DIGITAL IMAGE PROCESSING

Image Enhancement

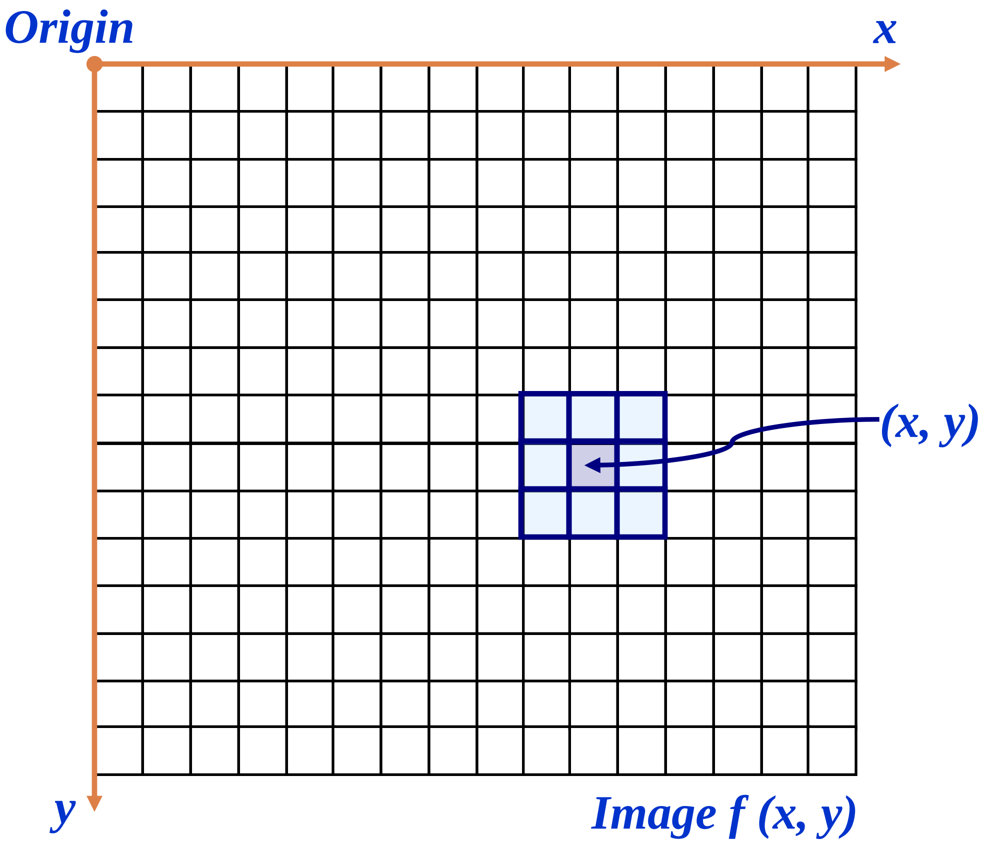
(Point Processing)

Contents

- In this lecture we will look at image enhancement point processing techniques:
  - What is point processing?
  - Negative images
  - Thresholding
  - Logarithmic transformation
  - Power law transforms
  - Grey level slicing
  - Bit plane slicing
  - Histogram Specification

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)



Point Processing

- The simplest spatial domain operations occur when the neighbourhood is simply the pixel itself
- In this case T is referred to as a *grey 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

Point Processing Example: Negative Images

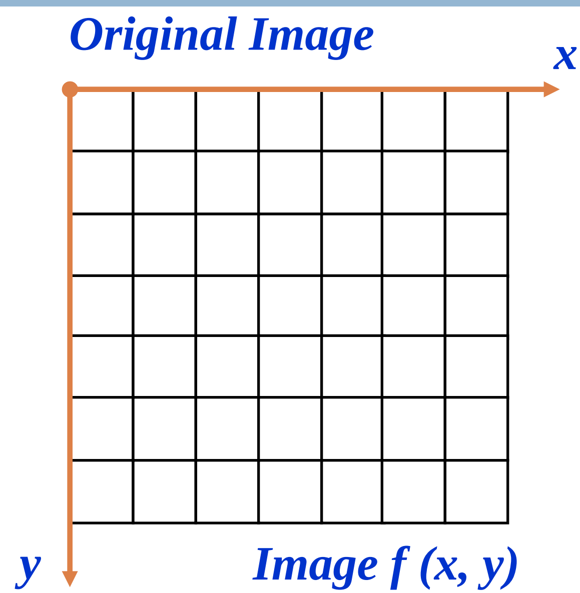
- Negative images are useful for enhancing white or grey detail embedded in dark regions of an image

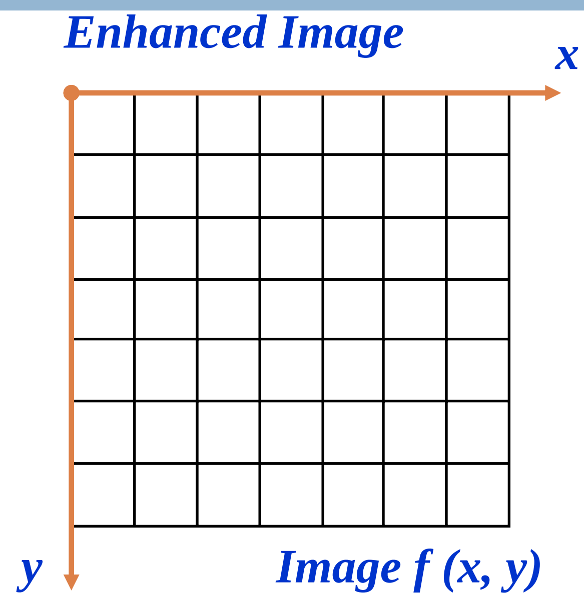
Negative Image

Original mage

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Point Processing Example: Negative Images (cont…)



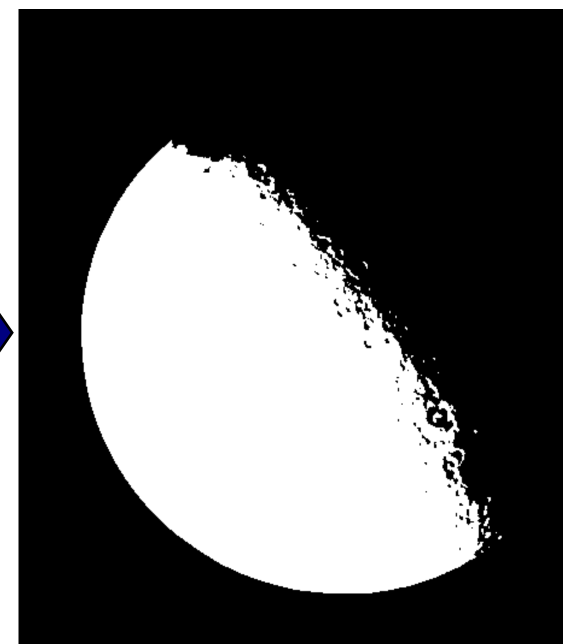


$$s = intensity_{max} - r$$

Point Processing Example: Thresholding

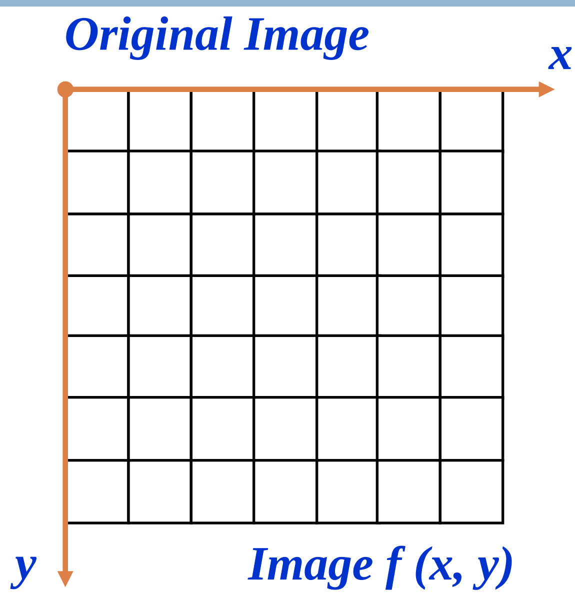
- Thresholding transformations are particularly useful for segmentation in which we want to isolate an object of interest from a background

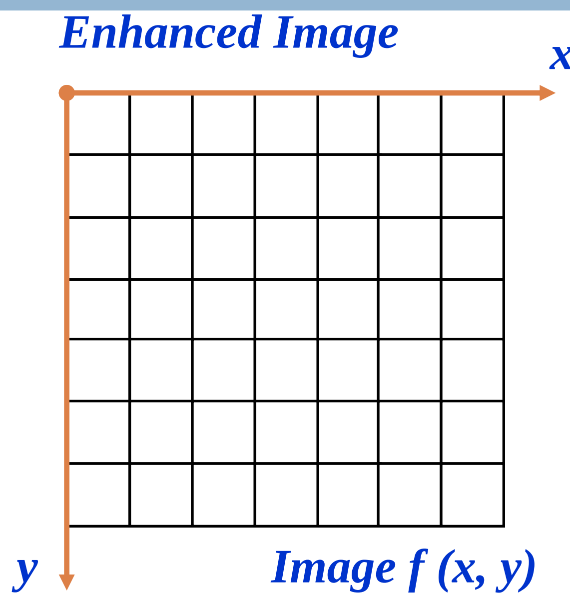
$$s = \begin{cases} 1.0 & r > \text{threshold} \\ 0.0 & r <= \text{threshold} \end{cases}$$



Images taken from Gonzalez & Woods, Digital Image Processing (2002)

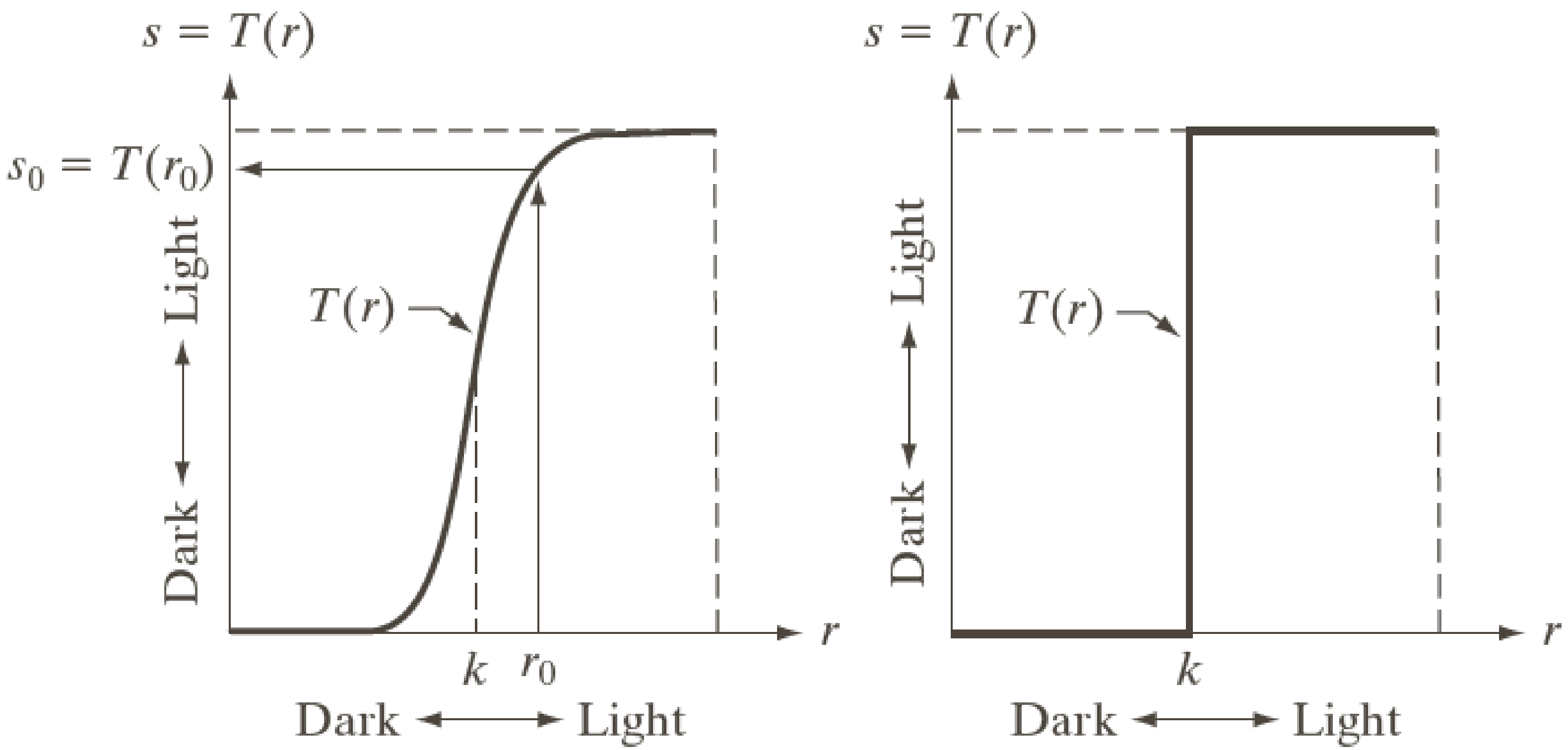
Point Processing Example: Thresholding (cont…)





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

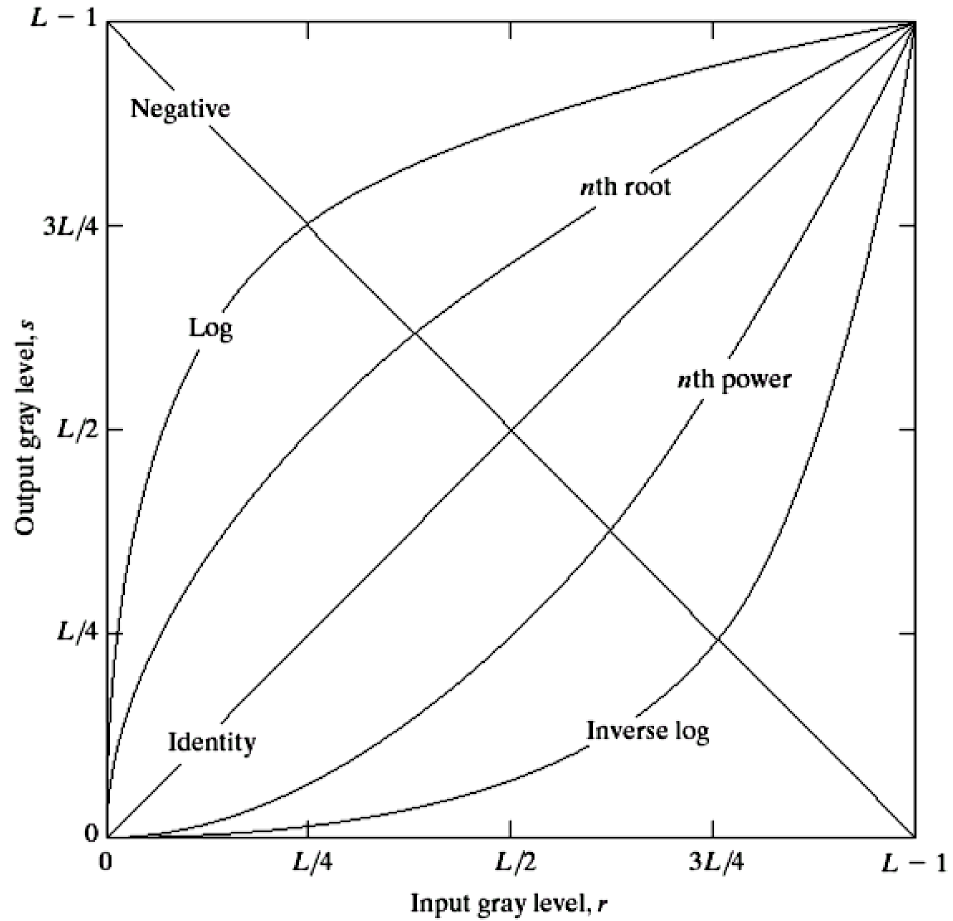
Intensity Transformations



Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Basic Grey Level Transformations

- There are many different kinds of grey level transformations
- Three of the most common are shown here
  - Linear
    - Negative/Identity
  - Logarithmic
    - Log/Inverse log
  - Power law
    - n^{th} power/n^{th} root



Logarithmic Transformations

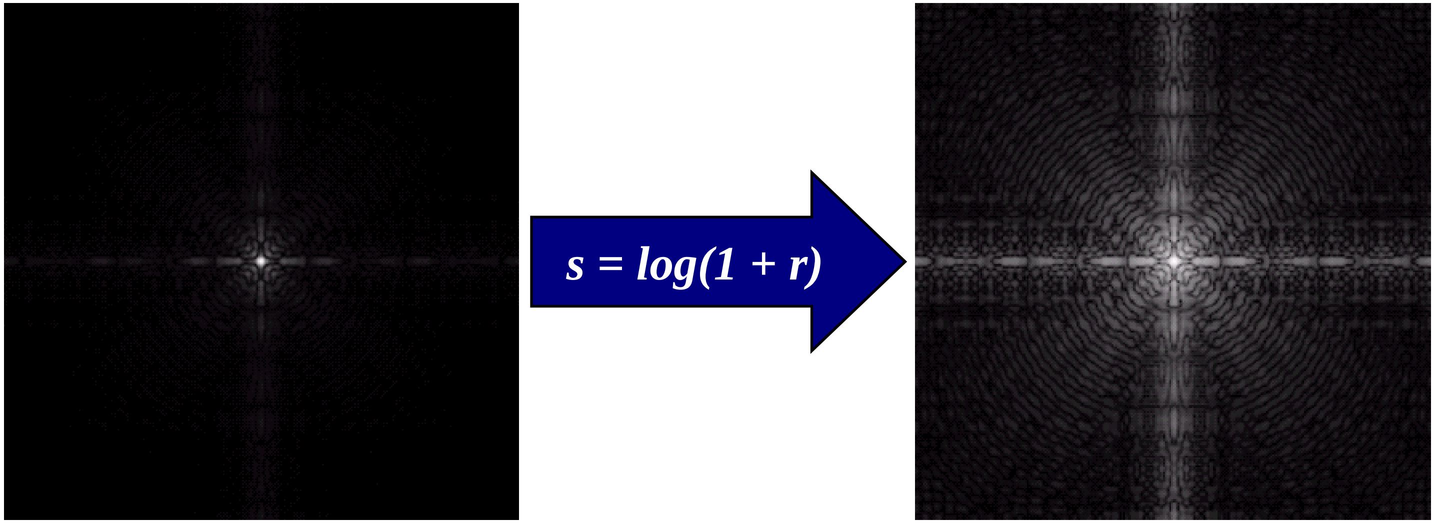
- The general form of the log transformation is

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

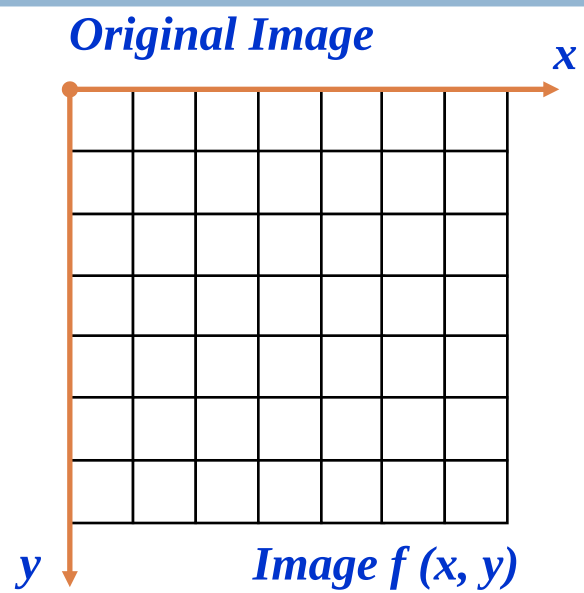
- The log transformation maps a narrow range of low input grey level values into a wider range of output values
- The inverse log transformation performs the opposite transformation

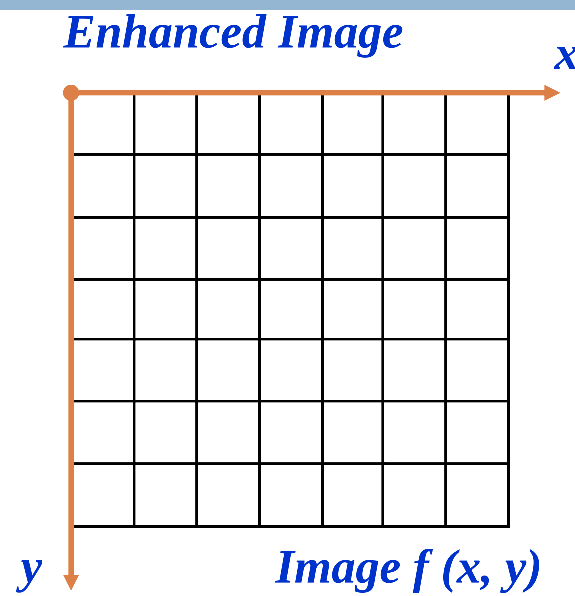
Logarithmic Transformations (cont…)

- Log functions are particularly useful when the input grey level values may have an extremely large range of values
- In the following example the Fourier transform of an image is put through a log transform to reveal more detail



Logarithmic Transformations (cont…)





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

We usually set c to 1

Grey levels must be in the range [0.0, 1.0]

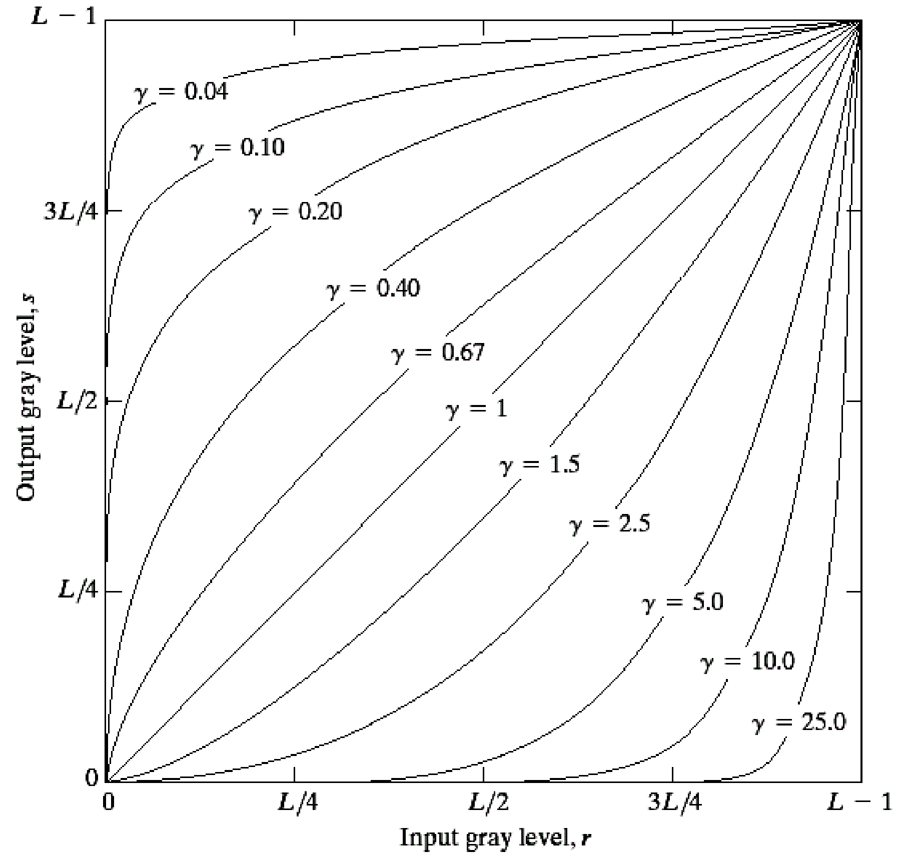
Power Law Transformations

- Power law transformations have the following form

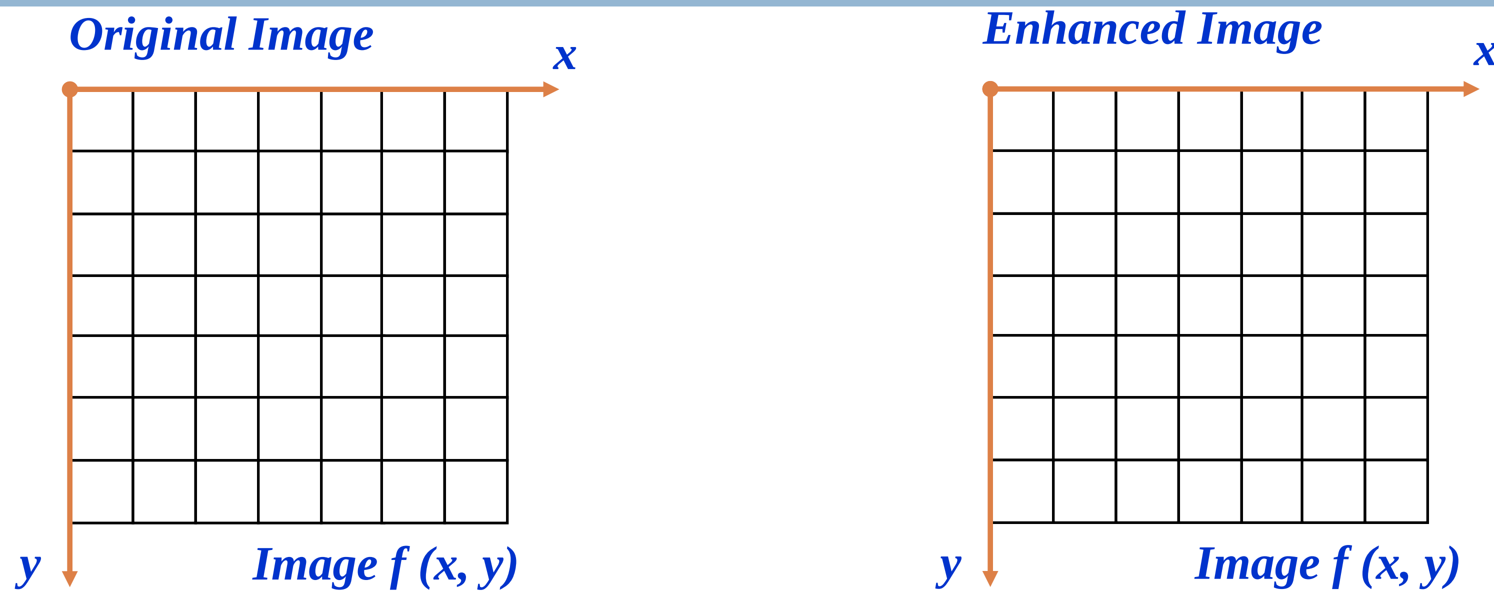
$$s = c * r^{\gamma}$$

- 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 Transformations (cont…)



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

- We usually set C to 1
- Grey levels must be in the range [0.0, 1.0]

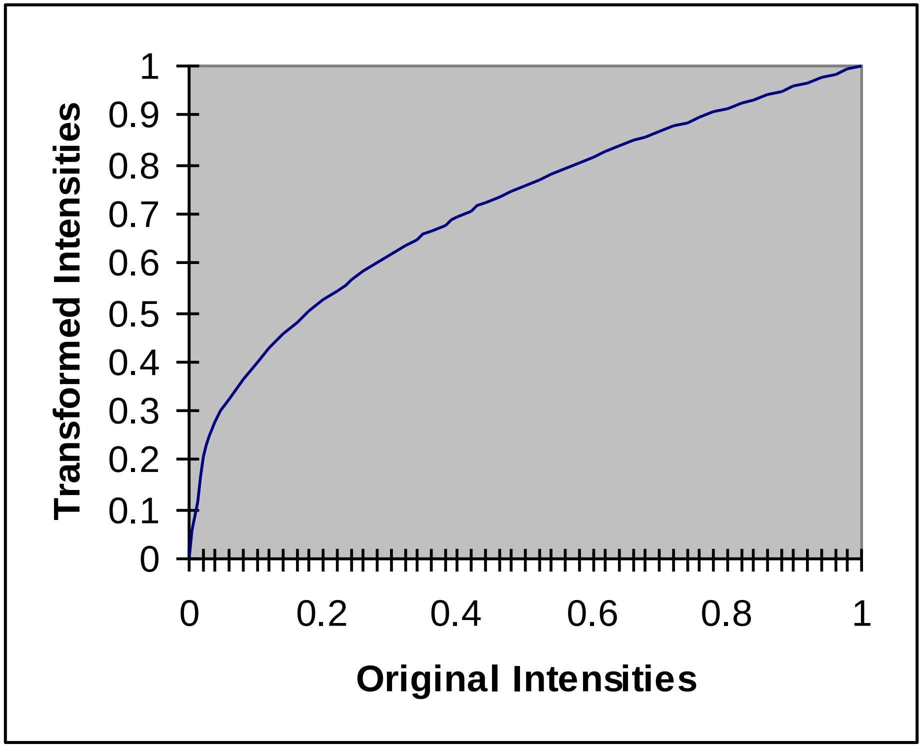
Power Law Example

Power Law Example (cont…)

$\gamma = 0.6$

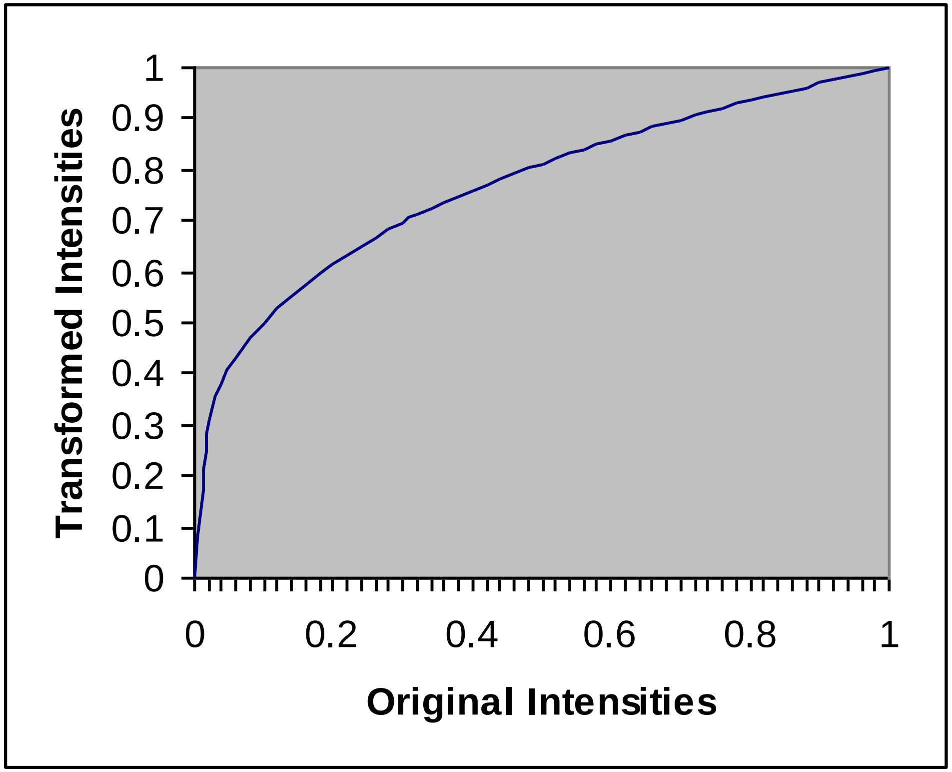
Power Law Example (cont…)

$\gamma = 0.4$



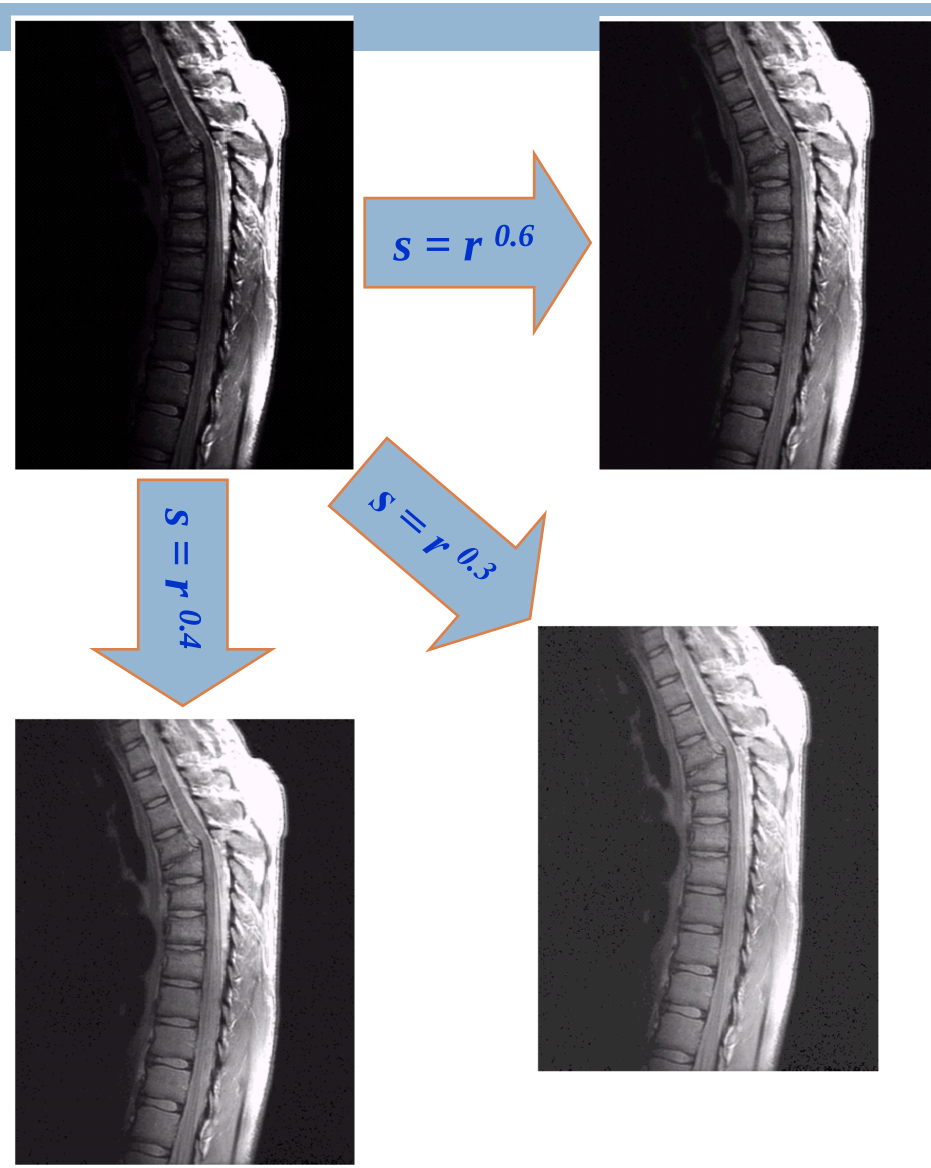
Power Law Example (cont…)

$$\gamma = 0.3$$



Power Law Example (cont…)

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

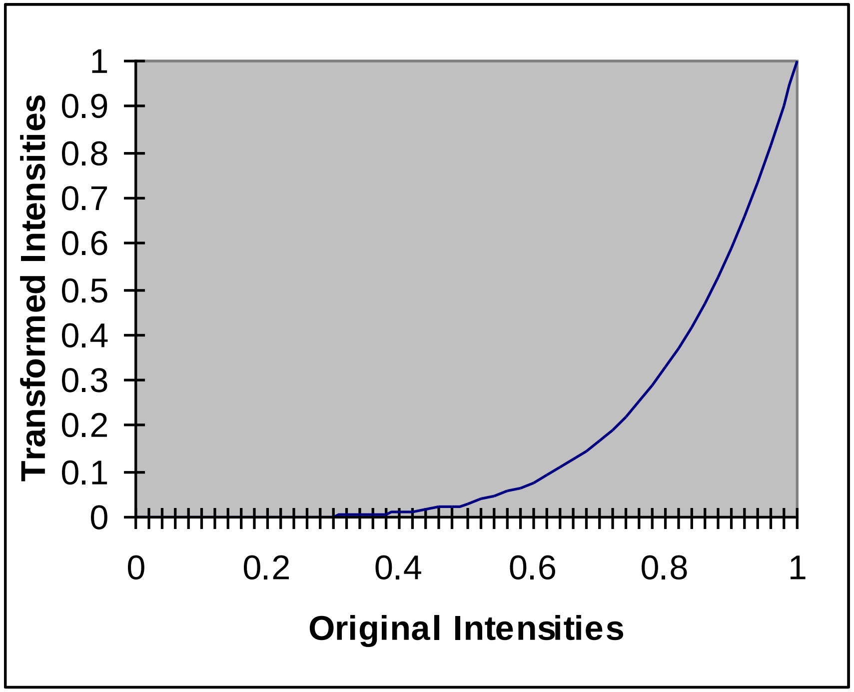


Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Power Law Example

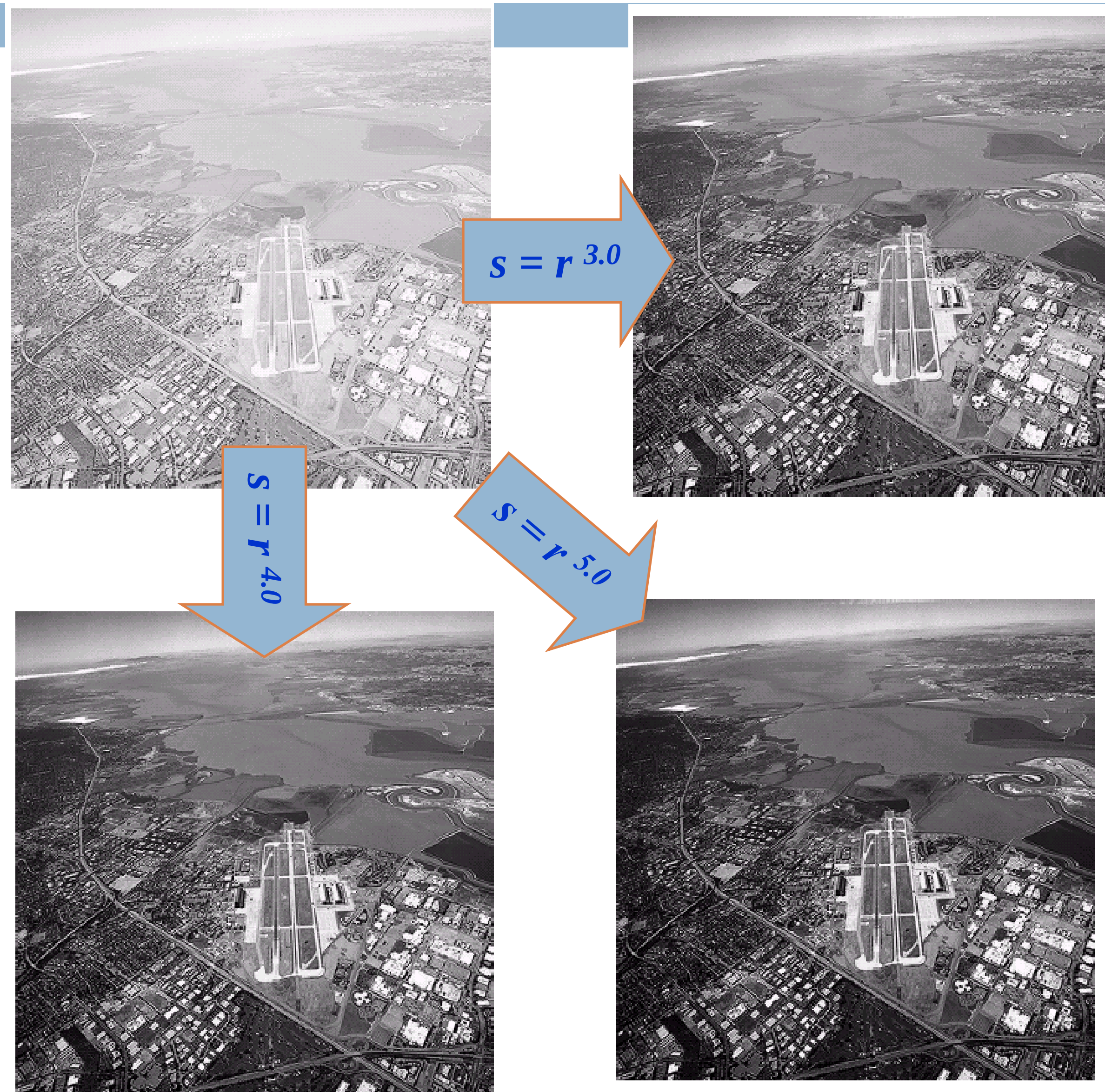
Power Law Example (cont…)

$$\gamma = 5.0$$



Power Law Transformations (cont…)

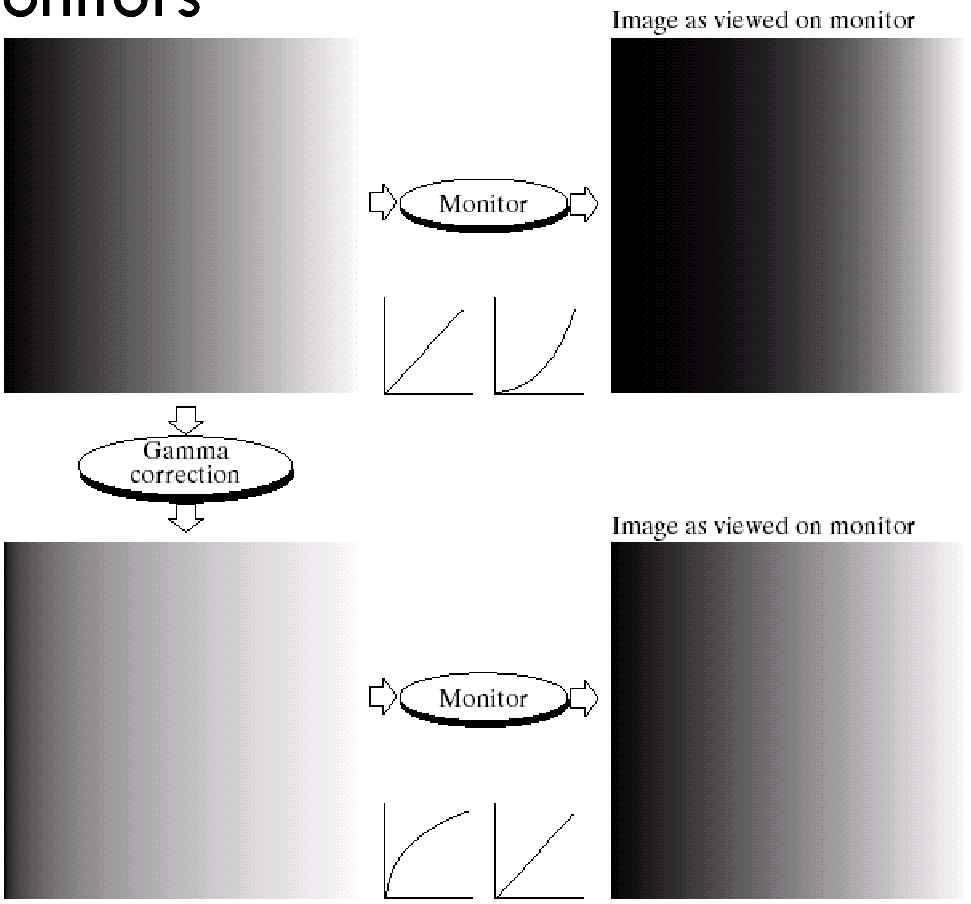
- An aerial photo of a runway is shown
- This time power law transforms are used to darken the image
- Different curves highlight different detail



Images taken from Gonzalez & Woods, Digital Image Processing (2002)

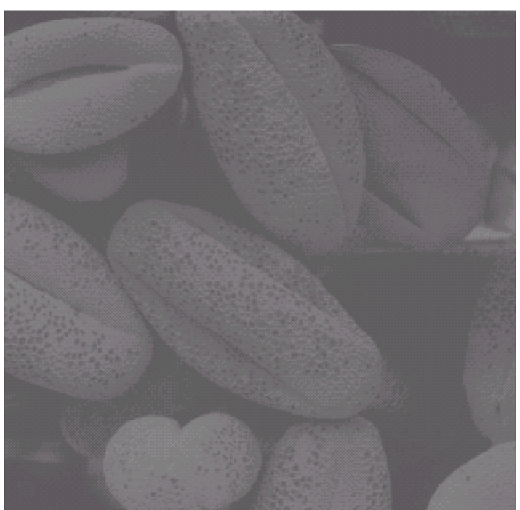
Gamma Correction

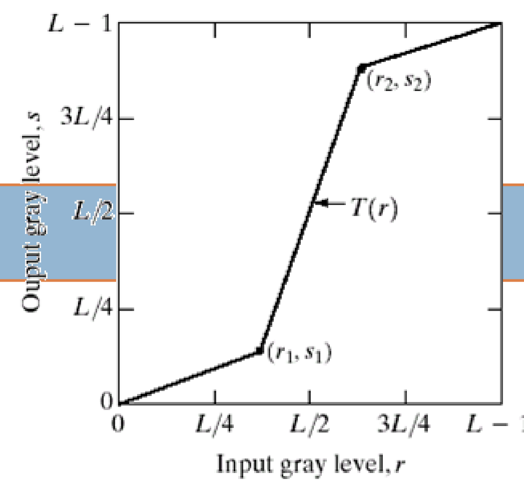
- Many of you might be familiar with gamma correction of computer monitors
- Problem is that display devices do not respond linearly to different intensities
- Can be corrected using a log transform



Piecewise Linear Transformation Functions

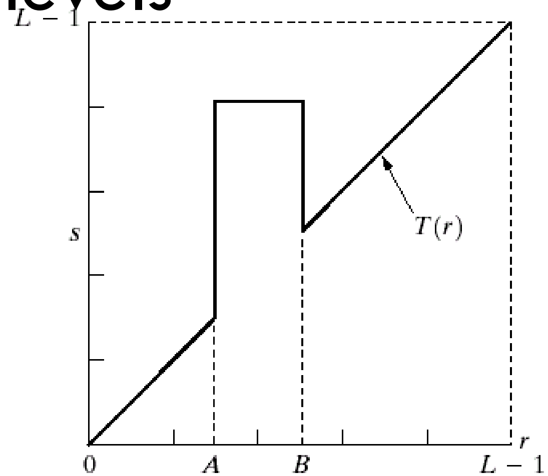
- Rather than using a well defined mathematical function we can use arbitrary user-defined transforms
- The images below show a contrast stretching linear transform to add contrast to a poor quality image

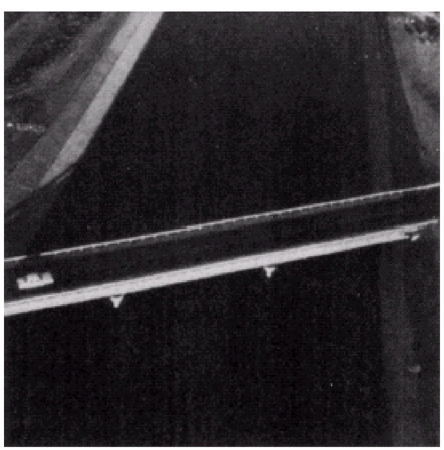


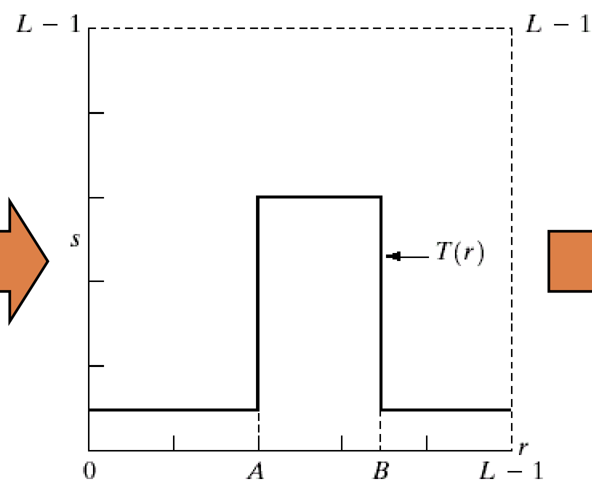


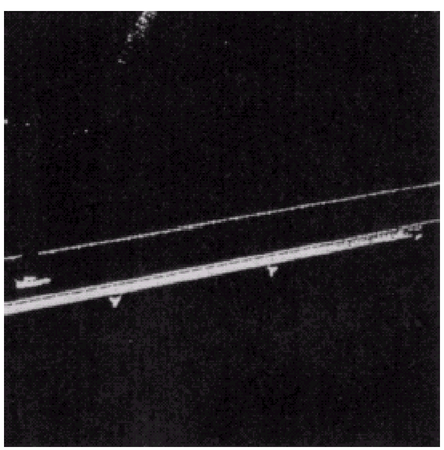
Gray Level Slicing

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



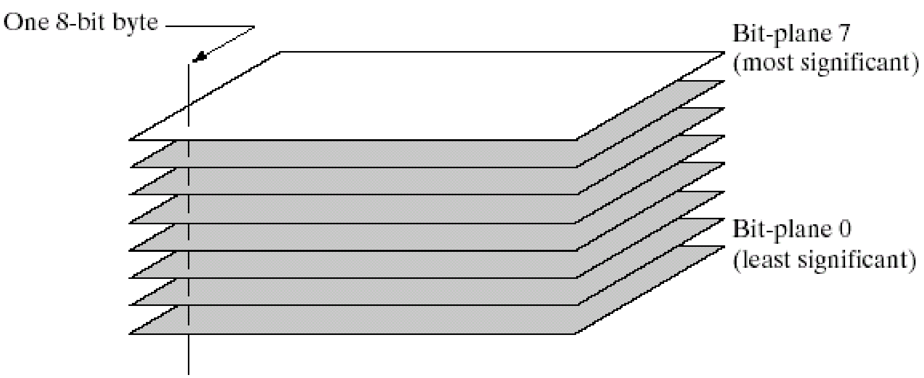


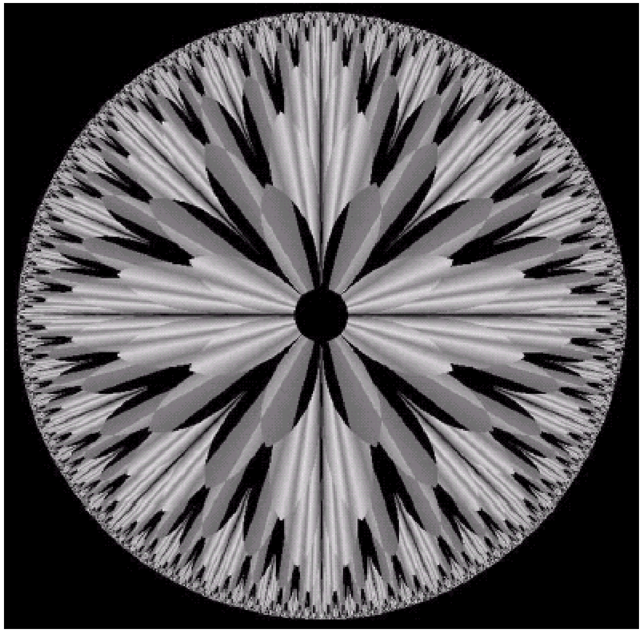




Bit Plane Slicing

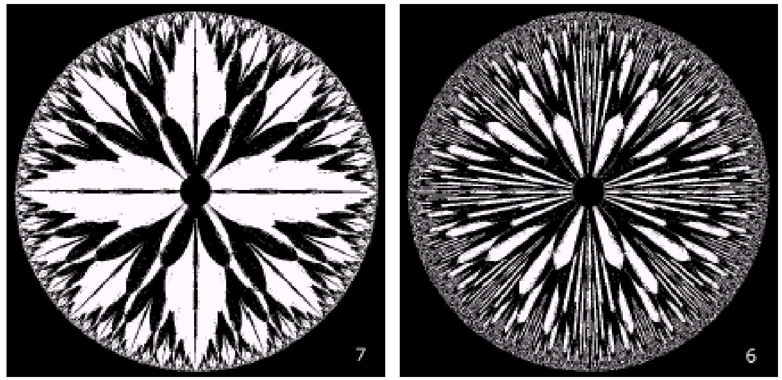
- Often by isolating particular bits of the pixel values in an image we can highlight interesting aspects of that image
  - Higher-order bits usually contain most of the significant visual information
  - Lower-order bits contain subtle details





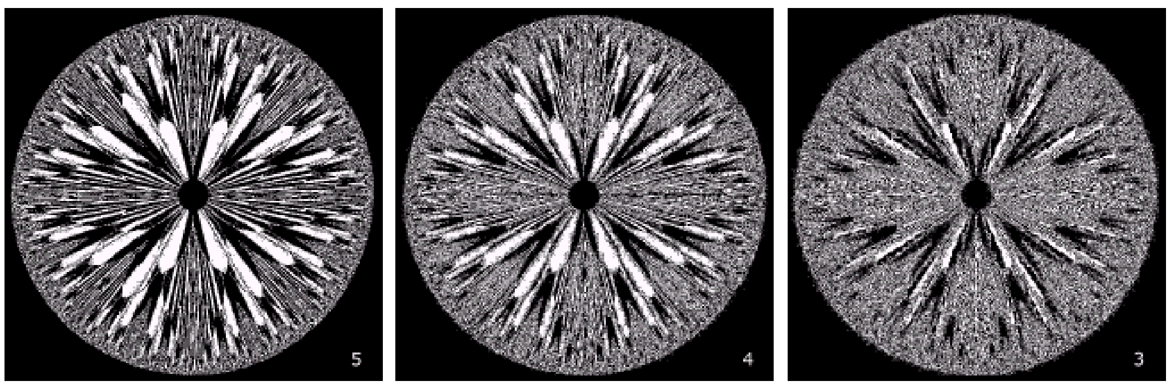
Bit Plane Slicing (cont…)

[10000000]



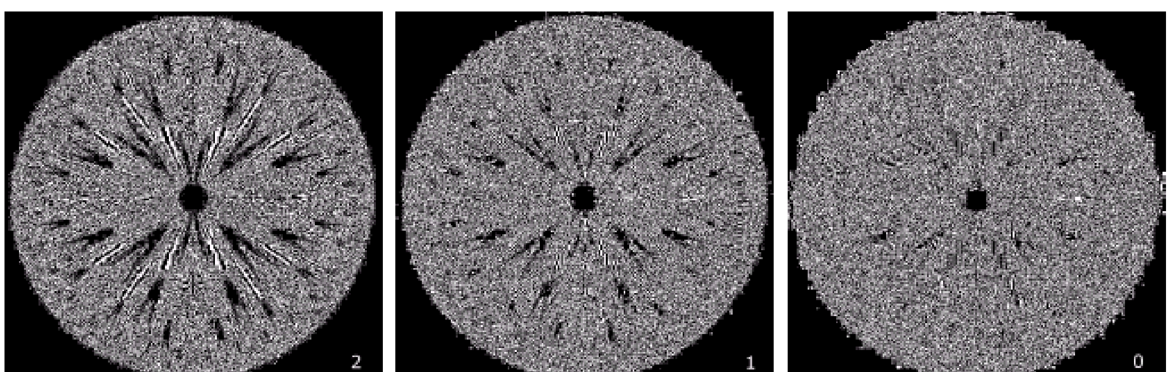
[01000000]

[00100000]



[00001000]

[00000100]

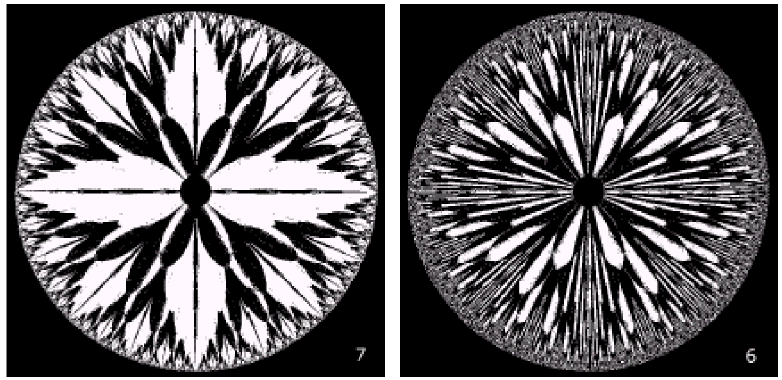


[00000001]

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

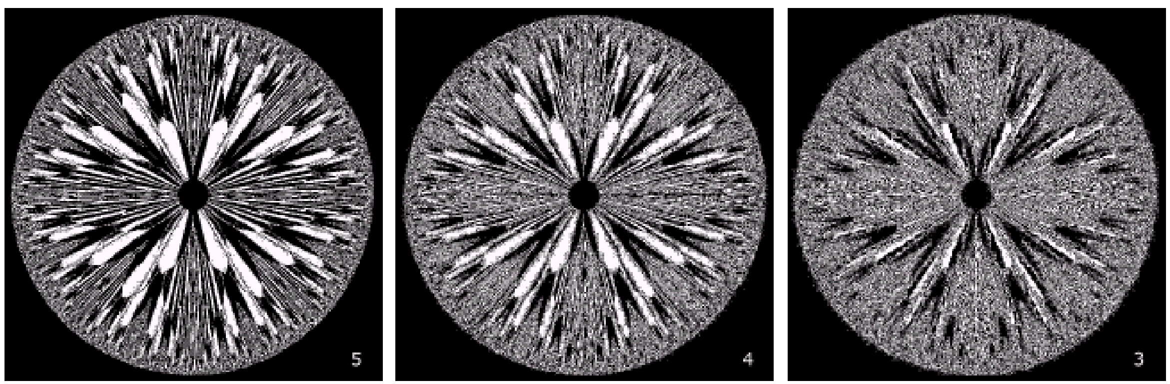
Bit Plane Slicing (cont…)

[10000000]



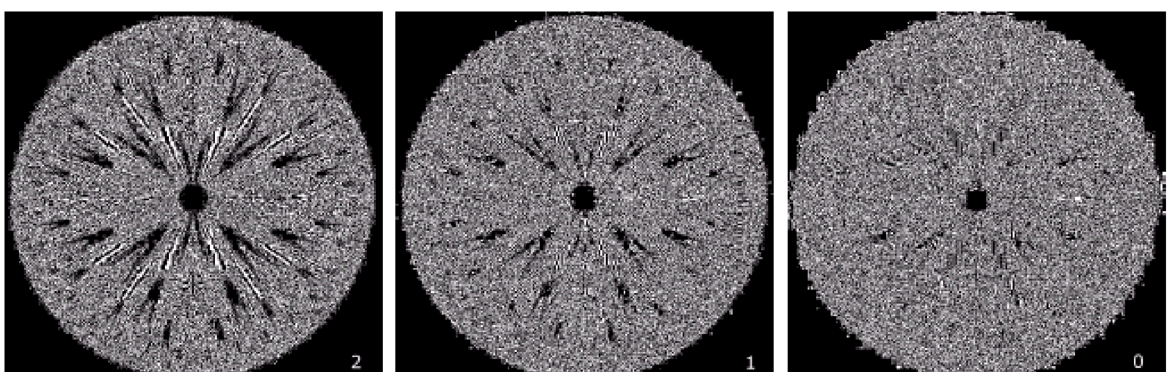
[01000000]

[00100000]



[00001000]

[00000100]



[00000001]

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Bit Plane Slicing (cont…)

a	b	c
d	e	f
g	h	i

FIGURE 3.14 (a) An 8-bit gray-scale image of size 500 × 1192 pixels. (b) through (i) Bit planes 1 through 8, with bit plane 1 corresponding to the least significant bit. Each bit plane is a binary image.

Bit Plane Slicing (cont…)

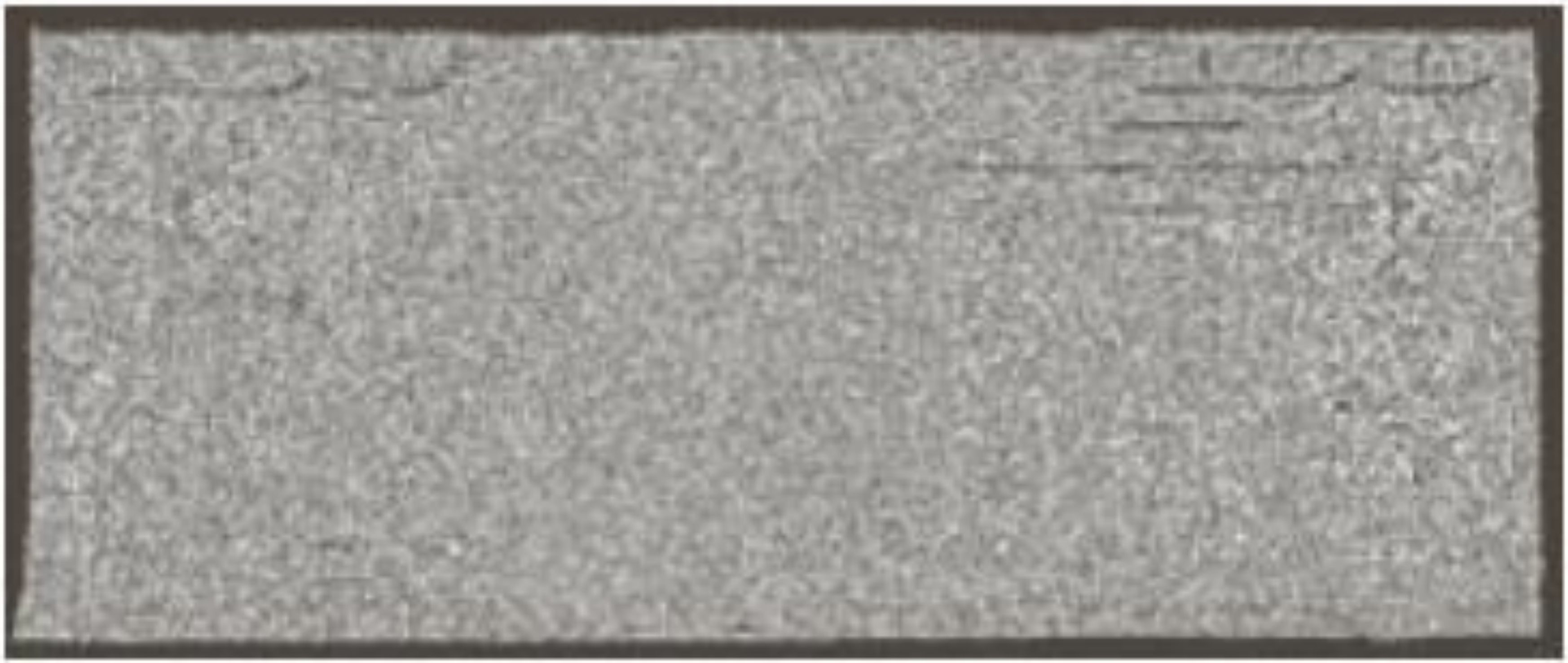
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Bit Plane Slicing (cont…)

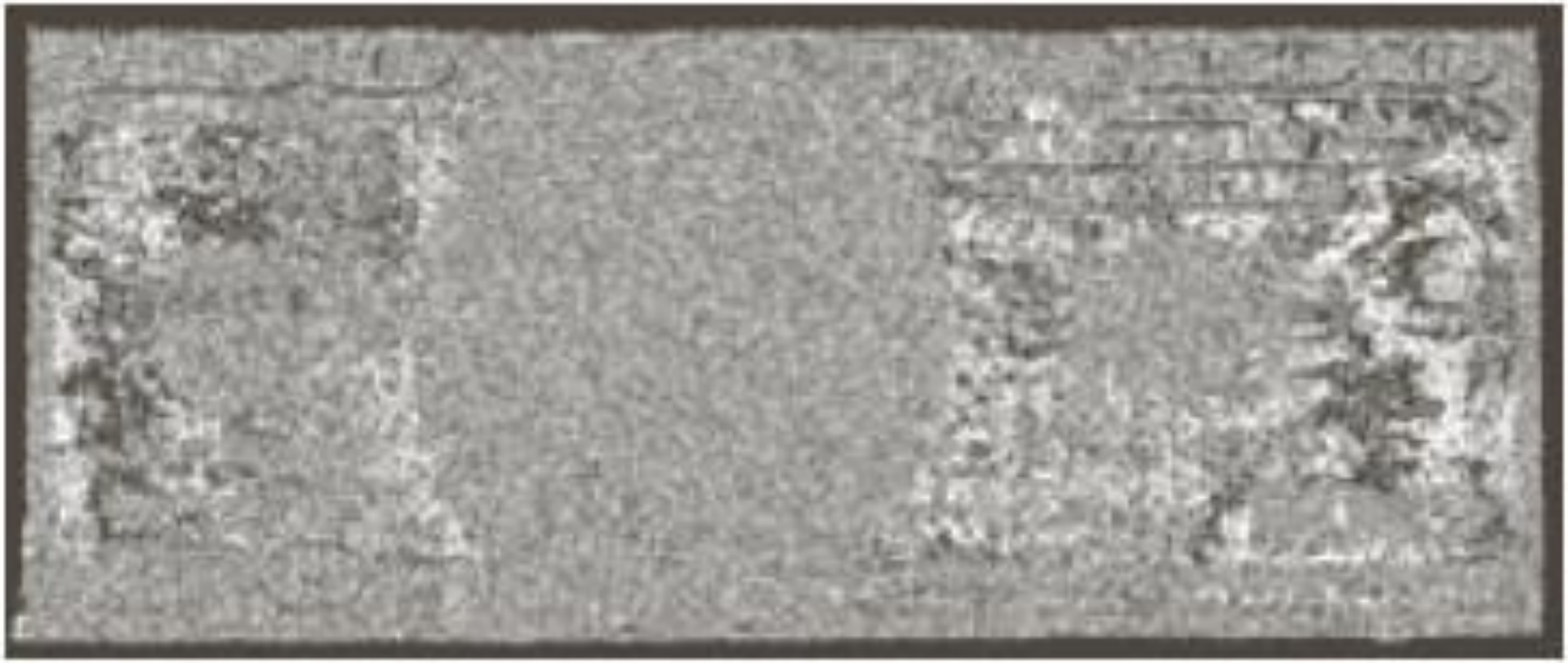
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Bit Plane Slicing (cont…)

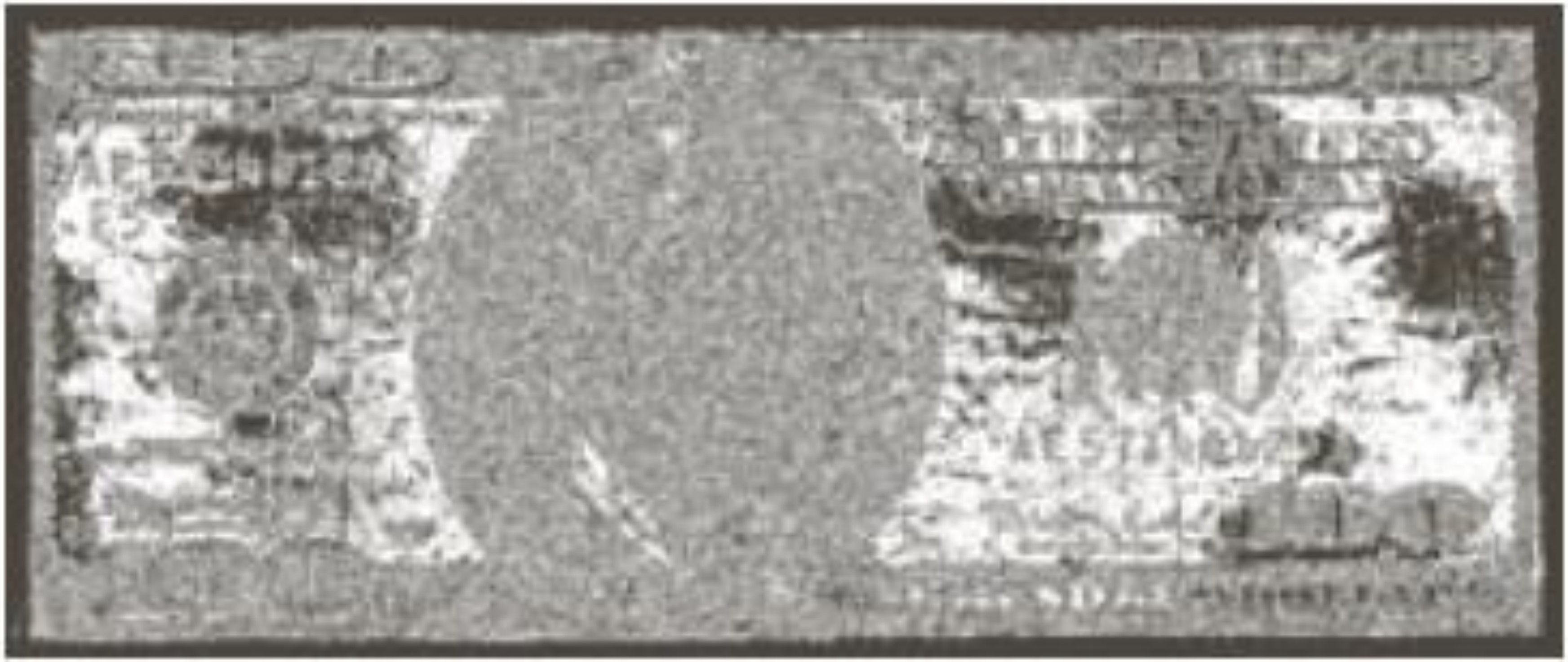
Bit Plane Slicing (cont…)



Bit Plane Slicing (cont…)



Bit Plane Slicing (cont…)



Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Bit Plane Slicing (cont…)

Bit Plane Slicing (cont…)

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Bit Plane Slicing (cont…)

Bit Plane Slicing (cont…)

Reconstructed image using only bit planes 8 and 7

Reconstructed image using only bit planes 8, 7 and 6

Reconstructed image using only bit planes 7, 6 and 5

Histogram Equalization – Example

- Suppose a 3-bit image (L=8) of size 64 x 64 (MN=4096) with intensity distribution shown:

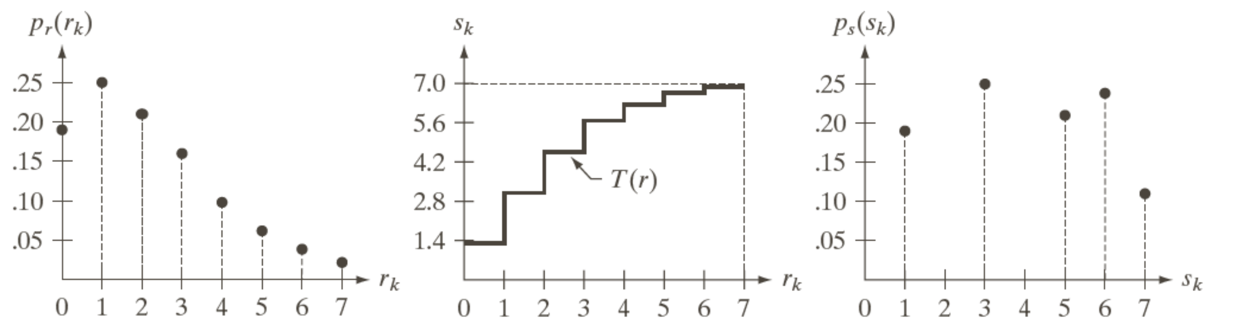
r_k	n_k	$p_r(r_k) = n_k/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02

$$s_0 = T(r_0) = (7)\sum_{j=0}^{0} \frac{n_0}{MN} = 7 p_r(r_0) = 1.33$$

$$s_1 = T(r_1) = (7)\sum_{j=0}^{1} \frac{n_j}{MN} = 7\sum_{j=0}^{1} p_r(r_j)$$

$$= 7 p_r(r_0) + 7 p_r(r_1) = 3.08$$

$$s_0 = 1.33 \rightarrow 1$$
$$s_1 = 3.08 \rightarrow 3$$
$$s_2 = 4.55 \rightarrow 5$$
$$s_3 = 6, s_4 = 6, s_5 = 7, s_6 = 7, s_7 = 7$$



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 Specification

- Histogram *equalization* is an automatic process which produces a uniform histogram of the output image
- Sometimes, it is desired to have the output image follow a certain pre-specified histogram
- The method to generate image that has a specified histogram is called *histogram matching* or *histogram specification*

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k}\frac{n_j}{n} = \frac{(L-1)}{MN}\sum_{j=0}^{k} n_j$$

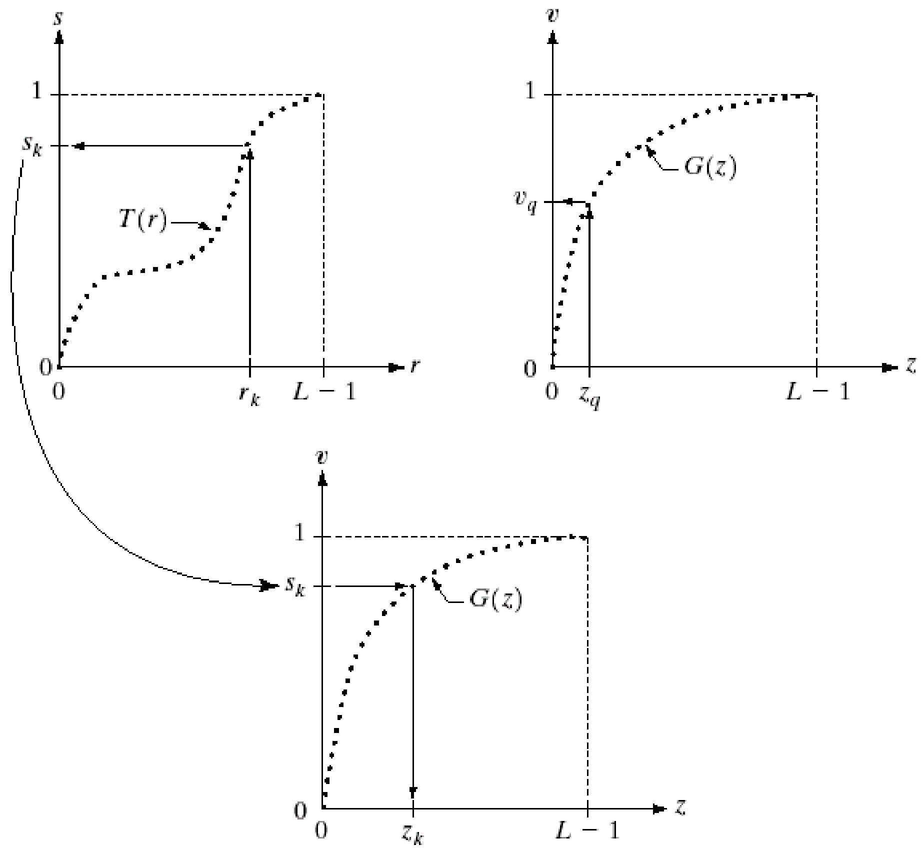
$$G(z_q) = (L-1)\sum_{i=0}^{q} p_z(z_i) = s_k$$

$$z_q = G^{-1}(s_k)$$

Histogram Specification- Steps

1. Compute the histogram $p_r(r)$ of the given image, perform histogram equalization and round the s_k to the integer range [0,L-1]
2. Compute all values of G using equation-5d for q=0, 1, 2,…., L-1 from the specified histogram
3. For every value of s_k use the stored value of G from step-2 to find the corresponding z_q so that $G(z_q)$ is closet to s_k

Histogram Specification- Steps

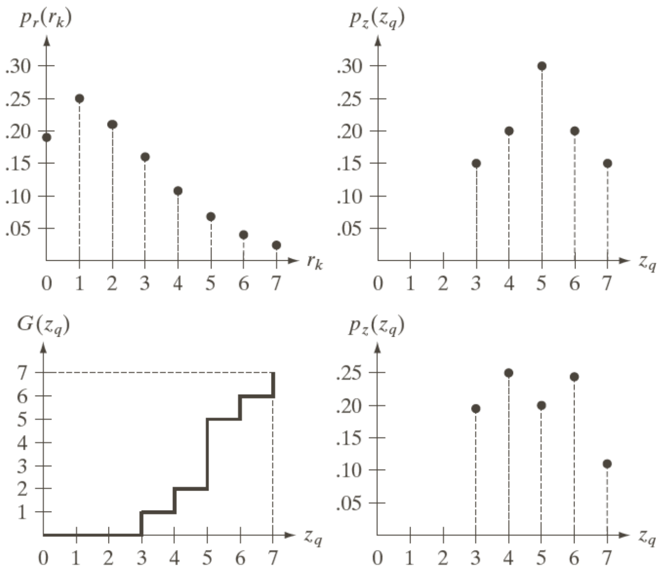


Histogram Specification- Example

Suppose a 3-bit image (L=8) of size 64 x 64 (MN=4096) with intensity distribution shown:

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

z_q	Specified $p_z(z_q)$	Actual $p_z(z_k)$
$z_0 = 0$	0.00	0.00
$z_1 = 1$	0.00	0.00
$z_2 = 2$	0.00	0.00
$z_3 = 3$	0.15	0.19
$z_4 = 4$	0.20	0.25
$z_5 = 5$	0.30	0.21
$z_6 = 6$	0.20	0.24
$z_7 = 7$	0.15	0.11



Step-1

$$s_0 = 1.33 \rightarrow 1$$

$$s_1 = 3.08 \rightarrow 3$$

$$s_2 = 4.55 \rightarrow 5$$

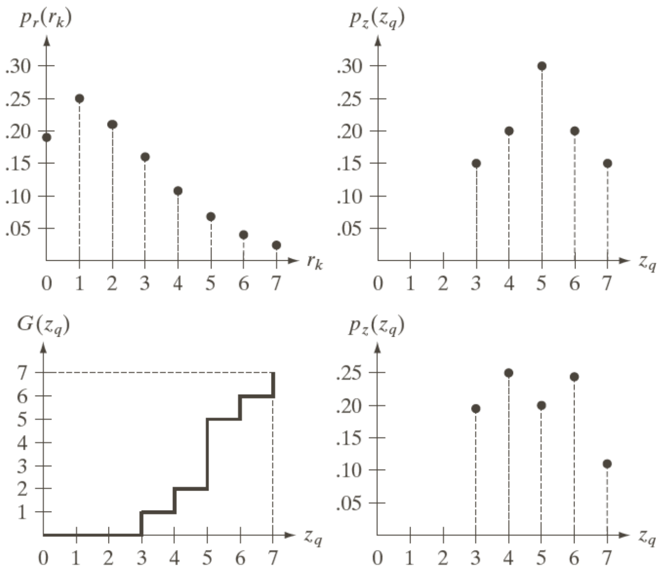
$$s_3 = 6, s_4 = 6, s_5 = 7, s_6 = 7, s_7 = 7$$

Histogram Specification- Example

Suppose a 3-bit image (L=8) of size 64 x 64 (MN=4096) with intensity distribution shown:

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

z_q	Specified $p_z(z_q)$	Actual $p_z(z_k)$
$z_0 = 0$	0.00	0.00
$z_1 = 1$	0.00	0.00
$z_2 = 2$	0.00	0.00
$z_3 = 3$	0.15	0.19
$z_4 = 4$	0.20	0.25
$z_5 = 5$	0.30	0.21
$z_6 = 6$	0.20	0.24
$z_7 = 7$	0.15	0.11



Step-2a

$$G(z_0) = 7\sum_{j=0}^{0} p_z(z_j) = 0,$$

$$G(z_1) = 0, G(z_2) = 0, G(z_3) = 1.05,$$

$$G(z_4) = 2.45, G(z_5) = 4.55,$$

$$G(z_6) = 5.95, G(z_7) = 7.00$$

Step-2b

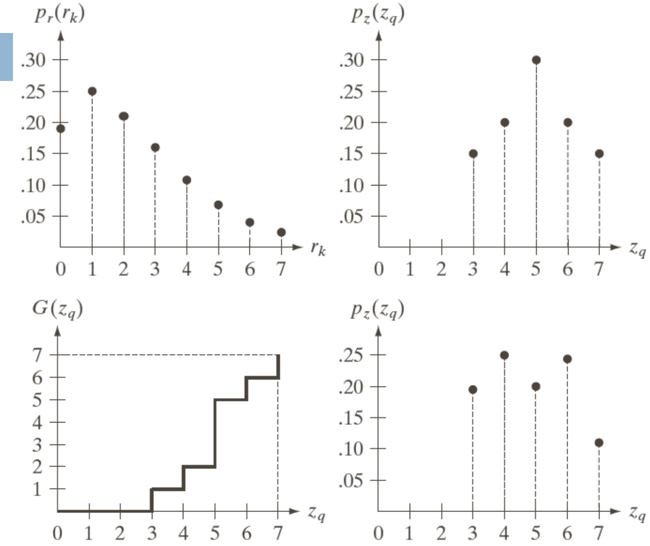
Round the values of G(z)

$G(z_q)$
0
0
0
1
2
5
6
7

Histogram Specification- Example

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

z_q	Specified $p_z(z_q)$	Actual $p_z(z_k)$
$z_0 = 0$	0.00	0.00
$z_1 = 1$	0.00	0.00
$z_2 = 2$	0.00	0.00
$z_3 = 3$	0.15	0.19
$z_4 = 4$	0.20	0.25
$z_5 = 5$	0.30	0.21
$z_6 = 6$	0.20	0.24
$z_7 = 7$	0.15	0.11



Third step

Find the smallest value of z_q so that value of G(z) is the closest to s_k.

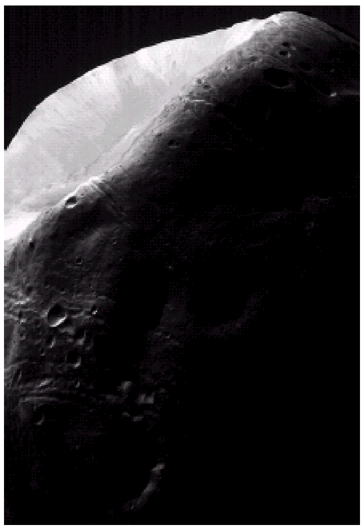
For example, $s_0=1$ and $G(z_3)=1$ i.e. $s_0 \rightarrow z_3$

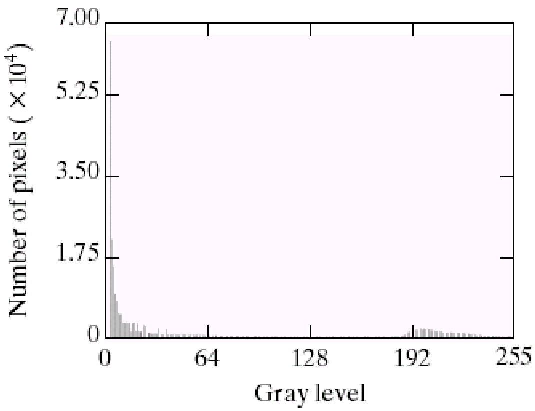
Which means that every pixel whose value=1 in the **histogram equalized** image would map to pixel valued 3 in the **histogram specified** image

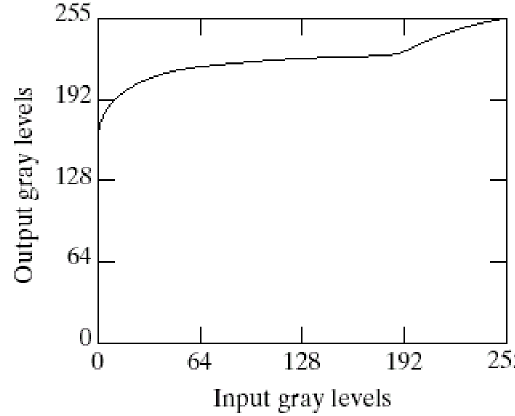
z_q	$G(z_q)$
$z_0 = 0$	0
$z_1 = 1$	0
$z_2 = 2$	0
$z_3 = 3$	1
$z_4 = 4$	2
$z_5 = 5$	5
$z_6 = 6$	6
$z_7 = 7$	7

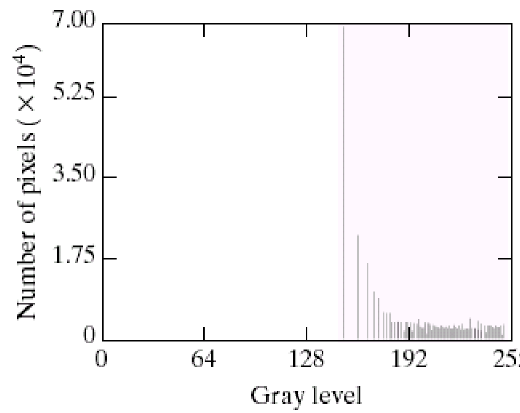
s_k	$\rightarrow$	z_q
1	$\rightarrow$	3
3	$\rightarrow$	4
5	$\rightarrow$	5
6	$\rightarrow$	6
7	$\rightarrow$	7

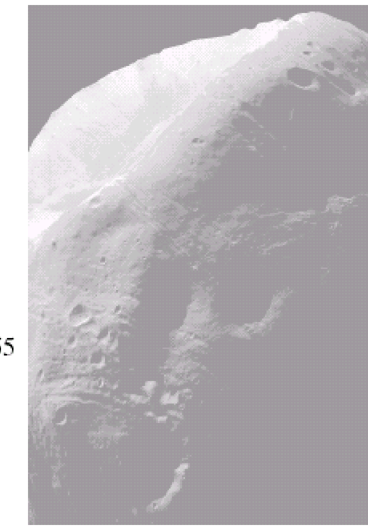
Histogram Matching.



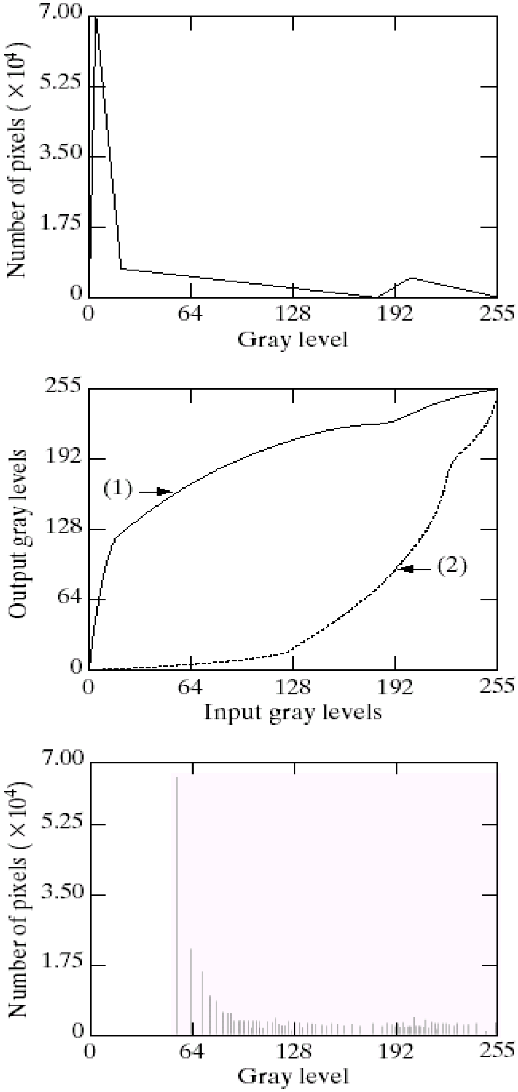








Histogram Matching.



Summary

- We have looked at different kinds of point processing image enhancement
- Next time we will start to look at neighbourhood operations – in particular *filtering* and *convolution*