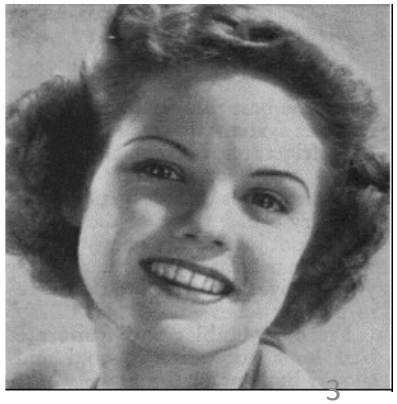
# CSC 447
# Digital Image Processing

Dr. Salma Hamdy
s.hamdy@cis.asu.edu.eg

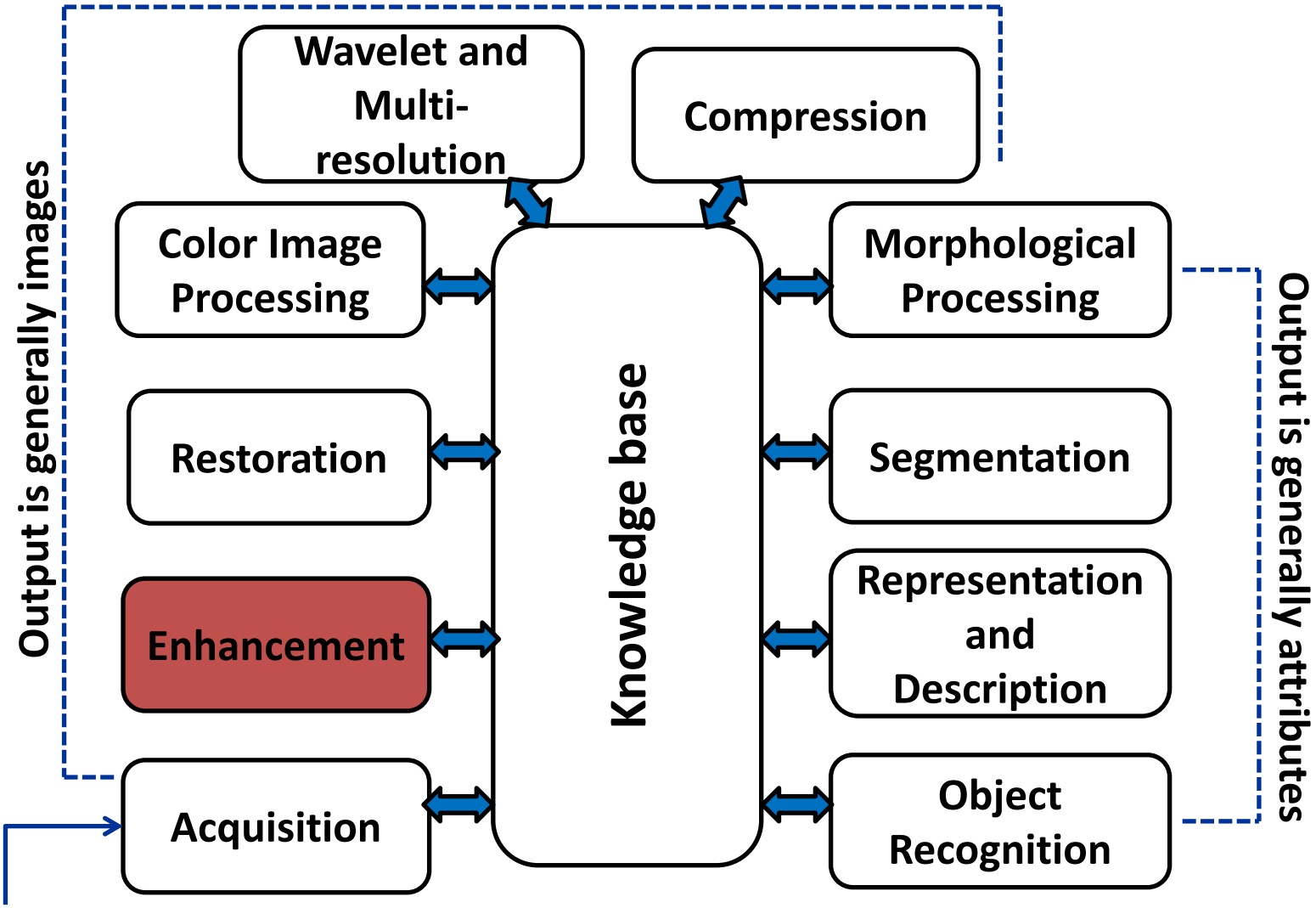
# Recall

1. Image digitization
2. Electromagnetic spectrum
3. Sampling and quantization
4. Color filter arrays
5. Image resolution

# Image Enhancement I



# Fundamental Steps of DIP



# Contents

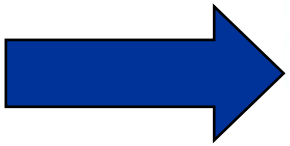
# What is Image Enhancement?

- **Image enhancement is the process of making images more useful visually.**

- **Reasons for doing this include:**
  - **Highlighting interesting detail in images.**
  - **Making images more visually appealing.**
    **Removing noise from images.**

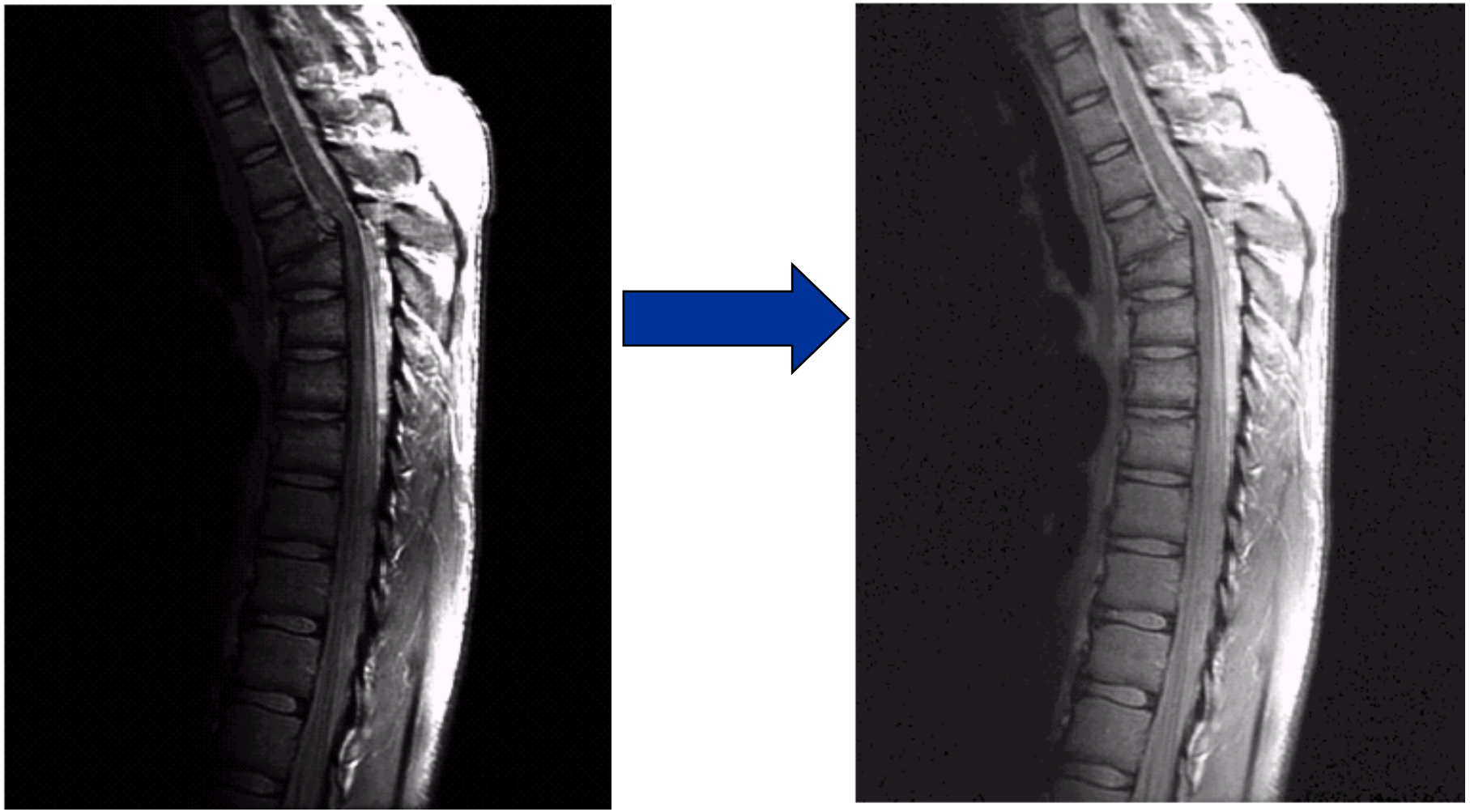
# Examples

Blurred & noisy image

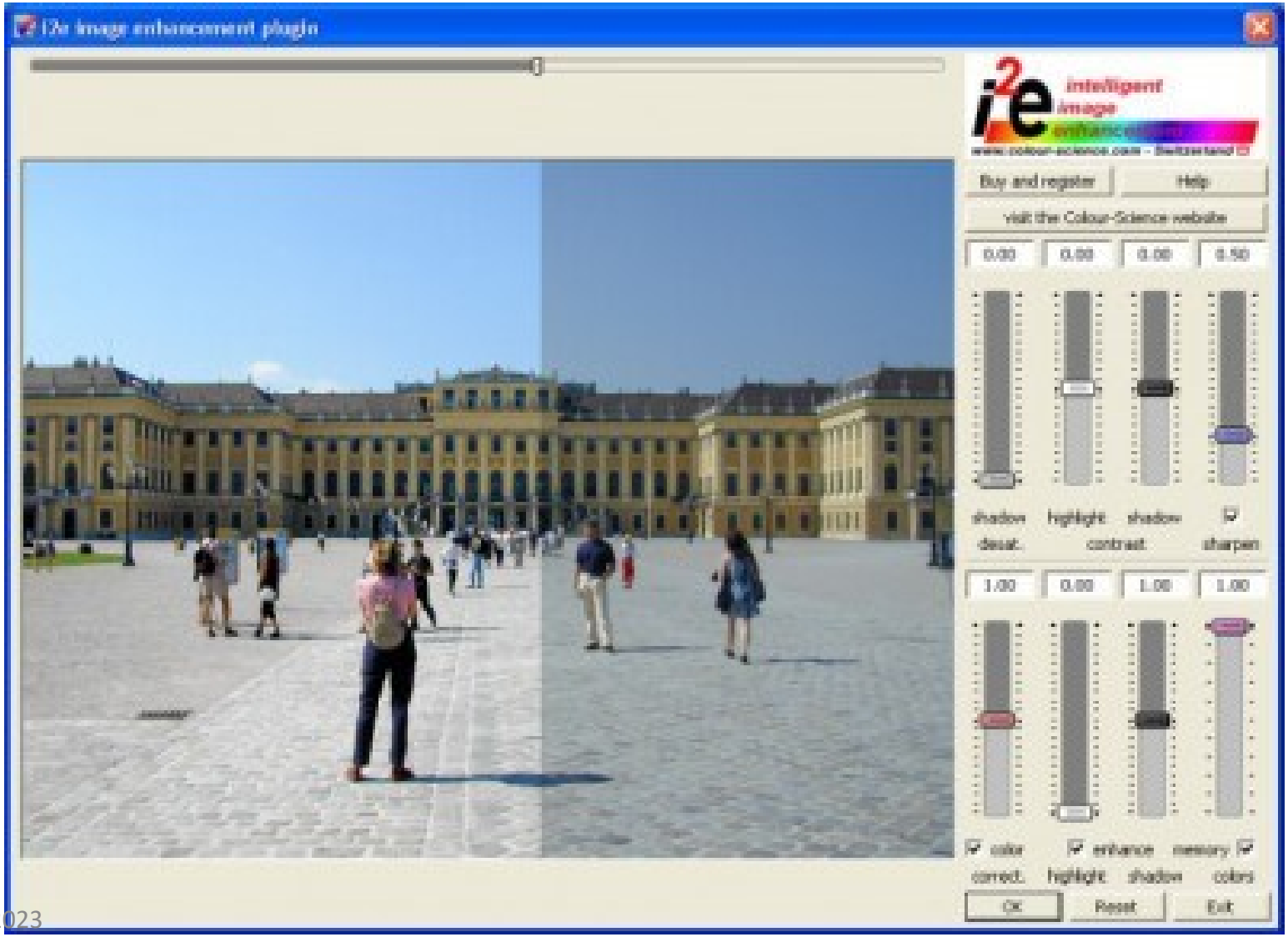


Restored image

# Examples – (cont.)

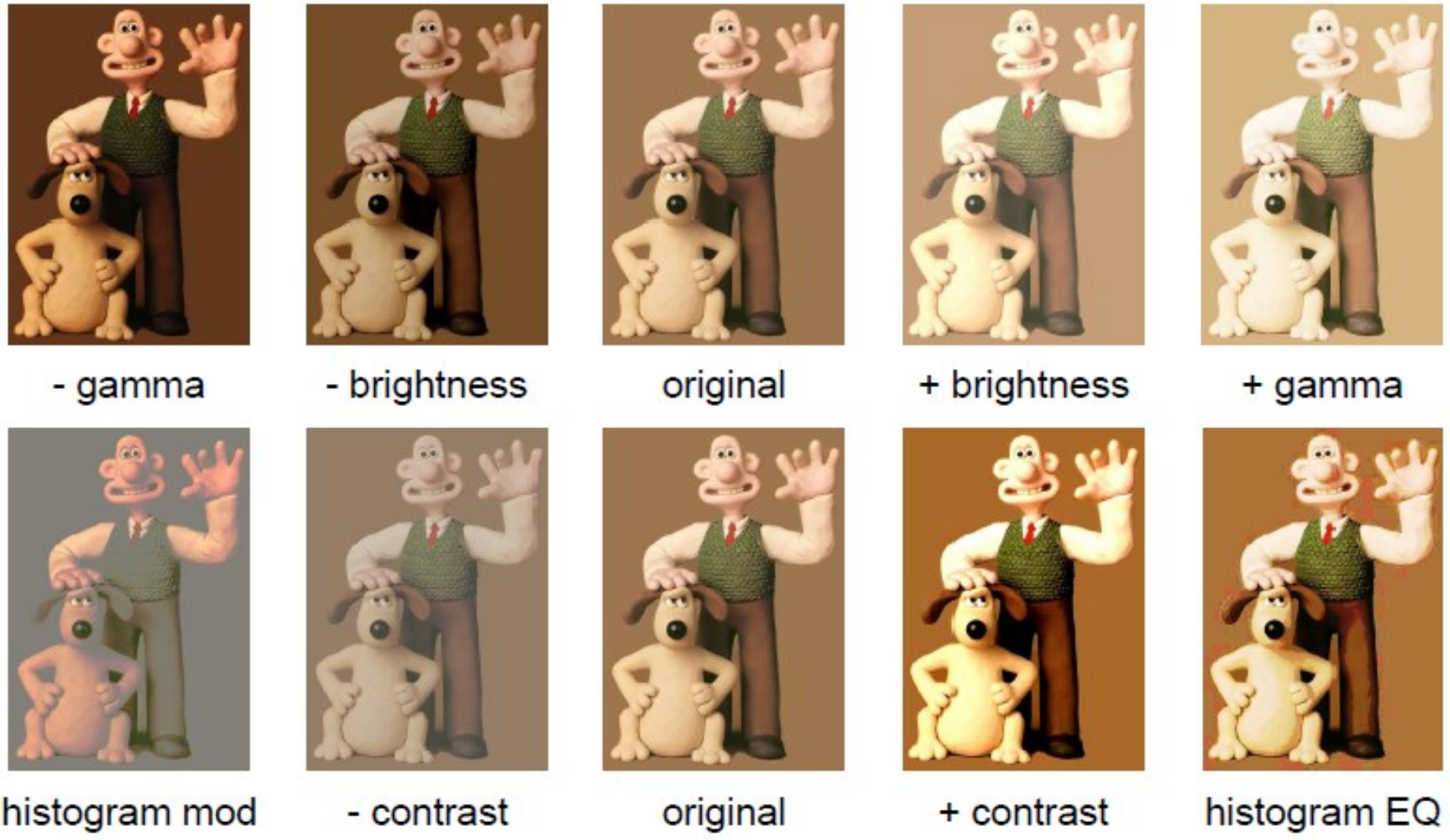


# Examples – (cont.)



# Examples – (cont.)

# Examples – (cont.)



# Examples – (cont.) Restoration

periodic noise

original

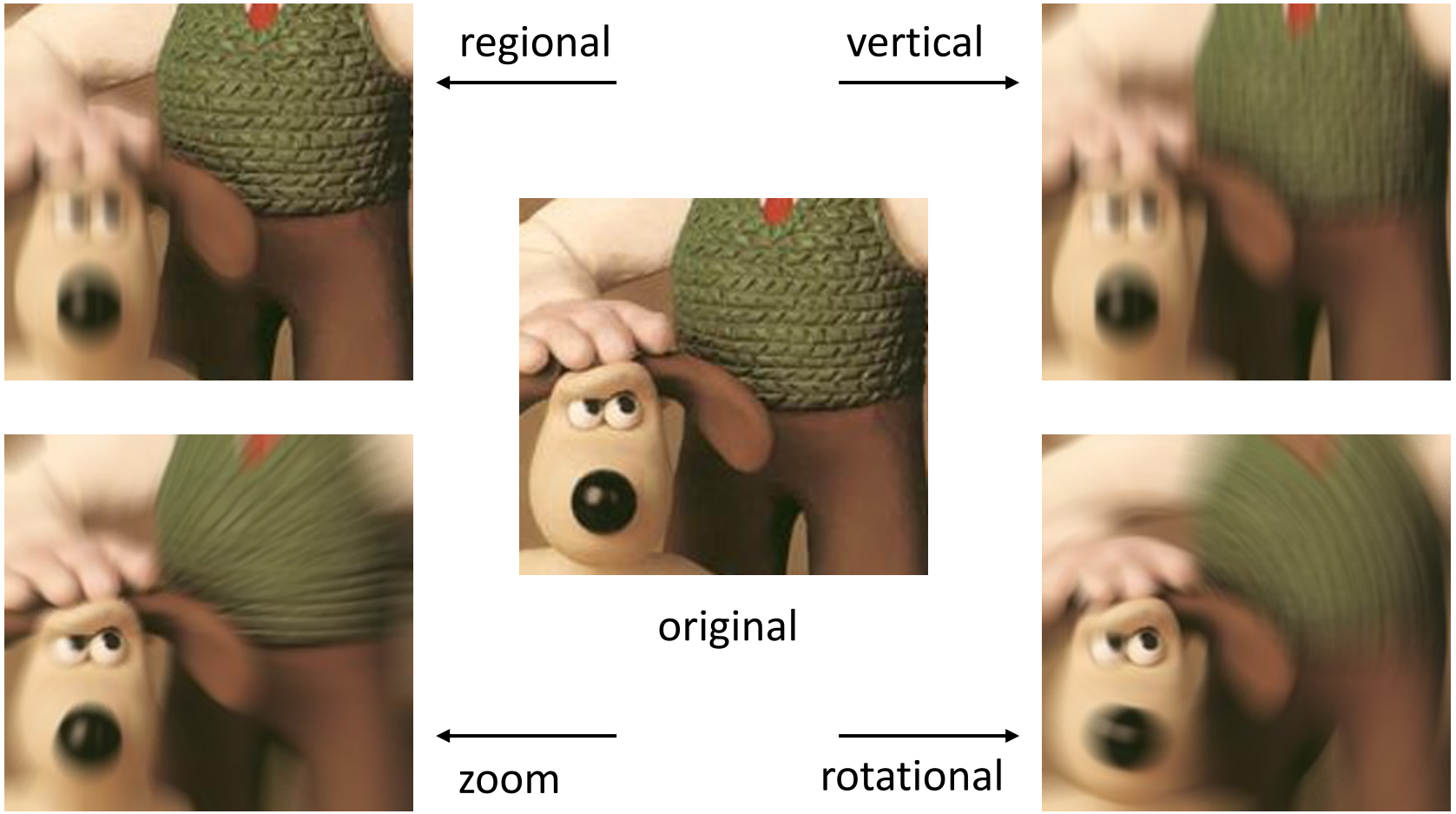
frequency tuned filter

# Examples – (cont.) Restoration

original

enhanced/restored

# Examples – (cont.) Motion Blur



# Image Enhancement Measures

- **Enhancement "parameters":**
  1. **Brightness**
  2. **Contrast.**
  3. **Histogram**

# Image Enhancement Measures – (cont.)

## Example: Brightness/contrast change?

# Image Enhancement Measures – (cont.)

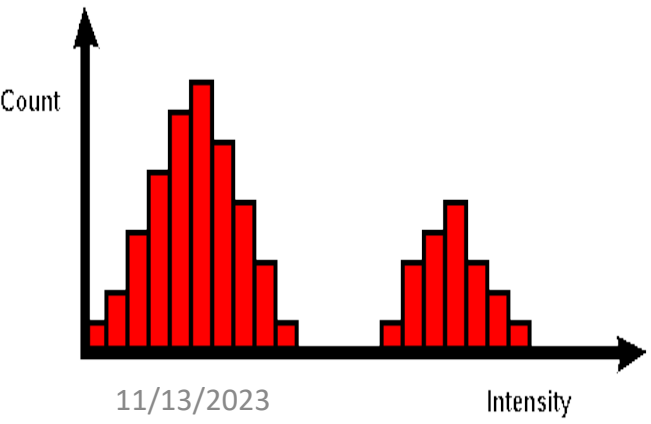
**Example:** Brightness/contrast change?

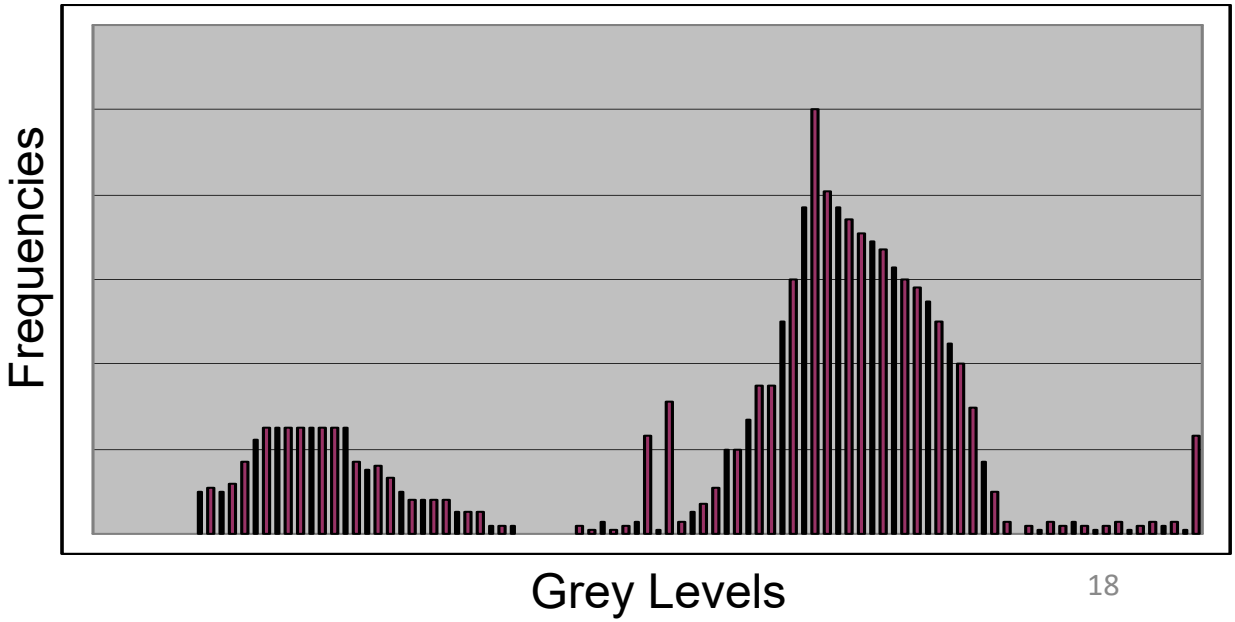
# Image Enhancement Measures – (cont.)

- **Histogram: A way of viewing the distribution of color/intensities in an image.**

$h(x)$ = the number of pixels in $\mathbf{I}$ with intensity value $x$

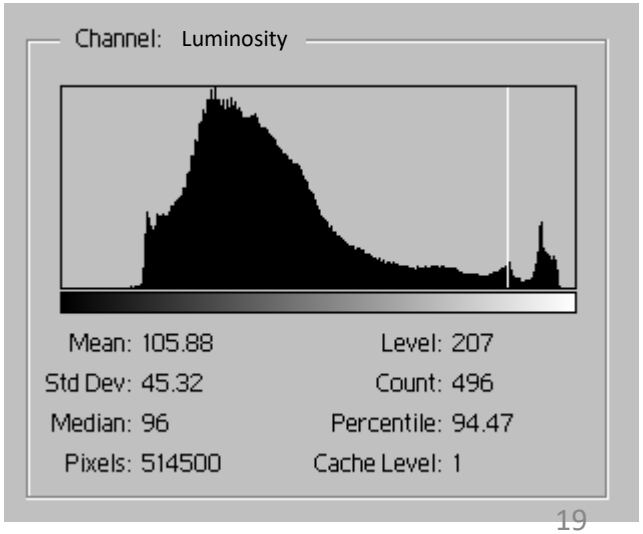
- **A plot of pixels values against pixel counts.**



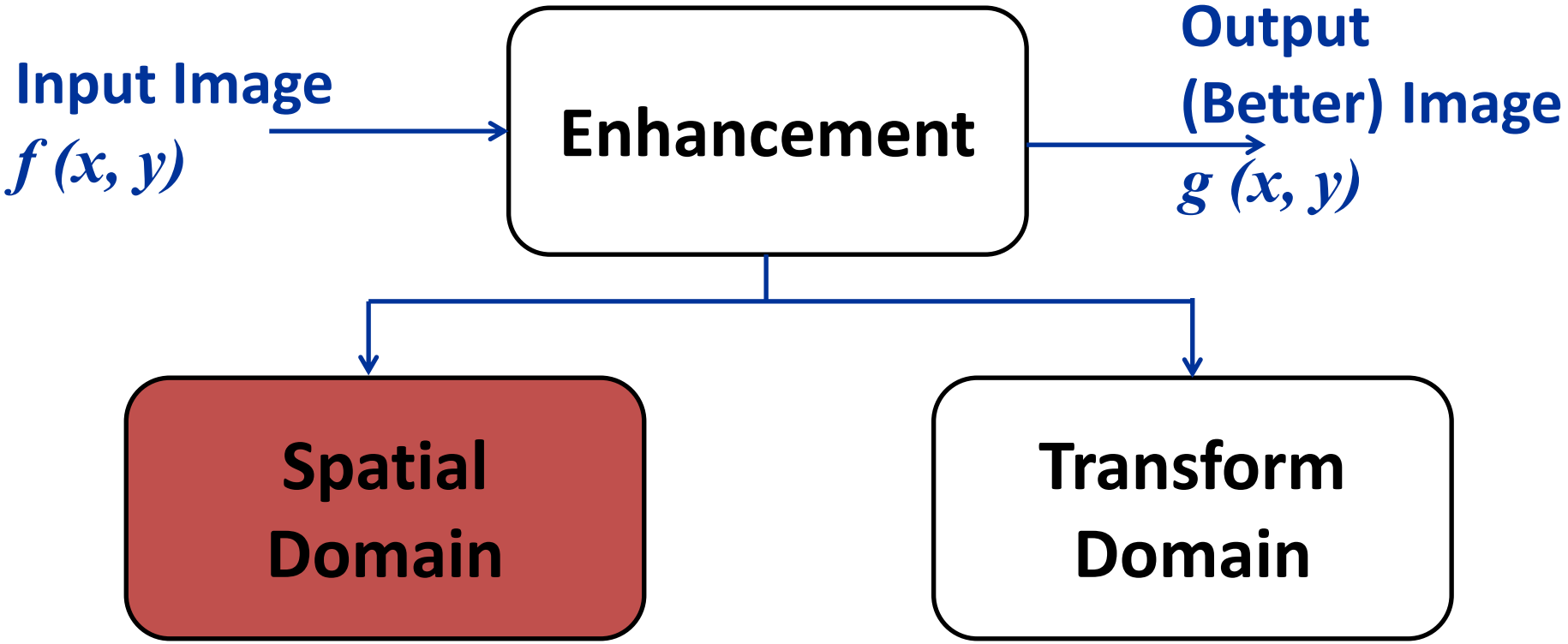



# Image Enhancement Measures – (cont.)

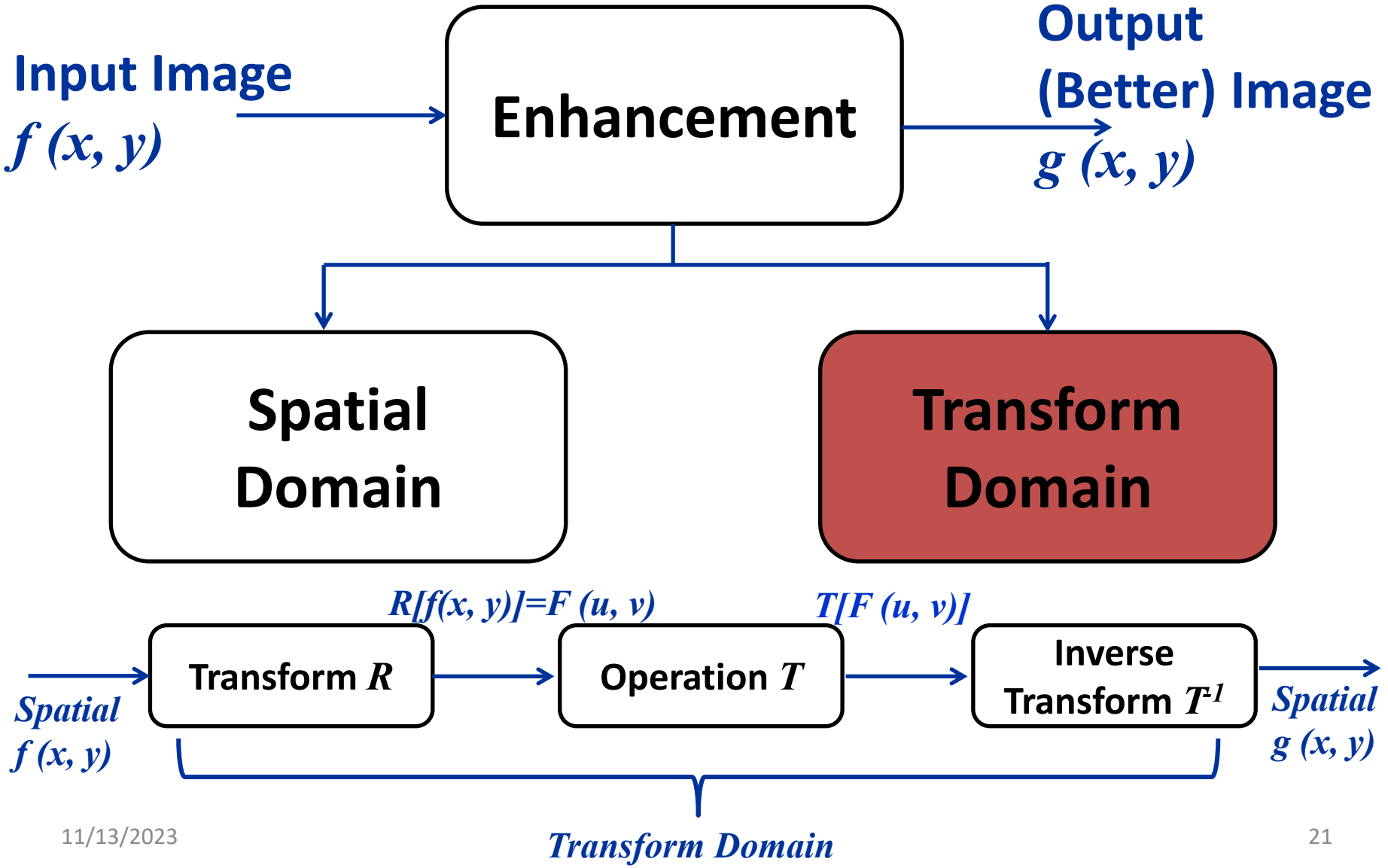
**Gray Scale Image has one histogram while color images has three; one for each band (channel).**



# Domains of Image Enhancement



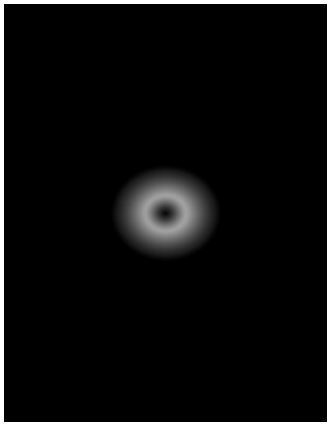

# Domains of Image Enhancement

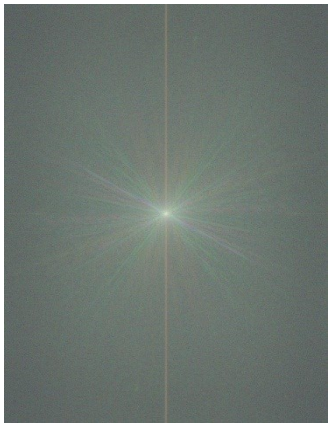


# Domains of Image Enhancement

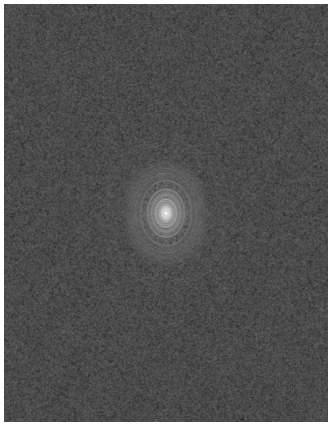
## Example: frequency domain visual effects.

Image & Mask

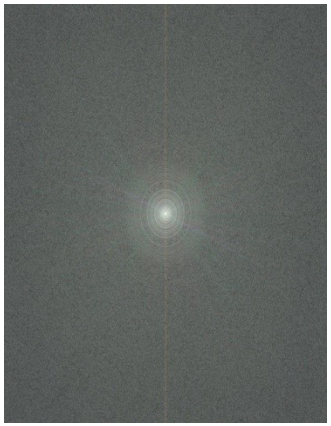




Transforms



Convolution via Fourier Transform



Pixel-wise Product

Inverse Transform

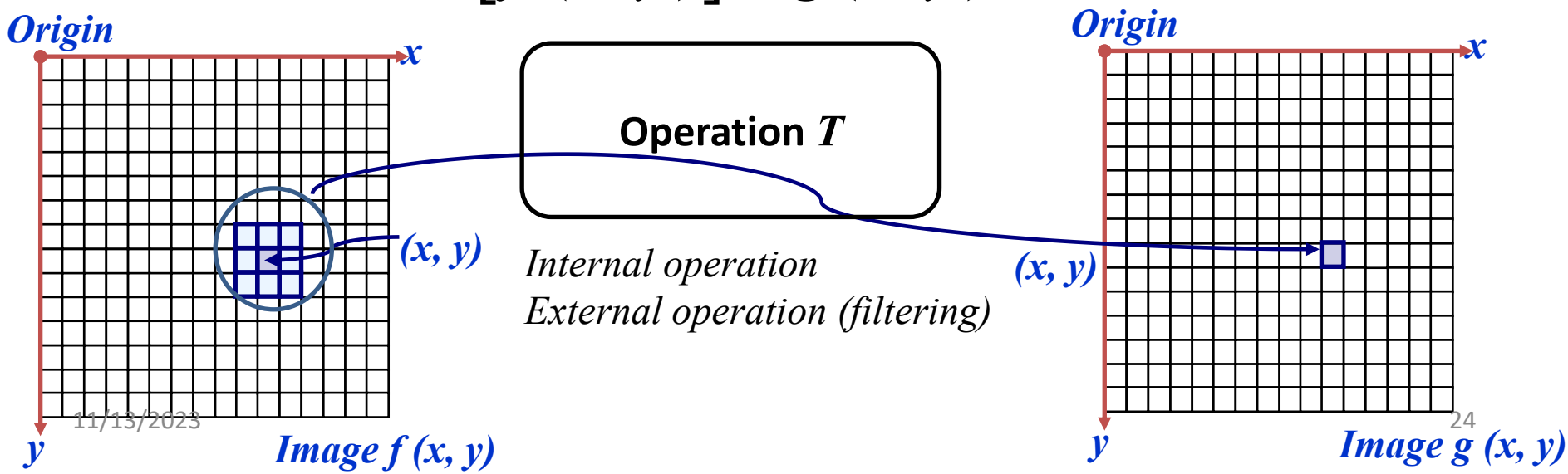
$$g(x,y) = \mathcal{F}^{-1}\{F(u,v)H(u,v)\}$$

# Spatial Enhancement

# Spatial Enhancement Techniques

- **Directly processing pixel value.**
- **For any specific location *(x, y)*, the value of *g* at that location is the result of applying *T* to the pixels in the neighborhood with origin *(x, y)*.**

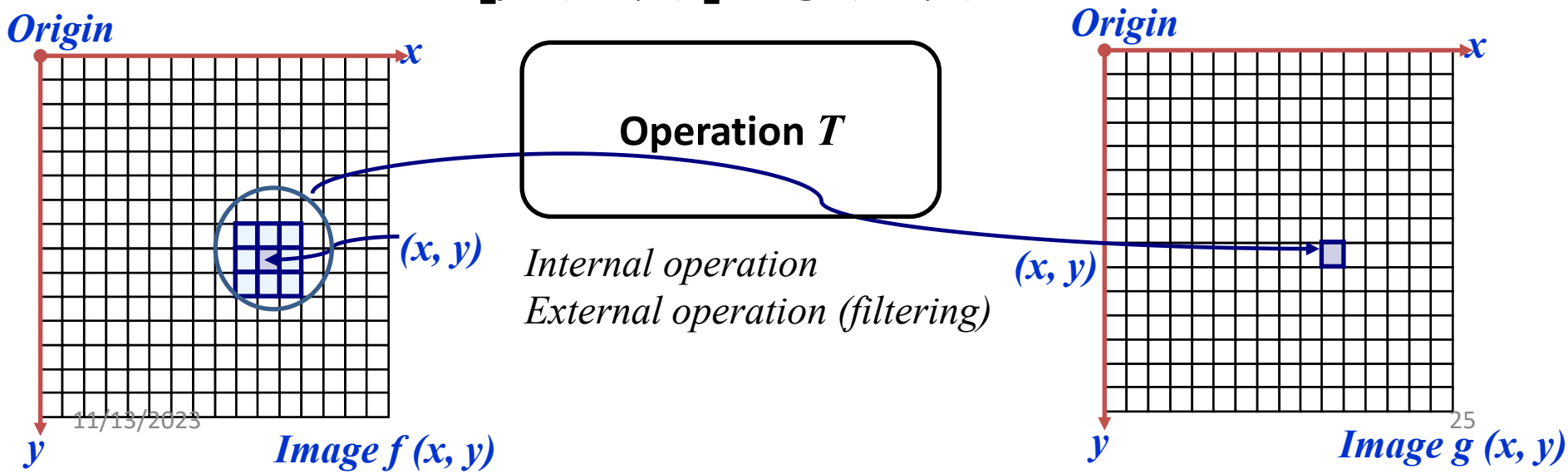
$$T[f(x,y)] = g(x,y)$$



# Spatial Enhancement Techniques

- **Point Processing.**
- **Histogram Processing.**
- **Neighborhood Processing (Filtering).**

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



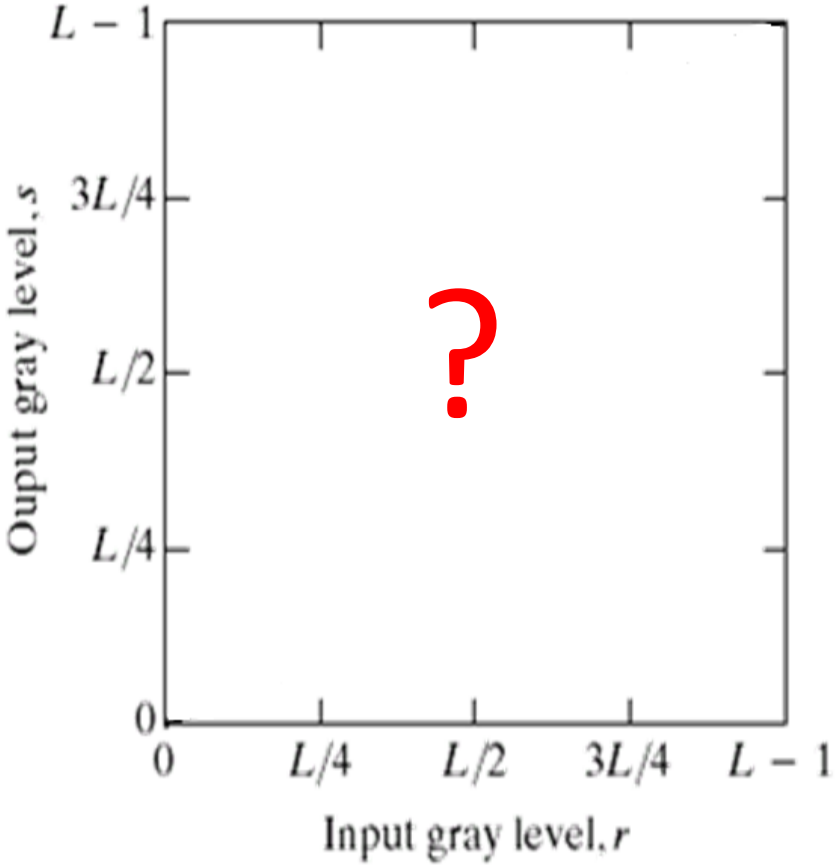
# 1. Point Processing

# Point Processing

- **When the neighborhood is $1 \times 1$, $T$ becomes an intensity transformation function of the form:**

$$g(x,y) = T\left[f(x,y)\right]$$

$$s = T(r)$$



# Point Processing – (cont.)

## Basic Intensity Transformation Functions

**1- Linear**

**Negative/Identity.**

**2- Logarithmic**

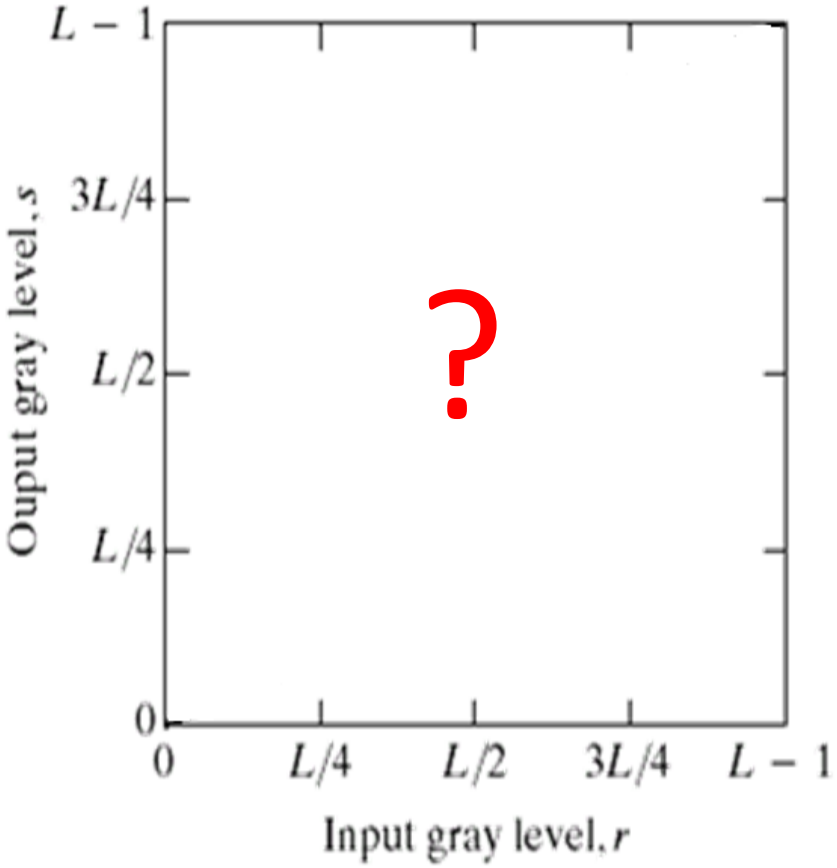
**log/inverse log.**

**3- Power-Law**

**$n^{th}$ power/$n^{th}$ root.**

**4- Piecewise**

**Level slicing/bit-plane slicing.**



# Point Processing – (cont.)

## Basic Intensity Transformation Functions

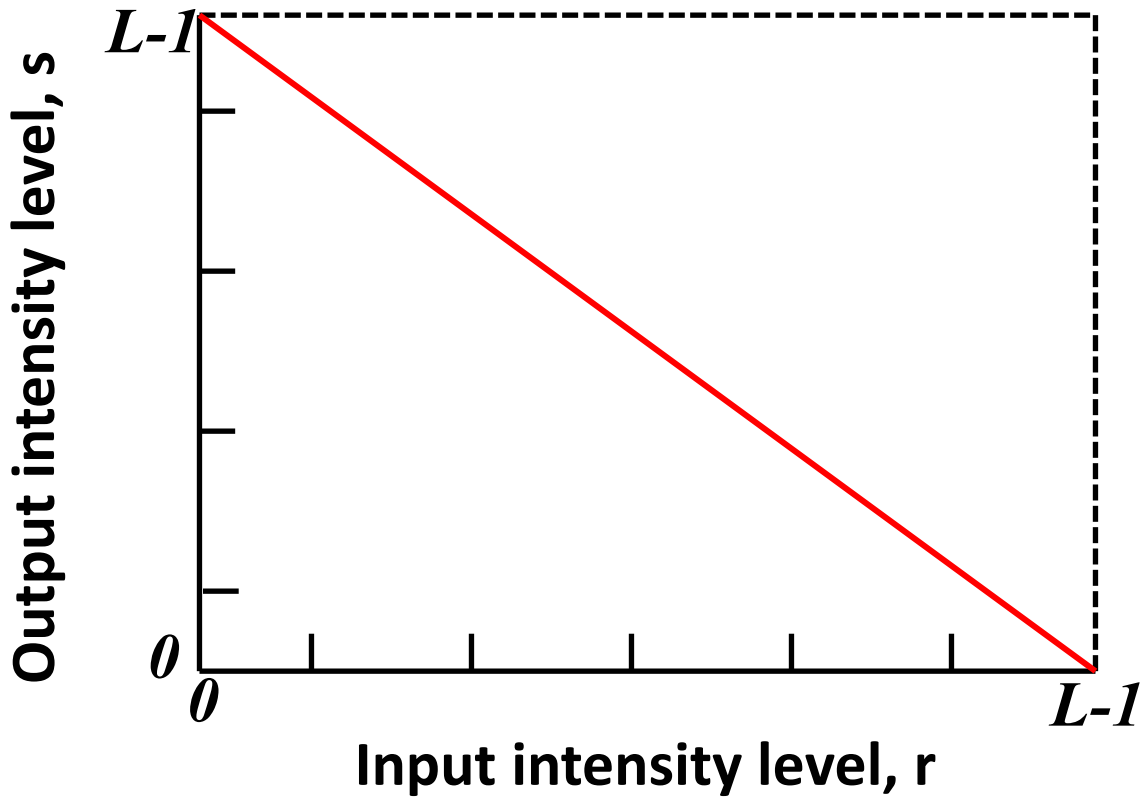
### 1- Image Negatives

# Point Processing – (cont.)

## Basic Intensity Transformation Functions

### 1- Image Negatives

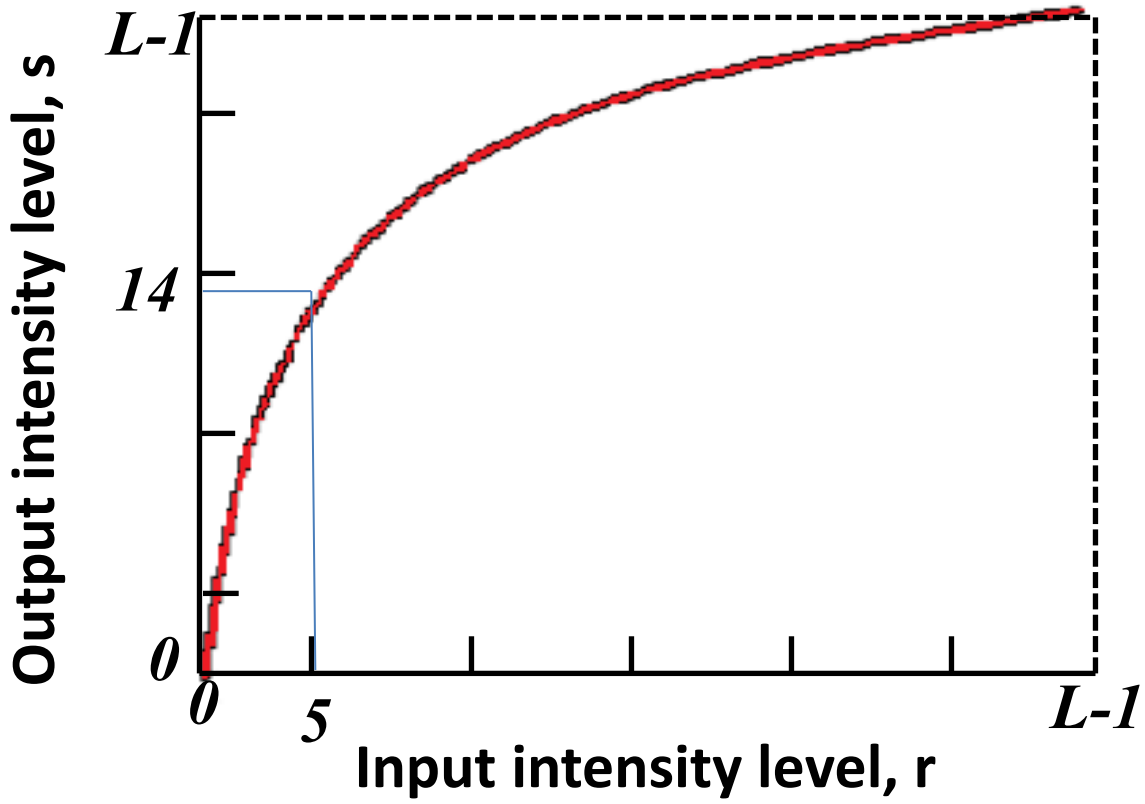
$$s = intensity_{max} - r$$



# Point Processing – (cont.)

## Basic Intensity Transformation Functions

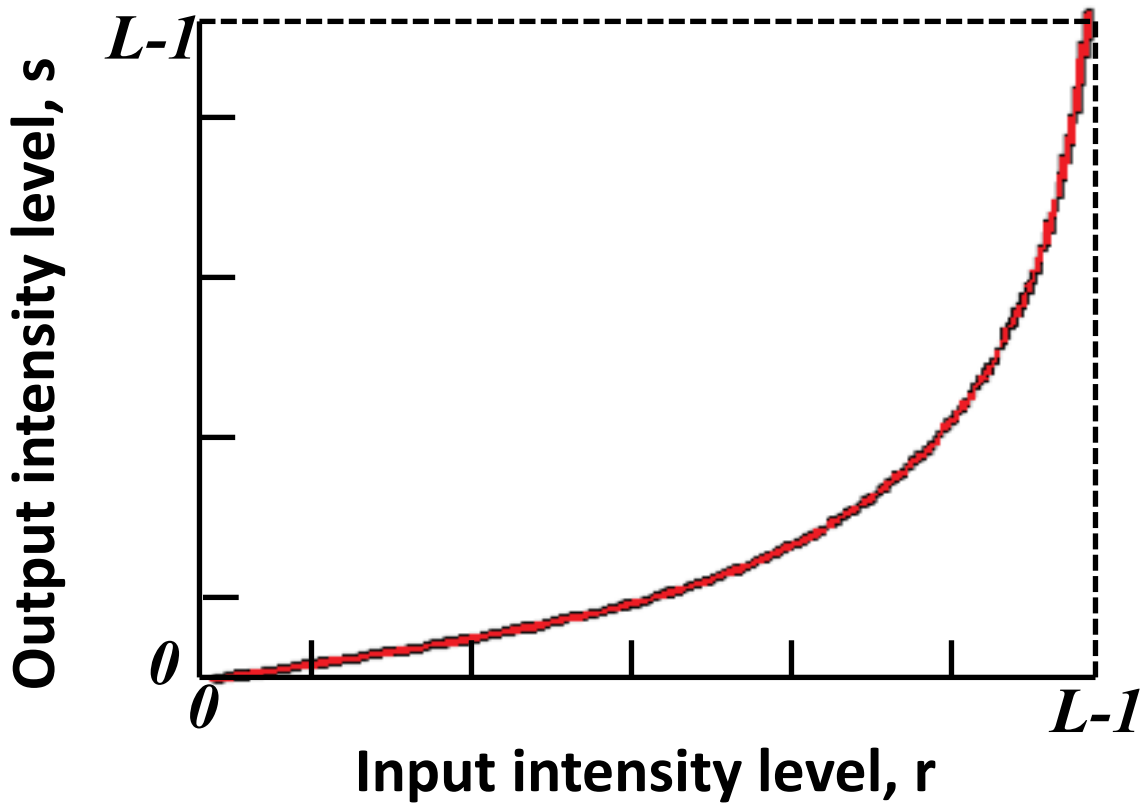
### 2- Log Transform $s = c\ log(1+r)$



# Point Processing – (cont.)

## Basic Intensity Transformation Functions

### 2- Log Transform (inv.)



# Point Processing – (cont.)

## Basic Intensity Transformation Functions

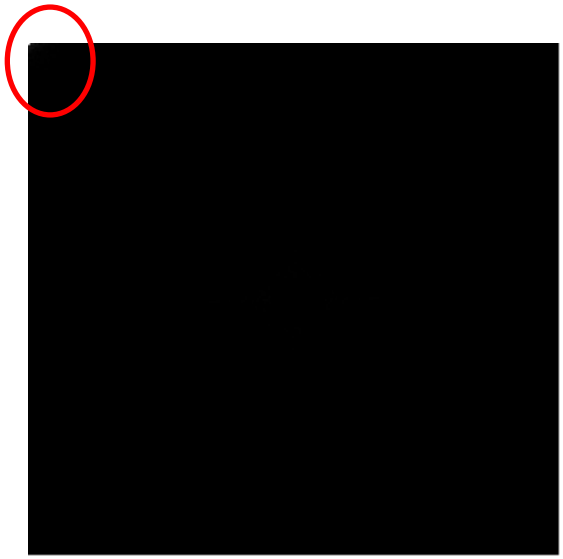
**2- Log Transform**

- **When the input grey level values may have an extremely large range of values.**
- **e.g *Fourier transform* can have values in the range [0-$10^6$].**

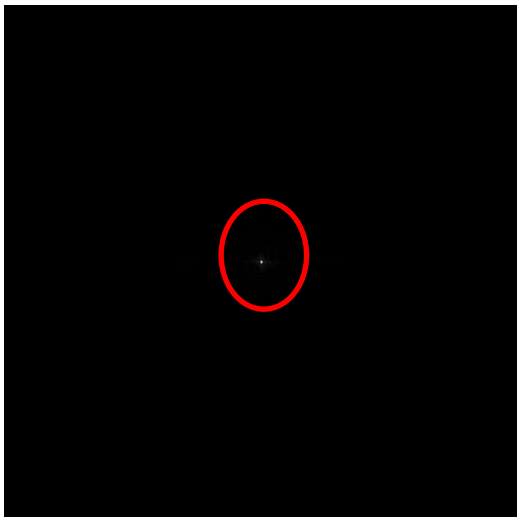
# Point Processing – (cont.)

## Example: the use of log transform

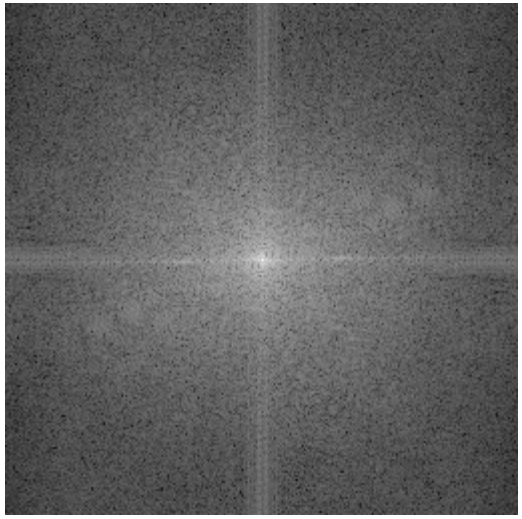
Test image



Spectrum [0-$10^6$]
(Fourier^2)



Centered spectrum



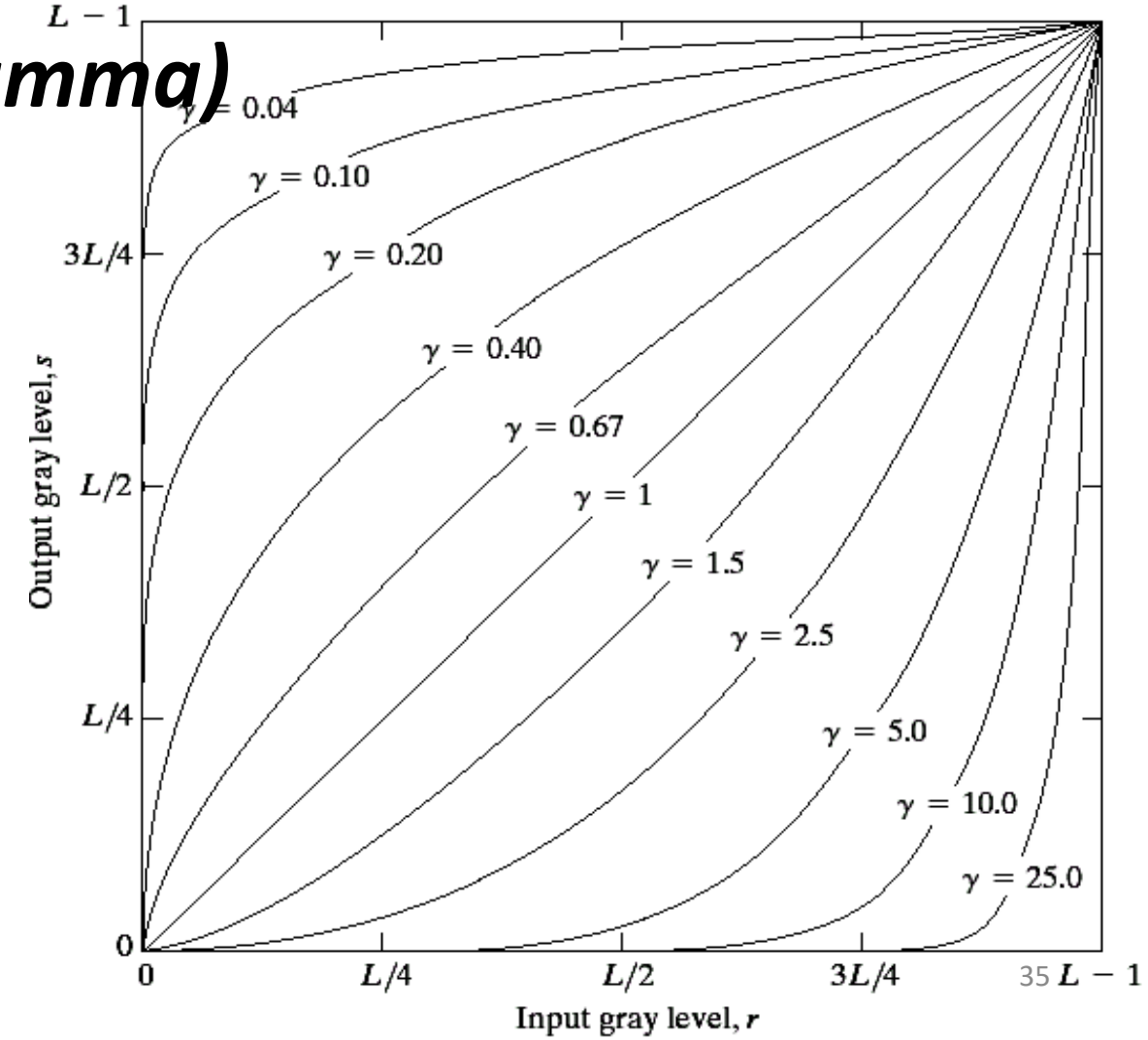
[0-6.17]
Scaled centered spectrum

# Point Processing – (cont.)

## Basic Intensity Transformation Functions

### 3- Power-Law *(Gamma)*

$$s = cr^{\gamma}$$



# Point Processing – (cont.)

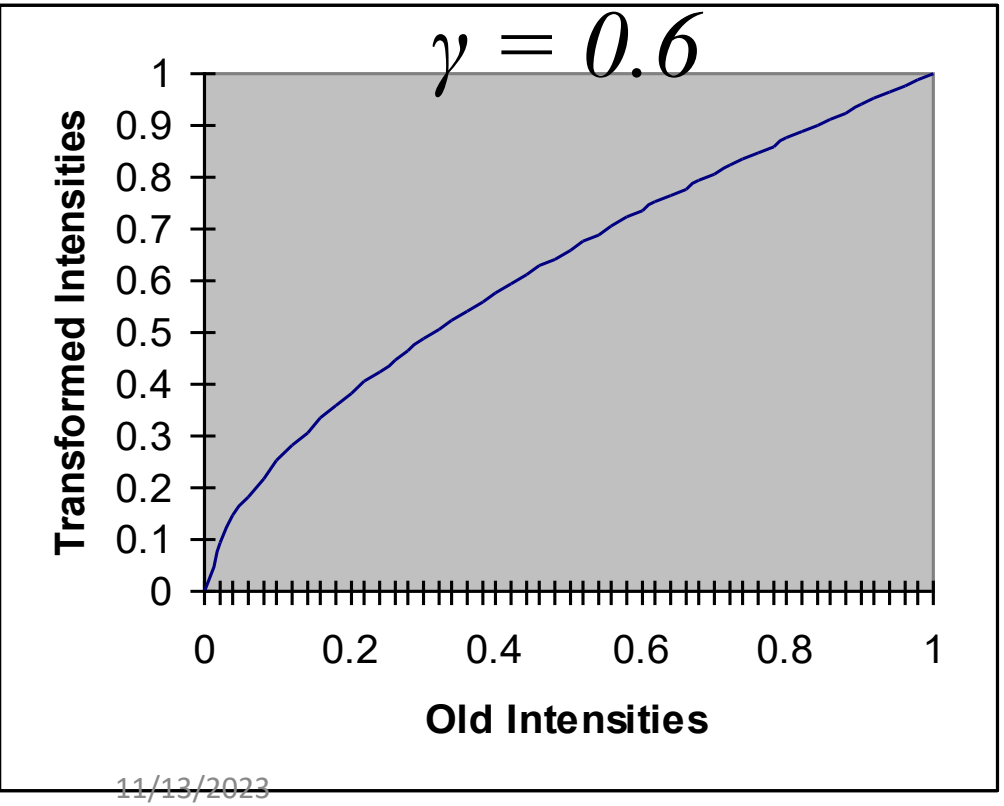
## Basic Intensity Transformation Functions

### 3- Power-Law *(Gamma)*

# Point Processing – (cont.)

## Basic Intensity Transformation Functions

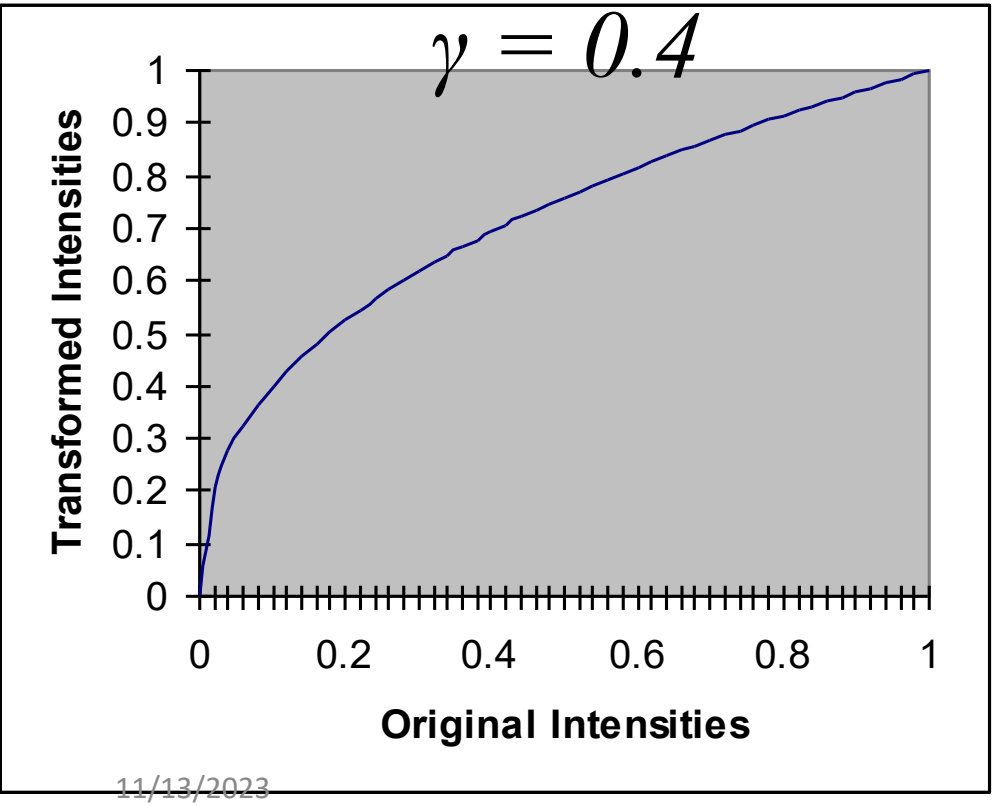
### 3- Power-Law *(Gamma)*




# Point Processing – (cont.)

## Basic Intensity Transformation Functions

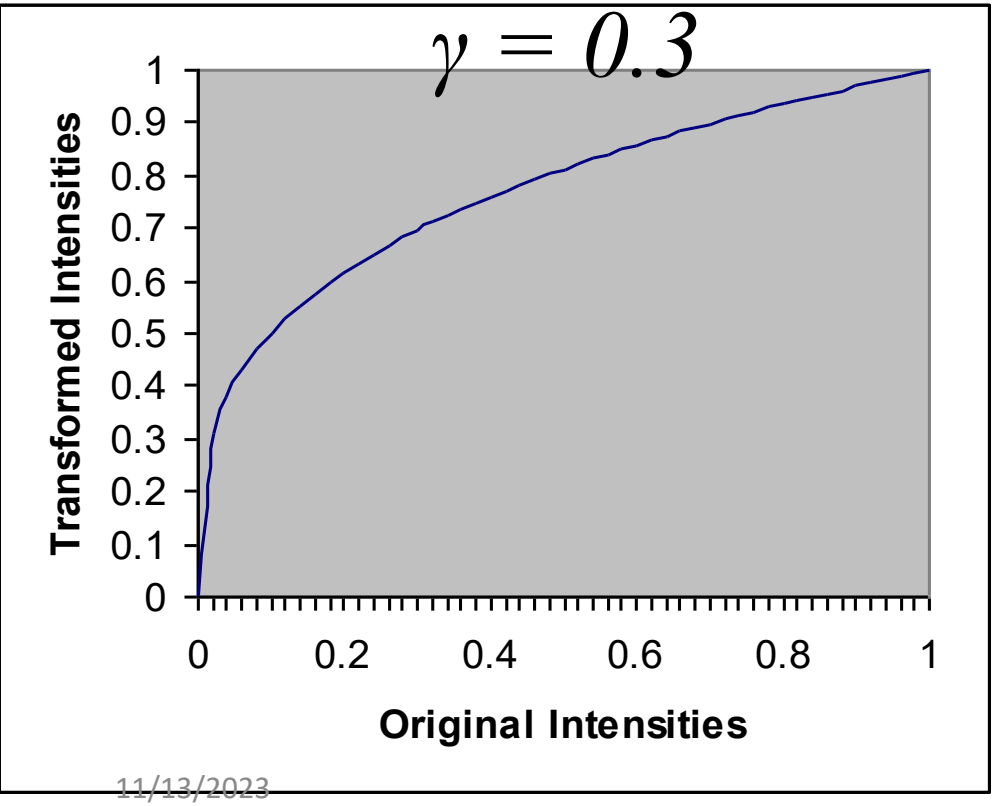
### 3- Power-Law *(Gamma)*




# Point Processing – (cont.)

## Basic Intensity Transformation Functions

### 3- Power-Law *(Gamma)*




# Point Processing – (cont.)

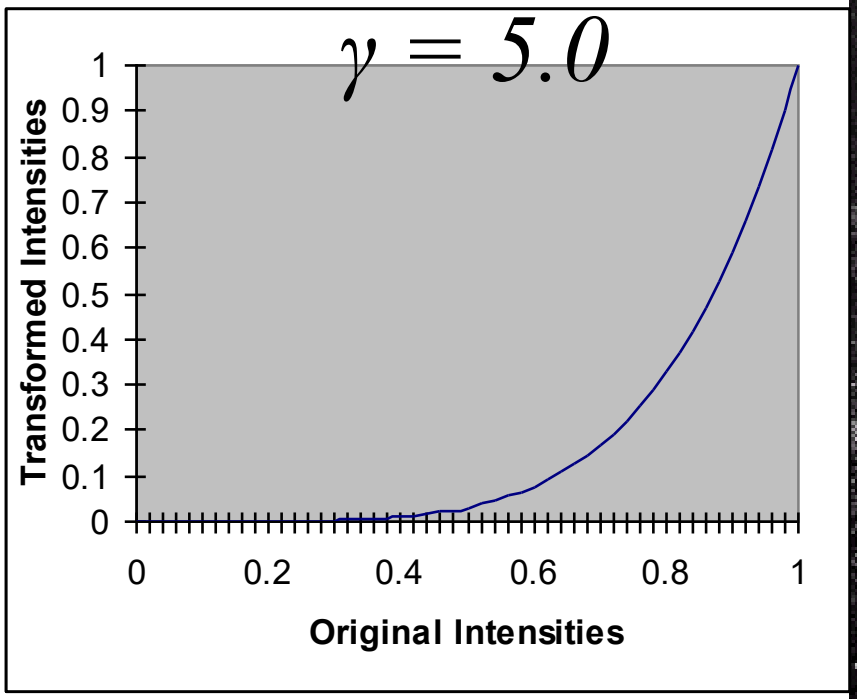
## Basic Intensity Transformation Functions

### 3- Power-Law *(Gamma)*

# Point Processing – (cont.)

## Basic Intensity Transformation Functions

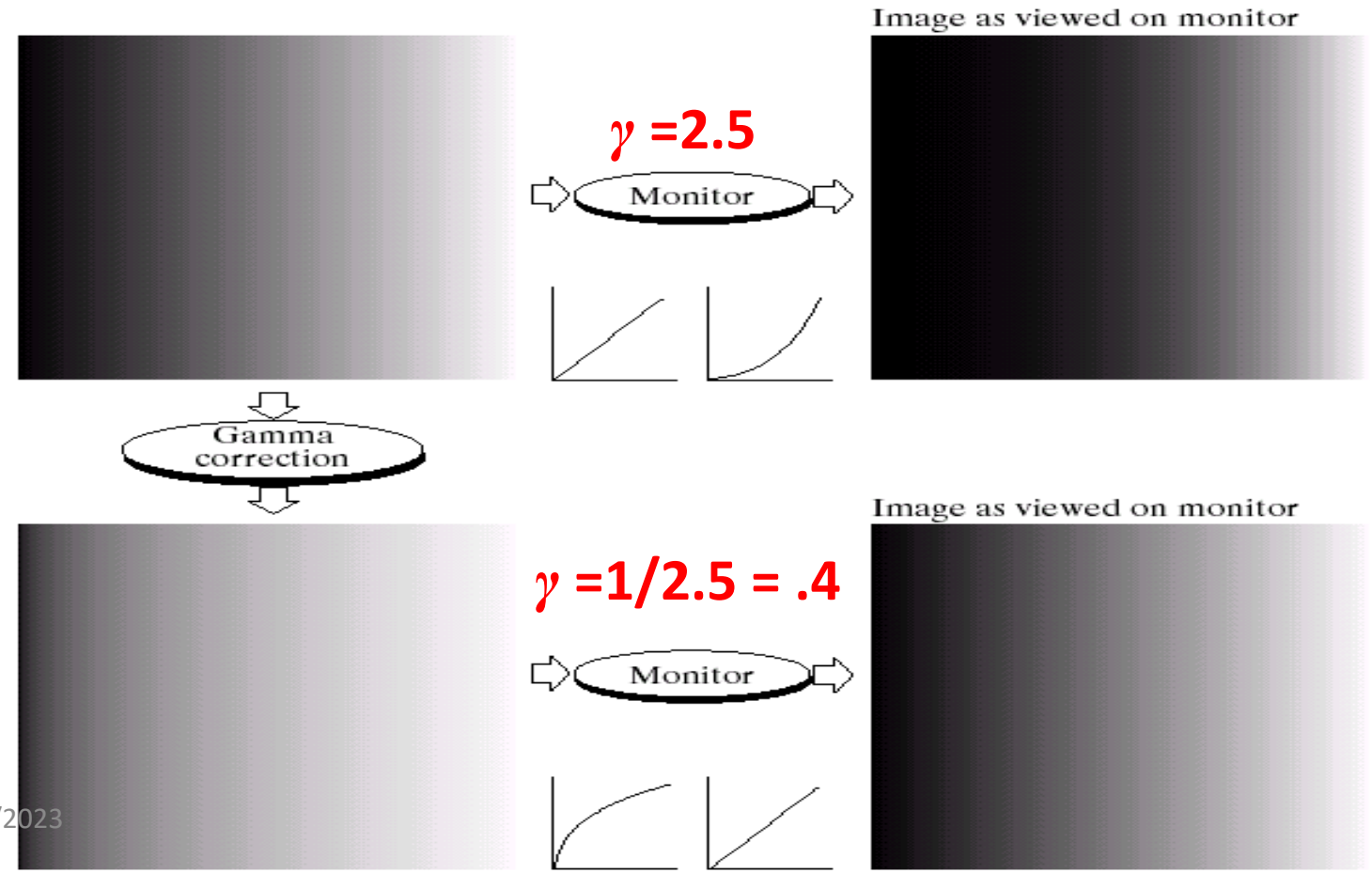
### 3- Power-Law *(Gamma)*




# Point Processing – (cont.)

## Basic Intensity Transformation Functions

### 3- Power-Law - *Gamma Correction*



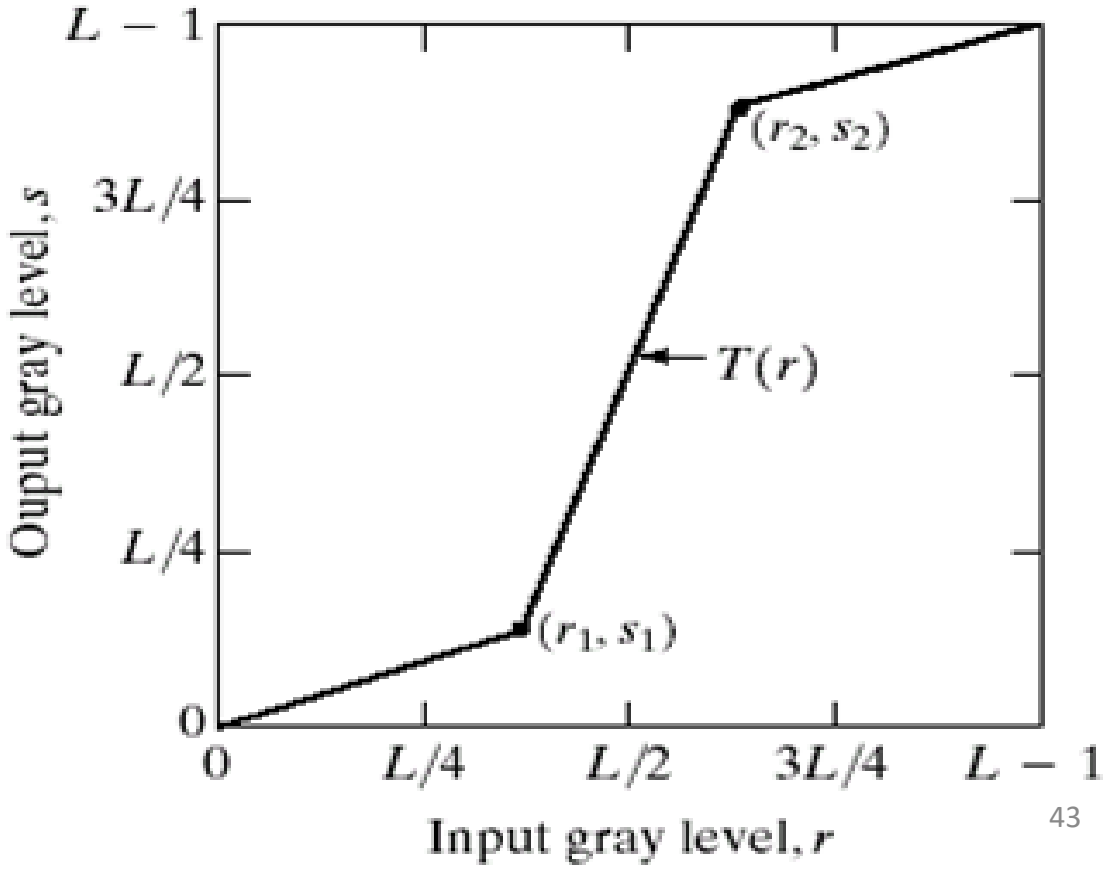
# Point Processing – (cont.)

## Basic Intensity Transformation Functions

### 4- Piecewise Linear Transformation Functions

**General function.**

**By changing the location of the two points (joints) we get several other Transformation function.**

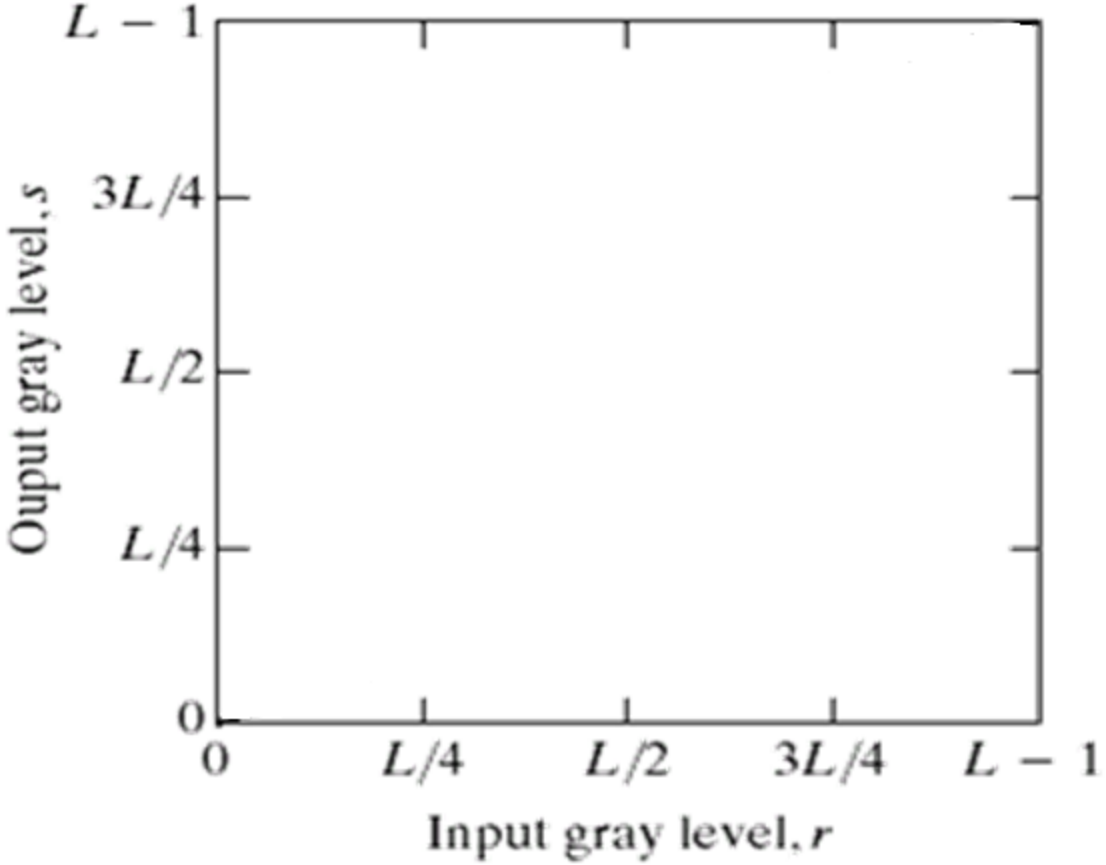


# Point Processing – (cont.)

## Basic Intensity Transformation Functions

### 4- Piecewise Linear Transformation Functions

$$(r_1, s_1) = (r_2, 0)$$
$$(r_2, s_2) = (r_2, L-1)$$

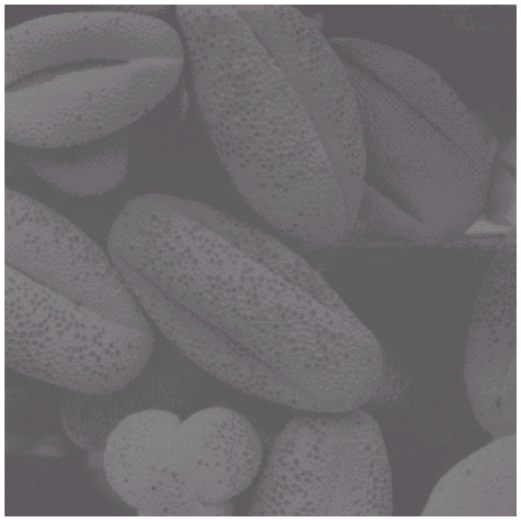


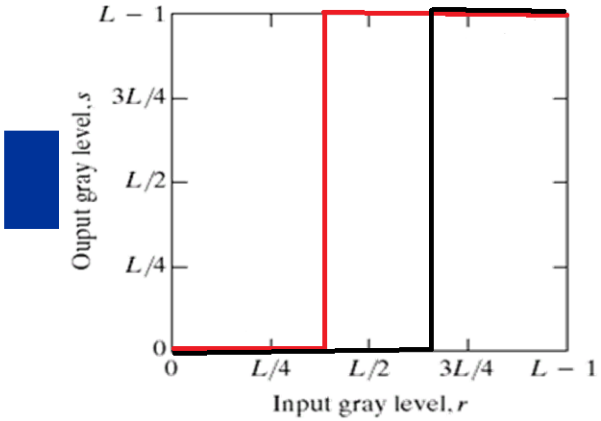
# Point Processing – (cont.)

## Basic Intensity Transformation Functions

### 4- Piecewise Linear Transformation Functions

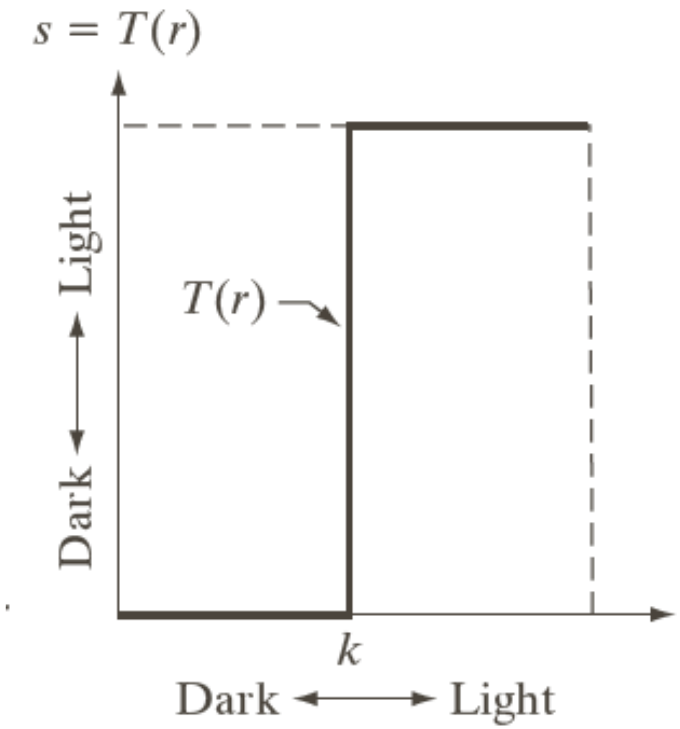
***Thresholding***

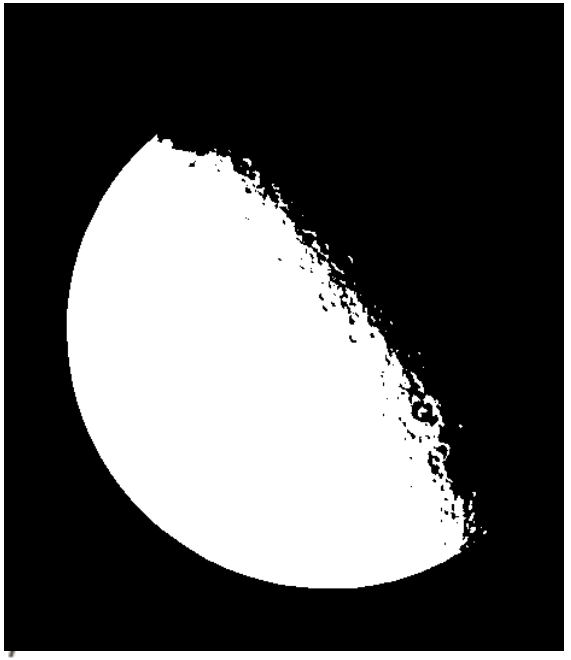





# Point Processing – (cont.)

- **Thresholding**



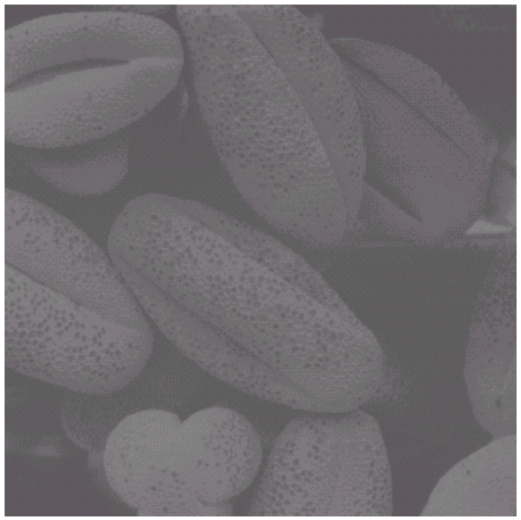



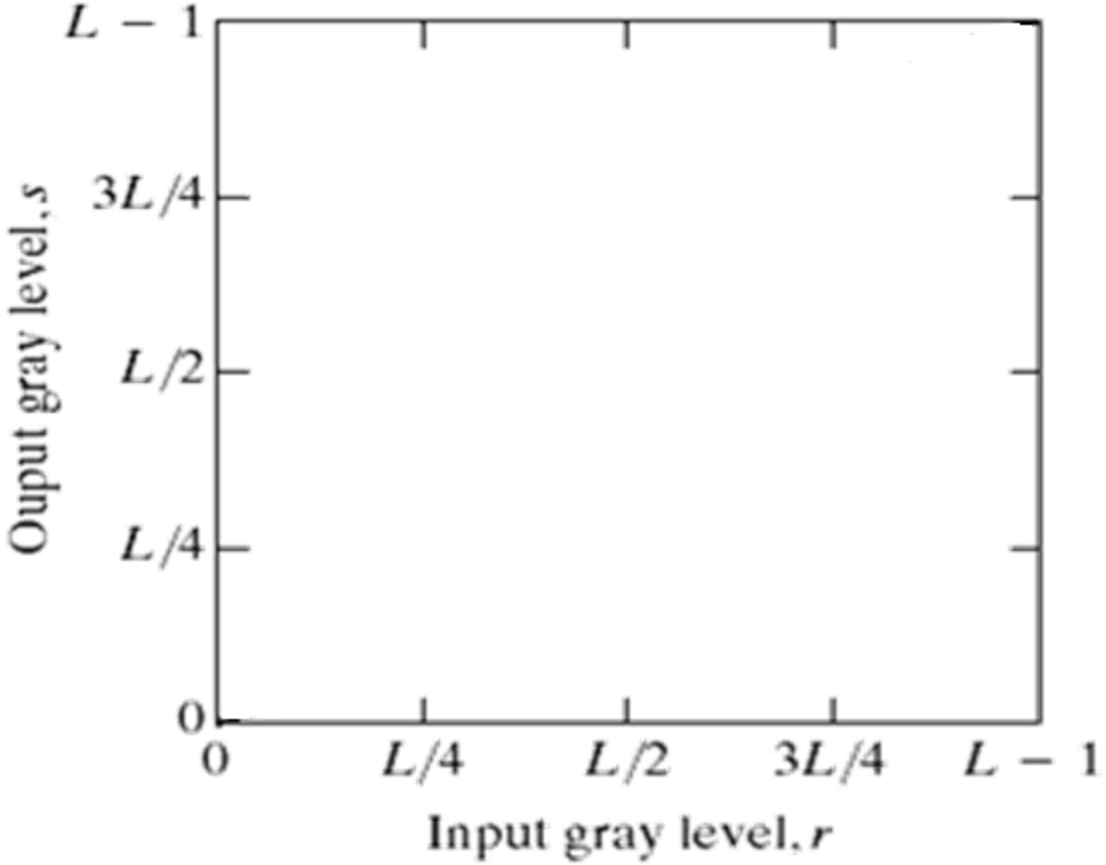
# Point Processing – (cont.)

## Basic Intensity Transformation Functions

### 4- Piecewise Linear Transformation Functions

$$(r_1, s_1) = (r_{\min}, 0)$$
$$(r_2, s_2) = (r_{\max}, L-1)$$



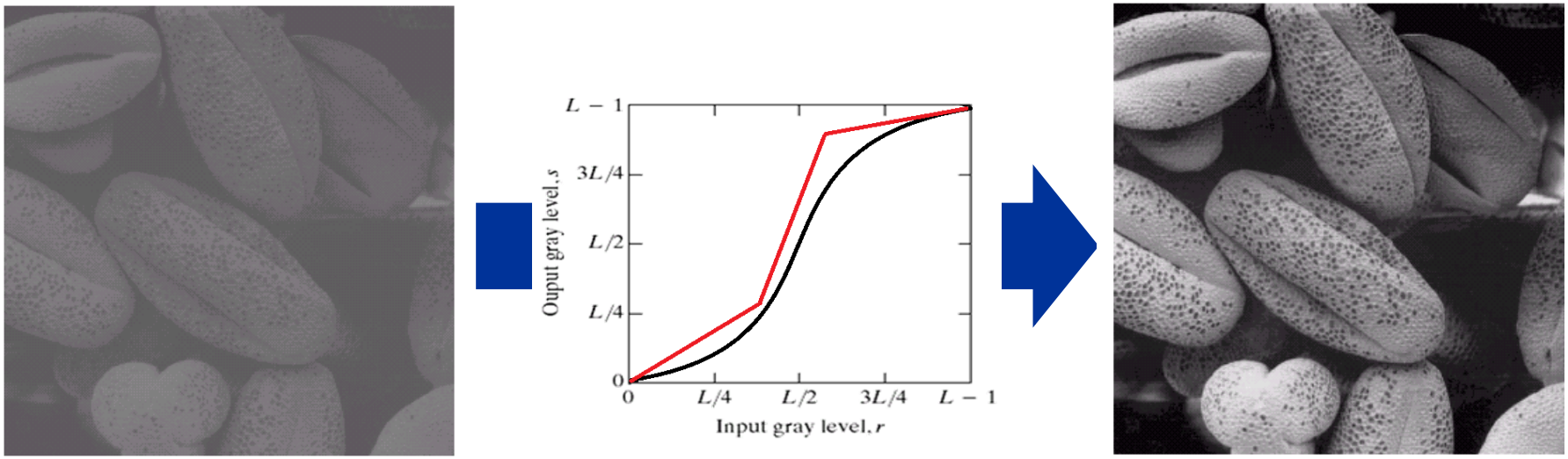


# Point Processing – (cont.)

## Basic Intensity Transformation Functions

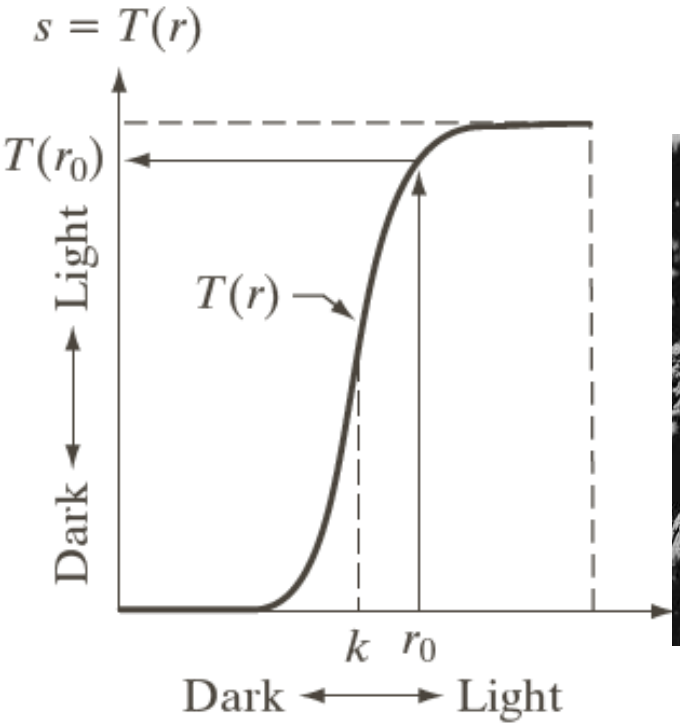
### 4- Piecewise Linear Transformation Functions

***Contrast Stretching***



# Point Processing – (cont.)

## Example: Contrast Stretching




# Point Processing – (cont.)

## Basic Intensity Transformation Functions

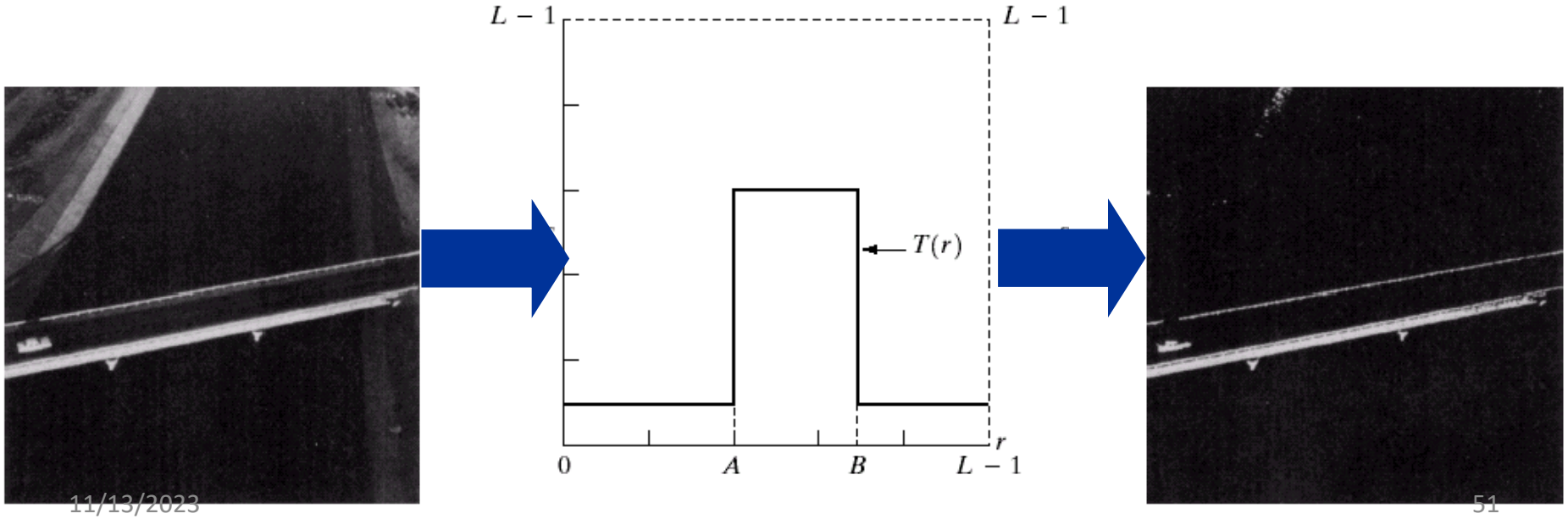
### 5. Intensity Level Slicing

- Similar to thresholding.
- Other levels can be suppressed or maintained.
- Useful for highlighting features in an image.

# Point Processing – (cont.)

## Basic Intensity Transformation Functions

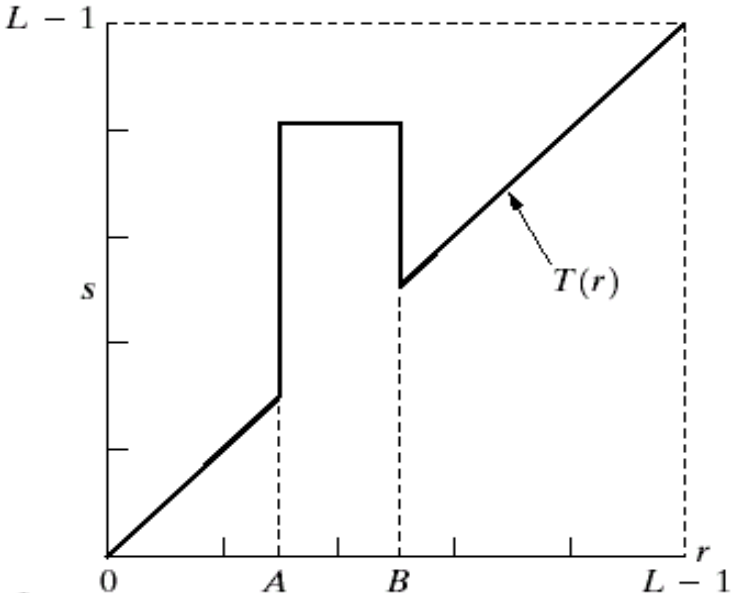
### 5. Intensity Level Slicing



# Point Processing – (cont.)

## Basic Intensity Transformation Functions

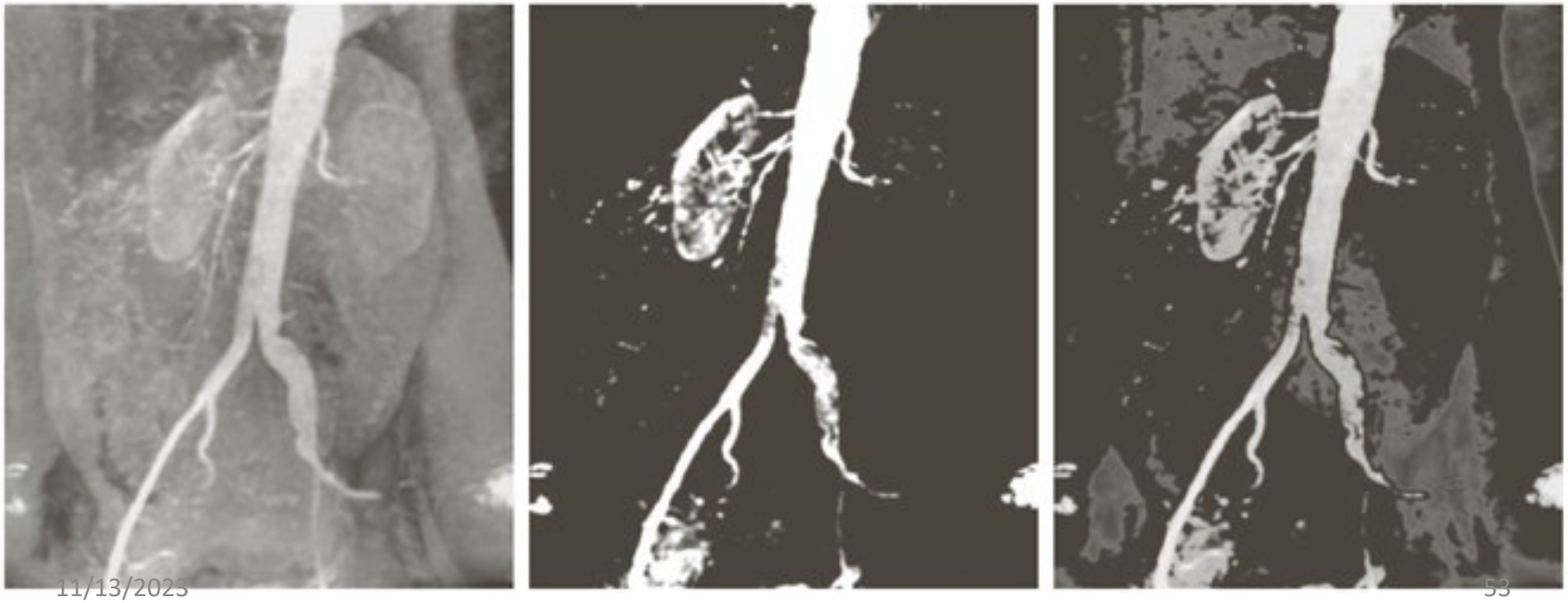
### 5. Intensity Level Slicing



# Point Processing – (cont.)

## Basic Intensity Transformation Functions

### 5. Intensity Level Slicing

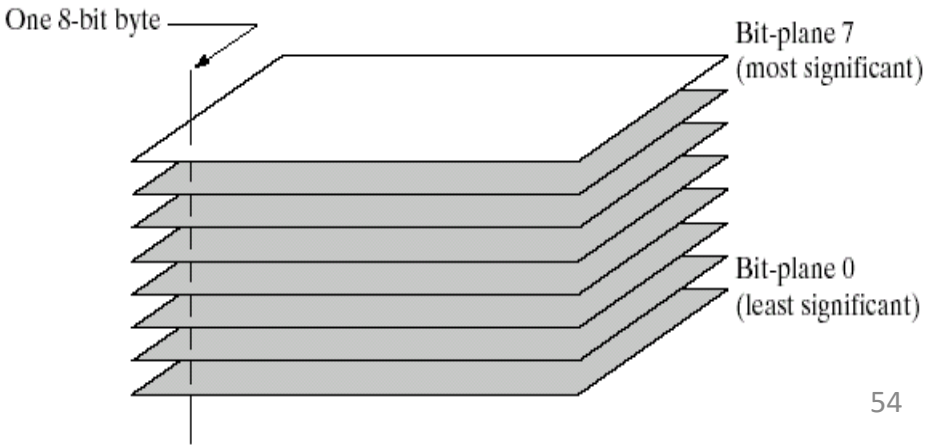


# Point Processing – (cont.)

## Basic Intensity Transformation Functions

### 6. Bit-plane Slicing (re-quantization)

- Highlighting the contribution of each bit to the total image appearance.
- Higher-order bit-planes vs. lower-order bit-planes.



# Point Processing – (cont.)

## Basic Intensity Transformation Functions

### 6. Bit-plane Slicing (re-quantization)

**Plane 1 contains the lowest order bit of all the pixels in the image.**

| | | |
|---|---|---|
| 10100111 | 10000101 | 01101111 |
| 10010000 | 10001100 | 10000111 |
| 10011111 | 10011010 | 10010100 |

**And plane 8 contains the highest order bit of all the pixels in the image.**

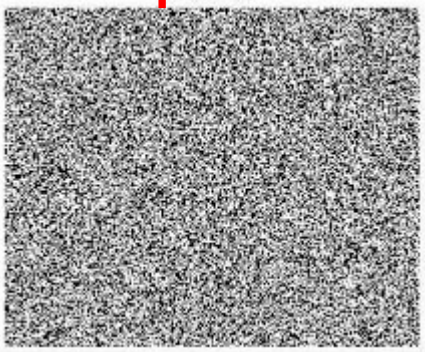
| | | |
|---|---|---|
| 10100111 | 10000101 | 01101111 |
| 10010000 | 10001100 | 10000111 |
| 10011111 | 10011010 | 10010100 |

# Point Processing – (cont.)

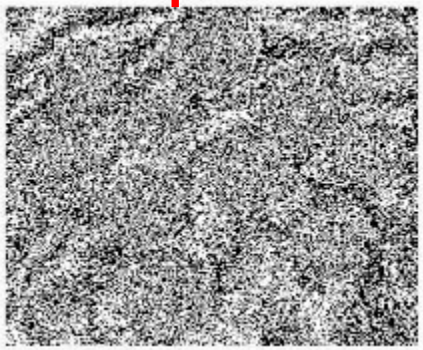
## Basic Intensity Transformation Functions

### 6. Bit-plane Slicing (re-quantization)

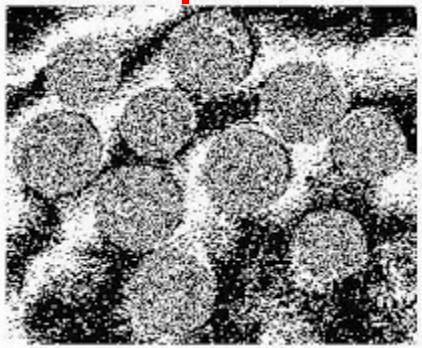
Bit plane 1



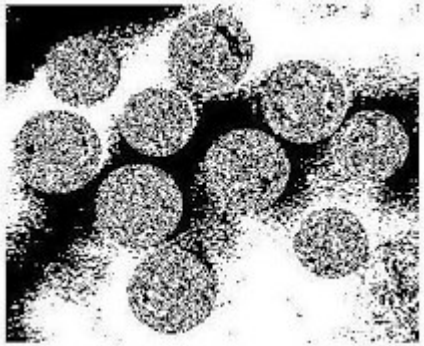
Bit plane 2



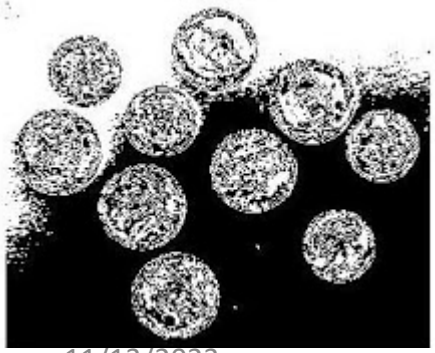
Bit plane 3



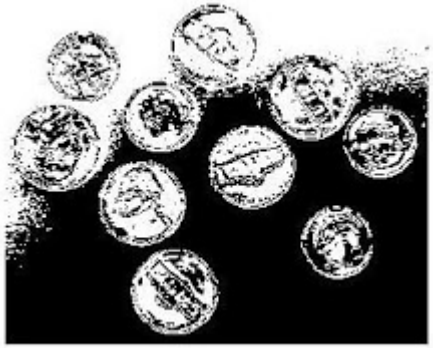
Bit plane 4



Bit plane 5

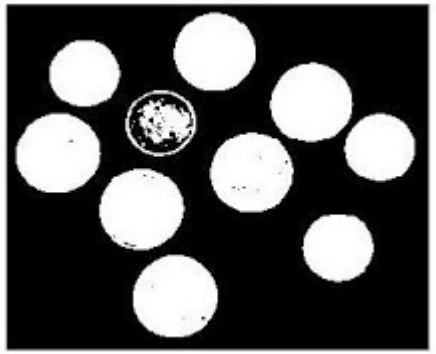


Bit plane 6



Bit plane 7

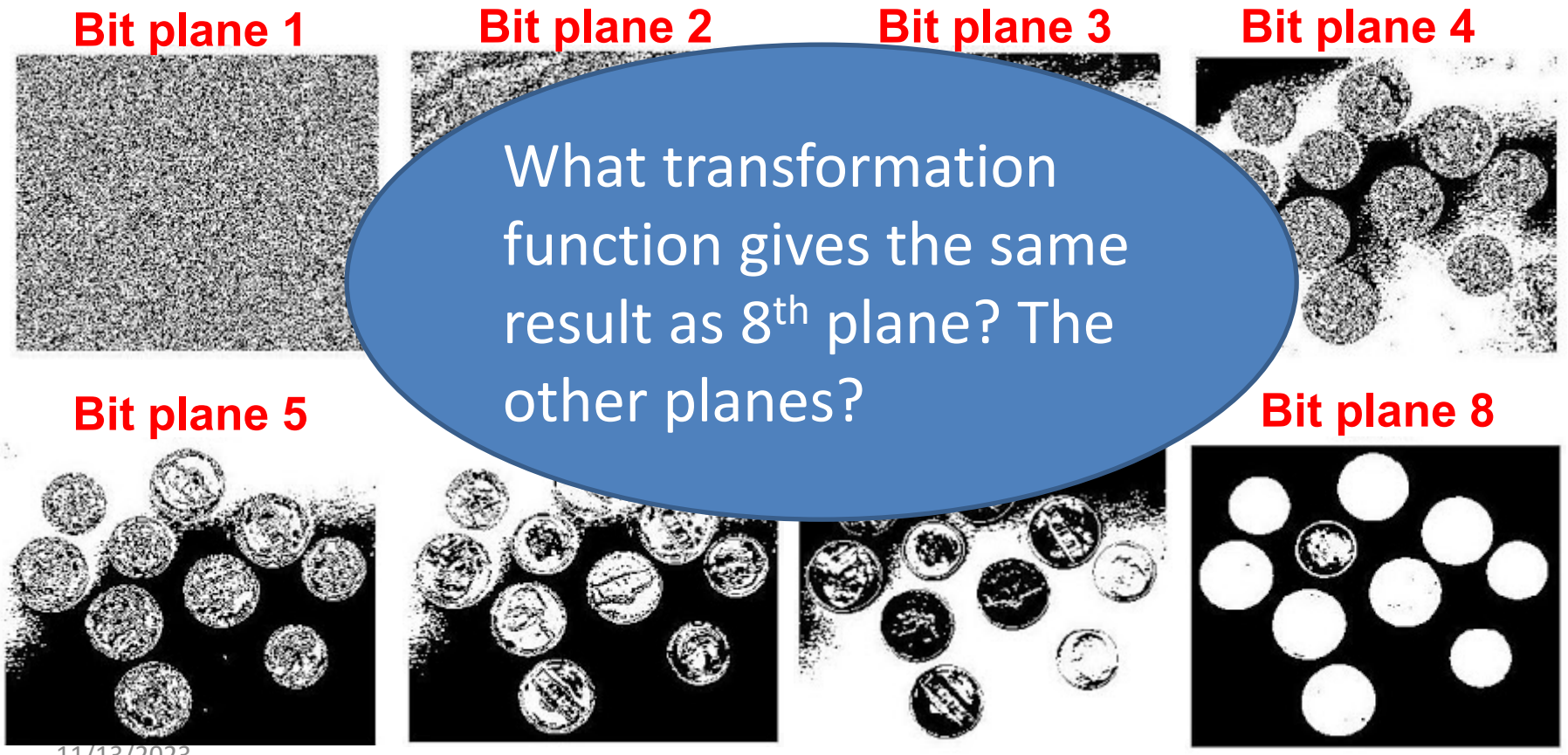
Bit plane 8



# Point Processing – (cont.)

## Basic Intensity Transformation Functions

### 6. Bit-plane Slicing (re-quantization)



# Point Processing – (cont.)

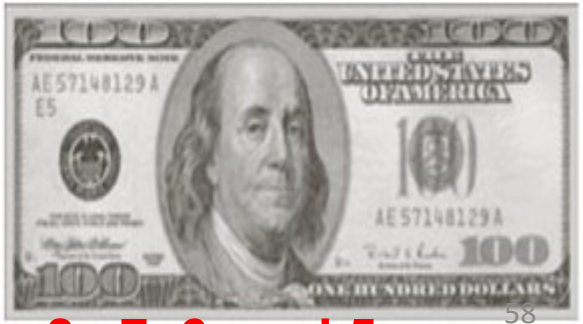
## Basic Intensity Transformation Functions

### 6. Bit-plane Slicing (re-quantization)

***Compression (construct image from fewer bit-planes)***

8 and 7

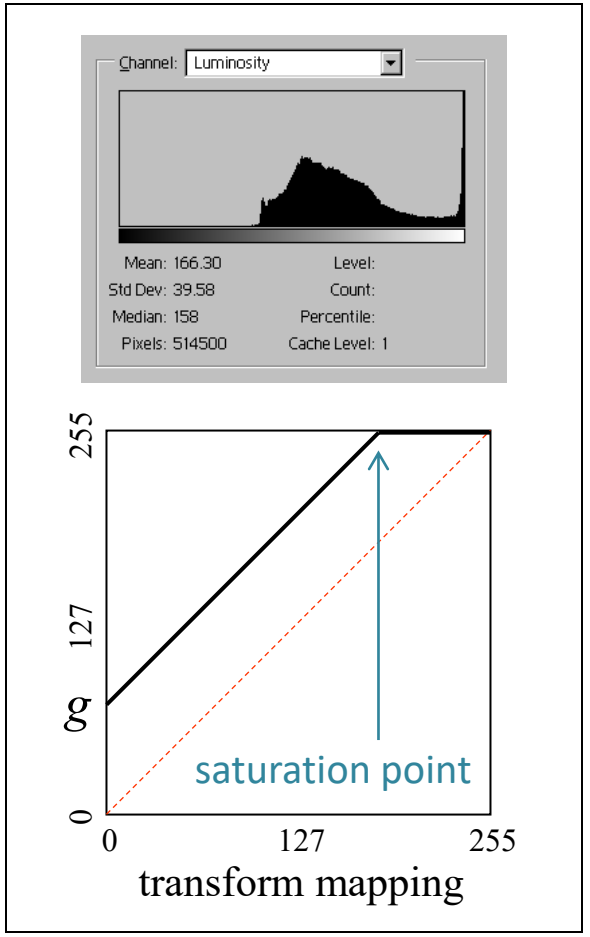
8 , 7, and 6



8 , 7, 6, and 5

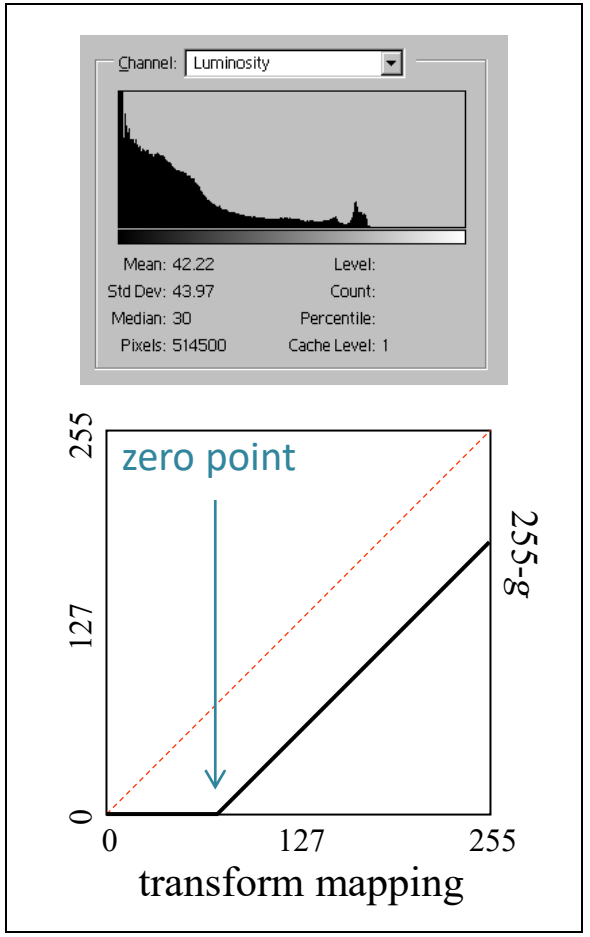
# Summary

## Increase Brightness



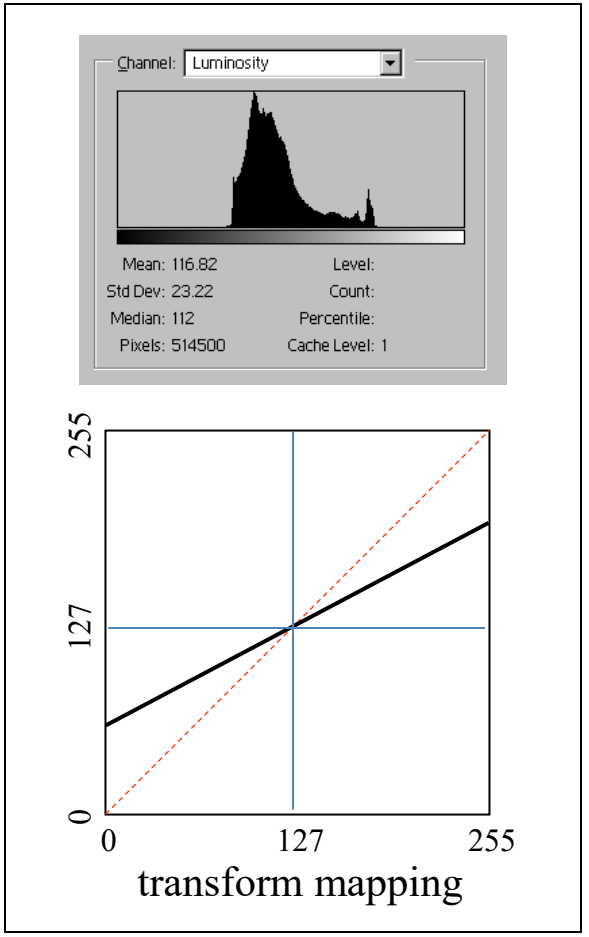
# Summary – (cont.)

## Decrease Brightness



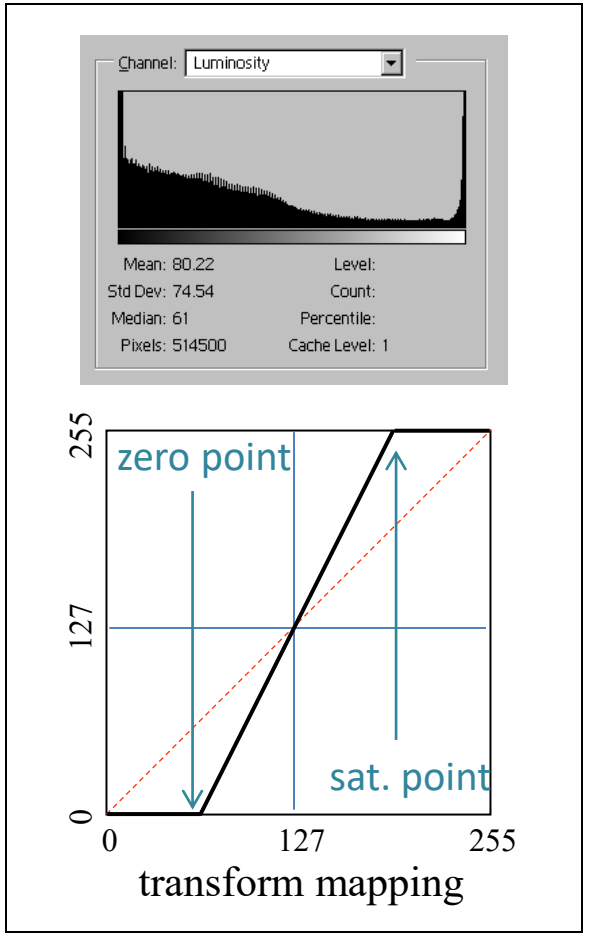
# Summary – (cont.)

## Decrease Contrast



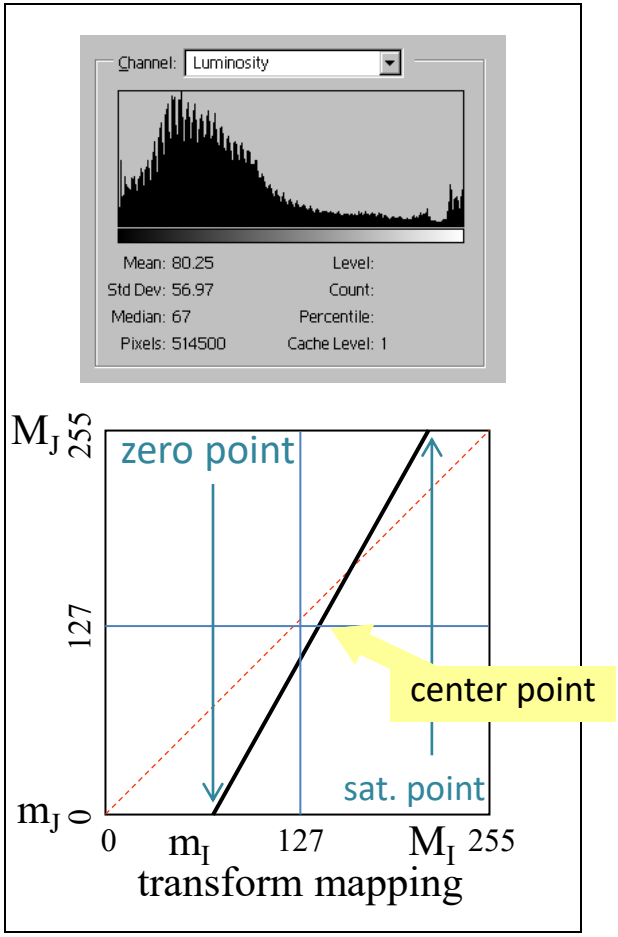
# Summary – (cont.)

## Increase Contrast



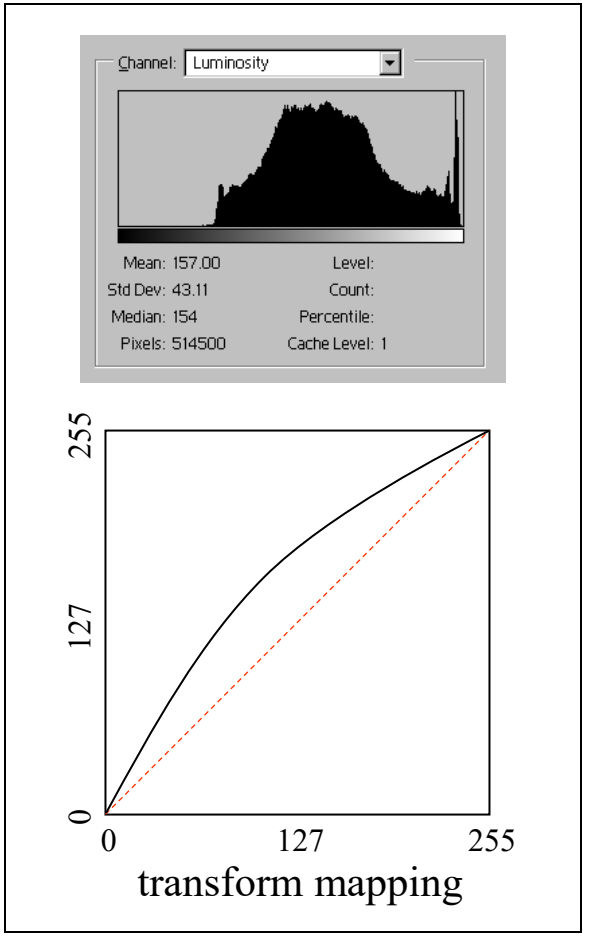
# Summary – (cont.)

## Contrast Stretch



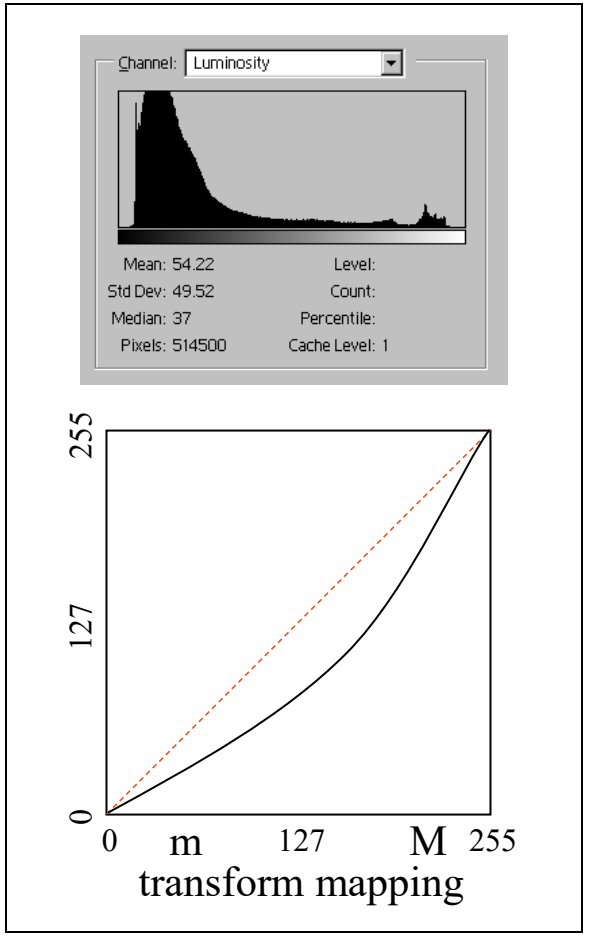
# Summary – (cont.)

## Increased Gamma



# Summary – (cont.)

## Decreased Gamma



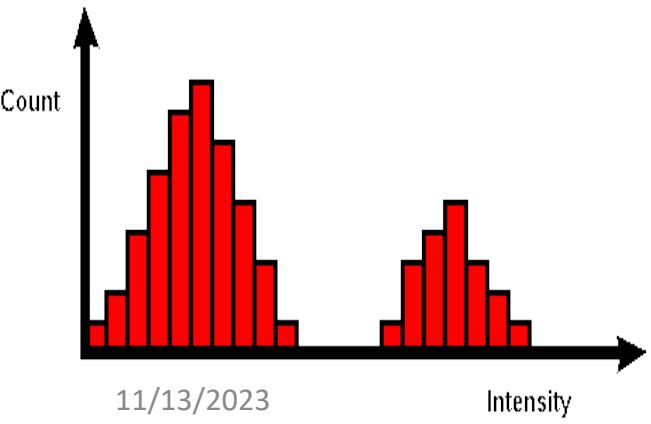
# 2. Histogram Processing

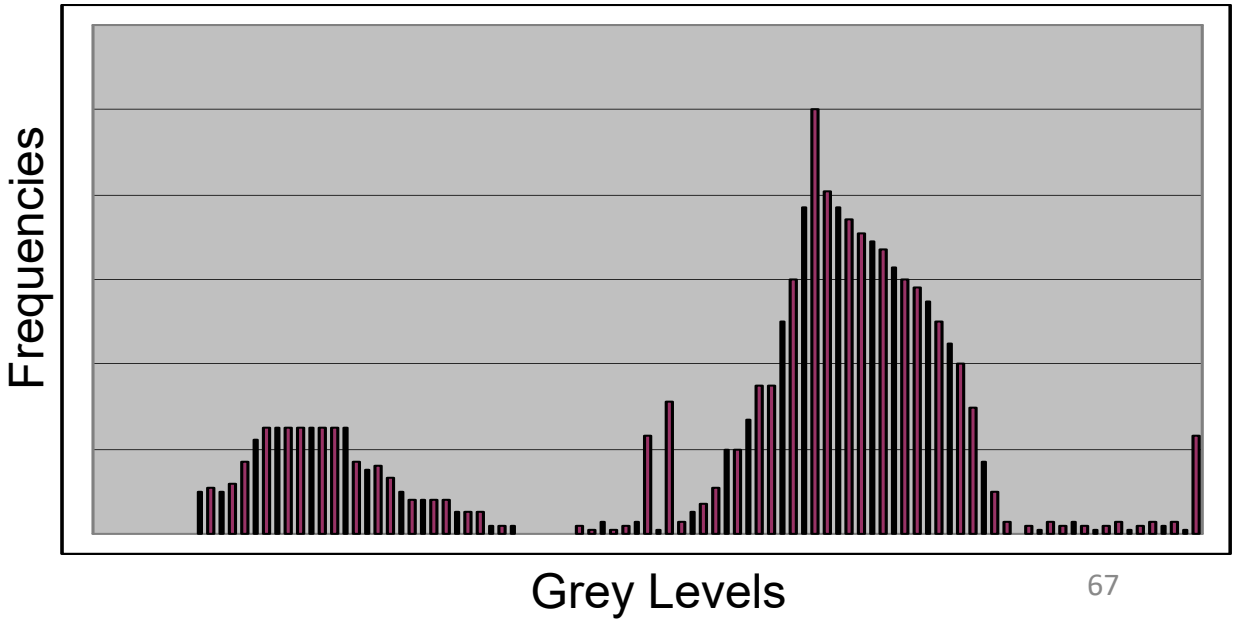
# Image Histogram

- **A way of viewing the distribution of color/intensities in an image.**

$h(x)$ = the number of pixels in $\mathbf{I}$ with intensity value $x$

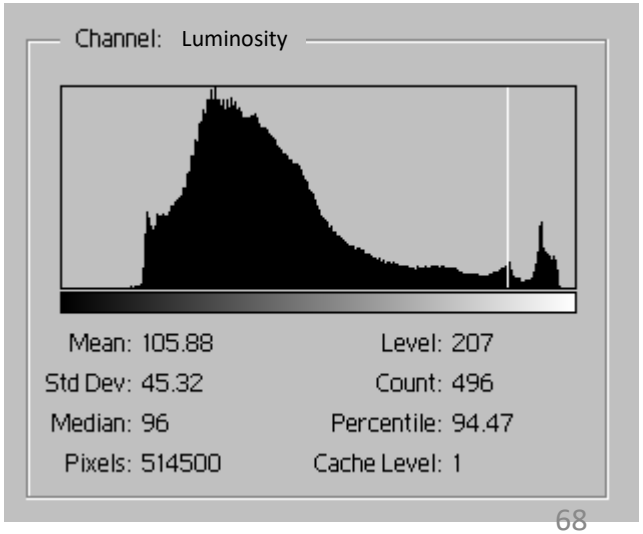
- **A plot of pixels values against pixel counts.**



# Image Histogram – (cont.)

**Gray Scale Image has one histogram while color images has three; one for each band (channel).**



# Image Histogram – (cont.)

## Color Image

- $\mathbf{I}$ has 3 histograms:

  $h_R(x)$ = number of pixels in $\mathbf{I}(:,:,1)$ with intensity value $x$

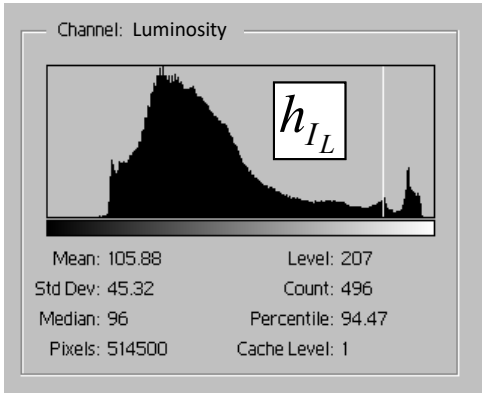
  $h_G(x)$ = number of pixels in $\mathbf{I}(:,:,2)$ with intensity value $x$

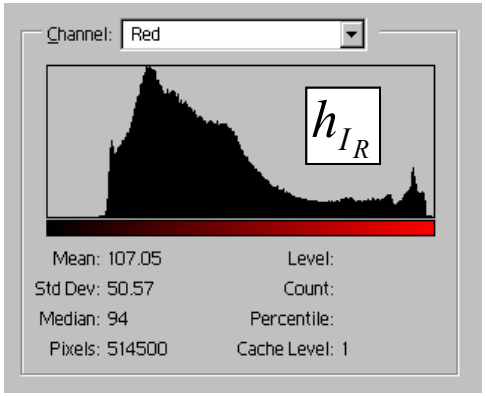
  $h_B(x)$ = number of pixels in $\mathbf{I}(:,:,3)$ with intensity value $x$

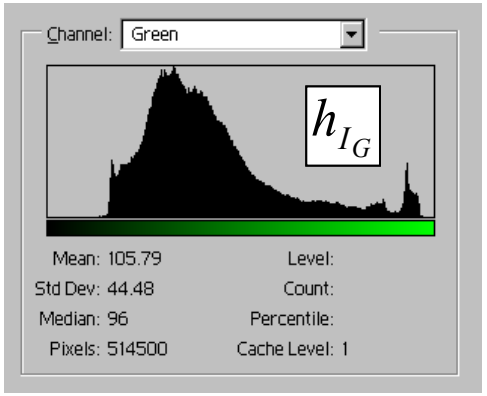
# Image Histogram – (cont.)

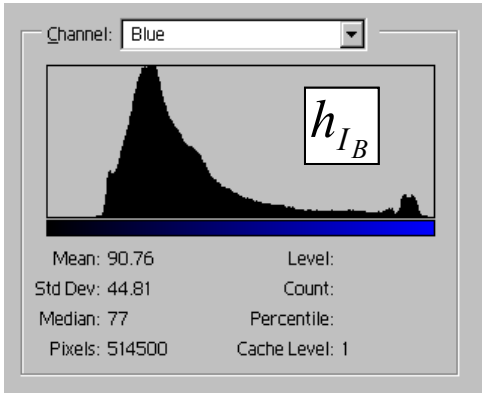
## Color Image

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



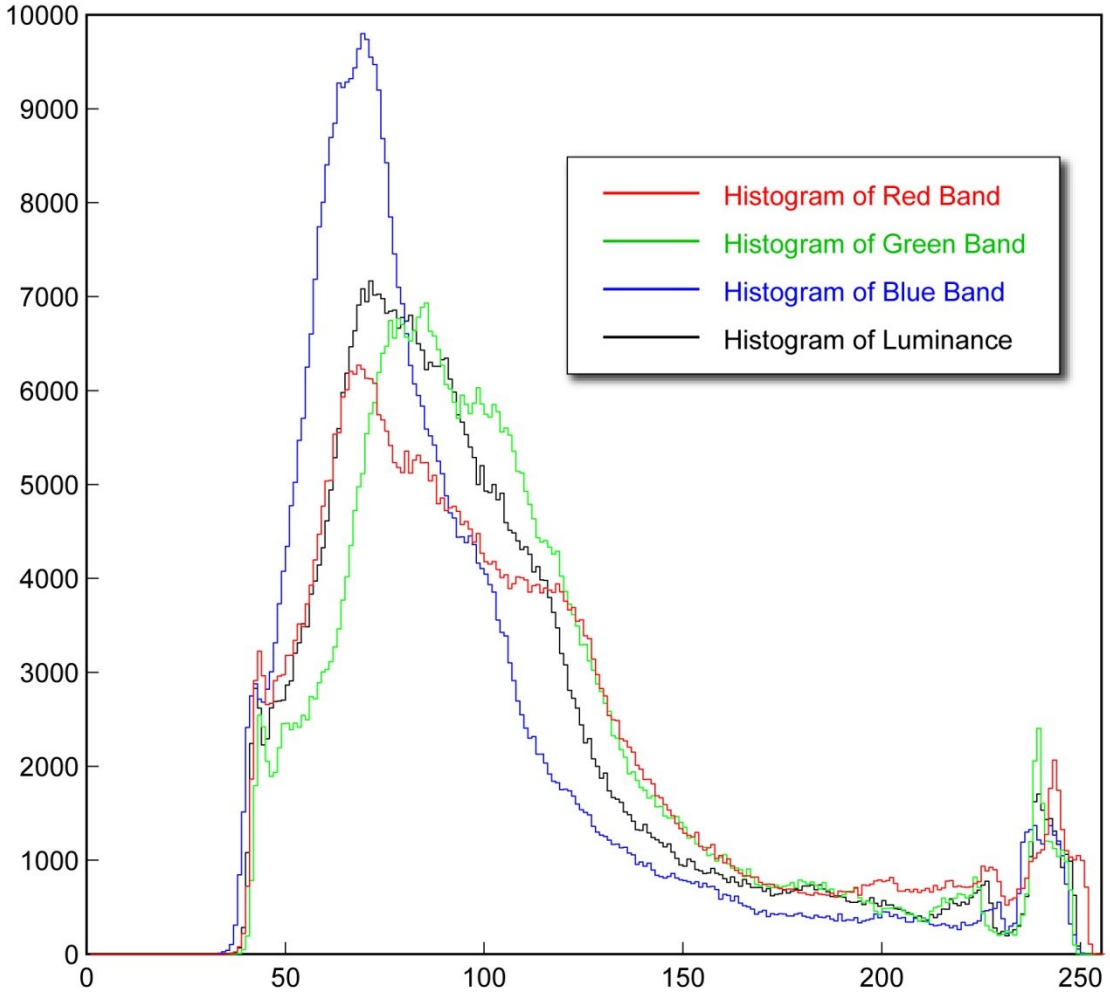



# Image Histogram – (cont.)

## Color Image



# Image Histogram – (cont.)

## Color to gray (in GIMP)

- **Lightness**

Lightness = ½ × (max(R,G,B) + min(R,G,B))

- **Luminosity**

Luminosity = 0.21 × R + 0.72 × G + 0.07 × B

- **Average**

Average Brightness = (R + G + B) ÷ 3

# Image Histogram – (cont.)

## Histogram Definition (normalized)

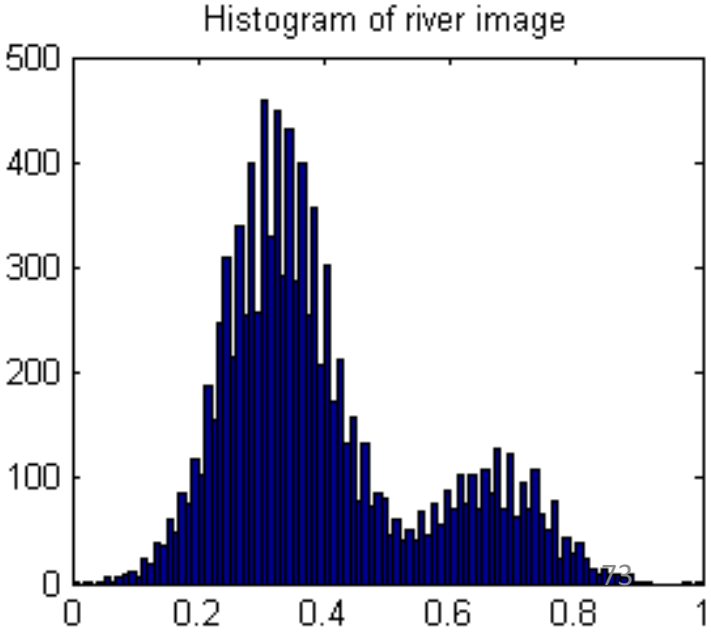
- **An estimate to the probability of occurrence of intensity level $r_k$ in an image.**

$$p(r_k) = \frac{n_k}{MN}, k = 0,1,2,\ldots,L-1$$

$MN$ : totoal number of pixels

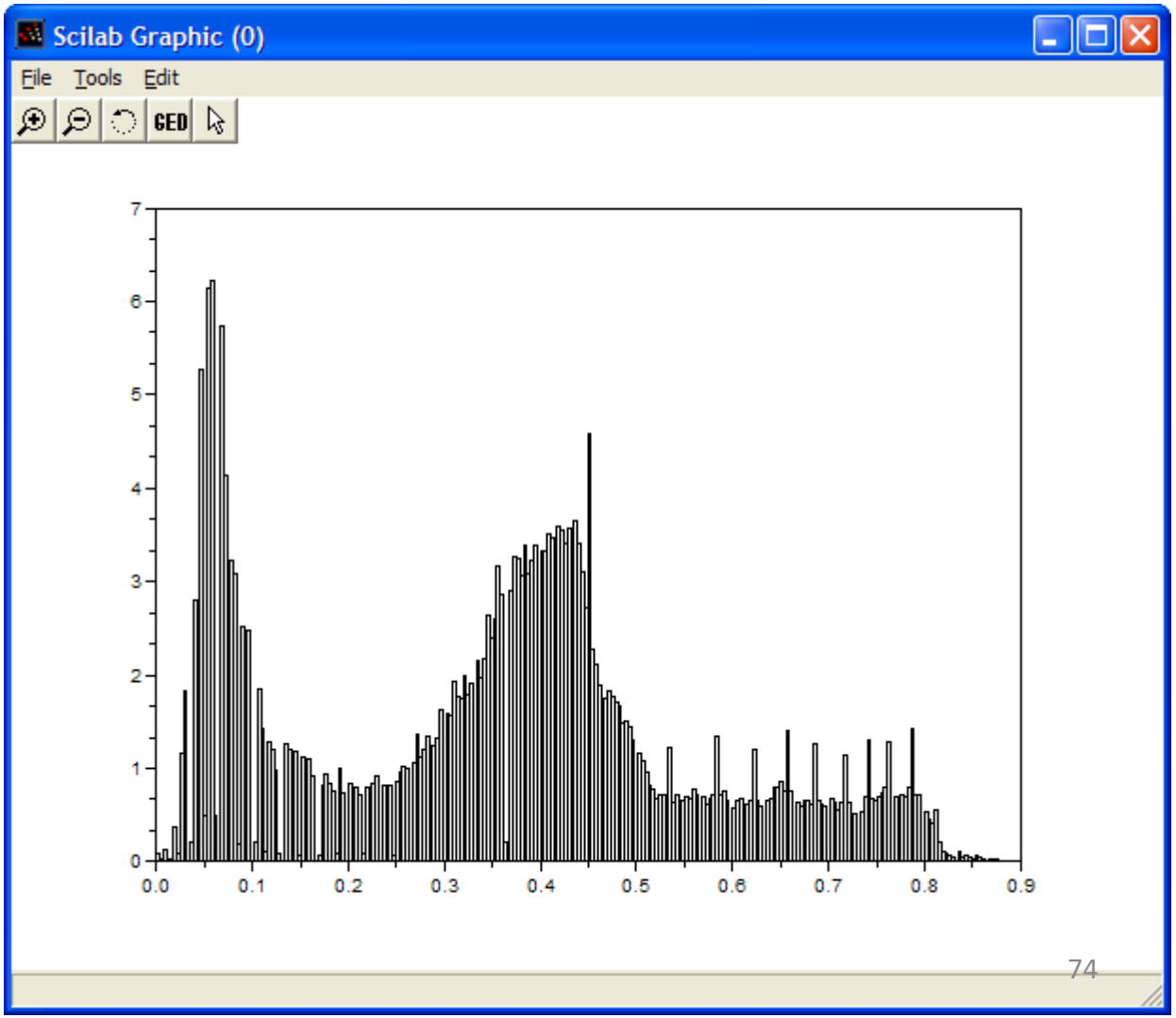
$n_k$ : number of pixels having value $r_k$

- **Not unique.**



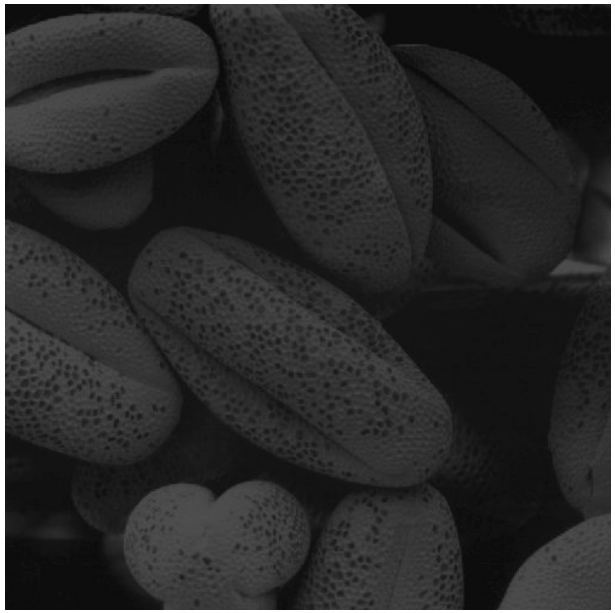
# Image Histogram – (cont.)

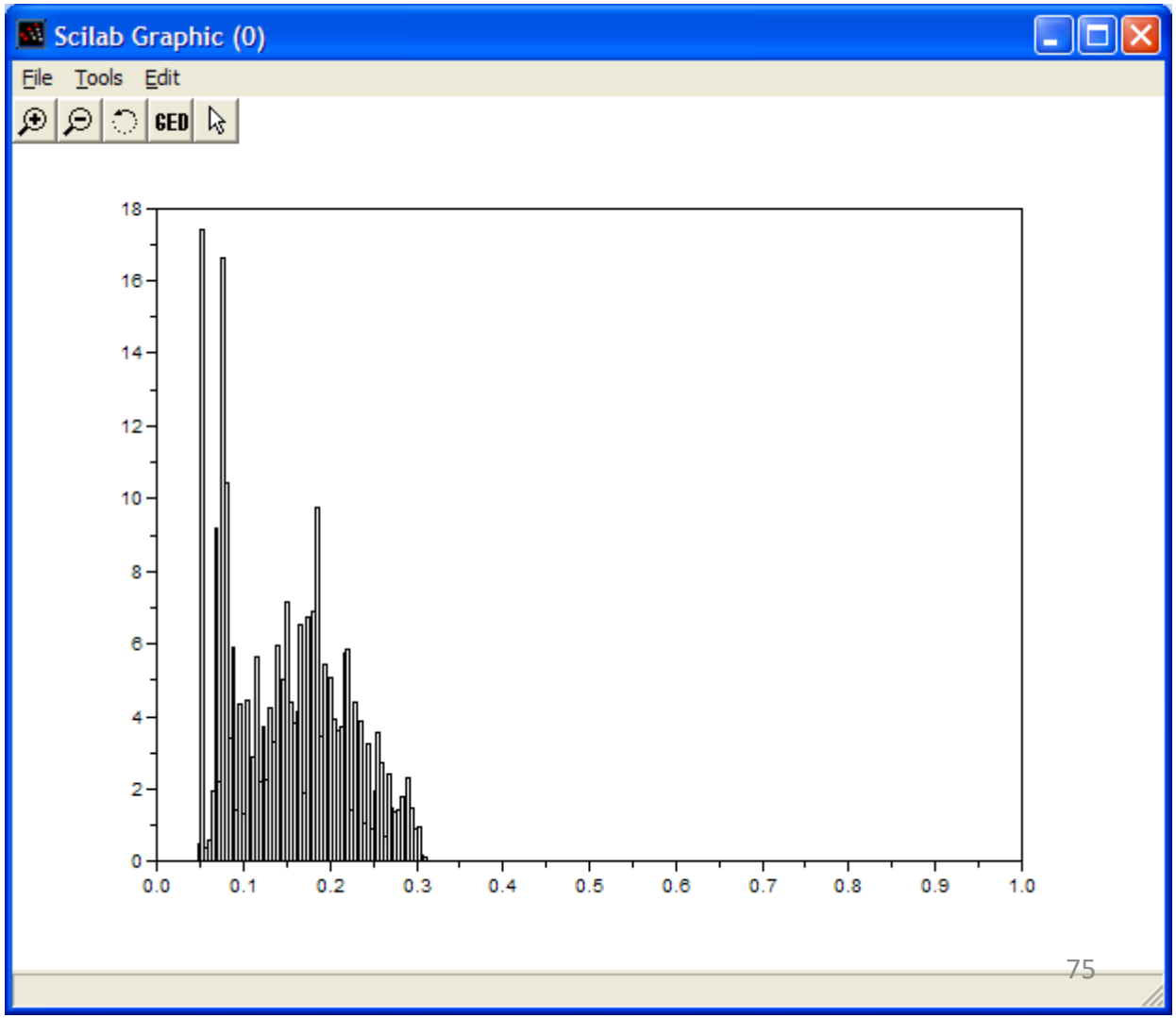
## Examples



# Image Histogram – (cont.)

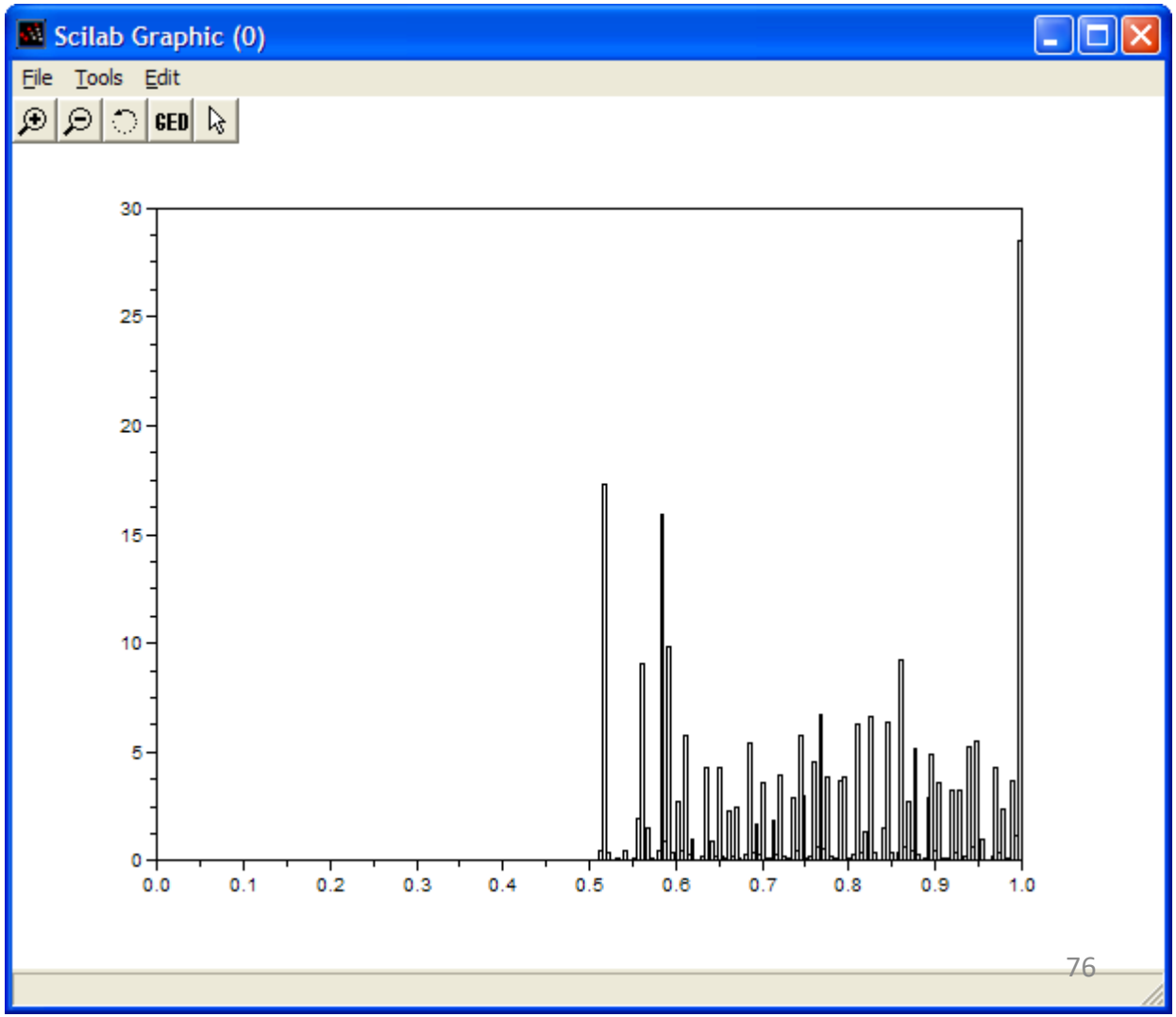
## Examples





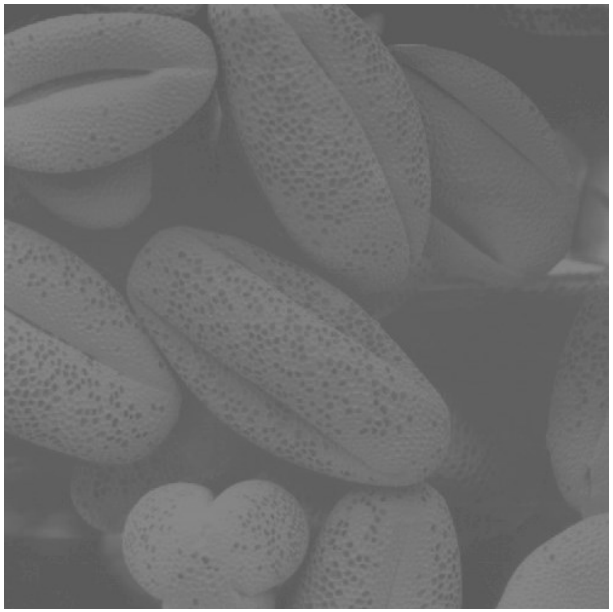
# Image Histogram – (cont.)

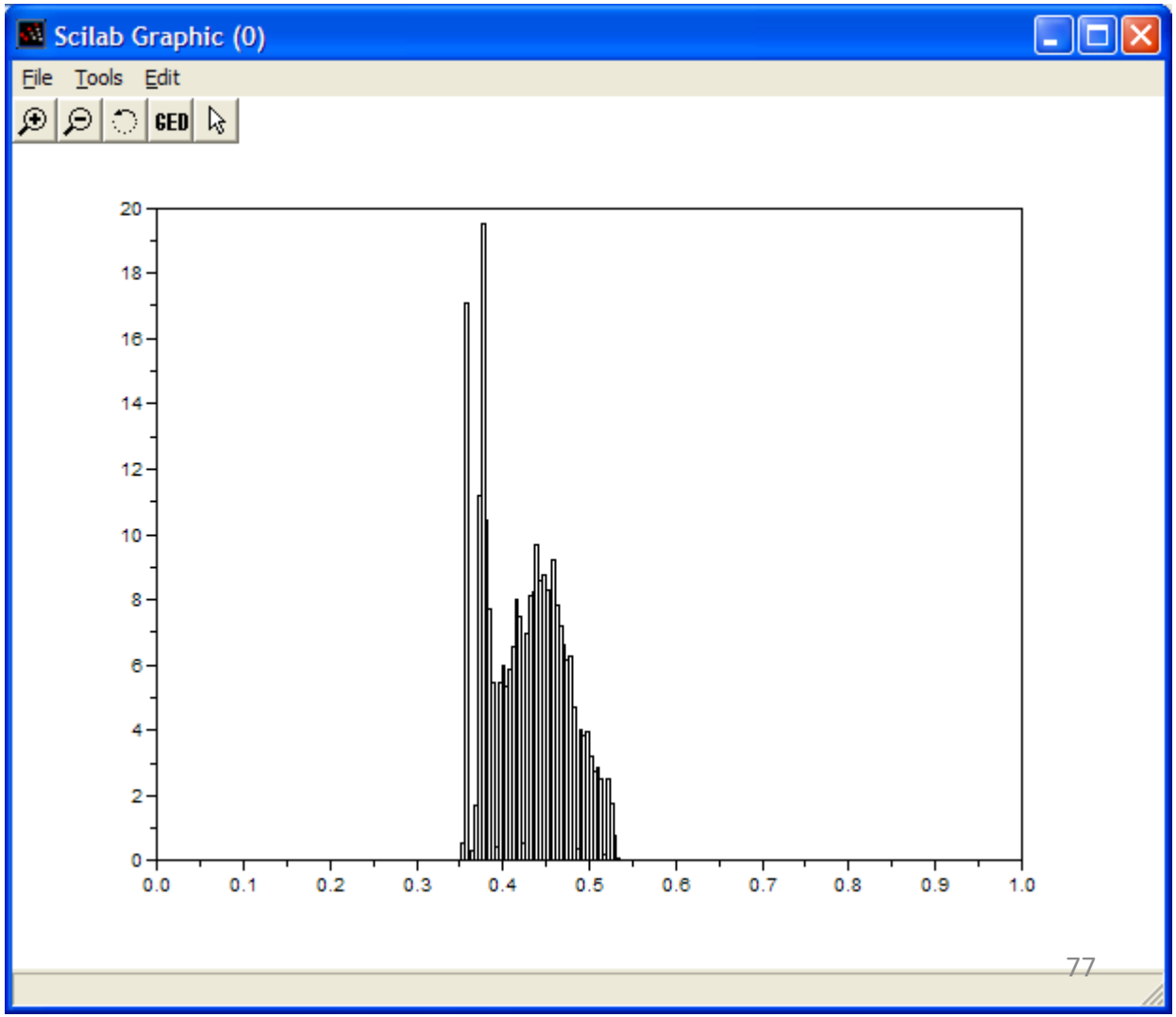
## Examples



# Image Histogram – (cont.)

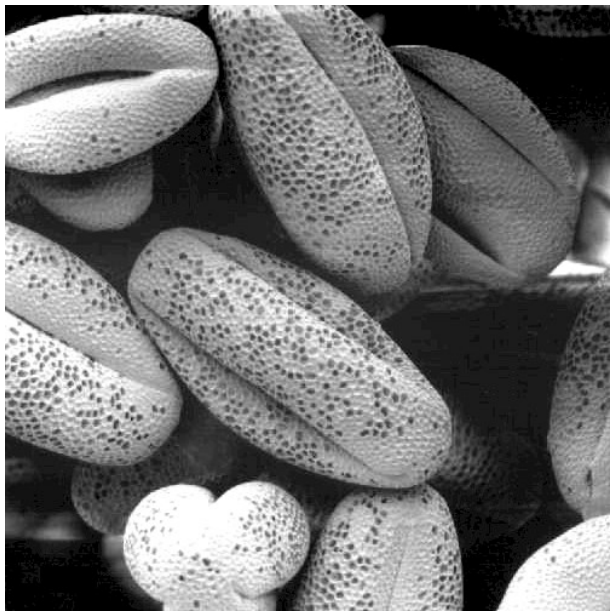
## Examples

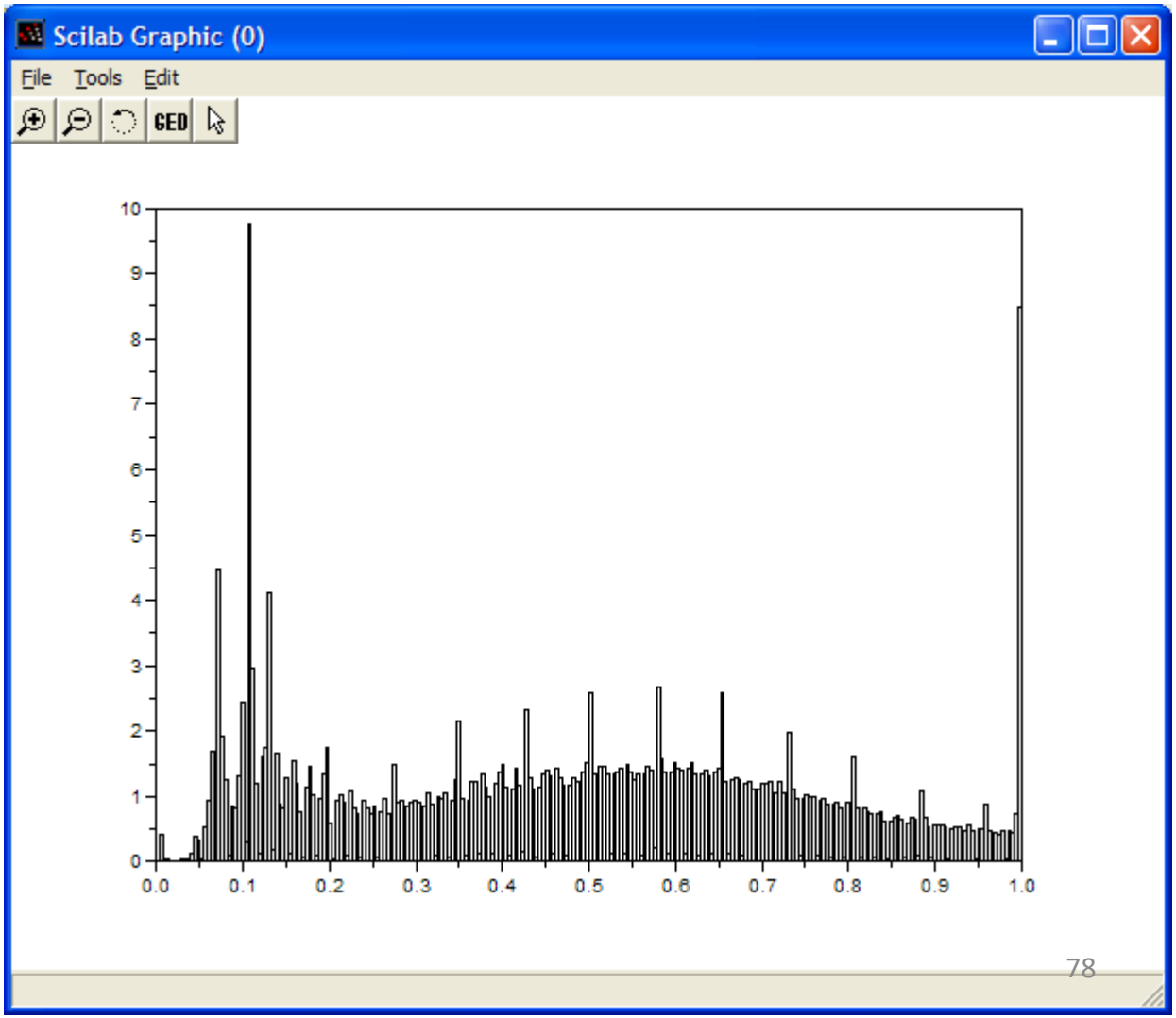




# Image Histogram – (cont.)

## Examples



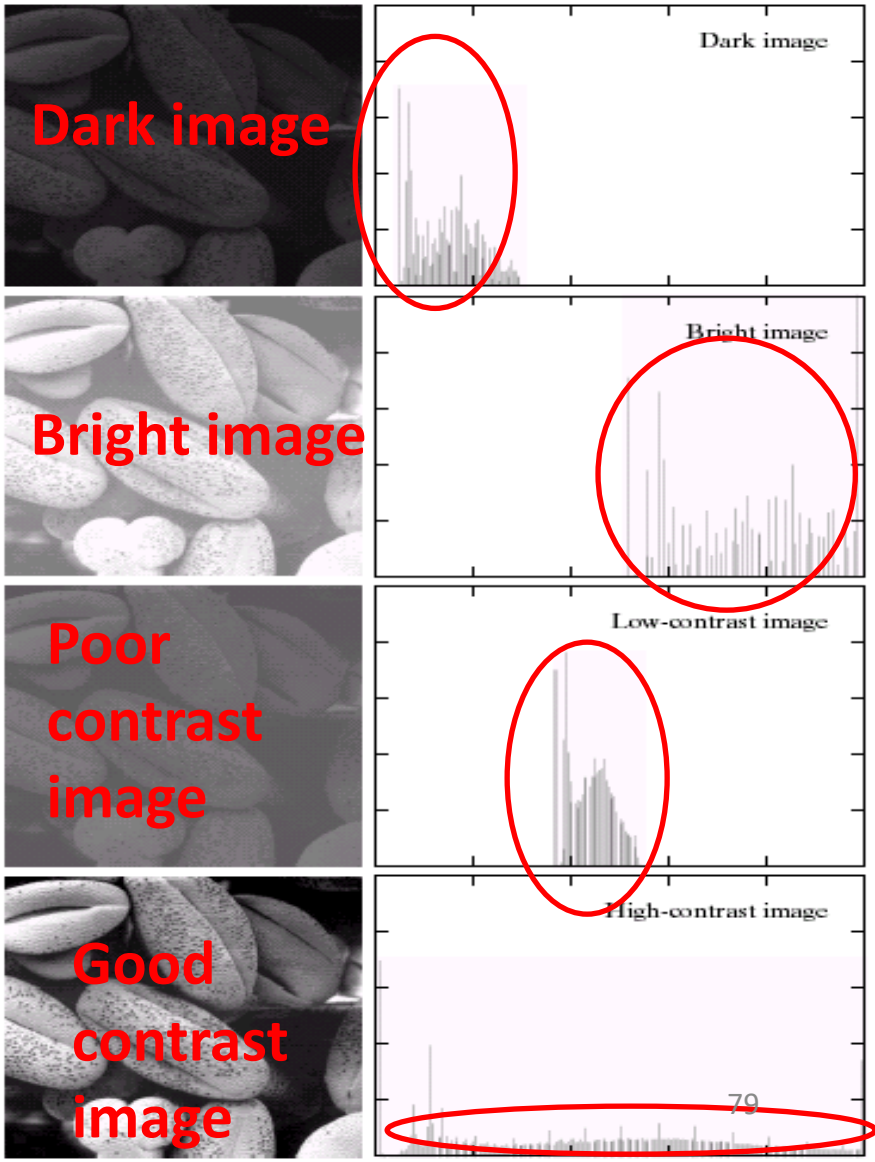


# Image Histogram – (cont.)

## Examples

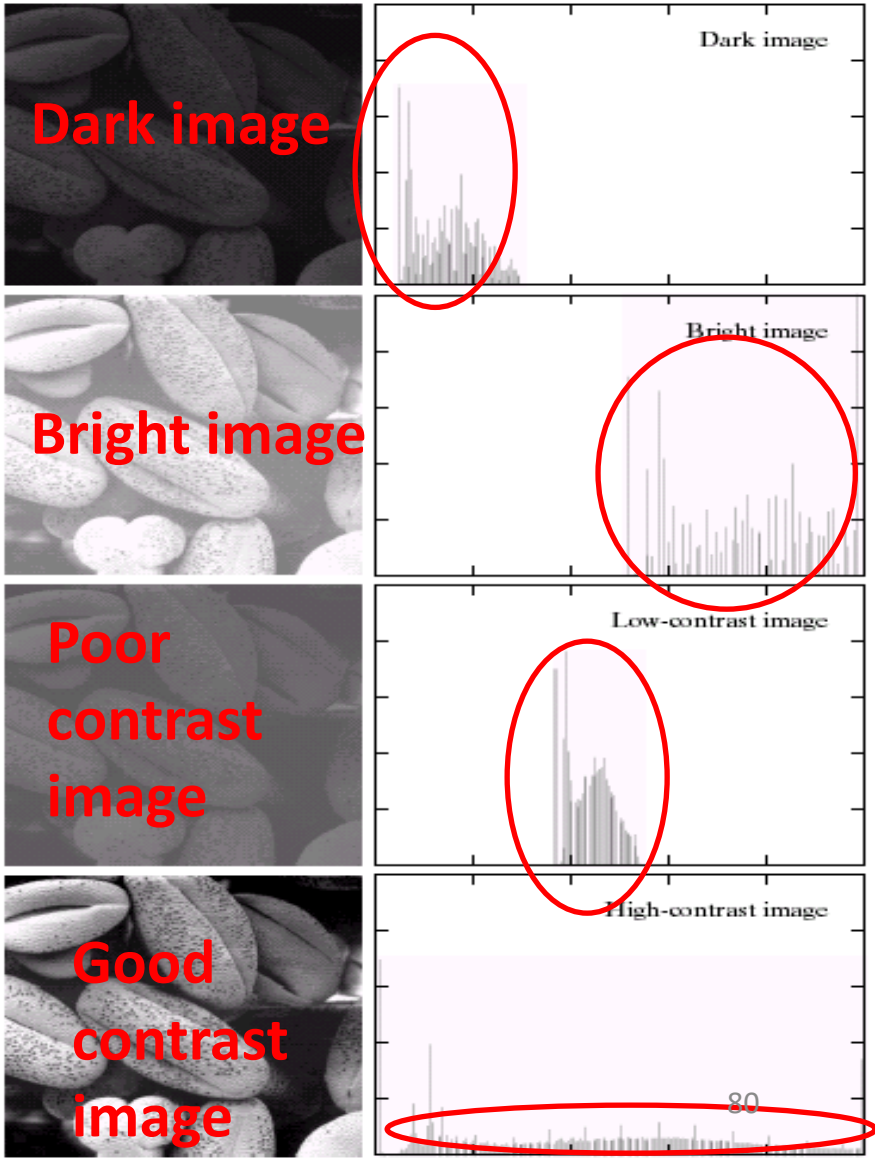
- **Note that the high contrast image has the most evenly spaced histogram.**

**Can we do that?**



# Histogram Processing

- **Histogram equalization**
- **Histogram matching**



# Histogram Processing – (cont.)

## Histogram Equalization

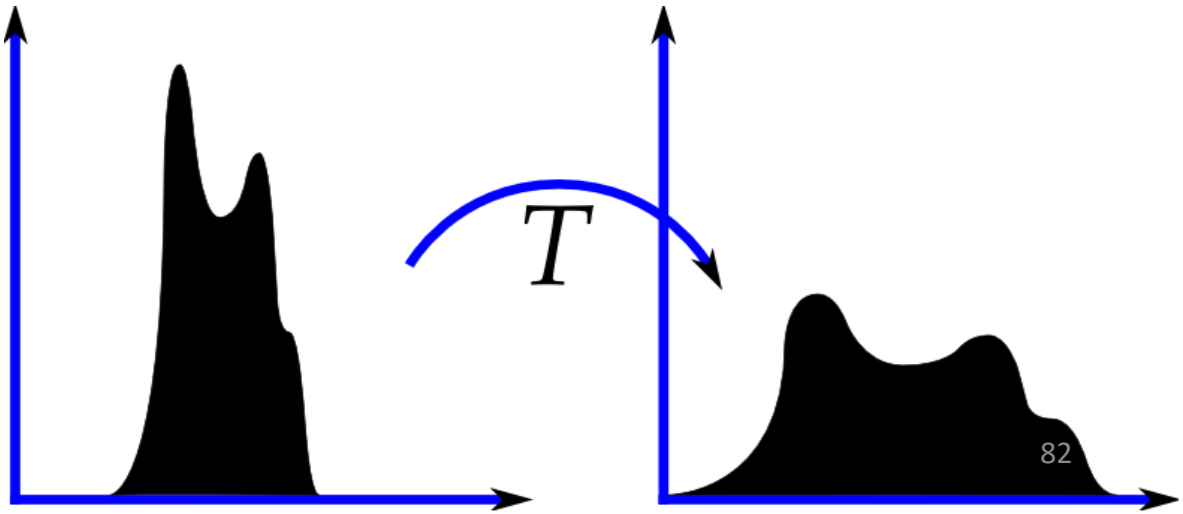
- Spreading out the frequencies in an image (or equalizing the image) is a simple way to improve dark or washed out images.
- Transfer the gray levels so that the histogram of the resulting image is equalized to be a constant (theoretically).
- Remap pixel values for image $\mathbf{I}$ so that its histogram is as close to constant as possible.

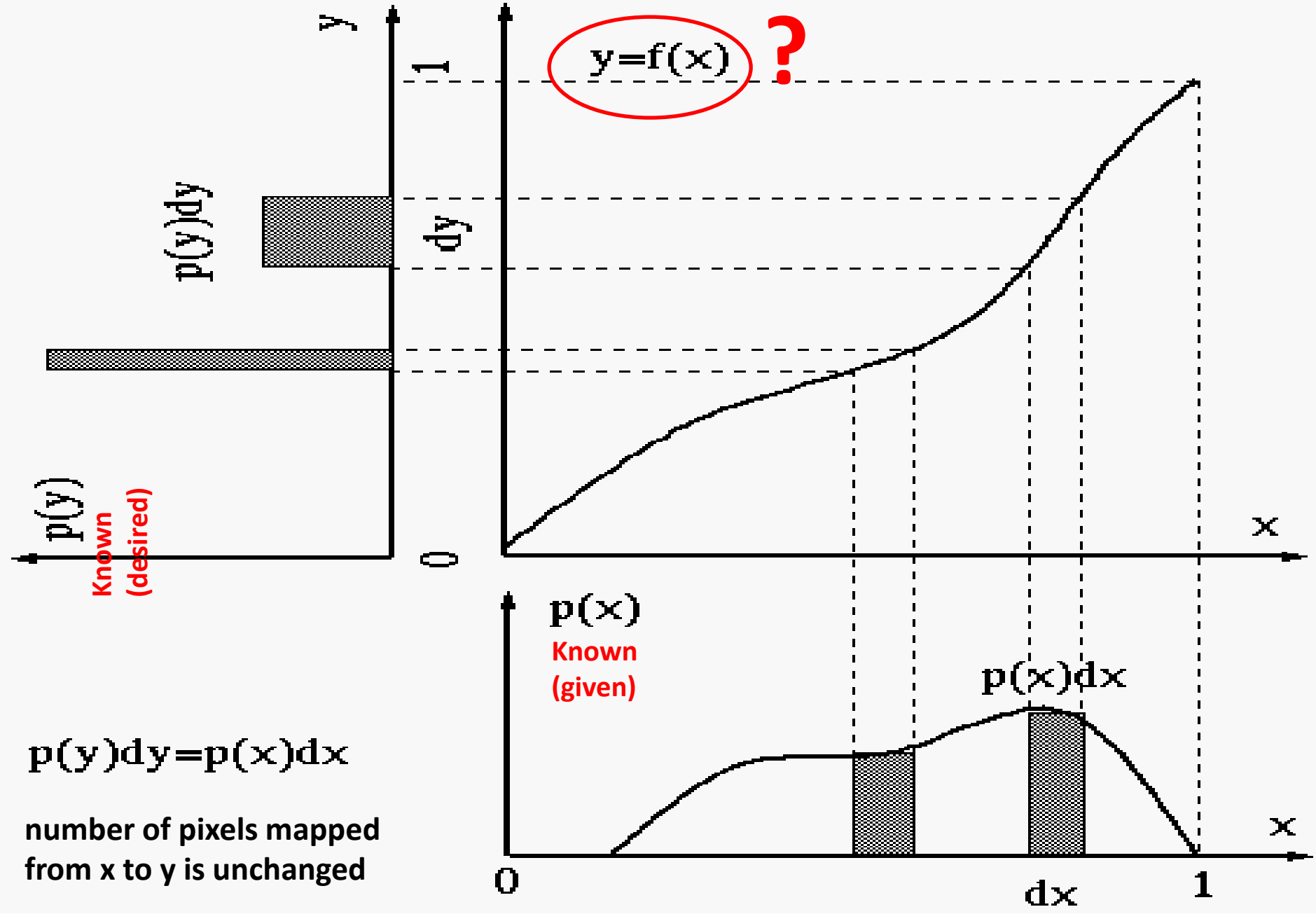
# Histogram Processing – (cont.)

## Histogram Equalization

Why?

- Treating intensity levels as random variables.
- Applying transform $T$ on the PDF of the input intensity to produce output intensities with equally spread probabilities (constant PDF).
- CHOICE OF $T$.






$$p(y)dy = p(x)dx$$

**number of pixels mapped from x to y is unchanged**

# Histogram Processing – (cont.)

## Histogram Equalization

- **For any given mapping function $y = f(x)$ between the input and output images the following holds**

$$p(y)\,dy = p(x)\,dx$$

**if the gray levels are assumed to be in the ranges between 0 and 1 then p(y) = 1 and**

$$dy = p(x)\,dx, \quad \text{or} \quad \frac{dy}{dx} = p(x)$$

$$y = f(x) = \int_0^x p(u)du = P(x) - P(0) = P(x)$$

where

$$P(x) = \int_0^x p(u)du, \qquad P(0) = 0$$

**The Cumulative Distribution Function of x**

# Histogram Processing – (cont.)

## Histogram Equalization

- **PDF of a random variable $X$ is a measure of the likelihood for $X$ to occur at a given point .**

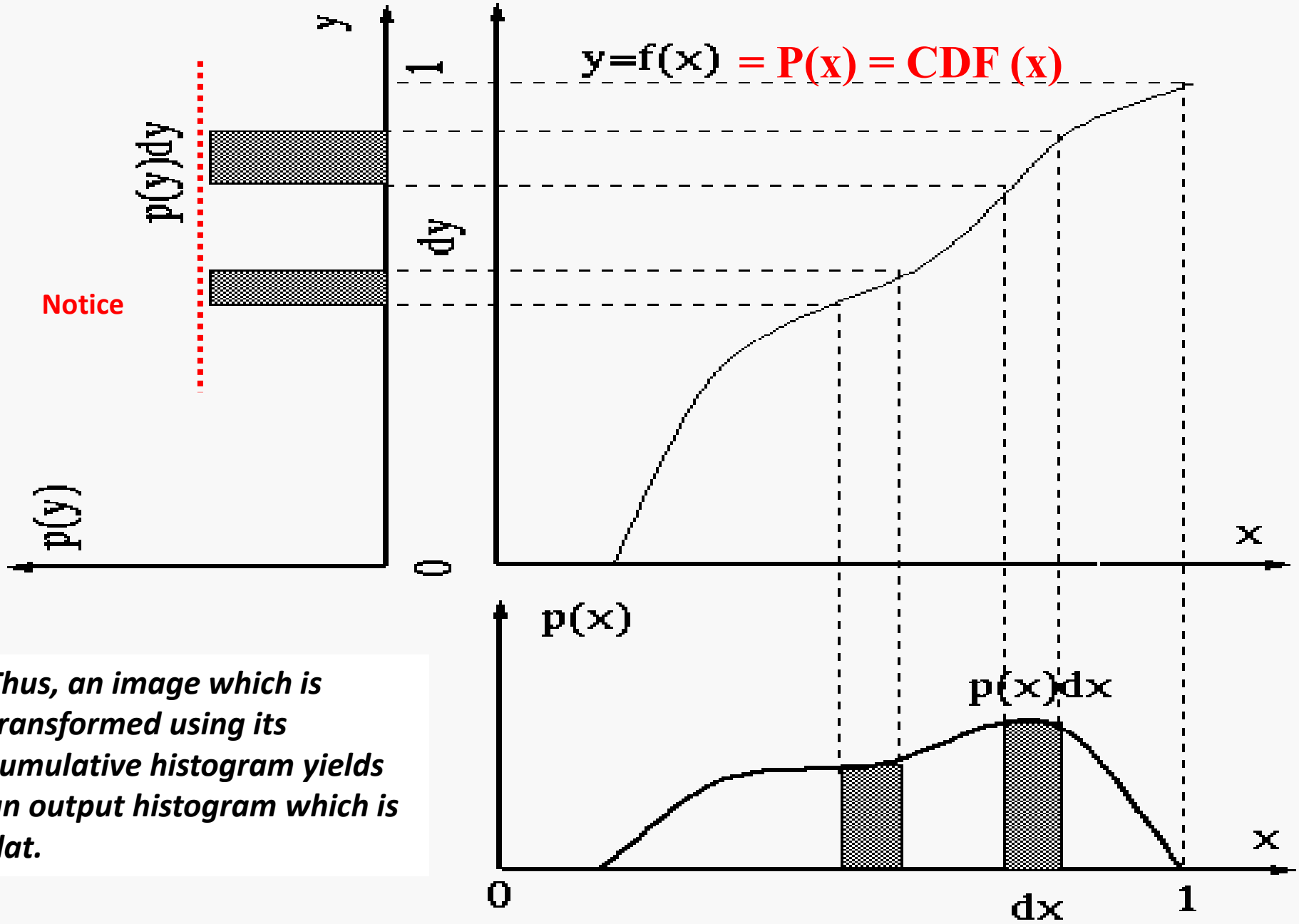
$$P_x = \text{Prob}(x_1 < X < x_2) = \int_{x_1}^{x_2} f(x)dx$$

This is the probability that an arbitrary pixel x has value f(x).

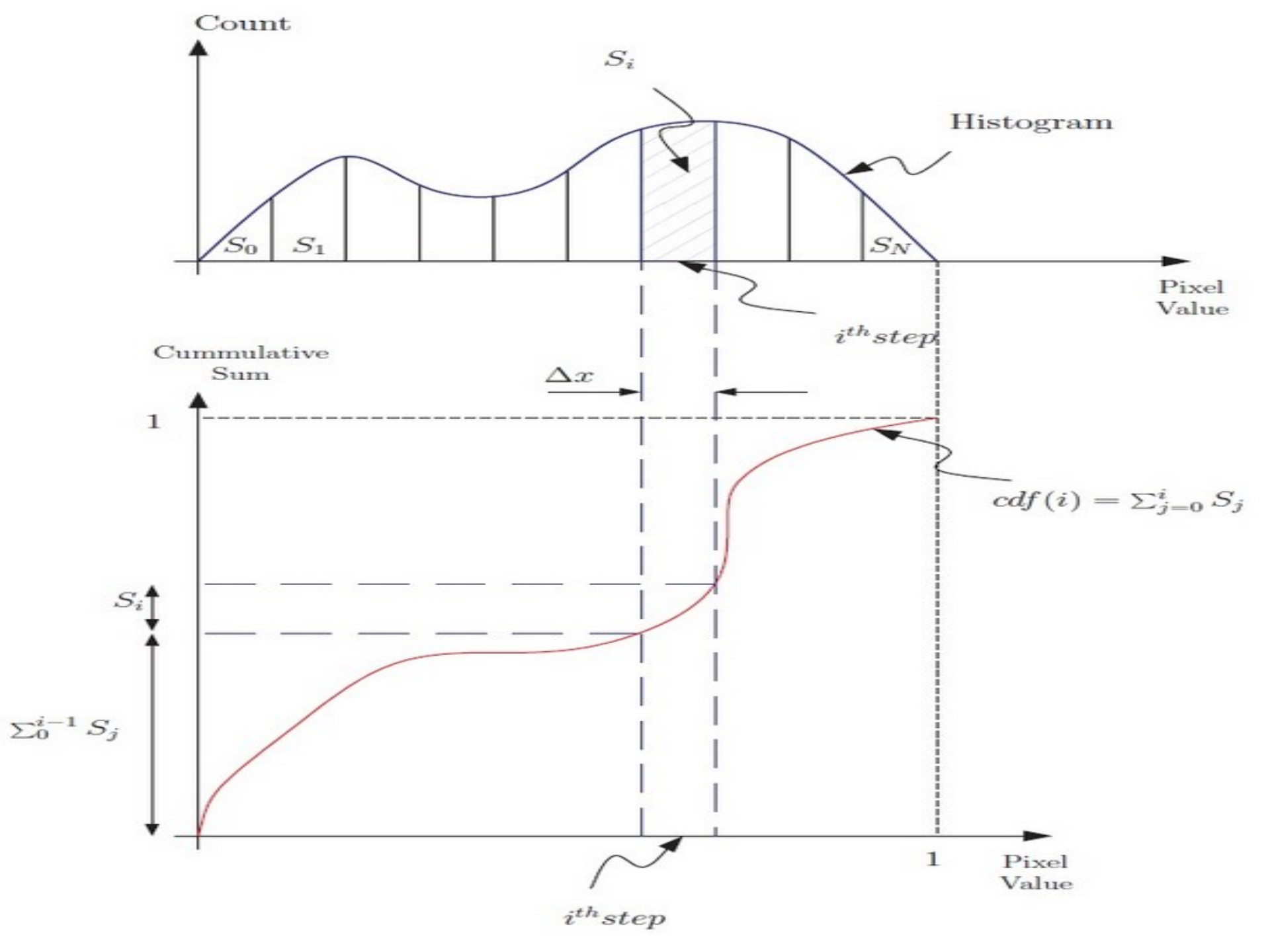
- **Integrating gives the Cumulative Distribution Function CDF of $X$:**

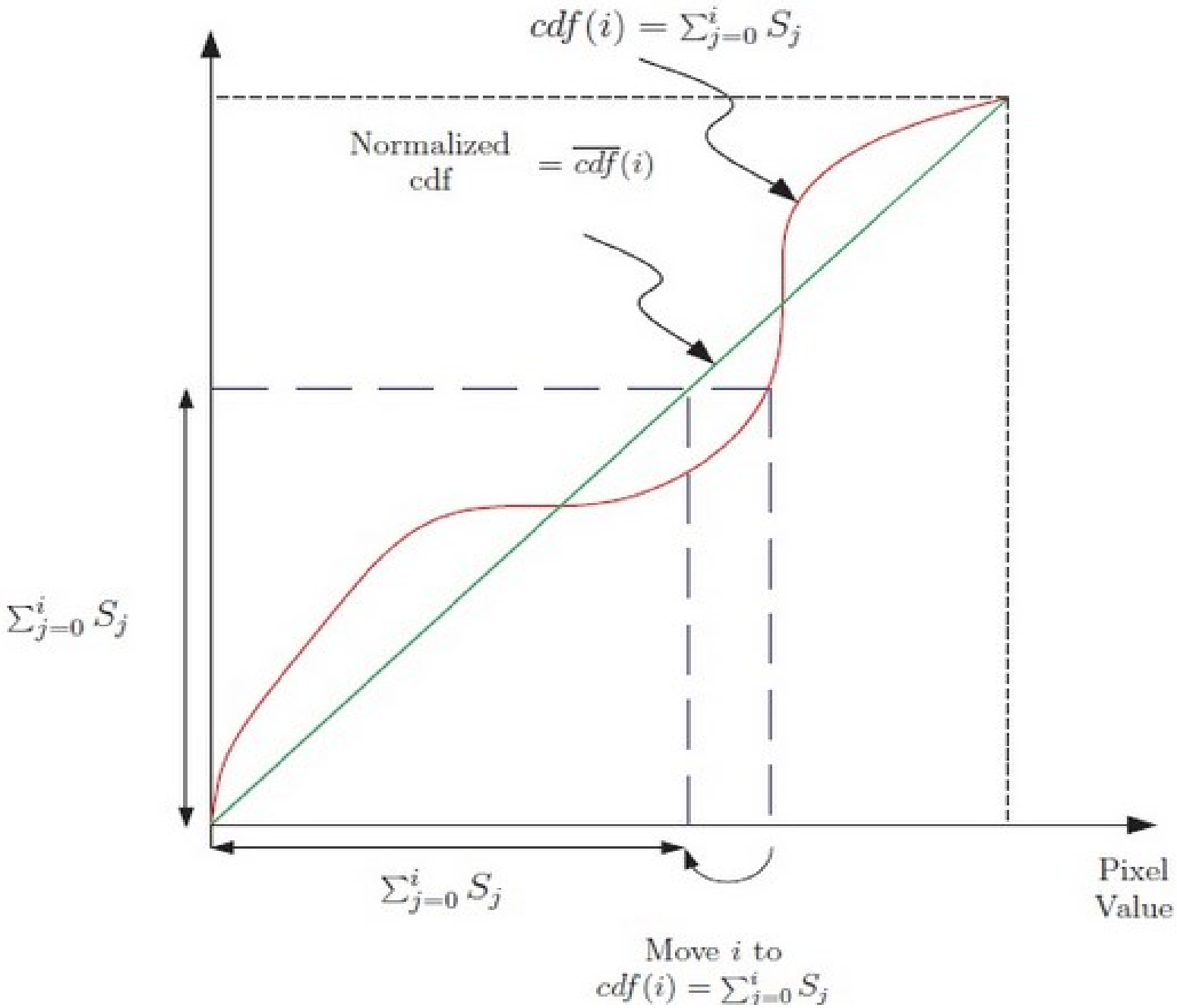
$$F_x = \text{Prob}(X \le x) = \int_{-\infty}^{x} f(x)dx$$

This is the probability that any given pixel from f has value less than or equal to x.




***Thus, an image which is transformed using its cumulative histogram yields an output histogram which is flat.***


Count
$S_i$
Histogram
$S_0$
$S_1$
$S_N$
Pixel Value
$i^{th}$ step
Cummulative Sum
$\Delta x$
1
$cdf\,(i) = \Sigma_{j=0}^{i} S_j$
$S_i$
$\Sigma_0^{i-1} S_j$
1
Pixel Value
$i^{th}$ step


$cdf(i) = \sum_{j=0}^{i} S_j$
Normalized cdf
$= \overline{cdf}(i)$
$\sum_{j=0}^{i} S_j$
$\sum_{j=0}^{i} S_j$
Pixel Value
Move $i$ to $cdf(i) = \sum_{i=0}^{i} S_j$

# Histogram Processing – (cont.)

## Histogram Equalization - Discrete

**if the gray levels are assumed to be in the ranges between 0 and L then p(y) = 1/(L-1) and**

$$y = (L-1)\int_0^x p\ (u)du$$

- **For discrete intensity values we have**

$$x \in \{0, 1, \cdots, L-1\}$$

$$y' = f[x] \triangleq \sum_{i=0}^{x} h[i] = H[x]$$

$$h[i] = \frac{n_i}{\sum_{i=0}^{L-1} n_i} = \frac{n_i}{N} \quad \text{and} \quad \sum_{i=0}^{L-1} h[i] = 1$$

# Histogram Processing – (cont.)

## Histogram Equalization - Discrete

$\mathbf{x == r_k}$ $\mathbf{y == s_k == f(x) == f(r_k) = T(r_k)}$

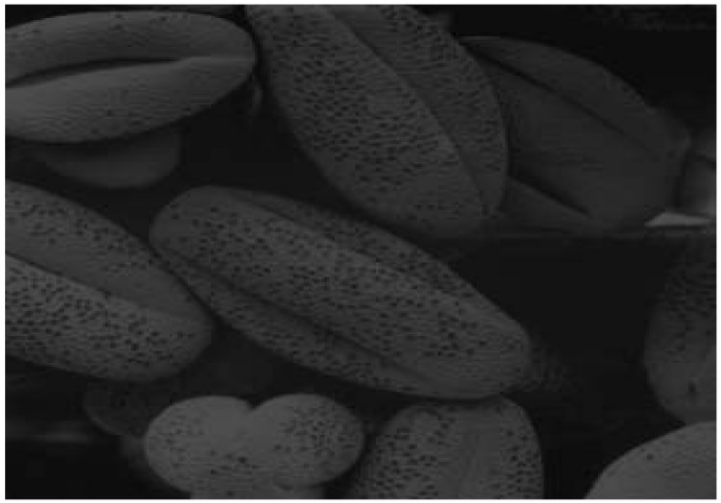
$$s_k = T(r_k) = \frac{(L-1)}{N}\sum_{j=0}^{k} n_j, \qquad k = 0, 1, 2, \ldots, L-1$$

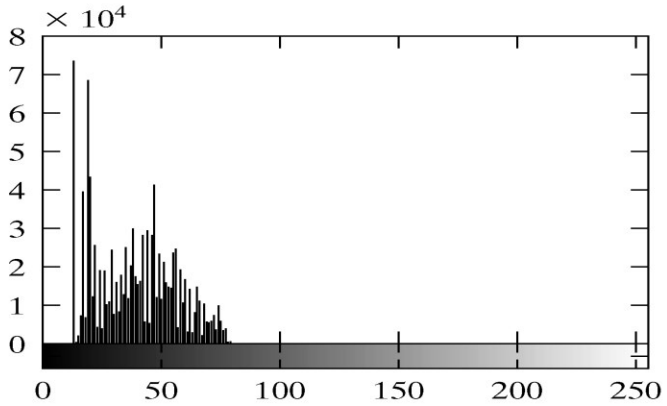
$r_k$: input intensity
$s_k$: output intensity
$k$: the intensity level counter
$n_j$: the histogram of intensity $j$
$N$: total number of pixels

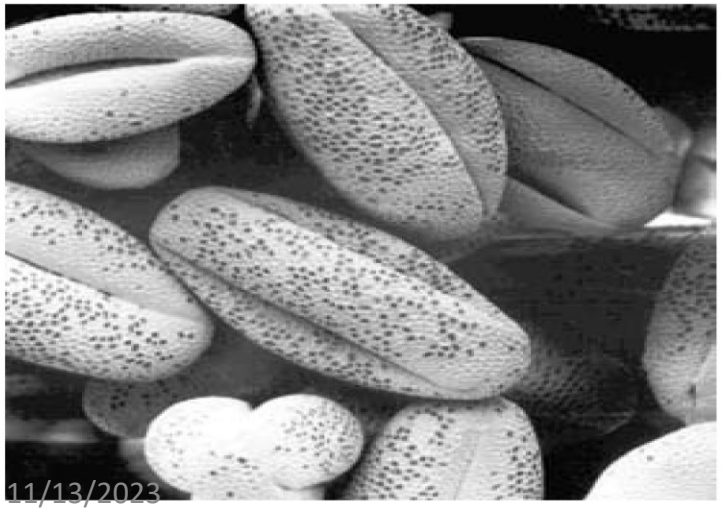
***Remember to round to the nearest integer.***

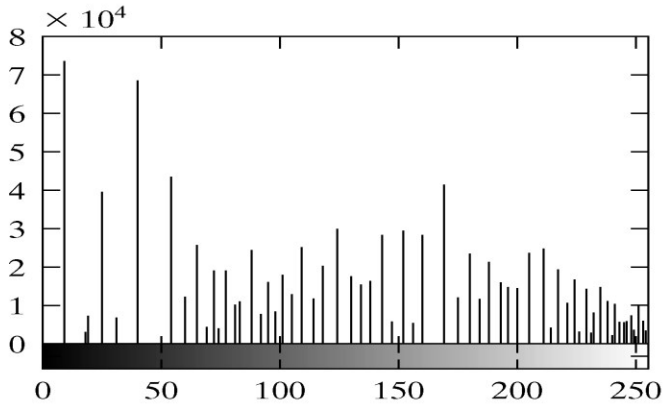
# Histogram Processing – (cont.)

## Histogram Equalization





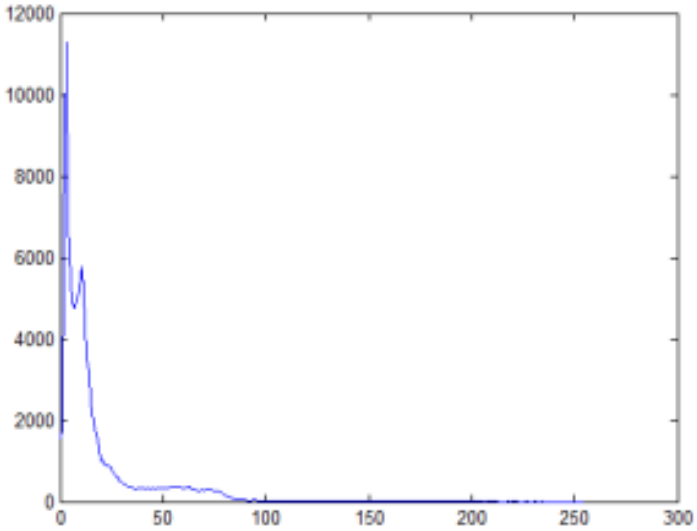





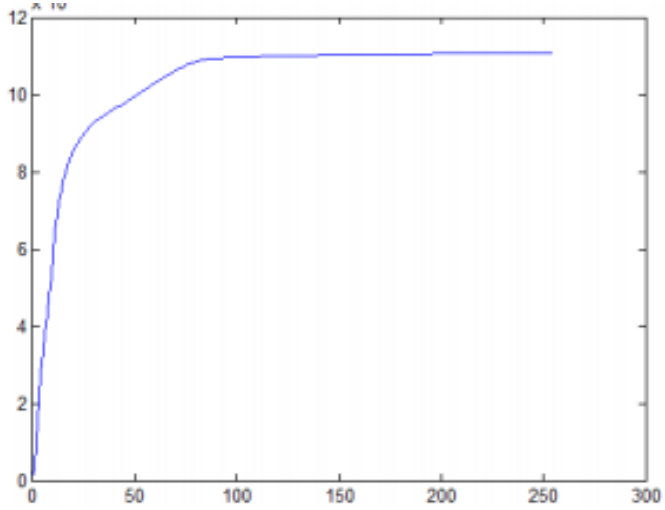
# Histogram Processing – (cont.)

## Histogram Equalization

Example image




Histogram of example image

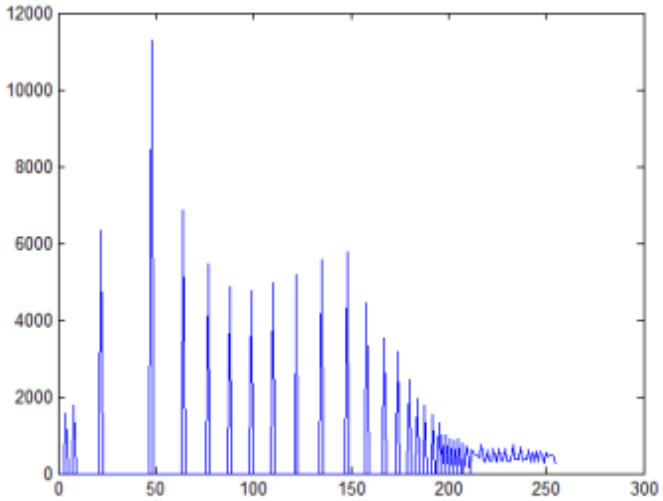



Cumulative histogram of example image

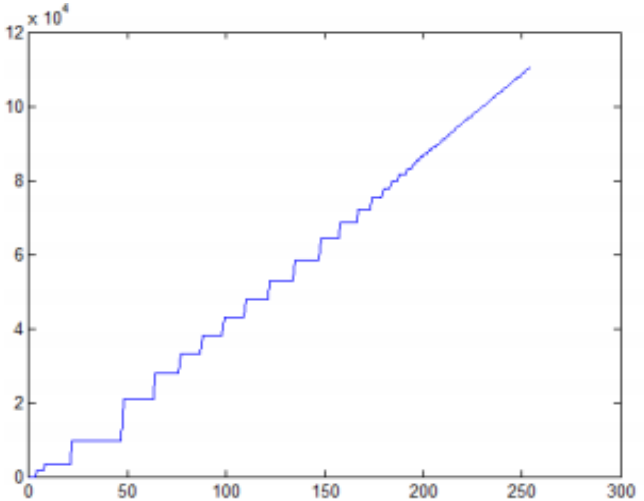
# Histogram Processing – (cont.)

## Histogram Equalization

Example image after applying algorithm



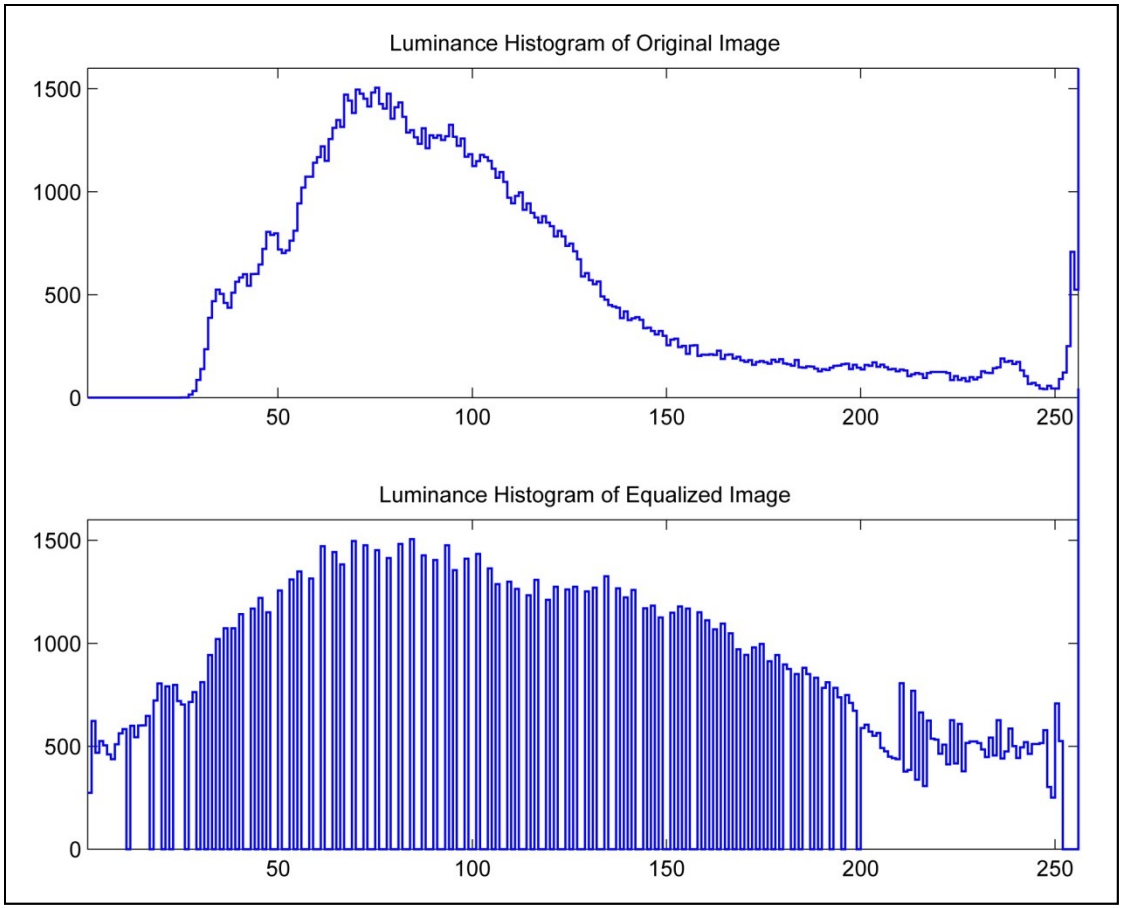

New equalized histogram




Equalized cumulative histogram

# Histogram Processing – (cont.)

## Histogram Equalization

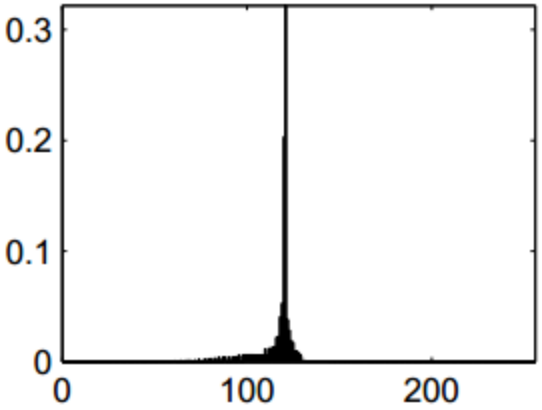


# Histogram Processing – (cont.)

## Histogram Equalization

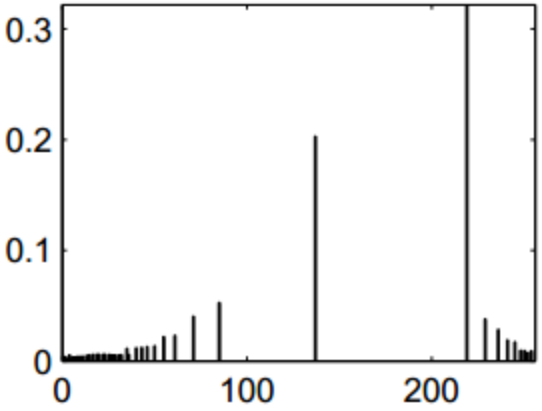
original image

original histogram




transformed image

transformed histogram



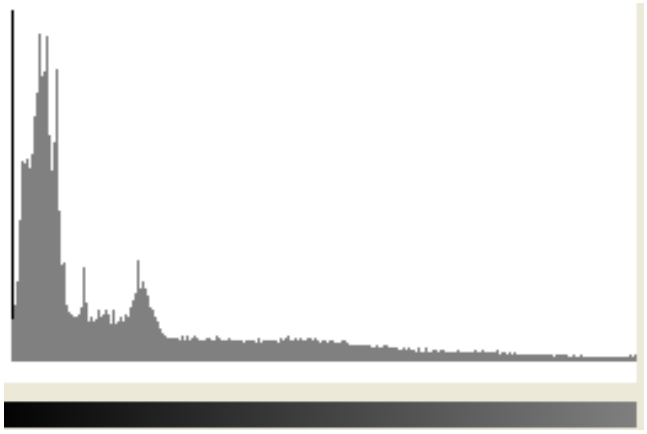
# Histogram Processing – (cont.)

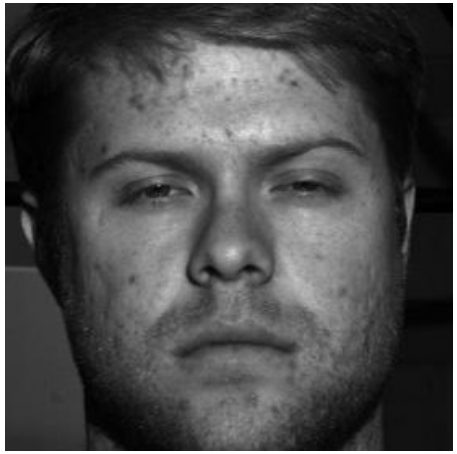
## Histogram Matching (Specification)

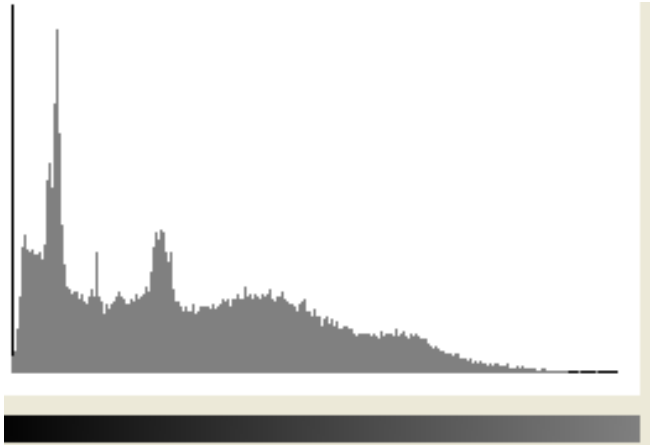
- **Mapping the histogram of an image to a known histogram of another image.**
- **A method of color adjustment of two images using the image histograms.**

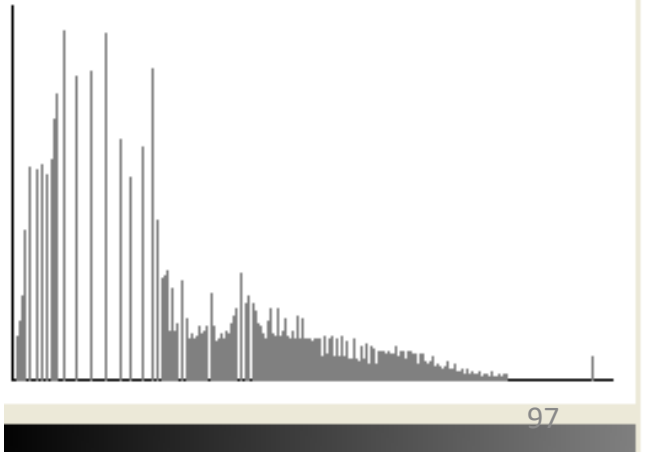
# Histogram Processing – (cont.)

## Histogram Matching (Specification)









# Histogram Processing – (cont.)

## Histogram Matching (Specification)

- **Equalize the histogram of the input image.**

$$y = f(x) = \int_0^x p_x(u)du$$

- **Equalize the specified (desired) histogram.**

$$y' = g(z) = \int_0^z p_z(u)du$$

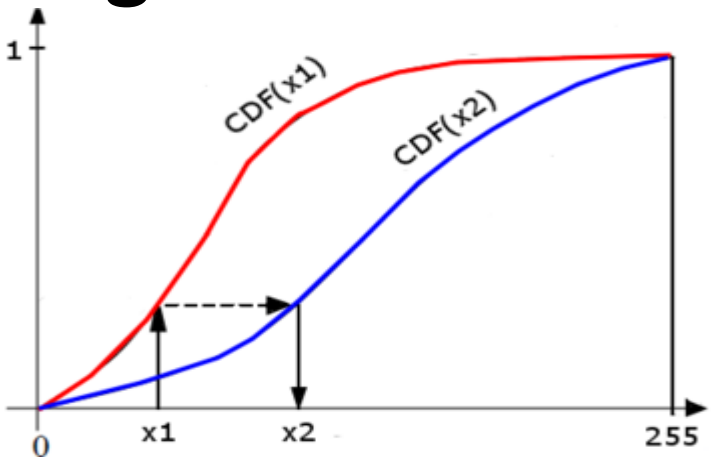
- **Relate the two equalized histograms.**

$$z = g^{-1}(y')$$

$$y = y'.$$

Since both images have the same equalized histogram, they are actually the same image.

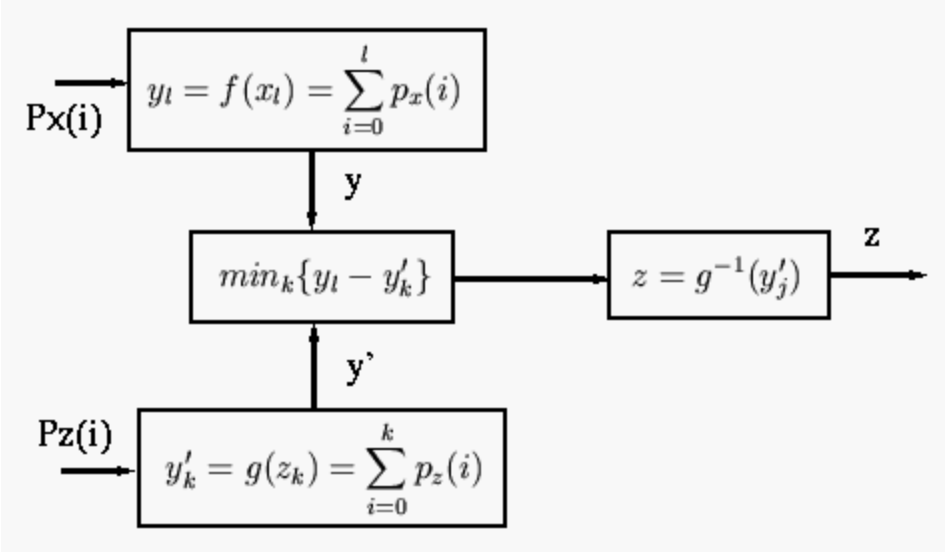
$$z = g^{-1}(y') = g^{-1}(y) = g^{-1}(f(x))$$



# Histogram Processing – (cont.)

## Histogram Matching (Specification) - Discrete

- **Discrete histograms and are not necessarily identical. We need to relate each gray level in x to a gray level in z, such that for every level i, find level j where hist1 best matches hist2.**



# Histogram Processing – (cont.)

## Histogram Matching (Specification) - Discrete

- **Step 1:** Find histogram of input image $h_x$, and find its cumulative $H_x$, the histogram equalization mapping function:

$$H_x[j] = \sum_{i=0}^{j} h_x[i]$$

- **Step 2:** Specify the desired histogram $h_z$, and find its cumulative $H_z$, the histogram equalization mapping function:

$$H_z[j] = \sum_{i=0}^{j} h_z[i]$$

- **Step 3:** Relate the two mapping above to build a lookup table for the over all mapping. Specifically, for each input level $i$, find an output level $j$ so that $H_z[j]$ best matches $H_x[i]$:

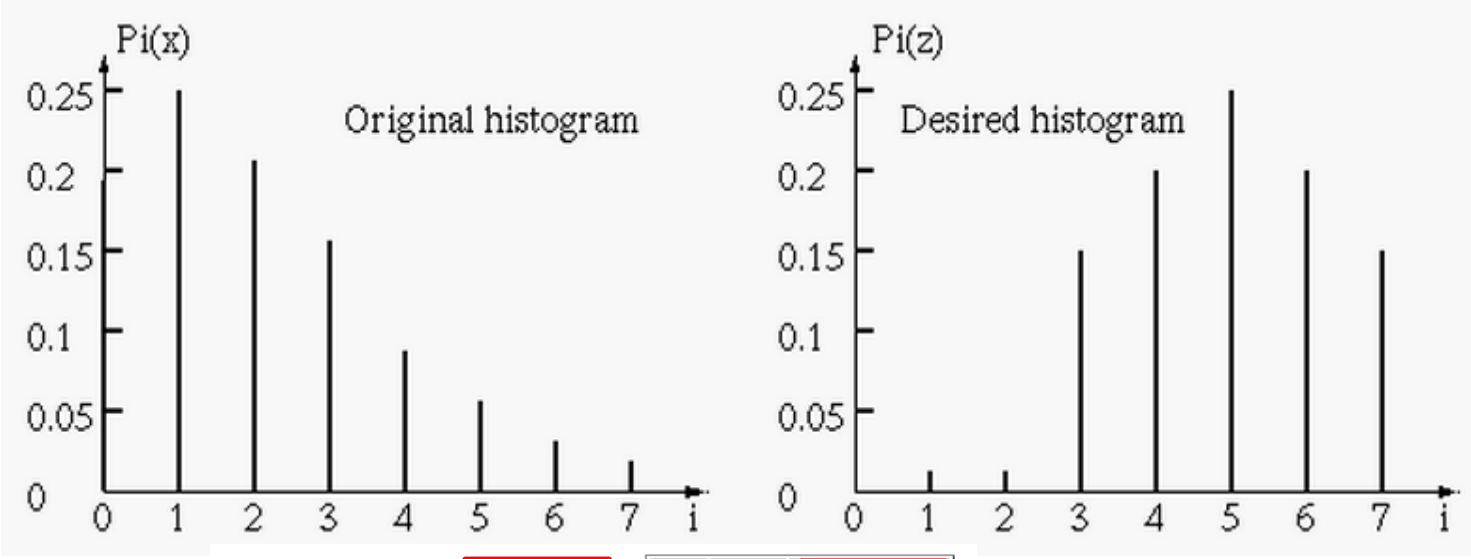
$$|H_x[i] - H_z[j]| = min_k |H_x[i] - H_z[k]|$$

and then we setup a lookup entry $lookup[i] = j$.

# Histogram Processing – (cont.)

## Histogram Matching (Specification) – Discrete

- **Example**




| $x_i$ | $n_j$ | $h_x$ | $y = H_x$ |
|---|---|---|---|
| 0 | 790 | 0.19 | 0.19 |
| 1 | 1023 | 0.25 | 0.44 |
| 2 | 850 | 0.21 | 0.65 |
| 3 | 656 | 0.16 | 0.81 |
| 4 | 329 | 0.08 | 0.89 |
| 5 | 245 | 0.06 | 0.95 |
| 6 | 122 | 0.03 | 0.98 |
| 7 | 81 | 0.02 | 1.00 |

| $z_i$ | $p_z$ | $y' = H_z$ |
|---|---|---|
| 0 | 0.0 | 0.0 |
| 1 | 0.0 | 0.0 |
| 2 | 0.0 | 0.0 |
| 3 | 0.15 | 0.15 |
| 4 | 0.20 | 0.35 |
| 5 | 0.30 | 0.65 |
| 6 | 0.20 | 0.85 |
| 7 | 0.15 | 1.0 |

# Histogram Processing – (cont.)

## Histogram Matching (Specification) – Discrete

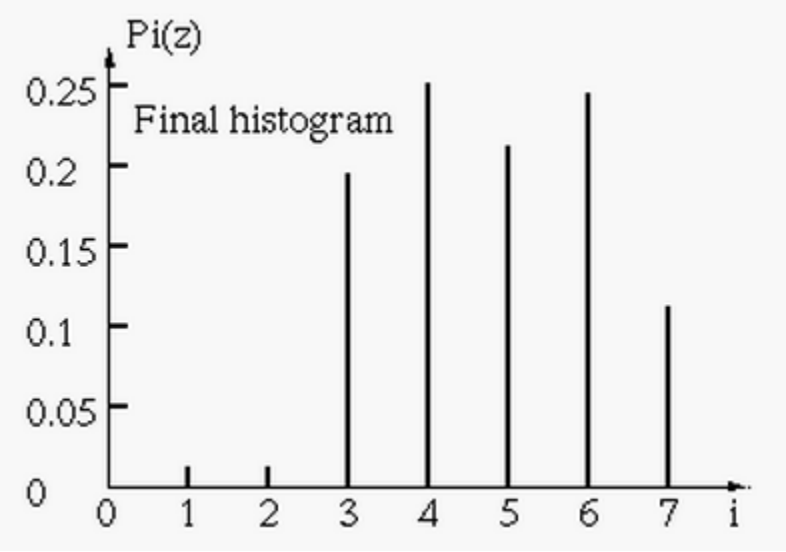
- **Example**

**Finding best level for best fit**

| $x_i = i$ | $y_j = H_x$ | $y'_j = H_z$ | $z_j = j$ |
|---|---|---|---|
| 0 | 0.19 | 0.0 | 3 |
| 1 | 0.44 | 0.0 | 4 |
| 2 | 0.65 | 0.0 | 5 |
| 3 | 0.81 | 0.15 | 6 |
| 4 | 0.89 | 0.35 | 6 |
| 5 | 0.95 | 0.65 | 7 |
| 6 | 0.98 | 0.85 | 7 |
| 7 | 1.0 | 1.0 | 7 |

**Resulting lookup table**

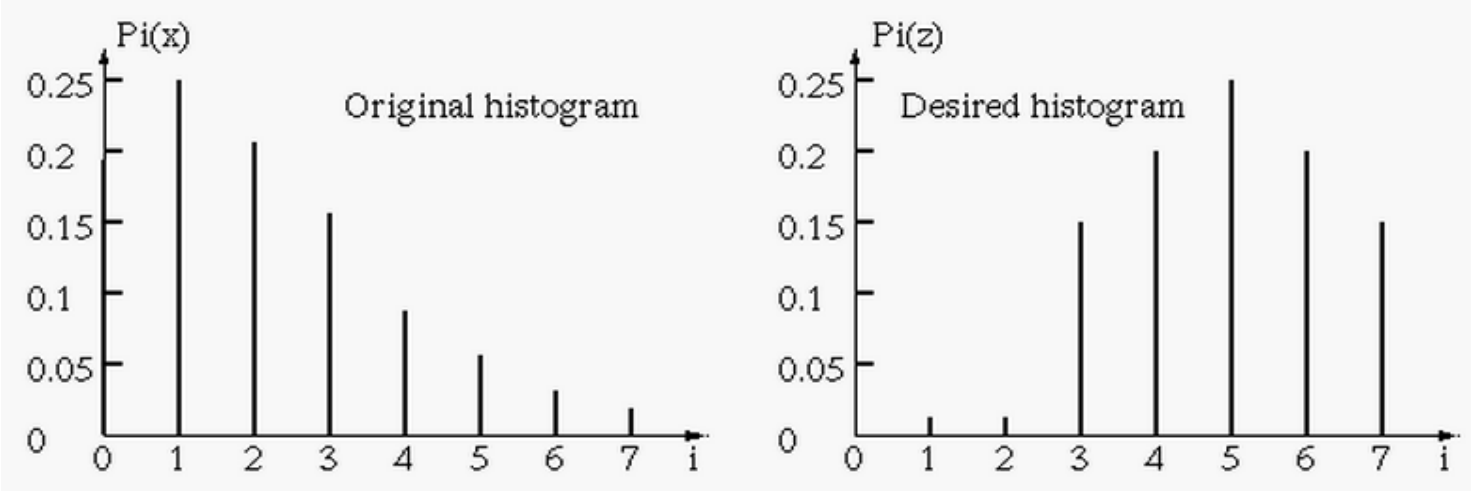
| i | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| j | 3 | 4 | 5 | 6 | 6 | 7 | 7 | 7 |

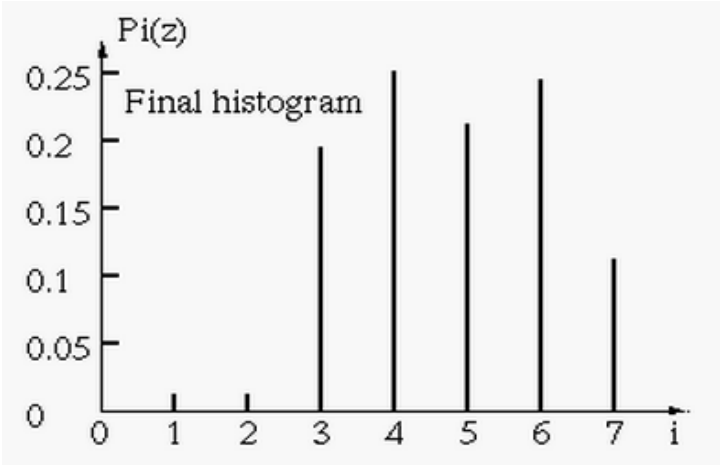


# Histogram Processing – (cont.)

## Histogram Matching (Specification) – Discrete

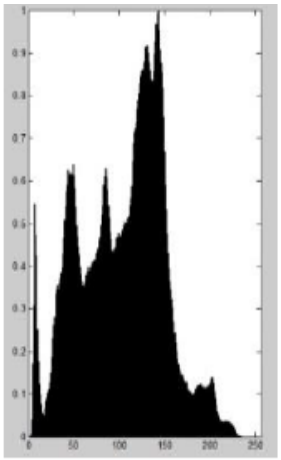
- **Example**



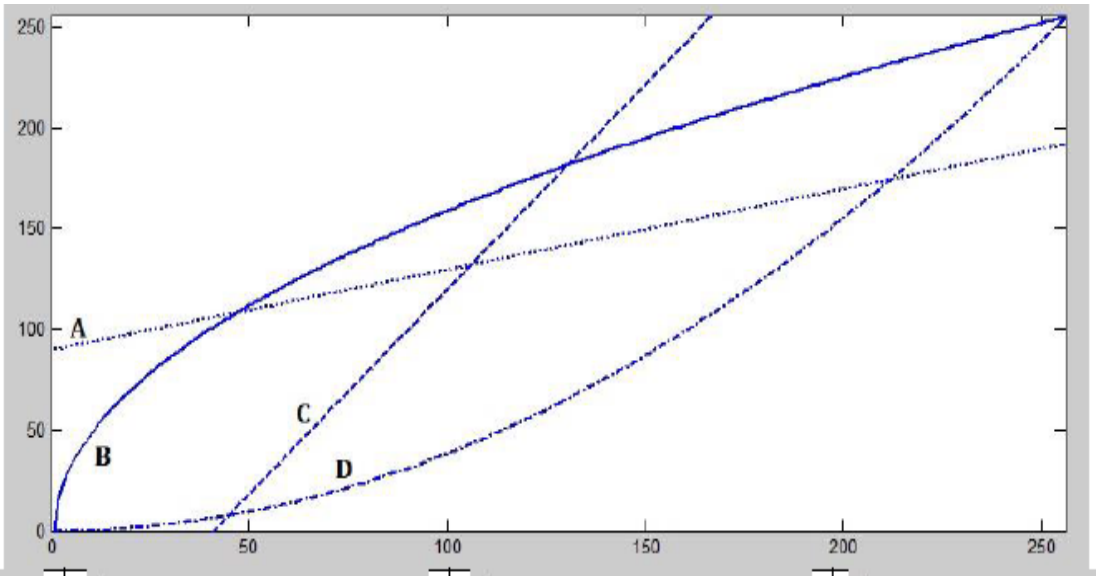



# Selected Problems

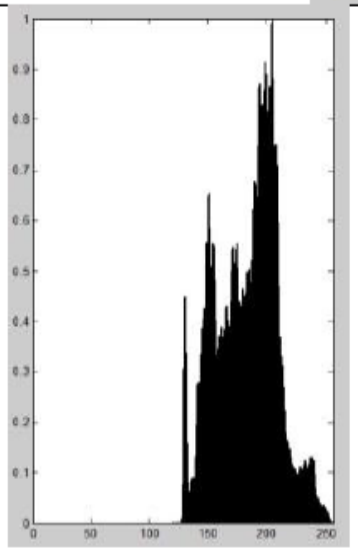
# Selected Problems



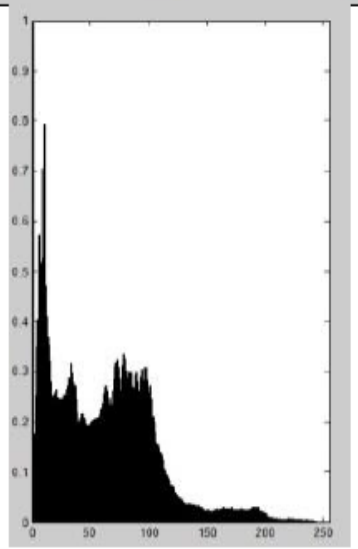
Original histogram



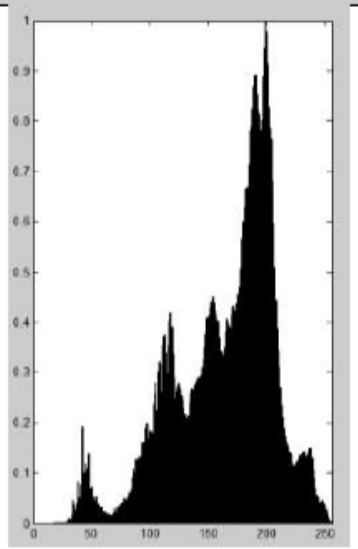

| Answers | | | |
|---|---|---|---|
| A | B | C | D |
| 1 | 3 | 4 | 2 |



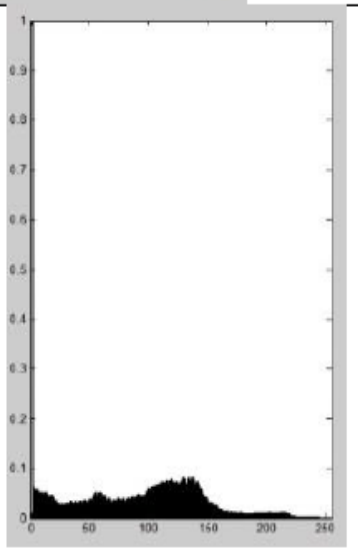
1



2



3



4

**2.21** Consider two 8-bit images whose intensity levels span the full range from 0 to 255.

**(a)** Discuss the limiting effect of repeatedly subtracting image (2) from image (1). Assume that the result is represented also in eight bits.

**(b)** Would reversing the order of the images yield a different result?

**2.22** Image subtraction is used often in industrial applications for detecting missing components in product assembly. The approach is to store a "golden" image that corresponds to a correct assembly; this image is then subtracted from incoming images of the same product. Ideally, the differences would be zero if the new products are assembled correctly. Difference images for products with missing components would be nonzero in the area where they differ from the golden image. What conditions do you think have to be met in practice for this method to work?

**3.5** **(a)** What effect would setting to zero the half of lower-order bit planes have on the histogram of an image in general?

**(b)** What would be the effect on the histogram if we set to zero the half of higher-order bit planes instead?

**3.8** In some applications it is useful to model the histogram of input images as Gaussian probability density functions of the form

$$p_r(r) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(r-m)^2}{2\sigma^2}}$$

where $m$ and $\sigma$ are the mean and standard deviation of the Gaussian PDF. The approach is to let $m$ and $\sigma$ be measures of average intensity and contrast of a given image. What is the transformation function you would use for histogram equalization?

# Notes on Image Enhancement

- **For human perception**
  - **The visual evaluation of image quality is a highly subjective process.**
  - **It is hard to standardize the definition of a good image.**
- **For machine perception**
  - **The evaluation task is easier.**
  - **A good image is one which gives the best machine recognition results.**
- **A certain amount of trial and error usually is required before a particular approach is selected.**

# Next Lecture

**Continuation of Image Enhancement in the Spatial Domain: Convolution and Spatial Filtering.**

# Assignment

**- Textbook Chapter 3:** 2

# Next Lecture

**Continuation of Image Enhancement in the Spatial Domain: Histogram Processing + Convolution and Spatial Filtering.**

# Assignment

**- Textbook Chapter 3:** 1, 2.

**- Check associated problems**

| | |
|---|---|
| **Chapter 3** | **4, 5, 6, 7, 8, 13** |

# References

- Gonzalez and Woods, Digital Image Processing.

- Peters, Richard Alan, II, “Point Processing", Lectures on Image Processing, Vanderbilt University, Nashville, TN, April 2008, Available on the web at the Internet Archive, http://www.archive.org/details/Lectures_on_Image_Processing.

- Histogram Equalization, Application Note by Robert Krutsch and David Tenorio, Microcontroller Solutions Group.

- GIMP documentation http://docs.gimp.org/en/

-http://dmmd.net/main_wp/intuitive-mathematics/histogram-equalization/

-http://homepages.inf.ed.ac.uk/rbf/HIPR2/histeq.htm

-http://fourier.eng.hmc.edu/e161/lectures/contrast_transform/node3.html