Image Processing

Lecture 02: Intensity Transformation

(Ch3 Intensity Transformation and Spatial Filtering – I)

Zhiguo Zhang

zhiguozhang@hit.edu.cn

Review of Last Lecture

- In the last lecture we learnt:
  - What is digital image processing?
  - Human vision system
  - Image acquisition
  - Sampling and Quantization
  - Resolution
  - Basic Relationships between Pixels
  - Key Stages in Image Processing

Contents of This Lecture

- In this lecture we will learn:
  - Various kinds of basic intensity transformation functions (point processing)
    - Thresholding
    - Logarithmic transformation
    - Power law transforms
    - Gray level slicing
    - Bit plane slicing
    - Image subtraction
    - Image averaging
  - Histogram processing (equalization)

What Is Image Enhancement?

- Image enhancement is the process of making images more useful.
- The reasons for doing this include:
  - Highlighting interesting detail in images
  - Removing noise from images
  - Making images more visually appealing

Image Enhancement Examples

Image Enhancement Examples (cont…)

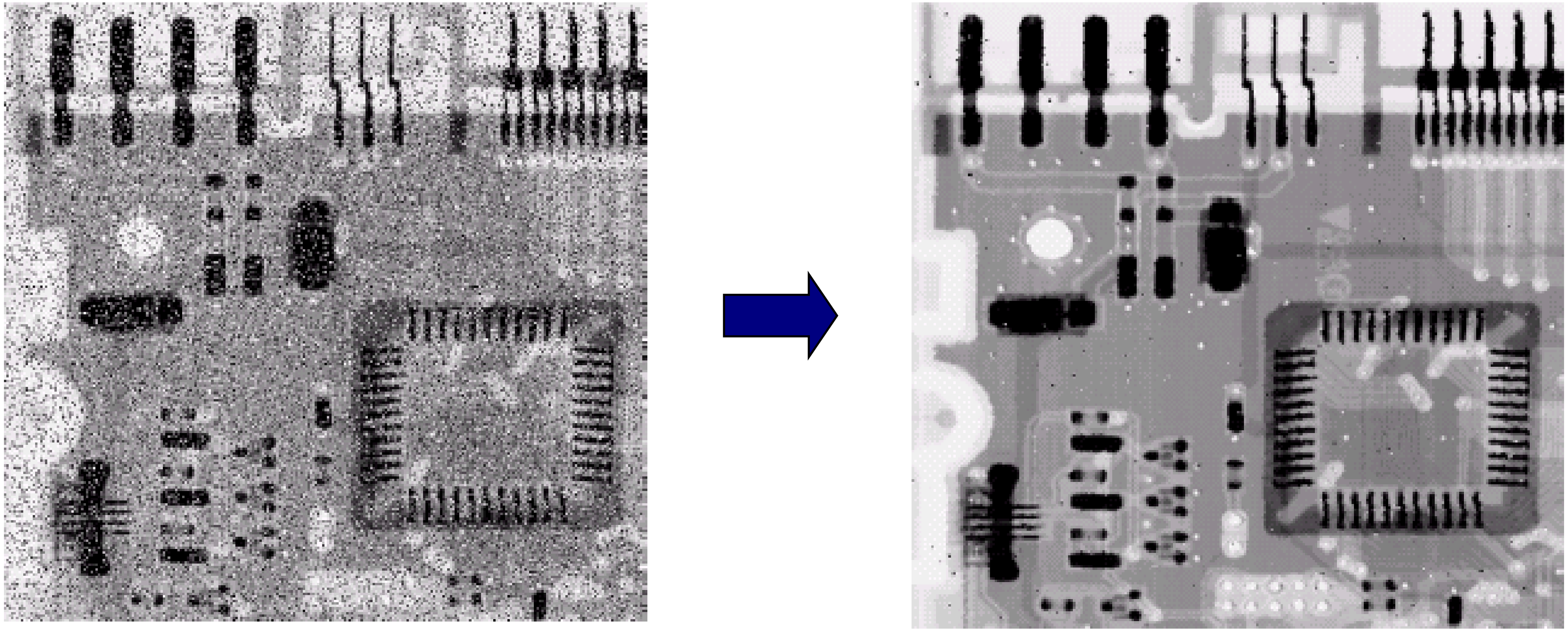
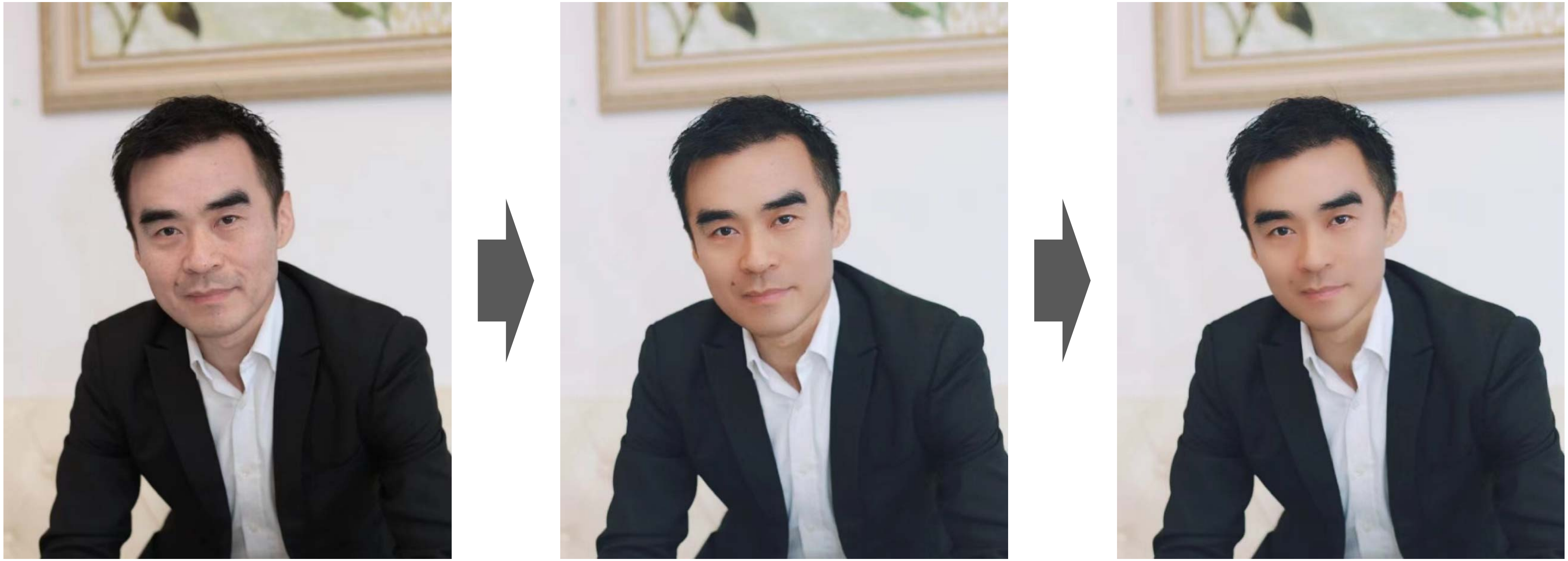


Image Enhancement Examples (cont…)



A Note About Gray Levels

- So far when we have spoken about image gray level values, we have said they are in the range [0, 255], where 0 is black and 255 is white.
- For many of the image processing operations in this lecture, gray levels are assumed to be given in the range [0, 1].

Spatial & Frequency Domains

- There are two broad categories of image enhancement techniques
  - Spatial domain techniques
    - Direct manipulation of image pixels
      - Point processing
      - Neighbourhood operations
  - Frequency domain techniques
    - Manipulation of Fourier transform or wavelet transform of an image
- For the moment we will concentrate on techniques that operate in the spatial domain

Basic Spatial Domain Image Enhancement

- Most spatial domain enhancement operations can be reduced to the form

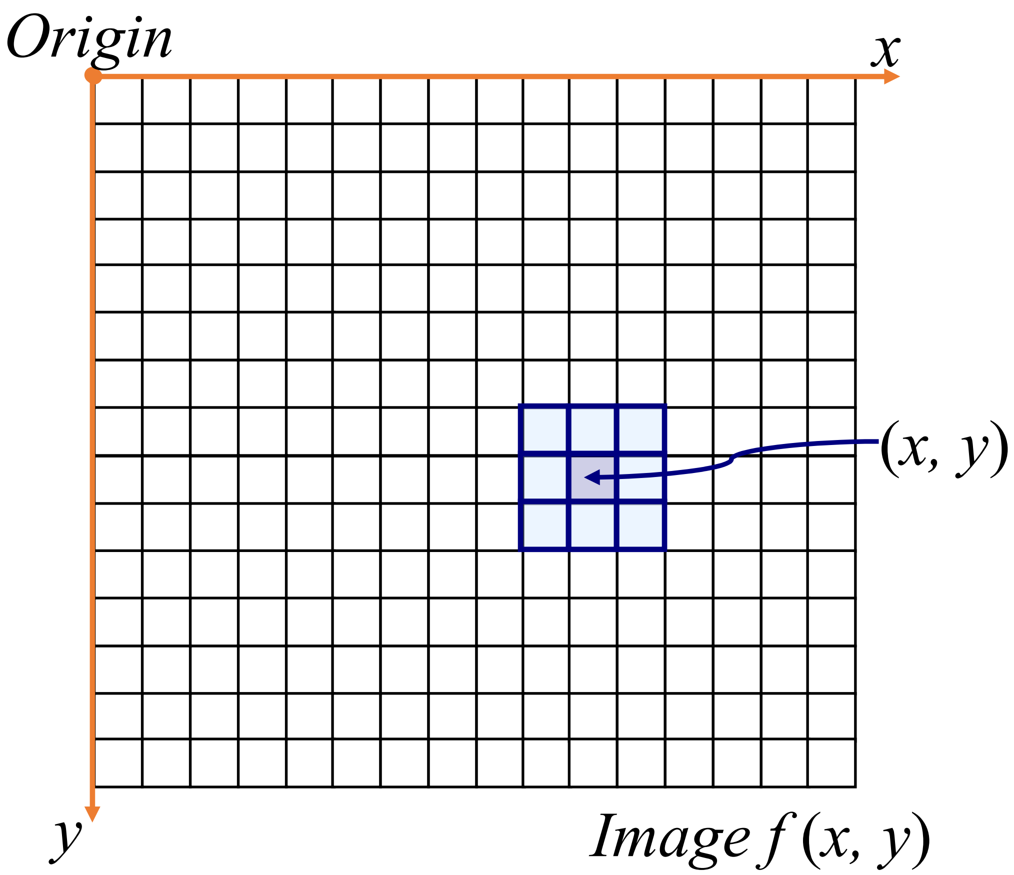
$$g(x, y) = T[f(x, y)]$$

where $f(x, y)$ is the input image,

$g(x, y)$ is the processed image and

T is some operator defined over

some neighbourhood of (x, y).

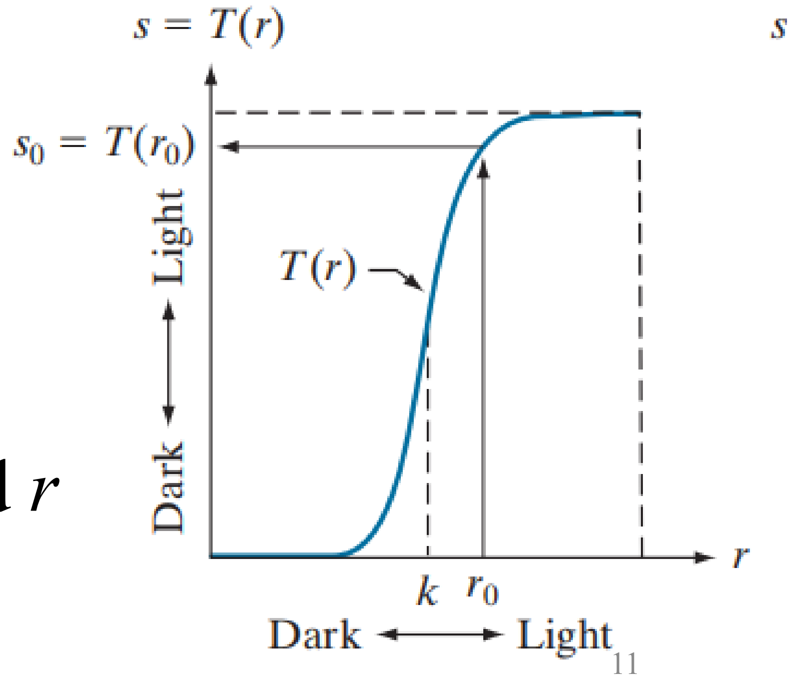


Basic Spatial Domain Image Enhancement

- The simplest spatial domain operations occur when the neighbourhood is simply the pixel itself: **Point Processing**
- In this case T is referred to as a gray level transformation function or a point processing operation.
- Point processing operations take the form:

$$s = T(r)$$

where s refers to the processed image pixel value and r refers to the original image pixel value.

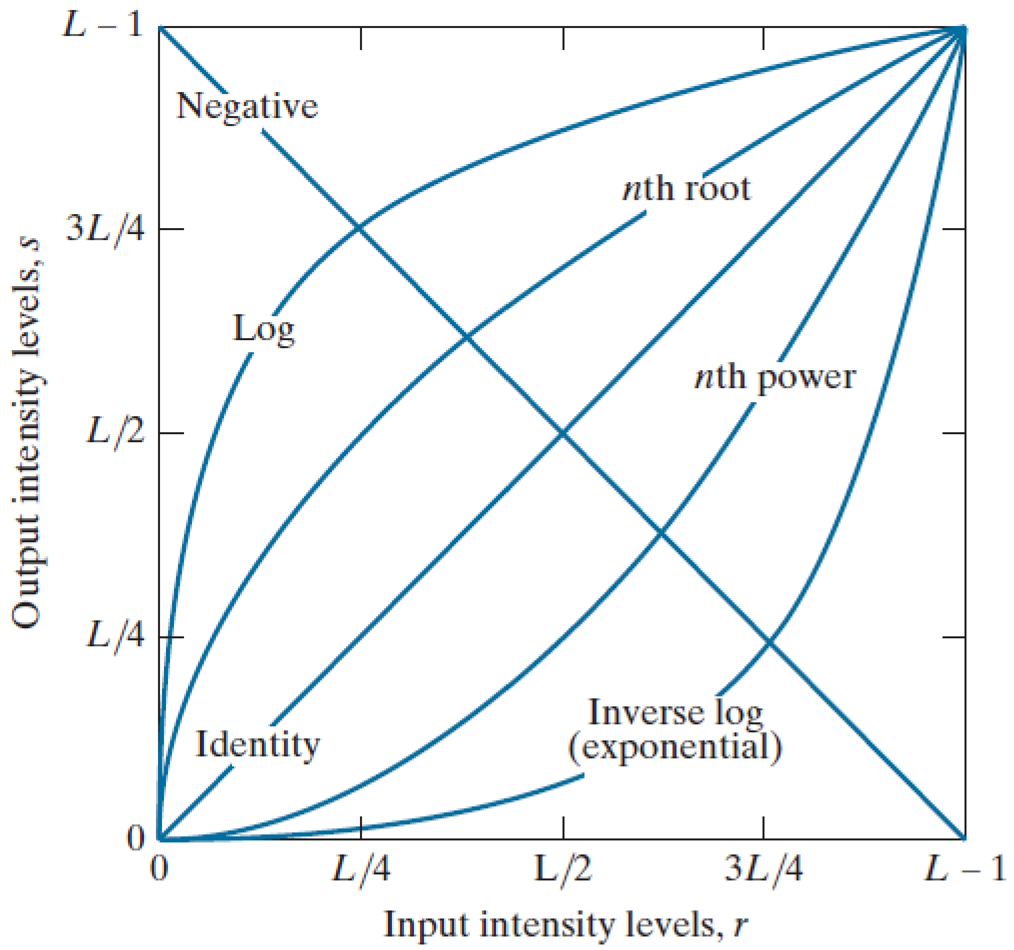


Basic Spatial Domain Image Enhancement

- Point processing
  - The neighborhood is of size 1×1
  - Gray-level transformation
- Mask processing or spatial filtering
  - The neighborhood is defined as a mask, a filter, or a window.
  - Filtering

Point Processing

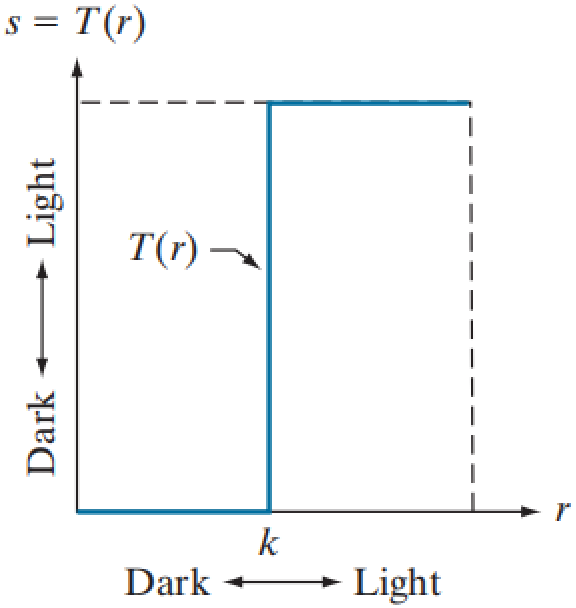
- In this lecture we will look at image enhancement point processing techniques:
  - Thresholding
  - Logarithmic transformation
  - Power law transforms
  - Gray level slicing
  - Bit plane slicing
  - Image subtraction
  - Image averaging



Thresholding

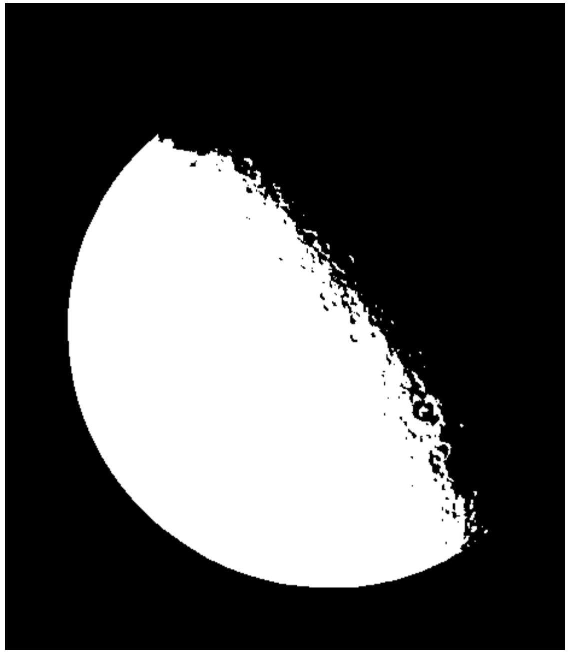
- Thresholding transformations are particularly useful for segmentation in which we want to isolate an object of interest from a background.

$$s = \begin{cases} 1.0 & r > threshold \\ 0.0 & r \leq threshold \end{cases}$$



Thresholding

$$s = \begin{cases} 1.0 & r > threshold \\ 0.0 & r \leq threshold \end{cases}$$



Thresholding

$$s = \begin{cases} 1.0 & r > threshold \\ 0.0 & r \le threshold \end{cases}$$

Logarithmic Transformation

- The general form of the log transformation is

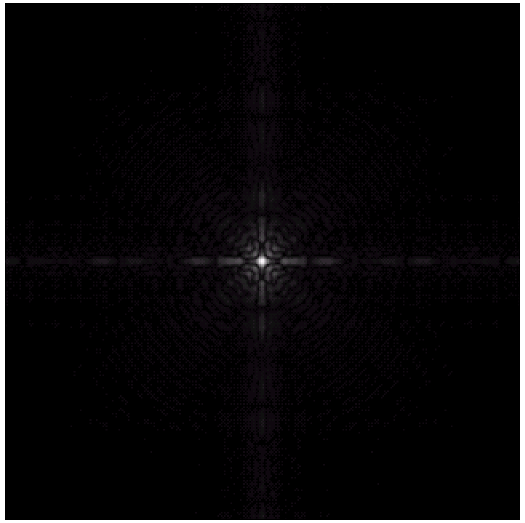
$$s = c \log(1 + r)$$

- Log functions are particularly useful when the input gray level values may have an extremely large range of values.

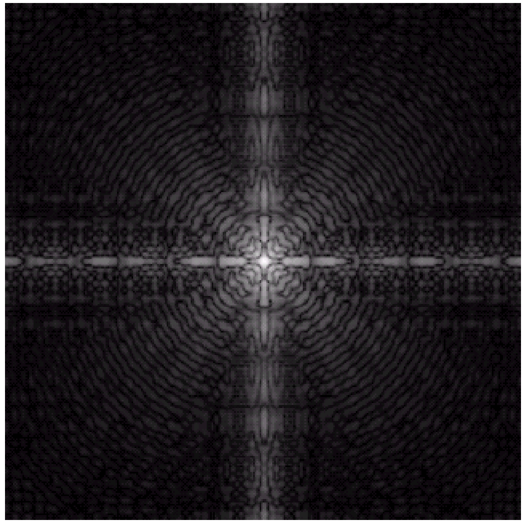
- In Fourier transform, we usually encounter spectrum values that range from 0 to 10^6 or higher. But an image display system cannot reproduce such a wide range of intensity values.

Logarithmic Transformation (cont…)

- In the following example the Fourier transform of an image is put through a log transform to reveal more detail.



$s = \log(1 + r)$



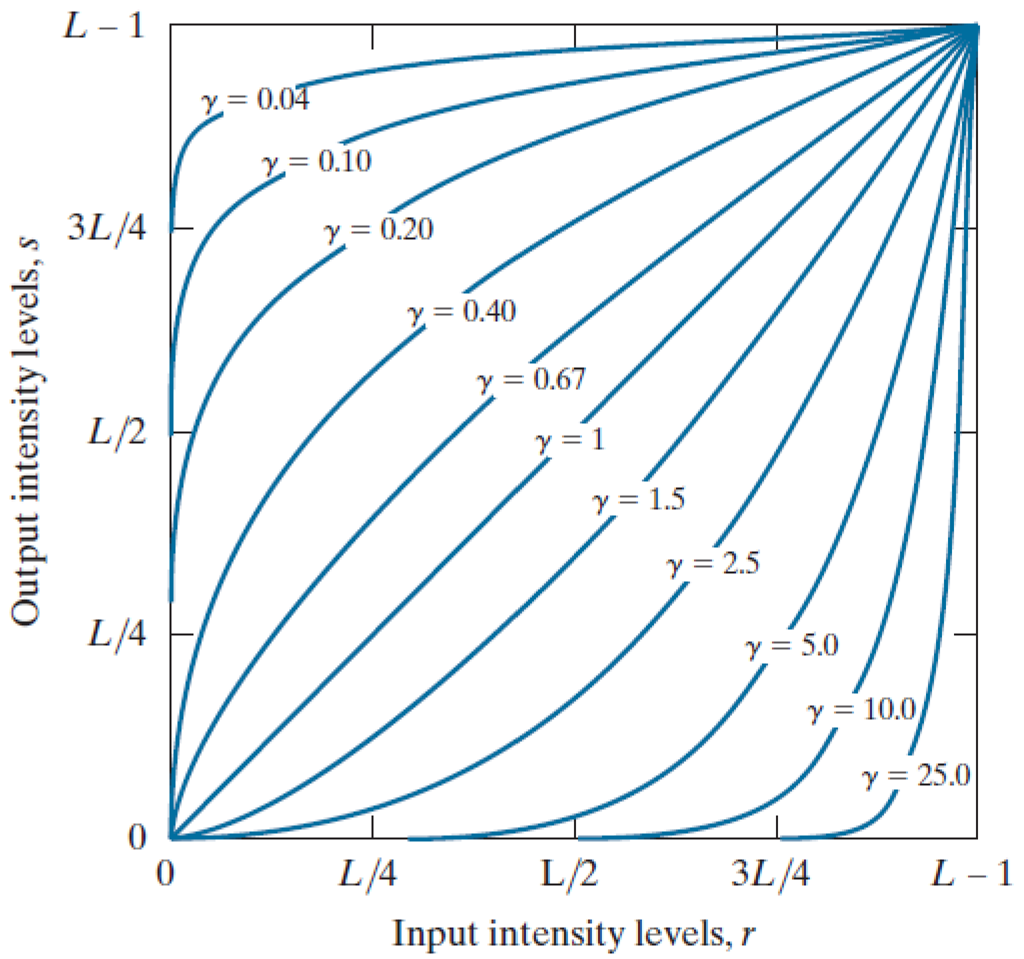
Power Law Transformations

- Power law transformations have the following form:

$$s = cr^{\gamma} \quad r \in [0.0, 1.0]$$

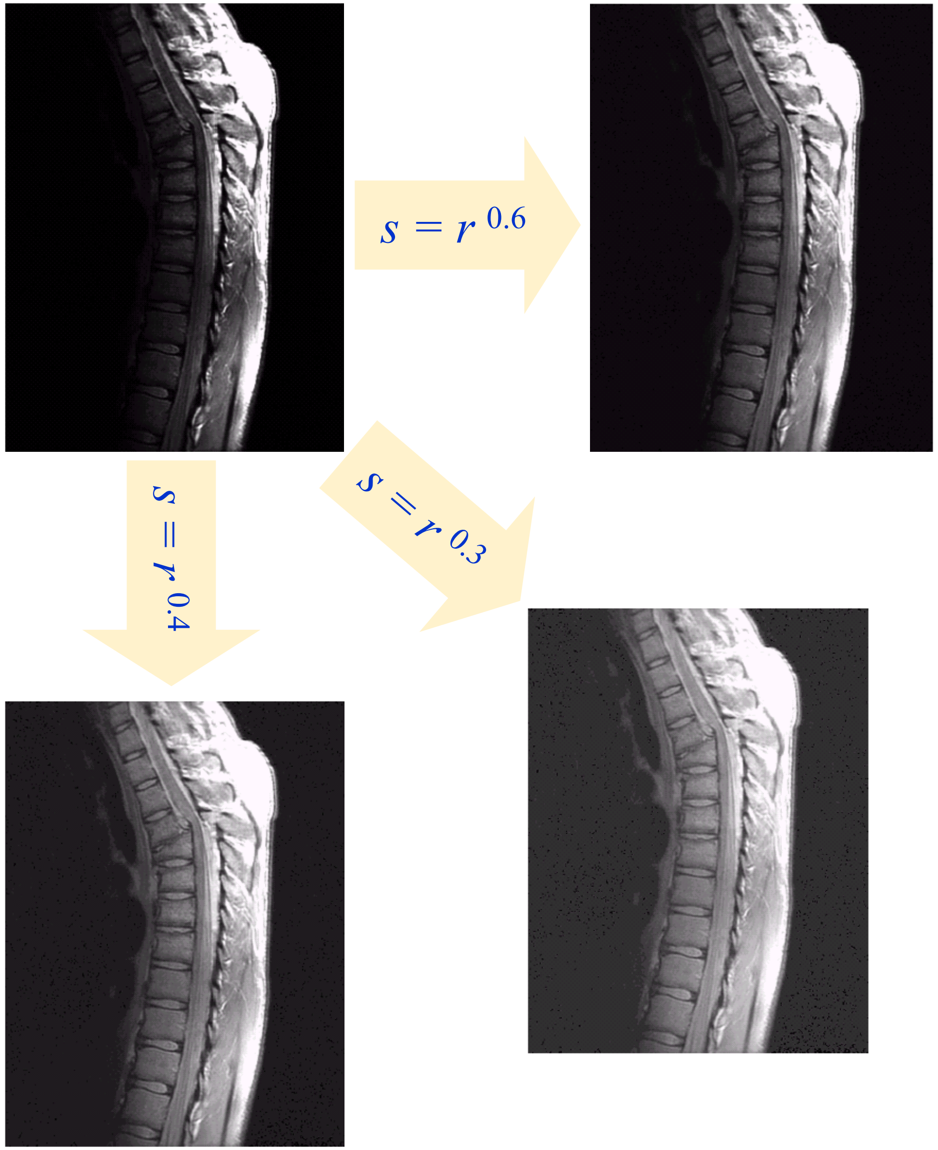
- Map a narrow range of dark input values into a wider range of output values or vice versa.

- Varying γ gives a whole family of curves.



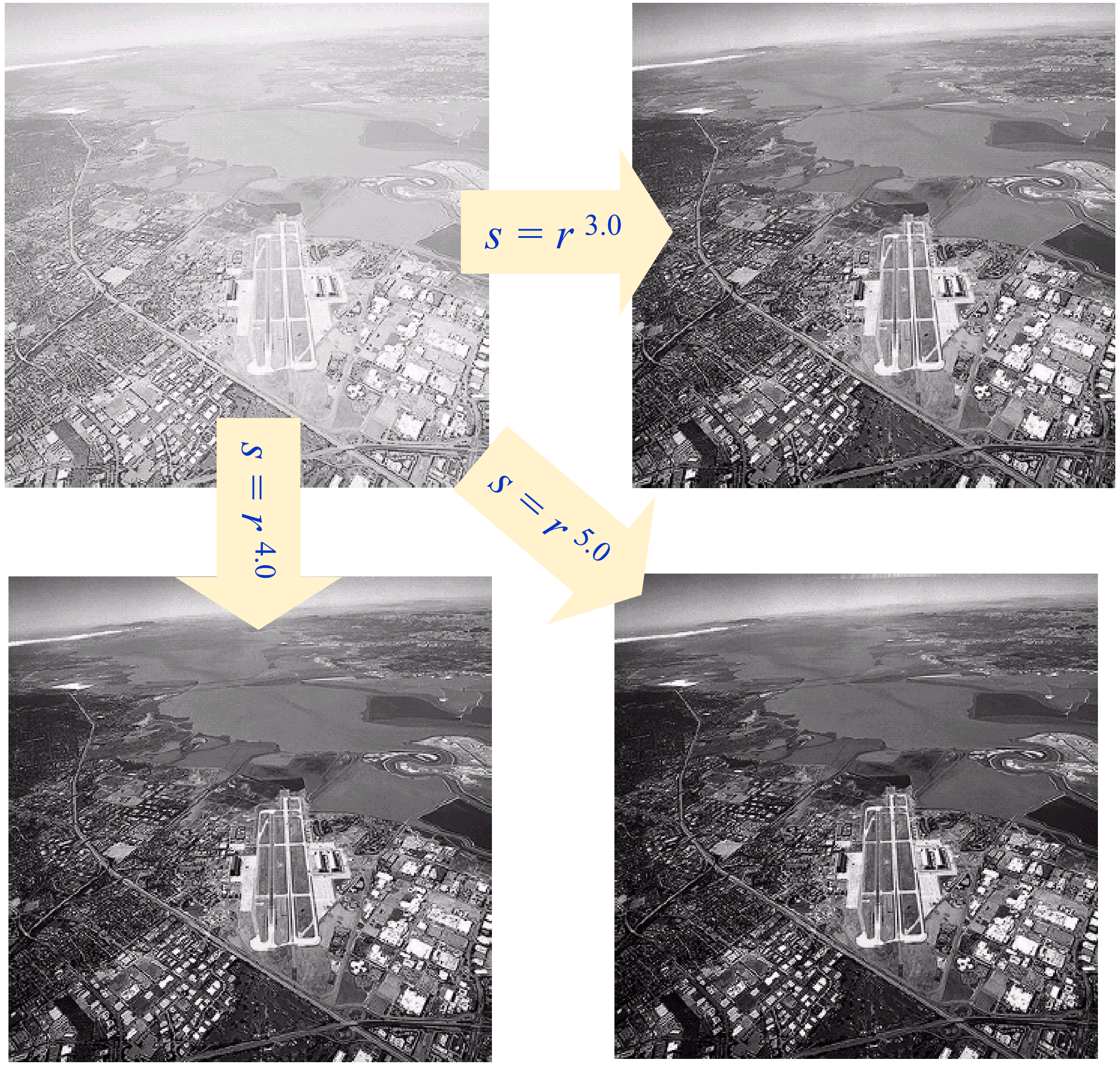
Power Law Example (cont…)

- The images to the right show a magnetic resonance image of a fractured human spine.
- Different curves highlight different detail.



Power Law Transformations (cont…)

- An aerial photo of a runway is shown.
- This time power law transforms are used to darken the image.



Gamma Correction

- Many of you might be familiar with gamma correction of computer monitors.
- Problem is that display devices and print devices do not respond linearly to different intensities, and they respond according to a power law:

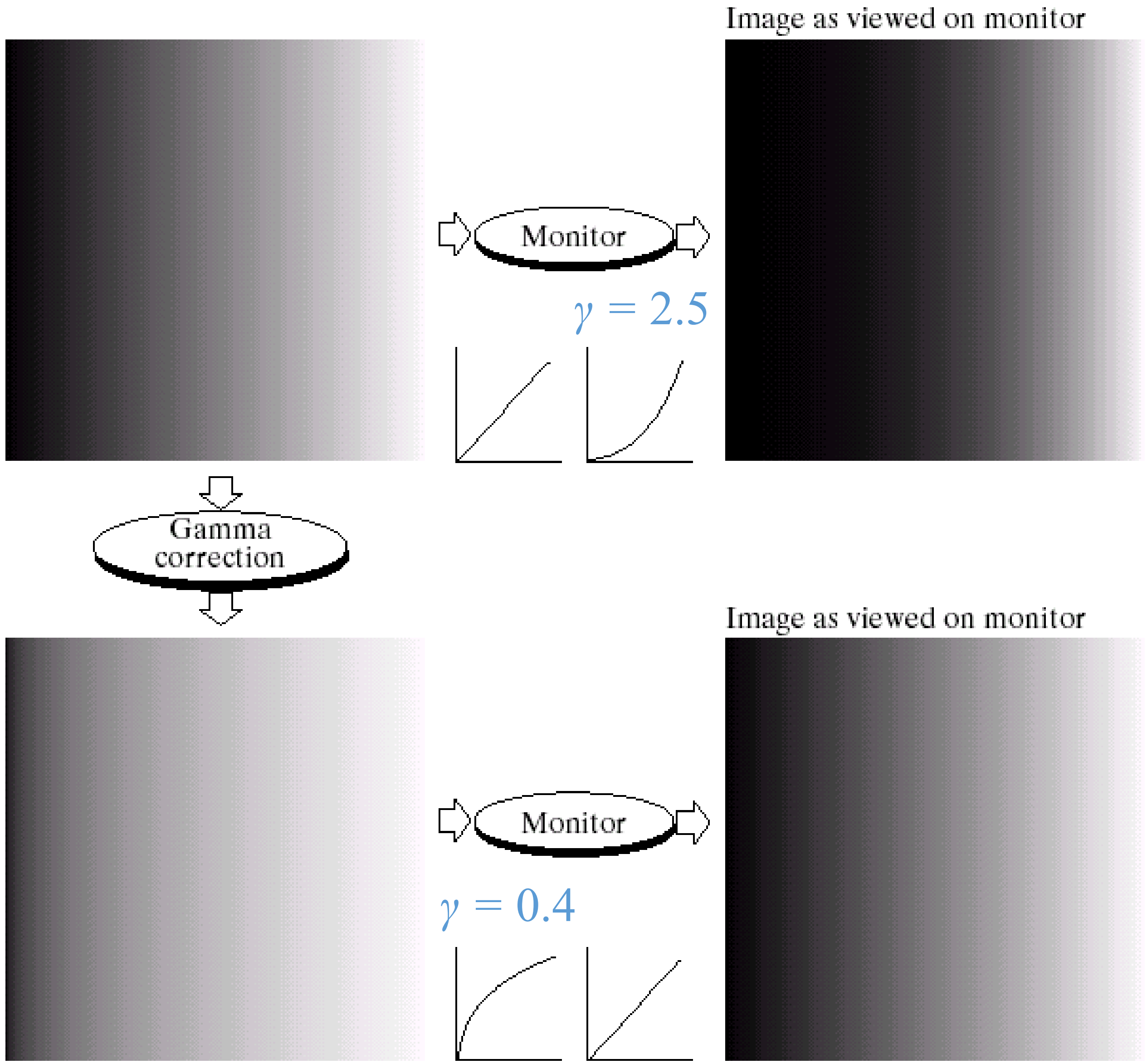
$$s = r^{\gamma}$$

- For cathode ray tube (CRT) display, γ =1.8~2.5.
- This can be corrected using a n-th root transform:

$$s = r^{1/\gamma}$$

Gamma Correction

(a) Intensity ramp image.



(b) Image as viewed on a simulated monitor with a gamma of 2.5.

(c) Gamma-corrected image.

(d) Corrected image as viewed on the same monitor.

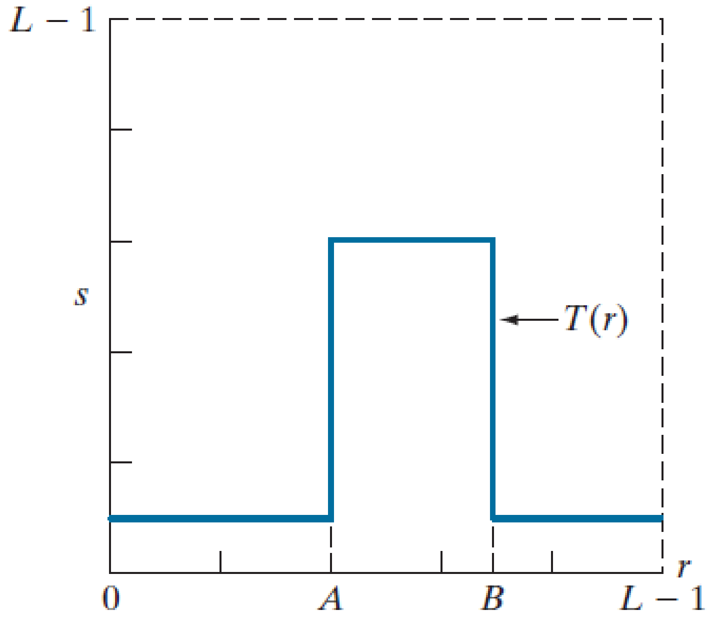
Compare (d) and (a).

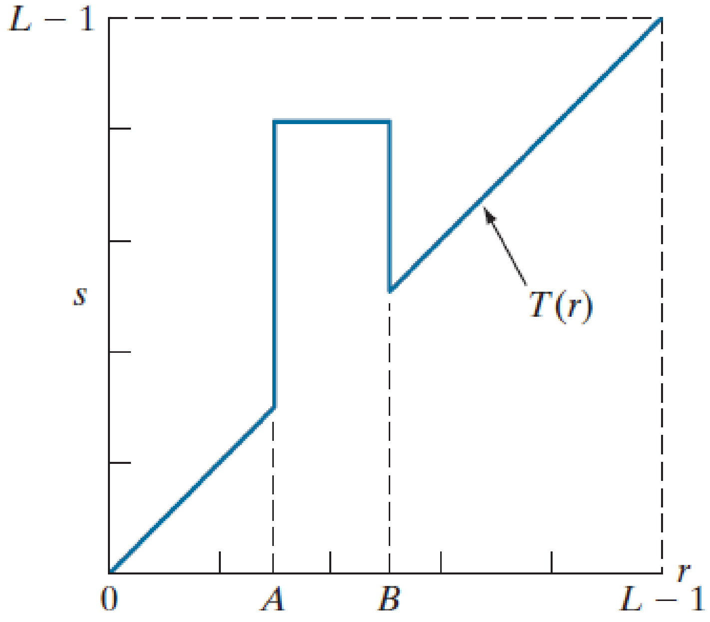
Gamma Correction

- Prior knowledge about gamma correction
  - Varying the value of gamma correction changes not only the brightness, but also the ratio of red to green to blue.
- Applications
  - Internet
  - Millions of people and millions of monitors
  - Gamma represents an “average ”of the types of monitors and computer systems
  - Scanners and printers have different values of gamma

Gray Level Slicing

- To highlight a specific range of gray levels
  - Similar to thresholding
  - Other levels can be suppressed or maintained
  - Useful for highlighting features in an image

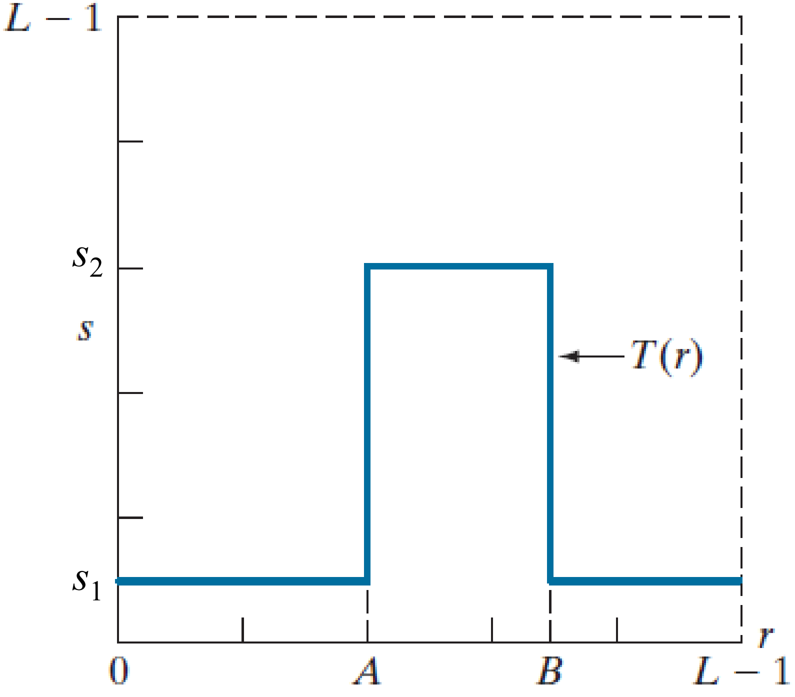


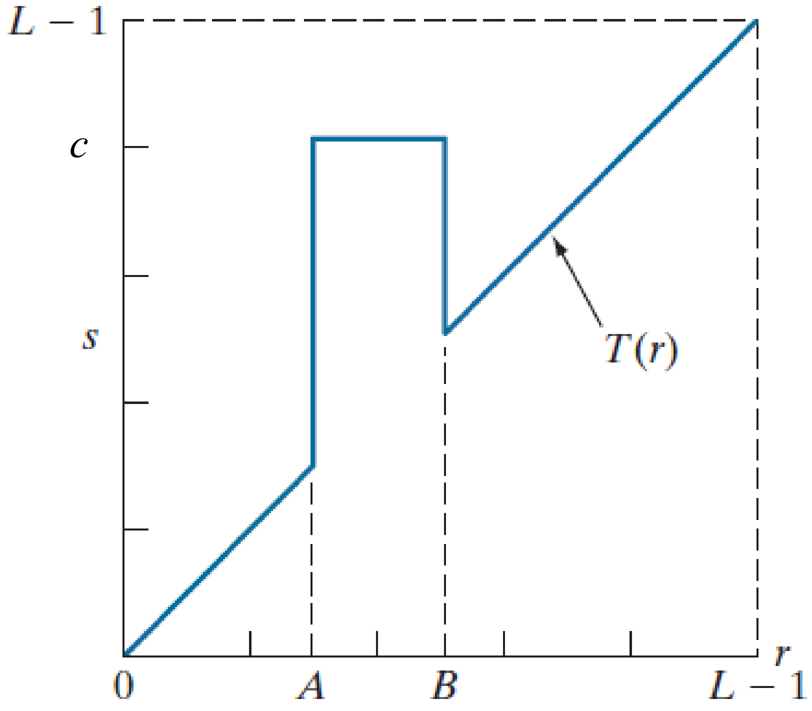


Gray Level Slicing

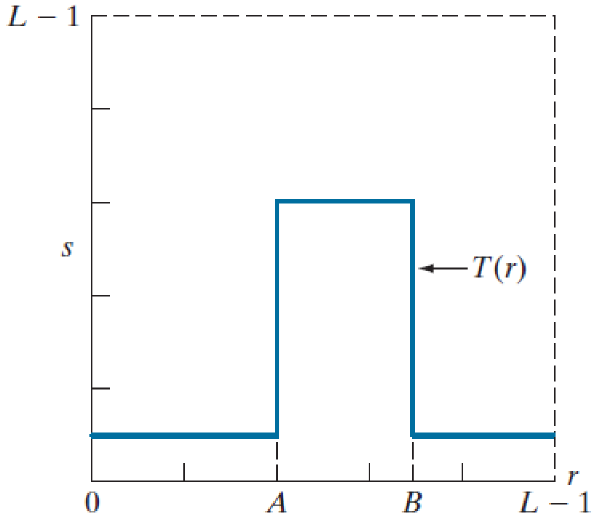
$$s = \begin{cases} s1 & r < A, r > B \\ s2 & A \le r \le B \end{cases}$$

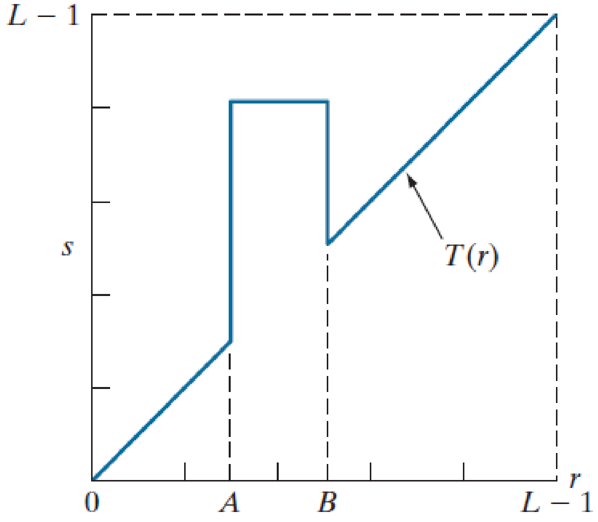
$$s = \begin{cases} r & r < A, r > B \\ c & A \le r \le B \end{cases}$$

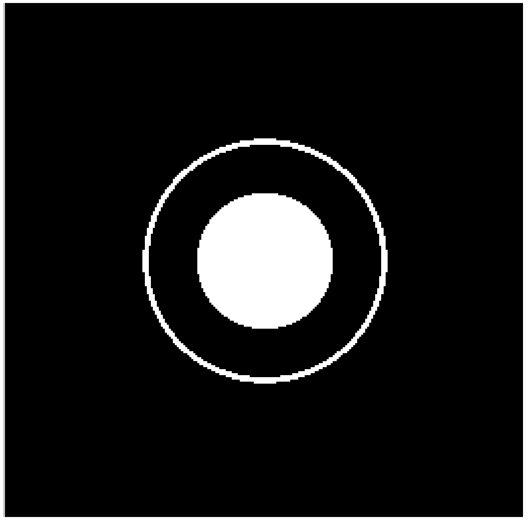


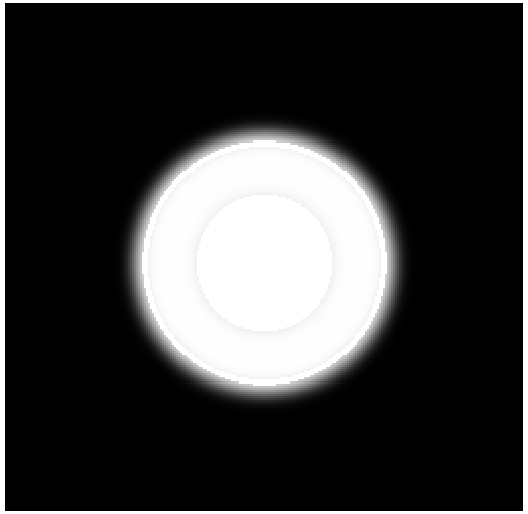


Gray Level Slicing

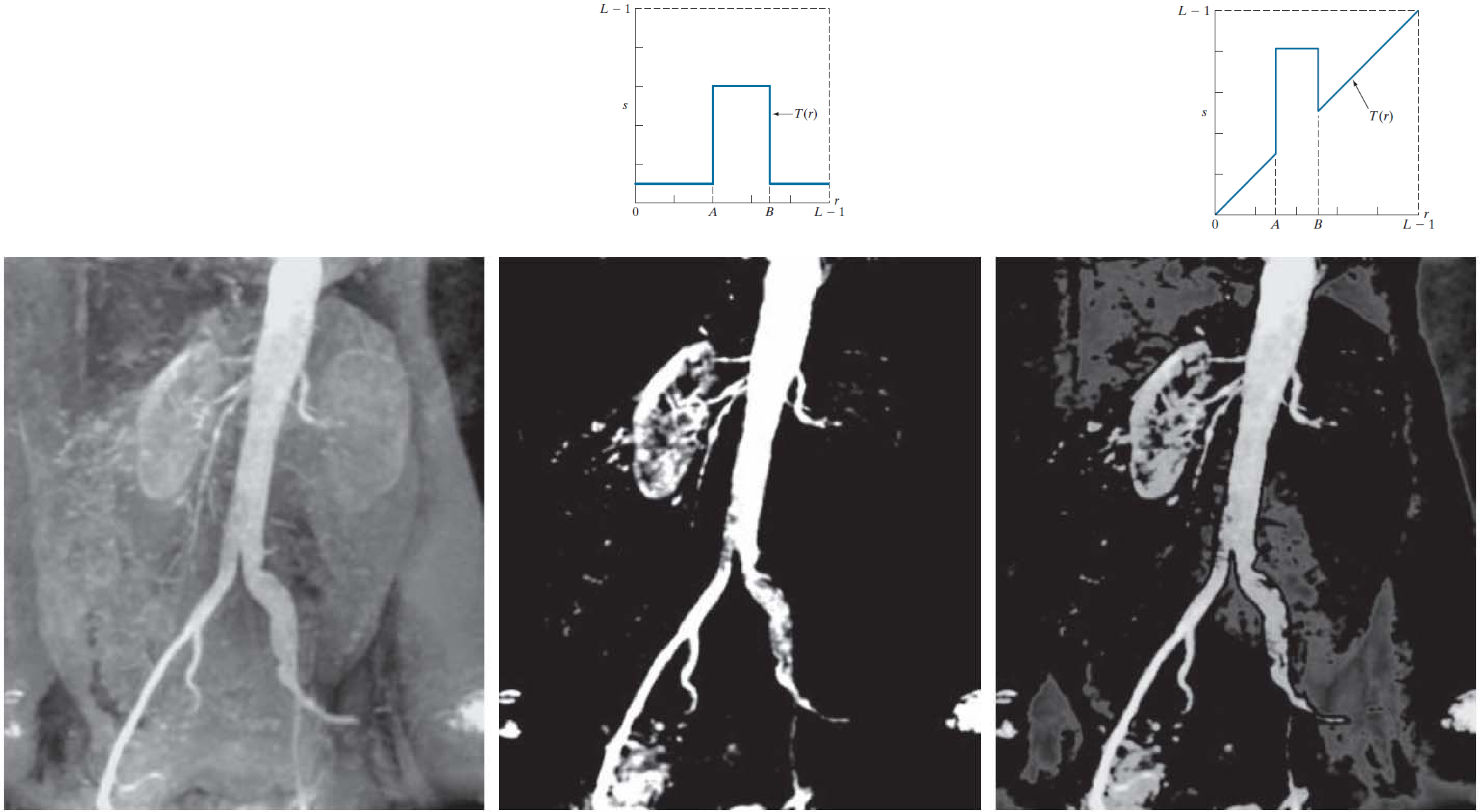






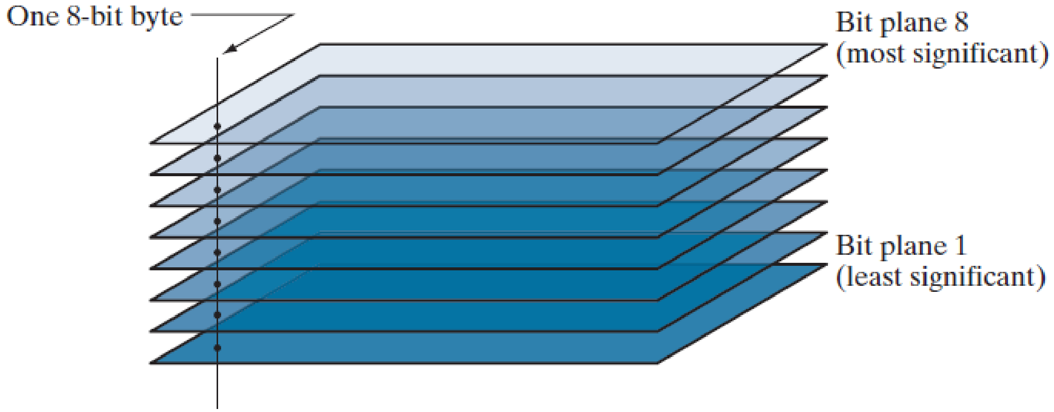


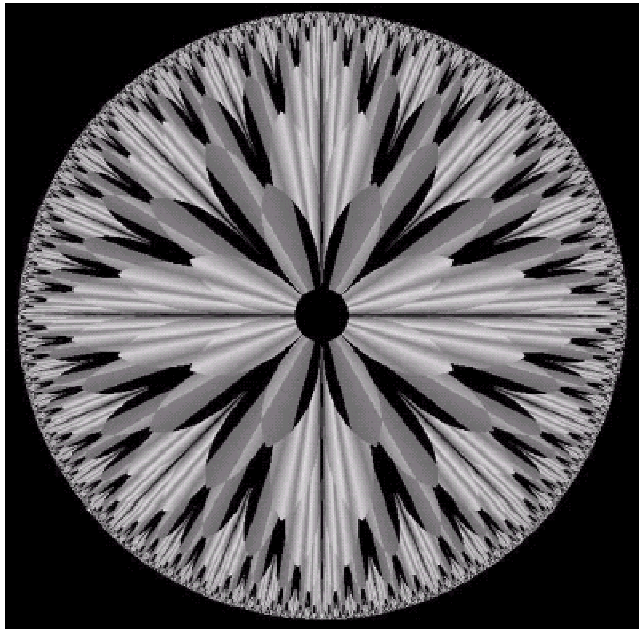
Gray Level Slicing



Bit Plane Slicing

- We can often highlight interesting aspects of that image by isolating particular bits of the pixel values in an image :
  - Higher-order bits usually contain most of the significant visual information.
  - Lower-order bits contain subtle details.





Bit Plane Slicing (cont…)

[10000000]

[01000000]

[00100000]

[00001000]

[00000100]

[00000001]

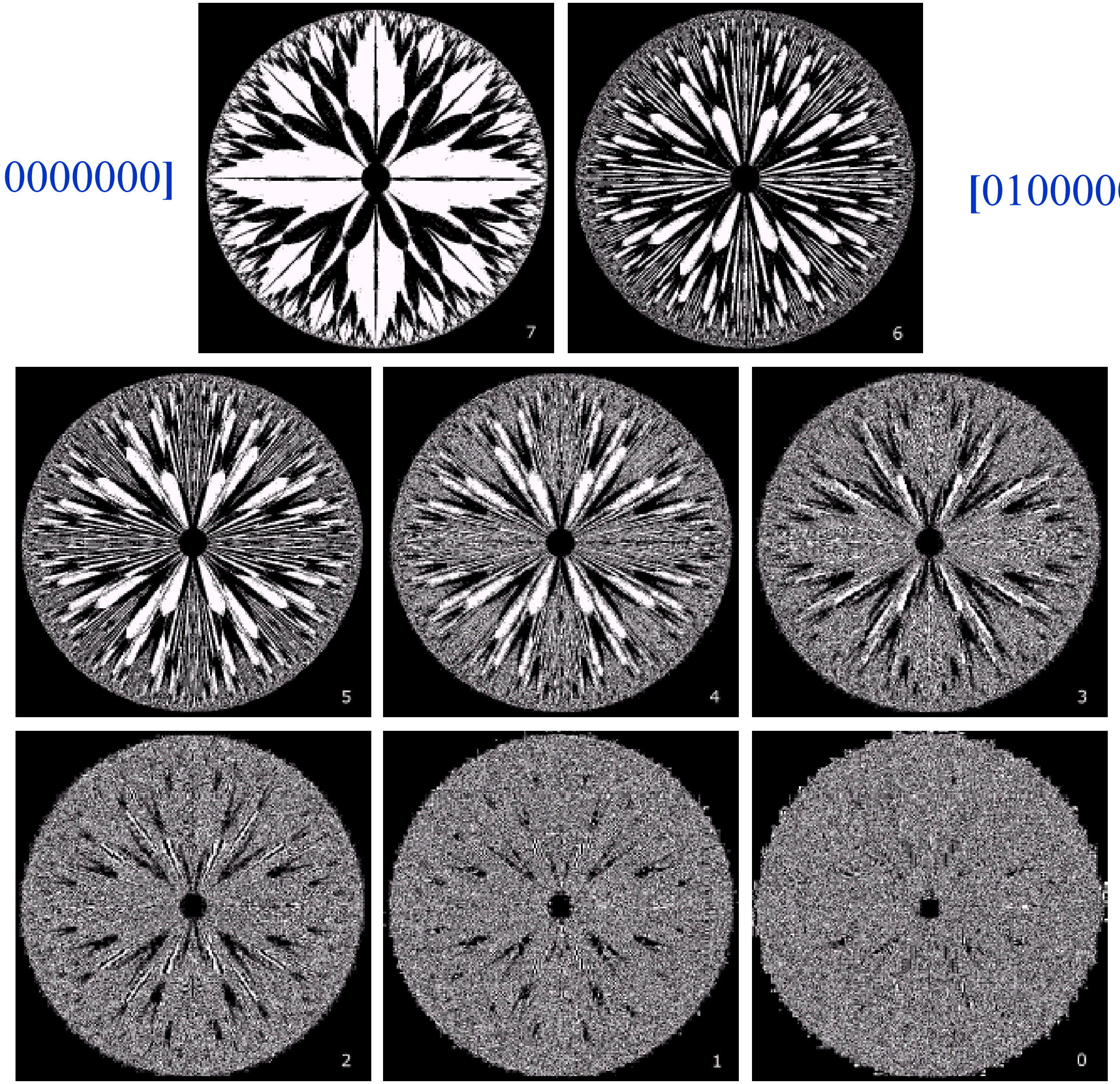


Image Subtraction

- Suppose *f* and *h* are two images, their difference can be calculated as:

$$g(x, y) = f(x, y) - h(x, y)$$

- Note: most images are displayed using 8 bits. Thus we expect image values not to be outside the range from 0 to 255.
- In subtraction, the results should be in the range –255 to 255. So some sort of scaling is required to display the results.

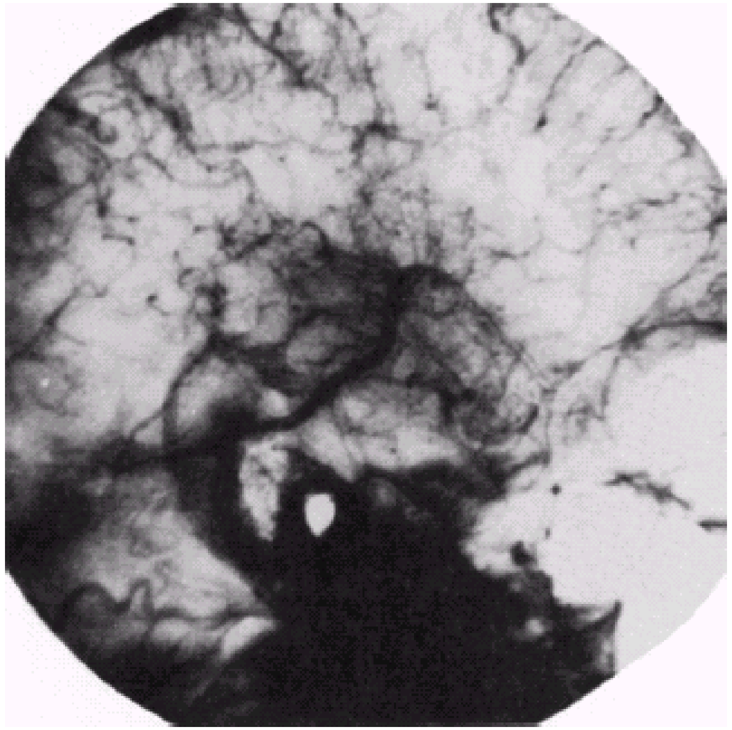
  a) add 255 to $g(x, y)$ and then divide by 2

  b) $y = x - min(x)$; $z = y \cdot 255/max(y)$.

Image Subtraction

Mask mode radiography: enhancement by image subtraction.

Mask image



An image (taken after injection of a contrast medium into the blood stream) with mask subtracted out

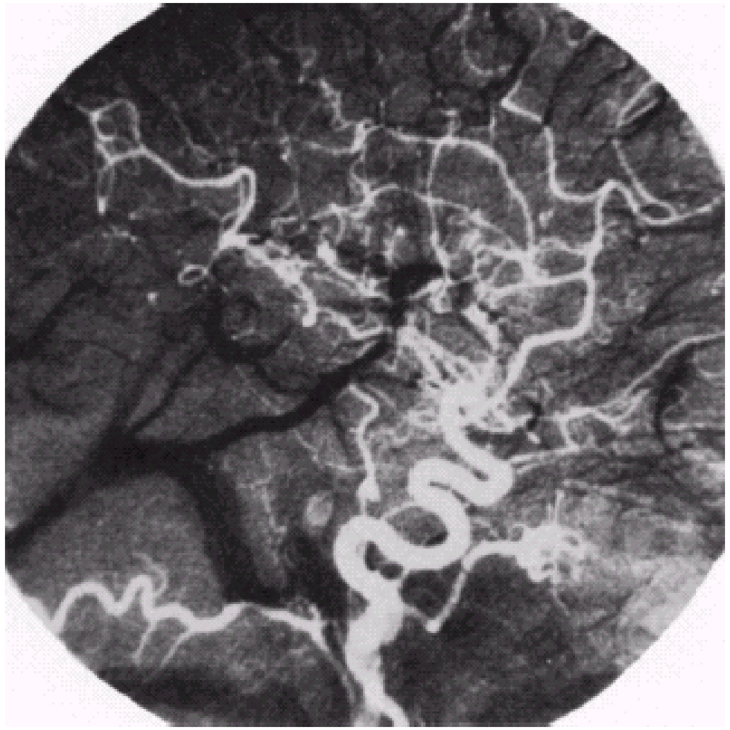
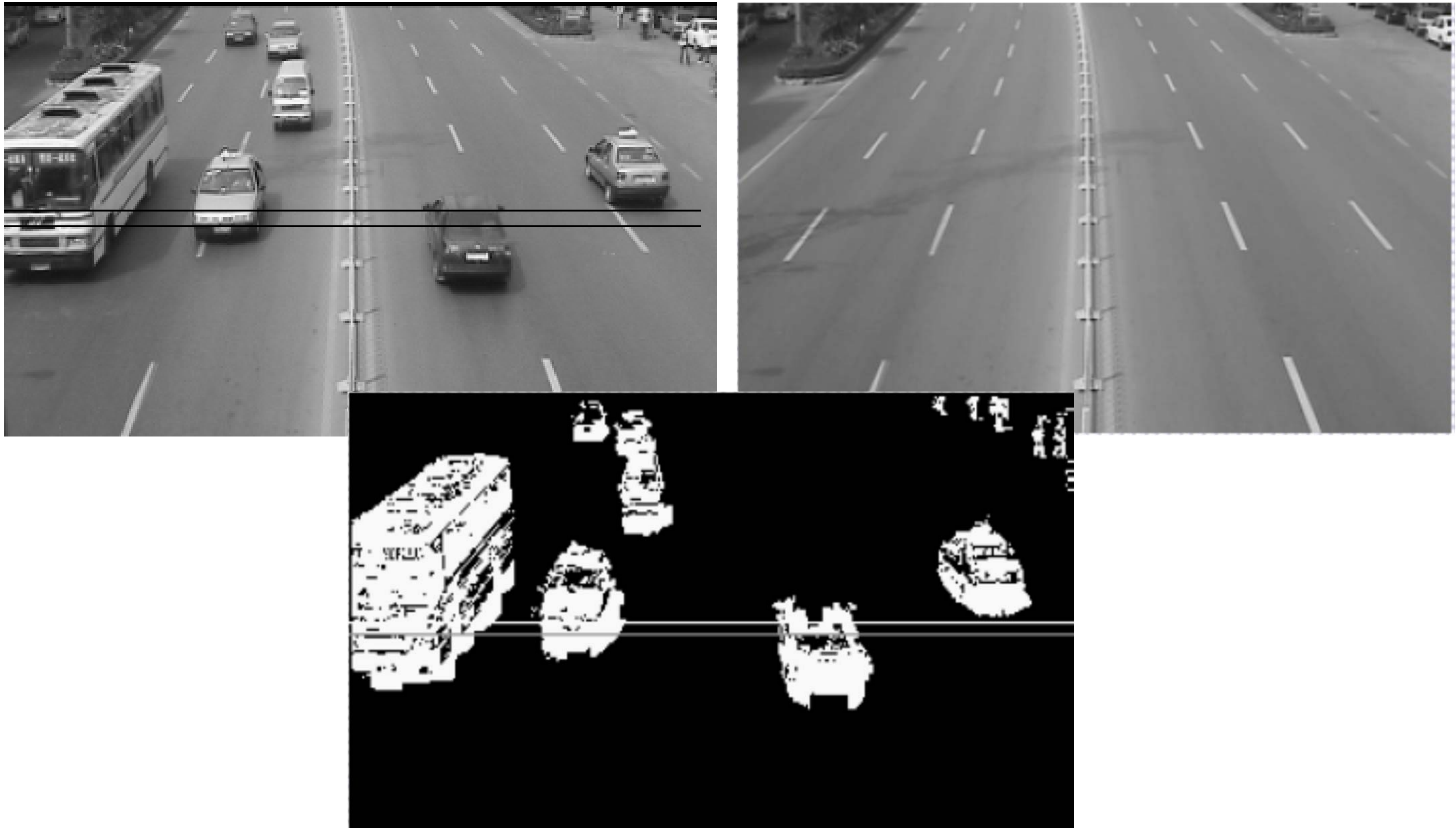


Image Subtraction

- Change detection is another major application by using image subtraction, such as:
  - Tracking moving vehicles
  - Tracking walking persons
  - Change detections

Image Subtraction



Examples: Change Detection

Image Averaging

- A noisy image:

$$g(x, y) = f(x, y) + n(x, y)$$

where $n(x, y)$ is the noise with zero average.

- Then averaging M different noisy images can reduce the noise:

$$\bar{g}(x, y) = \frac{1}{M} \sum_{i=1}^{M} g_i(x, y) \qquad \delta^2_{\bar{g}(x,y)} = \frac{1}{M} \sigma^2_{n(x,y)}$$

$$E\{\bar{g}(x, y)\} = f(x, y) \qquad \delta_{\bar{g}(x,y)} = \frac{1}{\sqrt{M}} \sigma_{n(x,y)}$$

Image Averaging

- As the number of noisy images used in the averaging process, *M*, increases,
  - the variability of the pixel values at each location decreases;
  - the averaged image $\bar{g}(x, y)$ approaches $f(x, y)$.

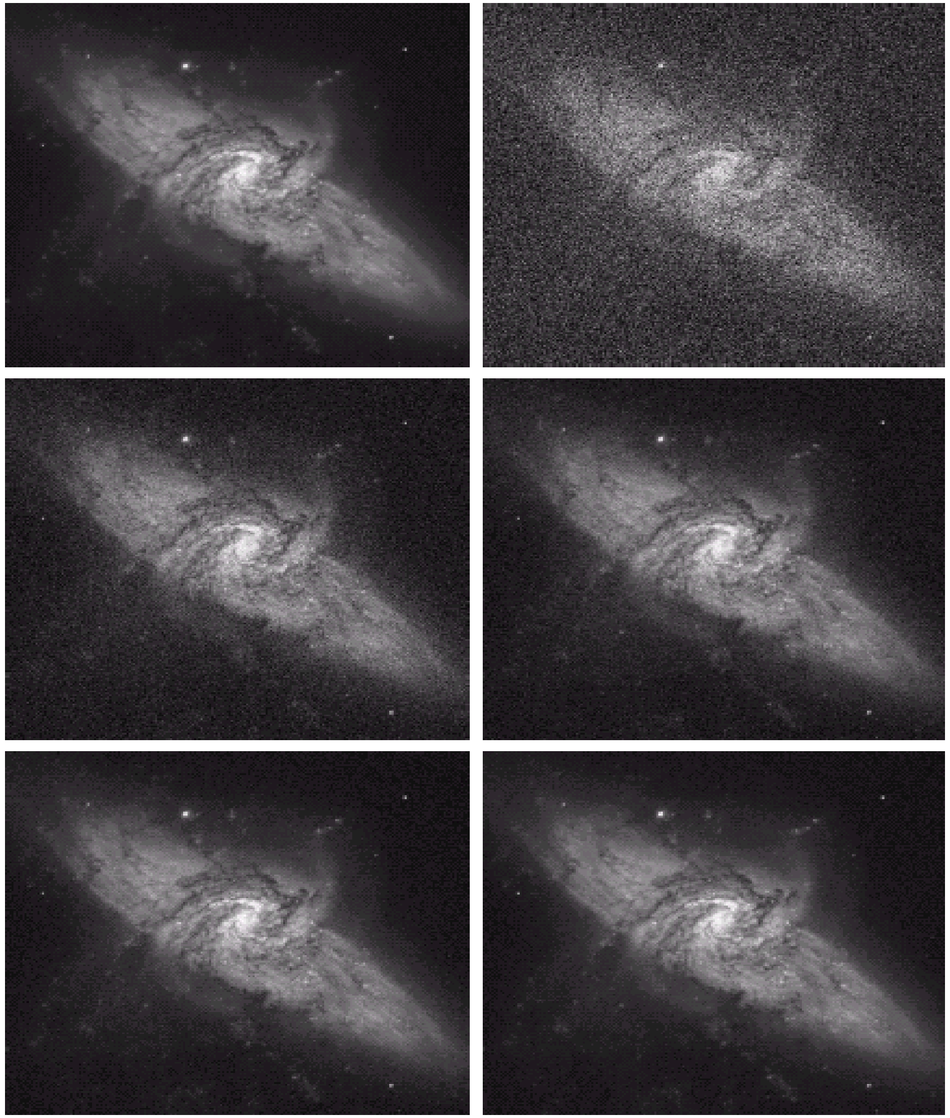
when $M \rightarrow \infty$,

$$\delta_{\bar{g}(x,y)} = \frac{1}{\sqrt{M}} \sigma_{n(x,y)} \rightarrow 0$$

$$\bar{g}(x, y) \rightarrow f(x, y)$$

Image Averaging

(a)
Image of Galaxy Pair NGC 3314



(b)
Image corrupted by additive Gaussian noise with zero mean and a standard deviation of 64 gray levels

(c)
Results of averaging $K = 8$ noisy images

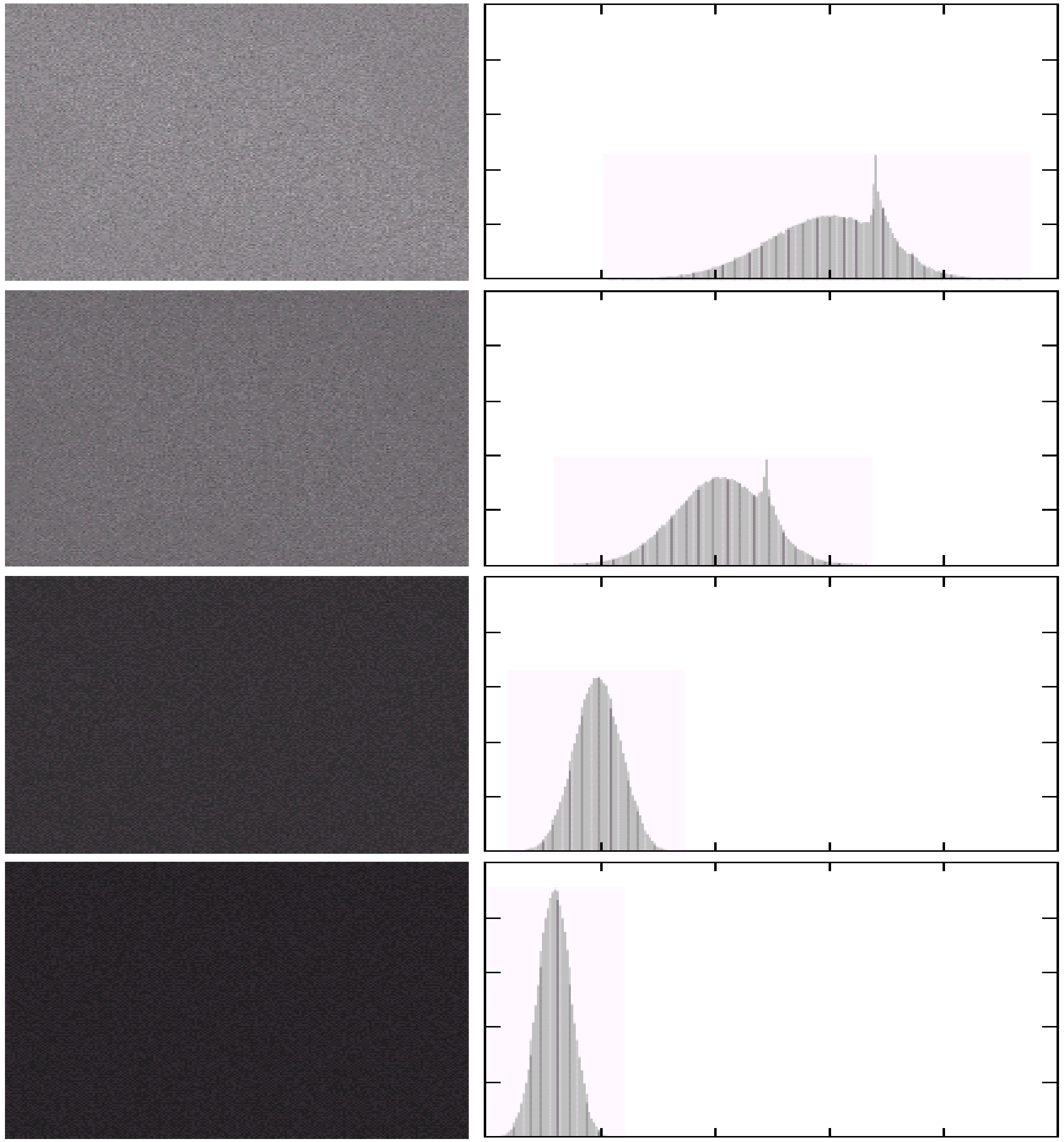
(d)
Results of averaging $K = 16$ noisy images

(e)
Results of averaging $K = 64$ noisy images

(f)
Results of averaging $K = 128$ noisy images

Image Averaging

Difference images between (a) and the four images in (c) through (f) in the last slide.

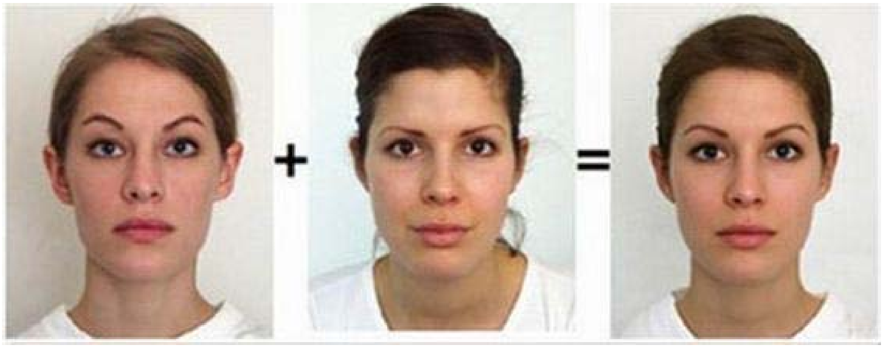


Corresponding histograms

?

Image Averaging

- Example: face averaging



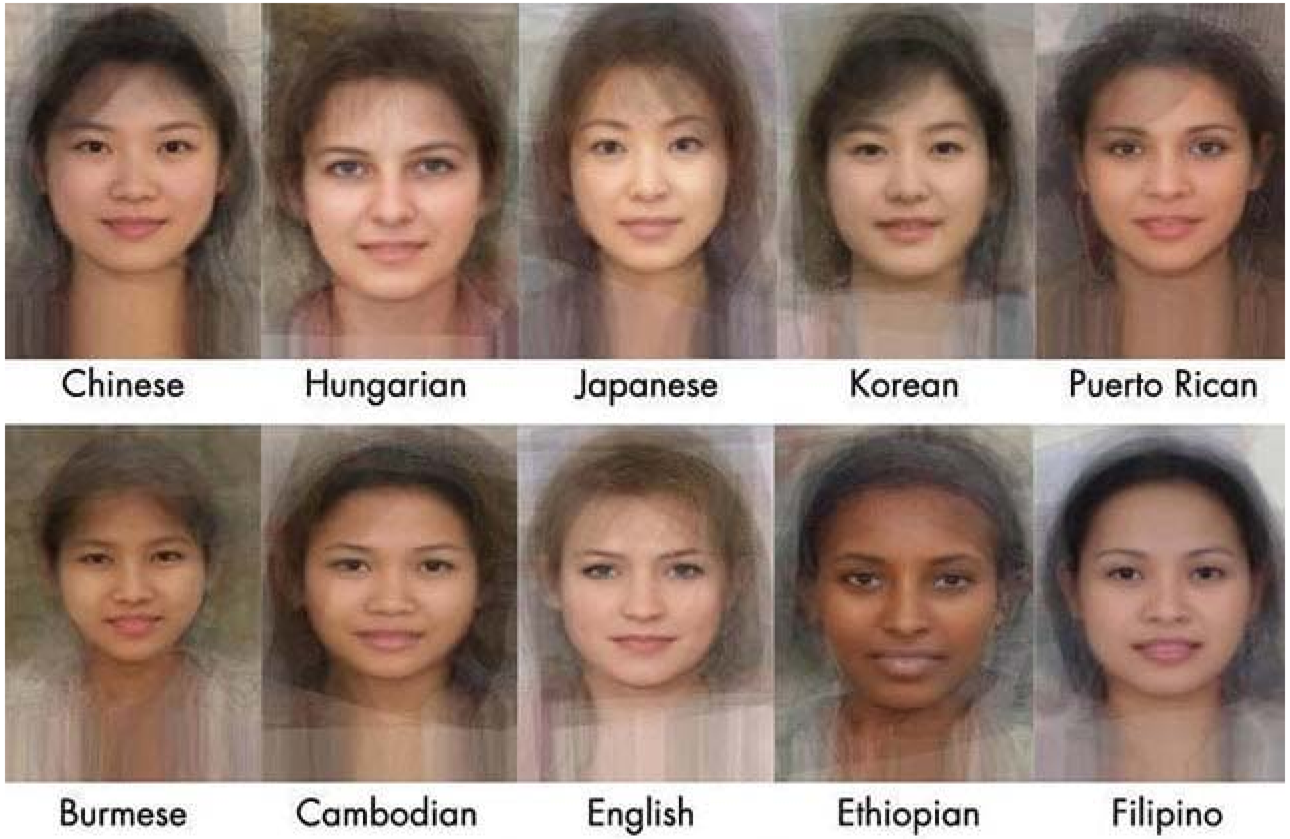


Image Histograms

- The histogram of a digital image with gray levels from 0 to $L - 1$ is a discrete function: $p_r(r_k) = n_k \qquad 0 \leq r_k \leq 1 \quad k = 0,1,2,\cdots,L-1$

  where r_k is the k-th gray level

  n_k is the number of pixels with the gray level r_k

- Normalized histogram: $p(r_k) = n_k/MN$, where MN is the total number of pixels in the image, and the sum of all components is 1.

Image Histograms

- The histogram of an image shows us the distribution of gray levels in the image. It is useful in image processing, especially in enhancement and segmentation.

1	2	3	4	5	6
6	4	3	2	2	1
1	6	6	4	6	6
3	4	5	6	6	6
1	4	6	6	2	3
1	3	6	4	6	6

1	2	3	4	5	6	gray level
5	4	5	6	2	14	histogram
5/36	4/36	5/36	6/36	2/36	14/36	normalized histogram

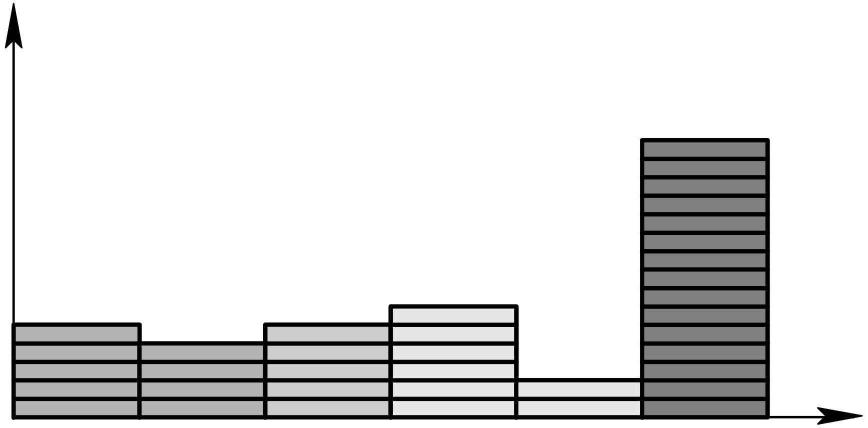
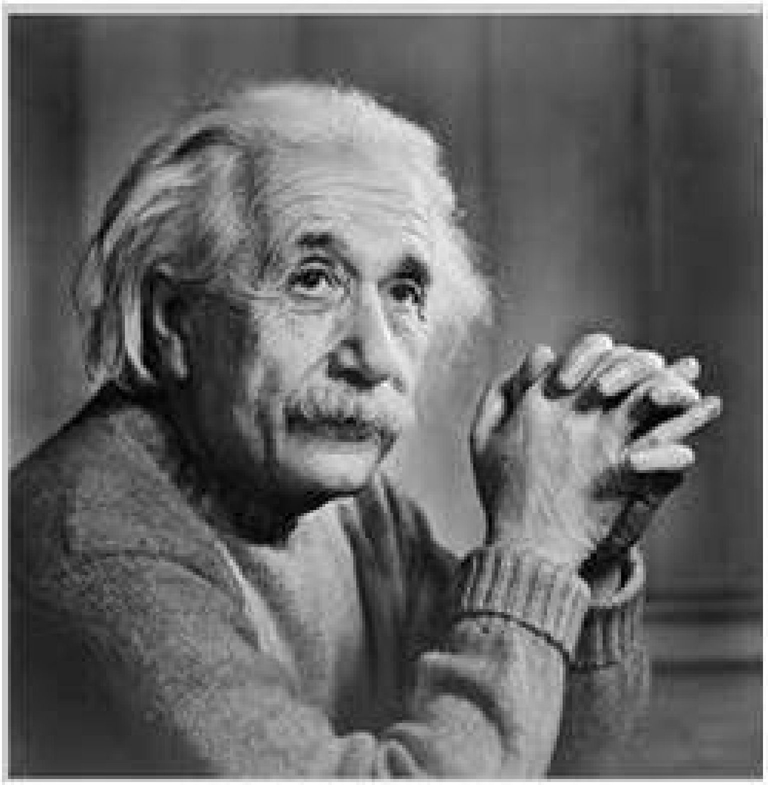
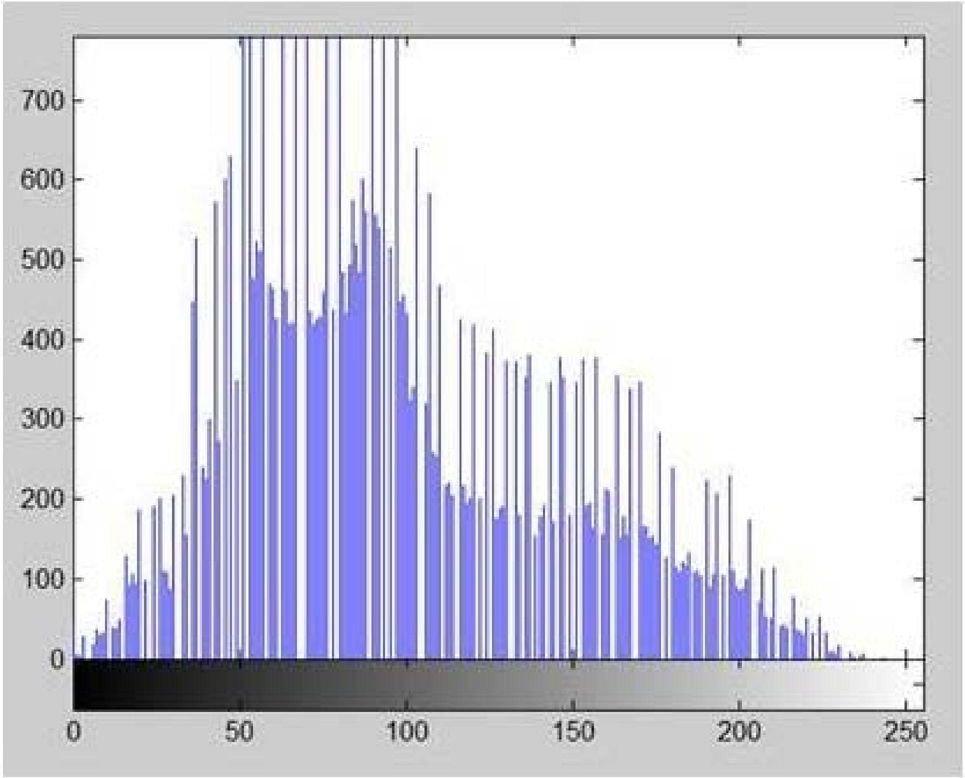


Image Histograms

an image

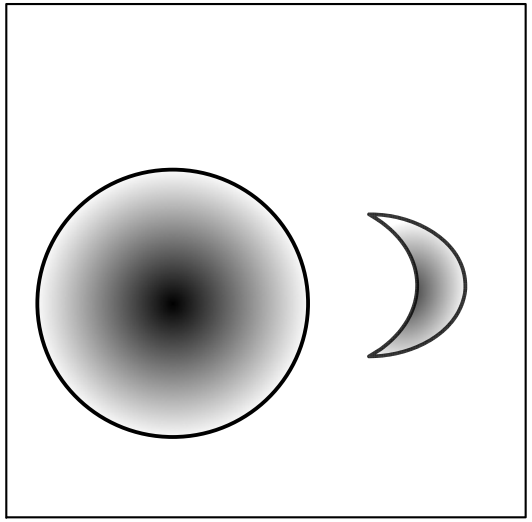


histogram



Properties of Image Histograms

- The histogram only shows the distribution of gray levels in the image, and it doesn't include the location information of pixels.
- One image has its corresponding histogram, but different images may have the same histograms.



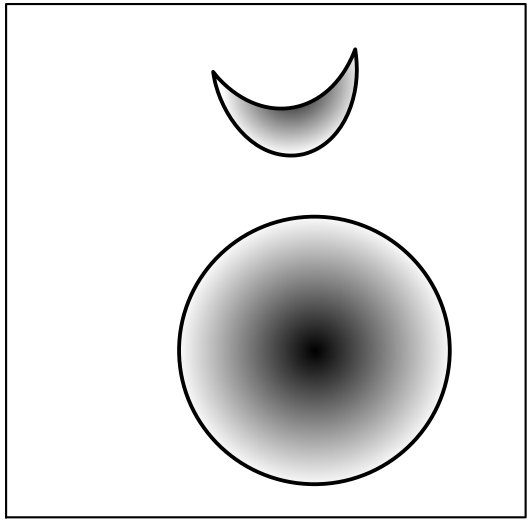
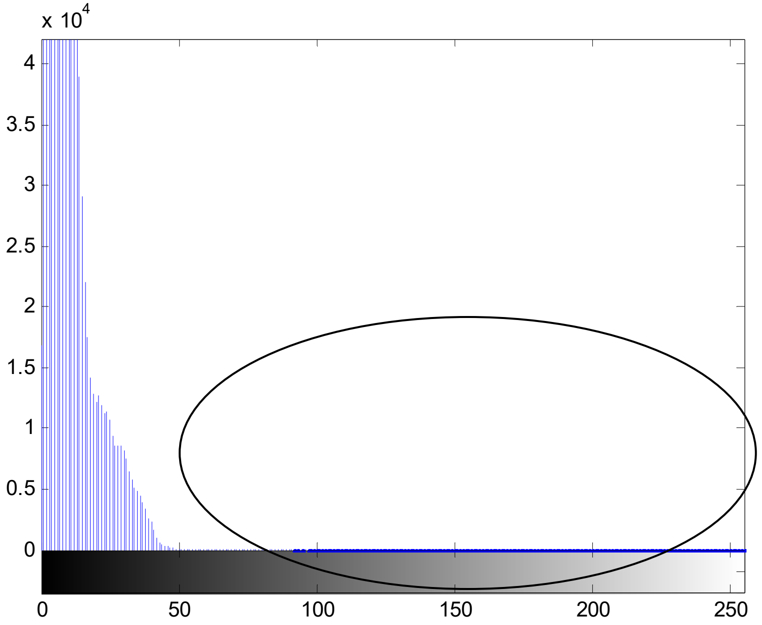


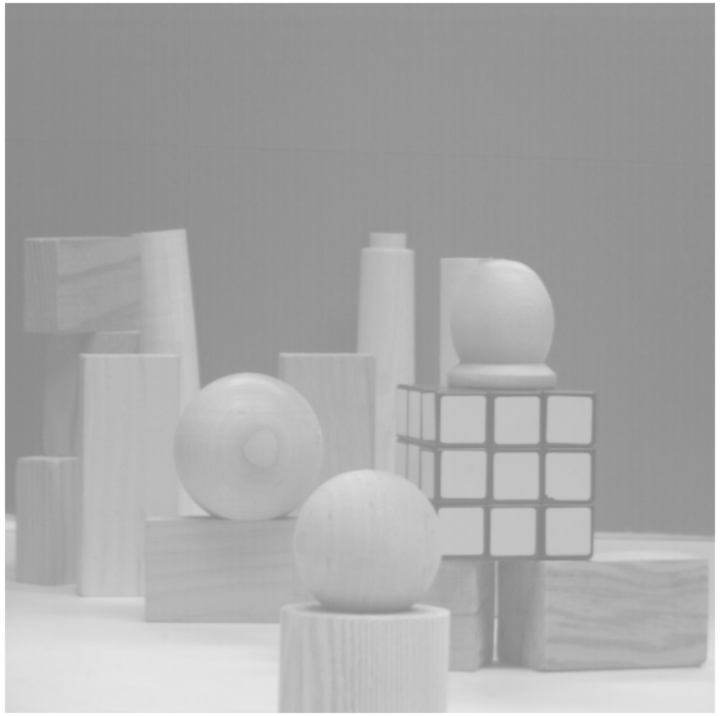
Image Histograms



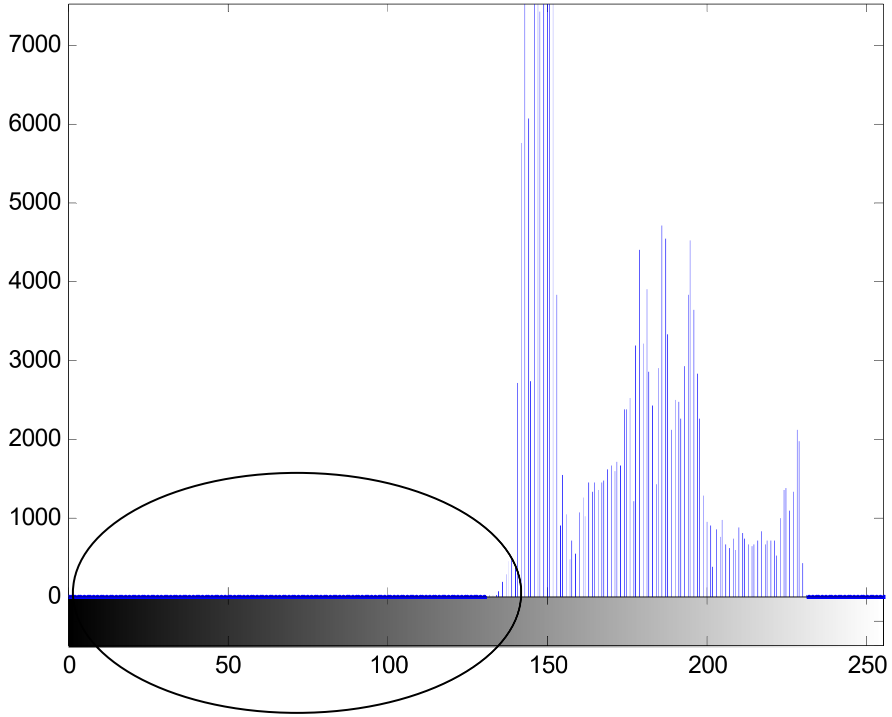
It is a baby in the cradle!

Histogram information reveals that image is *under-exposed*.

Image Histograms

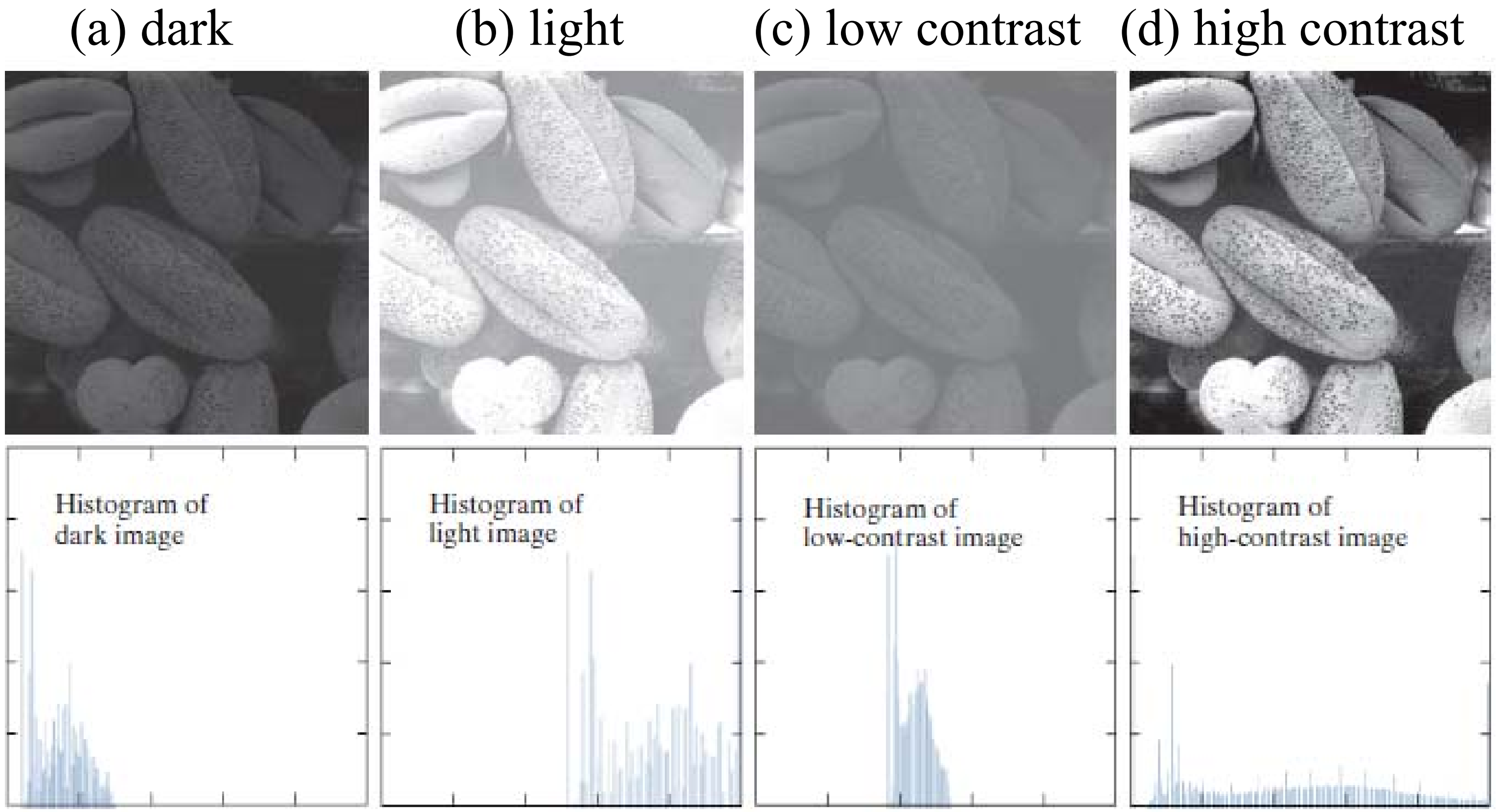


An *over-exposed* image



Histogram Examples

- A selection of images and their histograms.
- Notice the relationships between the images and their histograms.
- Note that the high contrast image has the most evenly spaced histogram.



Four image types and their corresponding histograms.
The horizontal axis of the histograms are values of r_k and the vertical axis are values of $p(r_k)$.

Histogram Equalization

- Histogram equalization
  - Basic idea: find a map $T(r)$ such that the histogram of the modified (equalized) image is flat (uniform).
- Assuming initially continuous intensity values, and let r represent the gray levels, which have been normalized to the interval [0,1], with $r=0$ representing black and $r=1$ representing white. For any r, define a transformation:

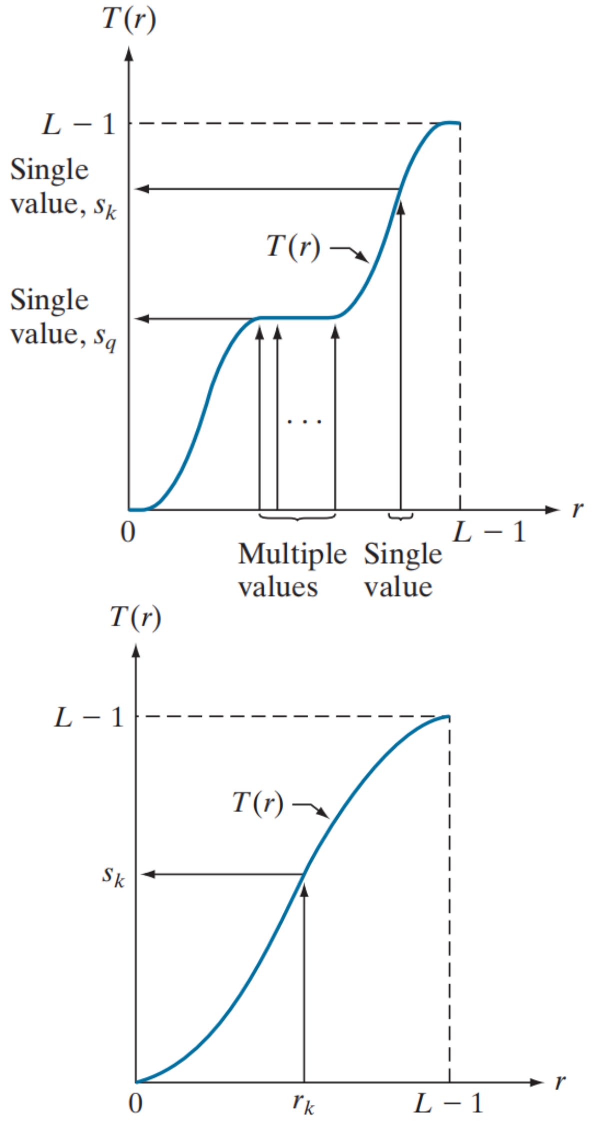
$$s = T(r)$$

Histogram Equalization

- The transformation, $s = T(r)$, produces a level s for every pixel value r in the input image. We assume that the transformation function $T(r)$ satisfies the following conditions:
  - condition (1): $T(r)$ is single-value and monotonically increasing in the interval
  - condition (2): $0 \leq r \leq 1$; $0 \leq T(r) \leq 1$ for $0 \leq r \leq 1$.

Histogram Equalization

- The condition (1) preserves the increasing order from black to white in the output image.
- The condition (2) guarantees that the output gray levels will be in the same range as the input levels.



Histogram Equalization

- Let $p_r(r)$ and $p_s(s)$ denote the probability density function (PDF) of random variables r and s, respectively. A basic result from probability theory is that, if $p_r(r)$ and $T(r)$ are known and $T^{-1}(s)$ satisfies condition (1). Then $p_s(s)$ can be obtained using a rather simple formula:

$$s = T(r), \qquad p_s(s) = p_r(r)\left|\frac{dr}{ds}\right|$$

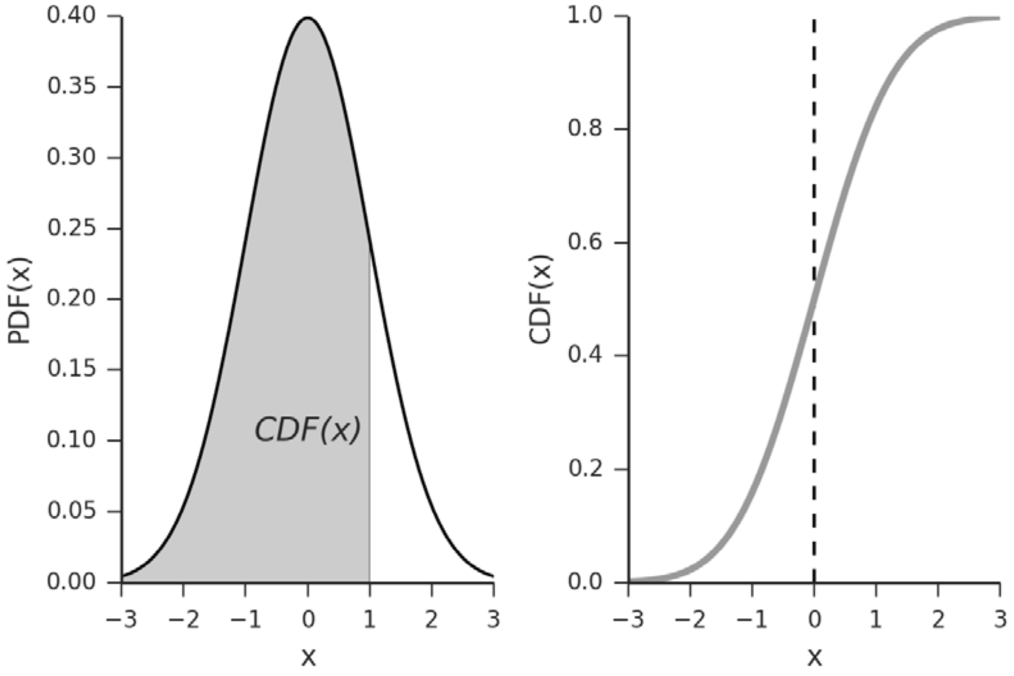
Histogram Equalization

- Thus the PDF of s is determined by the gray level PDF of input image and by the chosen transformation function.
- So for a given input image, we can change its histogram by some transformation, it is the idea of histogram equalization.
- Take the following equation as the transform function:

$$s = T(r) = (L-1)\underbrace{\int_0^r p_r(\omega)d\omega}_{\text{cumulative distribution function (CDF)}}$$

Histogram Equalization

- The integral of a PDF is called the cumulative distribution function (CDF) and is the area under the PDF.
- PDFs are always positive, so CDF should be single values and monotonically increasing.
- Similarly, CDF for variables in the range [0,1] is also in the range [0,1].



Histogram Equalization

- By taking the derivative of s with respect to r, one gets

$$\frac{ds}{dr} = \frac{dT(r)}{dr} = (L-1)\frac{d}{dr}[\int_0^r p_r(\omega)d\omega]$$

- By Leibniz's rule, we know that the derivative of a definite integral with respect to its upper limit is simply the integrand evaluated at the limit.

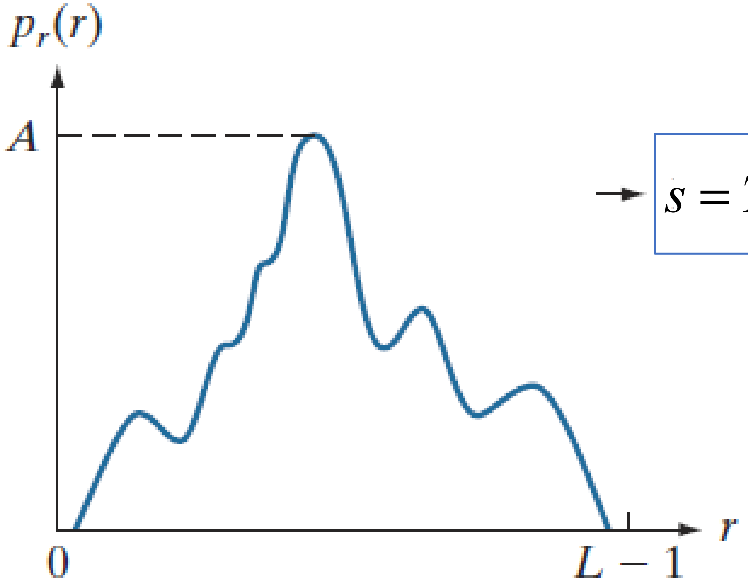
$$\frac{ds}{dr} = (L-1)\frac{d}{dr}[\int_0^r p_r(\omega)d\omega] = (L-1)p_r(r)$$

- So we have

$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right| = p_r(r)\left|\frac{1}{(L-1)p_r(r)}\right| = \frac{1}{L-1}, \qquad 0 \le s \le L-1.$$

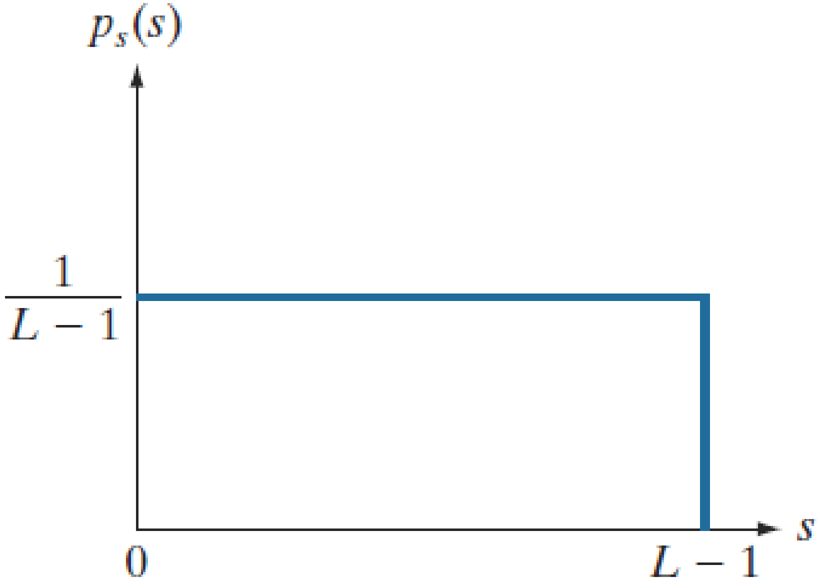
Histogram Equalization

- It is important to note that $T(r)$ depends on $p_r(r)$, but $p_s(s)$ is always uniform, independent of the form of $p_r(r)$.



An arbitrary PDF

$$\rightarrow \boxed{s = T(r) = (L-1)\int_0^r p_r(\omega)d\omega} \rightarrow$$



Equalized PDF

Histogram Equalization

- For discrete values (which is the case of a digital image), we use probability mass functions (PMFs) instead of PDFs. Define $p_r(r_k)$ as the probability of occurrence of gray level r_k.

$$p_r(r_k) = \frac{n_k}{MN} \quad 0 \le r_k \le 1 \quad k = 0,1,\cdots,l-1$$

where MN is the total number of pixels, n_k is the number of pixels that have a gray level r_k, l is the total number of possible gray levels.

- The transformation is

$$s_k = T(r_k) = (L-1)\underbrace{\sum_{j=0}^{k} p_r(r_j)}_{\text{CDF}}, \quad k = 0,1,\cdots,l-1$$

Histogram Equalization

Steps:

1) Find probability of the input image.
2) Calculate the CDF based on the PMF.
3) Multiply the CDF values by the maximum gray-level value $L - 1$ and round the results to obtain s_k.
4) Map the original gray-level value to the result obtained in Step 3.

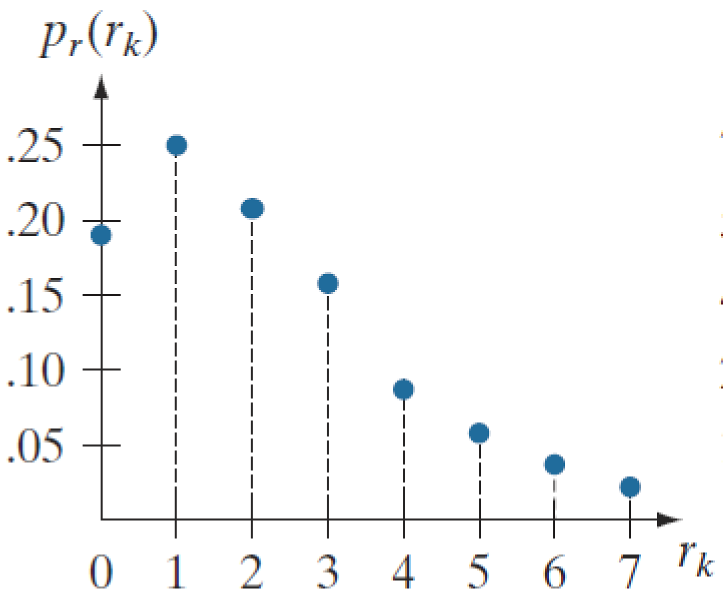
$$p_r(r_k) = \frac{n_k}{MN} \quad 0 \le r_k \le 1 \quad k = 0,1,\cdots,l-1$$

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j), \qquad k = 0,1,\cdots,l-1$$

Histogram Equalization

- An example, where the image is 64×64 pixels in size, with 8 gray levels. The distribution is as following table.

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



original histogram

Histogram Equalization

- Processing steps：

$$s_0 = T(r_0) = 7\sum_{j=0}^{0} P_r(r_j) = 7P_r(r_0) = 1.33$$

$$s_1 = T(r_1) = 7\sum_{j=0}^{1} P_r(r_j) = 7[P_r(r_0) + P_r(r_1)] = 3.08$$

$$s_2 = T(r_2) = 7\sum_{j=0}^{2} P_r(r_j) = 7[P_r(r_0) + P_r(r_1) + P_r(r_2)] = 4.55$$

…

- Finally we have

$$s_0 = 1.33,\ s_1 = 3.08,\ s_2 = 4.55,\ s_3 = 5.67,$$

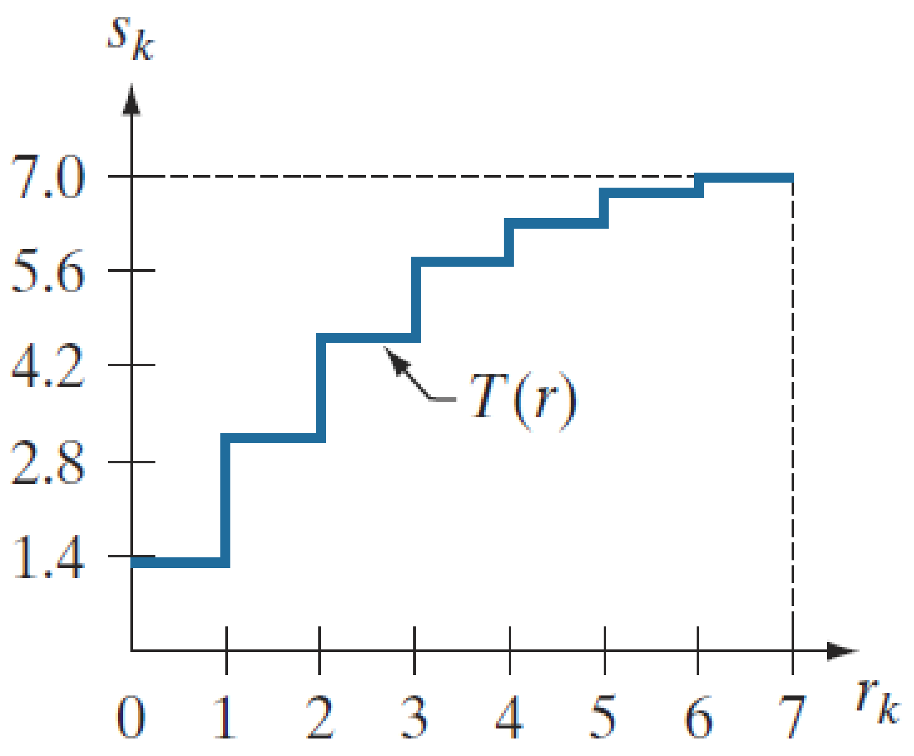
$$s_4 = 6.23,\ s_5 = 6.65,\ s_6 = 6.86,\ s_7 = 7.00.$$

Histogram Equalization

- The transform function is shown in the right figure.

- Since the gray level is 8, we should adjust the output values to the nearest integer numbers:

$$s_0 = 1.33 \approx 1; \qquad s_1 = 3.08 \approx 3;$$
$$s_2 = 4.55 \approx 5; \qquad s_3 = 5.67 \approx 6;$$
$$s_4 = 6.23 \approx 6; \qquad s_5 = 6.65 \approx 7;$$
$$s_6 = 6.86 \approx 7; \qquad s_7 = 7.$$



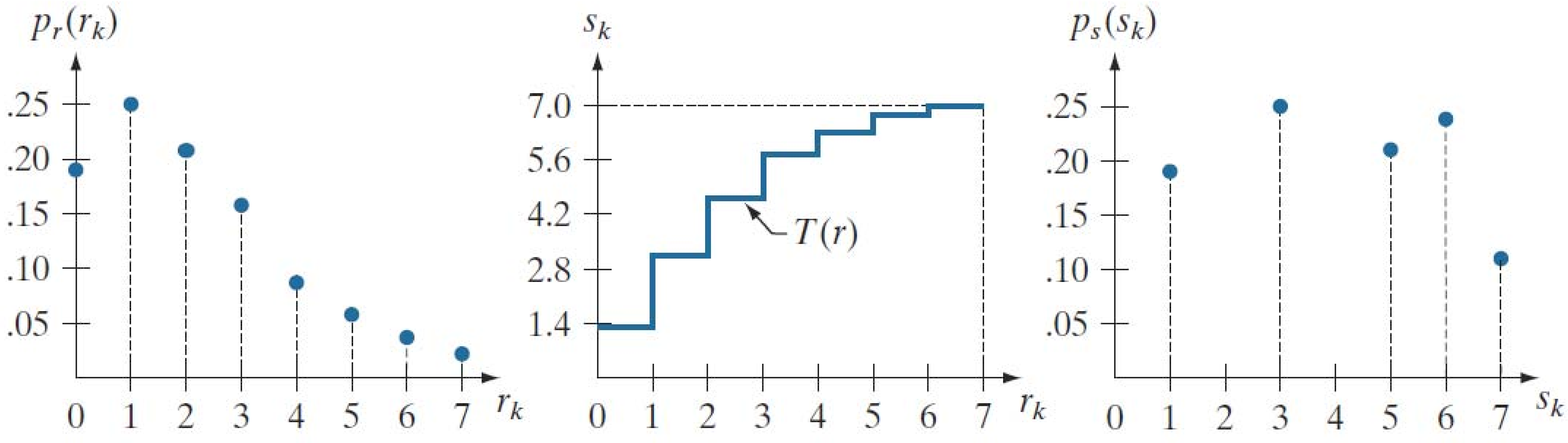
Histogram Equalization

- We can see that there are only 5 effective gray levels in the output image:

$$s_0 \approx 1,\ s_1 \approx 3,\ s_2 \approx 5,\ s_3 \approx 6,\ s_4 \approx 6,\ s_5 \approx 7,\ s_6 \approx 7,\ s_7 \approx 7$$

r_k	n_k	$p_r(r_k)$	s_k	n'_k	$p_s(s_k)$
$r_0 = 0$	790	0.19	$s_0 = 1$	790	0.19
$r_1 = 1$	1023	0.25	$s_1 = 3$	1023	0.25
$r_2 = 2$	850	0.21	$s_2 = 5$	850	0.21
$r_3 = 3$	656	0.16	$s_3 = 6$	985	0.24
$r_4 = 4$	329	0.08	$s_4 = 6$		
$r_5 = 5$	245	0.06	$s_5 = 7$	448	0.11
$r_6 = 6$	122	0.03	$s_6 = 7$		
$r_7 = 7$	81	0.02	$s_7 = 7$		

Histogram Equalization

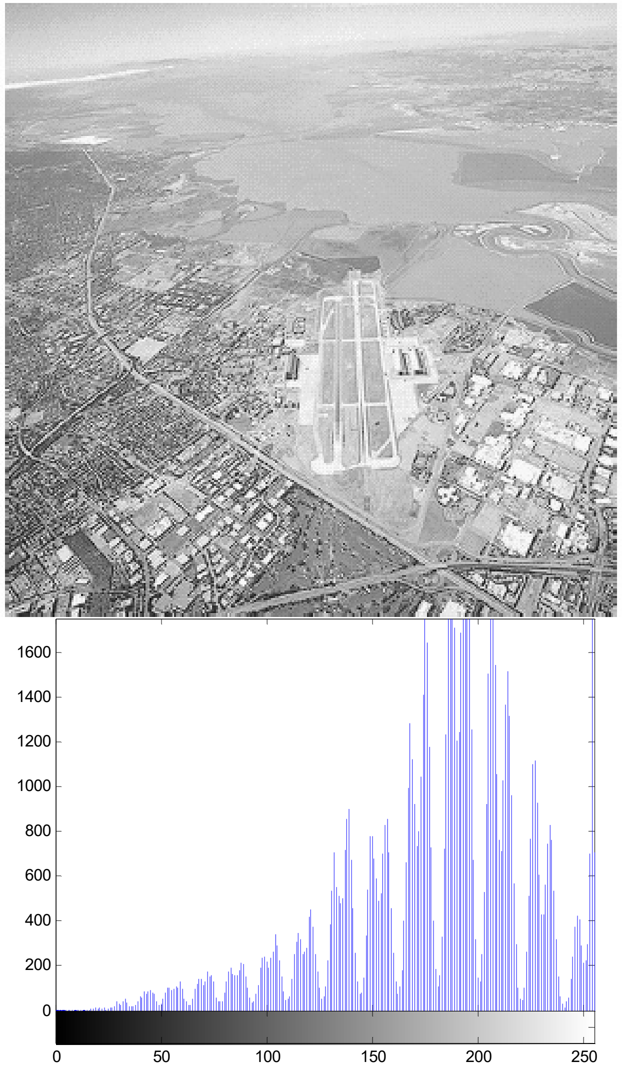


original histogram

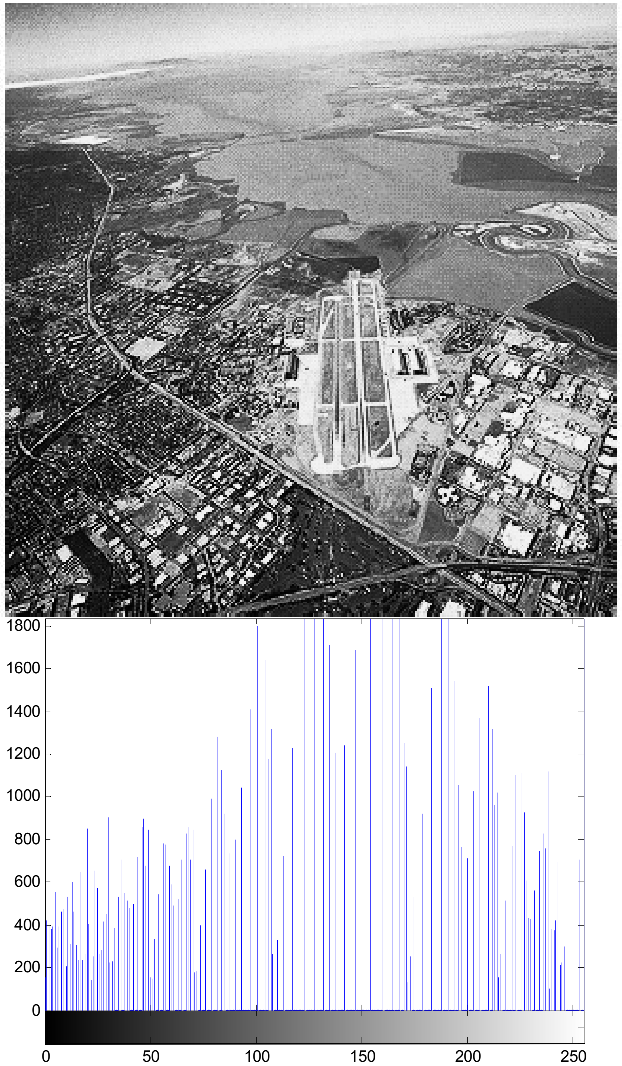
transform function

equalized histogram

Histogram Equalization



before equalization

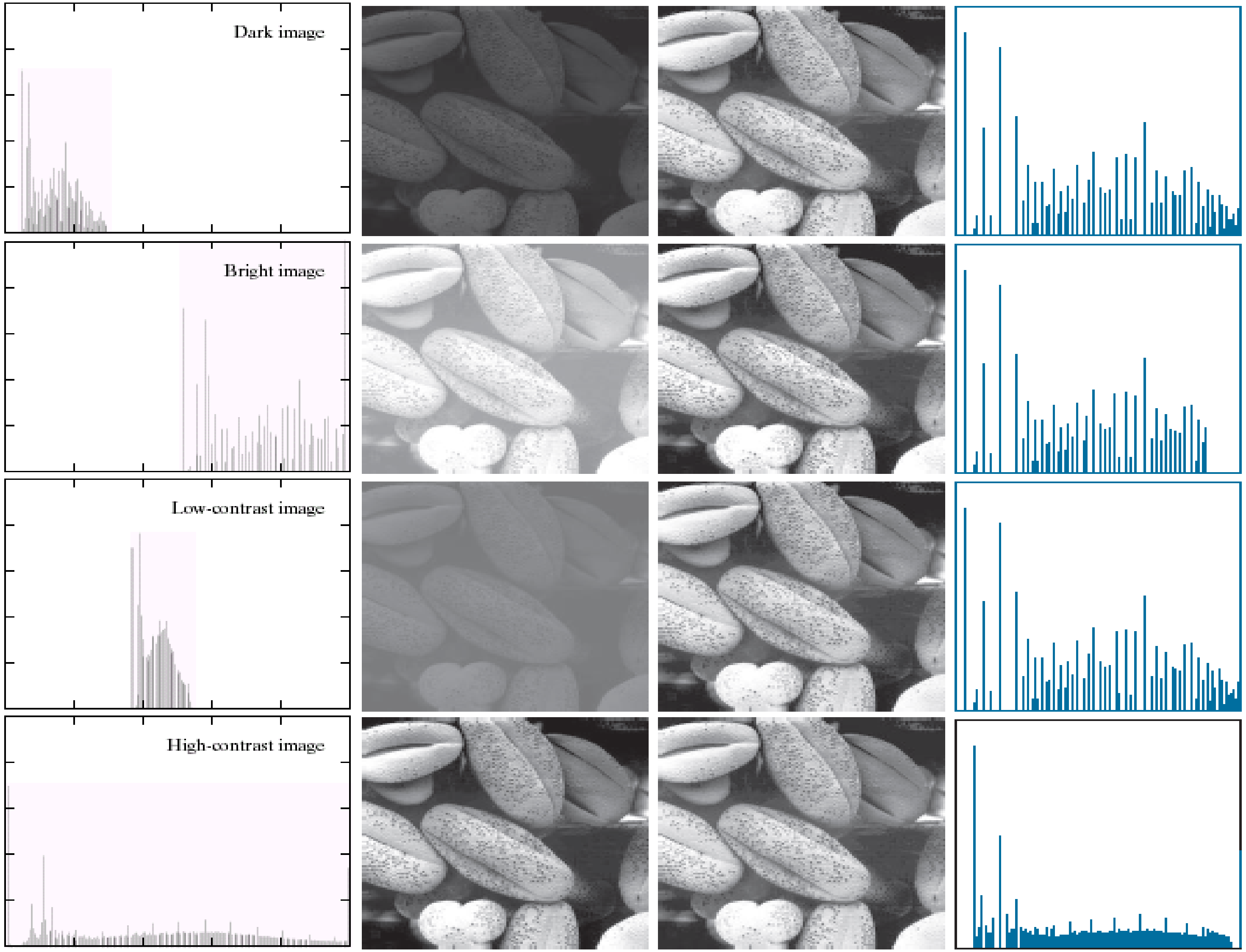


after equalization

Histogram Equalization

Column 1: histograms of four image types

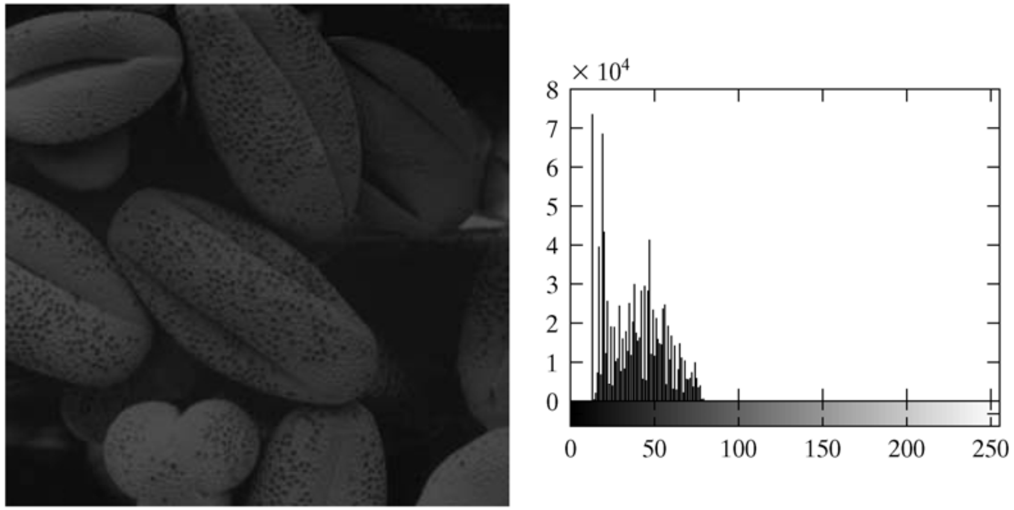
Column 2: four image types

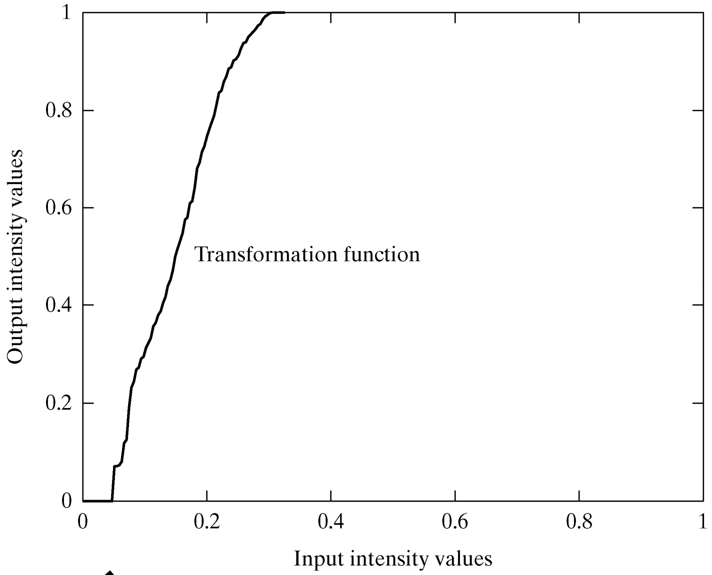


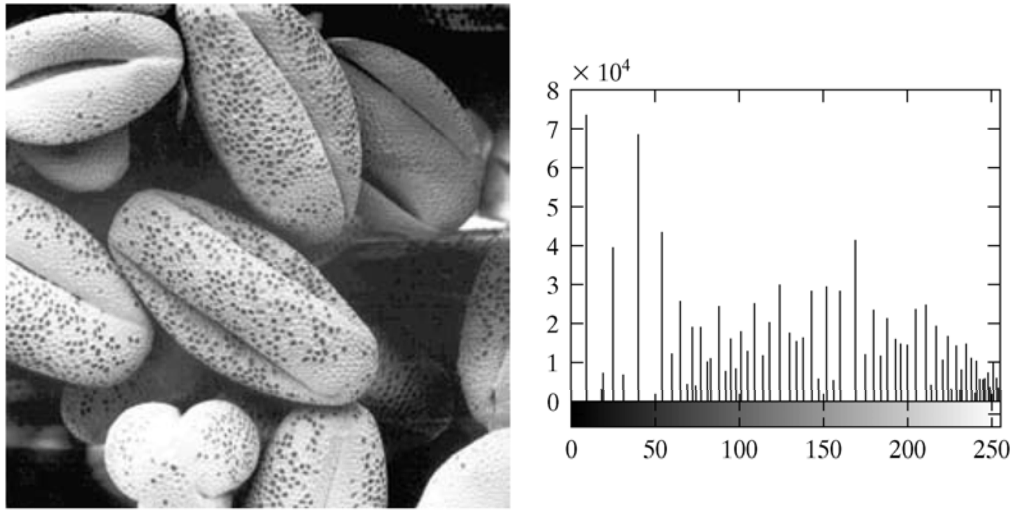
Column 3: corresponding histogram-equalized images

Column 4: histograms of equalized images

Equalization Transformation Function

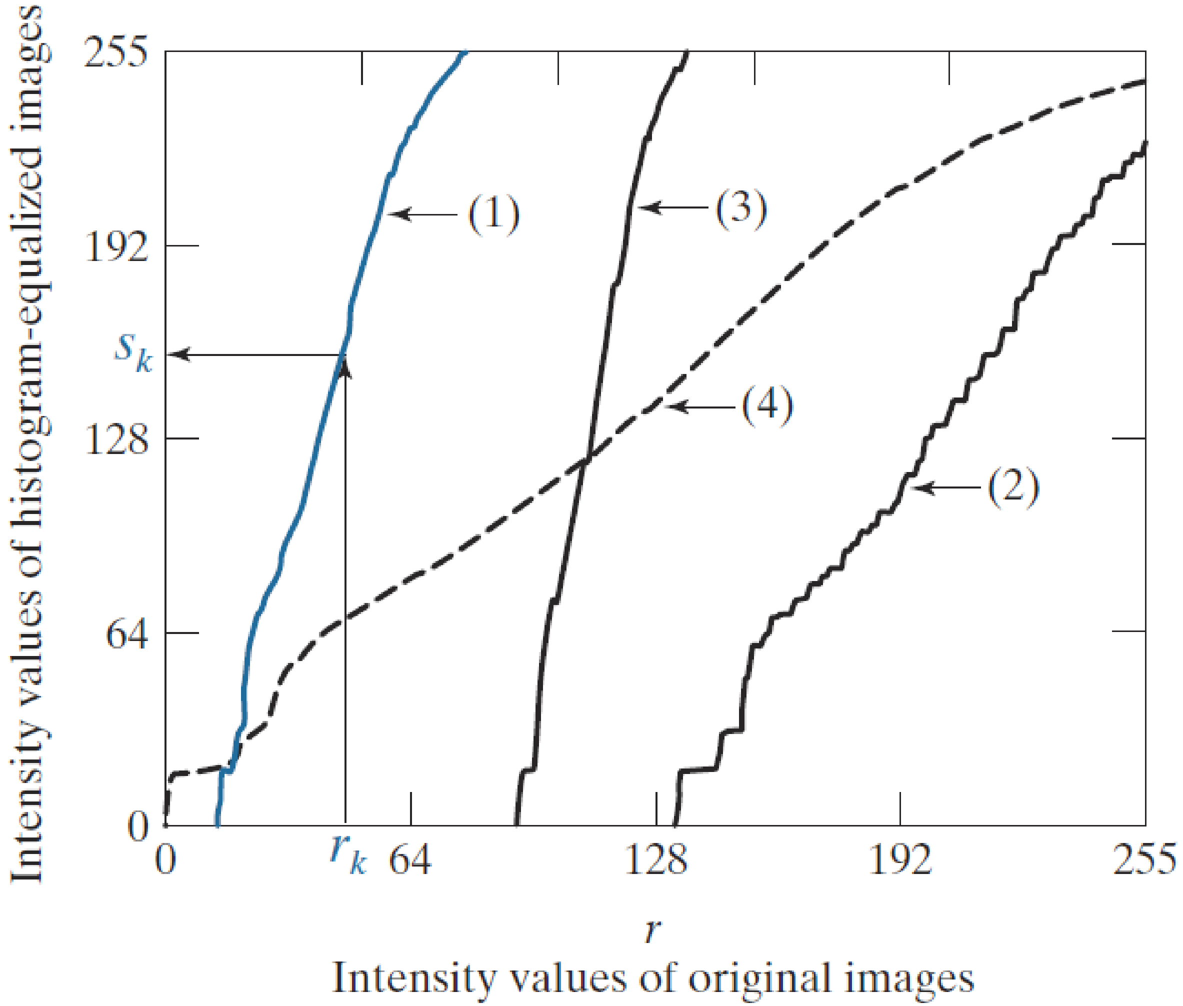




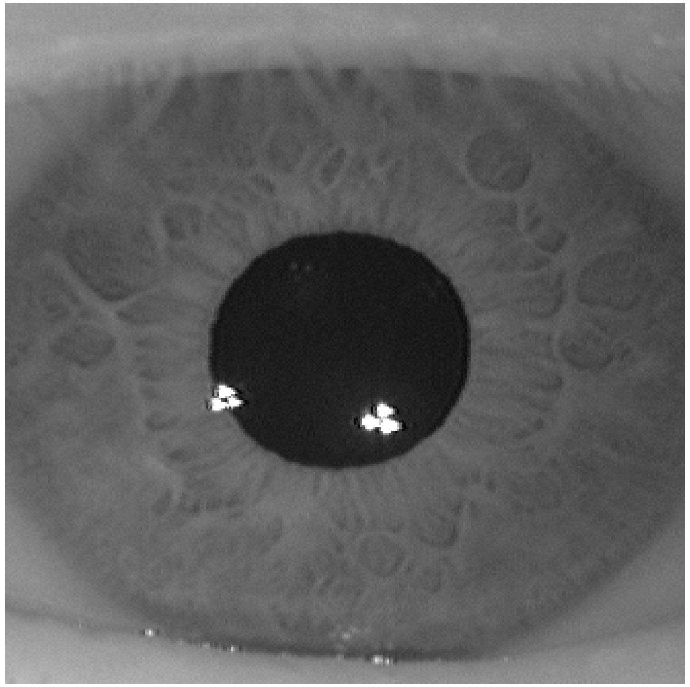


Equalization Transformation Functions

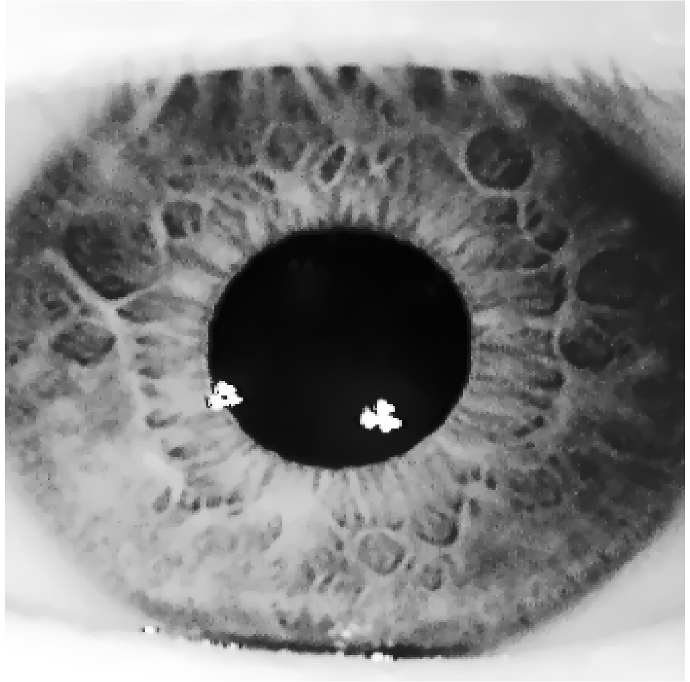
- Transformation functions for histogram equalization.
- Transformations (1) through (4) were obtained using the histograms of the images on Column 2 of the figure in Slide 64.
- Mapping of one intensity value r_k in image 1 to its corresponding value s_k is shown on the right.



Histogram Equalization



Before



After

Summary

- In this lecture we have learnt:
  - Various kinds of basic intensity transformation functions (point processing)
    - ➢Thresholding
    - ➢Logarithmic transformation
    - ➢Power law transforms
    - ➢Gray level slicing
    - ➢Bit plane slicing
    - ➢Image subtraction
    - ➢Image averaging
  - Histogram processing (equalization)

Optional Homework

Check the Textbook!

- **Chapter 3: Problems 3.2, 3.4, 3.5(a), 3.6**

- Homework answers will be provided at the end of each week.