Image Enhancement

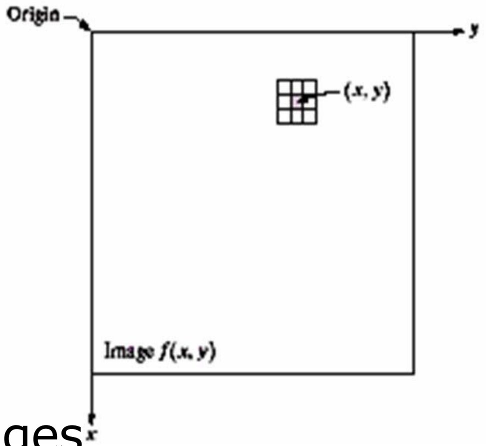
Alex Lin

Outline

- **Gray level transformations**
- Histogram processing
- Spatial filtering

Background

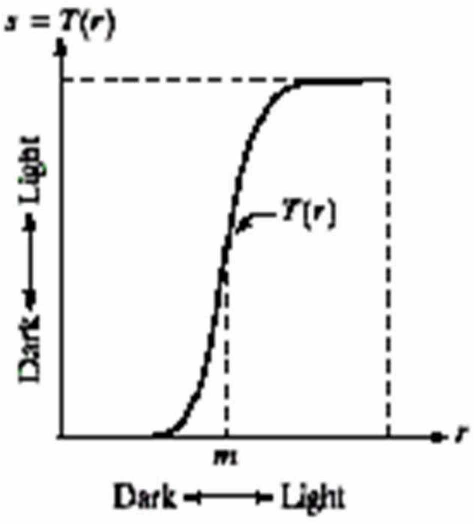
- Spatial domain methods
  - operate directly on the pixels $g(x,y)=T[f(x,y)]$
    - T operates over some neighborhood of (x,y)
      - neighborhood shape: square & rectangular arrays are the most predominant due to the ease of implementation
        - mask processing/filtering
        - masks/filters/kernels/templates/windows
        - e.g., image sharpening
      - the center of the window moves from pixel to pixel
      - the simplest form: gray-level transformation
        - $s=T(r)$
    - *T* can operate on a set of input images
      - e.g., sum of input images for noise reduction

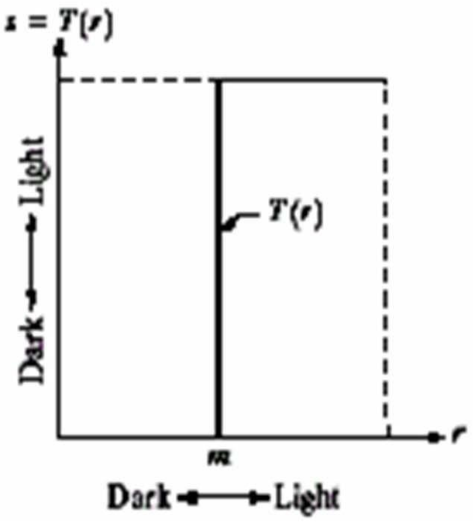


Point Processing

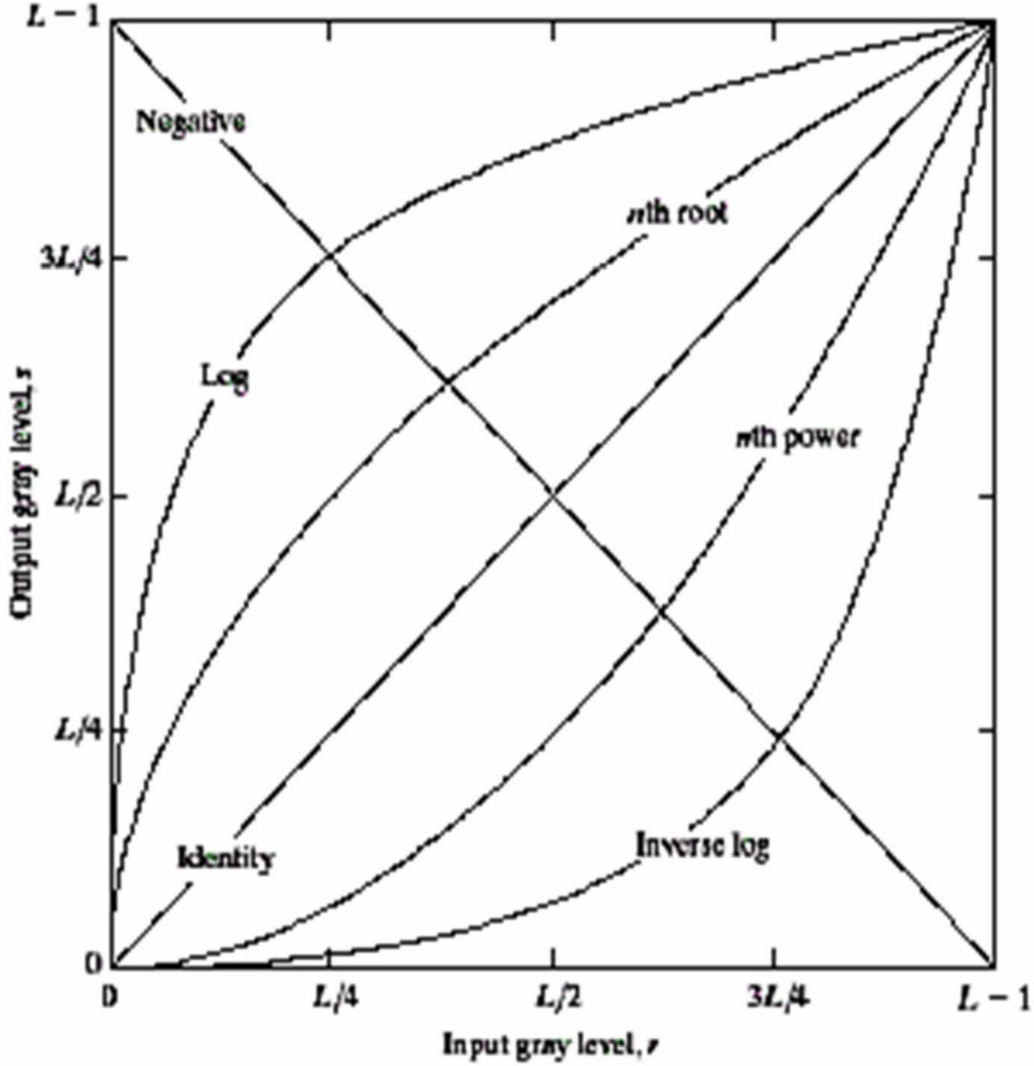
- Examples
  - contrast stretching: produce an image of higher contrast than the original
    - darken the levels below m & brighten the levels above m
  - the limiting case of contrast stretching: thresholding function

Contrast Stretching	Thresholding Function

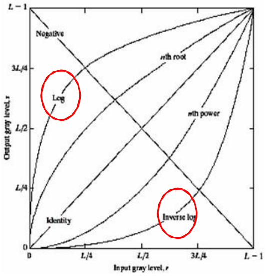




Some Basic Gray Level Transformations

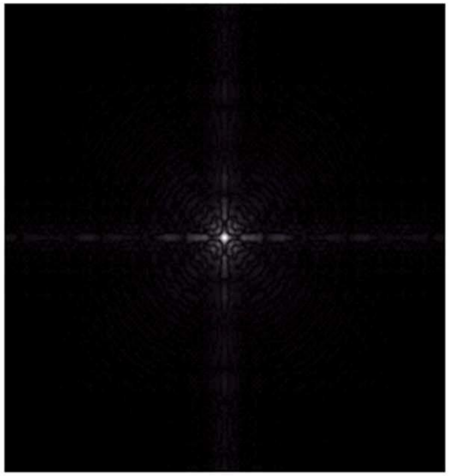


Gray Level Transformations – Log Transformations

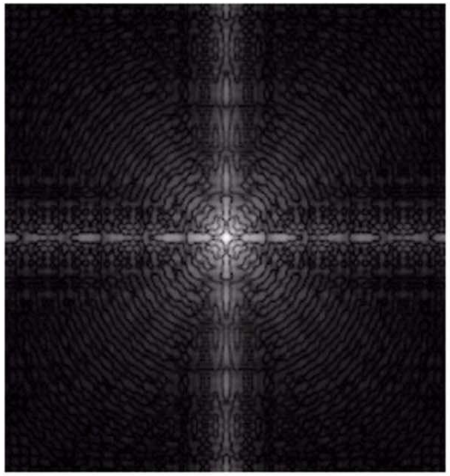


- $s = c\log(1+r)$, c: constant, $r \geq 0$
- Map a narrow range of low gray-level values in the input image into a wider range of output levels
- Compress the higher-level values
- An example: Fourier spectrum of wide range ($0 \sim 10^6$)
  - a significant degree of detail would lost in the display

Display in a 8-bit

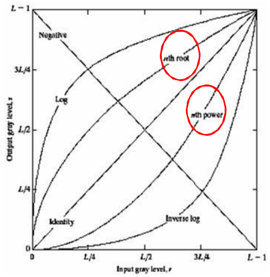


Fourier spectrum
($0 \sim 1.5 \times 10^6$)

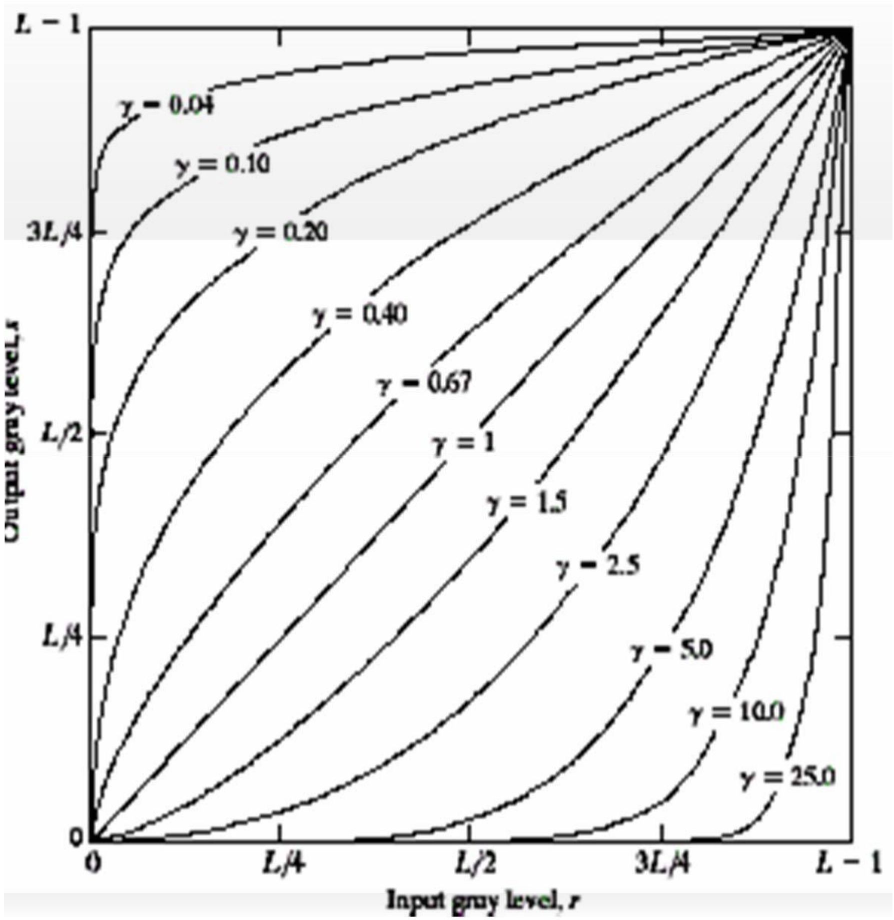


After log transformation
(c=1→ s:0~6.2)

Gray Level Transformations – Power-Law Transformations



- $s = cr^{\gamma}$, c & γ: positive constants
  - fractional γ : map a narrow range of dark input values into a wider range of output values
  - a family of possible transformation curves: varying γ
- A variety of devices used for image capture, printing & display respond according to a power law
  - device-dependent value of gamma



Example #1 of Power-Law Transformations – Gamma Correction

- Correct the power-law response phenomena
  - Example: The intensity-to-voltage response of CRT devices is a power function (exponents : 1.8~2.5)
    - darker darker than intended

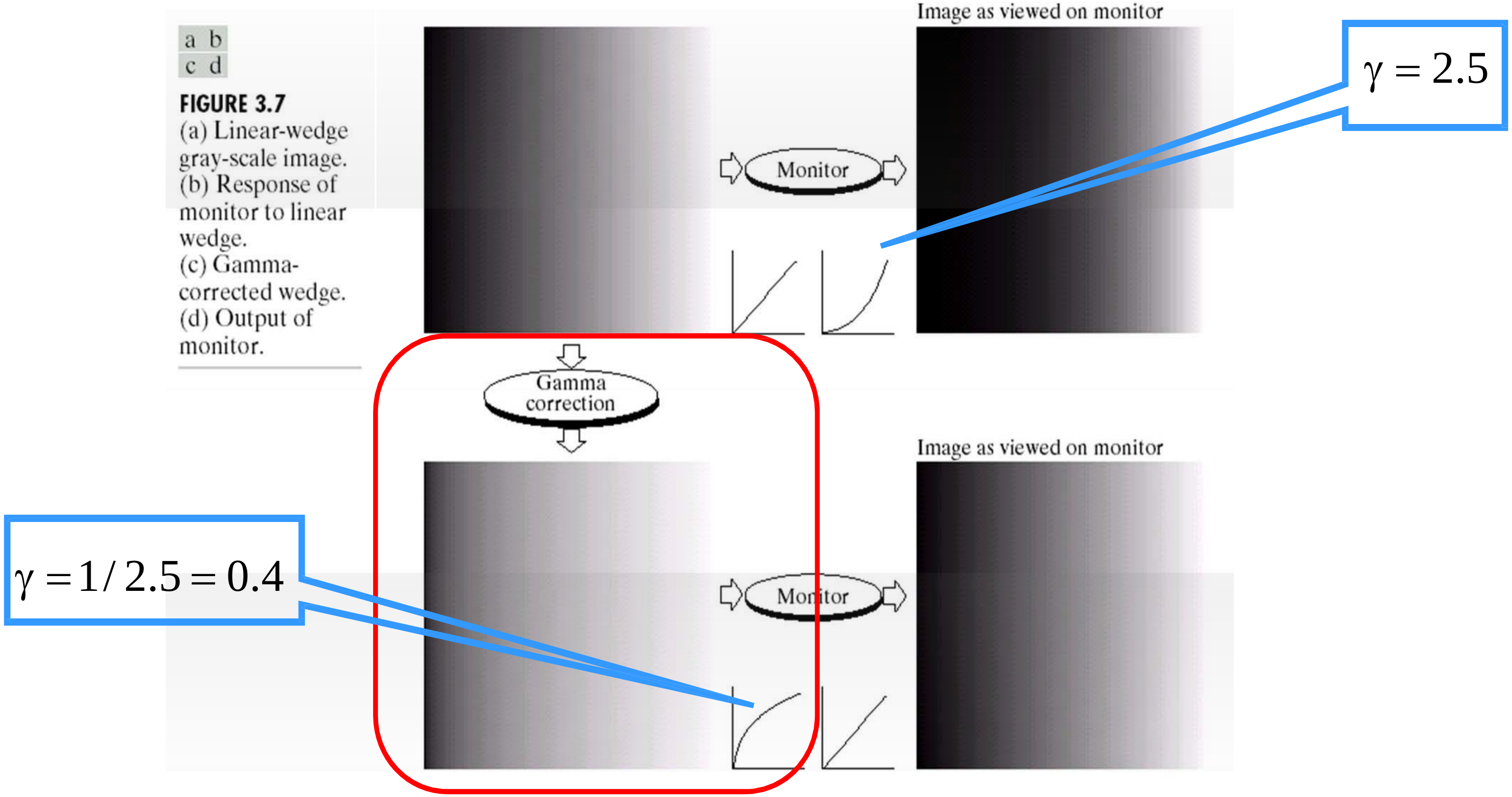
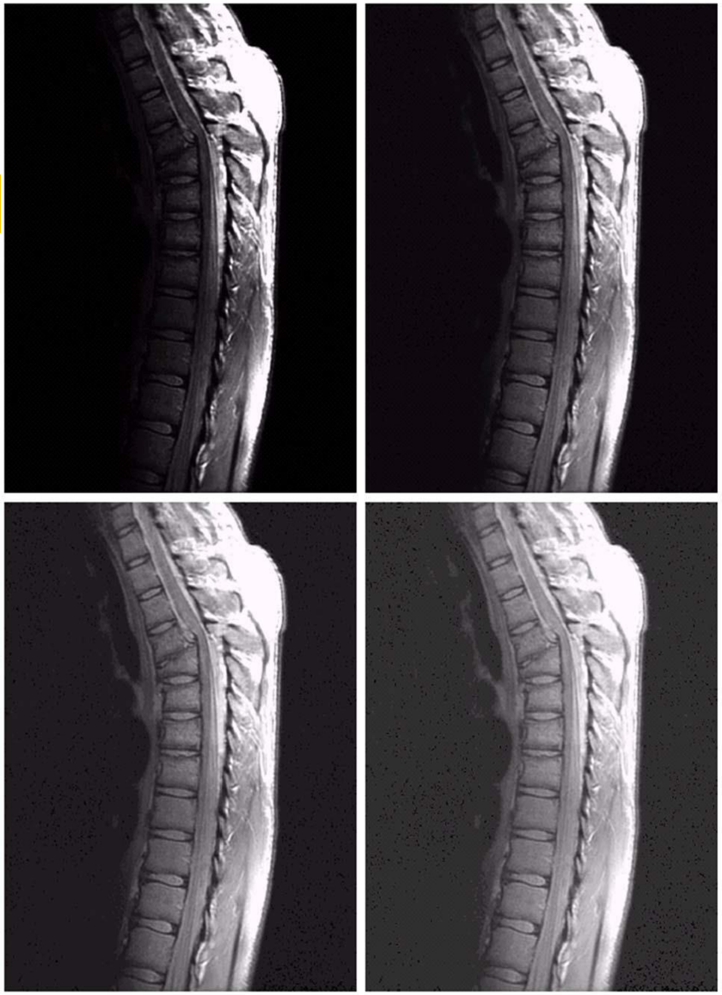


FIGURE 3.7 (a) Linear-wedge gray-scale image. (b) Response of monitor to linear wedge. (c) Gamma-corrected wedge. (d) Output of monitor.

Example #2 of Power-Law Transformations – Contrast Manipulation

Original image

$c=1, \gamma=0.6$



✓$c=1, \gamma=0.4$

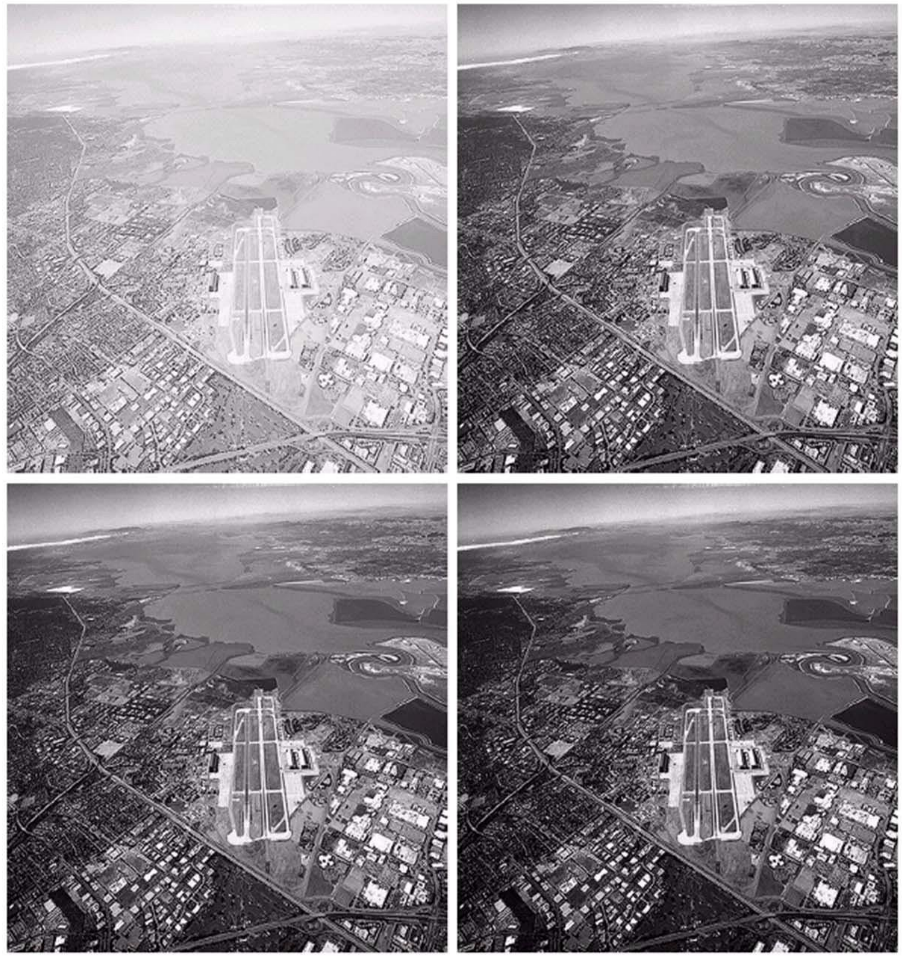
$c=1, \gamma=0.3$

Magnetic resonance (MR) image of a fractured human spine

Example #3 of Power-Law Trans formations – Contrast Manipulation

Original image

$c=1, \gamma=3.0$



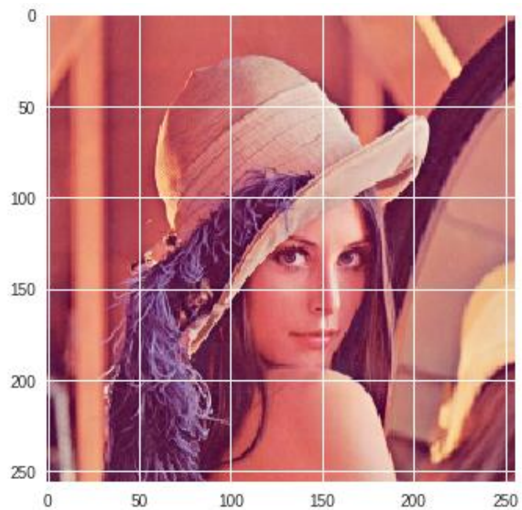
✓$c=1, \gamma=4.0$

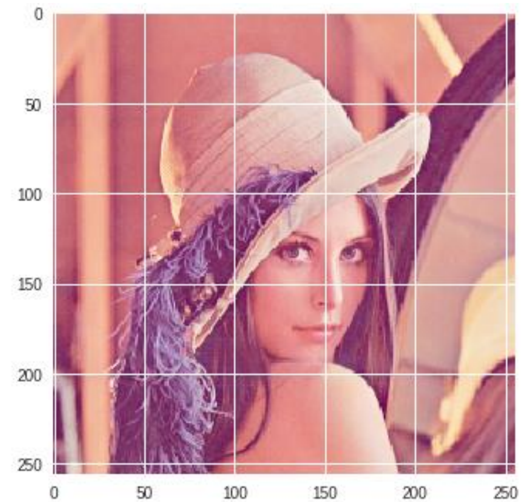
$c=1, \gamma=5.0$

A Washed-out appearance: a compression of gray levels is desirable

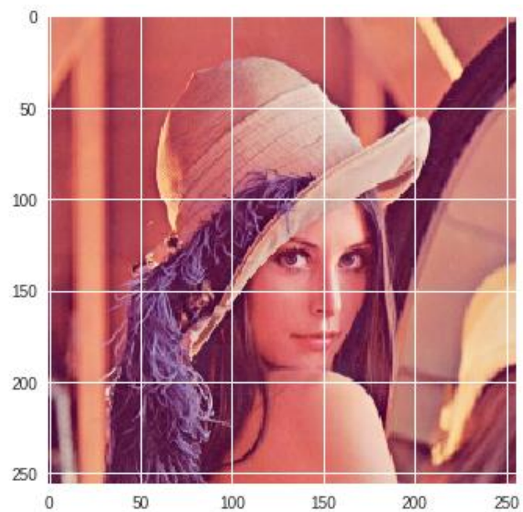
Example: Gamma_Correction

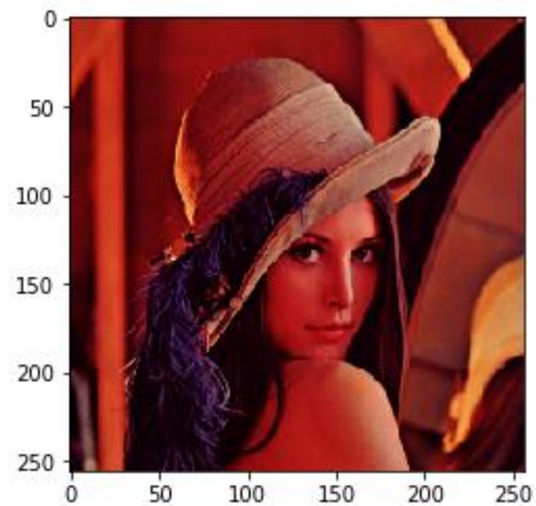
γ=0.67



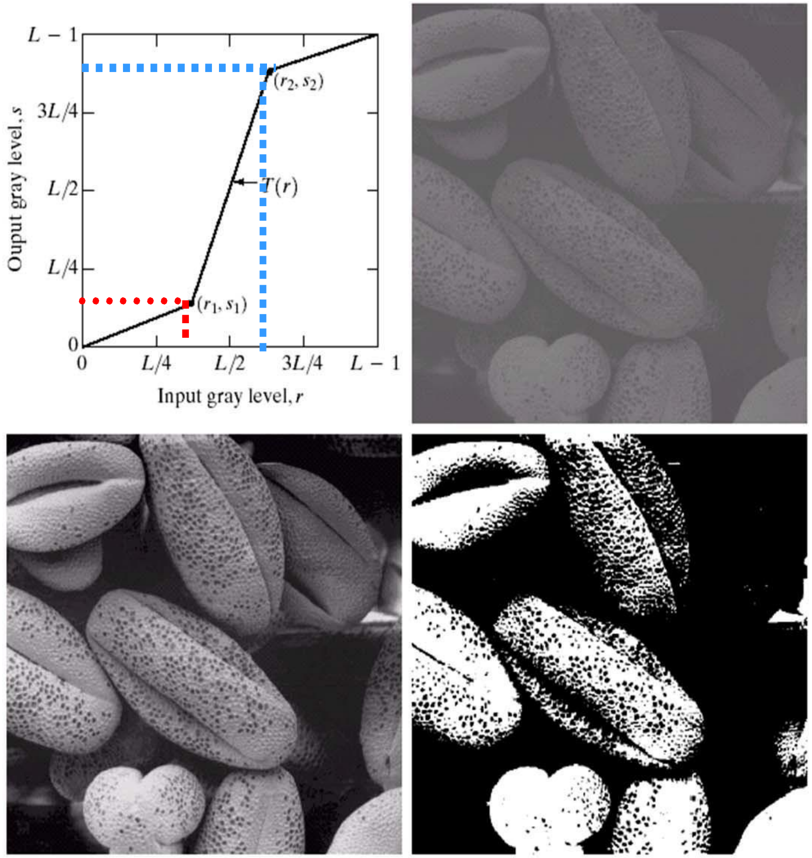


γ=2.67





Piecewise-Linear Transformation Functions

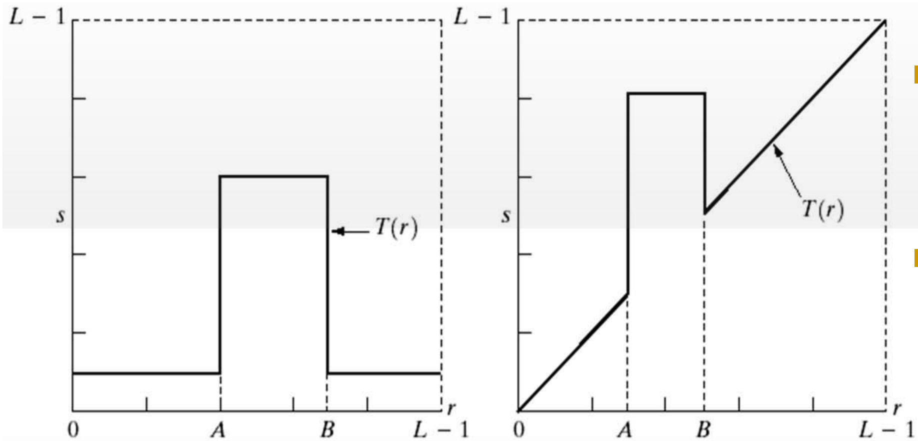


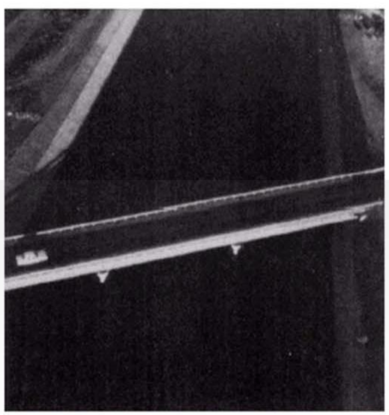
$(r_1,s_1)=(r_{min},0)$
$(r_2,s_2)=(r_{max},L-1)$

$r_1=r_2=m$
(Thresholding function)

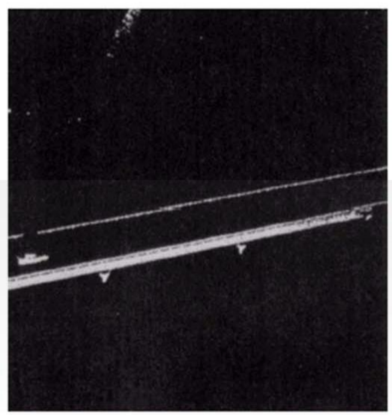
- Advantage: arbitrarily complex
- Disadvantage: the specification requires considerably more user inputs
- Example: contrast-stretching transformation
  - increase the dynamic range of the gray levels of the input image

Gray-Level Slicing





Original image

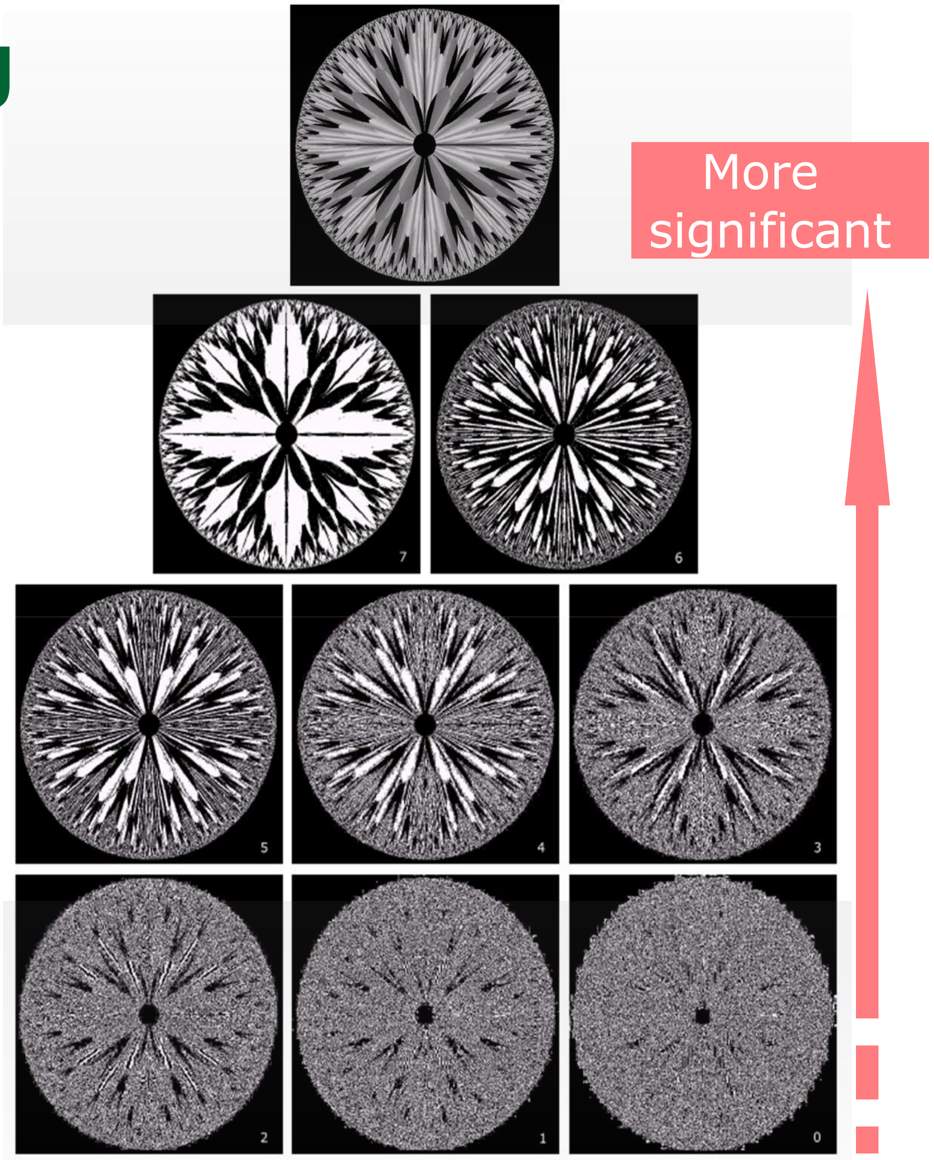


Transformed image by (a)

- Highlight a specific range of gray levels is often desired
- Various ways to accomplish this
  - highlight some range and reduce all others to a constant level
  - Highlight some range but preserve all other levels

Bit-plane Slicing

- Instead of highlighting gray-level ranges, highlighting the contribution made to total image appearance by specific bits
- Higher-order bits: the majority of the visually significant data
- Lower-order bits: subtle details
- Obtain bit-plane images
  - thresholding gray-level transformation function

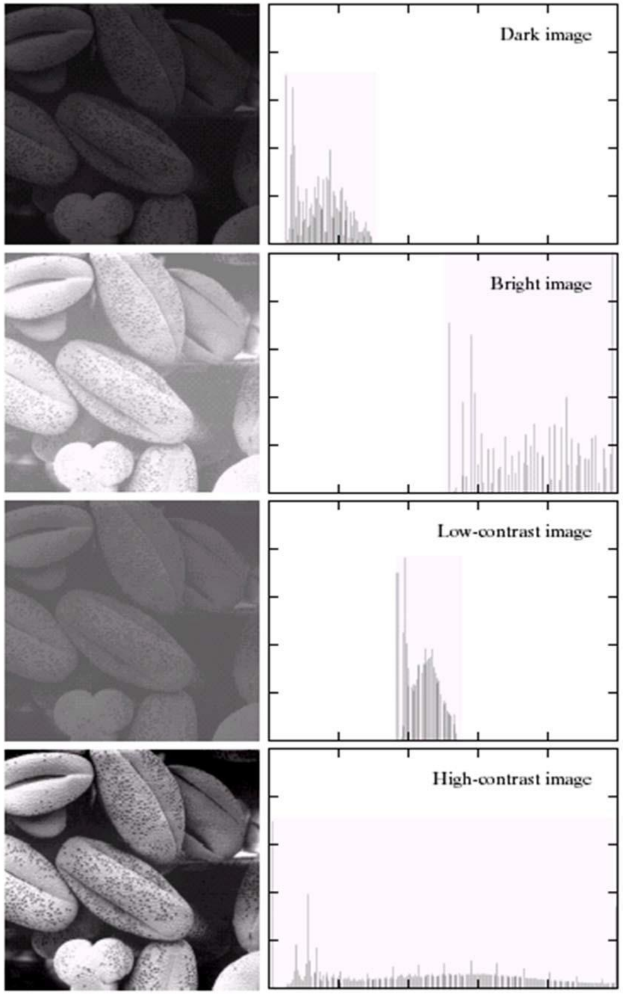


Outline

- Gray level transformations
- **Histogram processing**
- Spatial filtering

Histogram Processing

- Histogram: A plot r_k vs. $H(r_k) = n_k$
  - r_k : kth gray level
  - n_k: number of pixels in the image having gray level r_k
  - $k=0,1,..,L-1$
- Normalized histogram $p(r_k) = n_k / n$
- Purposes
  - image enhancement
  - image compression
  - segmentation
- Simple to calculate
  - economic hardware implementation
  - proper for real-time image processing

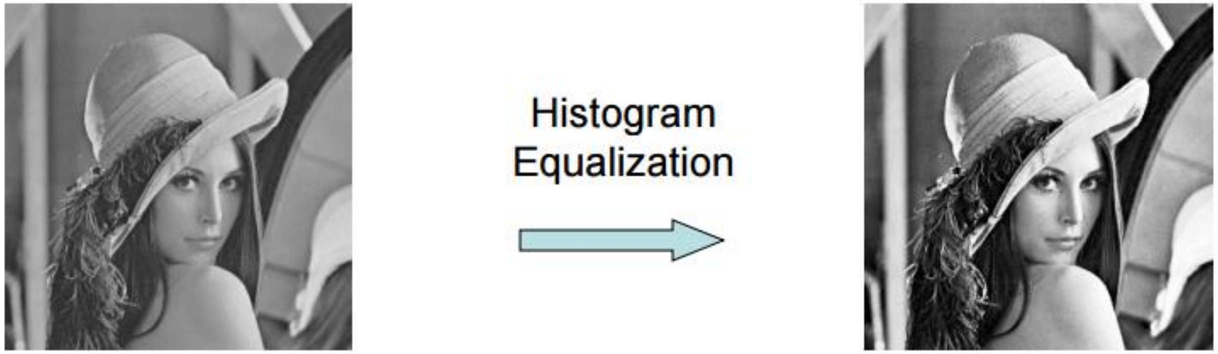


Histogram Equalization

- 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

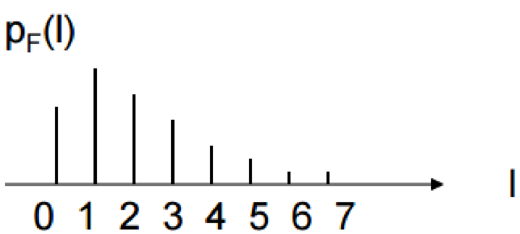
- For a discrete image f which takes values k=0,…,K-1, use $$\tilde{g}(l)=\sum_{k=0}^{l} p_F(k), l=0,1,\ldots,K-1.$$
- To convert the transformed values to the range of (0, L-1):

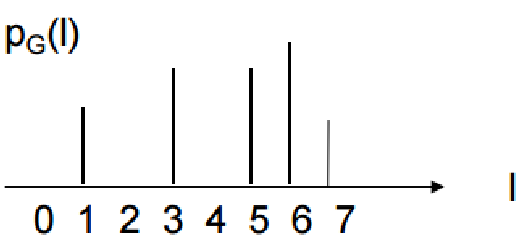
$$g(l)=round\left\{\left(\sum_{k=0}^{l} p_F(k)\right)*(L-1)\right\}$$

  – Note: {x} is the rounding of x

Example

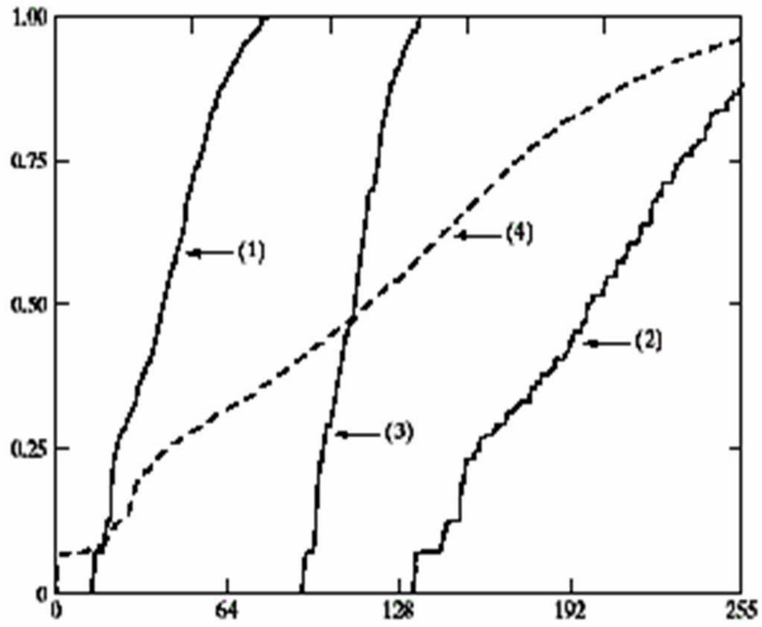
f_k	$p_F(l)$	$\tilde{g}_l = \sum_{k=0}^{l} p_F(k)$	$g_l = [\tilde{g}_l * 7]$	$p_G(l)$	g_k
0	0.19	0.19	[1.33]=1	0	0
1	0.25	0.44	[3.08]=3	0.19	1
2	0.21	0.65	[4.55]=5	0	2
3	0.16	0.81	[5.67]=6	0.25	3
4	0.08	0.89	[6.03]=6	0	4
5	0.06	0.95	[6.65]=7	0.21	5
6	0.03	0.98	[6.86]=7	0.16+0.08=0.24	6
7	0.02	1.00	[7]=7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

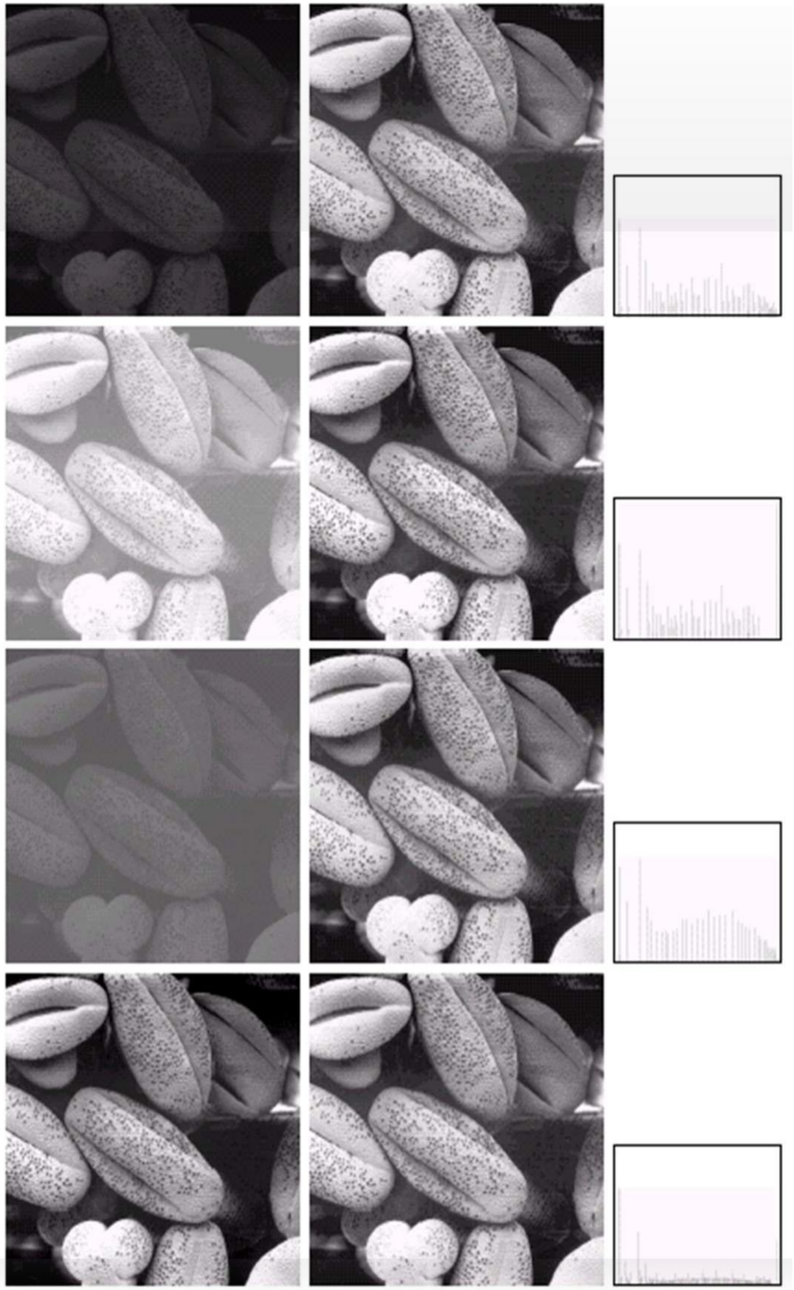




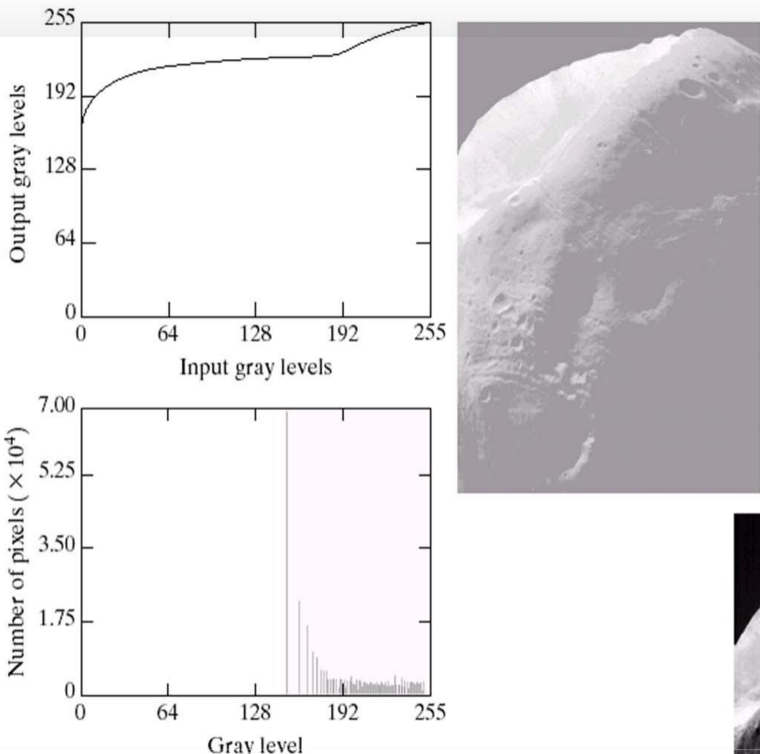
Results of Histogram Equalization (1/2)

Transformation functions



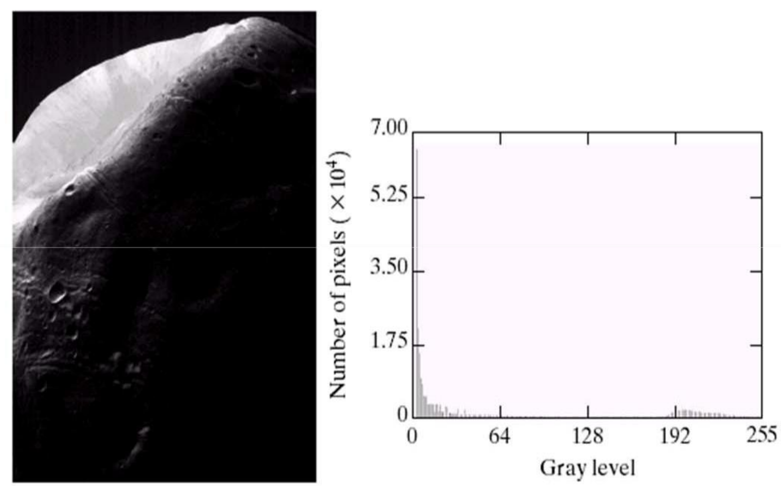


Results of Histogram Equalization (2/2)



a b
c

FIGURE 3.21 (a) Transformation function for histogram equalization. (b) Histogram-equalized image (note the washed-out appearance). (c) Histogram of (b).

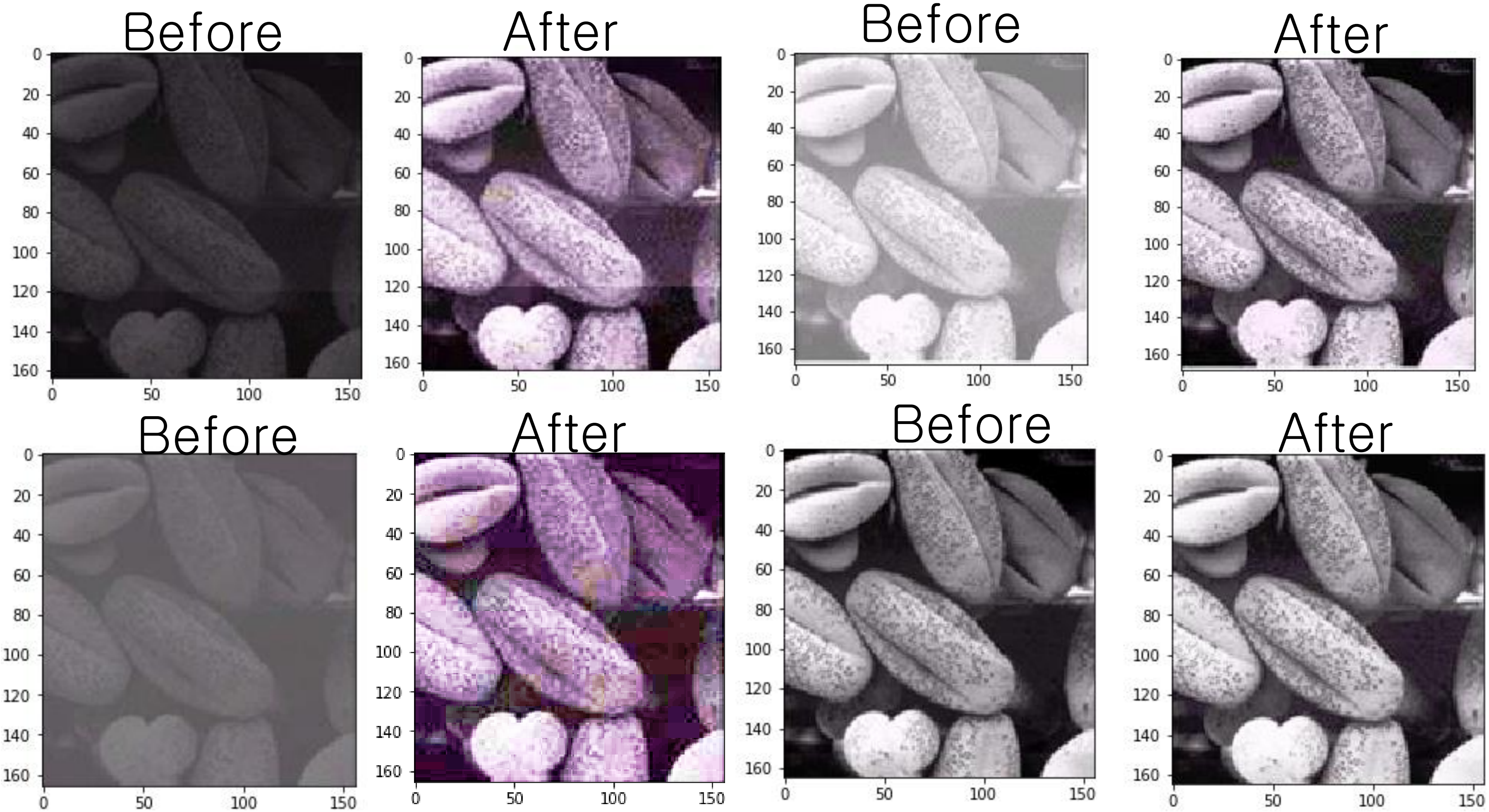


a b

FIGURE 3.20 (a) Image of the Mars moon Photos taken by NASA's *Mars Global Surveyor.* (b) Histogram. (Original image courtesy of NASA.)

OpenCV Example 4.2: Histogram Equalization

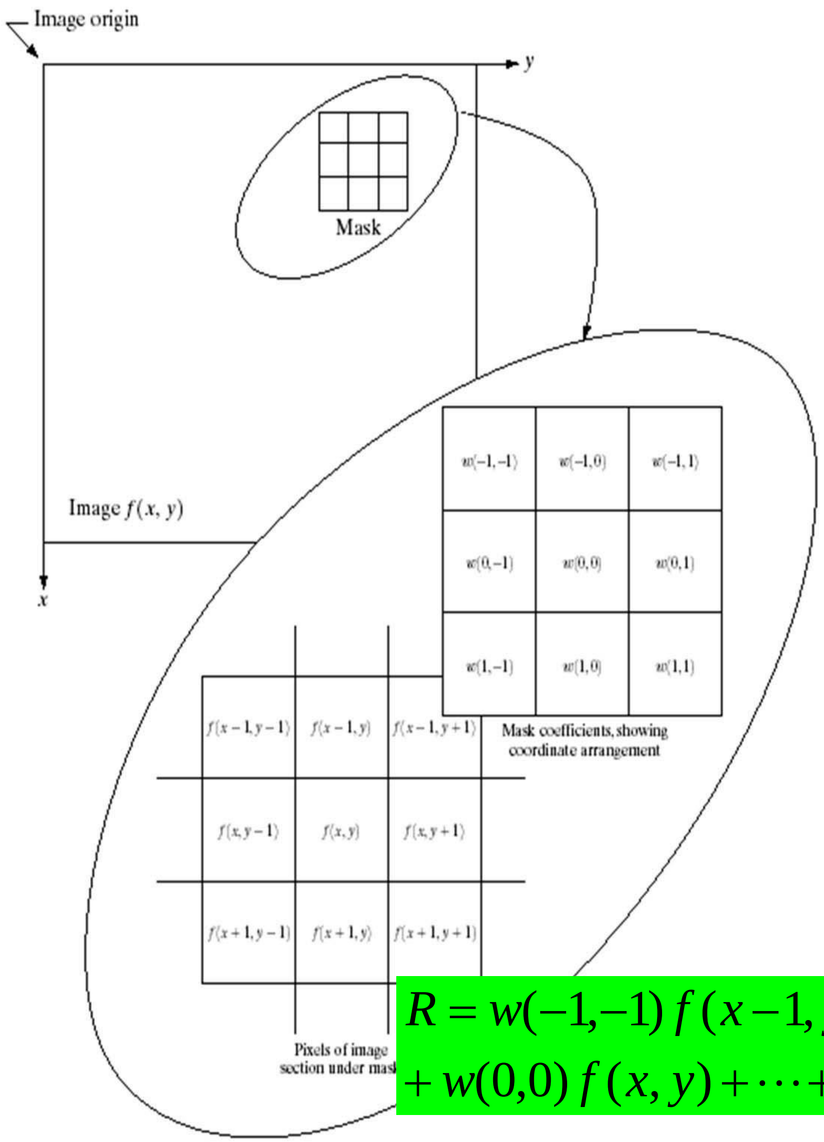
- The four processed results are not identical. Do you know why?



Outline

- Gray level transformations
- Histogram processing
- **Spatial filtering**

Basics of Spatial Filtering (1/2)



- The concept of filtering comes from the use of the Fourier transform for signal processing in the *frequency domain.*
- At each point (x,y), the response of the filter at that point is calculated using a predefined relationship.
- Linear spatial filtering: the response is the sum of products

$$R = w(-1,-1)f(x-1,y-1) + w(-1,0)f(x-1,y) + \ldots$$
$$+ w(0,0)f(x,y) + \cdots + w(1,0)f(x+1,y) + w(1,1)f(x+1,y+1)$$

Basics of Spatial Filtering (2/2)

- Linear spatial filtering: convolving a mask with an image.
  - **filtering mask / convolution mask / convolution kernel**
- Nonlinear spatial filtering
  - the filtering operation is based conditionally on the values of the pixels in the neighborhood under consideration, but not explicitly use coefficients in the sum-of-products manner
  - e.g., median filter for noise reduction: compute the median gray-level value in the neighborhood

When the Center of the Filter Approaches the Border of the Image？

- One or more rows/columns of the mask will be located outside the image plane
- Solution
  - limit the excursions of the center of the mask to be at a distance no less than $(n\text{-}1)/2$ pixels from the border
    - the resulting filtered image is smaller than the original
  - partial filter mask: filter all pixels only with the section of the mask that is fully contained in the image
  - Padding
    - add rows & columns of 0's (or other constant gray level), or replicate rows/columns
    - the padding is stripped off at the end of the process
    - side effect: an effect near the edges that becomes more prevalent as the size of the mask increases

Smoothing Linear Filters (1/2)

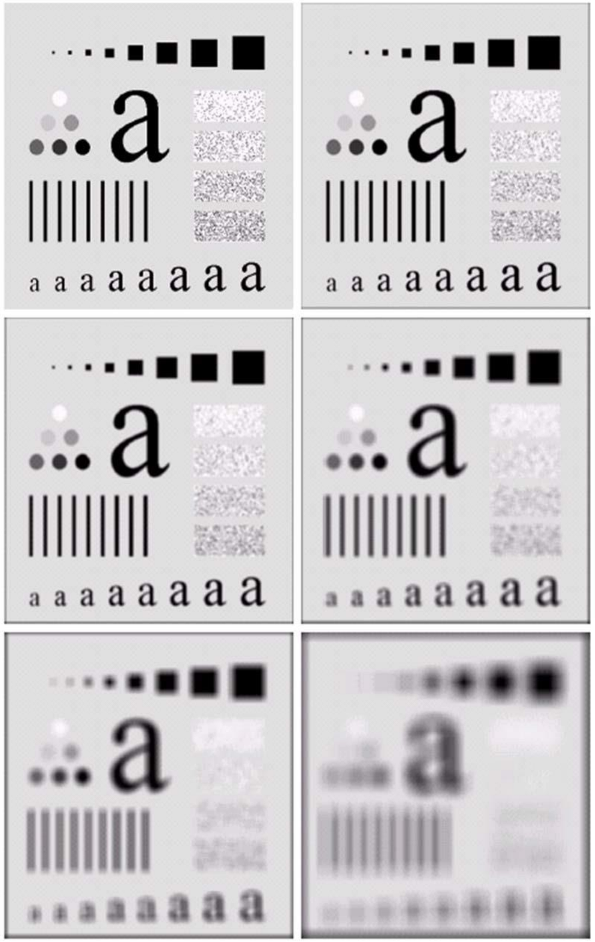
- Averaging filters/low pass filters: replace the value of every pixel in an image by the average of the gray levels of the pixels contained in the neighborhood of the filter mask
- Reduce sharp transitions in gray levels
  - side effect: blur edges
- Reduce irrelevant detail in an image and get a gross representation of objects of interest
  - irrelevant: pixel regions that are small with respect to the size of the filter mask
- Applications
  - noise reduction
  - smooth false contouring

Smoothing Linear Filters (2/2)

$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad \frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

- Computationally efficient: instead of being 1/9 (1/16), the coefficients of the filter are all 1's
- Box filter: a spatial averaging filter in which all coefficients are equal
- Weighted average: give more importance (weight) to some pixels
  - reduce blurring in the smoothing process
    - the pixel at the center of the mask is multiplied by a higher value than any other
    - the other pixels are inversely weighted as a function of their distance from the center of the mask
  - The attractive feature of "16": integer power of 2 for computer implementation

Examples of Smoothing Linear Filters (1/2)



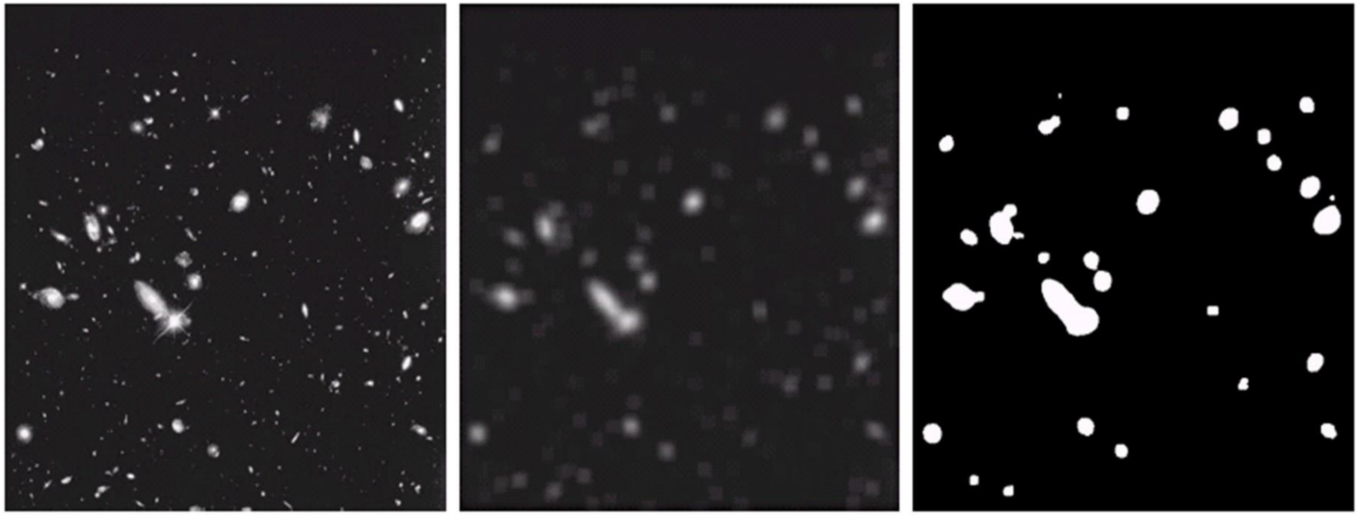
- $n=3$: general slight blurring through the entire image, but details are of approximately the same size as the filter mask are affected considerably more
- $n=9$: the significant further smoothing of the noisy rectangles
- $n=15$ & 35: generally eliminate small objects from an image
- $n=35$: black order due to zero padding

a	b
c	d
e	f

FIGURE 3.35 (a) Original image, of size 500 × 500 pixels. (b)–(f) Results of smoothing with square averaging filter masks of sizes $n = 3, 5, 9, 15$, and 35, respectively. The black squares at the top are of sizes 3, 5, 9, 15, 25, 35, 45, and 55 pixels, respectively; their borders are 25 pixels apart. The letters at the bottom range in size from 10 to 24 points, in increments of 2 points; the large letter at the top is 60 points. The vertical bars are 5 pixels wide and 100 pixels high; their separation is 20 pixels. The diameter of the circles is 25 pixels, and their borders are 15 pixels apart; their gray levels range from 0% to 100% black in increments of 20%. The background of the image is 10% black. The noisy rectangles are of size 50 × 120 pixels.

Examples of Smoothing Linear Filters (2/2)

- Goal: get a gross representation of objects of interest, such that the intensity of smaller objects blends with the background and larger objects become bloblike
  - the size of the mask
- Smoothing linear filtering + thresholding to eliminate objects based on their intensity



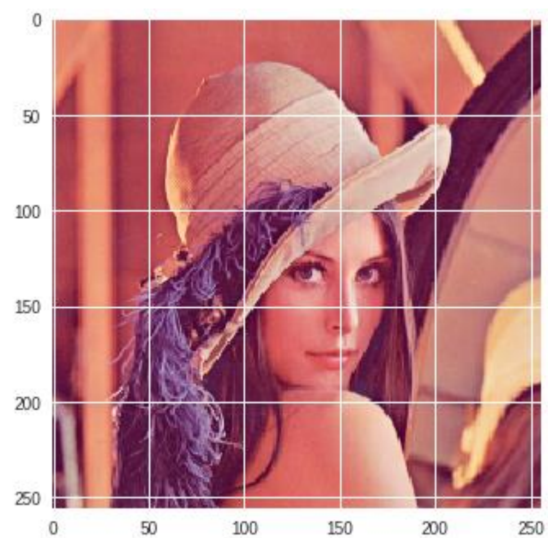
a b c

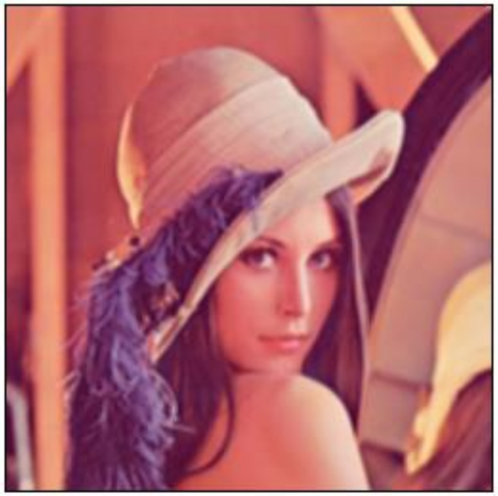
FIGURE 3.36 (a) Image from the Hubble Space Telescope. (b) Image processed by a 15 × 15 averaging mask. (c) Result of thresholding (b). (Original image courtesy of NASA.)

Example: Smoothing Filtering

- How do you feel about the following implementation for a 3x3 convolution?

```
for k in range(c):
    for i in range(1,h-1):
        for j in range(1,w-1):
            #print(k)
            top_left = img_rgb.item(i-1,j-1,k)
            top = img_rgb.item(i-1,j,k)
            top_right = img_rgb.item(i-1,j+1,k)

            center_left = img_rgb.item(i,j-1,k)
            center = img_rgb.item(i,j,k)
            center_right = img_rgb.item(i,j+1,k)

            btm_left = img_rgb.item(i+1,j-1,k)
            btm = img_rgb.item(i+1,j,k)
            btm_right = img_rgb.item(i+1,j+1,k)

            averaged_value = (top_left + top + top_right + center_left + center + center_right + btm_left + btm + btm_right)/9

            new_img_rgb.itemset((i,j,k),averaged_value)
```



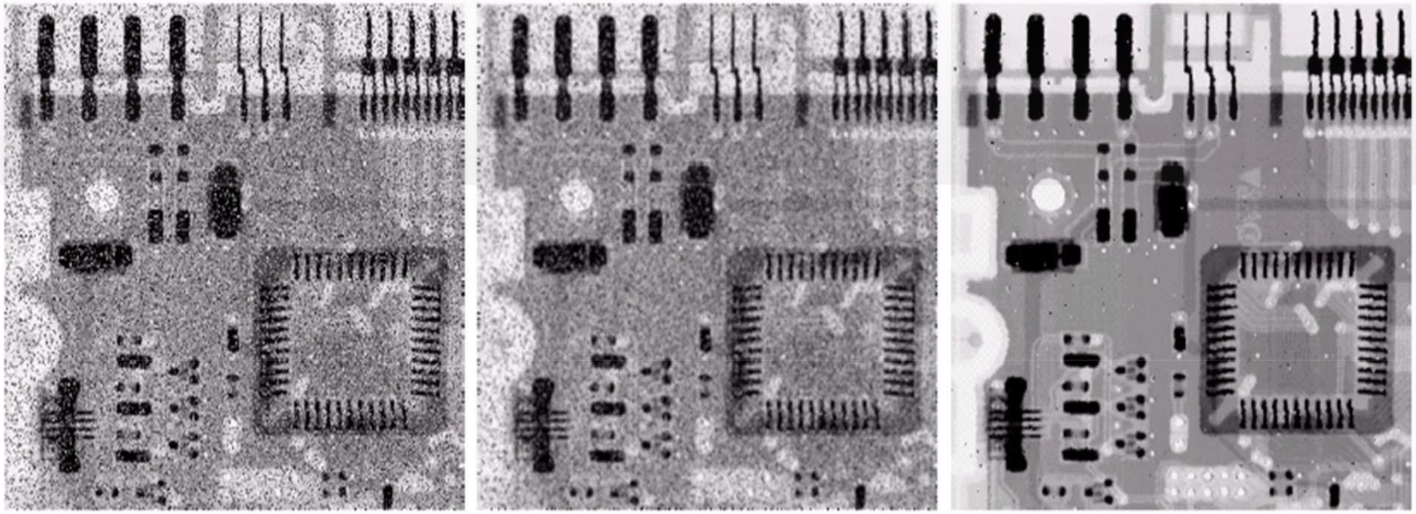


Order-Statistics Filters

- Nonlinear spatial filters whose response is based on ordering (ranking) the pixels in the area to be filtered
- Examples: max filter, min filter & median filter
- Median filter
  - replace the value of a pixel by the median of the gray levels in the neighborhood of that pixel
  - force points with distinct gray levels to be more like their neighbors
  - for 3x3 neighborhood (10,20,20,20,15,20,20,25,100) -> median is 20
  - excellent noise-reduction capabilities, with considerably less blurring than linear smoothing filters of similar size
  - particular effective for impulse noise (salt-and-pepper noise

An Example – Applying Median Filters

- The image processed with the averaging filter has less visible noise, but the price paid is significant



a b c

FIGURE 3.37 (a) X-ray image of circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a 3 × 3 averaging mask. (c) Noise reduction with a 3 × 3 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

Example: Median Filter

- Do you know why some salt and pepper are still there?
- Do you have any idea what part could be sped-up?