TDI2131 Digital Image Processing

Image Enhancement in Spatial Domain
Lecture 3

John See
Faculty of Information Technology
Multimedia University

Some portions of content adapted from Zhu Liu, AT&T Labs. Most figures from Gonzalez/Woods

Lecture Outline

- Introduction
- Point Processing
- Basic Gray Level Transformation Functions
- Piecewise-Linear Transformation Functions
- Histogram Processing – Histogram Equalization

Some Announcements

- Consultation Hours: **Thursdays, 2-6pm**
- Some of you have very poor attendance (as I'm beginning to see it...) -- I will **NOT** hesitate to bar you.
- Please get a copy of Matlab installed – so that you can work on your tutorial exercises and coming assignments.

What is Image Enhancement?

Original image with low contrast

Enhanced image

Image Enhancement

- Images are obtained from various sensor outputs
- Sometimes, they are NOT suitable for use in applications, e.g. Mars Probe Images, X-ray images
- Image Enhancement – A set of image processing operations applied on images to produce good images useful for a specific application.

Principle Objective of Enhancement

- Process an image so that the result will be more suitable than the original image for a specific application
- The suitability depends on each specific application
  - A method useful for enhancing a certain image may not necessarily be the best approach for enhancing other types of images

What is a good, suitable image?

- For human visual
  - The visual evaluation of image quality is a highly subjective process
  - Hard to standardize the definition of a good image
- For machine perception
  - The evaluation task is staightforward
  - 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 image enhancement approach is selected

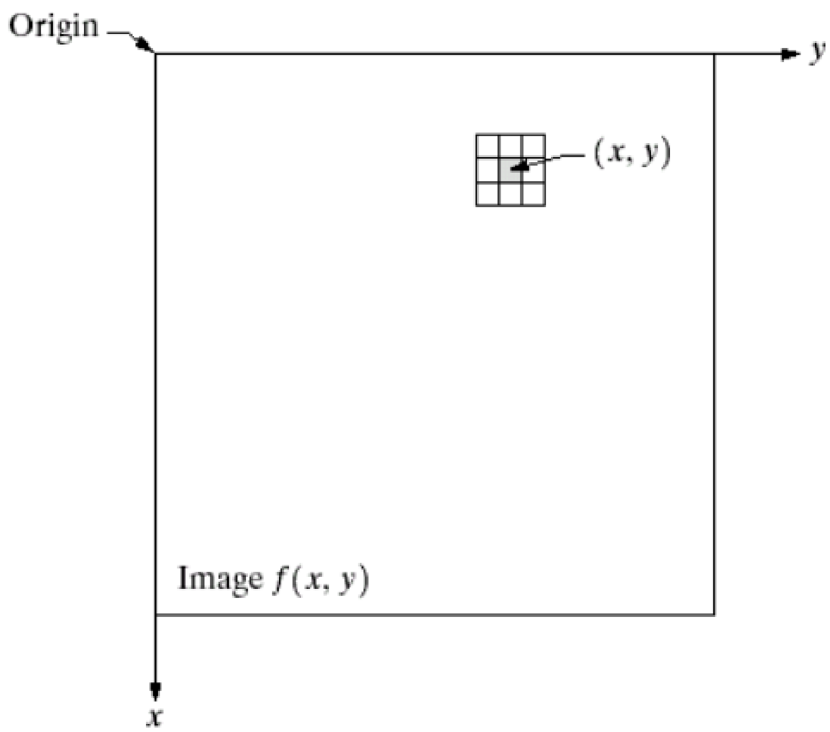
2 Domains

- Spatial Domain (image plane)
  - Techniques are based on direct manipulation of pixels in an image
- Frequency Domain
  - Tehniques are based on modifying the spectral transform (in our course, we'll use Fourier transform) of an image
- There are some enhancement techniques based on various combinations of methods from these 2 domains

3 Types of Processing

- Any image processing operation transforms the gray values of the pixels
- Divided into 3 classes based on the information required to perform the transformation
  - Point processing – Gray values change without any knowledge of its surroundings
  - Neighborhood processing – Gray values change depending on the gray values in a small neighborhood of pixels around the given pixel
  - Transforms – Gray values are represented in a different domain, but equivalent form, e.g. Fourier, wavelet

Spatial Domain



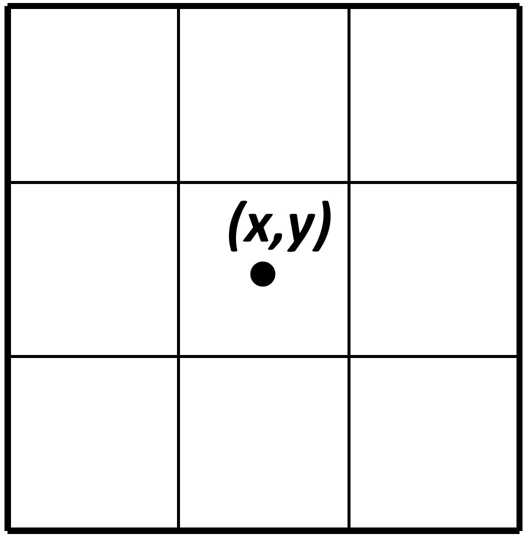
- Procedures that operate directly on pixels

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

  – where

  - $f(x, y)$ is the input image
  - $g(x, y)$ is the processed image
  - T is the operator on f defined over some neighborhood of (x, y)

Mask/Filter



- Neighborhood of a point (x,y) can be defined by using a square/rectangular (commonly used) or circular sub-image area centered at (x,y)
- The center of the sub-image is moved from pixel to pixel starting from the top corner of the processed image
- To be covered in next lecture!

Point Processing

- Neighborhood = 1x1 pixel
- g depends on only the value of f at (x, y)
- T = gray level (or intensity or mapping) transformation function

$$s = T(r)$$

where

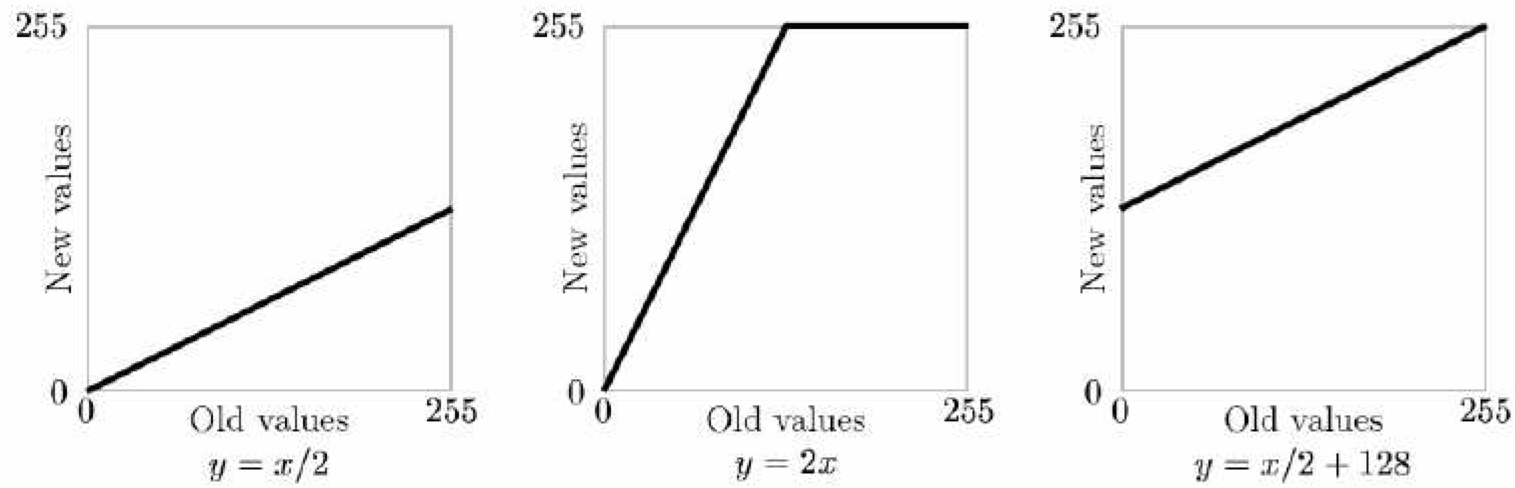
r = gray level of $f(x, y)$

s = gray level of $g(x, y)$

Arithmetic Operations

- These operations act by applying a simple arithmetic function $s = T(r)$ to each gray level in the image
- T is a function that maps r to s.
- Addition, subtraction, scaling (multiplication & division), complement, etc.

Arithmetic Operations



b3: $y = x/2$ b4: $y = 2x$ b5: $y = x/2 + 128$

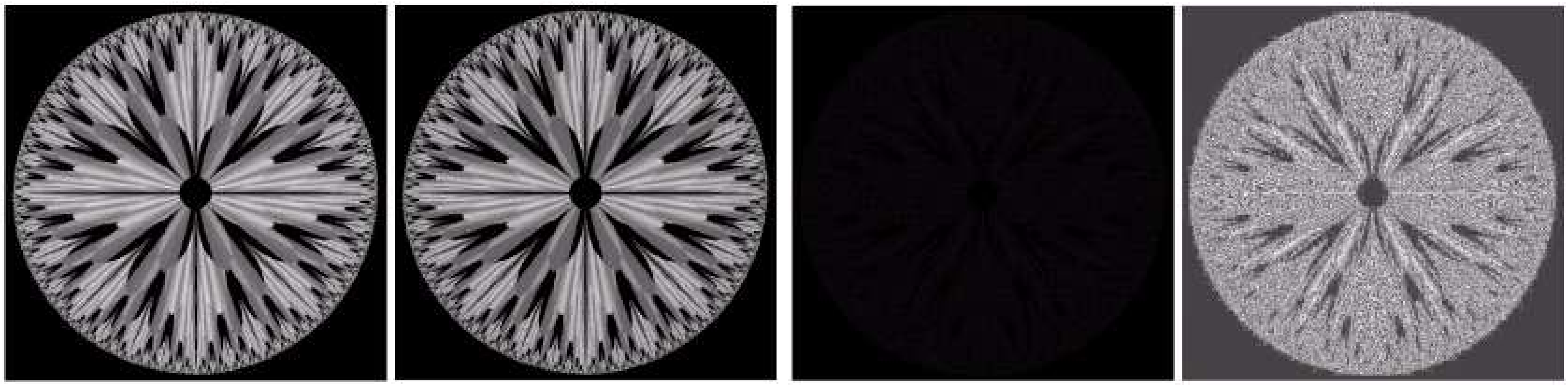
Image Subtraction

- The difference between two images $f(x,y)$ and $h(x,y)$,

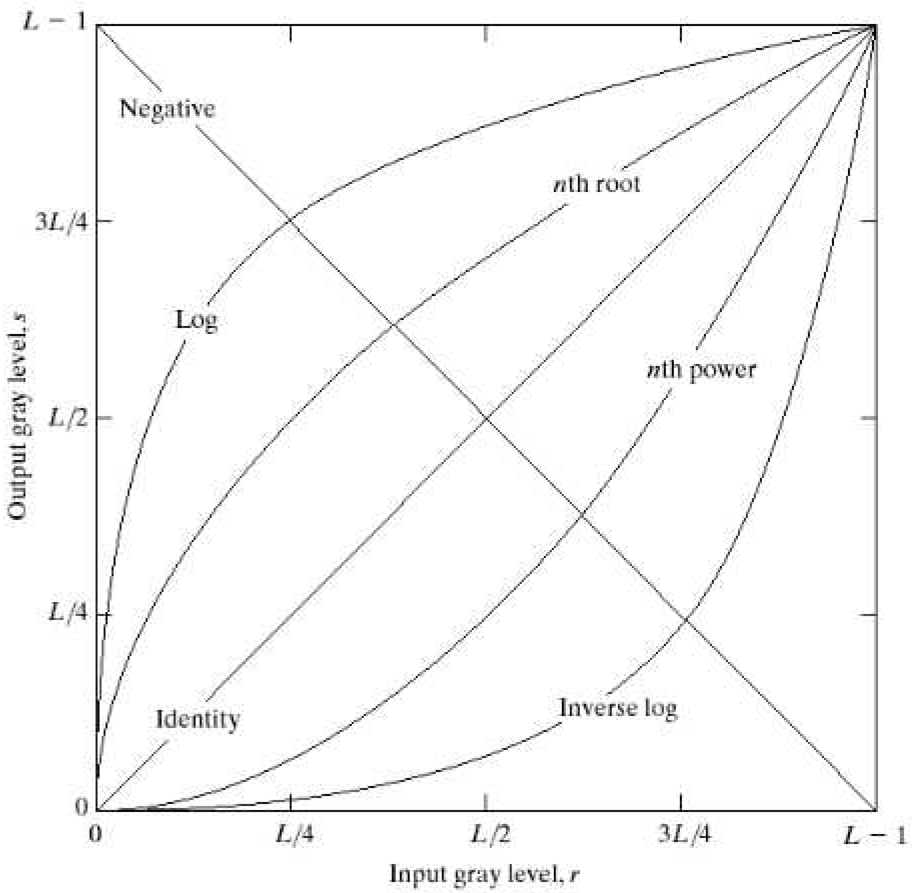
$$g(x,y) = f(x,y) - h(x,y)$$

  - is obtained by computing the difference between all pairs of corresponding pixels

- Usefulness: Enhancement of differences between images

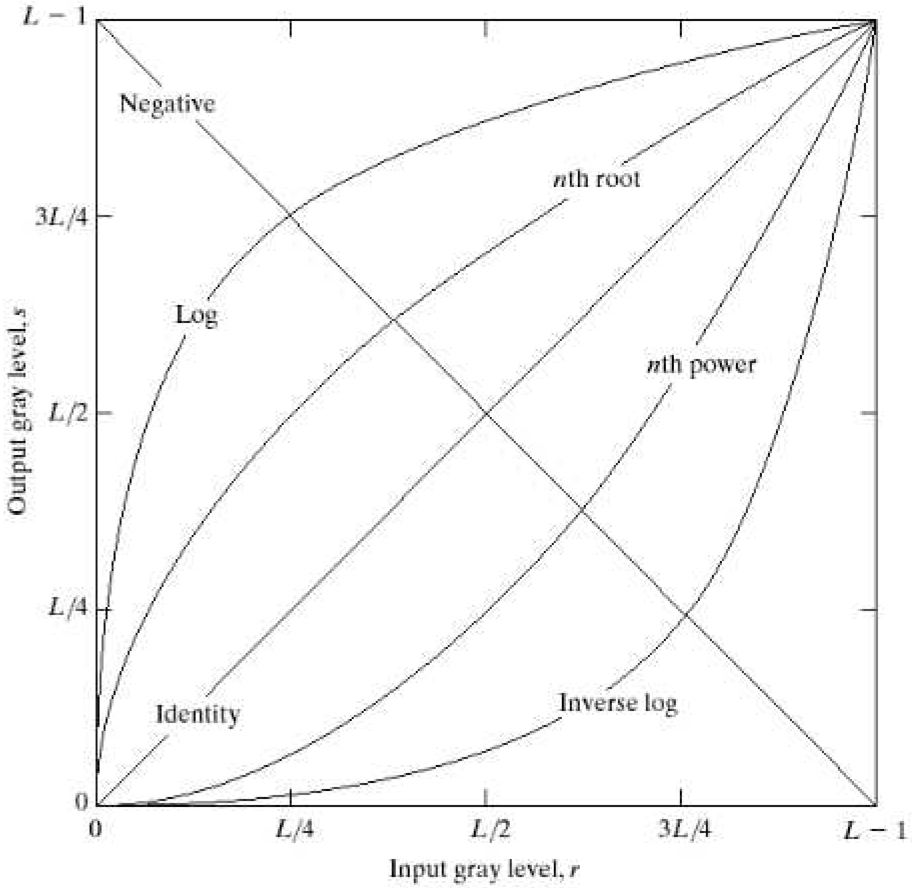


Basic Gray-level Transformation Functions



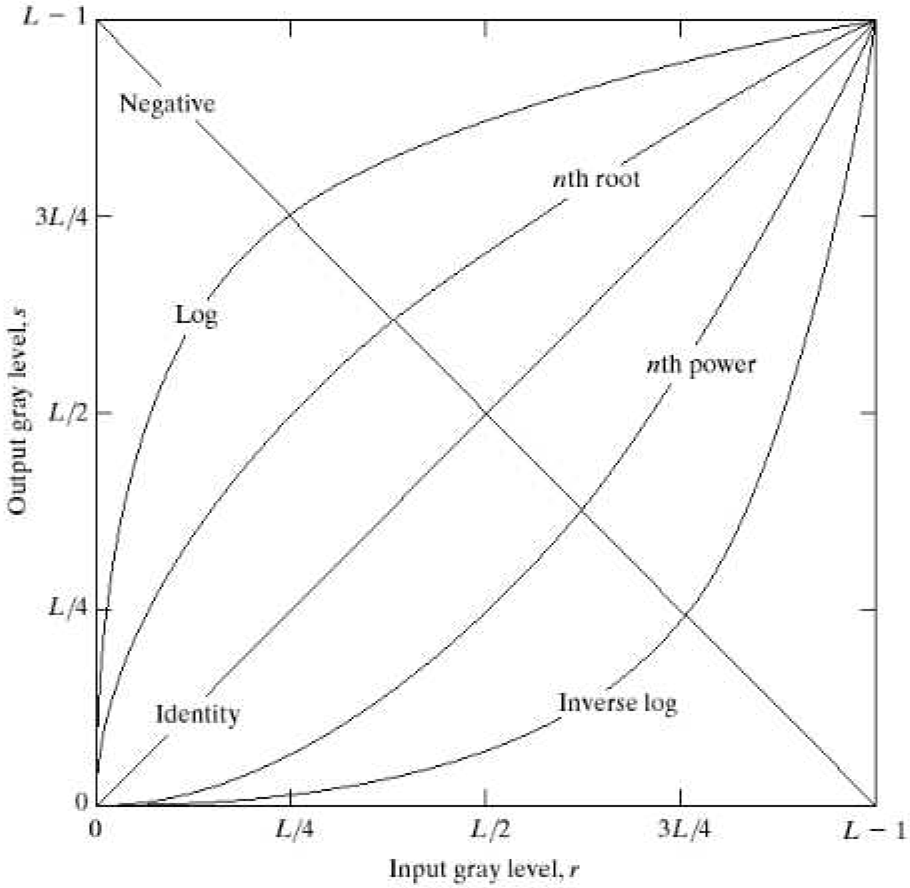
- Linear function
  - Negative and identity transformations
- Logarithm function
  - Log and inverse-log transformations
- Power-law function
  - n^{th} power and n^{th} root transformations

Identity Function



- Output intensities are identical to input intensities
- What "goes in", "comes out" the same

Negative Transformation



- For an image with gray levels in the range [0, L-1]

  where $L = 2^n$, n = 1, 2, ...

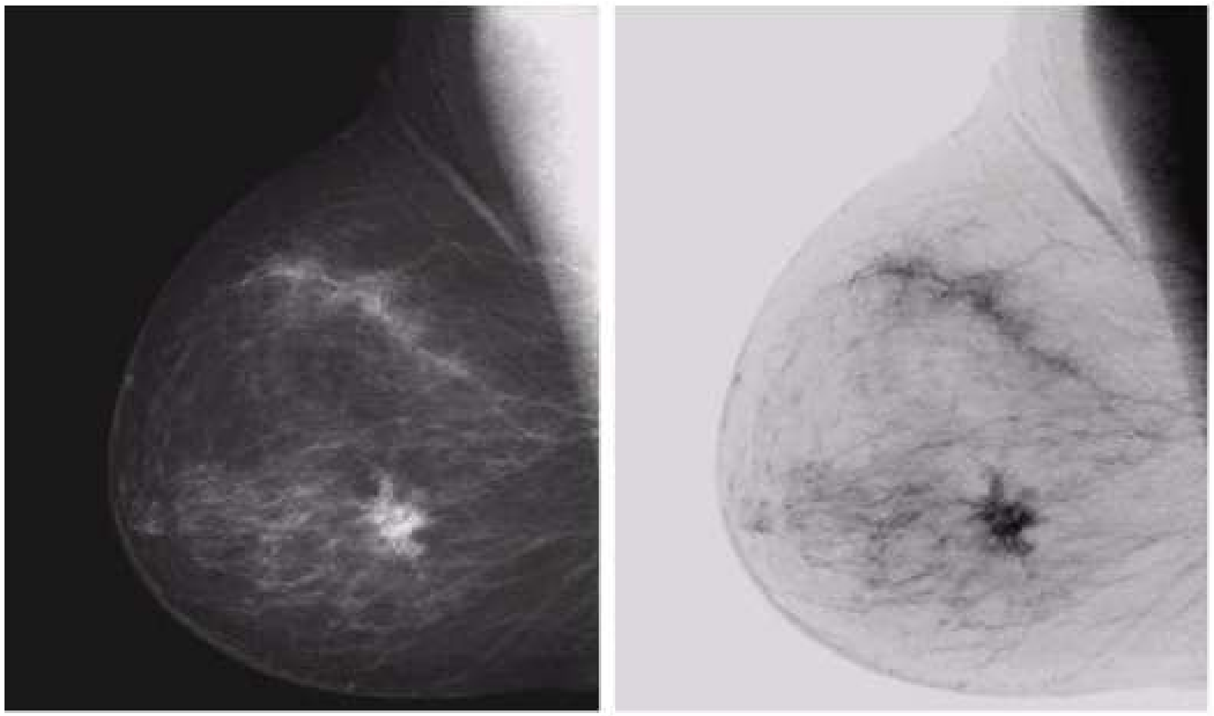
- Negative transformation:

  $$s = L - 1 - r$$

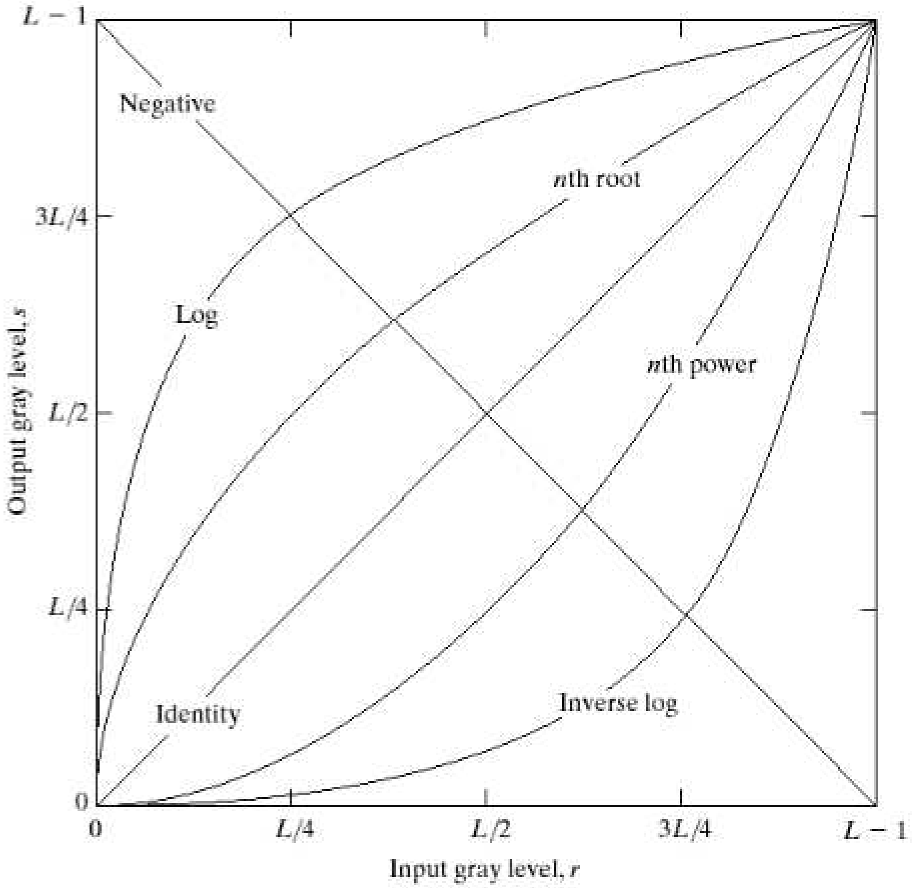
- Reversing the intensity levels of an image

Negative Transformation

- Suitable for enhancing white or gray detail embedded in dark regions of an image, especially when the black area is dominant in size



Log Transformation



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

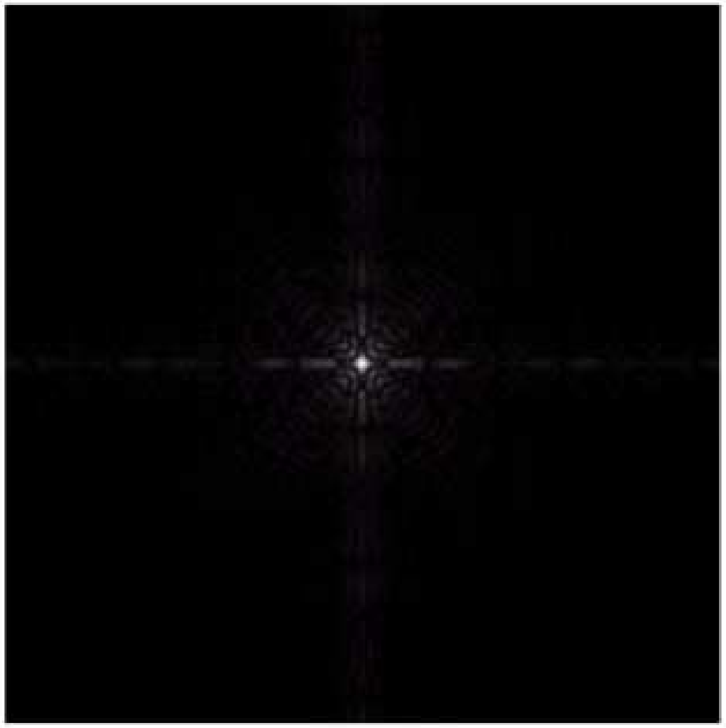
- c is a constant and $r \geq 0$
- Log curve maps a narrow range of low gray levels in the input image into a wider range of output levels
  - Expand the values of dark pixels
  - Compress higher value lighter pixels

Log Transformations

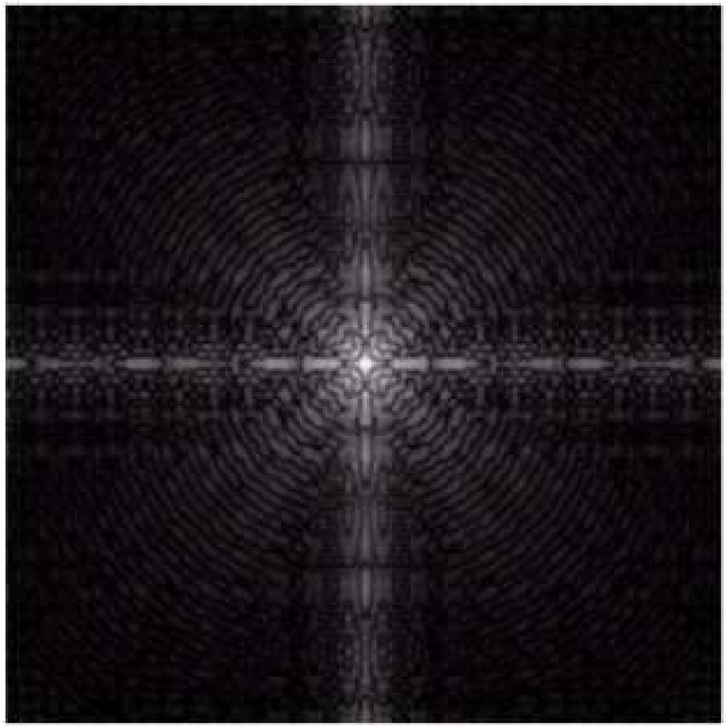
- Compresses the dynamic range of images with large variations in pixel values
- E.g. Image with dynamic range – Fourier spectrum image (to be discussed in Lecture 5)
  - Intensity range from 0 to 10^6 or higher
- We can't see the significant degree of detail as it will be lost in the display (remember: our human eyes have limitations!)

Log Transformations on Spectrum Images

- Log transformations bring up the details that are not visible due to large dynamic range of values



Fourier Spectrum with range: 0 to 1.5×10^6

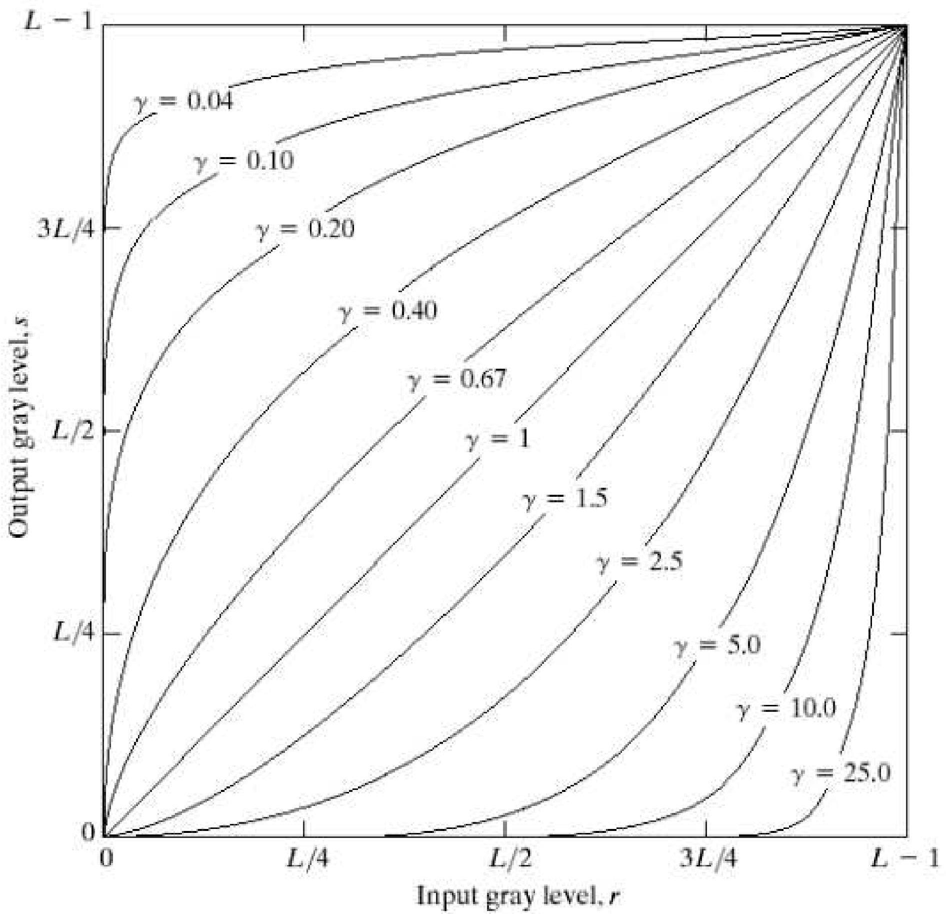


Result after applying log transformation with c=1, range: 0 to 6.2

Inverse Log Transformations

- Do the opposite of Log Transformations
- Used to expand the higher value pixels in an image while compressing darker-level values

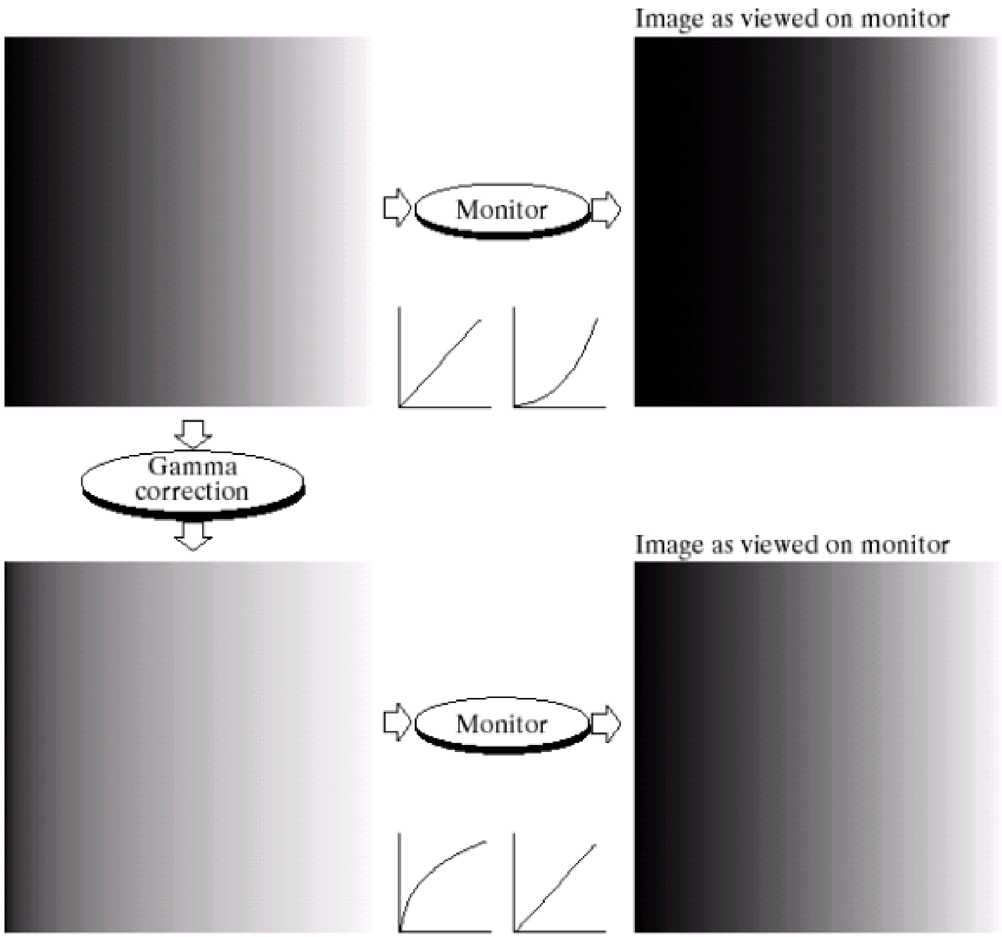
Power-Law Transformation



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

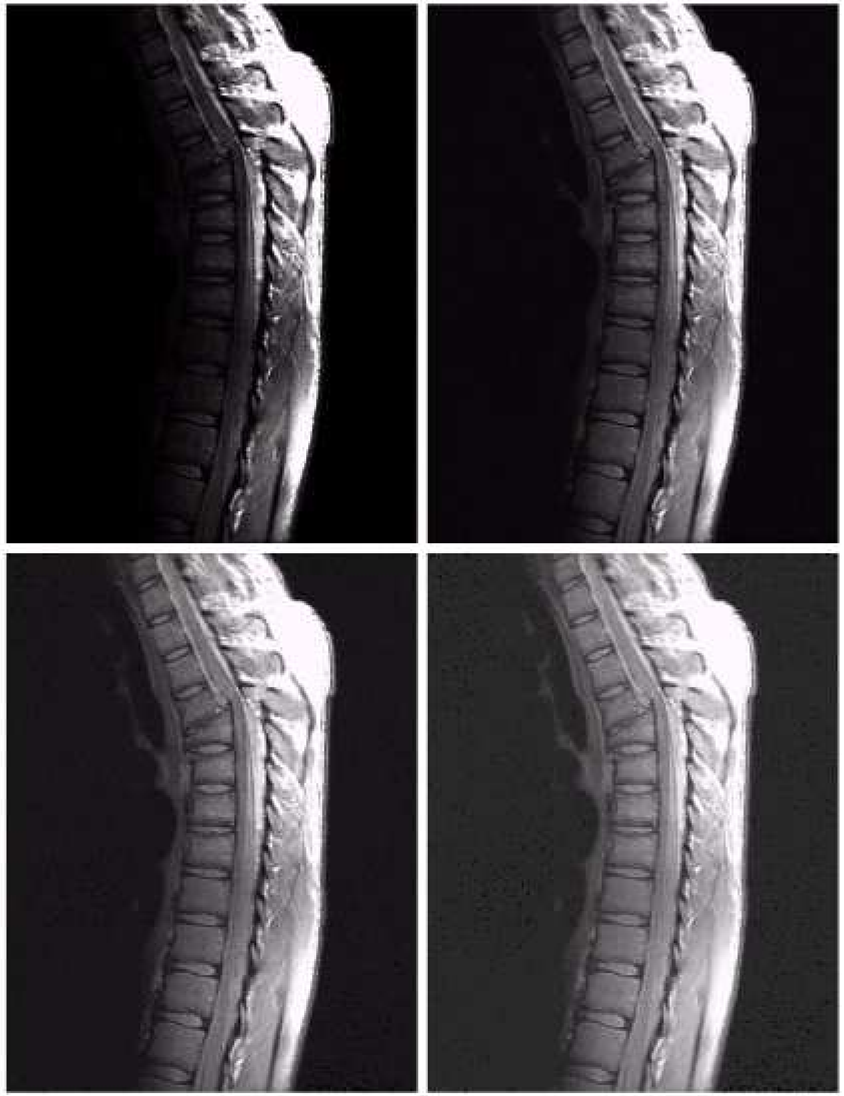
- c and γ are positive constants
- Power-law curves with fractional values of γ map a narrow range of dark input values into a wider range of output value, with the opposite being true for higher values of input levels
- $c = \gamma = 1$, results in the identity function

Gamma Correction



- CRT devices have a power function, with γ varying from 1.8 to 2.5
- Macintosh (1.8), PC (2.5)
- The picture appears darker
- Gamma correction is done by preprocessing the image before inputting it to the monitor with $s = cr^{1/\gamma}$

Application: MRI of Human Spine

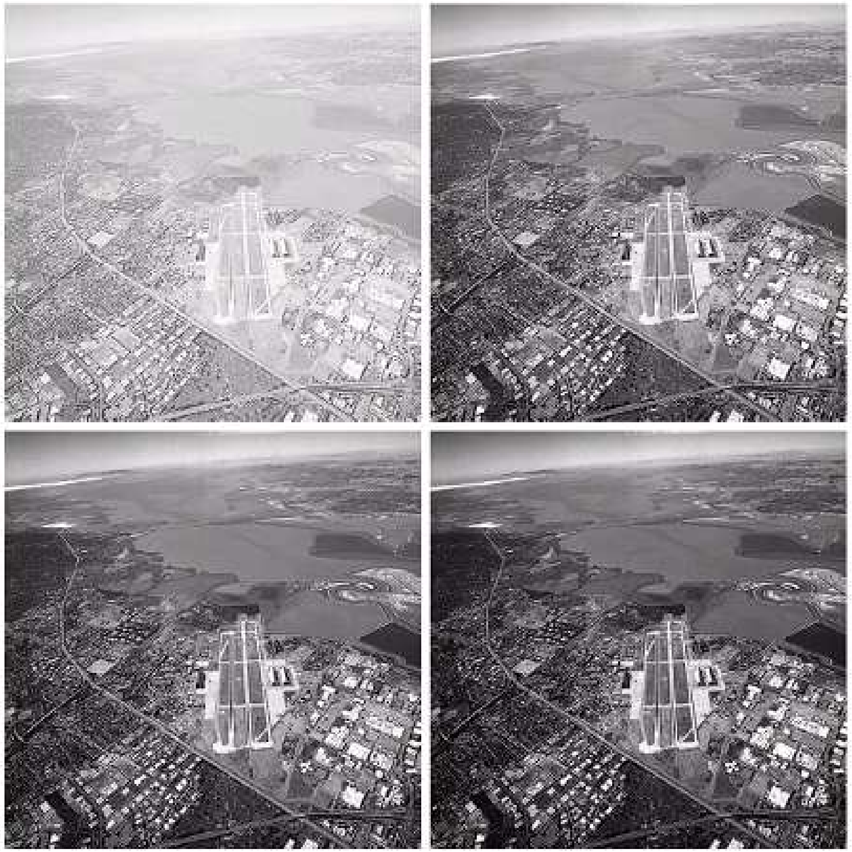


- Problem: Picture is too dark
- Solution: Expansion of lower gray levels is desirable, $\gamma < 1$
- $\gamma = 0.6$ (insufficient)
- $\gamma = 0.4$ (best result)
- $\gamma = 0.3$ (contrast lacking)

Decreasing Gamma?

- When γ is reduced too much, the image begins to reduce contrast to the point where the image starts to have a “wash-out” look, especially in the background

Application: Aerial Imagery



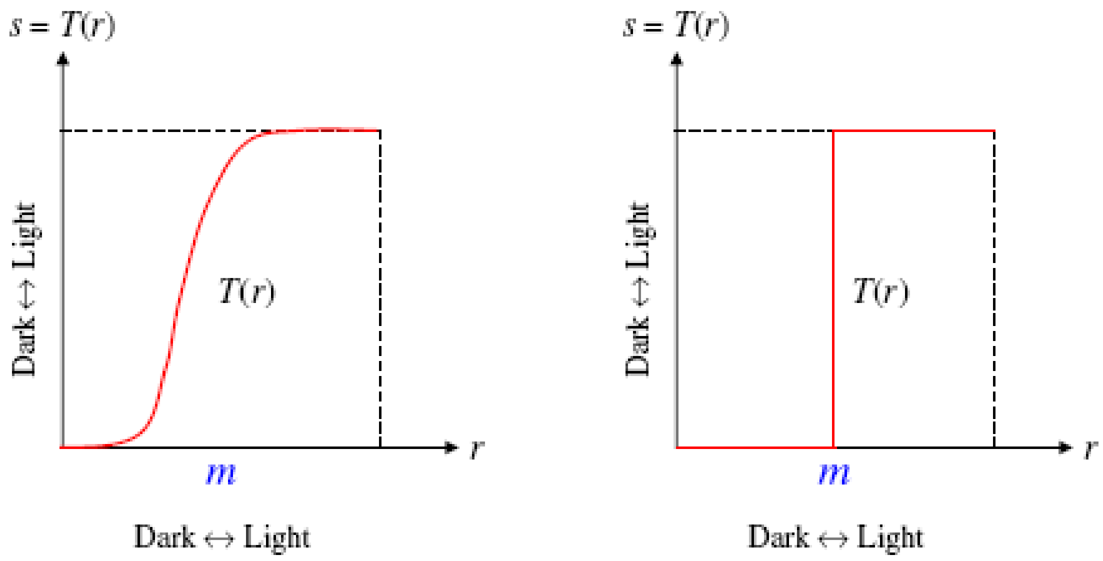
- Problem: Image has “wash-out” appearance
- Solution: Compression of higher gray levels is desirable, $\gamma > 1$
- $\gamma = 3.0$ (suitable)
- $\gamma = 4.0$ (suitable)
- $\gamma = 5.0$ (high contrast, some finer details are lost)

Piecewise-Linear Transformation Functions

- Advantage
  - The form of piecewise functions can be arbitrarily complex
  - Some important transformations can be formulated only as piecewise functions
- Disadvantage
  - Specification requires considerably more user input

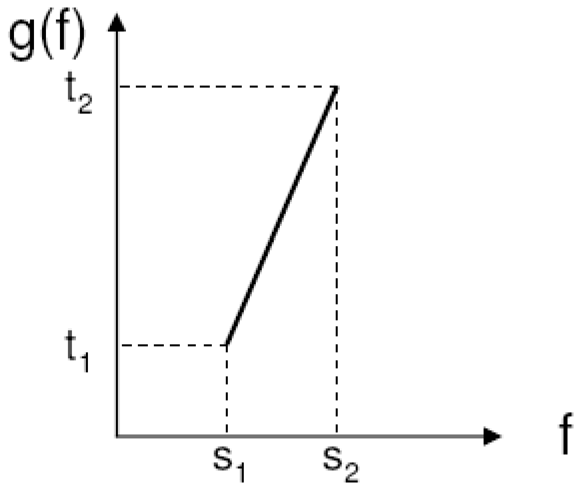
Contrast Stretching

- Produce higher contrast than the original by
  - Darkening the levels below *m* in the original image
  - Brightening the levels above *m* in the original
  - Thresholding: produce a binary image



Linear Stretching

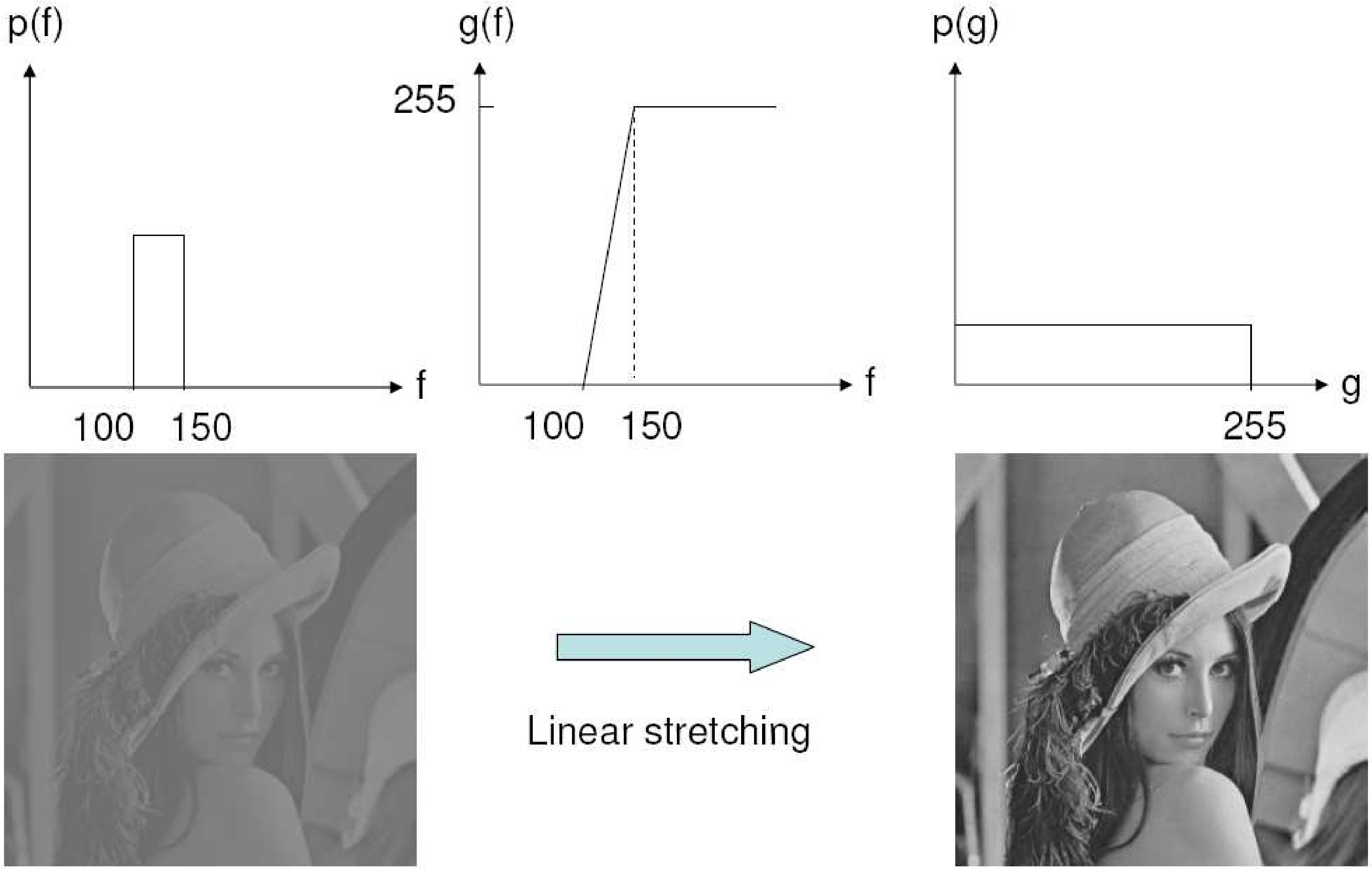
- Enhance the dynamic range by linear stretching the original gray levels to the range of 0 to 255
- Example
  - The original gray levels are [100, 150]
  - The target gray levels are [0, 255]
  - The transformation function



$$g(f) = (f - s_1)/(s_2 - s_1)*(t_2 - t_1) + t_1$$

$$g(f) = ((f - 100)/50)*255 + 0, \qquad \text{for } 100 \leq f \leq 150$$

Illustration of Linear Stretching

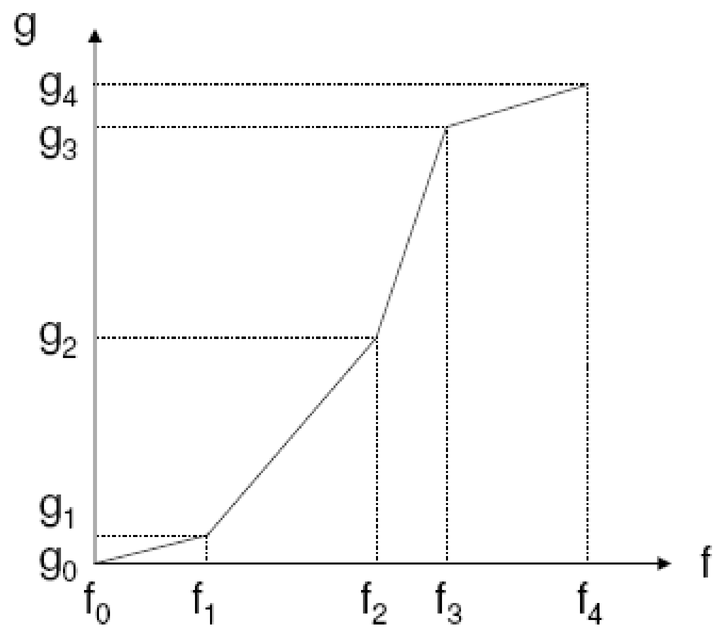


Piecewise Linear Stretching

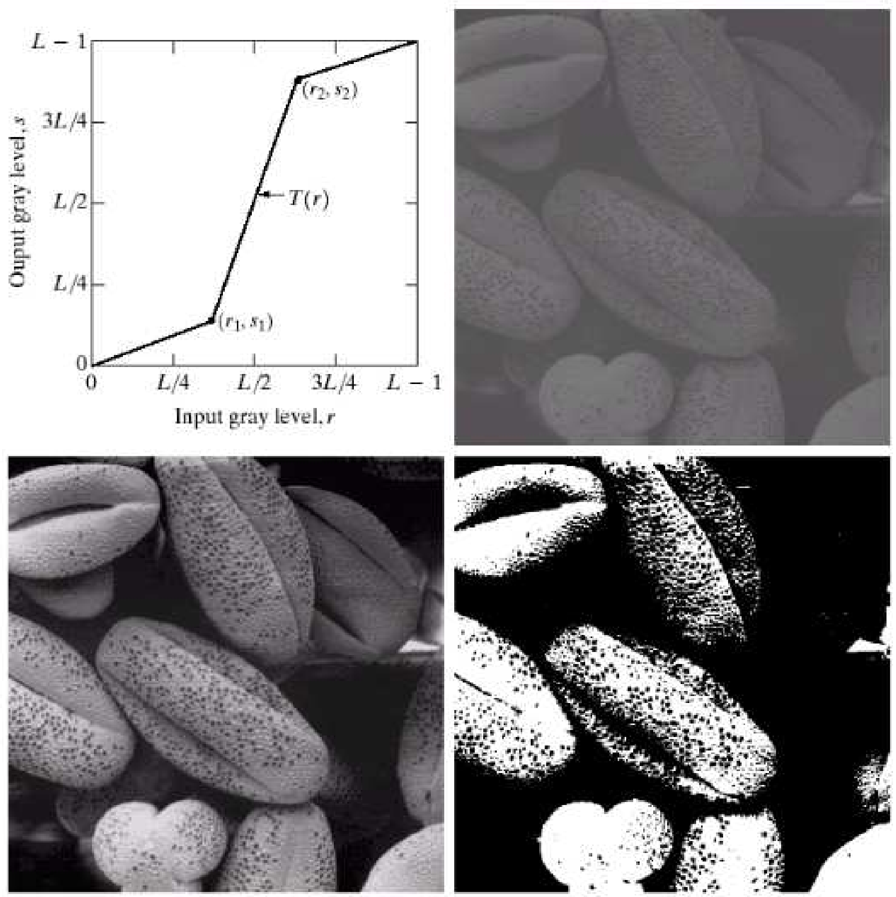
- K segments
  - Starting position of input $\{f_k, k = 0, \ldots K-1\}$
  - Starting position of output $\{g_k, k = 0, \ldots K-1\}$
  - Transform function

$$g(f) = \frac{f - f_k}{f_{k+1} - f_k} * (g_{k+1} - g_k) + g_k;$$

$$for\ f_k < f \le f_{k+1}, \quad k = 0,1,\ldots,K-1.$$

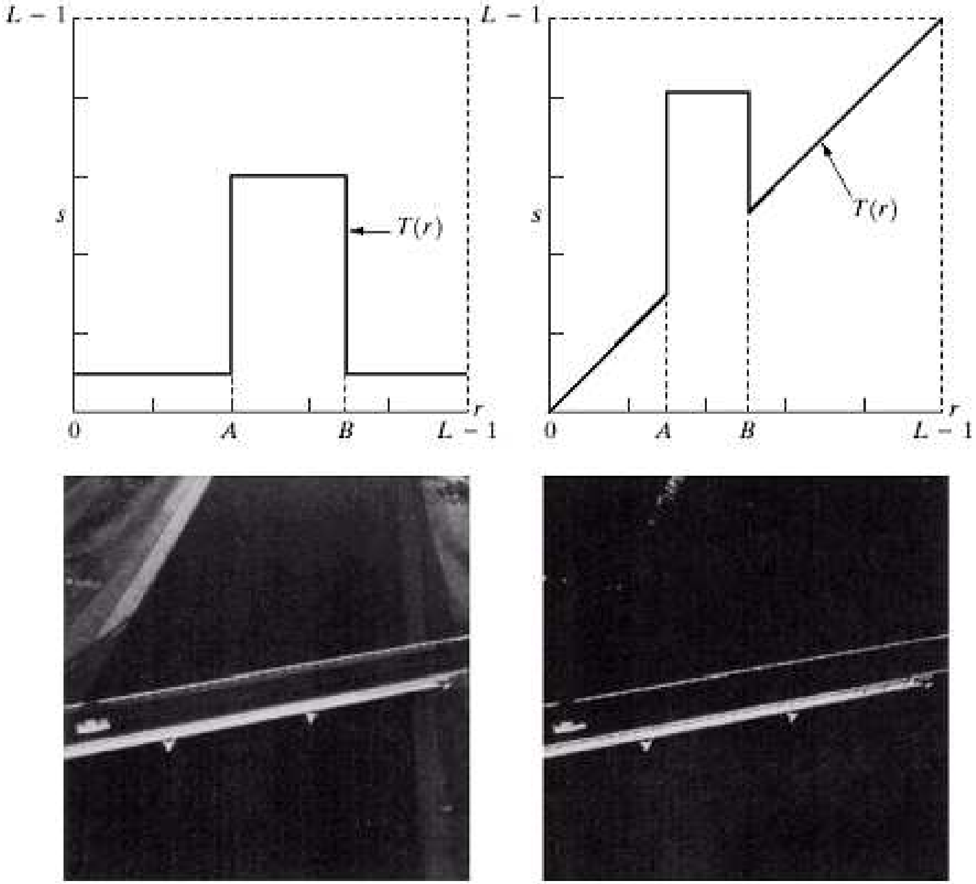


Application: Contrast Stretching



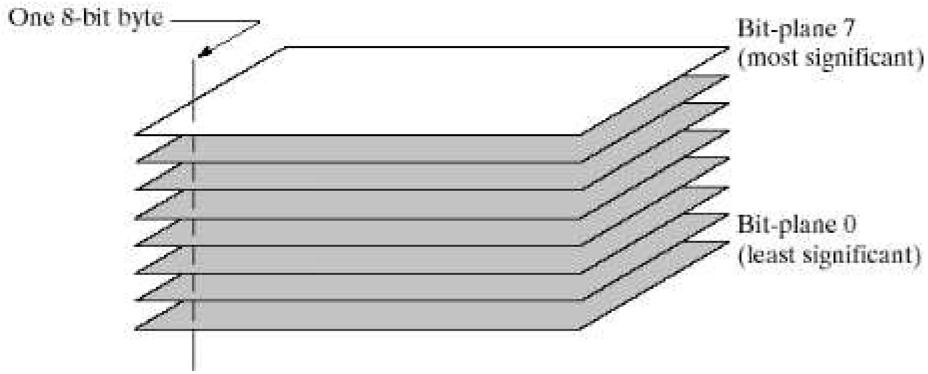
- Problem: Low contrast image, result of poor illumination, lack of dynamic range
- Solution: Contrast stretching using the given transformation function *(bottom left)*
- Result of thresholding *(bottom right)*

Gray-level Slicing



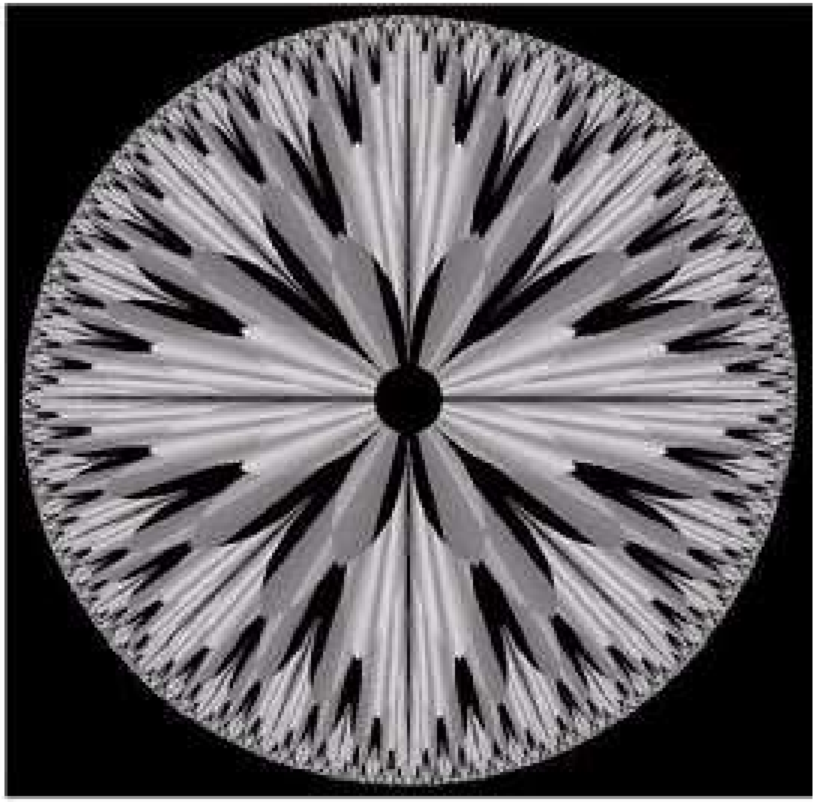
- Highlighting a specific range of gray levels in an image
  - Display high value for gray levels in the range of interest and low value for all other gray levels
- *(left)* Highlights range [A,B] and reduces all others to a constant level
- *(right)* Highlights range [A,B] but preserves all other levels

Bit-plane Slicing



- Highlighting the contribution made to total image appearance by specific bits
- Assume each pixel is represented by 8 bits
- Higher-order bits contain the majority of the visually significant data – Useful for analyzing relative importance played by each bit

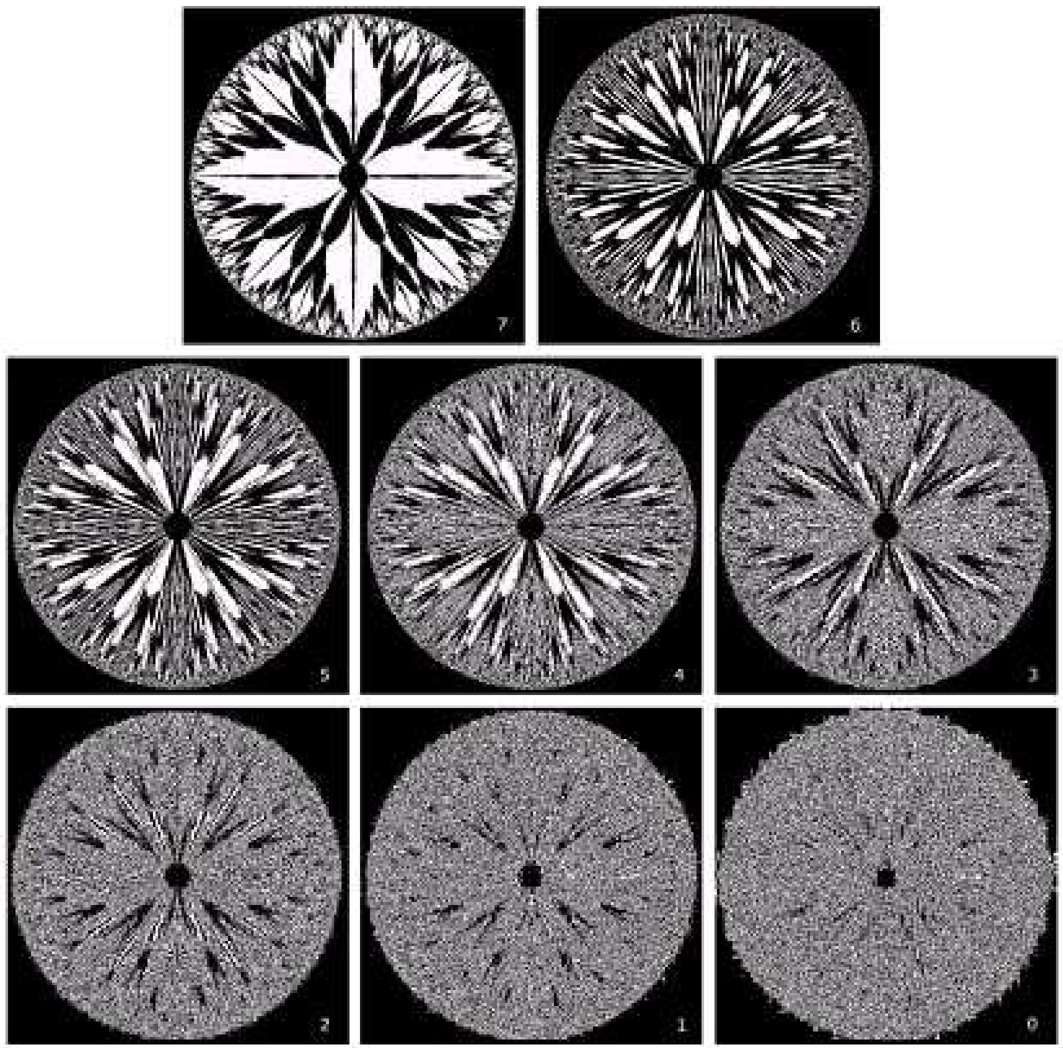
Example: Slicing Fractals



An 8-bit fractal image

- The (binary) image for bit-plane 7 can be obtained by processing the input image with a thresholding gray-level transformation
  - Map all levels between 0 and 127 to 0
  - Map all levels between 128 and 255 to 255

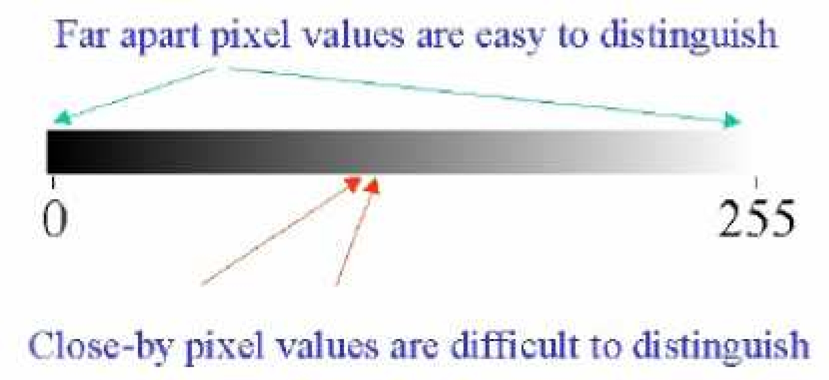
Example: Slicing Fractals



<table>
<tr><td colspan="3">Bit-plane 7</td><td colspan="3">Bit-plane 6</td></tr>
<tr><td colspan="2">Bit-
plane 5</td><td colspan="2">Bit-
plane 4</td><td colspan="2">Bit-
plane 3</td></tr>
<tr><td colspan="2">Bit-
plane 2</td><td colspan="2">Bit-
plane 1</td><td colspan="2">Bit-
plane 0</td></tr>
</table>

How to Enhance Contrast?

- We know that Contrast Stretching is one particular technique



- Low contrast – image values concentrated in a narrow range
- Contrast enhancement – change the image value distribution to cover a wide range
- Contrast of an image can be revealed by its histogram

Image Histogram

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

$$h(r_k) = n_k$$

where

- r_k : the kth gray level
- n_k : the number of pixels in the image having gray level r_k
- $h(r_k)$: histogram of a digital image with gray levels r_k

Normalized Histogram

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

$$p(r_k) = n_k/n$$

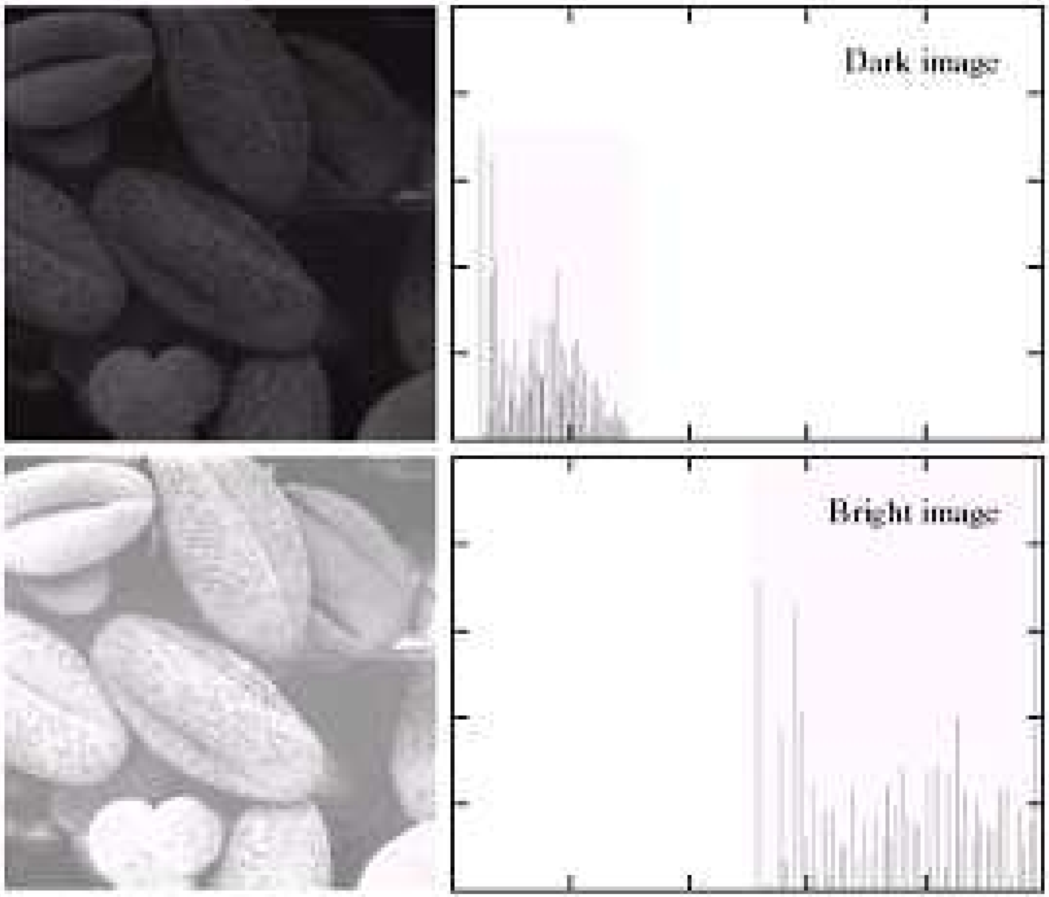
for $k = 0, 1, \ldots, L\text{-}1$

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

Histogram Processing

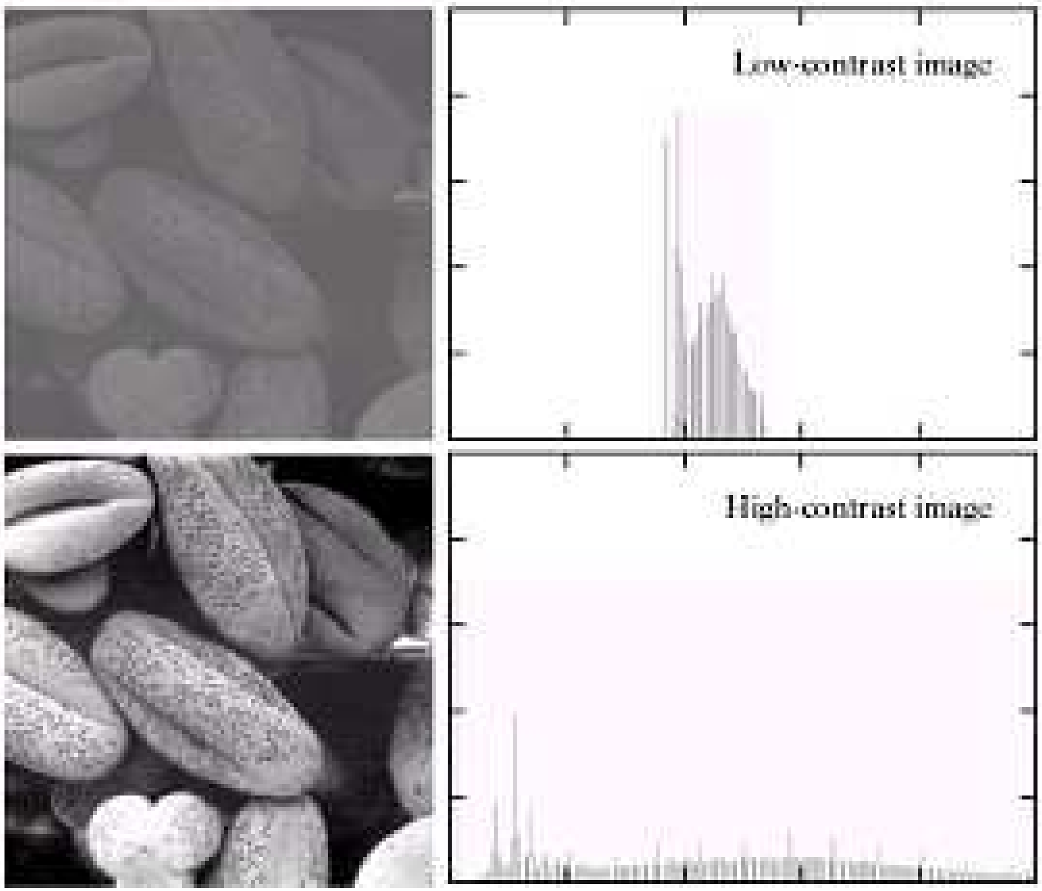
- Basic for numerous spatial domain processing techniques
- Used effectively for image enhancement
- Information inherent in histograms is also useful in image compression and segmentation

Histogram & Image Contrast



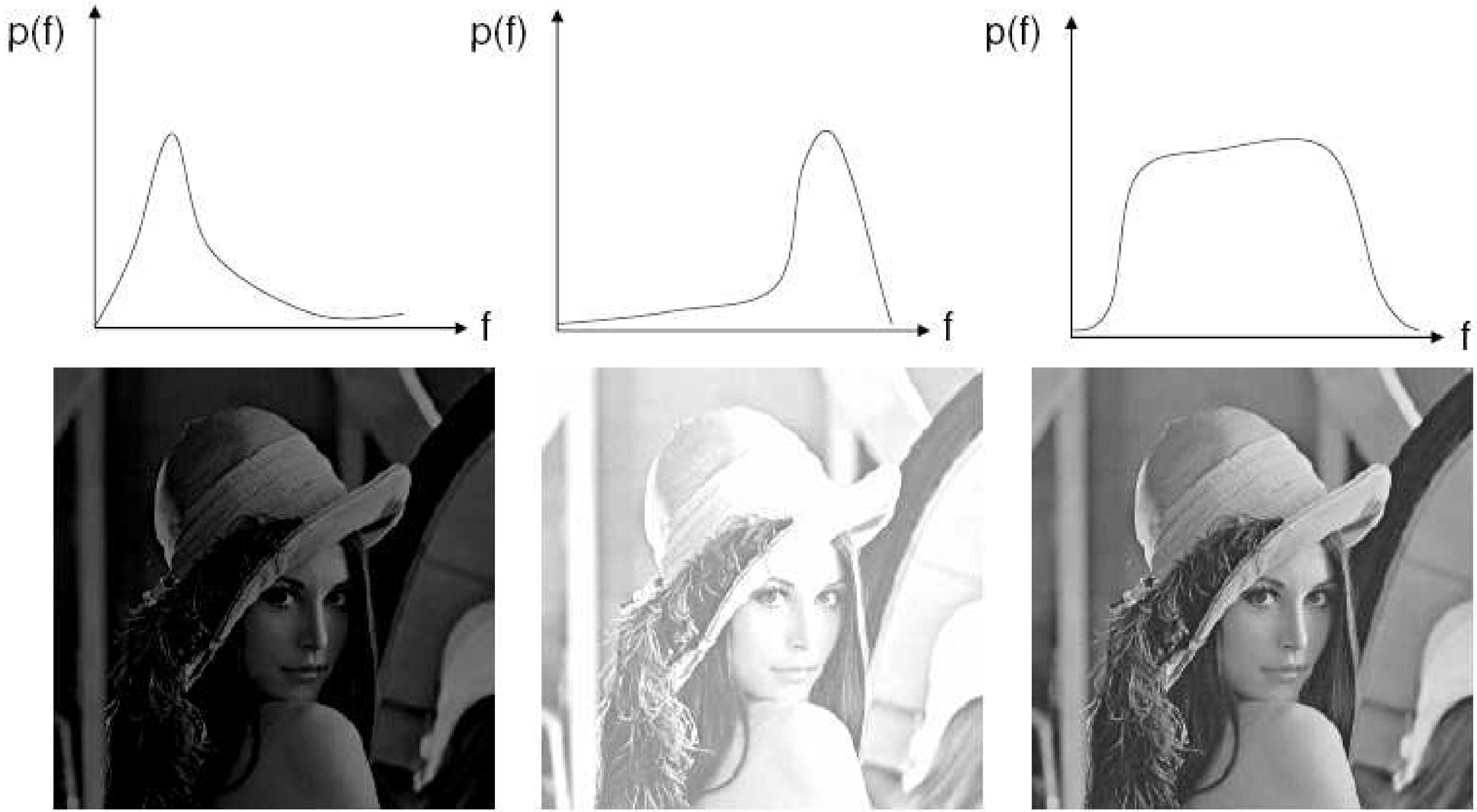
- Dark Image
  - Components of histogram are concentrated on the low side of the gray scale
- Bright Image
  - Components of histogram are concentrated on the high side of the gray scale

Histogram & Image Contrast



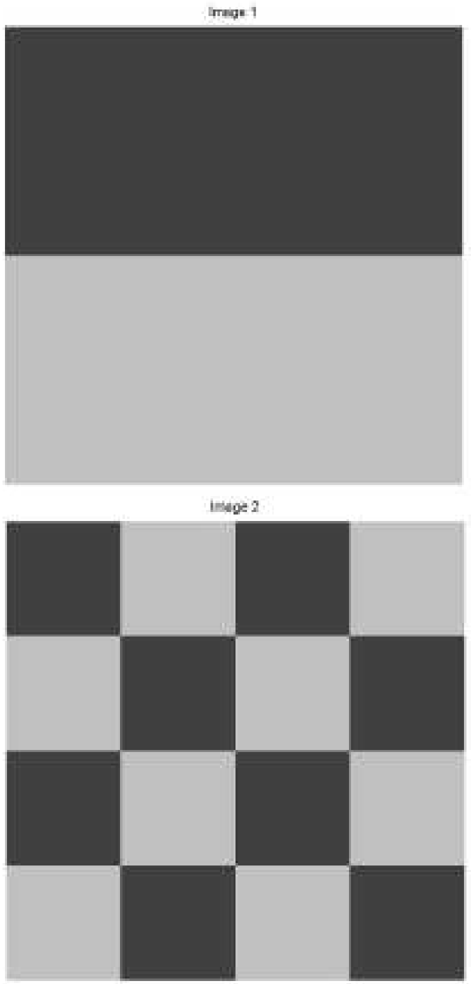
- Low-contrast Image
  - Histogram is narrow and centred towards the middle of the gray scale
- High-contrast Image
  - Histogram covers a broad range of the gray scale and the distribution of pixels is not too far from uniform, with very few vertical lines being much higher than others

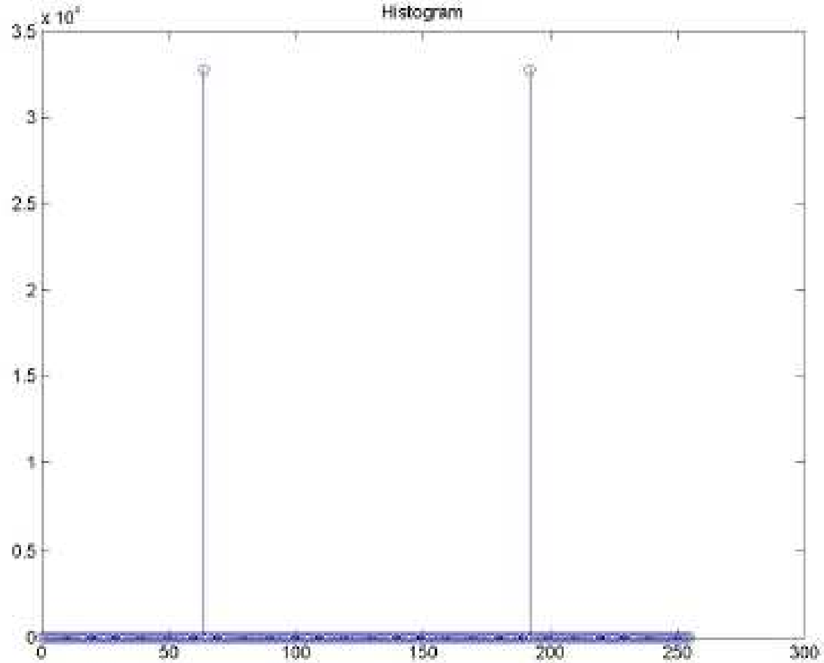
Histogram & Image Contrast



(a) Too dark (b) Too bright (c) Well balanced

Different Images with Same Histogram!



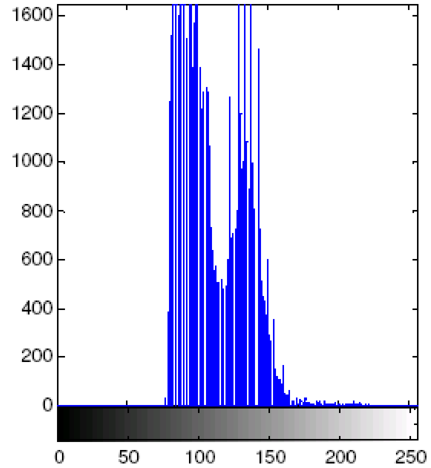


Histogram reflects the pixel intensity distribution, not the spatial distribution!

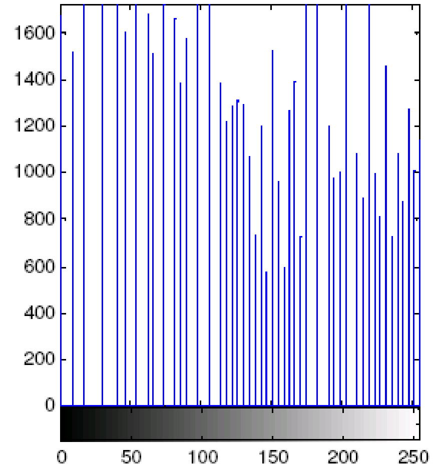
Correcting the Pouting Child

Original image with low contrast

Enhanced image



Original girl image with low contrast

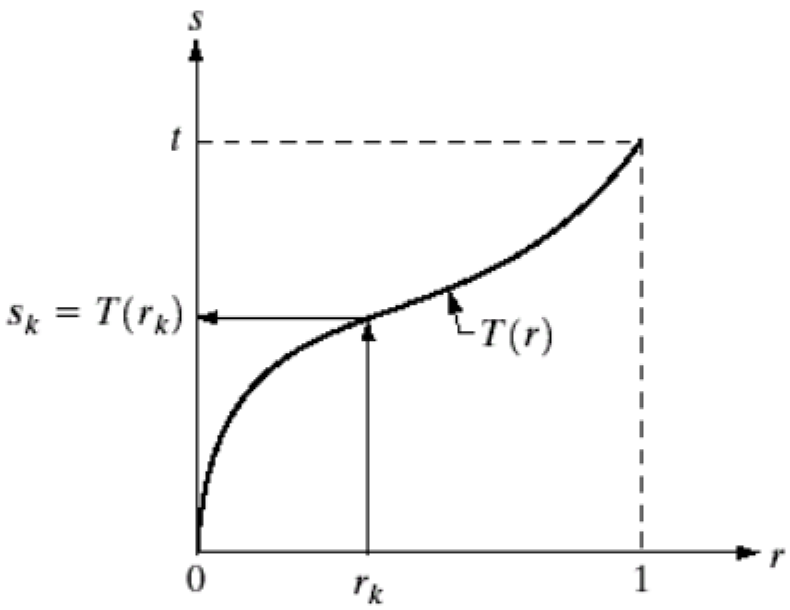


Enhancement image with histogram equalization

Histogram Equalization

- Histogram EQUALization
  - Aim: To “equalize” the histogram, to “flatten”, “distrubute as uniform as possible”
- As the low-contrast image's histogram is narrow and centred towards the middle of the gray scale, by distributing the histogram to a wider range will improve the quality of the image
- Adjust probability density function of the original histogram so that the probabilities spread equally

Histogram Transformation



$s = T(r)$

- Where $0 \leq r \leq 1$
- $T(r)$ satisfies

  a) $T(r)$ is single-valued and monotonically increasingly in the interval $0 \leq r \leq 1$

  b) $0 \leq T(r) \leq 1$ for $0 \leq r \leq 1$
- Single-valued guarantees that the inverse transformation will exist

Histogram Equalization

- Transforms an image with an arbitrary histogram to one with a flat histogram
- Suppose f has PDF, $p_F(f)$, $0 \le f \le 1$
- Transform function (continuous version) $g(f) = \int_0^f p_F(t)dt$
- g is uniformly distributed in (0, 1)

Discrete Implementation

- For a discrete image f which takes values k = 0, ... , K-1, use

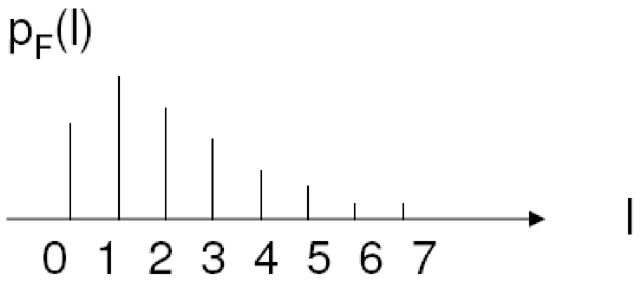
$$\tilde{g}(l) = \sum_{k=0}^{l} p_F(k), l = 0,1,\ldots,K-1.$$

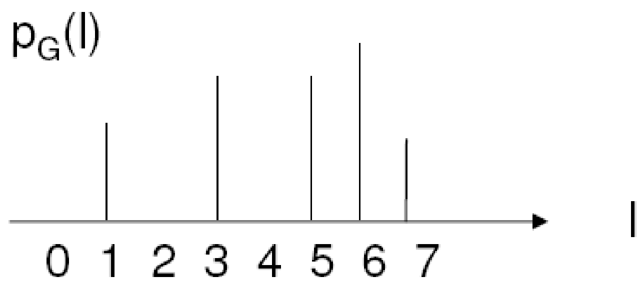
- To convert the transformed values to the range of (0, L-1):

$$g(l) = \left\lfloor \left(\sum_{k=0}^{l} p_F(k) \right) * (L-1) \right\rfloor$$

Example: Discrete Implementation

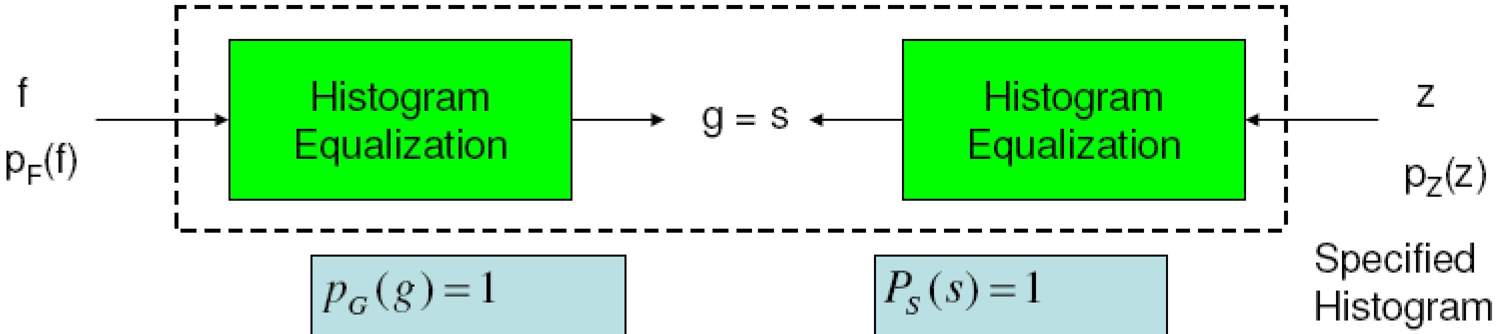
f_k	$p_F(l)$	$\tilde{g}_l = \sum_{k=0}^{l} p_F(k)$	$g_l = [\tilde{g}_l * 7]$	$p_G(l)$	g_k
0	0.19	0.19	[1.33]=1	0	0
1	0.25	0.44	[3.08]=3	0.19	1
2	0.21	0.65	[4.55]=5	0	2
3	0.16	0.81	[5.67]=6	0.25	3
4	0.08	0.89	[6.03]=6	0	4
5	0.06	0.95	[6.65]=7	0.21	5
6	0.03	0.98	[6.86]=7	0.16+0.08=0.24	6
7	0.02	1.00	[7]=7	0.06+0.03+0.02=0.11	7





Histogram Specification (Matching)

- What if the desired histogram is not flat?



$$p_G(g) = 1$$
$$g = g(f) = \int_0^f p_F(f)df$$

$$P_S(s) = 1$$
$$s = s(z) = \int_0^z p_Z(z)dz$$

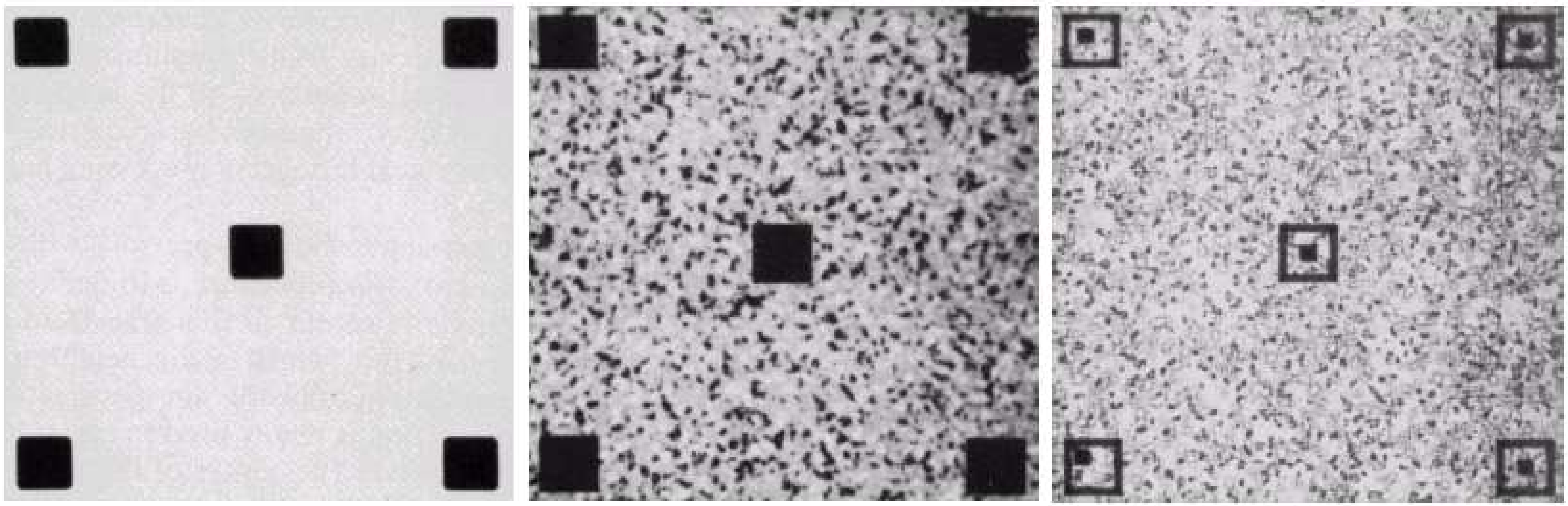
$$z = s^{-1}(g) = s^{-1}(g(f))$$

A Global Method?

- So far, would you consider Histogram Processing (and the other transformations covered so far) as a global method?
- Global: The pixels are modified by a transformation function based on the gray-level content of an entire image

Local Histogram Equalization

- Dealing with things locally



a b c

FIGURE 3.23 (a) Original image. (b) Result of global histogram equalization. (c) Result of local histogram equalization using a 7×7 neighborhood about each pixel.

Next week...

- Spatial Filtering (Neighborhood Processing)
  - Taking into consideration information from neighboring pixels

Recommended Readings

- Digital Image Processing (2nd Edition), Gonzalez & Woods,
  - Chapter 3:
    - 3.1 – 3.3 (Week 3)
    - 3.4 (Extra)
    - 3.5 – 3.8 (Week 4)