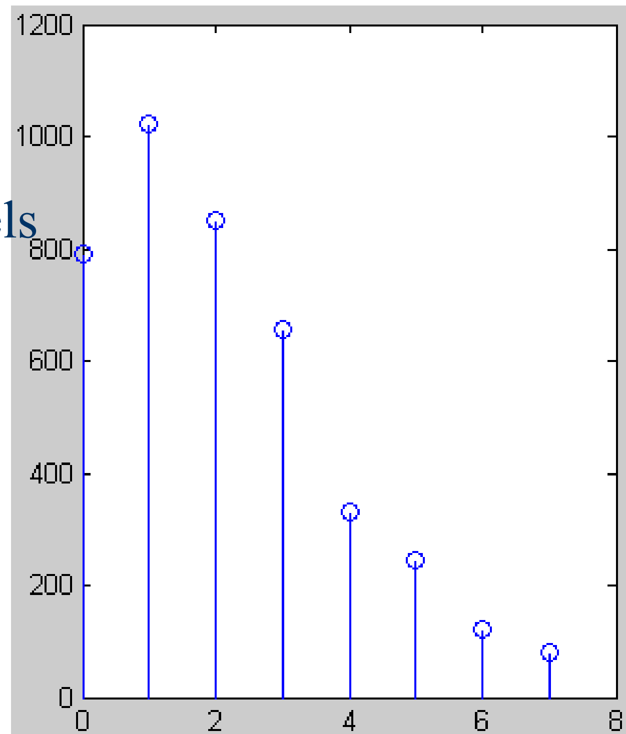
Consider an 8-level 64 x 64 image with gray values (0, 1, …, 7). The normalized gray values are (0, 1/7, 2/7, …, 1). The normalized histogram is given below:

| $k$ | $r_k$ | $n_k$ | $p(r_k) = n_k/n$ |
|---|---|---|---|
| 0 | 0 | 790 | 0.19 |
| 1 | 1/7 | 1023 | 0.25 |
| 2 | 2/7 | 850 | 0.21 |
| 3 | 3/7 | 656 | 0.16 |
| 4 | 4/7 | 329 | 0.08 |
| 5 | 5/7 | 245 | 0.06 |
| 6 | 6/7 | 122 | 0.03 |
| 7 | 1 | 81 | 0.02 |

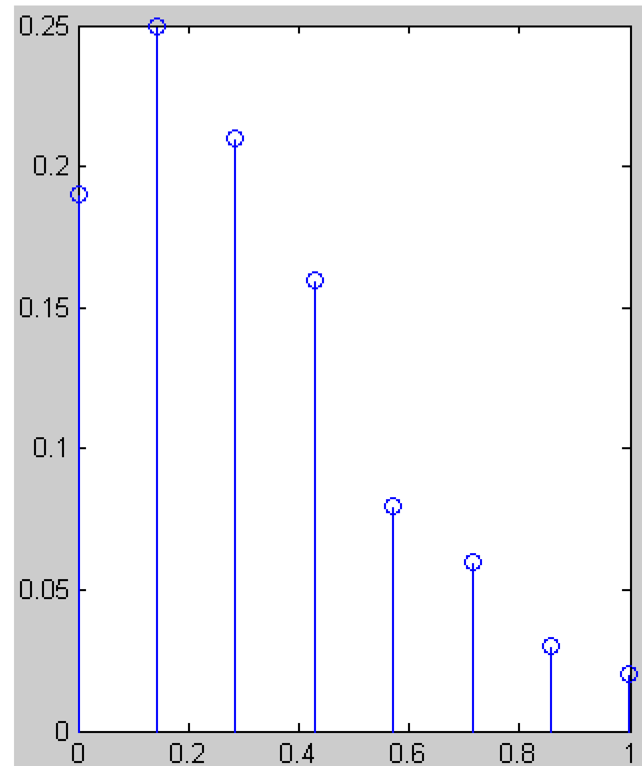
NB: The gray values in output are also (0, 1/7, 2/7, …, 1).

# Example:

# pixels




Fraction of # pixels




Normalized gray value

Applying the transformation, $s_k = T(r_k) = \sum_{j=0}^{k} p_{in}(r_j)$

we have

$$s_0 = T(r_0) = \sum_{j=0} p_{\text{in}}(r_j) = p_{\text{in}}(r_0) = 0.19 \rightarrow 1/7$$

$$s_1 = T(r_1) = \sum_{j=0}^{1} p_{\text{in}}(r_j) = p_{\text{in}}(r_0) + p_{\text{in}}(r_1) = 0.44 \rightarrow 3/7$$

$$s_2 = T(r_2) = \sum_{j=0}^{2} p_{\text{in}}(r_j) = p_{\text{in}}(r_0) + p_{\text{in}}(r_1) + p_{\text{in}}(r_2) = 0.65 \rightarrow 5/7$$

$$s_3 = T(r_3) = \sum_{j=0}^{3} p_{\text{in}}(r_j) = p_{\text{in}}(r_0) + p_{\text{in}}(r_1) + \cdots + p_{\text{in}}(r_3) = 0.81 \rightarrow 6/7$$

$$s_4 = T(r_4) = \sum_{j=0}^{4} p_{\text{in}}(r_j) = p_{\text{in}}(r_0) + p_{\text{in}}(r_1) + \cdots + p_{\text{in}}(r_4) = 0.89 \rightarrow 6/7$$

$$s_5 = T(r_5) = \sum_{j=0}^{5} p_{\text{in}}(r_j) = p_{\text{in}}(r_0) + p_{\text{in}}(r_1) + \cdots + p_{\text{in}}(r_5) = 0.95 \rightarrow 1$$

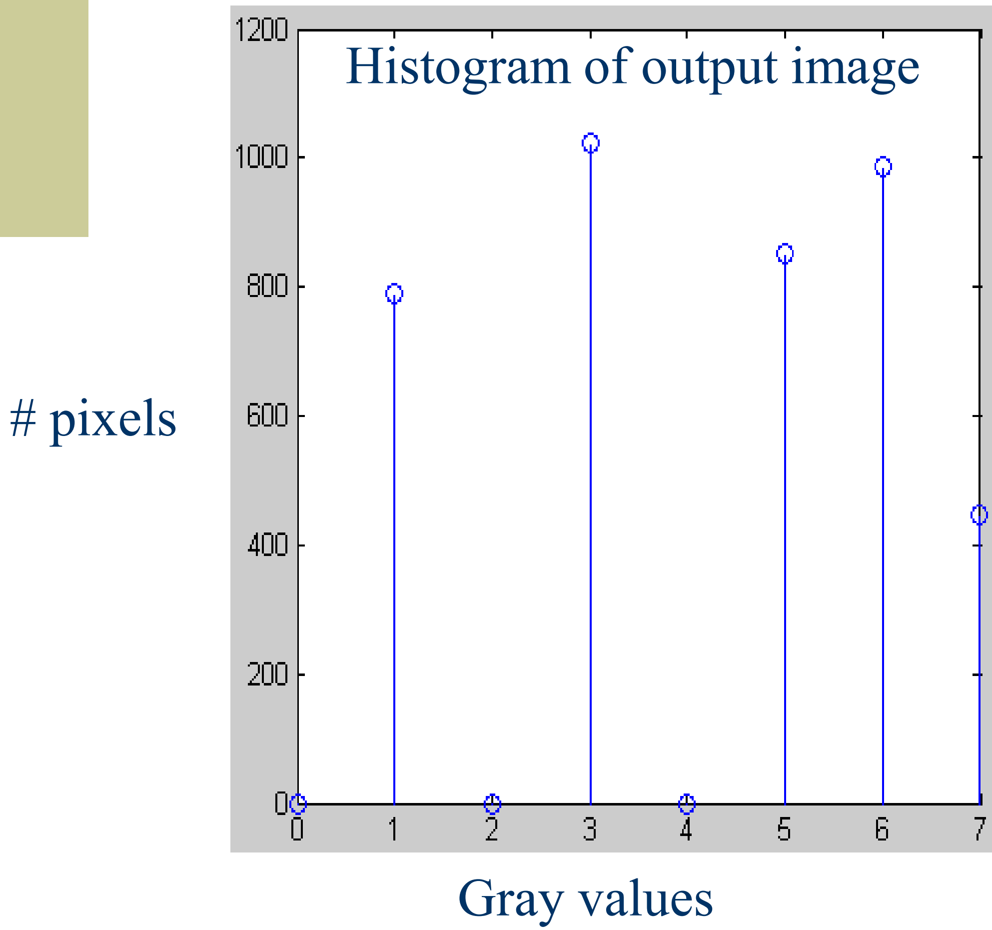
$$s_6 = T(r_6) = \sum_{j=0}^{6} p_{\text{in}}(r_j) = p_{\text{in}}(r_0) + p_{\text{in}}(r_1) + \cdots + p_{\text{in}}(r_6) = 0.98 \rightarrow 1$$

$$s_7 = T(r_7) = \sum_{j=0}^{7} p_{\text{in}}(r_j) = p_{\text{in}}(r_0) + p_{\text{in}}(r_1) + \cdots + p_{\text{in}}(r_7) = 1.00 \rightarrow 1$$

- Notice that there are only five distinct gray levels --- (1/7, 3/7, 5/7, 6/7, 1) in the output image. We will relabel them as ($s_0$, $s_1$, …, $s_4$ ).

- With this transformation, the output image will have histogram

| $k$ | $s_k$ | $n_k$ | $p(s_k) = n_k/n$ |
|---|---|---|---|
| 0 | 1/7 | 790 | 0.19 |
| 1 | 3/7 | 1023 | 0.25 |
| 2 | 5/7 | 850 | 0.21 |
| 3 | 6/7 | 985 | 0.24 |
| 4 | 1 | 448 | 0.11 |




```
>> clear
>> h=[0 790 0 1023 0 850 985 448];
>> stem(0:7,h)
```

- Note that the histogram of output image is only approximately, and not exactly, uniform. This should not be surprising, since there is no result that claims uniformity in the **discrete** case.

# Histogram Equalization: same example ☺

Suppose that a 3-bit image (L=8) of size 64 × 64 pixels (MN = 4096) has the intensity distribution shown in following table.
Get the histogram equalization transformation function and give the $p_s(s_k)$ for each $s_k$.

| $r_k$ | $n_k$ | $p_r(r_k) = n_k/MN$ |
|---|---|---|
| $r_0 = 0$ | 790 | 0.19 |
| $r_1 = 1$ | 1023 | 0.25 |
| $r_2 = 2$ | 850 | 0.21 |
| $r_3 = 3$ | 656 | 0.16 |
| $r_4 = 4$ | 329 | 0.08 |
| $r_5 = 5$ | 245 | 0.06 |
| $r_6 = 6$ | 122 | 0.03 |
| $r_7 = 7$ | 81 | 0.02 |

# Solution

| $r_k$ | $n_k$ | $p_r(r_k) = n_k/MN$ |
|---|---|---|
| $r_0 = 0$ | 790 | 0.19 |
| $r_1 = 1$ | 1023 | 0.25 |
| $r_2 = 2$ | 850 | 0.21 |
| $r_3 = 3$ | 656 | 0.16 |
| $r_4 = 4$ | 329 | 0.08 |
| $r_5 = 5$ | 245 | 0.06 |
| $r_6 = 6$ | 122 | 0.03 |
| $r_7 = 7$ | 81 | 0.02 |

$$\boxed{s_k = \frac{L-1}{MN}\sum_{j=0}^{k} n_j}$$

$$s_0 = T(r_0) = 7\sum_{j=0}^{0} p_r(r_j) = 7 \times 0.19 = 1.33 \quad \rightarrow 1$$

$$s_1 = T(r_1) = 7\sum_{j=0}^{1} p_r(r_j) = 7 \times (0.19 + 0.25) = 3.08 \quad \rightarrow 3$$

$$s_2 = 4.55 \quad \rightarrow 5 \qquad s_3 = 5.67 \quad \rightarrow 6$$

$$s_4 = 6.23 \quad \rightarrow 6 \qquad s_5 = 6.65 \quad \rightarrow 7$$

$$s_6 = 6.86 \quad \rightarrow 7 \qquad s_7 = 7.00 \quad \rightarrow 7$$

# Solution cont.

final transform:

$r_0 \rightarrow s_0 = 1 \Rightarrow 790$ pixels map to $1$

$r_1 \rightarrow s_1 = 3 \Rightarrow 1023$ pixels map to $3$

$r_2 \rightarrow s_2 = 5 \Rightarrow 850$ pixels map to $5$

$r_3 \rightarrow s_3 = 6 \Rightarrow 656 + 329 = 985$ pixels map to $6$

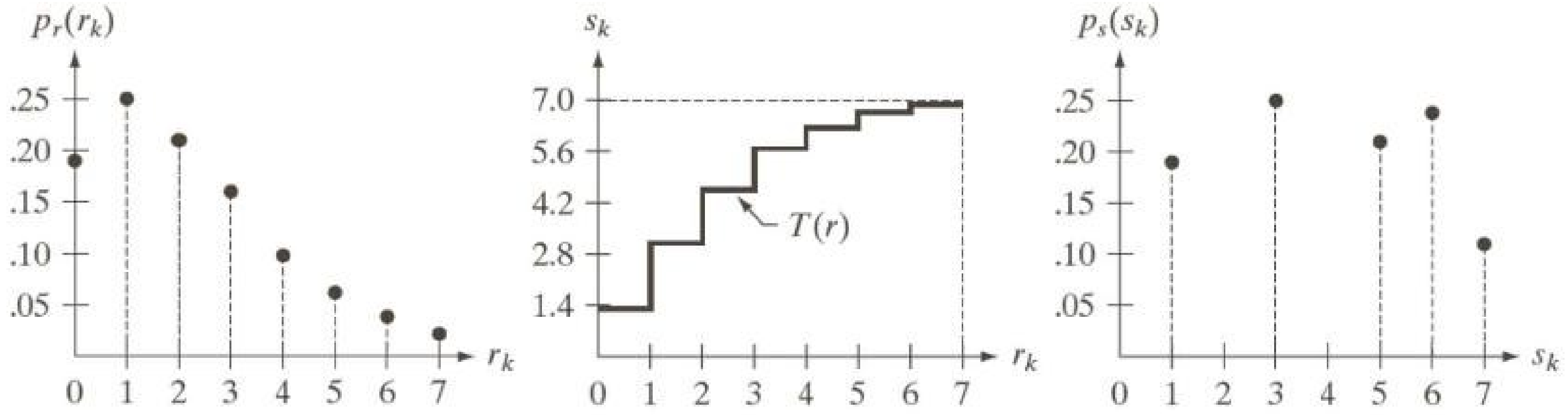
$r_4 \rightarrow s_4 = 6 \Rightarrow 656 + 329 = 985$ pixels map to $6$

$r_5 \rightarrow s_5 = 7 \Rightarrow 245 + 122 + 81 = 458$ pixels map to $7$

$r_6 \rightarrow s_6 = 7 \Rightarrow 245 + 122 + 81 = 458$ pixels map to $7$

$r_7 \rightarrow s_7 = 7 \Rightarrow 245 + 122 + 81 = 458$ pixels map to $7$

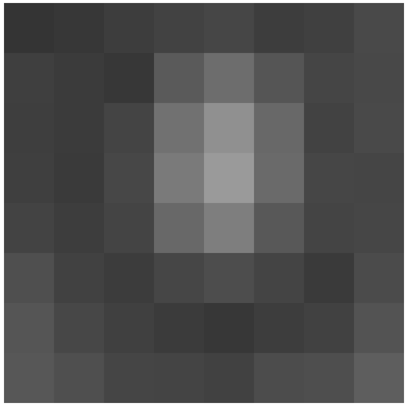
# Solution cont.




a b c

**FIGURE 3.19** Illustration of histogram equalization of a 3-bit (8 intensity levels) image. (a) Original histogram. (b) Transformation function. (c) Equalized histogram.

# Histogram Equalization: more practical Example



$$\begin{bmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 55 & 90 & 109 & 85 & 69 & 72 \\ 62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\ 67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\ 79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\ 87 & 79 & 69 & 68 & 65 & 76 & 78 & 94 \end{bmatrix}$$

An 8x8 image

# Histogram Equalization: Example

Fill in the following table/histogram

| Value | Count | Value | Count | Value | Count | Value | Count | Value | Count |
|---|---|---|---|---|---|---|---|---|---|
| 52 | | 64 | | 72 | | 85 | | 113 | |
| 55 | | 65 | | 73 | | 87 | | 122 | |
| 58 | | 66 | | 75 | | 88 | | 126 | |
| 59 | | 67 | | 76 | | 90 | | 144 | |
| 60 | | 68 | | 77 | | 94 | | 154 | |
| 61 | | 69 | | 78 | | 104 | | | |
| 62 | | 70 | | 79 | | 106 | | | |
| 63 | | 71 | | 83 | | 109 | | | |

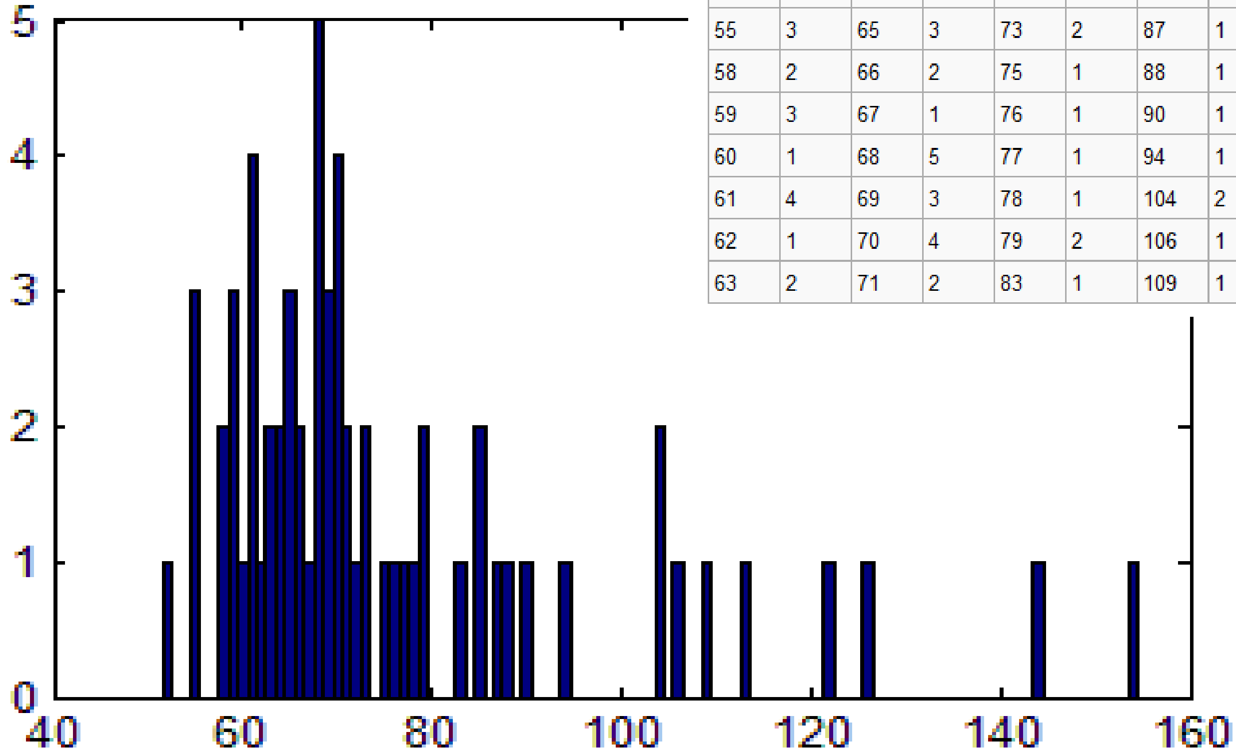
Image Histogram (Non-zero values)

# Histogram Equalization: Example

Image Histogram (Non-zero values shown)

| Value | Count | Value | Count | Value | Count | Value | Count | Value | Count |
|---|---|---|---|---|---|---|---|---|---|
| 52 | 1 | 64 | 2 | 72 | 1 | 85 | 2 | 113 | 1 |
| 55 | 3 | 65 | 3 | 73 | 2 | 87 | 1 | 122 | 1 |
| 58 | 2 | 66 | 2 | 75 | 1 | 88 | 1 | 126 | 1 |
| 59 | 3 | 67 | 1 | 76 | 1 | 90 | 1 | 144 | 1 |
| 60 | 1 | 68 | 5 | 77 | 1 | 94 | 1 | 154 | 1 |
| 61 | 4 | 69 | 3 | 78 | 1 | 104 | 2 | | |
| 62 | 1 | 70 | 4 | 79 | 2 | 106 | 1 | | |
| 63 | 2 | 71 | 2 | 83 | 1 | 109 | 1 | | |

# Histogram Equalization: Example




| Value | Count | Value | Count | Value | Count | Value | Count | Value | Count |
|---|---|---|---|---|---|---|---|---|---|
| 52 | 1 | 64 | 2 | 72 | 1 | 85 | 2 | 113 | 1 |
| 55 | 3 | 65 | 3 | 73 | 2 | 87 | 1 | 122 | 1 |
| 58 | 2 | 66 | 2 | 75 | 1 | 88 | 1 | 126 | 1 |
| 59 | 3 | 67 | 1 | 76 | 1 | 90 | 1 | 144 | 1 |
| 60 | 1 | 68 | 5 | 77 | 1 | 94 | 1 | 154 | 1 |
| 61 | 4 | 69 | 3 | 78 | 1 | 104 | 2 | | |
| 62 | 1 | 70 | 4 | 79 | 2 | 106 | 1 | | |
| 63 | 2 | 71 | 2 | 83 | 1 | 109 | 1 | | |

# Histogram Equalization: Example

## Cumulative Distribution Function (cdf)

| Value | cdf | Value | cdf | Value | cdf | Value | cdf | Value | cdf |
|---|---|---|---|---|---|---|---|---|---|
| 52 | | 64 | | 72 | | 85 | | 113 | |
| 55 | | 65 | | 73 | | 87 | | 122 | |
| 58 | | 66 | | 75 | | 88 | | 126 | |
| 59 | | 67 | | 76 | | 90 | | 144 | |
| 60 | | 68 | | 77 | | 94 | | 154 | |
| 61 | | 69 | | 78 | | 104 | | | |
| 62 | | 70 | | 79 | | 106 | | | |
| 63 | | 71 | | 83 | | 109 | | | |

## Image Histogram/Prob Mass Function

| Value | Count | Value | Count | Value | Count | Value | Count | Value | Count |
|---|---|---|---|---|---|---|---|---|---|
| 52 | 1 | 64 | 2 | 72 | 1 | 85 | 2 | 113 | 1 |
| 55 | 3 | 65 | 3 | 73 | 2 | 87 | 1 | 122 | 1 |
| 58 | 2 | 66 | 2 | 75 | 1 | 88 | 1 | 126 | 1 |
| 59 | 3 | 67 | 1 | 76 | 1 | 90 | 1 | 144 | 1 |
| 60 | 1 | 68 | 5 | 77 | 1 | 94 | 1 | 154 | 1 |
| 61 | 4 | 69 | 3 | 78 | 1 | 104 | 2 | | |
| 62 | 1 | 70 | 4 | 79 | 2 | 106 | 1 | | |
| 63 | 2 | 71 | 2 | 83 | 1 | 109 | 1 | | |

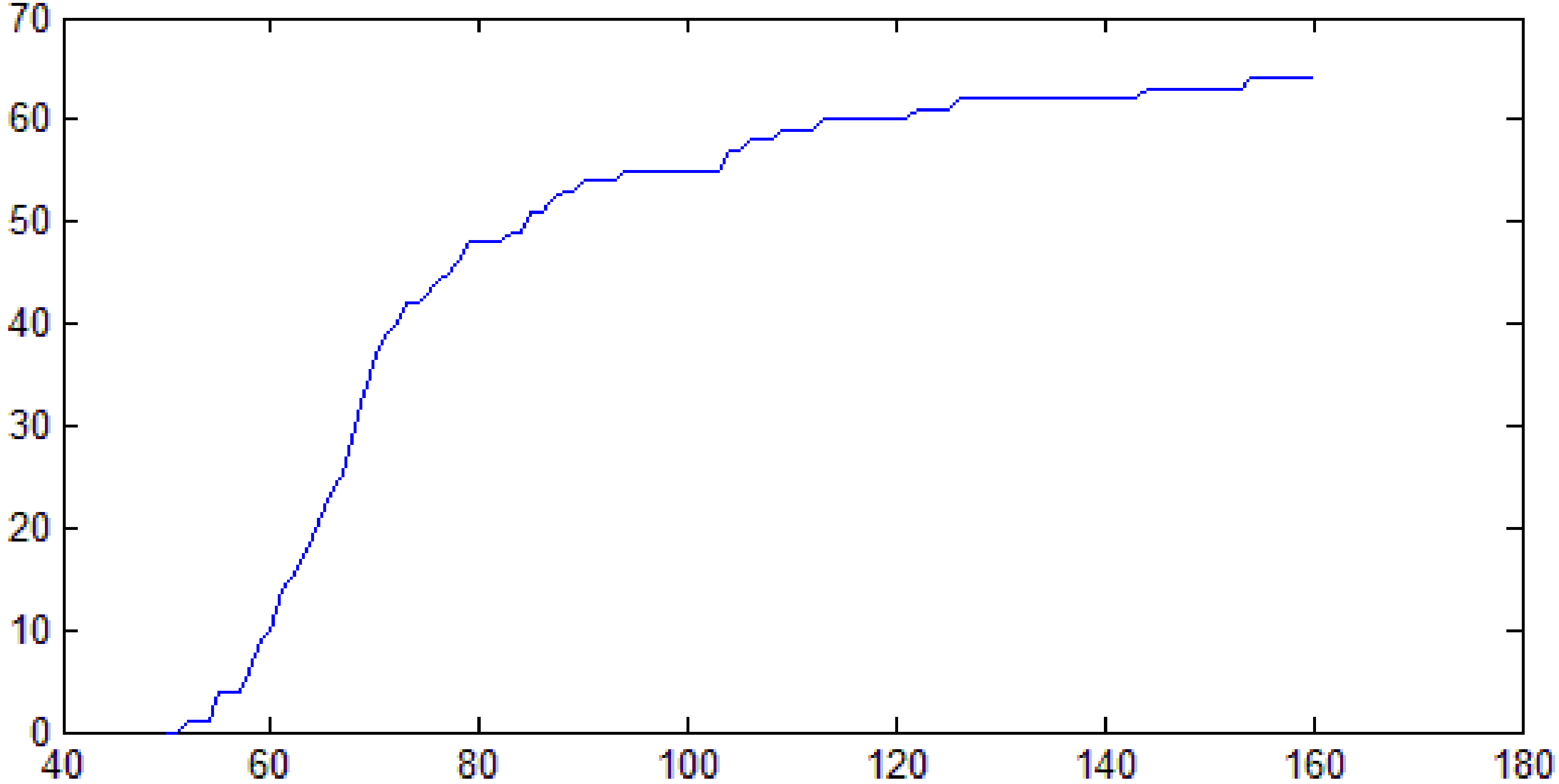
# Histogram Equalization: Example

## Cumulative Distribution Function (cdf)

| Value | cdf | Value | cdf | Value | cdf | Value | cdf | Value | cdf |
|---|---|---|---|---|---|---|---|---|---|
| 52 | 1 | 64 | 19 | 72 | 40 | 85 | 51 | 113 | 60 |
| 55 | 4 | 65 | 22 | 73 | 42 | 87 | 52 | 122 | 61 |
| 58 | 6 | 66 | 24 | 75 | 43 | 88 | 53 | 126 | 62 |
| 59 | 9 | 67 | 25 | 76 | 44 | 90 | 54 | 144 | 63 |
| 60 | 10 | 68 | 30 | 77 | 45 | 94 | 55 | 154 | 64 |
| 61 | 14 | 69 | 33 | 78 | 46 | 104 | 57 | | |
| 62 | 15 | 70 | 37 | 79 | 48 | 106 | 58 | | |
| 63 | 17 | 71 | 39 | 83 | 49 | 109 | 59 | | |

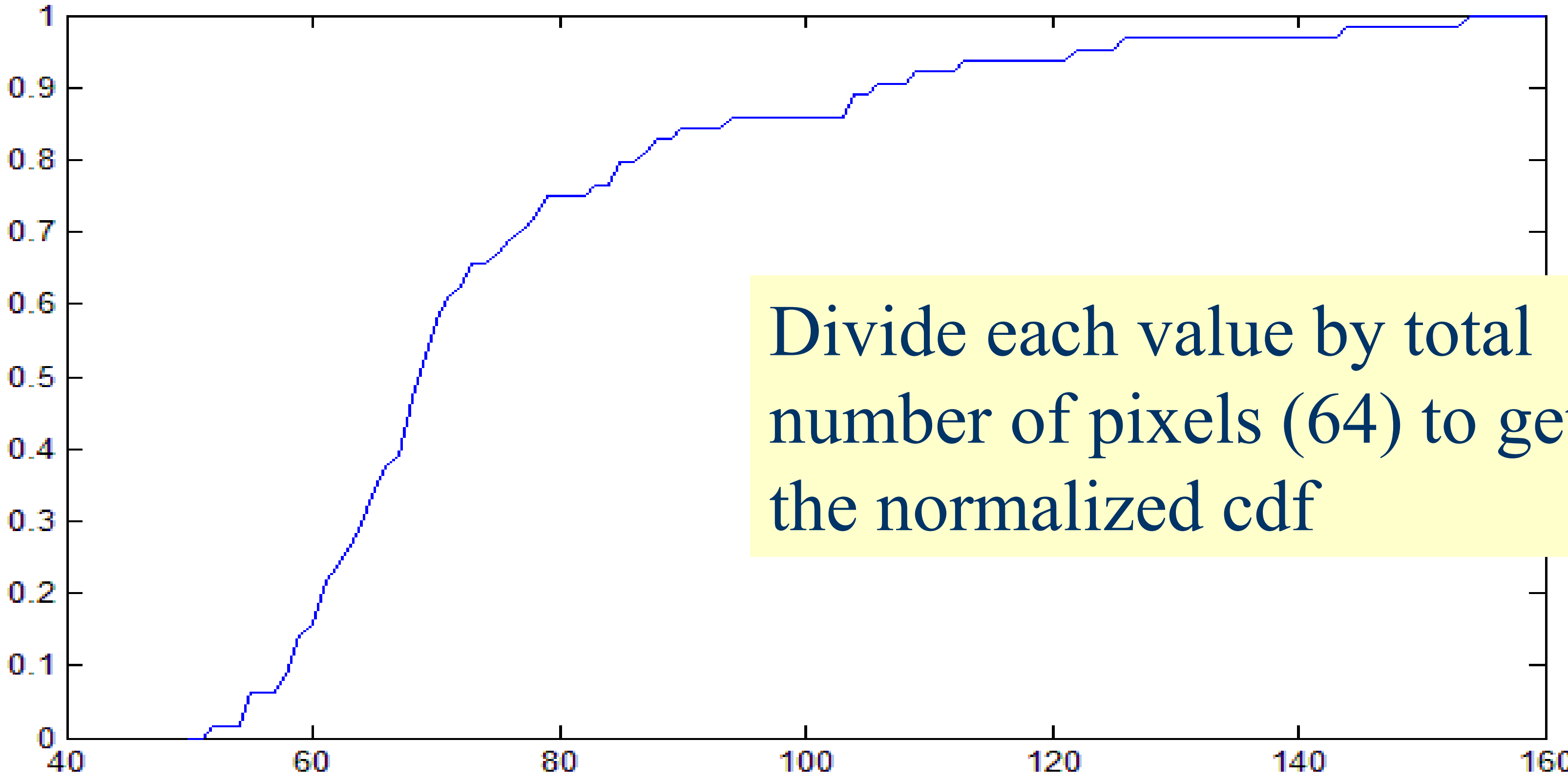
# Histogram Equalization: Example

Cumulative Distribution Function (cdf)



# Histogram Equalization: Example

Normalized Cumulative Distribution Function (cdf)




Divide each value by total number of pixels (64) to get the normalized cdf

# Histogram Equalization: Example

| Value | cdf | Value | cdf | Value | cdf | Value | cdf | Value | cdf |
|---|---|---|---|---|---|---|---|---|---|
| 52 | 1 | 64 | 19 | 72 | 40 | 85 | 51 | 113 | 60 |
| 55 | 4 | 65 | 22 | 73 | 42 | 87 | 52 | 122 | 61 |
| 58 | 6 | 66 | 24 | 75 | 43 | 88 | 53 | 126 | 62 |
| 59 | 9 | 67 | 25 | 76 | 44 | 90 | 54 | 144 | 63 |
| 60 | 10 | 68 | 30 | 77 | 45 | 94 | 55 | 154 | 64 |
| 61 | 14 | 69 | 33 | 78 | 46 | 104 | 57 | | |
| 62 | 15 | 70 | 37 | 79 | 48 | 106 | 58 | | |
| 63 | 17 | 71 | 39 | 83 | 49 | 109 | 59 | | |

$$J(r,c) = 255 \cdot P_I\left[I(r,c)\right].$$

If cdf is normalized

$$s = round(255.cdf(r))$$

If cdf is NOT normalized

$$s = round(255.\frac{cdf(r)}{M \times N})$$

$$s = round(255.(46/64))$$

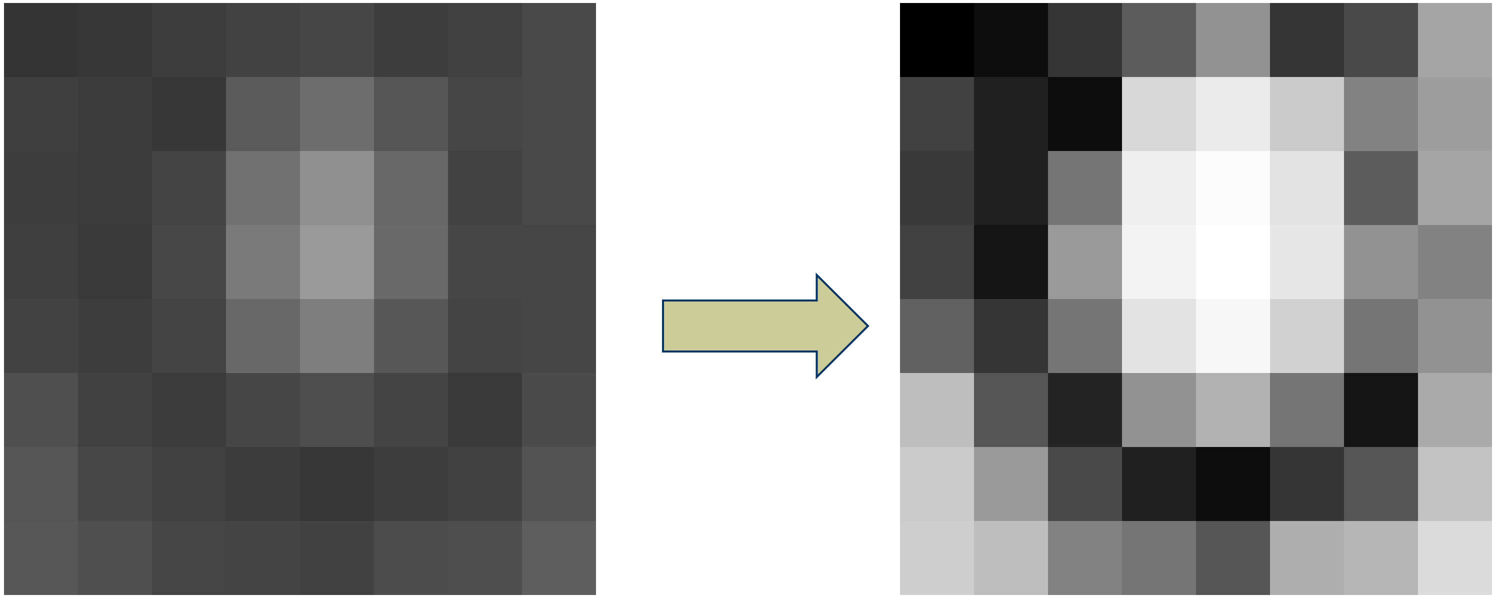
$$s = 183$$

$$\begin{bmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 55 & 90 & 109 & 85 & 69 & 72 \\ 62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\ 67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\ 79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\ 87 & 79 & 69 & 68 & 65 & 76 & 78 & 94 \end{bmatrix}$$

Original Image

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | 183 | |

# Histogram Equalization: Example



# Matlab Examples

histogramsEq.m:

for an equalization example

And

histograms1.m: this shows the concept of the color histogram used in many applications like image retrieval

# Local Histogram Processing

- Histogram equalization/specification are **global** methods.
  - The intensity transformation is computed using pixels from the entire image.
- Global transformations are not appropriate for enhancing little details in an image.
  - The number of pixels in these areas might be very small, contributing very little to the computation of the transformation.

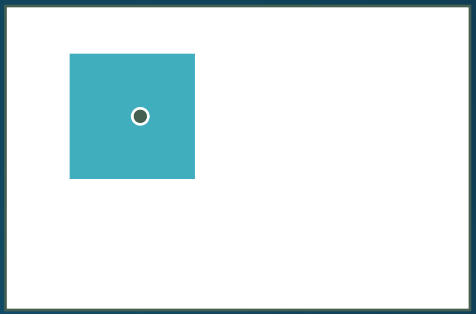
# Local Histogram Processing

## Idea:

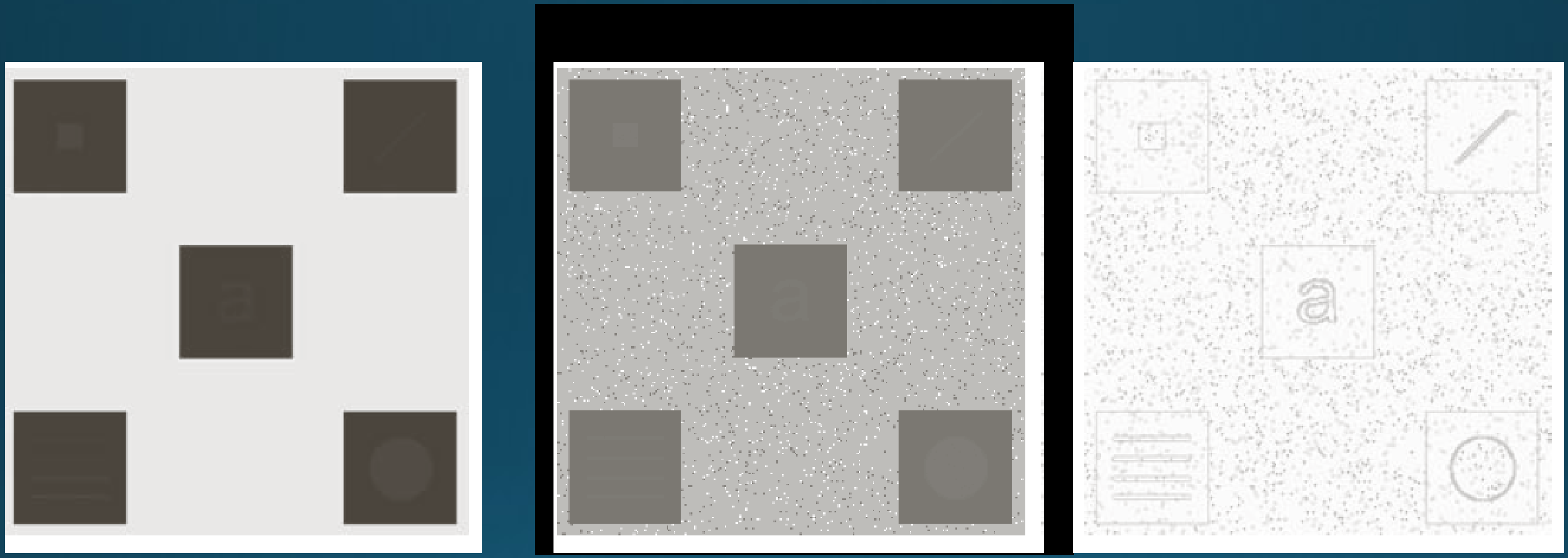
Define a transformation function based on the intensity distribution in a neighborhood of every pixel in the image!

# Local Histogram Processing (cont'd)

1. Define a neighborhood and move its center from pixel to pixel.
2. At each location, the histogram of the points in the neighborhood is computed. Obtain histogram equalization or histogram specification transformation.
3. Map the intensity of the pixel <u>centered</u> in the neighborhood.
4. Move to the next location and repeat the procedure.



# Local Histogram Processing: Example




global histogram equalization

local histogram equalization 3 x 3 neighborhood

# Histogram Statistics

Mean
(average intensity)

$$m = \sum_{i=0}^{L-1} r_i p(r_i) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y)$$

n-th moment
around mean

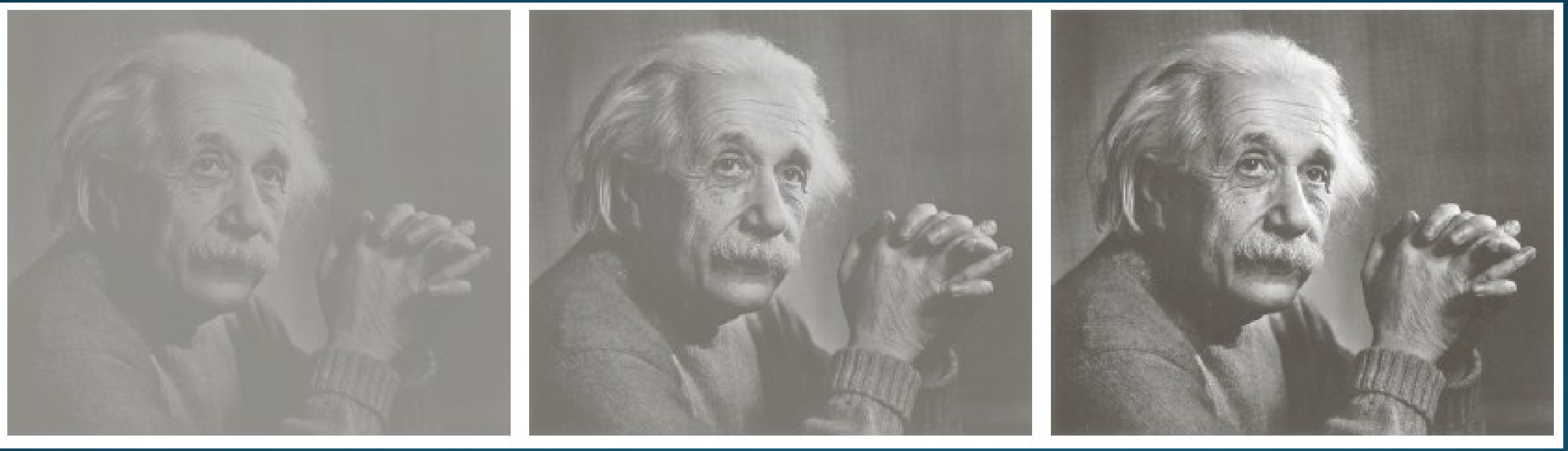
$$\mu_n(r) = \sum_{i=0}^{L-1} (r_i - m)^n p(r_i)$$

Variance
(2nd moment)

$$\sigma^2 = \mu_2(r) = \sum_{i=0}^{L-1} (r_i - m)^2 p(r_i) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [f(x,y) - m]^2$$

# Example: Comparison of Standard Deviation Values

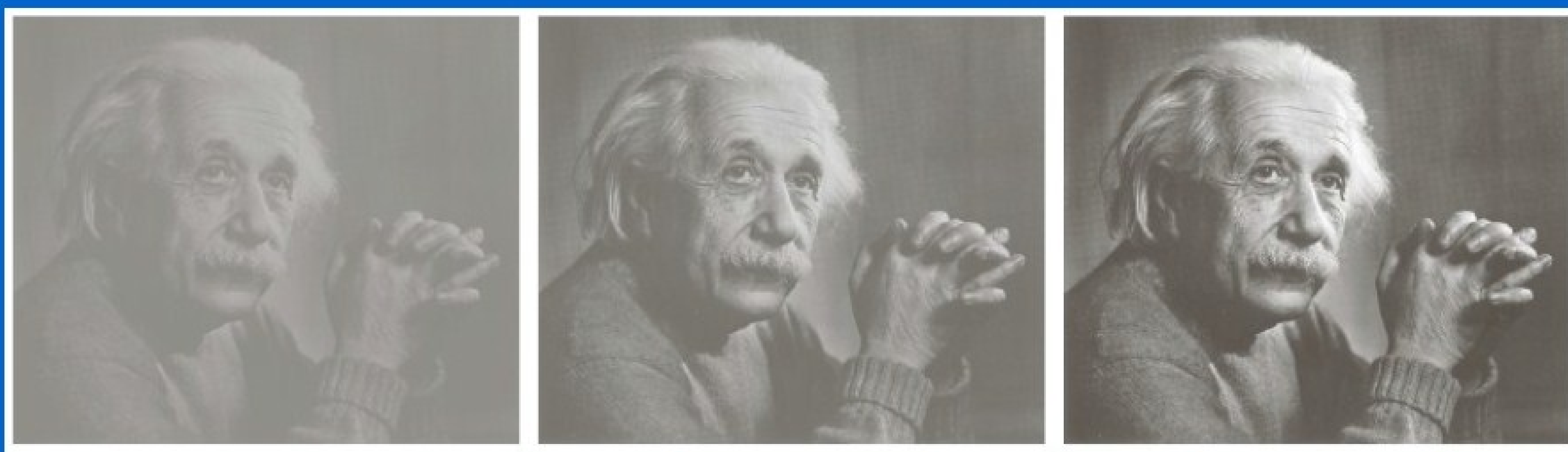
σ is useful for estimating image contrast!



$$\sigma = 14.3$$

# Example: Using Histogram Statistics for Estimating Image Contrast

σ is useful for estimating image contrast!



$\sigma = 14.3$ $\sigma = 31.6$ $\sigma = 49.2$

# Local Histogram Statistics

- Compute image histogram statistics in a local region.

Local average intensity

$$m_{s_{xy}} = \sum_{i=0}^{L-1} r_i p_{s_{xy}}(r_i)$$

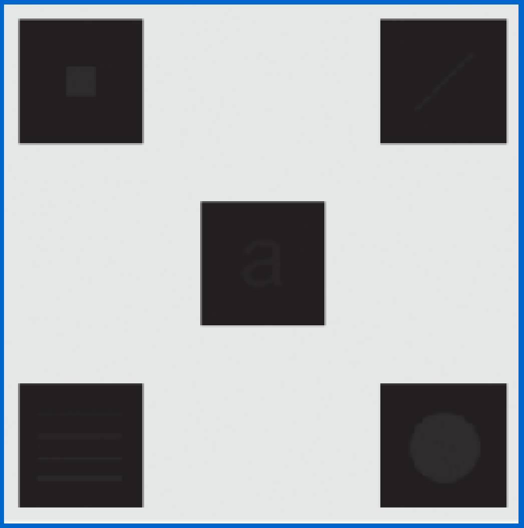
$s_{xy}$ denotes a neighborhood

Local variance

$$\sigma^2_{s_{xy}} = \sum_{i=0}^{L-1} (r_i - m_{s_{xy}})^2 p_{s_{xy}}(r_i)$$

# Example: Using Local Histogram Statistics for Image Enhancement

- Could be useful for local image enhancement.



Task: enhance dark areas without affecting much bright areas.

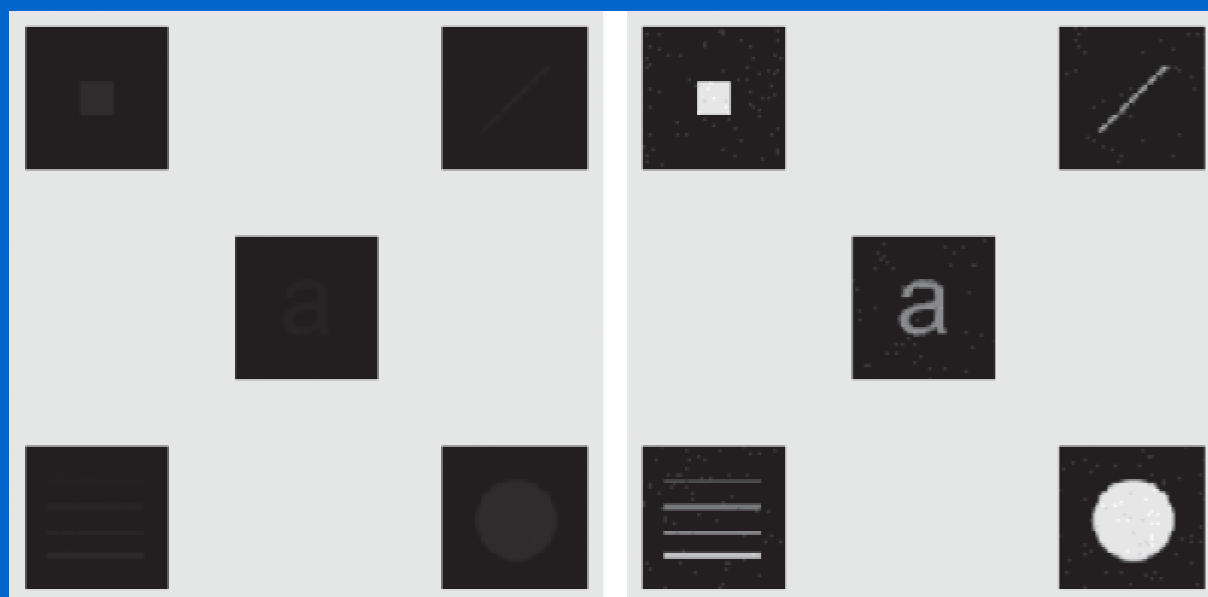
Idea: Find dark, low contrast areas using local statistics and enhance them.

# Example (cont'd)

$$g(x,y)=\begin{cases} \text{Eq if} \quad m_{s_{xy}} \le k_0 m_G \text{ and } k_1\sigma_G \le \sigma_{s_{xy}} \le k_2\sigma_G \\ f(x,y), & \text{otherwise} \end{cases}$$

$m_G$ : global mean; $\sigma_G$ : global standard deviation

$m_G=161$ $\sigma_G=103$ $E=22.8$ $k_0=0.25$ $k_1=0$ $k_2=0.1$



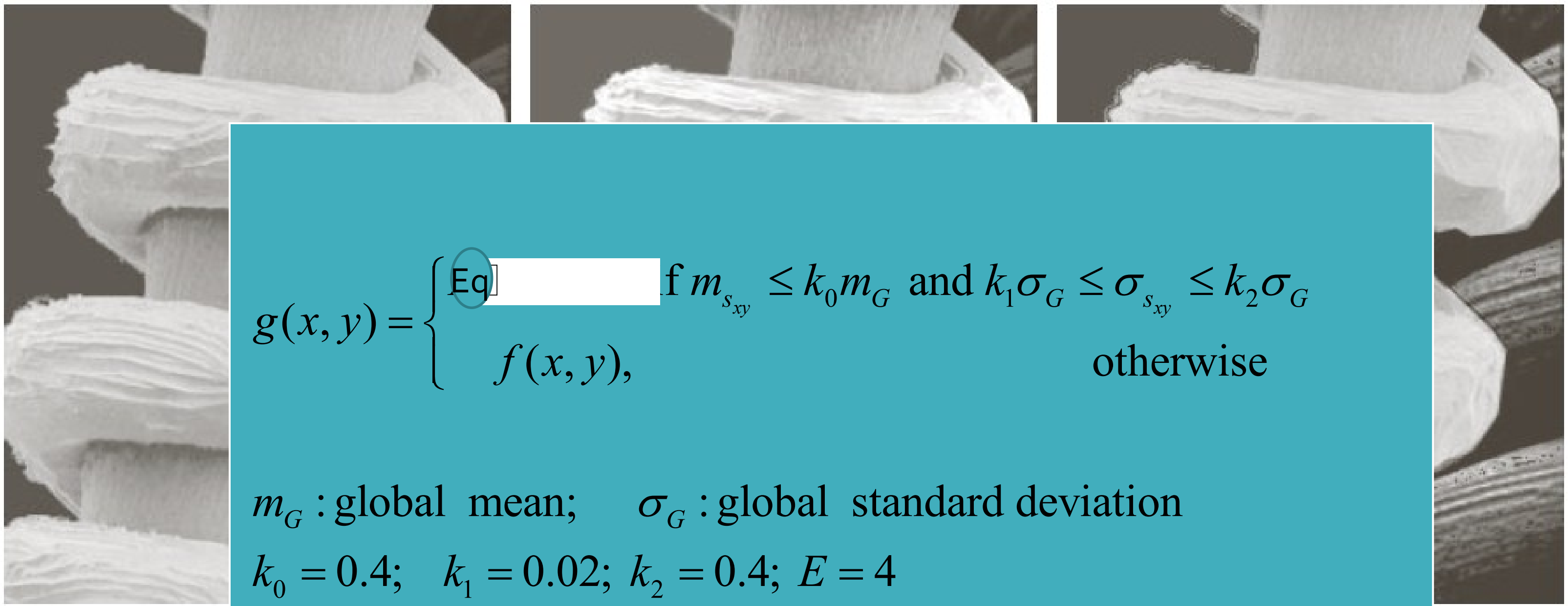
# Using Histogram Statistics for Image Enhancement

- Useful when parts of the image might contain hidden features.

<u>Task:</u> enhance dark areas without changing bright areas.

<u>Idea:</u> Find dark, low contrast areas using local statistics.

# Using Histogram Statistics for Image Enhancement: 2nd Example



$$g(x,y)=\begin{cases}\text{Eq}\qquad\qquad \text{f}\ m_{s_{xy}} \le k_0 m_G \ \text{ and } k_1\sigma_G \le \sigma_{s_{xy}} \le k_2\sigma_G \\ f(x,y), \qquad\qquad\qquad\qquad\qquad\qquad \text{otherwise}\end{cases}$$

$m_G$ : global mean; $\sigma_G$ : global standard deviation

$k_0 = 0.4;\quad k_1 = 0.02;\ k_2 = 0.4;\ E = 4$

a b c

**FIGURE 3.27** 130×.
(b) Result of togram
statistics. (Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene.)