Digital Image Processing

Lecture # 3A

Image Transformations and Spatial Filtering

Contents

- Image Enhancement
- Types of Enhancement Operations
- Point Processing
  - Linear, Logarithmic & Power Law Transformations
  - Contrast Stretching
  - Gray Level Slicing
  - Bit-Plane Slicing

Image Enhancement

Image Enhancement

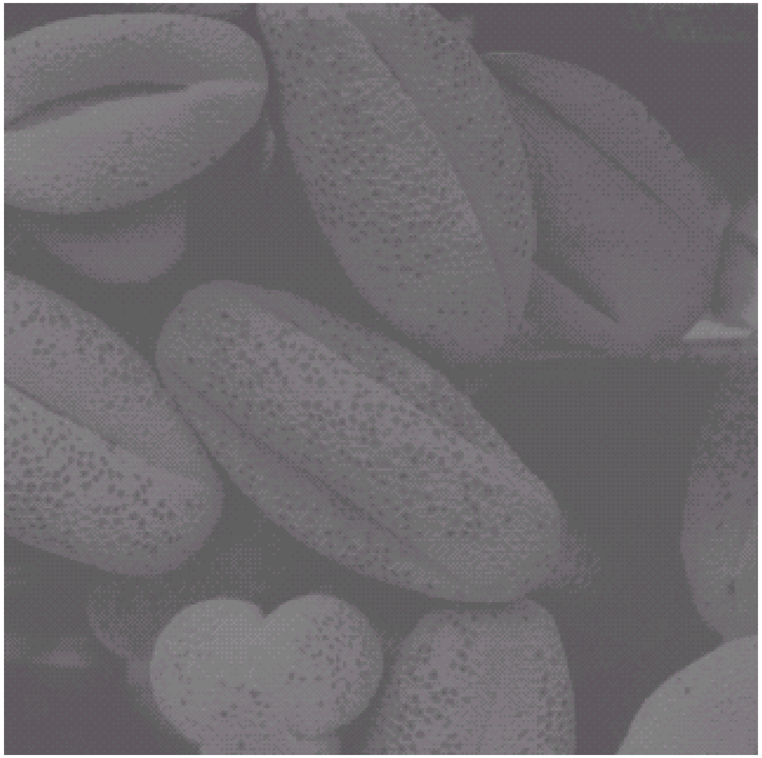


Image Enhancement

Process an image so that the result is more suitable than the original image for a specific application

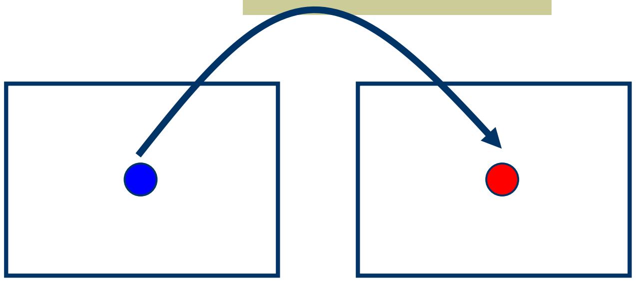
- Image Enhancement Methods
  - Spatial Domain: Direct manipulation of pixels in an image
  - Frequency Domain: Process the image by modifying the Fourier transform of an image

This Chapter – Spatial Domain

Types of image enhancement operations

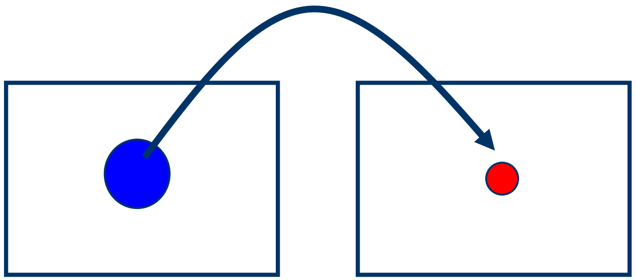
- ## Point/Pixel operations

Output value at specific coordinates *(x,y)* is dependent only on the input value at *(x,y)*



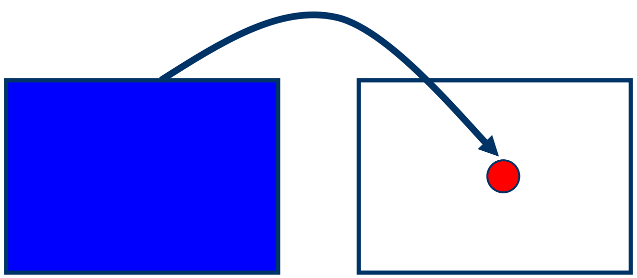
- ## Local operations

The output value at *(x,y)* is dependent on the input values in the neighborhood of *(x,y)*



- ## Global operations

The output value at *(x,y)* is dependent on all the values in the input image



Basic Concepts

- Most spatial domain enhancement operations can be generalized as:

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

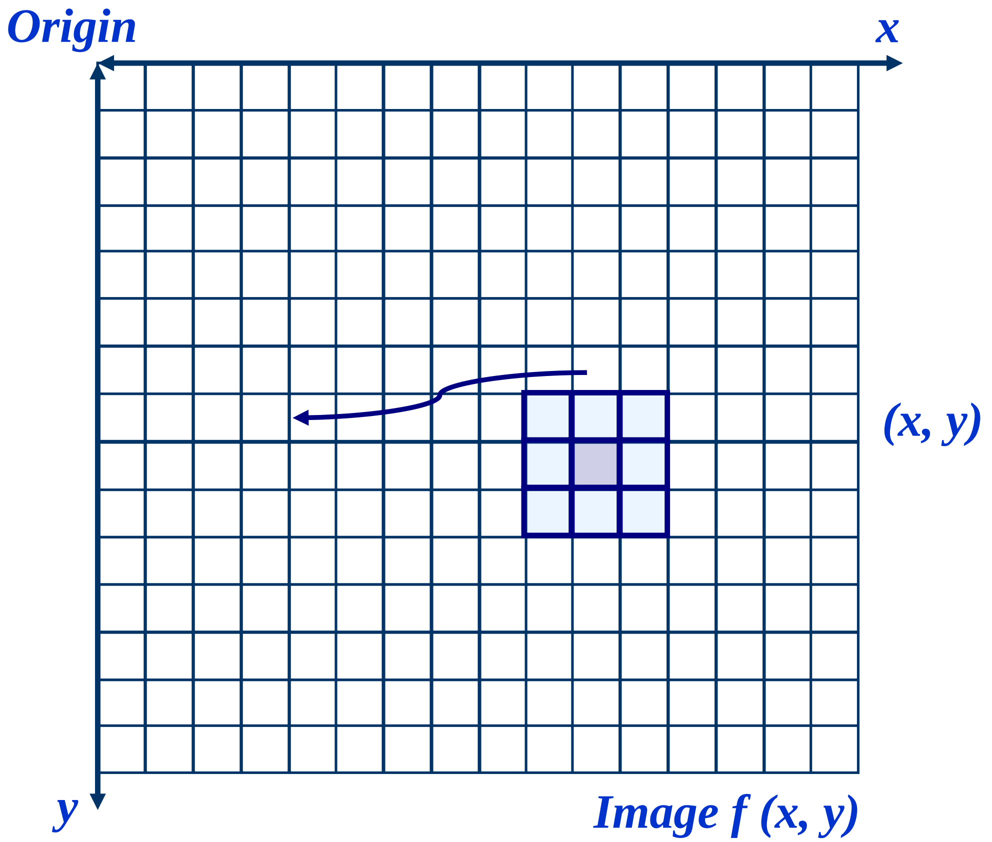
$f\ (x, y)$ = the input image

$g\ (x, y)$ = the processed/output image

T = some operator defined over some neighbourhood of (x, y)

Basic Concepts

A square or rectangular sub-image area centered at (x,y)



Point Processing

- In a digital image, point = pixel
- Point processing transforms a pixel's value as function of its value alone;
- It does not depend on the values of the pixel's neighbors.

Point Processing

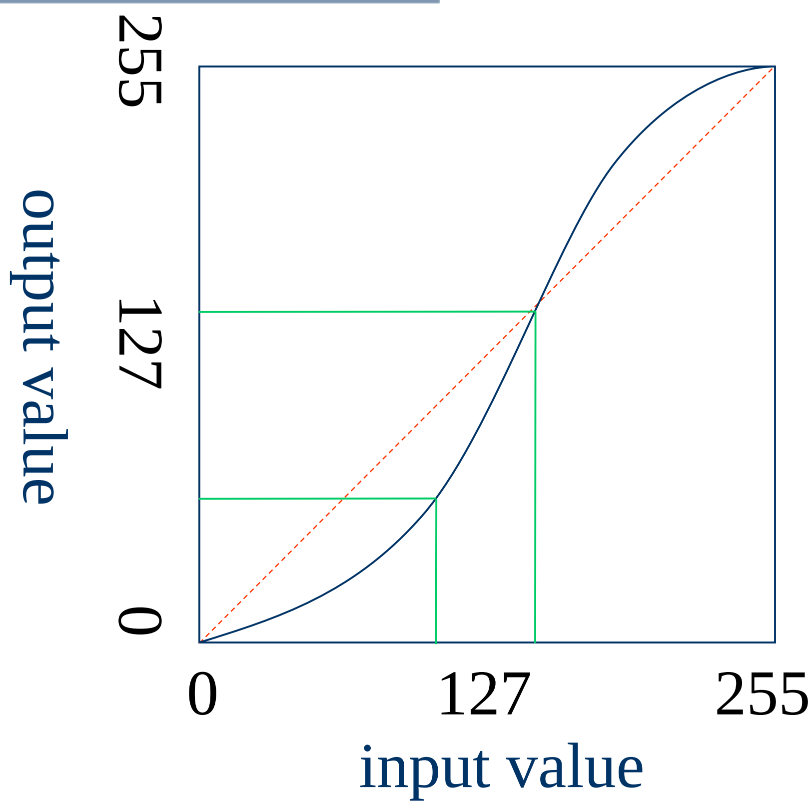
- Neighborhood of size 1x1:
- *g* depends only on *f* at *(x,y)*
- *T*: Gray-level/intensity transformation/ mapping function

$$s = T(r)$$

  - r = gray level of *f* at *(x,y)*
  - s = gray level of *g* at *(x,y)*

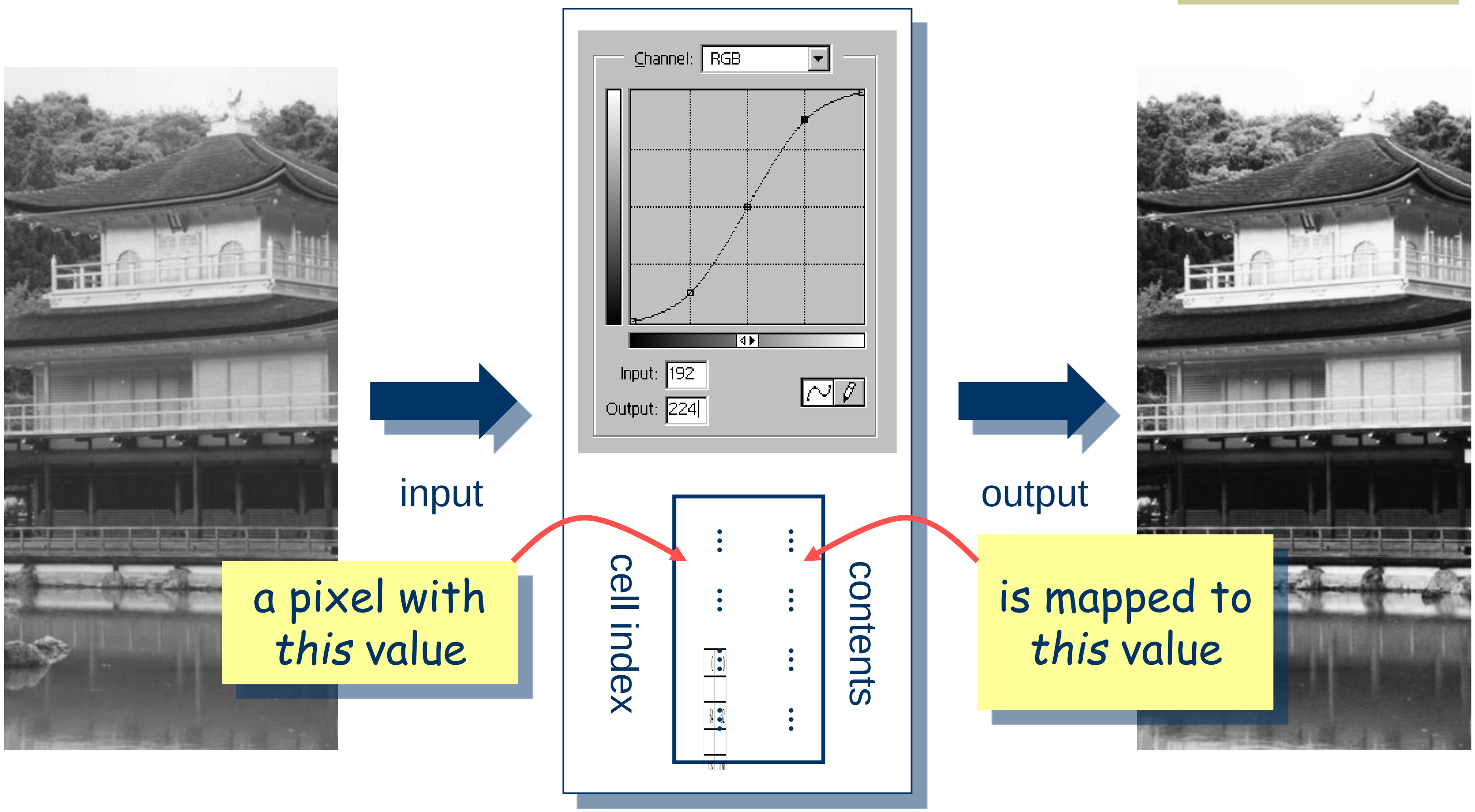
Point Processing using Look-up Tables

A look-up table (LUT) implements a functional mapping.



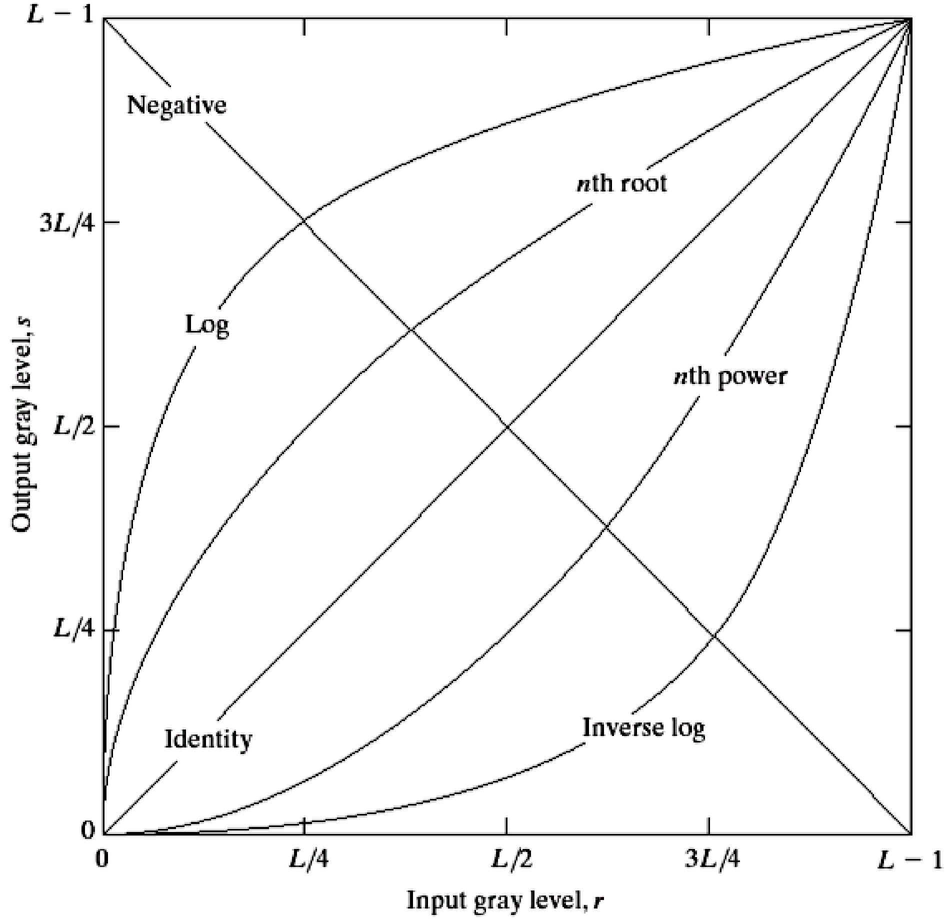
E.g.:	index	value
	...	...
	101	64
	102	68
	103	69
	104	70
	105	70
	106	71
	...	...
	input	output

Point Processing using Look-up Tables



Point Processing 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



Point Processing Example: Negative Images

- Reverses the gray level order
- For L gray levels, the transformation has the form:

$$s = (L - 1) - r$$

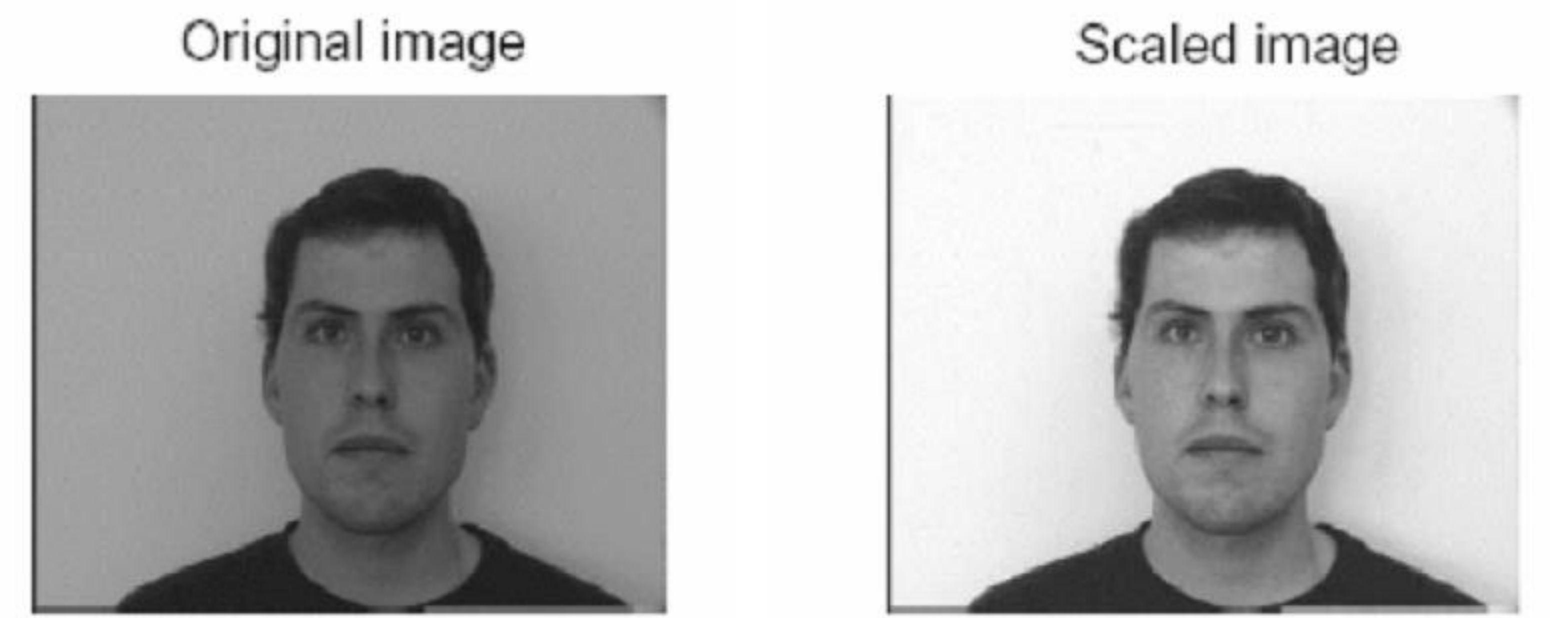
- Negative images are useful for enhancing white or grey detail embedded in dark regions of an image

Point Processing Example: Negative Images

Input image (X-ray image) **Output image (negative)**

Point Processing Example: Intensity Scaling

$$s = T(r) = ar$$

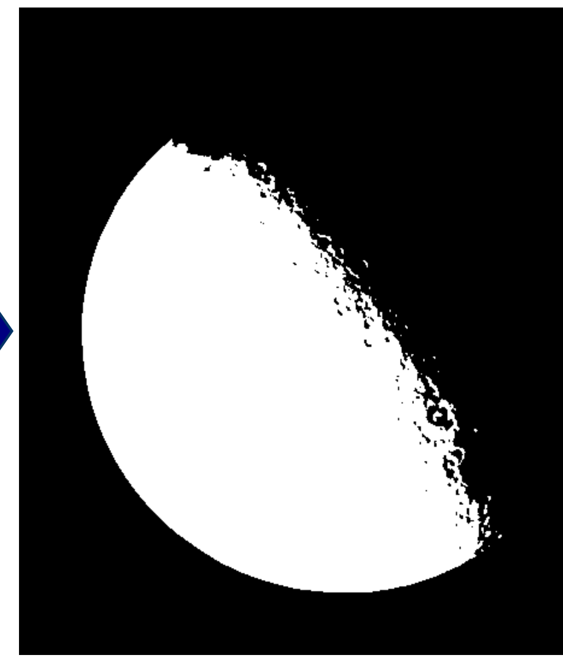


$f(x, y)$ $\qquad$ $a \cdot f(x, y)$

Point Processing Example: Thresholding

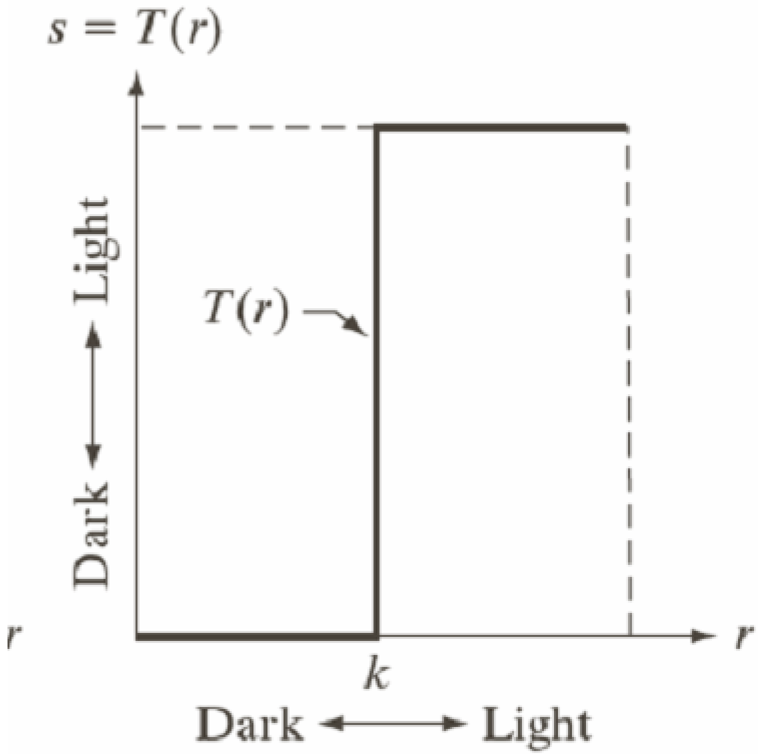
- Segmentation of an object of interest from a background

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



Point Processing Example: Thresholding

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



Logarithmic Transformations

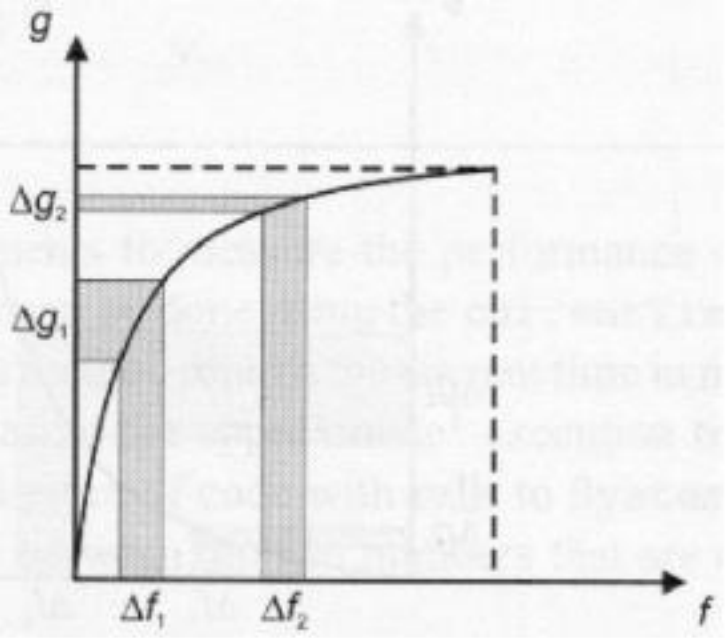
- The general form of the log transformation is

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

- 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

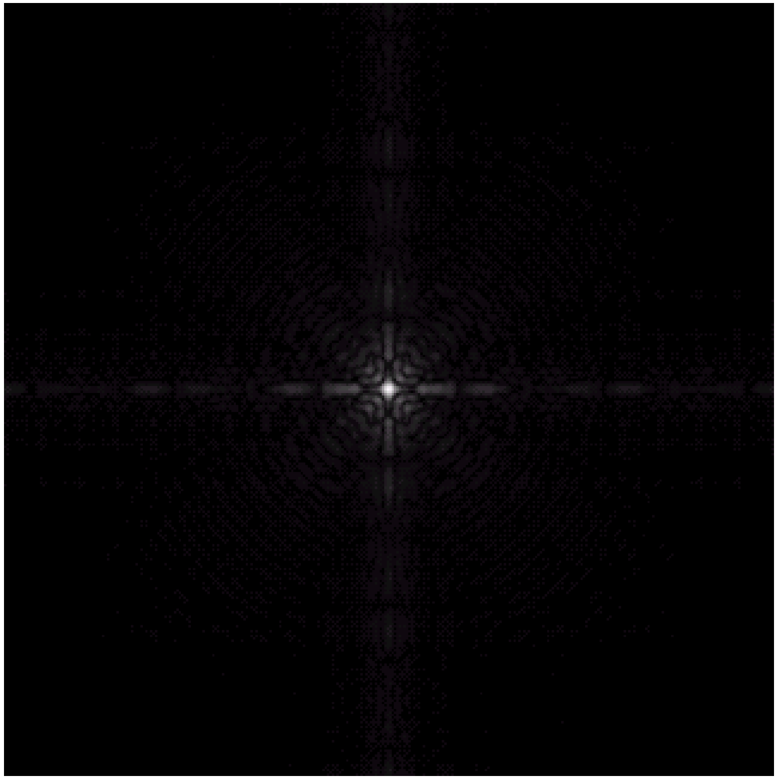
- **Properties**
  - For lower amplitudes of input image the range of gray levels is expanded
  - For higher amplitudes of input image the range of gray levels is compressed



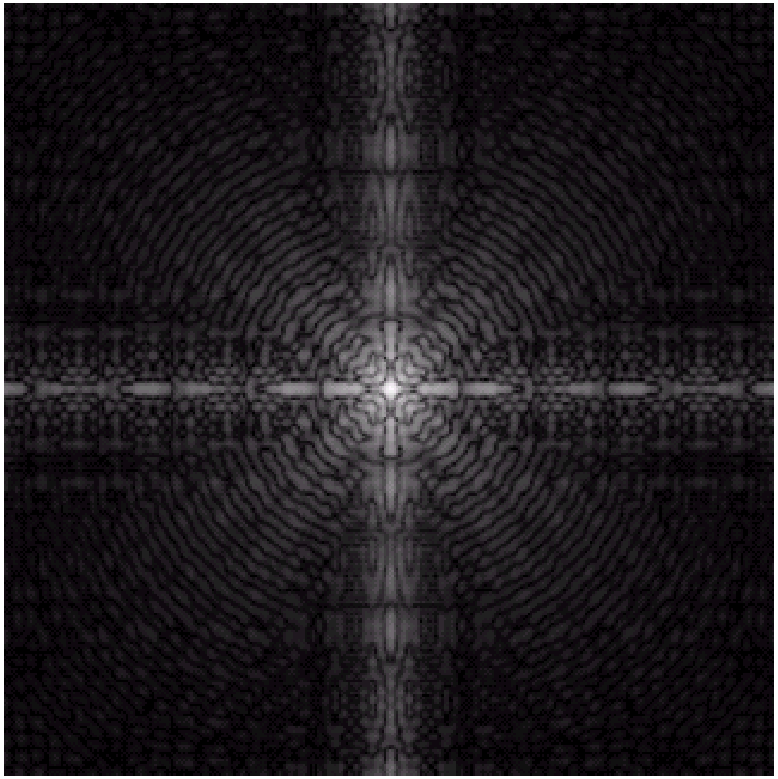
Logarithmic Transformations

- **Application**
  - This transformation is suitable for the case when the dynamic range of a processed image far exceeds the capability of the display device (e.g. display of the Fourier spectrum of an image)
  - Also called "dynamic-range compression / expansion"

Logarithmic Transformations



Fourier spectrum: image values ranging from 0 to $1.5x10^6$



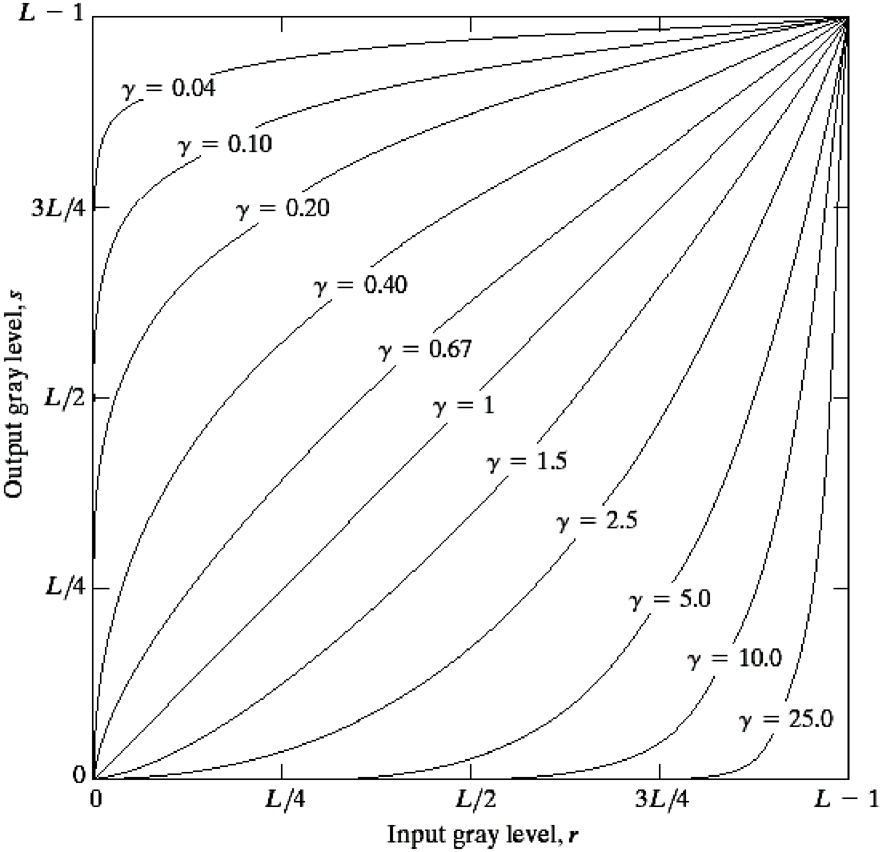
The result of log transformation with c = 1

Power Law Transformations

- Power law transformations have the following form

- 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

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



Power Law Transformations

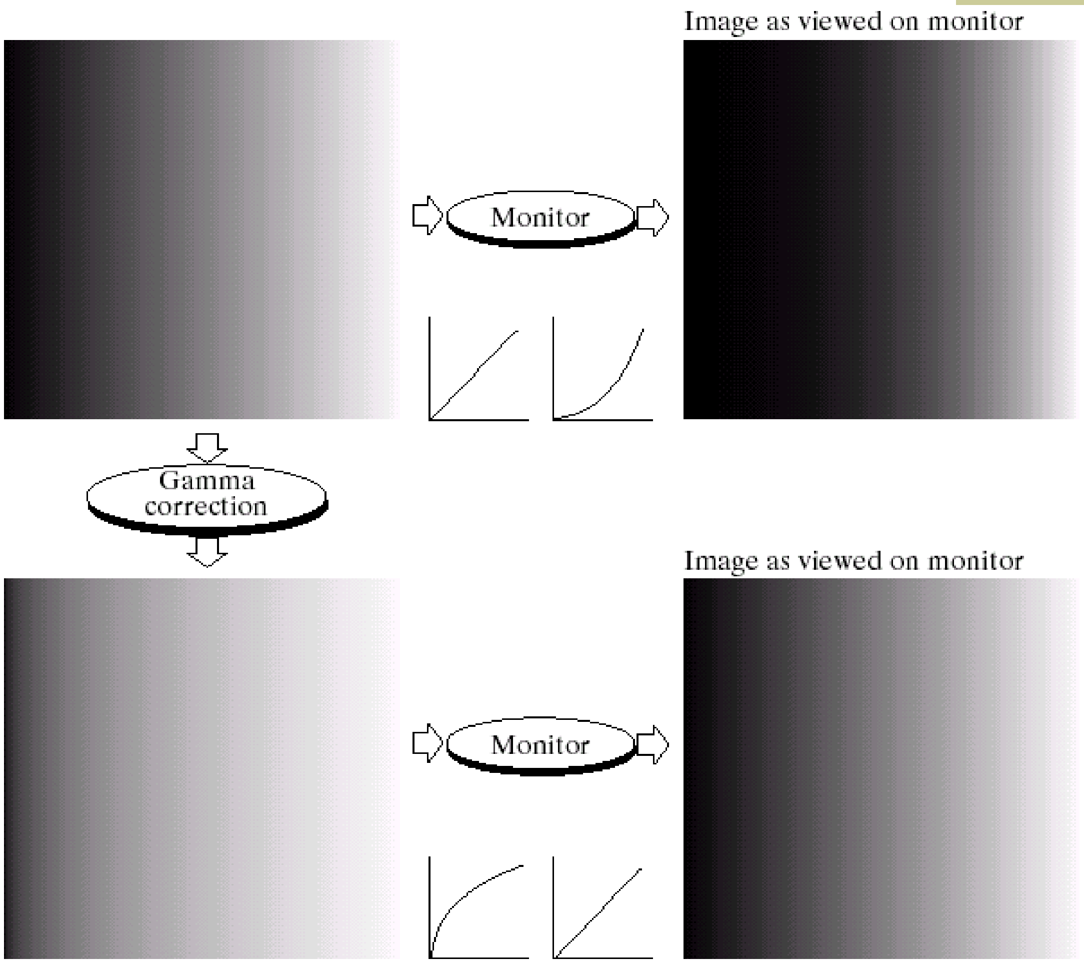
- For $\gamma < 1$: Expands values of dark pixels, compress values of brighter pixels
- For $\gamma > 1$: Compresses values of dark pixels, expand values of brighter pixels
- If $\gamma=1$ & $c=1$: Identity transformation ($s = r$)

- A variety of devices (image capture, printing, display) respond according to a power law and need to be corrected

- **Gamma (γ) correction**

  The process used to correct the power-law response phenomena

Power Law Transformations: Gamma Correction



a b
c d

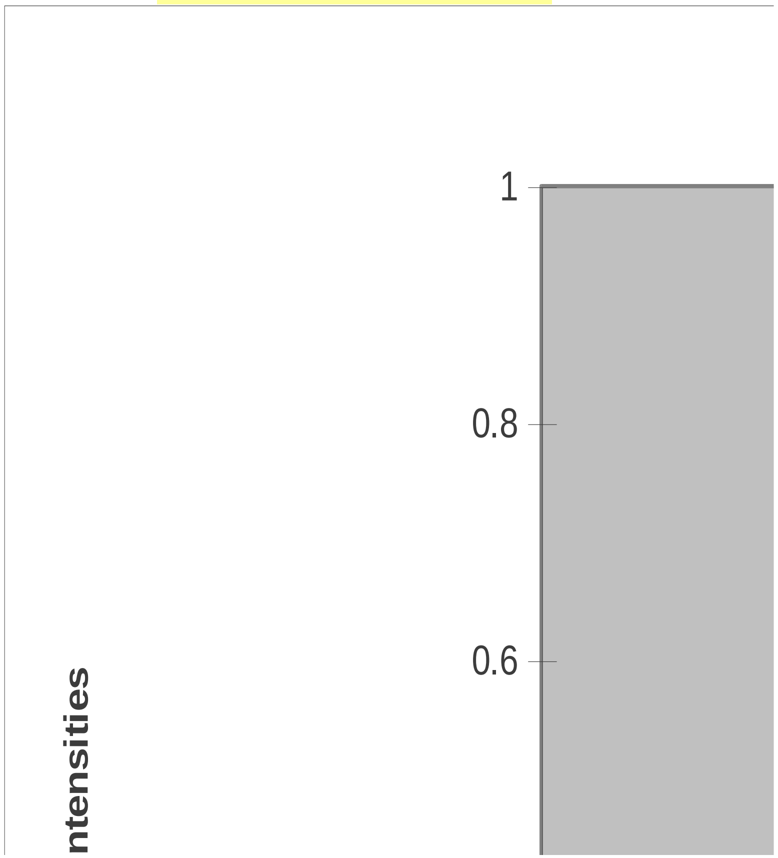
FIGURE 3.7
(a) Linear-wedge gray-scale image.
(b) Response of monitor to linear wedge.
(c) Gamma-corrected wedge.
(d) Output of monitor.

Power Law Transformations Contrast Enhancement

The images to the right show a magnetic resonance (MR) image of a fractured human spine

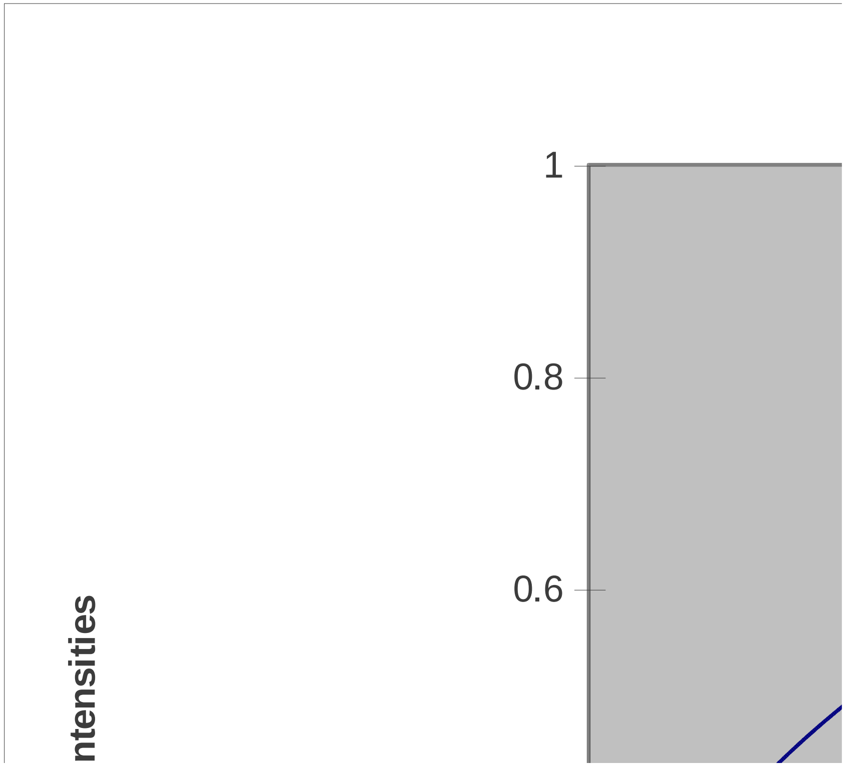
Power Law Transformations
Contrast Enhancement

$\gamma = 0.6$



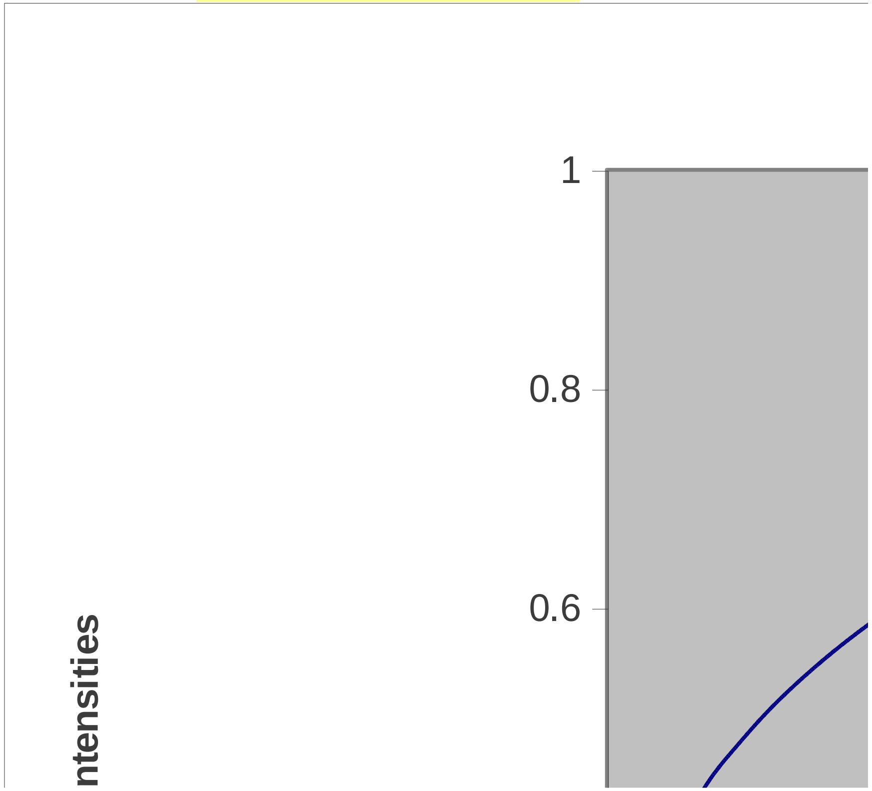
Power Law Transformations Contrast Enhancement

$\gamma = 0.4$



Power Law Transformations Contrast Enhancement

$\gamma = 0.3$



Power Law Transformations Contrast Enhancement

MR image of fractured human spine

Result after Power law transformation

$c = 1, \gamma = 0.6$

Result after Power law transformation

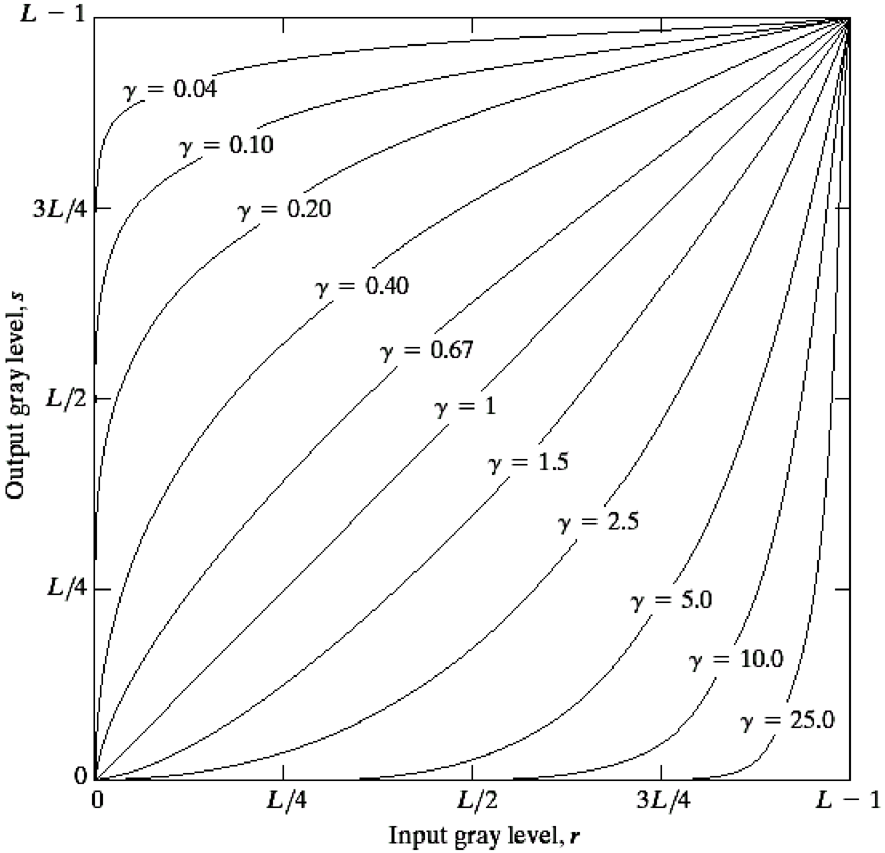
$c = 1, \gamma = 0.4$

Result after Power law transformation

$c = 1, \gamma = 0.3$

Power Law Transformations Contrast Enhancement

When the γ is reduced too much, the image begins to reduce contrast to the point where the image started to have very slight "wash-out" look.

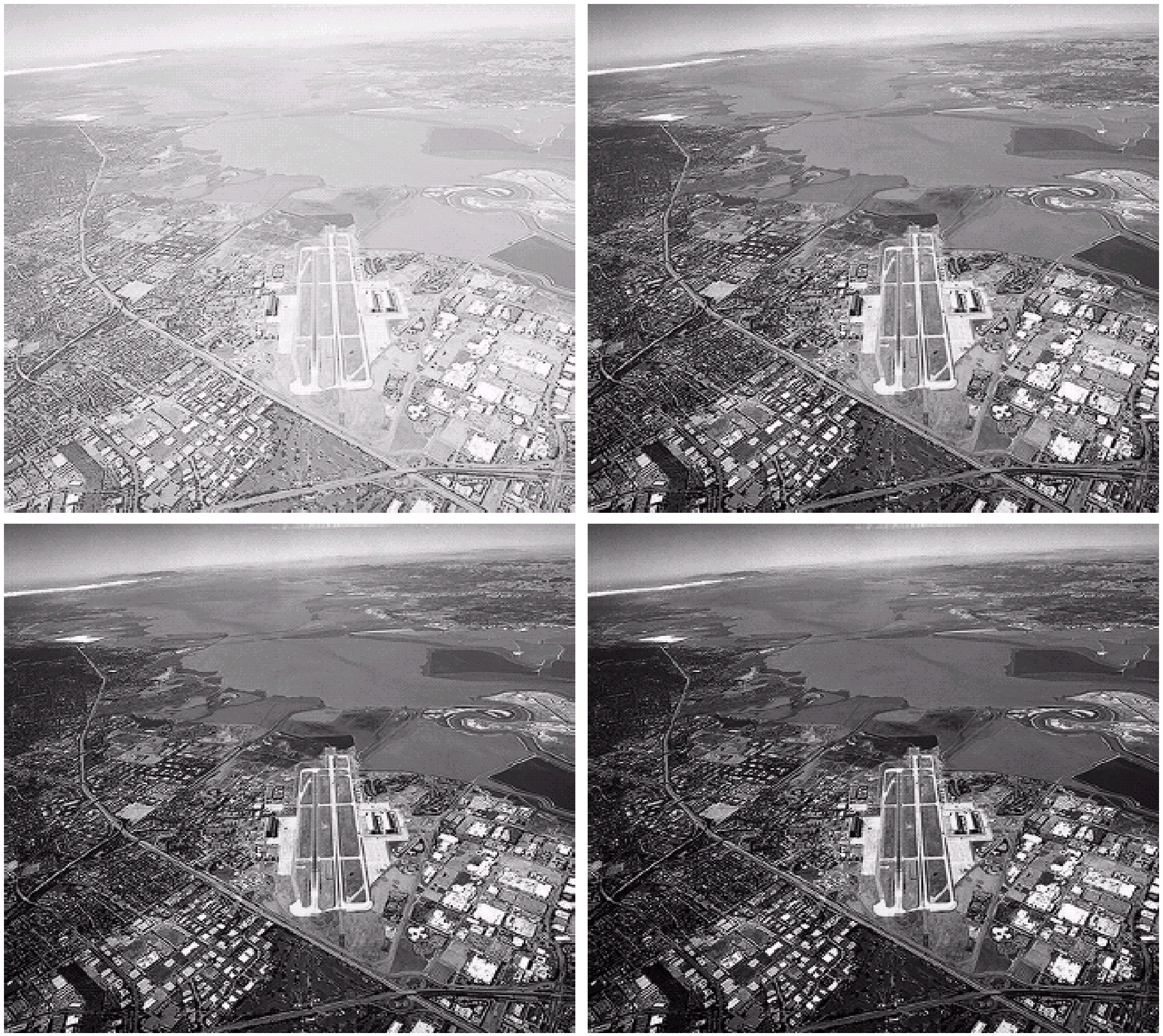


Power Law Transformations Contrast Enhancement

Image has a washed-out appearance – needs $\gamma > 1$

Image Enhancement

Aerial Image



Result of Power law transformation c = 1, γ = 3.0 (suitable)

Result of Power law transformation c = 1, γ = 4.0 (suitable)

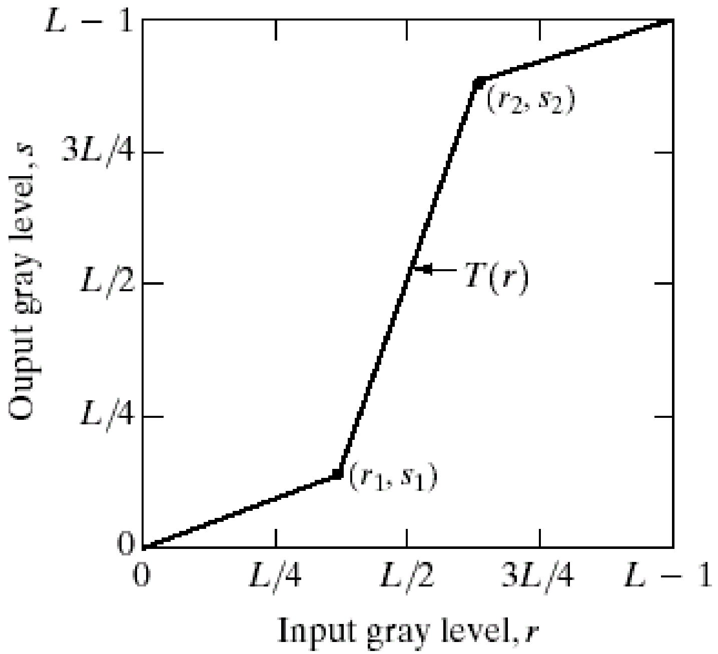
Result of Power law transformation c = 1, γ = 5.0 (high contrast, some regions are too dark)

Piecewise Linear Transformation Functions

- Contrast stretching
- Intensity level slicing
- Bit-Plane slicing

Contrast Stretching

- Objective
  - Increase the dynamic range of the gray levels for low contrast images
- Rather than using a well defined mathematical function we can use arbitrary user-defined transforms

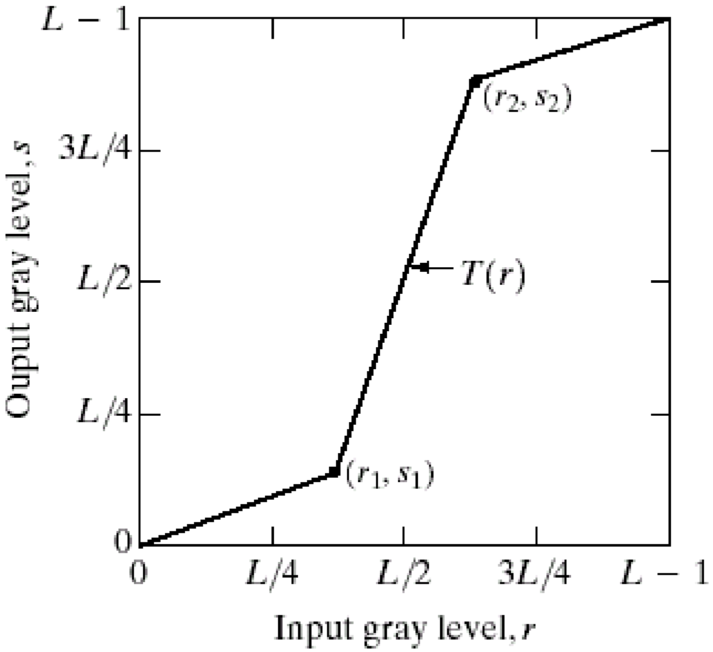


- If $r_1 = s_1$ & $r_2 = s_2$, no change in gray levels
- If $r_1 = r_2$, $s_1 = 0$ & $s_2 = L-1$, then it is a threshold function. The resulting image is binary

Contrast Stretching

$r_1 = r_{min}$ & $s_1 = 0$

$r_2 = r_{max}$ & $s_2 = L\text{-}1$



Contrast Stretching

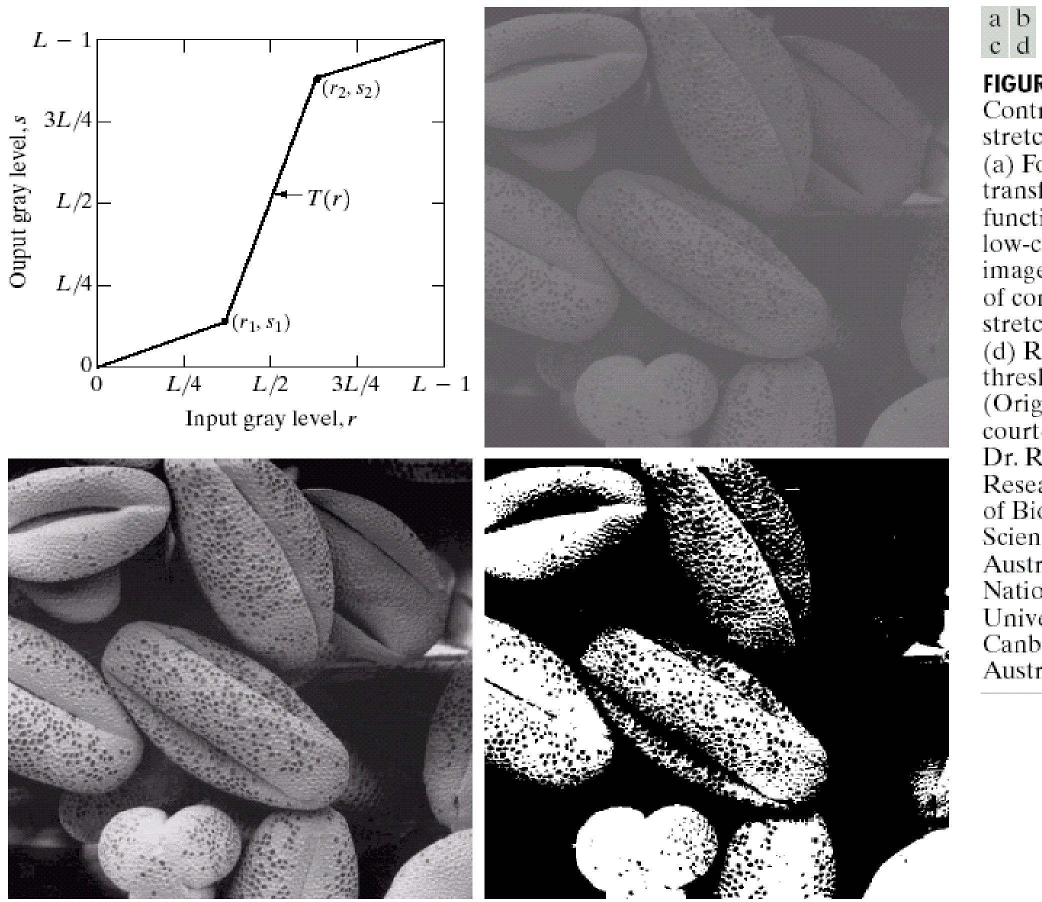


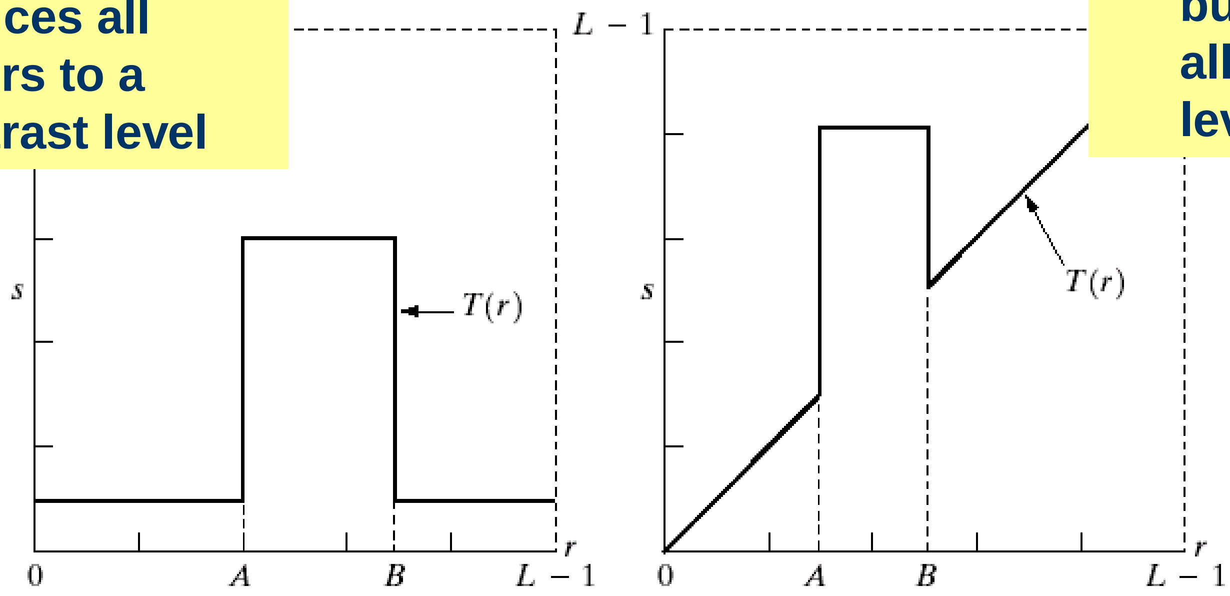
FIGURE 3.10 Contrast stretching. (a) Form of transformation function. (b) A low-contrast image. (c) Result of contrast stretching. (d) Result of thresholding. (Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)

Grey Level Slicing

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

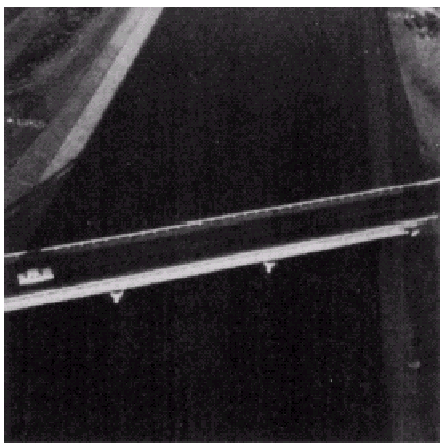
Grey Level Slicing

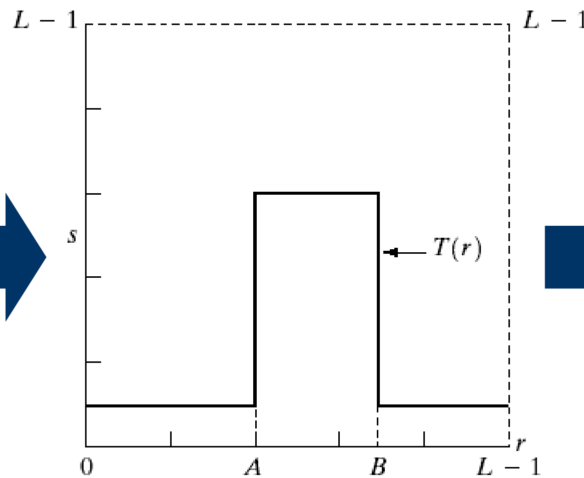
Highlights range [A,B] of gray levels and reduces all others to a contrast level

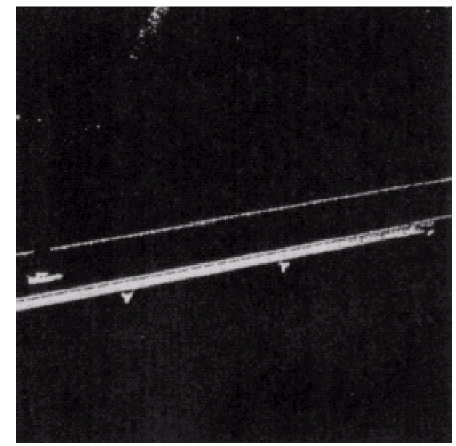


Highlights range [A,B] but preserves all other gray levels

Grey Level Slicing



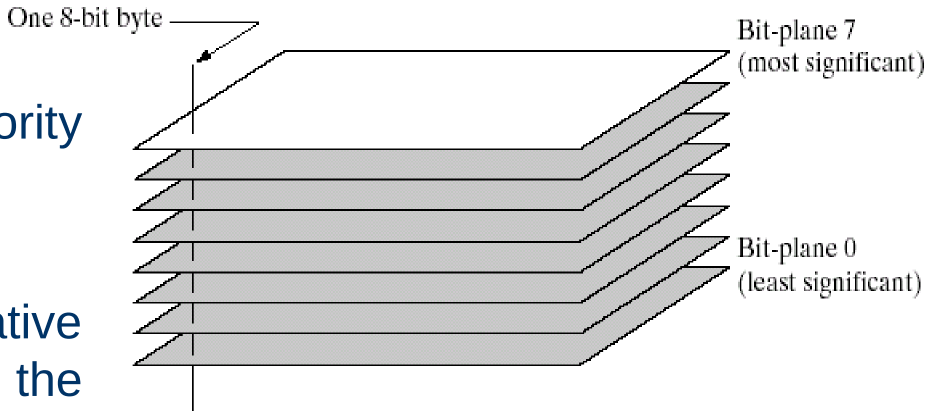




Bit Plane Slicing

- **Objective:** Highlights the contribution made to the image appearance by specific bits

- Suppose an image is of 8 bits i.e. each pixel is represented by 8 bits
- Higher-order bits contain the majority of the visually significant data
- Useful for analyzing the relative importance played by each bit of the image. It is useful in compression



Bit Plane Slicing

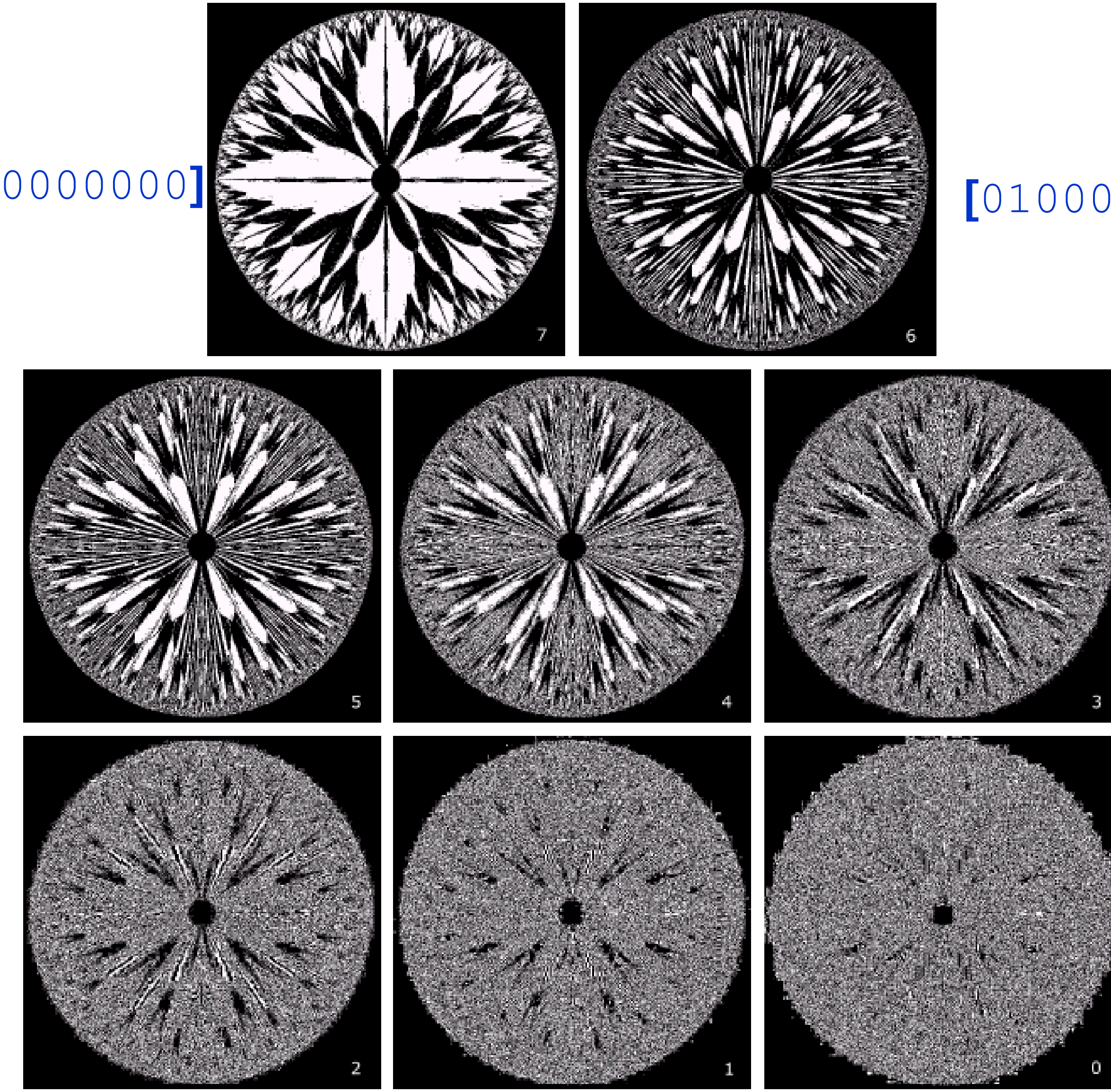
(MSB) 11111010 (LSB)

250 (11111010)	126(01111110)	26 (00011010)	255 (11111111)
0 (00000000)	42 (00101010)	32 (00100000)	21 (00010101)
1 (00000001)	2 (00000010)	16 (00010000)	22 (00010110)
99 (01100011)	198 (11000110)	8 (00001000)	96 (01100000)

[10000000] [01000000]

[00100000] [00001000]

[00000100] [00000001]



Bit Plane Slicing

Bit planes 1 through 8

Bit Plane Slicing

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

Acknowledgements

- Digital Image Processing", Rafael C. Gonzalez & Richard E. Woods, Addison-Wesley, 2002
- Peters, Richard Alan, II, Lectures on Image Processing, Vanderbilt University, Nashville, TN, April 2008
- Brian Mac Namee, Digitial Image Processing, School of Computing, Dublin Institute of Technology
- Computer Vision for Computer Graphics, Mark Borg

Material in these slides has been taken from, the following resources