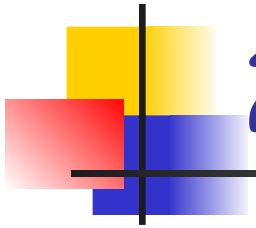
Image Enhancement in the Spatial Domain

Principle Objective of Enhancement

- Enhancement refers to accentuation, or sharpening, of image features such as edges, boundaries, or contrast to make graphic display more useful for display and analysis.
- Does not increase the inherent information content in the data, only increase the dynamic range of chosen feature.

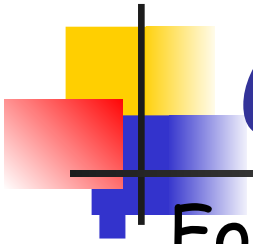
Principle Objective of Enhancement

- Process an image so that the result will be more suitable than the original image for a specific application.
- The suitableness is up to each application.
- A method which is quite useful for enhancing an image may not necessarily be the best approach for enhancing another images



2 domains

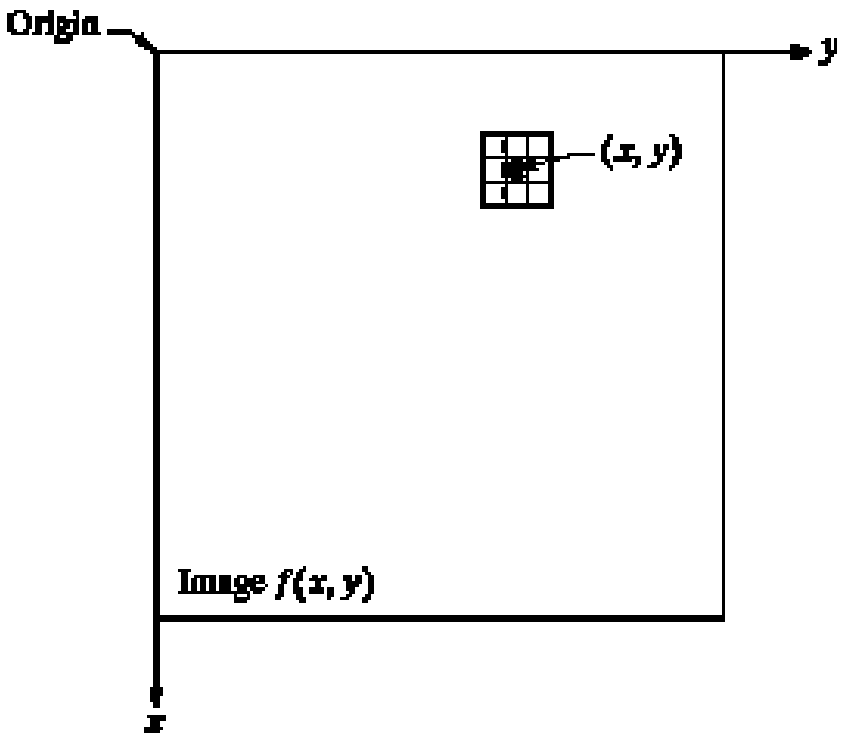
- Spatial Domain : (image plane)
  - Techniques are based on direct manipulation of pixels in an image
- Frequency Domain :
  - Techniques are based on modifying the Fourier transform of an image
- There are some enhancement techniques based on various combinations of methods from these two categories.



Good images

- For human visual
  - The visual evaluation of image quality is a highly subjective process.
  - It is hard to standardize the definition of a good image.
- For machine perception
  - The evaluation task is easier.
  - 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.

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 an operator on f defined over some neighborhood of (x,y)

Spatial Domain

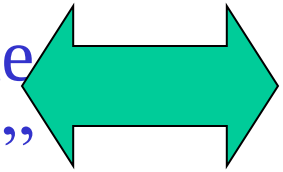
What is spatial domain

The space where all pixels form an image

In spatial domain we can represent an image by f(x,y) where x and y are coordinates along x and y axis with respect to an origin

There is duality between Spatial and Frequency Domains

Images in the spatial domain are pictures in the xy plane where the word "distance" is meaningful.

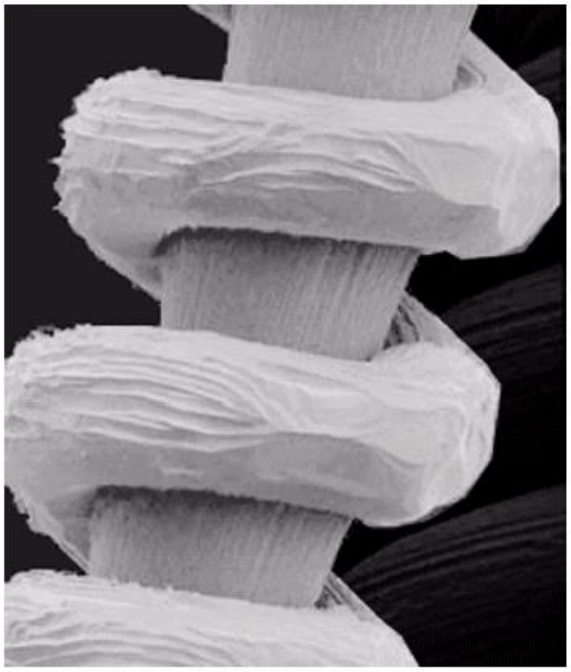


Using the Fourier transform, the word "distance" is lost but the word "frequency" becomes alive.

Image Enhancement

Image Enhancement means improvement of images to be suitable for specific applications.

Example:



Note: each image enhancement technique that is suitable for one application may not be suitable for other applications.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Image Enhancement Example

Original image

Enhanced image using Gamma correction

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Image Enhancement in the Spatial Domain

= Image enhancement using processes performed in the Spatial domain resulting in images in the Spatial domain.

We can written as $$g(x,y) = T[f(x,y)]$$

where f(x,y) is an original image, g(x,y) is an output and T[ ] is a function defined in the area around (x,y)

Note: T[ ] may have one input as a pixel value at (x,y) only or multiple inputs as pixels in neighbors of (x,y) depending in each function.

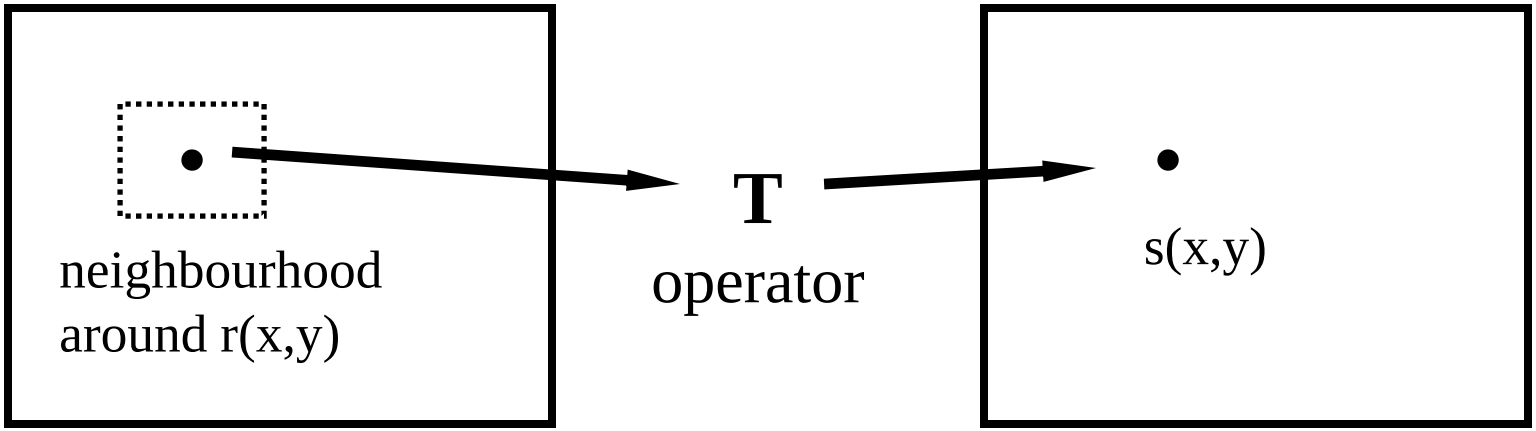
Ex. Contrast enhancement uses a pixel value at (x,y) only for an input while smoothing filte use several pixels around (x,y) as inputs.

Types of Image Enhancement in the Spatial Domain

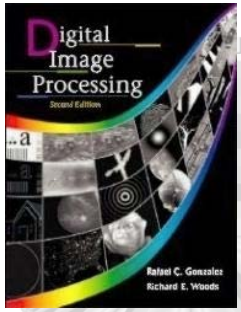
- Single pixel methods
    - Gray level transformations
      Example
        - Historgram equalization
        - Contrast stretching
    - Arithmetic/logic operations
      Examples
        - Image subtraction
        - Image averaging
- Multiple pixel methods
    Examples
        Spatial filtering
            - Smoothing filters
            - Sharpening filters

Spatial Domain Methods

The value of a pixel at location (x,y) in the enhanced image is the result of performing some operation on the pixels in the neighbourhood of (x,y) in the input image.



For computational reasons the neighbourhood is usually square but it can be any shape.



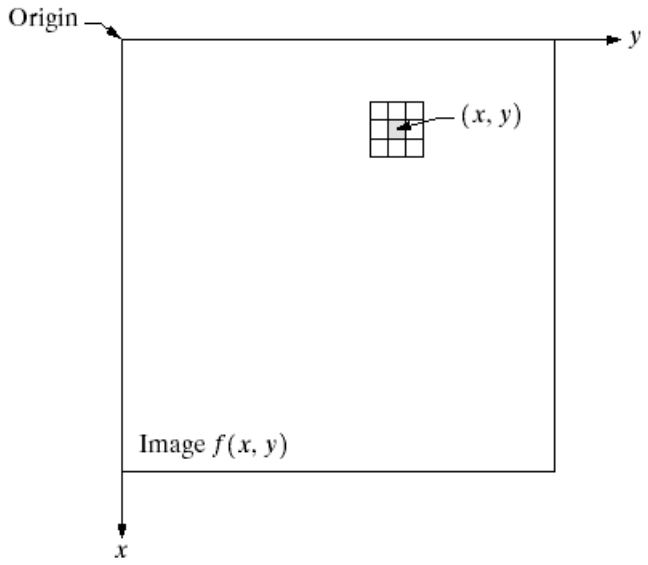
Chapter 3
Image Enhancement in the Spatial Domain

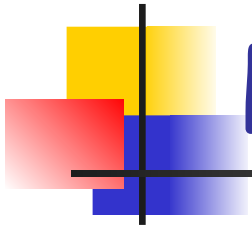
Spatial domain refers to the aggregate of pixels composing an image. Spatial domain methods are procedures that operate directly on these pixels.

$$g(x, y) = T[f(x, y)] \qquad (3.1\text{-}1)$$

where $f(x, y)$ is the input image, $g(x, y)$ is the processed image, and T is an operator on f, defined over some neighborhood of (x, y). In addition, T can operate on a *set* of input images, such as performing the pixel-by-pixel sum of K images for noise reduction.

FIGURE 3.1 A 3×3 neighborhood about a point (x, y) in an image.





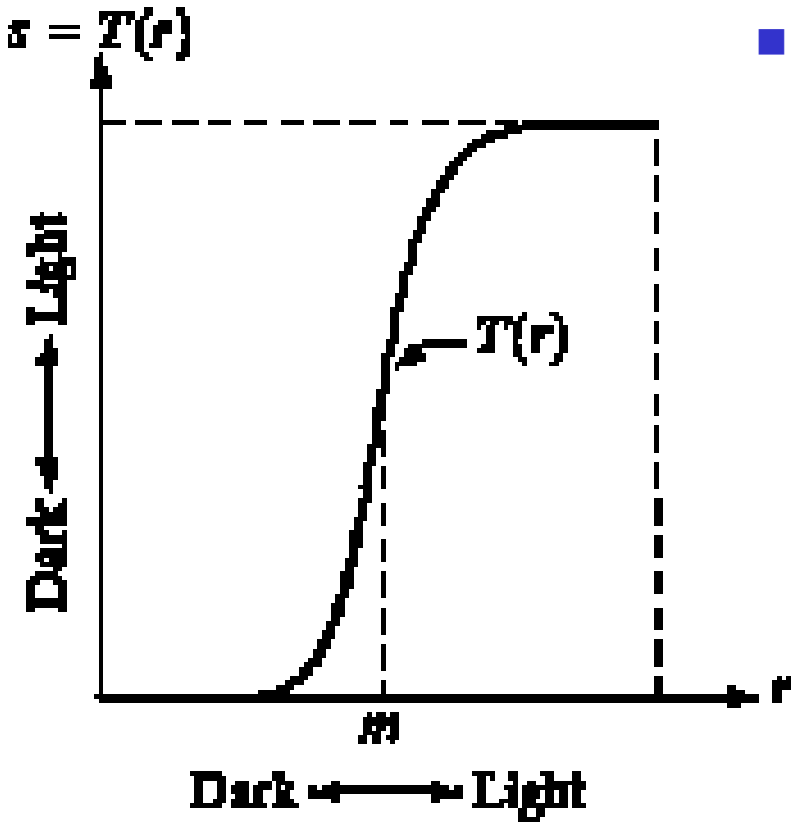
Point Processing

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

$$s = T(r)$$

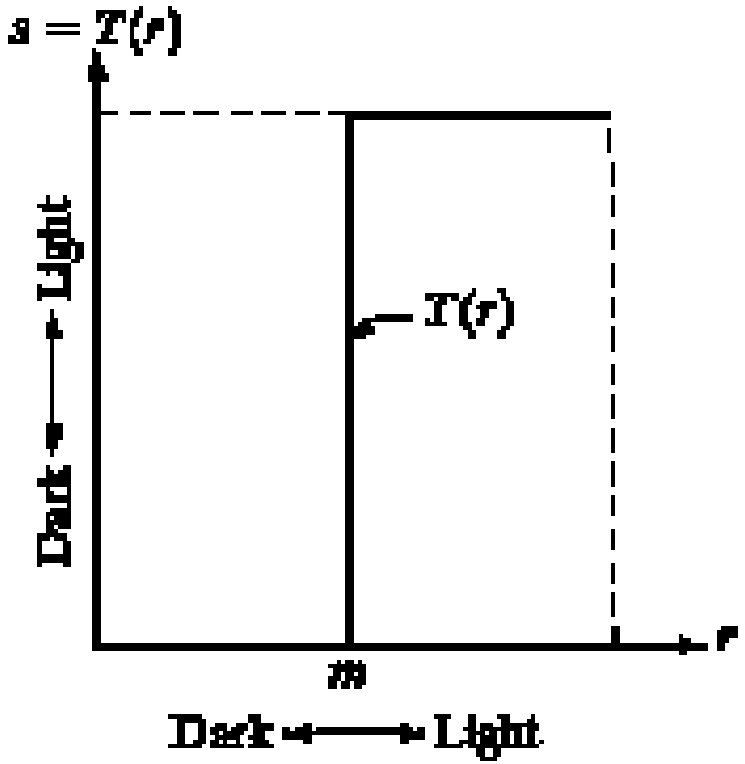
- Where
  - r = gray level of $f(x,y)$
  - s = gray level of $g(x,y)$

Contrast Stretching



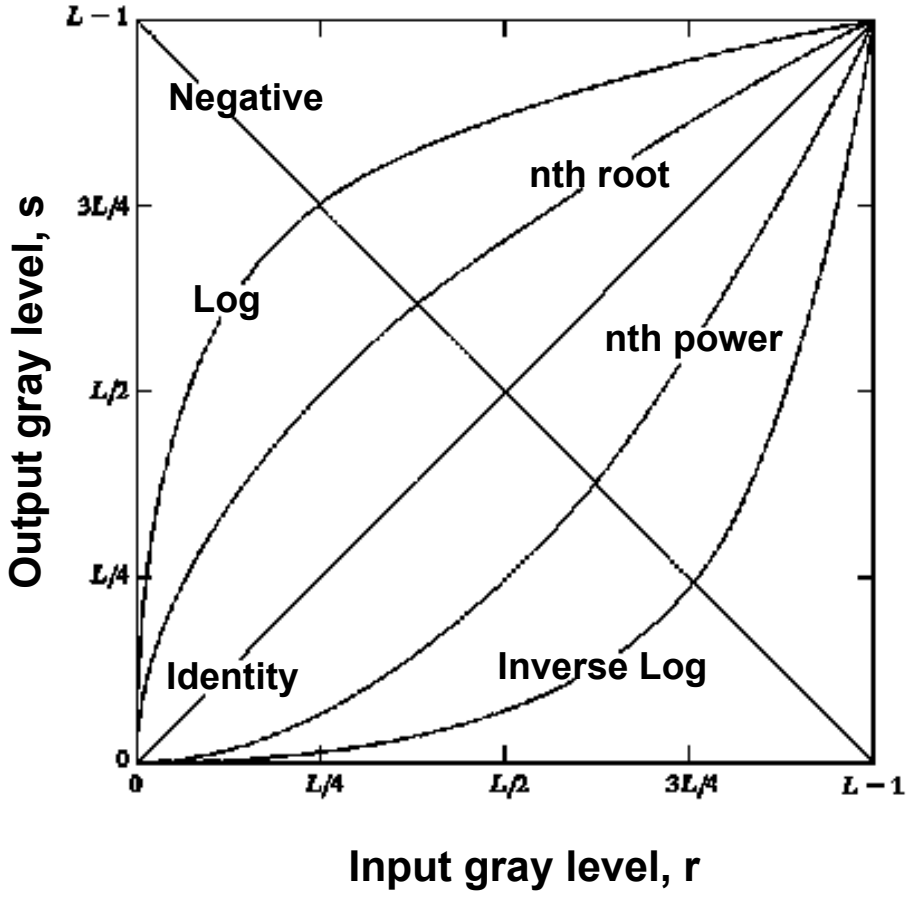
- Produce higher contrast than the original by
  - darkening the levels below m in the original image
  - Brightening the levels above m in the original image

Thresholding

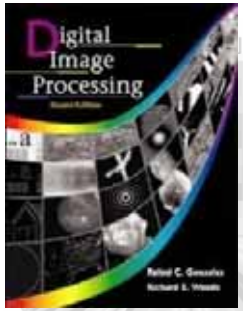


- Produce a two-level (binary) image

3 basic gray-level transformation functions



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



3.2 Basic gray level transformations

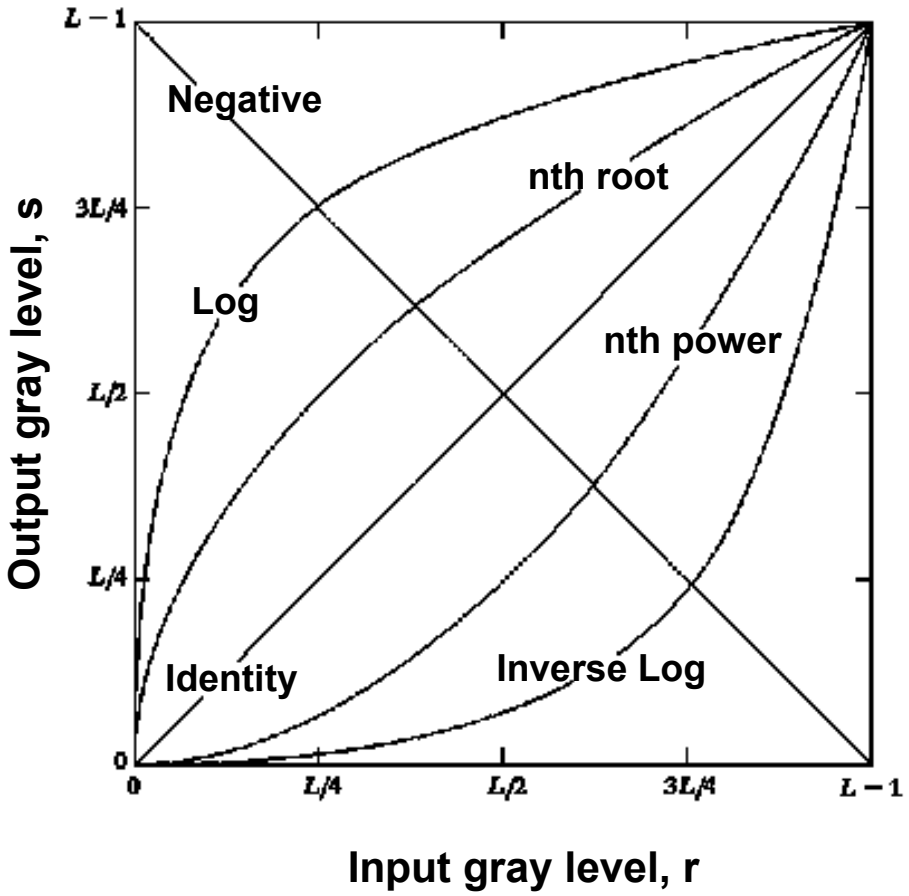
- Three basic functions used in image enhancement
  - Linear (negative and identity transformation)

    $s=L-1-r$
  - Logarithmic (log and inverse log)

    $s=c\ log(1+r)$
  - Power law (*n*th power and *n*th root transformation)

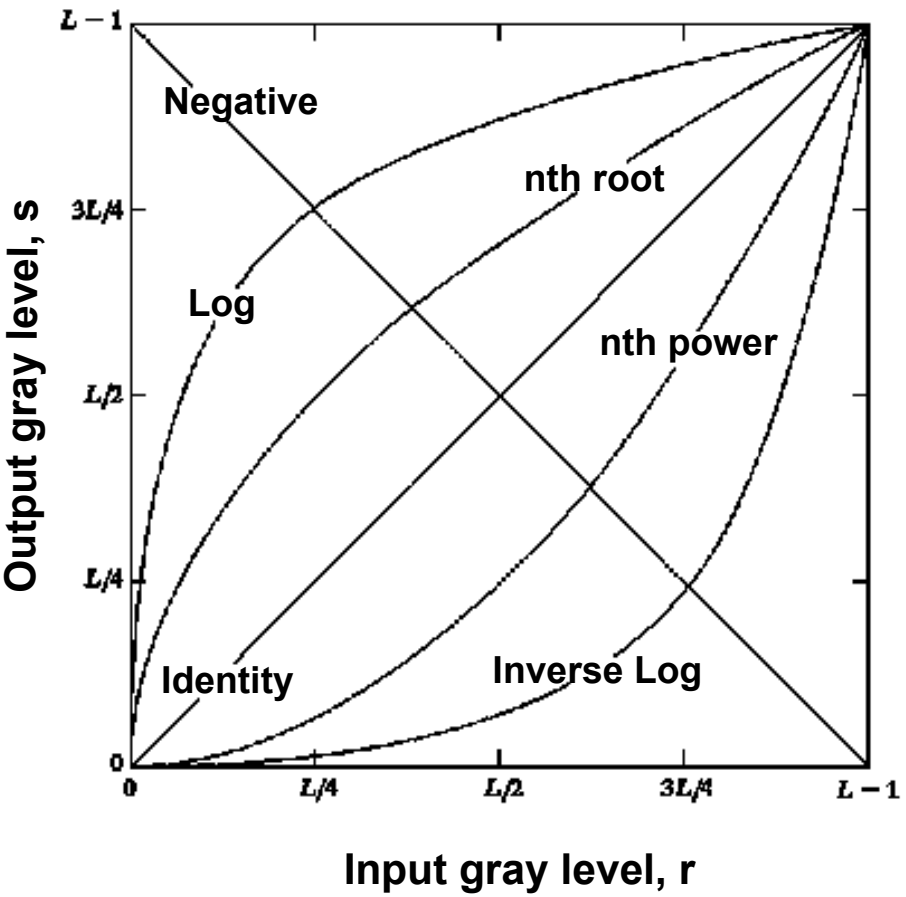
    $s=cr^{\gamma}$ *or* $s=c(r+\varepsilon)^{\gamma}$

Identity function



- Output intensities are identical to input intensities.
- Is included in the graph only for completeness.

Image Negatives



- An image with gray level in the range $[0, L-1]$ where $L = 2^n$; n = 1, 2…
- Negative transformation :

$$s = L - 1 - r$$

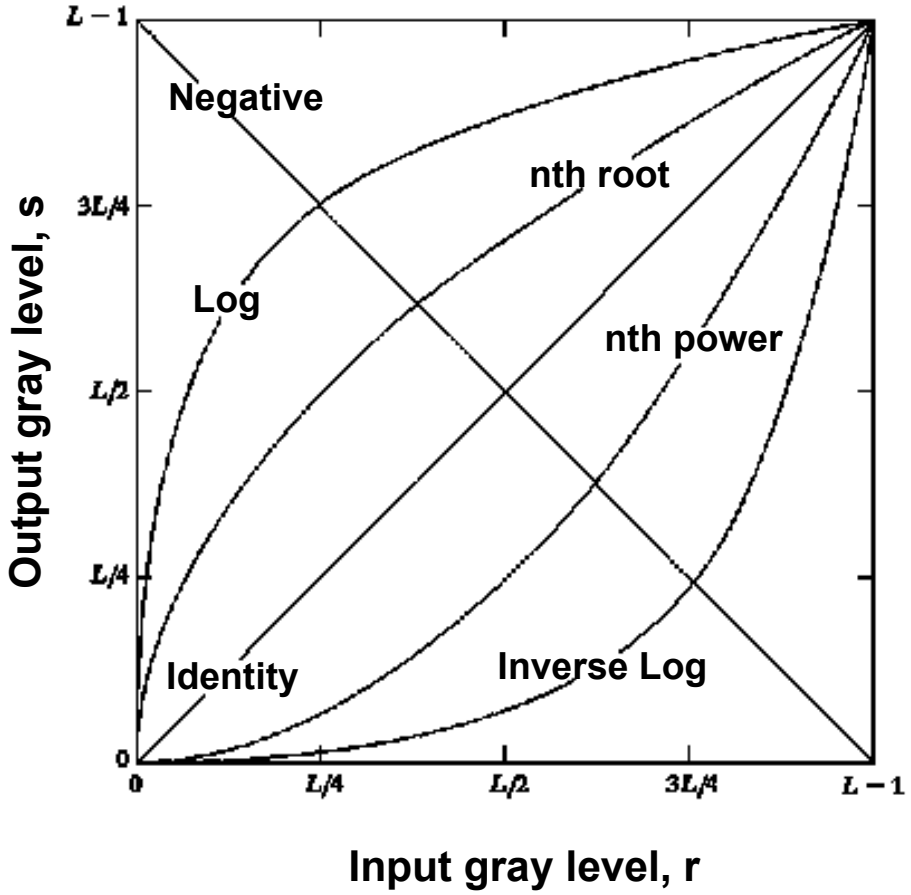
- Reversing the intensity levels of an image.
- Suitable for enhancing white or gray detail embedded in dark regions of an image, especially when the black area dominant in size.

Example of Negative Image

Original mammogram showing a small lesion of a breast

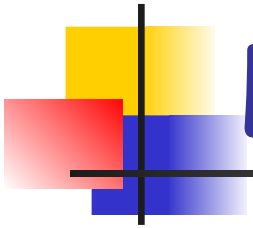
Negative Image : gives a better vision to analyze the image

Log Transformations



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

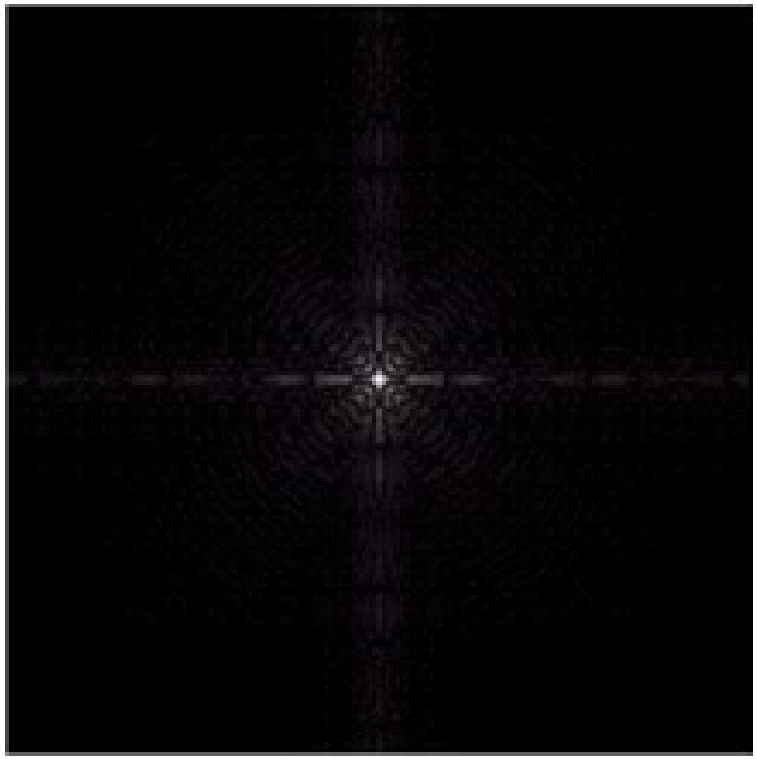
- c is a constant and $r \geq 0$
- Log curve maps a narrow range of low gray-level values in the input image into a wider range of output levels.
- Used to expand the values of dark pixels in an image while compressing the higher-level values.



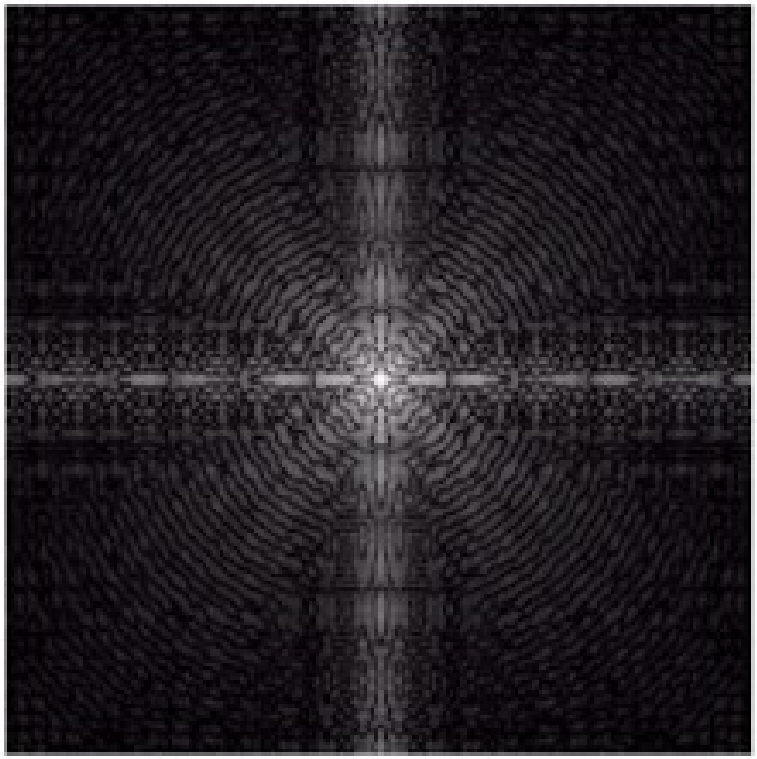
Log Transformations

- It compresses the dynamic range of images with large variations in pixel values
- Example of image with dynamic range: Fourier spectrum image
- It can have 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.

Example of Logarithm Image

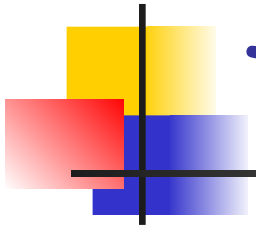


Fourier Spectrum with range = 0 to 1.5×10^6



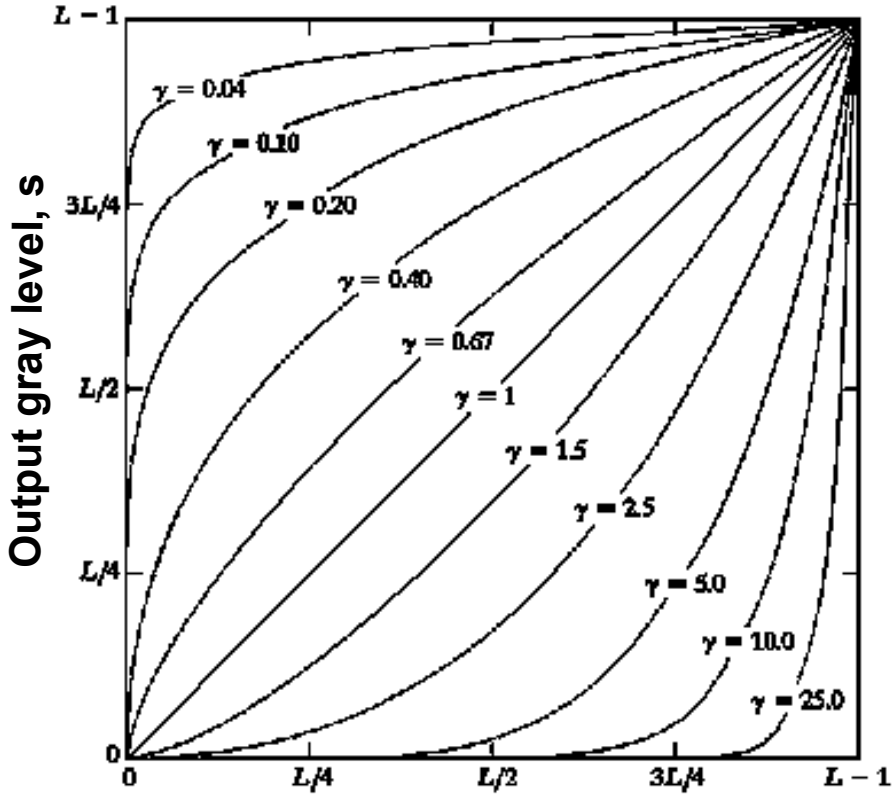
Result after apply the log transformation with c = 1, range = 0 to 6.2

Inverse Logarithm Transformations



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

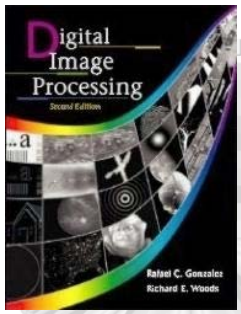
Power-Law Transformations



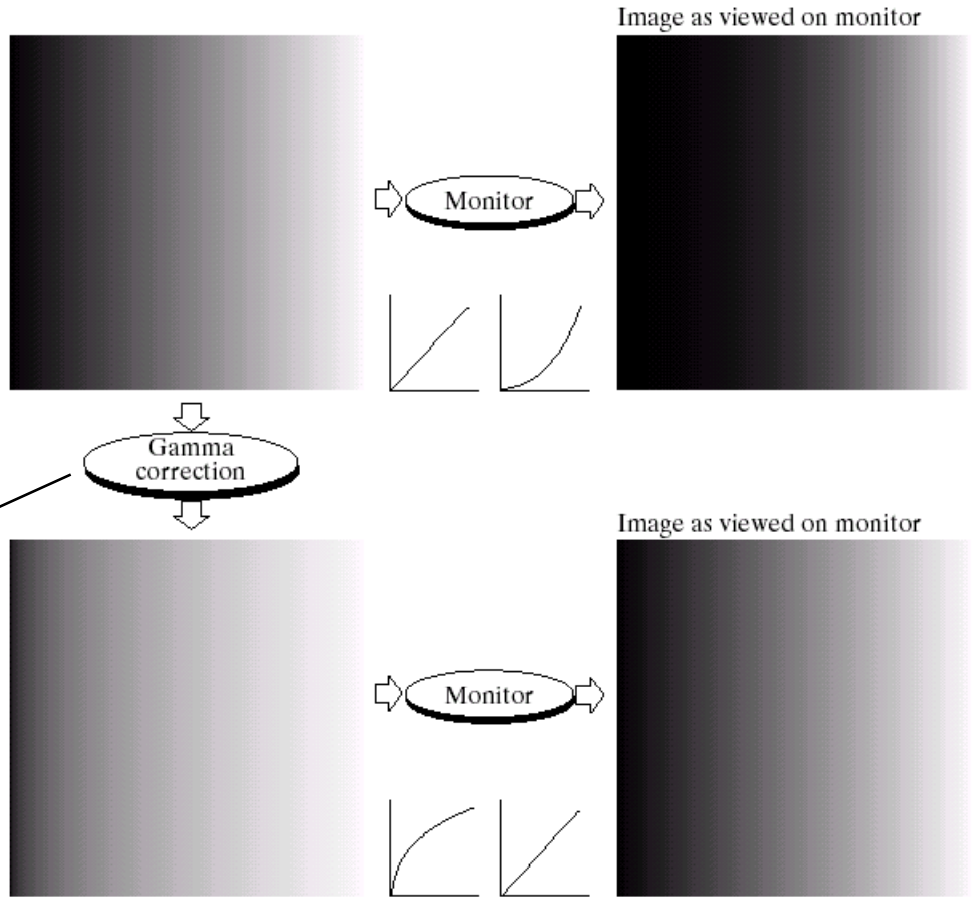
Plots of $s = cr^{\gamma}$ for various values of γ (c = 1 in all cases)

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

- 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 values, with the opposite being true for higher values of input levels.
- $c = \gamma = 1$ ⇒ Identity function



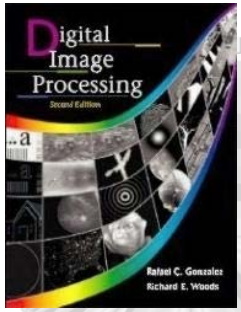
Chapter 3
Gamma Correction



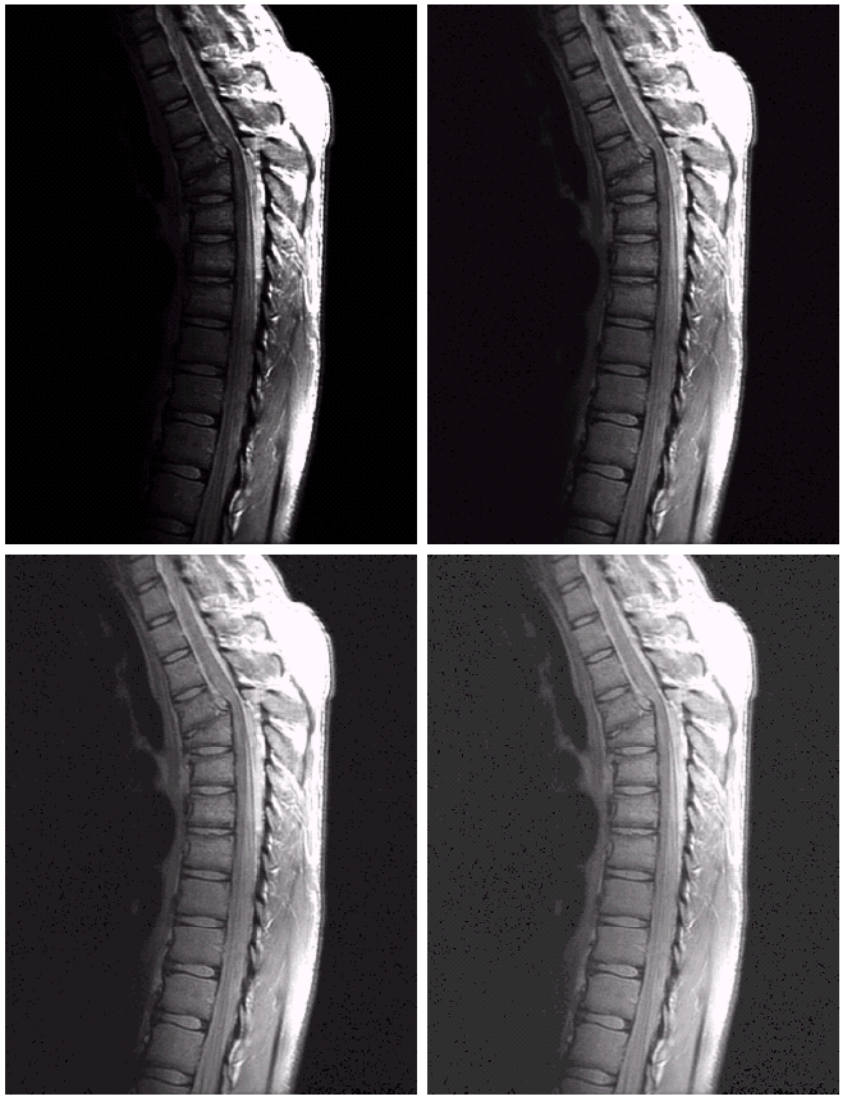
a b
c d

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

$$s = r^{1/2.5} = r^{0.4}$$

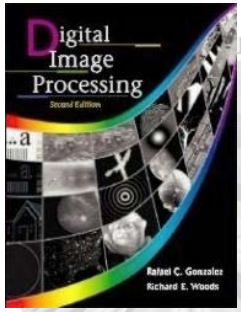


Chapter 3
Power-Law Transformations



a b
c d

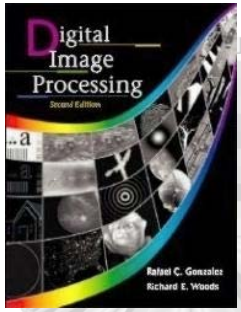
FIGURE 3.8
(a) Magnetic resonance (MR) image of a fractured human spine.
(b)–(d) Results of applying the transformation in Eq. (3.2-3) with $c = 1$ and $\gamma = 0.6, 0.4$, and 0.3, respectively. (Original image for this example courtesy of Dr. David R. Pickens, Department of Radiology and Radiological Sciences, Vanderbilt University Medical Center.)



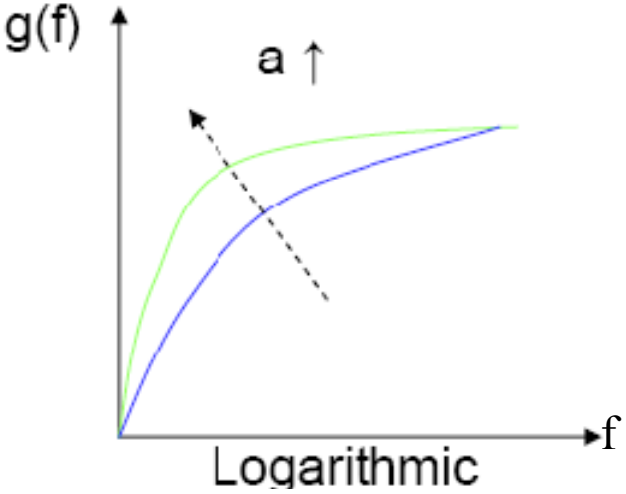
Chapter 3
Power-Law Transformations

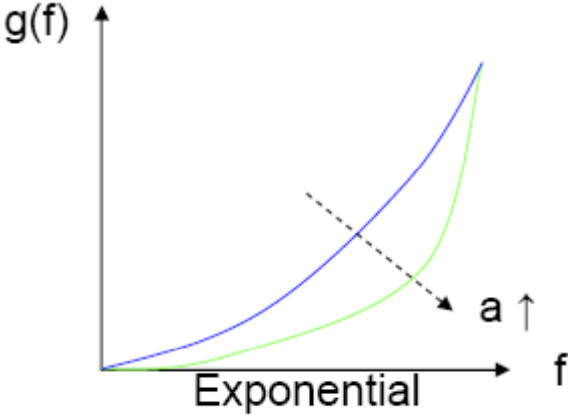
a b
c d

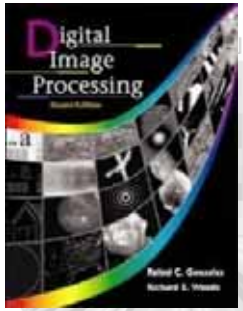
FIGURE 3.9
(a) Aerial image. (b)–(d) Results of applying the transformation in Eq. (3.2-3) with $c = 1$ and $\gamma = 3.0, 4.0$, and 5.0, respectively. (Original image for this example courtesy of NASA.)



Chapter 3
Log /Power-Law Transformations

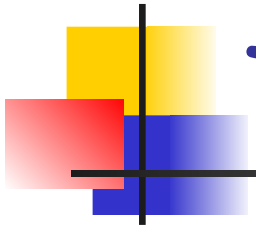






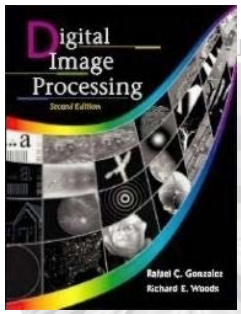
3.2 Basic gray level transformations

- Piecewise-Linear Transformation Functions
  - Contrast stretching
  - Gray-level slicing
  - Bit-plane slicing



Piecewise-Linear Transformation Functions

- Advantage:
  - The form of piecewise functions can be arbitrarily complex
- Disadvantage:
  - Their specification requires considerably more user input

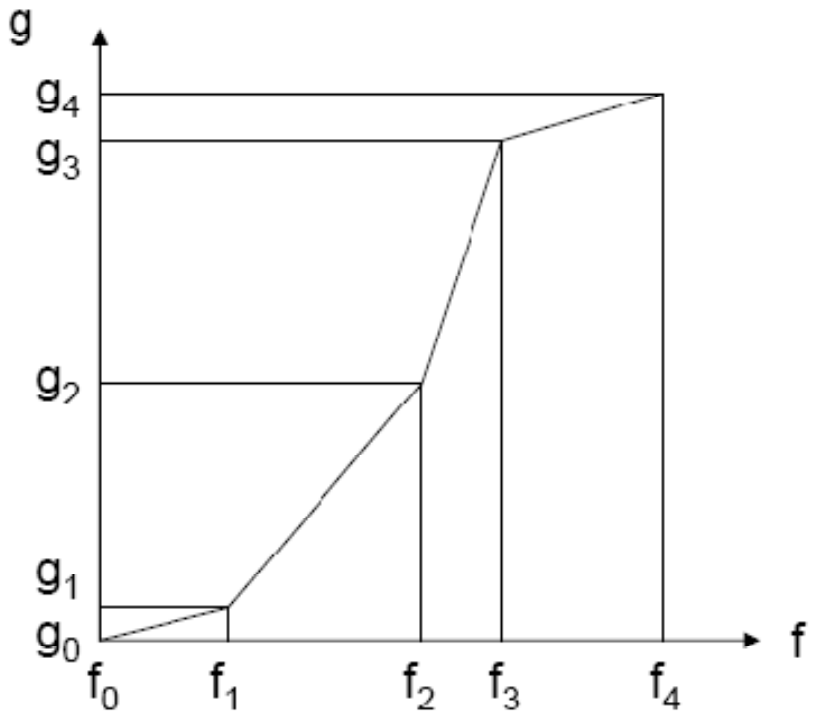


Chapter 3
Piecewise Linear Stretching

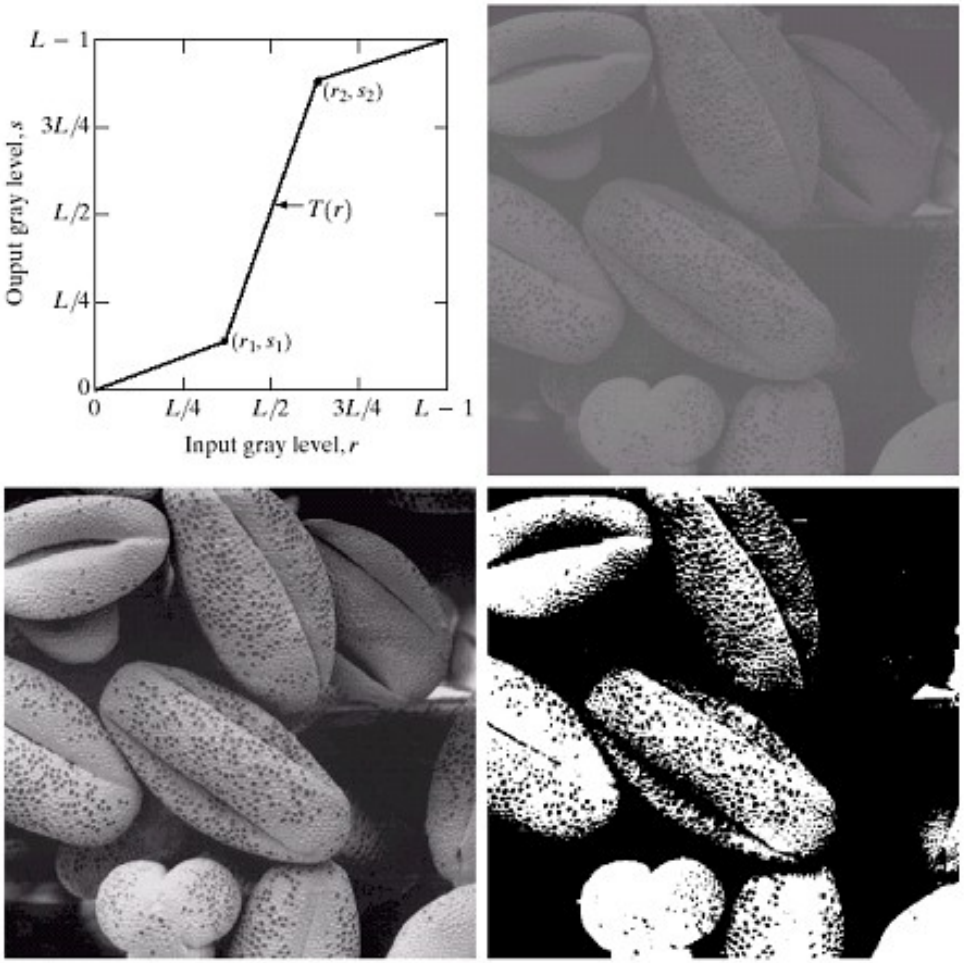
- K segments
  - Starting position of input $\{f_k, k = 0, \ldots K-1\}$
  - Starting position of output $\{g_k, k = 0, \ldots, K-1\}$
  - Transform function

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

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



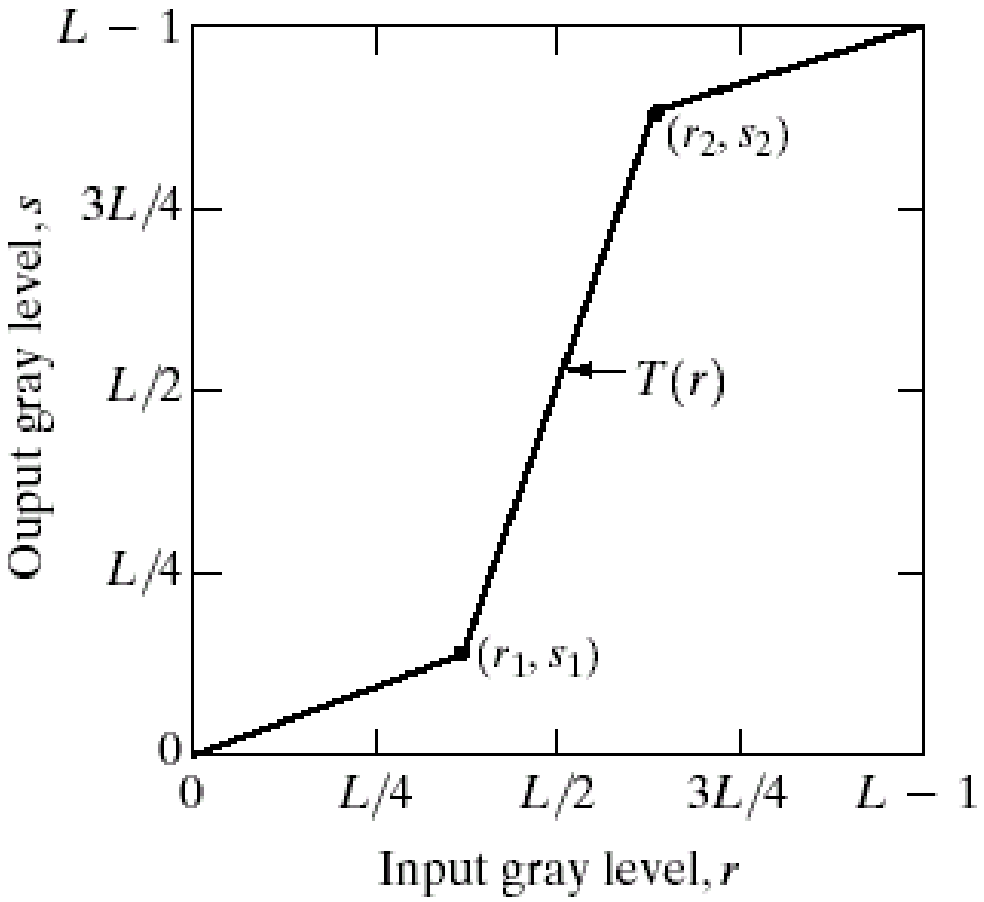
Contrast Stretching



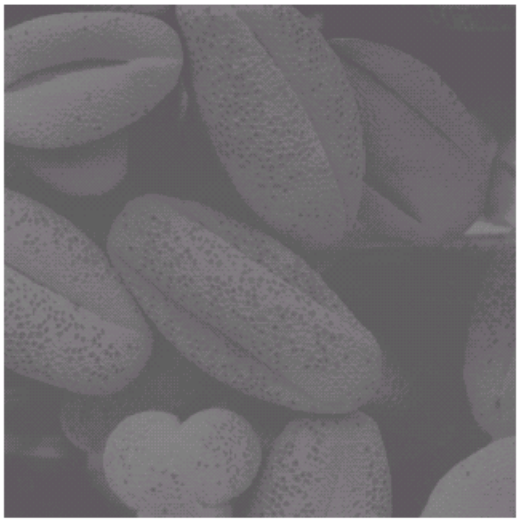
- increase the dynamic range of the gray levels in the image
- (b) a low-contrast image : result from poor illumination, lack of dynamic range in the imaging sensor, or even wrong setting of a lens aperture of image acquisition
- (c) result of contrast stretching: $(r_1,s_1) = (r_{min},0)$ and $(r_2,s_2) = (r_{max},L-1)$
- (d) result of thresholding

Contrast Stretching

Contrast means the difference between the brightest and darkest intensities



Before contrast enhancement



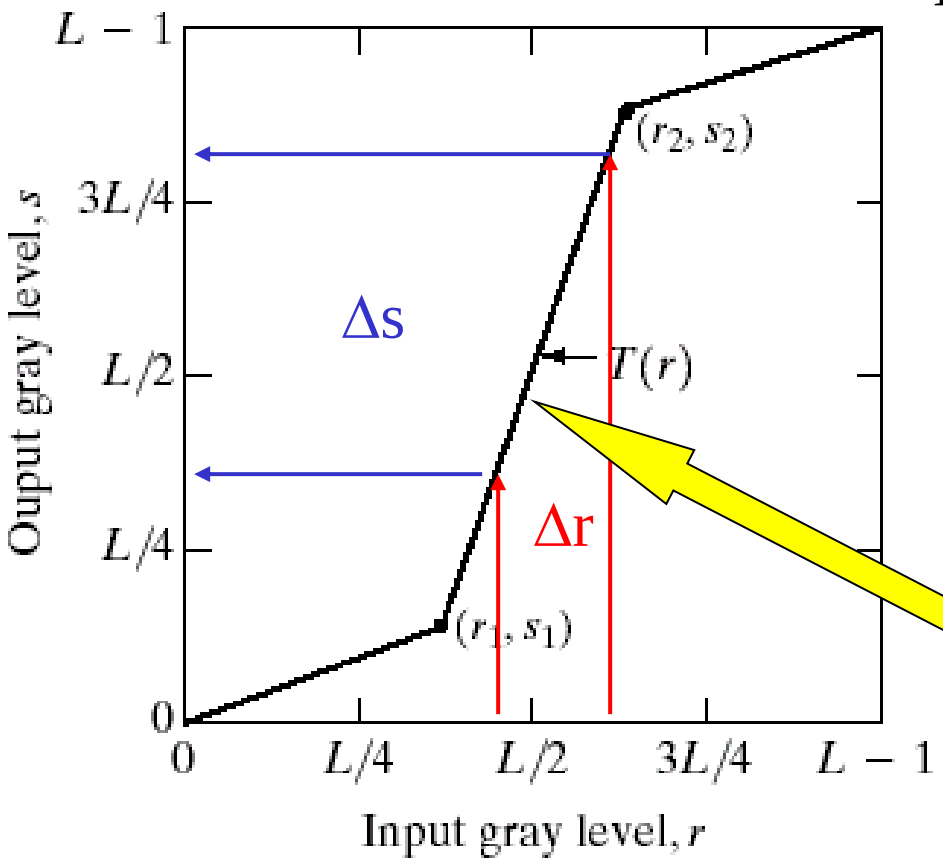
After

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

How to know where the contrast is enhanced ?

Notice the slope of T(r)

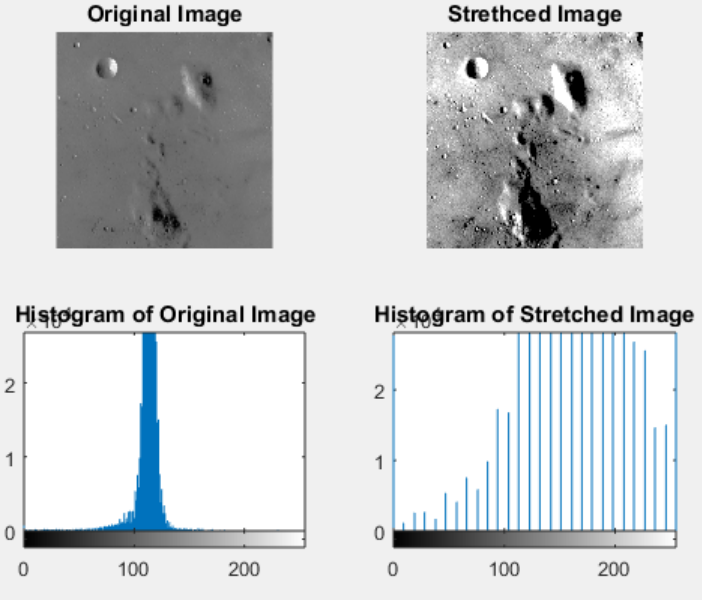
- if Slope > 1 → Contrast increases
- if Slope < 1 → Contrast decrease
- if Slope = 1 → no change



Smaller Δr yields wider Δs = increasing Contrast

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

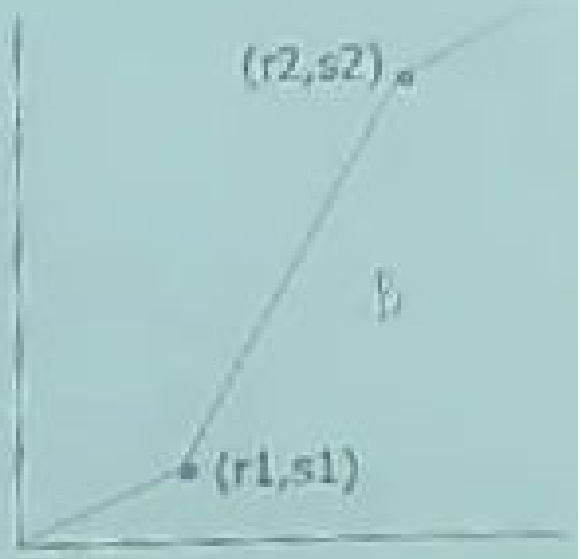
$$s = (r - r_{min}) \frac{(I_{max} - I_{min})}{(r_{max} - r_{min})} + I_{min}$$



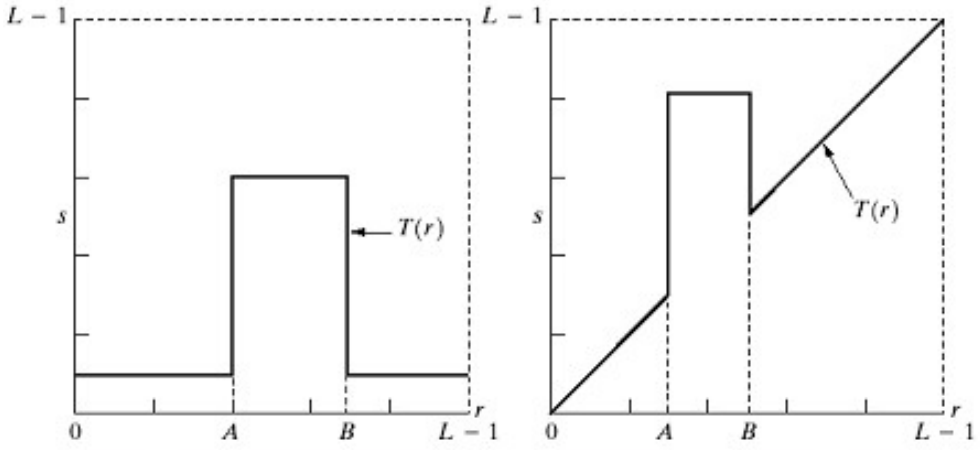
$$(10, 60) \rightarrow (120, 80)$$

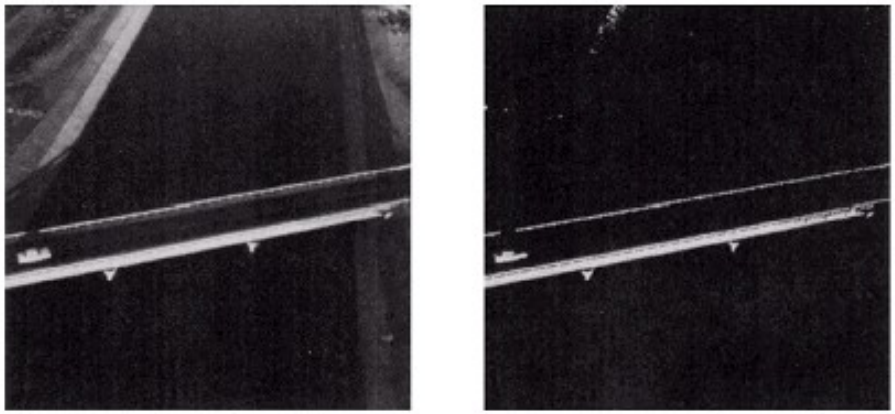
Contrast Stretching

- The location (r1,s1) and (r2,s2) control the transformation.
  - If (r1=s1) and (r2=s2), then identical function
  - If (r1=r2) and (s1=0 and s2=L-1) thresholding function



Gray-level slicing



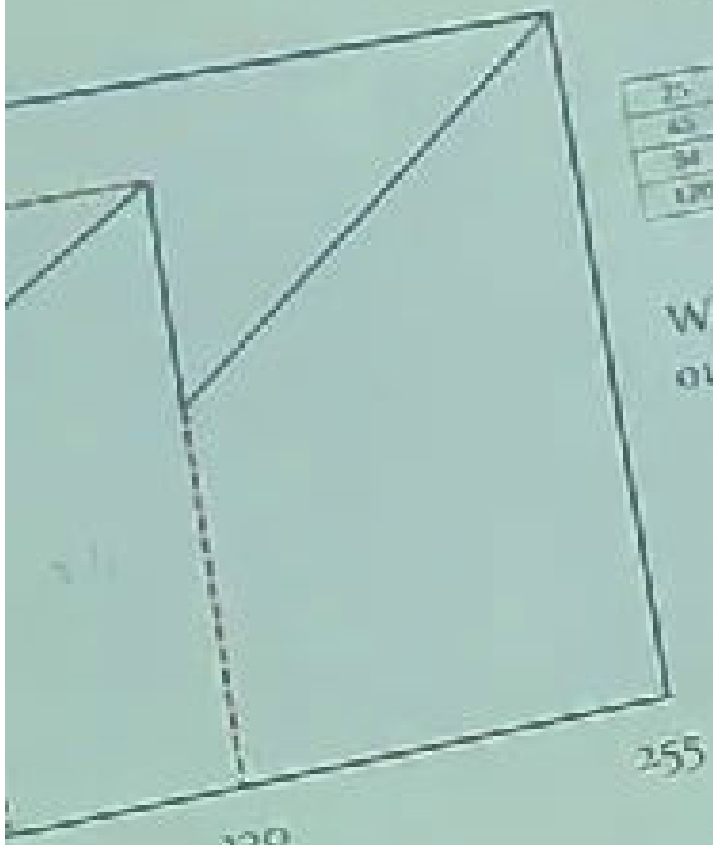


- Highlighting a specific range of gray levels in an image
  - Display a high value of all gray levels in the range of interest and a low value for all other gray levels
- (a) transformation highlights range [A,B] of gray level and reduces all others to a constant level
- (b) transformation highlights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

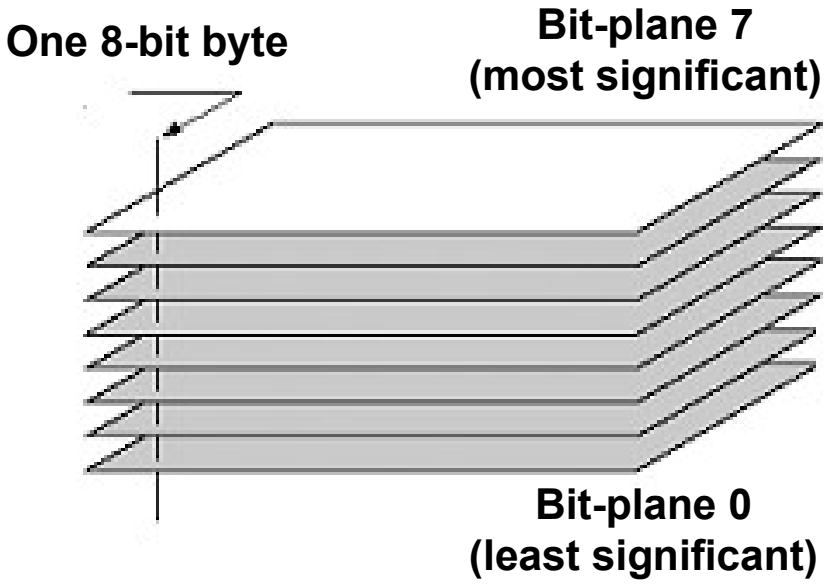
I =

25	110	180	210
45	130	205	230
80	200	240	240
120	90	250	200

What will be the output image?

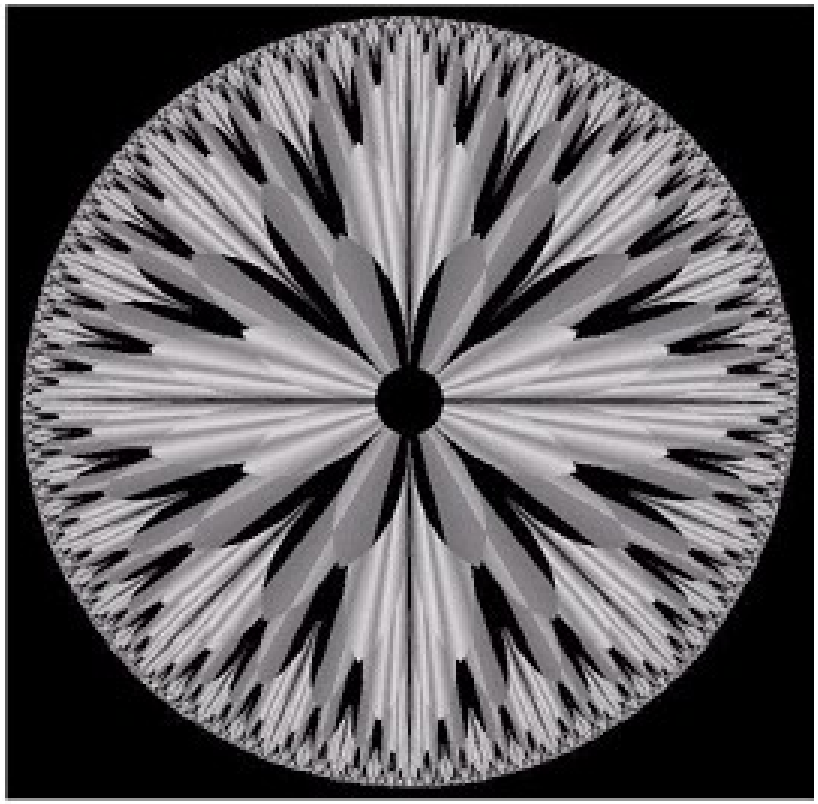


Bit-plane slicing



- Highlighting the contribution made to total image appearance by specific bits
- Suppose 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

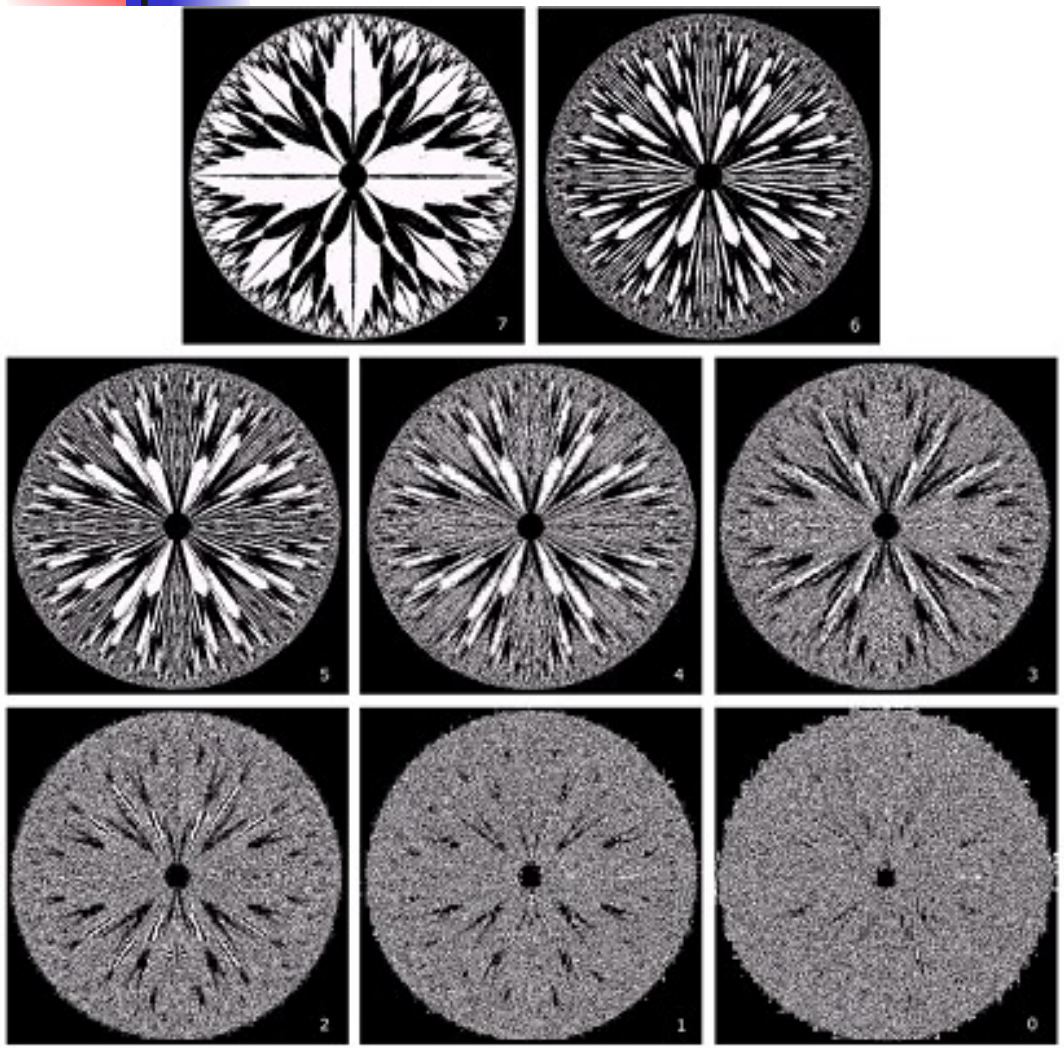
Example



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 129 and 255 to 255

8 bit planes



Bit-plane 7	Bit-plane 6	
Bit-plane 5	Bit-plane 4	Bit-plane 3
Bit-plane 2	Bit-plane 1	Bit-plane 0

[Insert Homework Question]

Simple Image Statistics- Histogram

- Let S be a set and define #S as the cardinality of this set
  - The histogram $h_A(l)$ $(l=0,\ldots,255)$ of the image A is defined as:

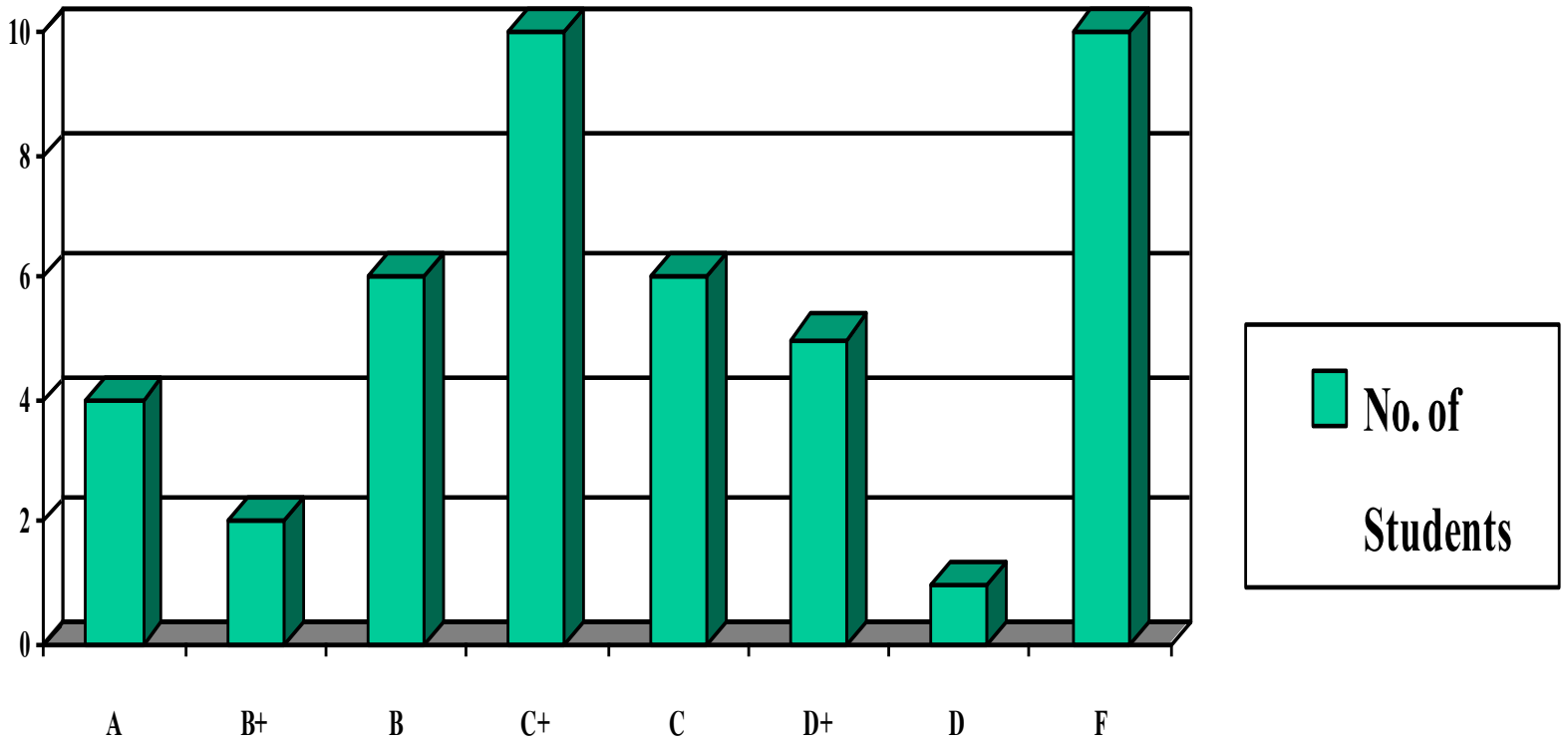
$h_A(l) = \#[(i,j) \mid A(i,j)=l,\ i=0,\ldots,N-1;\ j=0,\ldots,M-1]$

$$\sum_{l=0}^{255} h_A(l) = \text{Number of pixels in } A$$

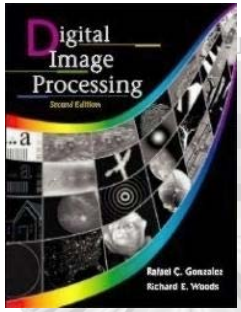
```
For(i=0;i<255;i++)
   H[i]=0;
for(i=0;i<N;i++)
   for(j=0;j<M;j++)
         H[A(i,j)]++;
```

Histogram

Histogram = Graph of population frequencies



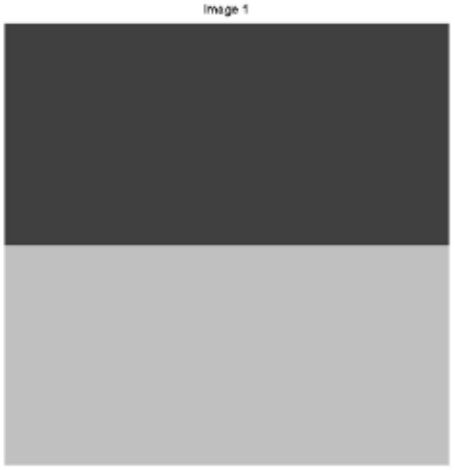
Grades of the course 178 xxx

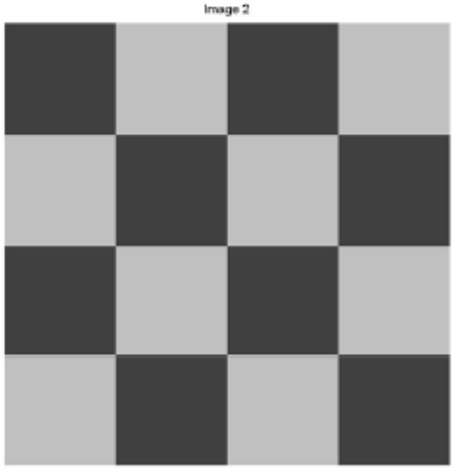


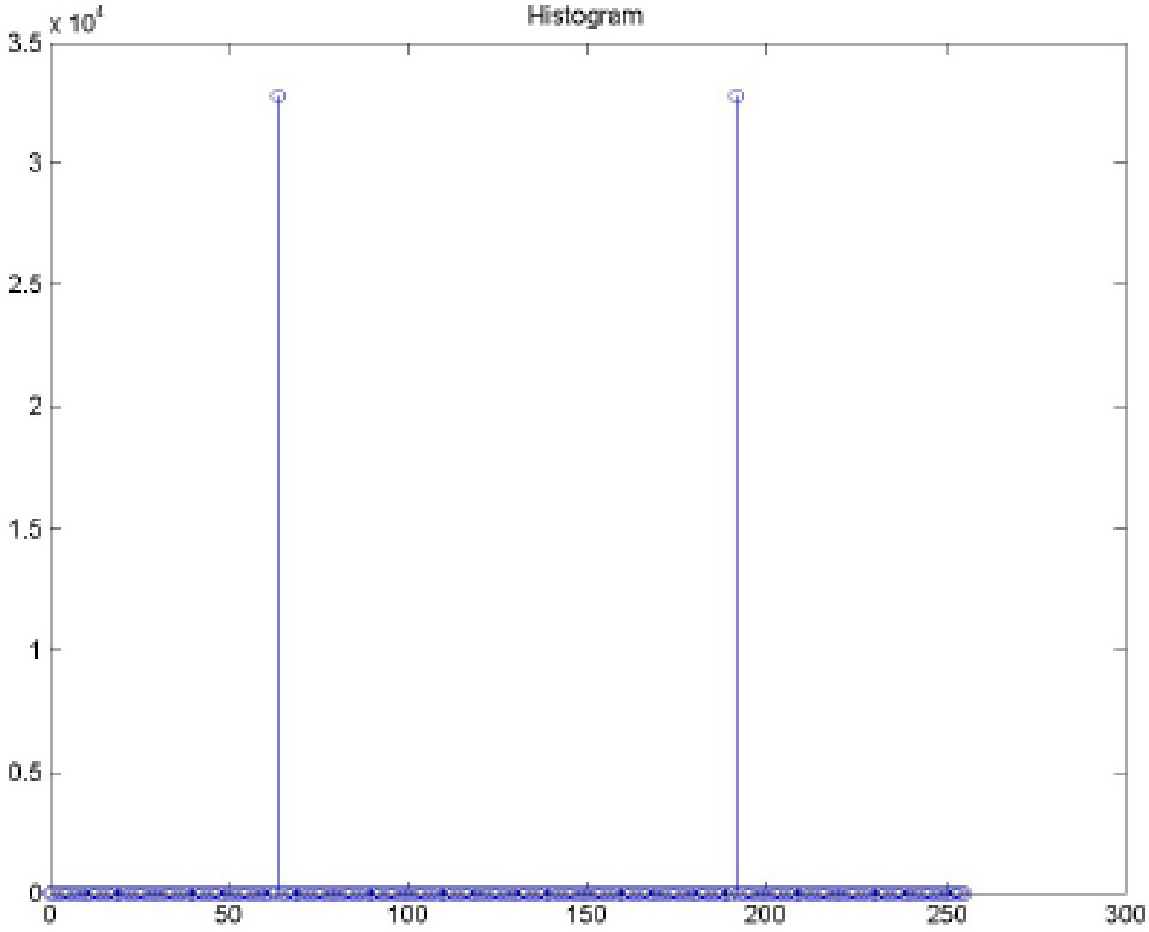
Chapter 3
Histogram Processing

Different images

But have the same histogram

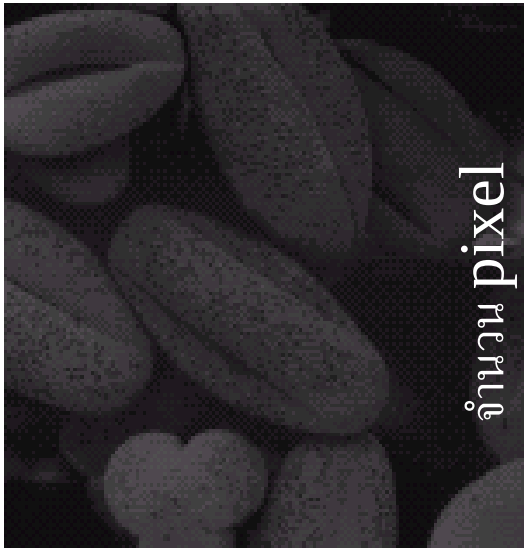


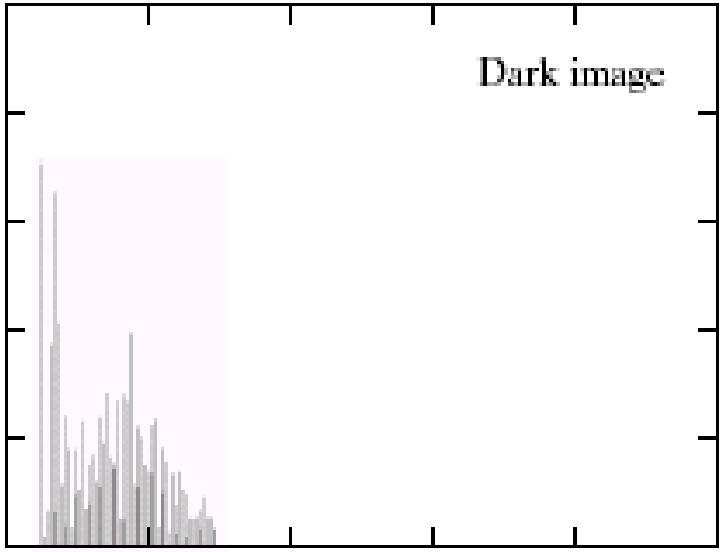




Histogram of an Image

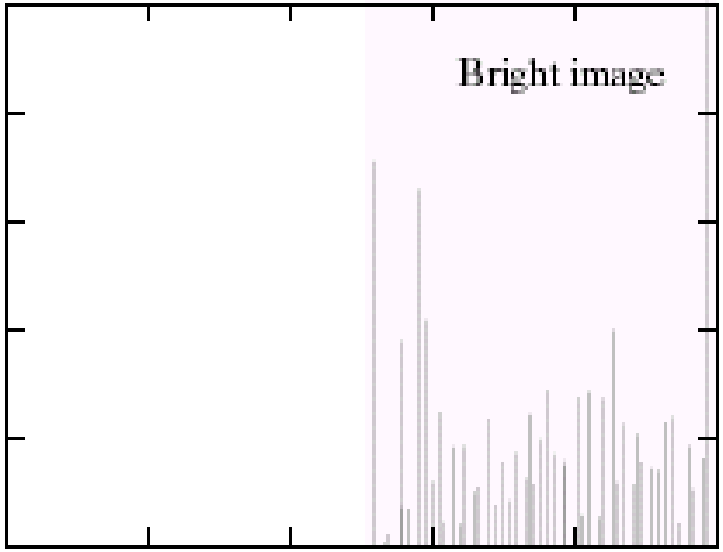
= graph of no. of pixels vs intensities





$$h(r_k) = n_k$$

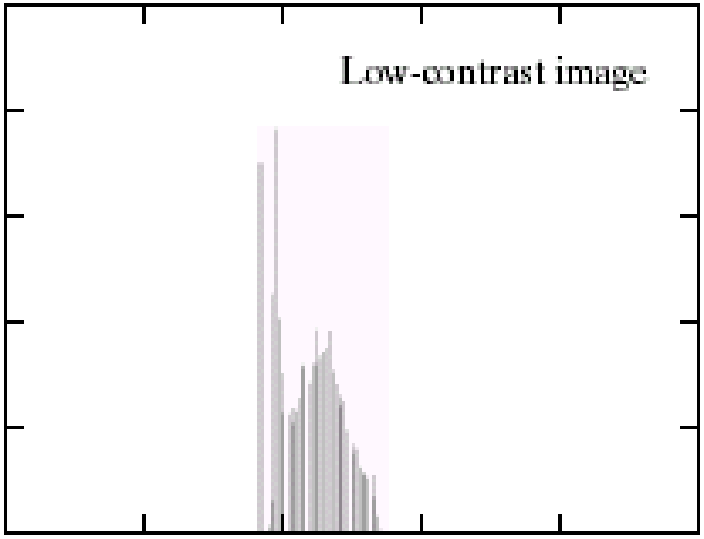
Dark image has histogram on the left



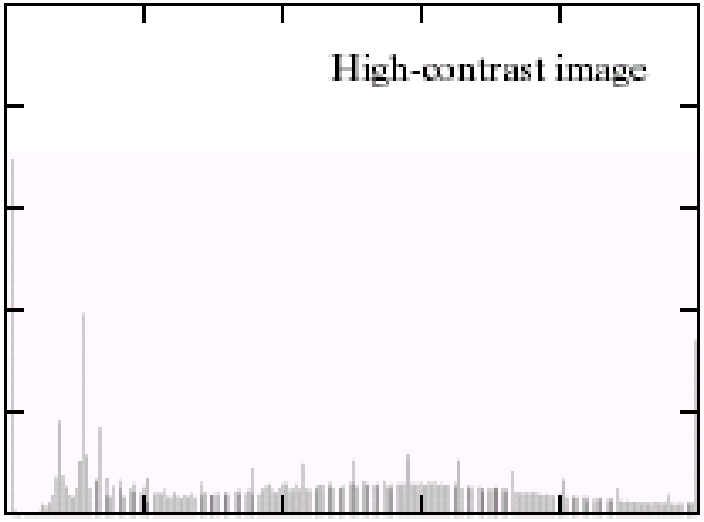
Bright image has histogram on the right

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Histogram of an Image (cont.)

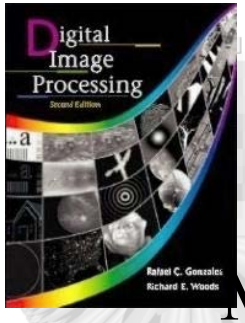


low contrast image
has narrow histogram



high contrast image
has wide histogram

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

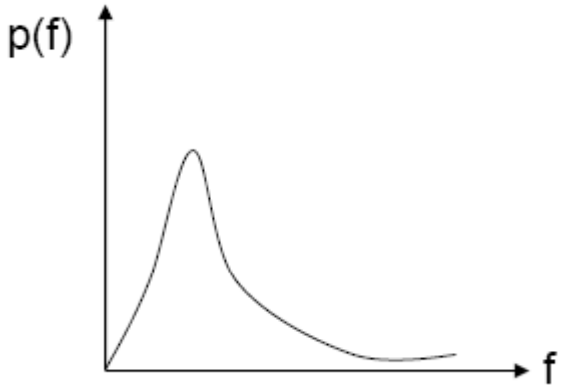


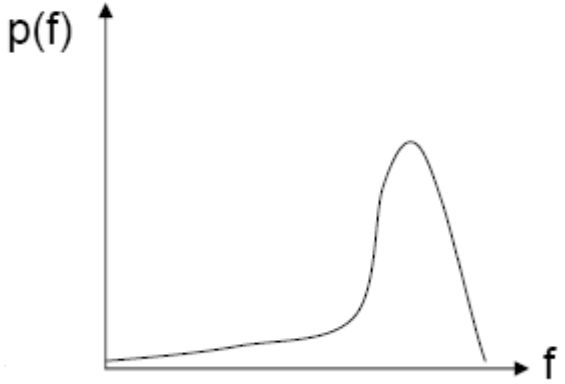
Chapter 3
Histogram Processing

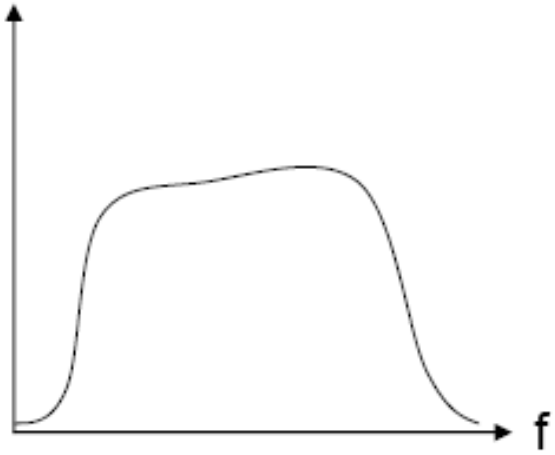
More examples of histograms

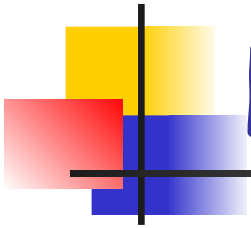
(a) Too dark

(b) Too bright









Histogram Processing

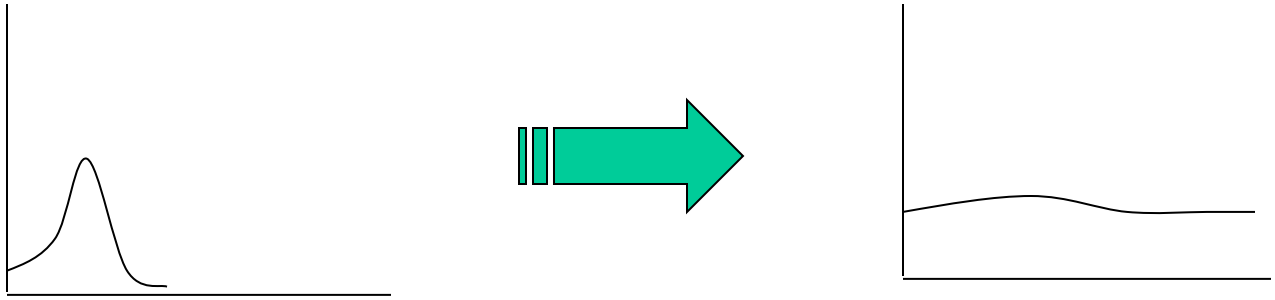
- Basic for numerous spatial domain processing techniques
- Used effectively for image enhancement
- Information inherent in histograms also is useful in image compression and segmentation

Histogram Processing

= intensity transformation based on histogram information to yield desired histogram

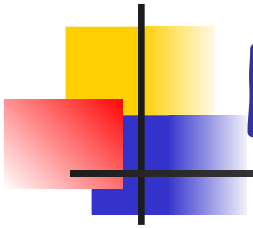
- Histogram equalization

To make histogram distributed uniformly



- Histogram matching

To make histogram as the desire



Histogram Processing

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

Make (min, max) to (0,255)

Histogram Stretching

- Aim is to spread the histogram to cover the entire dynamic range.
- increase contrast.

The mapping function is $\frac{s_{max} - s_{min}}{r_{max} - r_{min}}$

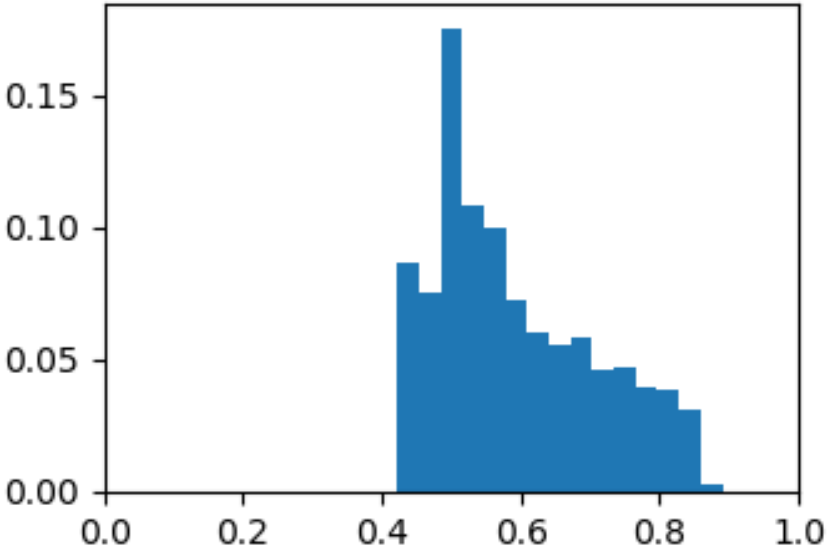
(s_{max} and s_{min} are maximum and minimum of pixel values in the stretched histogram and r_{max} and r_{min} are that of original image)

Transformation is $\frac{s_{max} - s_{min}}{r_{max} - r_{min}}(r - r_{min}) + s_{min}$

$$\frac{(r - r_{min})}{} \times 255$$

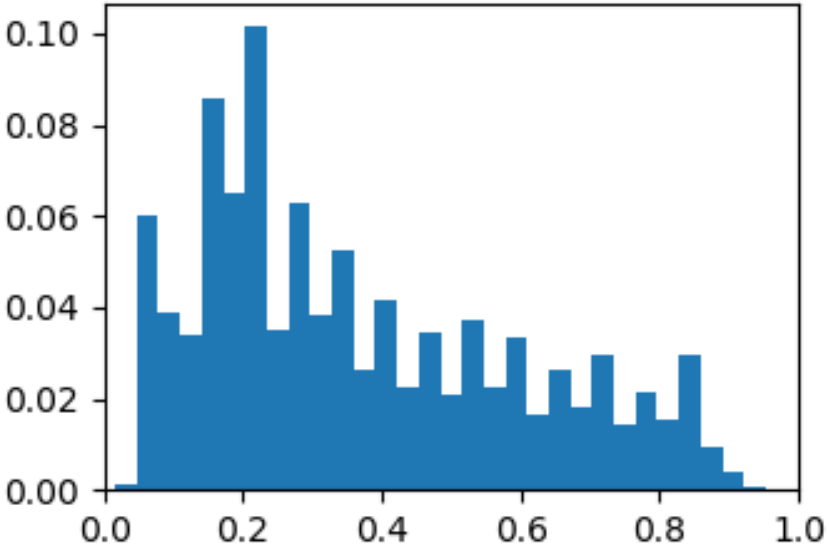
Low contrast orginal

Histogram of low contrast image

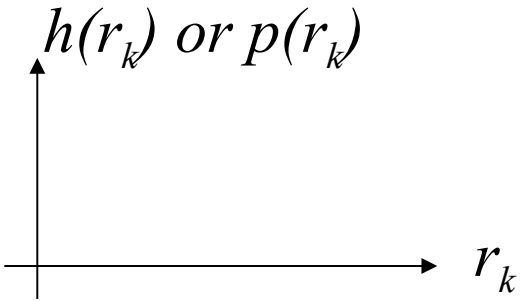


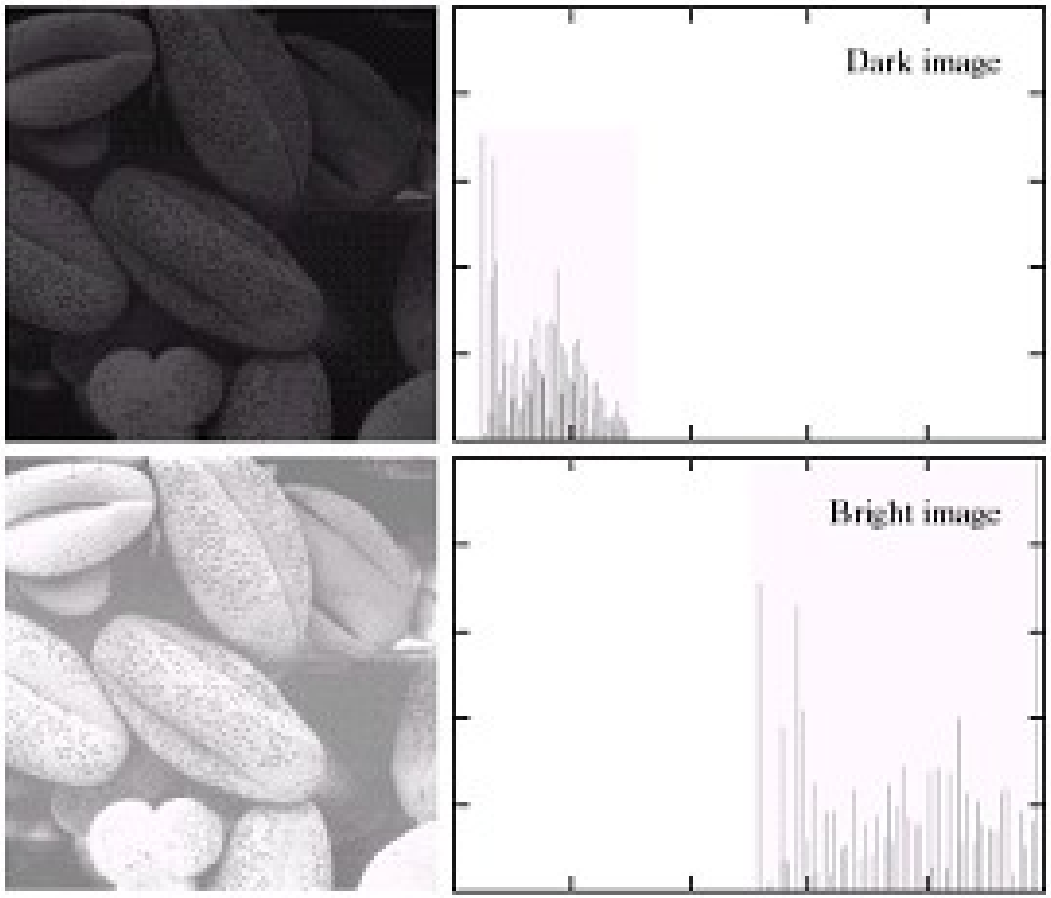
Contrast Stretched

Histogram of contrast stretched image



Example





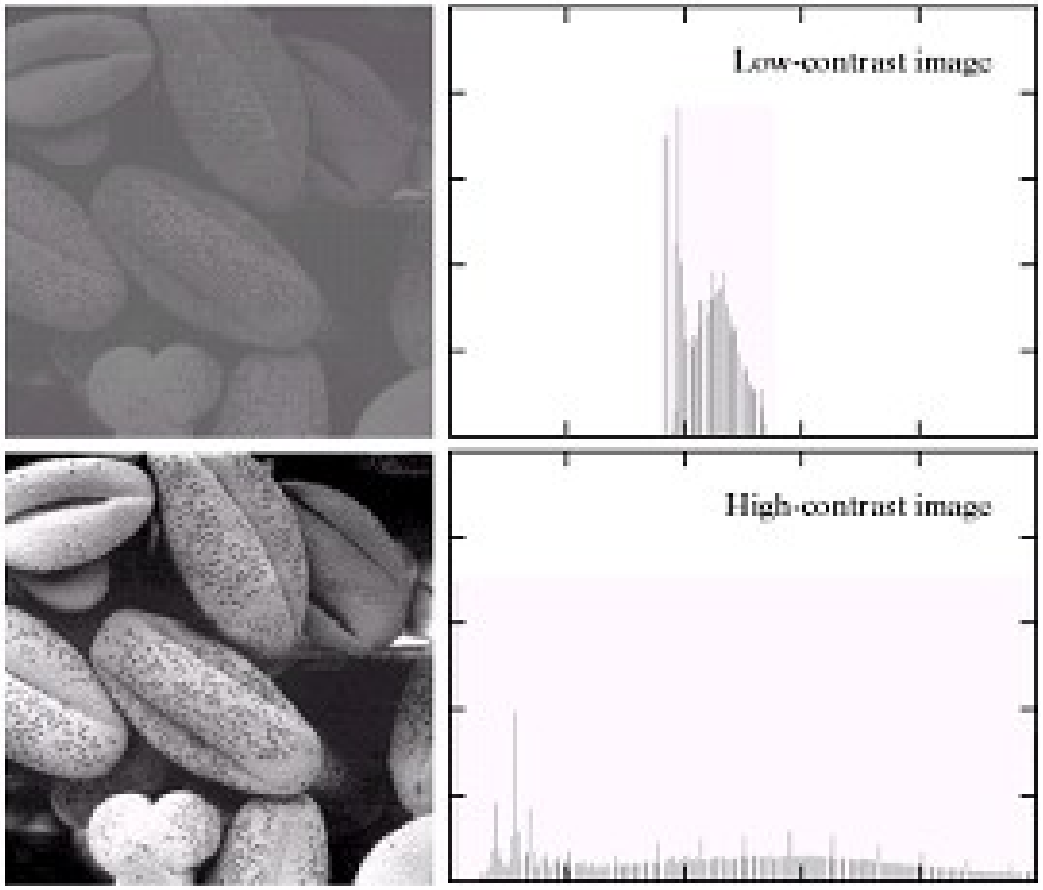
Dark image

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.

Example



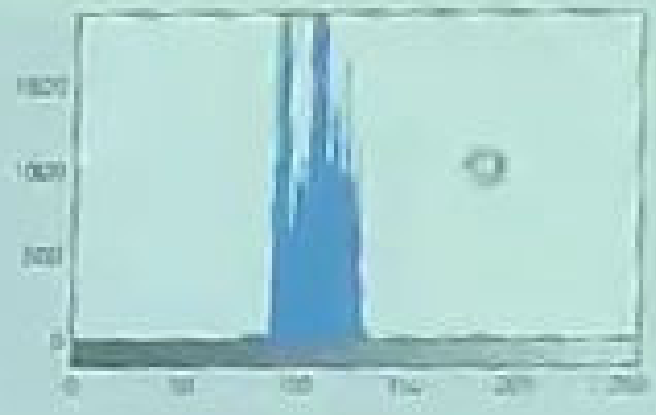
Low-contrast image

histogram is narrow and centered toward the middle of the gray scale

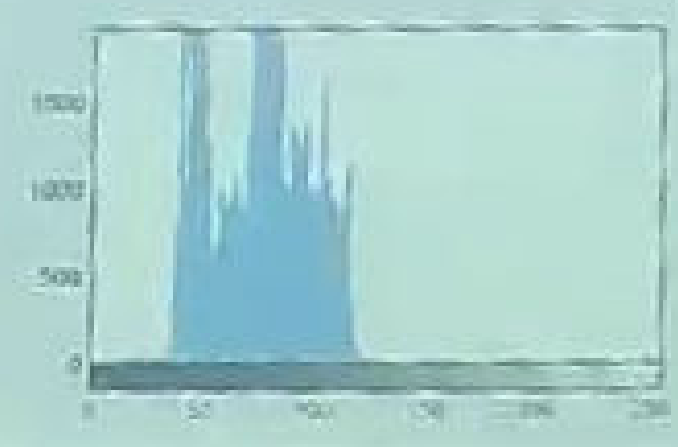
High-contrast image

histogram covers 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 the others

Max: 198
Min: 71



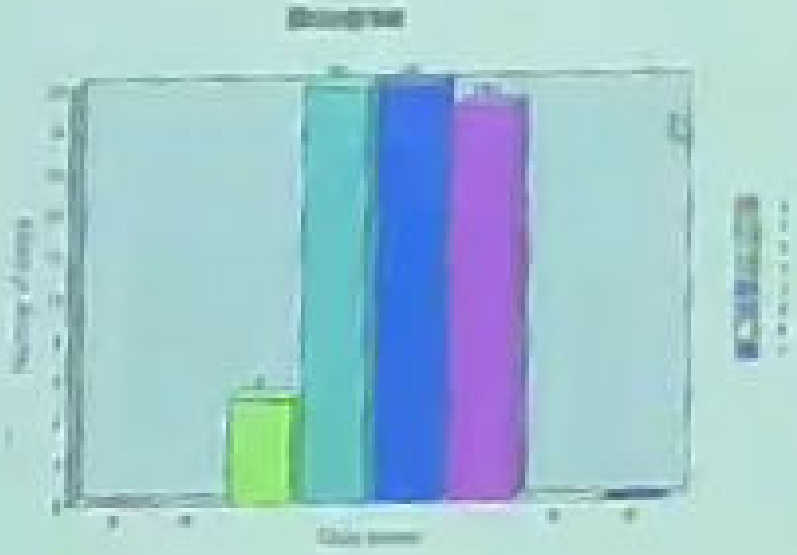
Max: 255
Min: 0



HOMEWORK

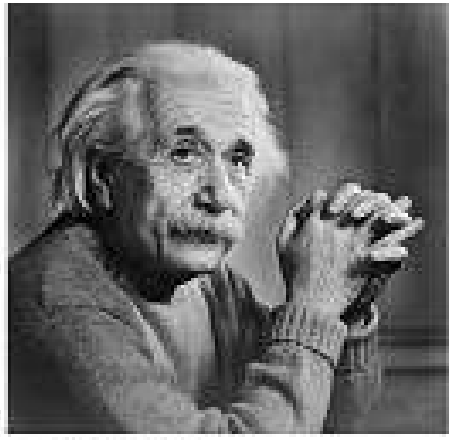
Perform histogram stretching of the image

Gray level	No of pixels
0	0
1	0
2	6
3	20
4	20
5	10
6	0
7	0



When there will be no effect of histogram stretching?

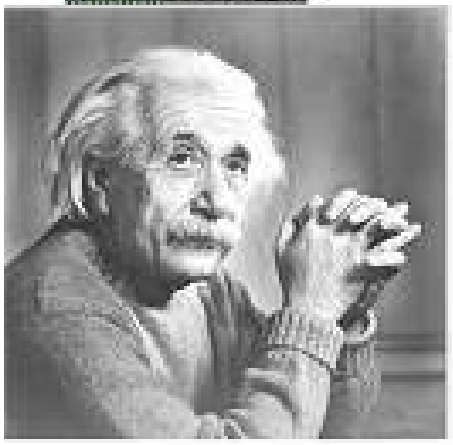
Old image



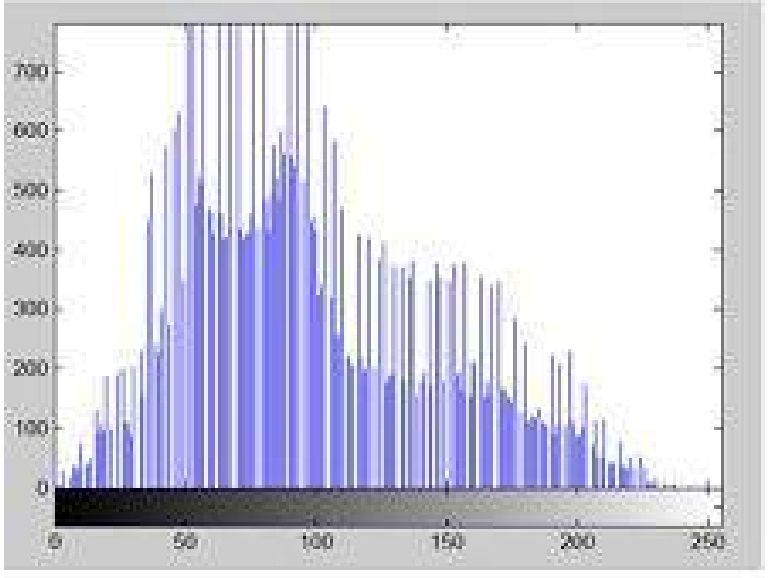
Offset = 50

S = r + offset

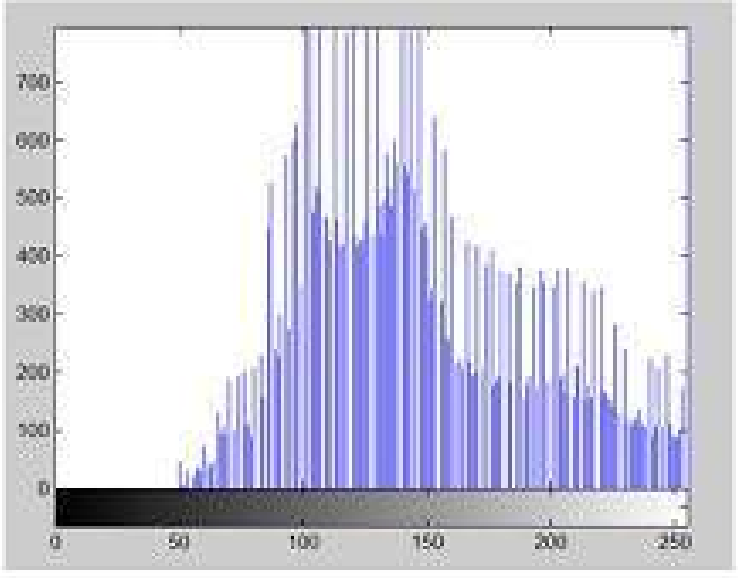
New image



Old histogram



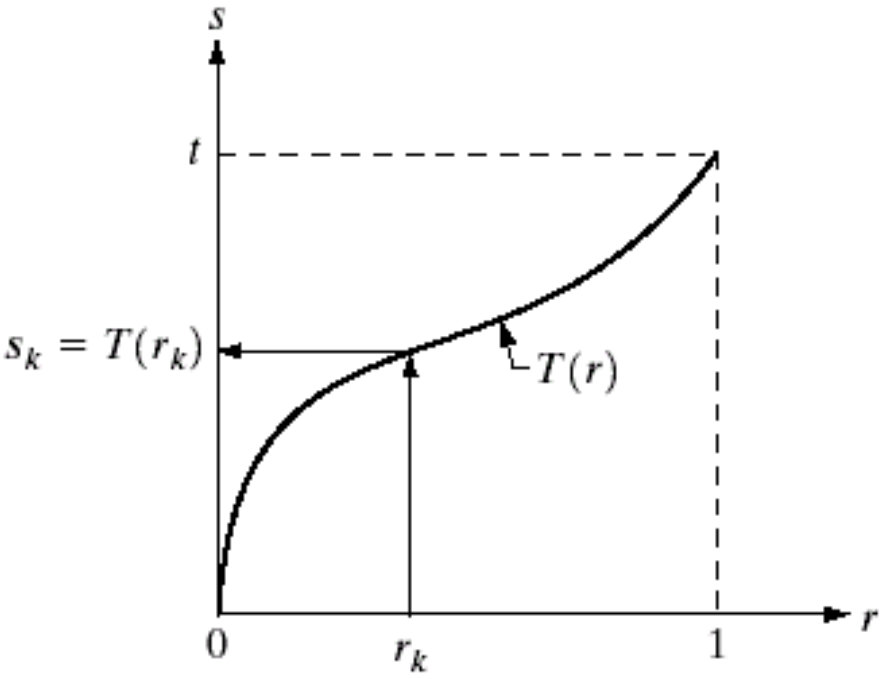
New Histogram



Monotonically Increasing Function

= Function that is only increasing or constant

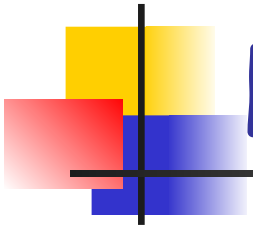
$$s = T(r)$$



Properties of Histogram processing function

1. Monotonically increasing function

2. $0 \le T(r) \le 1$ for $0 \le r \le 1$

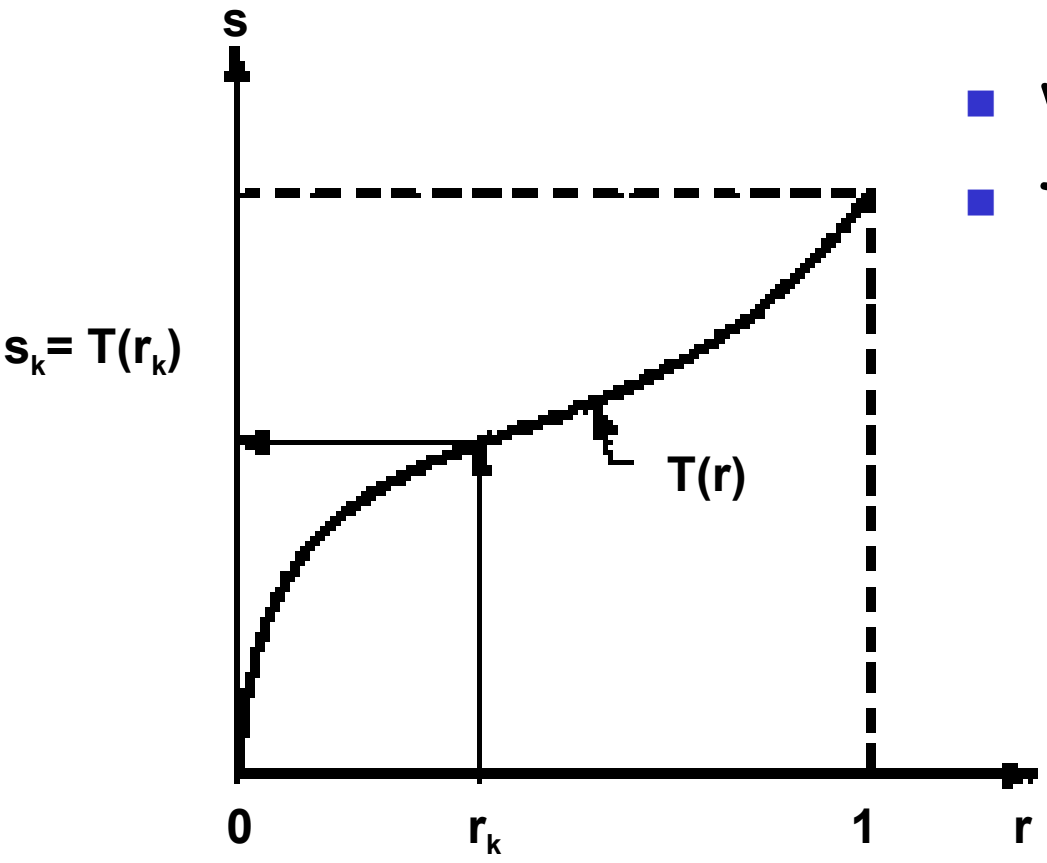


Histogram Equalization

- As the low-contrast image's histogram is narrow and centered toward the middle of the gray scale, if we distribute the histogram to a wider range the quality of the image will be improved.
- We can do it by adjusting the probability density function of the original histogram of the image so that the probability spread equally

Contrast Stretching means histogram stretching

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 $0 \leq r \leq 1$

Probability Density Function

Histogram is analogous to Probability Density Function (PDF) which represent density of population

Let s and r be Random variables with PDF $p_s(s)$ and $p_r(r)$ respectively and relation between s and r is

r ε [0,1]

$$s = T(r)$$

s is the transformation of r and r is normalised

r in the image can be viewed as random variable in [0,1]

We get

T(r) is continuous and differentiable over the range

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

Histogram Equalization

Let

$$s = T(r) = \int_0^r p_r(w)dw$$

We get

$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right| = p_r(r)\cdot\left|\frac{1}{\frac{ds}{dr}}\right|$$

$$= p_r(r)\cdot\left|\frac{1}{\frac{d\left(\int_0^r p_r(w)dw\right)}{dr}}\right| = p_r(r)\cdot\left|\frac{1}{p_r(r)}\right| = 1 \;!$$

Histogram Equalization

Formula in the previous slide is for a continuous PDF

For Histogram of Digital Image, we use

$$s_k = T(r_k) = \sum_{j=0}^{k} p_r(r_j)$$

$$= \sum_{j=0}^{k} \frac{n_j}{N}$$

n_j = the number of pixels with intensity = j
N = the number of total pixels

Discrete transformation function

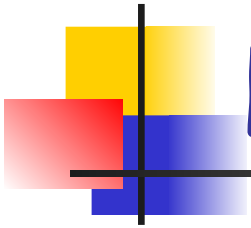
- The probability of occurrence of gray level in an image is approximated by

$$p_r(r_k) = \frac{n_k}{n} \qquad \text{where } k = 0, 1, ..., L\text{-}1$$

- The discrete version of transformation

$$s_k = T(r_k) = \sum_{j=0}^{k} p_r(r_j)$$

$$= \sum_{j=0}^{k} \frac{n_j}{n} \qquad \text{where } k = 0, 1, ..., L\text{-}1$$



Histogram Equalization

- Thus, an output image is obtained by mapping each pixel with level r_k in the input image into a corresponding pixel with level s_k in the output image
- In discrete space, it cannot be proved in general that this discrete transformation will produce the discrete equivalent of a uniform probability density function, which would be a uniform histogram

Histogram Equalization Example

Intensity	# pixels
0	20
1	5
2	25
3	10
4	15
5	5
6	10
7	10
Total	100

Accumulative Sum of P_r
20/100 = 0.2
(20+5)/100 = 0.25
(20+5+25)/100 = 0.5
(20+5+25+10)/100 = 0.6
(20+5+25+10+15)/100 = 0.75
(20+5+25+10+15+5)/100 = 0.8
(20+5+25+10+15+5+10)/100 = 0.9
(20+5+25+10+15+5+10+10)/100 = 1.0
1.0

Histogram Equalization Example (cont.)

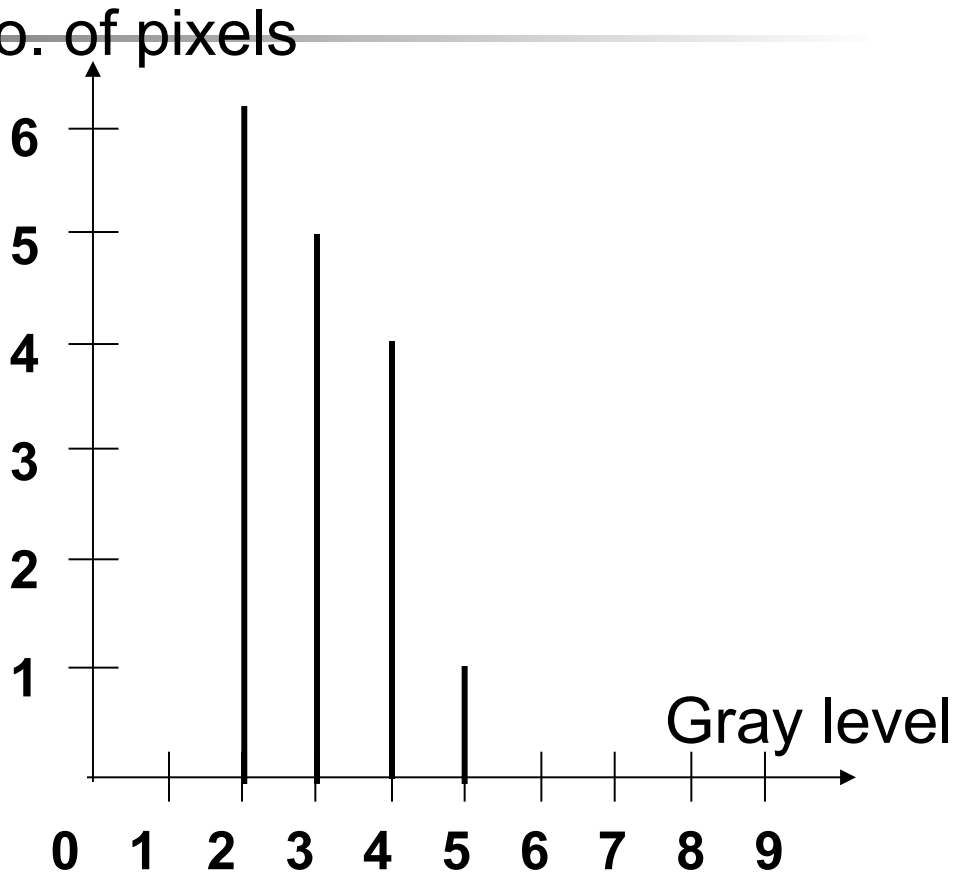
Intensity (r)	No. of Pixels (n_j)	Acc Sum of P_r	Output value	Quantized Output (s)
0	20	0.2	0.2x7 = 1.4	1
1	5	0.25	0.25*7 = 1.75	2
2	25	0.5	0.5*7 = 3.5	3
3	10	0.6	0.6*7 = 4.2	4
4	15	0.75	0.75*7 = 5.25	5
5	5	0.8	0.8*7 = 5.6	6
6	10	0.9	0.9*7 = 6.3	6
7	10	1.0	1.0x7 = 7	7
Total	100			

Example

2	3	3	2
4	2	4	3
3	2	3	5
2	4	2	4

4x4 image

Gray scale = [0,9]



histogram

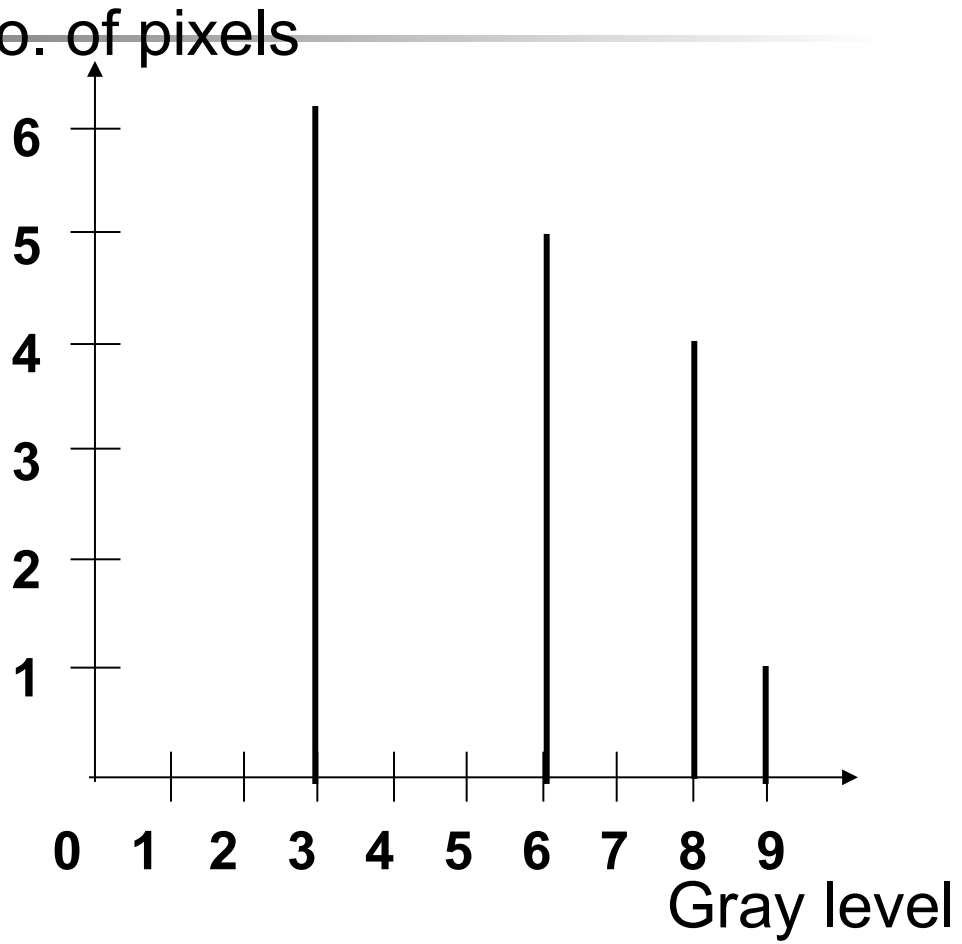
Gray Level(j)	0	1	2	3	4	5	6	7	8	9
No. of pixels	0	0	6	5	4	1	0	0	0	0
$\sum_{j=0}^{k} n_j$	0	0	6	11	15	16	16	16	16	16
$s = \sum_{j=0}^{k} \frac{n_j}{n}$	0	0	6 / 16	11 / 16	15 / 16	16 / 16	16 / 16	16 / 16	16 / 16	16 / 16
s x 9	0	0	3.3 $\approx$3	6.1 $\approx$6	8.4 $\approx$8	9	9	9	9	9

Example

3	6	6	3
8	3	8	6
6	3	6	9
3	8	3	8

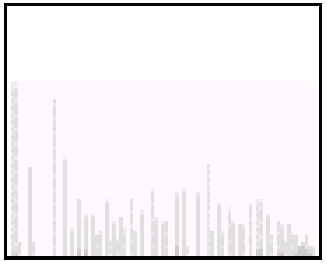
Output image

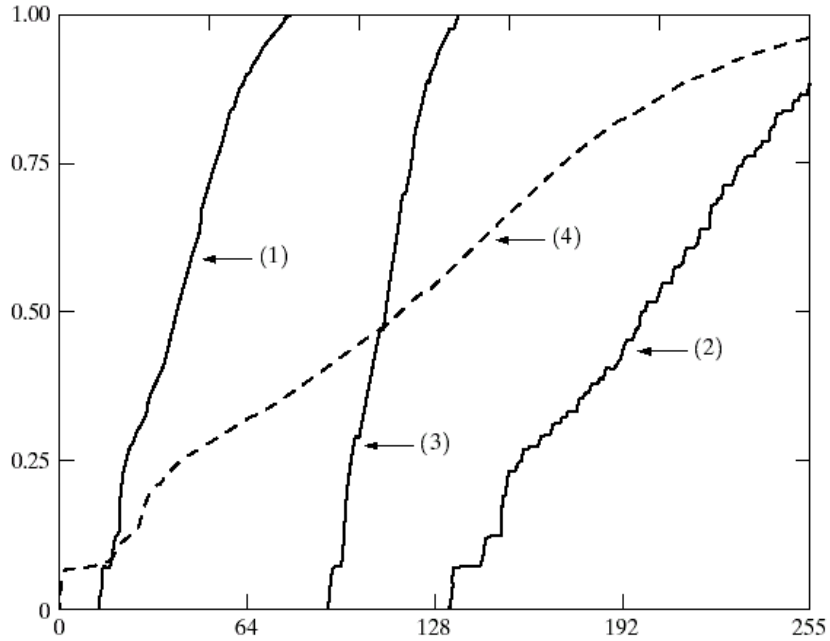
Gray scale = [0,9]

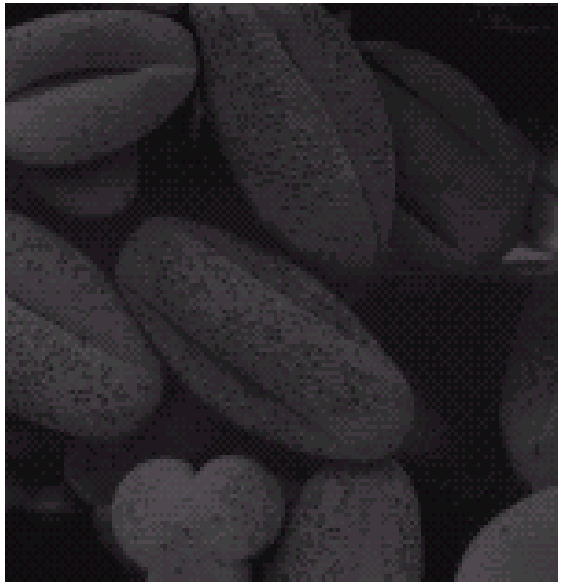


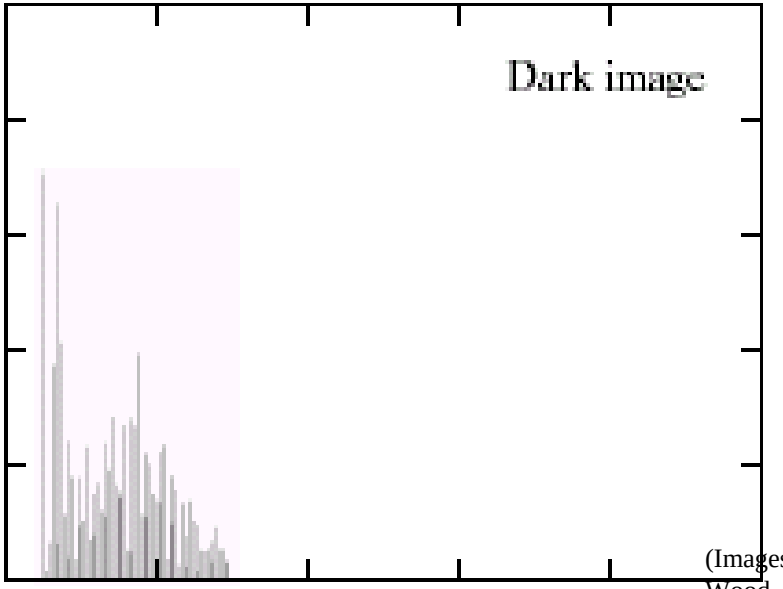
Histogram equalization

Histogram Equalization



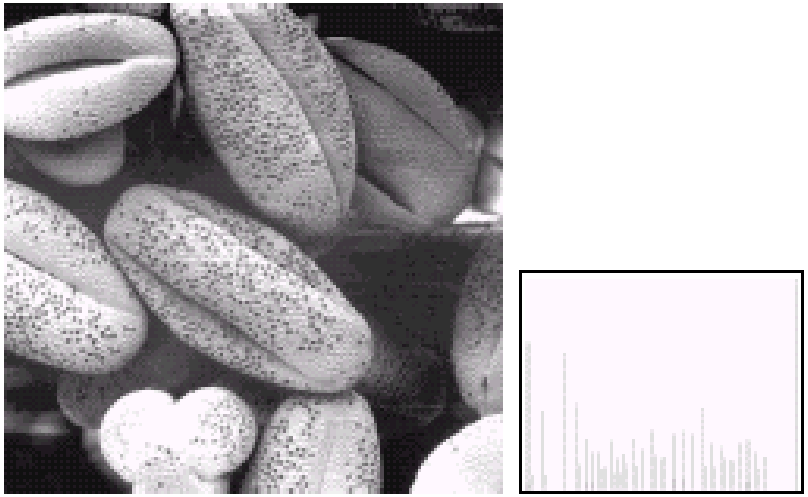


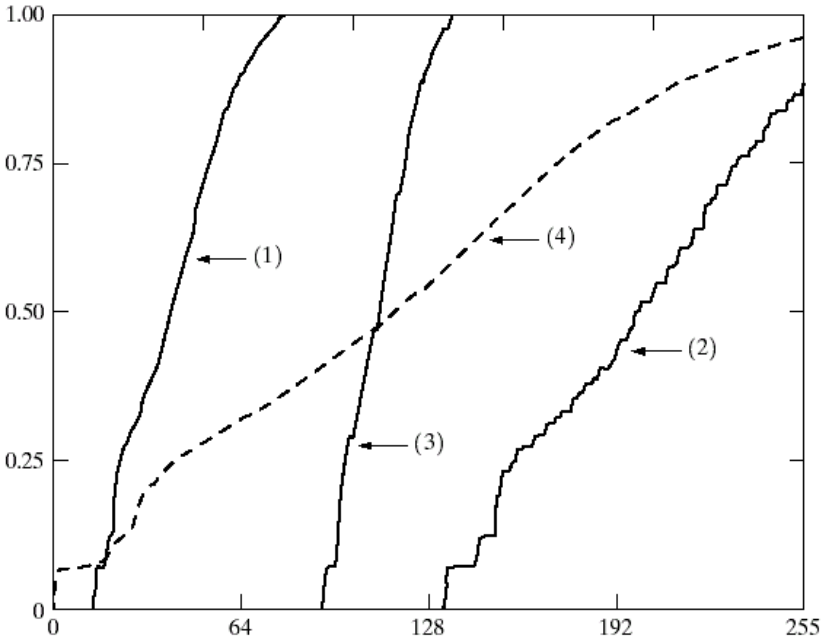


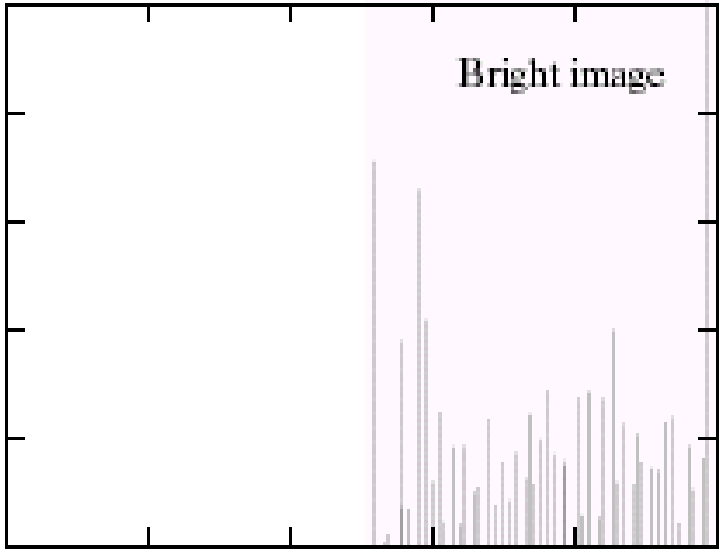


(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Histogram Equalization (cont.)

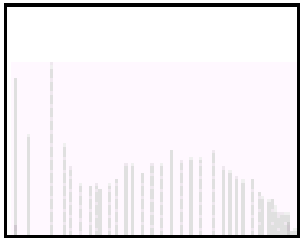


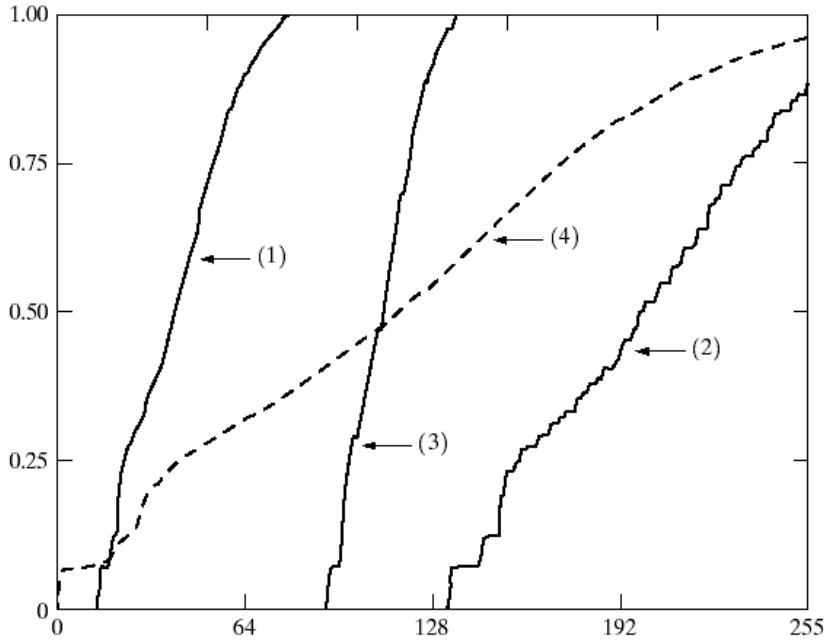


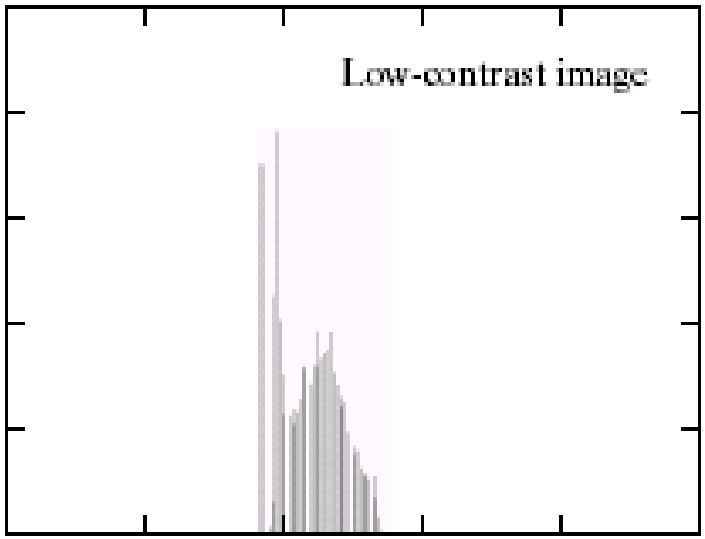


(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Histogram Equalization (cont.)

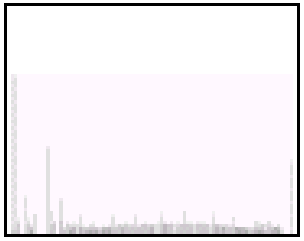


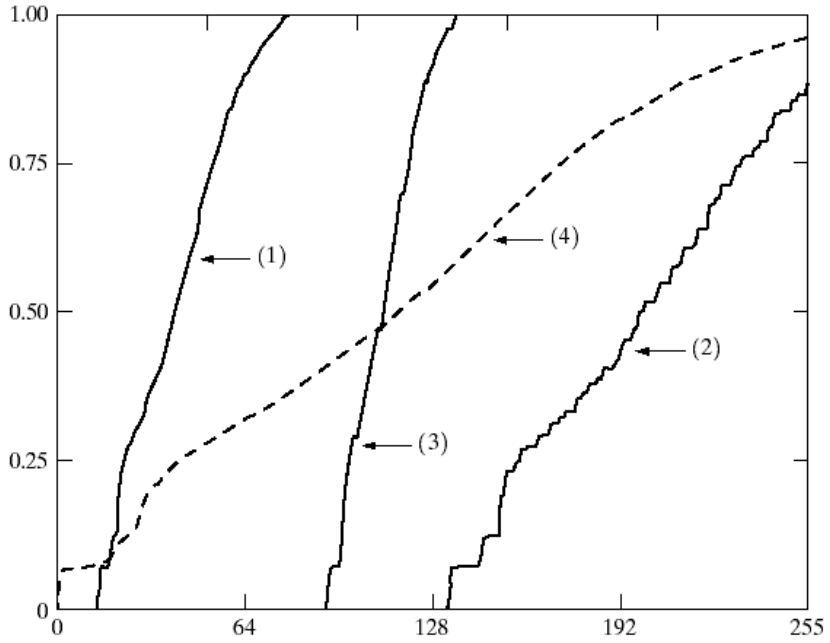


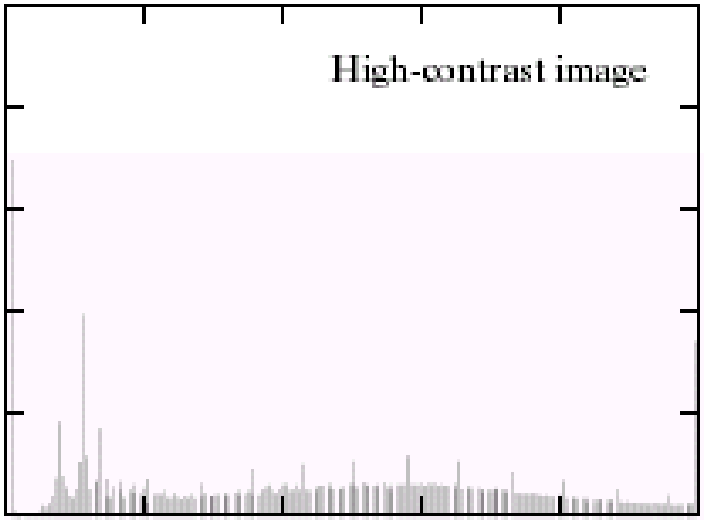


(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Histogram Equalization (cont.)

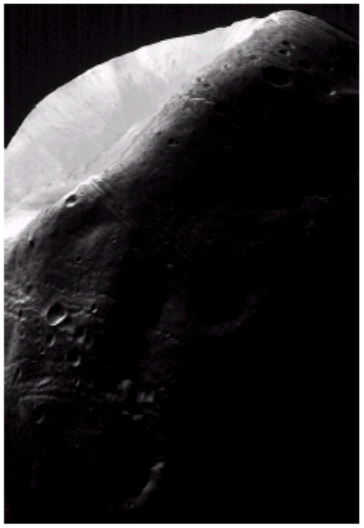


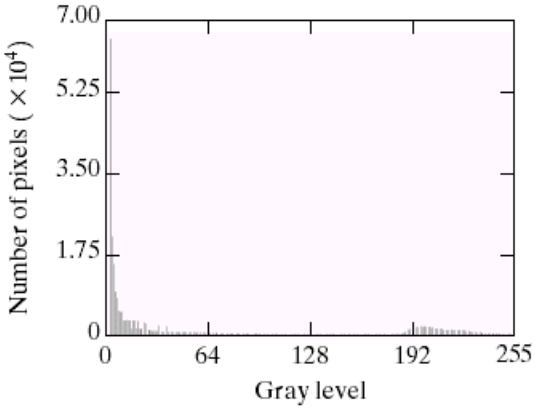


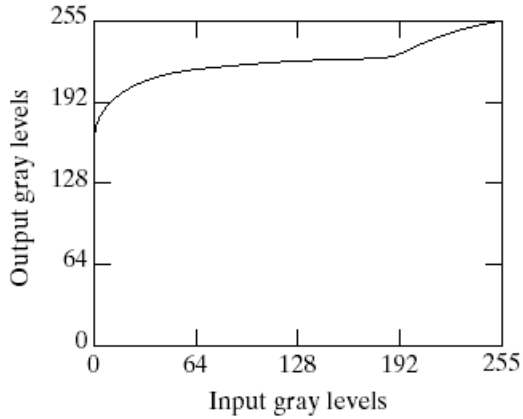


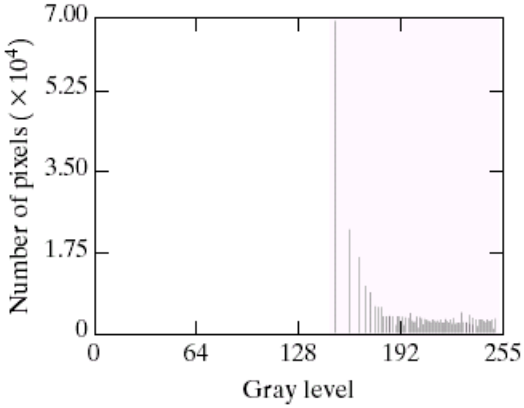
(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

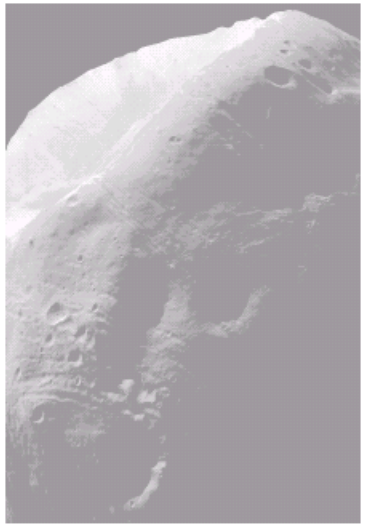
Histogram Equalization (cont.)







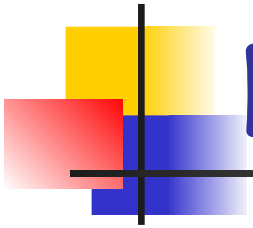




Original image

After histogram equalization, the image become a low contrast image

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.



Note

- It is clearly seen that
  - Histogram equalization distributes the gray level to reach the maximum gray level (white) because the cumulative distribution function equals 1 when $0 \leq r \leq L-1$
  - If the cumulative numbers of gray levels are slightly different, they will be mapped to little different or same gray levels as we may have to approximate the processed gray level of the output image to integer number
  - Thus the discrete transformation function can't guarantee the one to one mapping relationship

Homework

Perform histogram equalization of the following image (having gray level 0-7)

i)

0	0	0	7
0	0	7	7
0	0	7	7
0	7	7	7

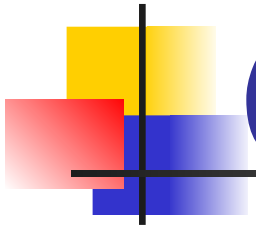
ii)

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

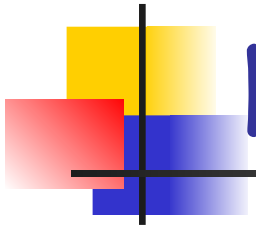
iii)

4	4	4	4
4	4	4	4
4	4	4	4
4	4	4	4

Histogram Matching (Specification)

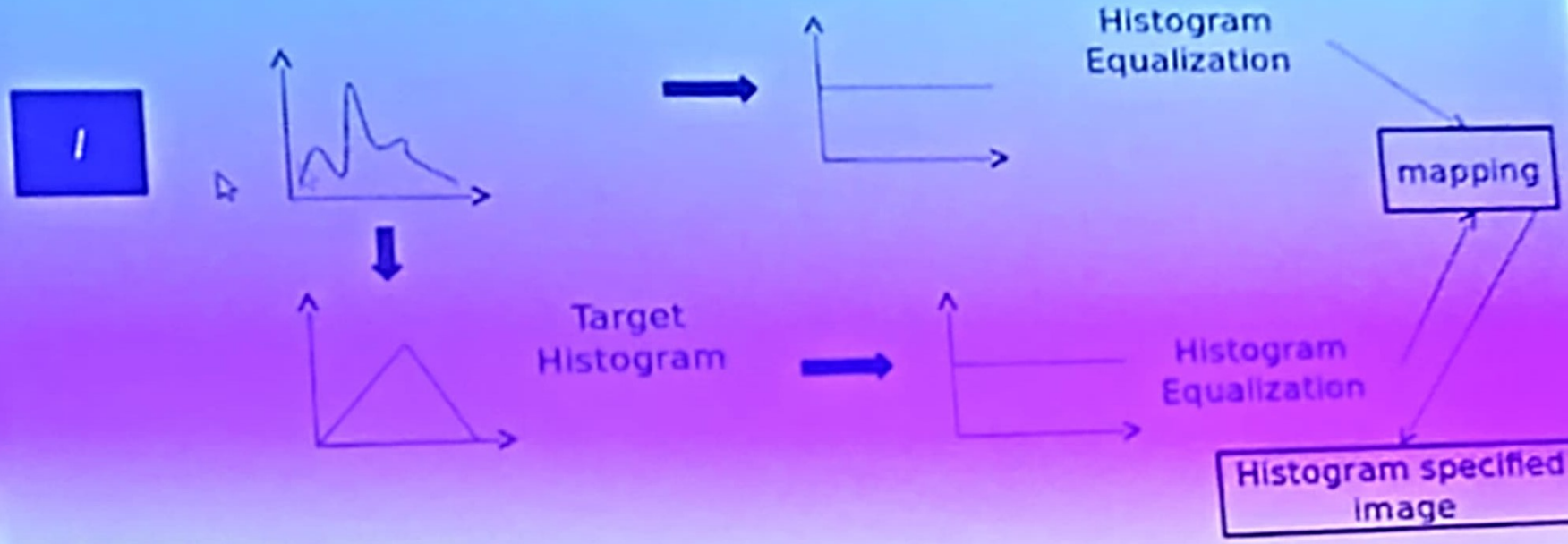


- Histogram equalization has a disadvantage which is that it can generate only one type of output image.
- With Histogram Specification, we can specify the shape of the histogram that we wish the output image to have.
- It doesn't have to be a uniform histogram



Note

- Histogram specification is a trial-and-error process
- There are no rules for specifying histograms, and one must resort to analysis on a case-by-case basis for any given enhancement task.


I
Histogram Equalization
mapping
Target Histogram
Histogram Equalization
Histogram specified image

Histogram Matching : Algorithm

To transform image histogram to be a desired histogram

Concept : from Histogram equalization, we have

$$s = T(r) = \int_0^r p_r(w)dw$$

We get $p_s(s) = 1$

We want an output image to have PDF $p_z(z)$

Apply histogram equalization to $p_z(z)$, we get

$$v = G(z) = \int_0^z p_z(u)du$$

We get $p_v(v) = 1$

Since $p_s(s) = p_v(v) = 1$ therefore s and v are equivalent

Therefore, we can transform r to z by

r → T( ) → s → G^{-1}( ) → z

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

$$= (L-1)\sum_{j=0}^{k} \frac{n_j}{N} \qquad k = 0,1,\ldots L-1$$

n_j = the number of pixels with intensity r_j
N = the number of total pixels

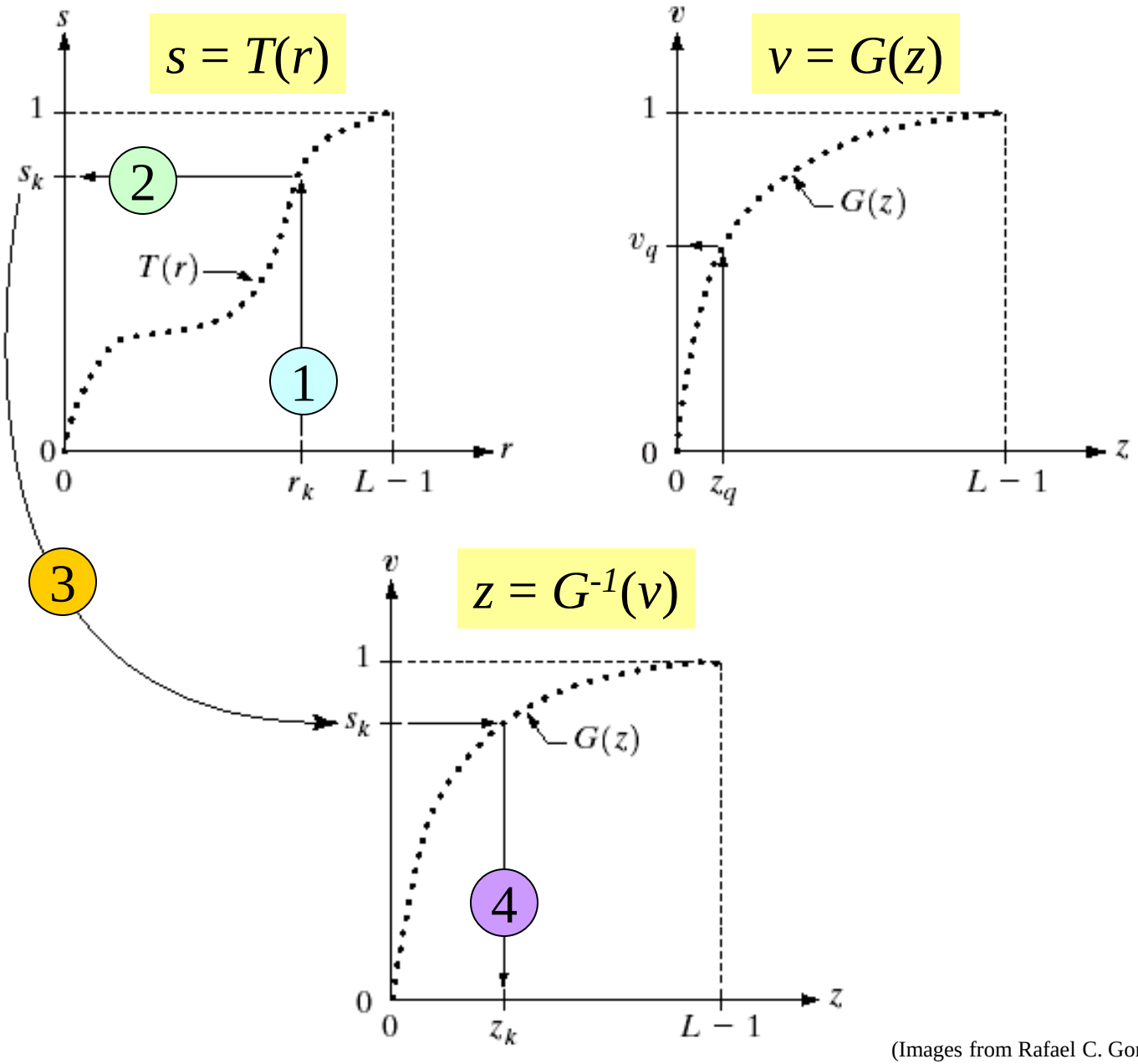
$$v_k = G(z_k) = (L-1)\sum_{j=0}^{q} p_z(z_i) = s_k$$

$$k = 0,1,\ldots L-1$$

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

Note: All mapping from r to s and from s to z are table lookup between a given pixel value

Histogram Matching : Algorithm (cont.)



(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

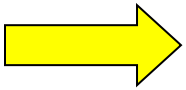
Histogram Matching Example

Example

Input image histogram

Intensity (s)	# pixels
0	20
1	5
2	25
3	10
4	15
5	5
6	10
7	10
Total	100

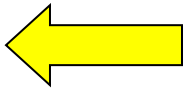
Original data



Desired Histogram

Intensity (z)	# pixels
0	5
1	10
2	15
3	20
4	20
5	15
6	10
7	5
Total	100

User define



Histogram Matching Example (cont.)

1. Apply Histogram Equalization to both tables

r	(n_j)	ΣP_r	s
0	20	0.2	1
1	5	0.25	2
2	25	0.5	3
3	10	0.6	4
4	15	0.75	5
5	5	0.8	6
6	10	0.9	6
7	10	1.0	7

$s_k = T(r_k)$

z	(n_j)	ΣP_z	v
0	5	0.05	0
1	10	0.15	1
2	15	0.3	2
3	20	0.5	4
4	20	0.7	5
5	15	0.85	6
6	10	0.95	7
7	5	1.0	7

$v_k = G(z_k)$

Histogram Matching Example (cont.)

2. Get a map

r → s

r	s
0	1
1	2
2	3
3	4
4	5
5	6
6	6
7	7

$s_k = T(r_k)$

s → v

v → z

v	z
0	0
1	1
2	2
4	3
5	4
6	5
7	6
7	7

$z_k = G^{-1}(v_k)$

We get

r	z
0	1
1	2
2	2
3	3
4	4
5	5
6	5
7	6

Actual Output Histogram

z	# Pixels
0	0
1	20
2	30
3	10
4	15
5	15
6	10
7	0

0	2	1	3	4
1	3	4	3	3
0	1	3	1	4
3	1	4	2	0
0	4	2	4	4

input: Image A

Pixel Value	Histogram	Equalized Histogram
0	4	4
1	5	9
2	3	12
3	6	18
4	7	25

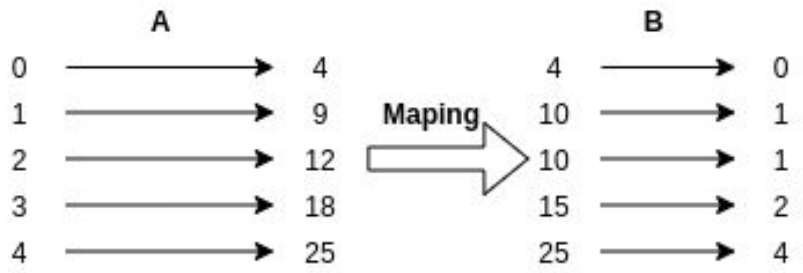
Equalized Histogram of A

2	1	2	1	0
3	3	2	4	4
1	3	2	4	4
0	0	3	2	1
1	3	1	4	0

Target: Image B

Pixel Value	Histogram	Equalized Histogram
0	4	4
1	6	10
2	5	15
3	5	20
4	5	25

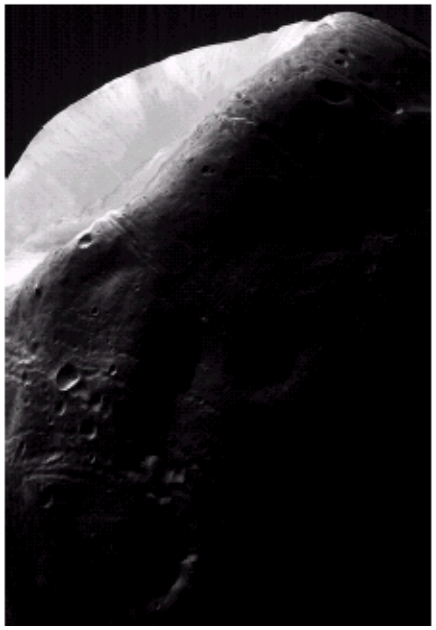
Equalized Histogram of B

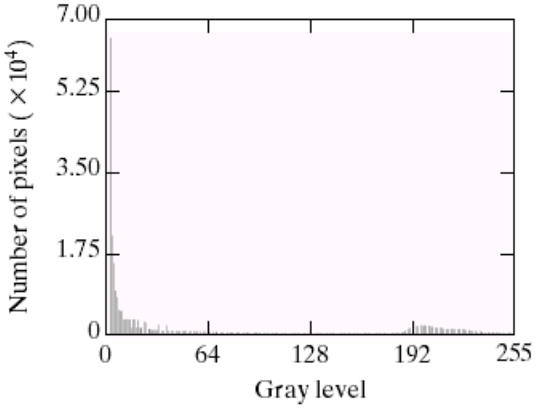


0	1	1	2	4
1	2	4	2	2
0	1	2	1	4
2	1	4	1	0
0	4	1	4	4

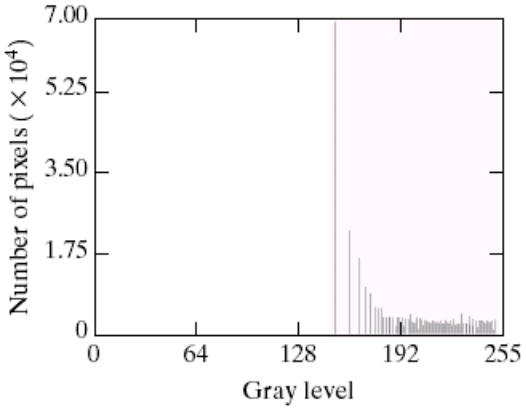
Modified: Image A

Histogram Matching Example (cont.)

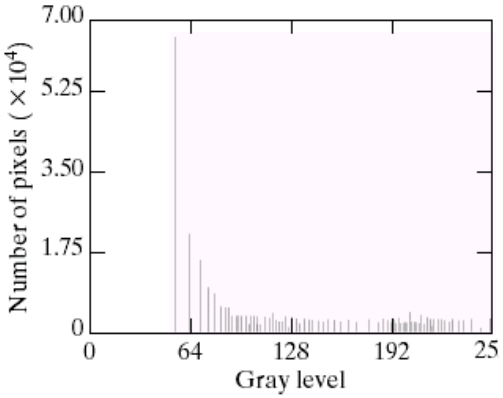




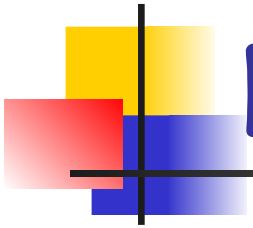
Original image



After histogram equalization



After histogram matching



Note

- Histogram processing methods are global processing, in the sense that pixels are modified by a transformation function based on the gray-level content of an entire image.
- Sometimes, we may need to enhance details over small areas in an image, which is called a local enhancement.

Perform histogram specification of the 8x8 image.

1	2	5	5	5	6	7	7
2	1	4	4	5	2	4	4
4	4	2	0	5	6	4	4
4	4	0	2	5	5	2	5
2	2	0	0	5	5	5	5
6	2	0	5	5	2	0	0
1	1	3	5	5	3	0	1
6	1	1	4	4	1	1	1

Grey levels (r_k)	0	1	2	3	4	5	6	7
Number of pixels (n_k)	9	10	10	5	15	10	4	3

Homework

- Perform histogram specification of the 8×8 image.

Target

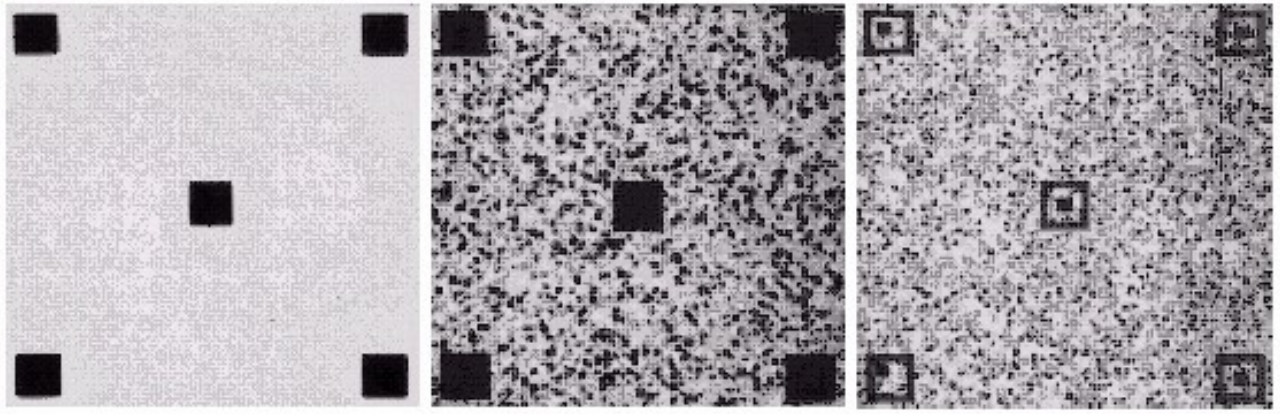
Grey levels (r_k)	0	1	2	3	4	5	6	7
Number of pixels (p_k)	0	0	0	0	20	20	16	8

Answer

Grey levels (r_k)	0	1	2	3	4	5	6	7
Map to	4	4	5	5	6	6	7	7

- computationally intensive, can be used in local sense

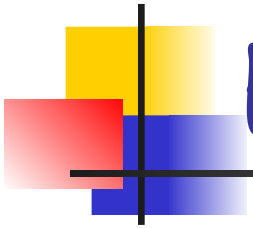
Local Enhancement



(a) (b) (c)

a) Original image (slightly blurred to reduce noise)
b) global histogram equalization (enhance noise & slightly increase contrast but the construction is not changed)
c) local histogram equalization using 7x7 neighborhood (reveals the small squares inside larger ones of the original image.

- define a square or rectangular neighborhood and move the center of this area from pixel to pixel.
- at each location, the histogram of the points in the neighborhood is computed and either histogram equalization or histogram specification transformation function is obtained.
- another approach used to reduce computation is to utilize nonoverlapping regions, but it usually produces an undesirable checkerboard effect.



Explain the result in c)

- Basically, the original image consists of many small squares inside the larger dark ones.
- However, the small squares were too close in gray level to the larger ones, and their sizes were too small to influence global histogram equalization significantly.
- So, when we use the local enhancement technique, it reveals the small areas.
- Note also the finer noise texture is resulted by the local processing using relatively small neighborhoods.

Local Enhancement : Histogram Statistic for Image Enhancement

We can use statistic parameters such as Mean, Variance of Local area for image enhancement

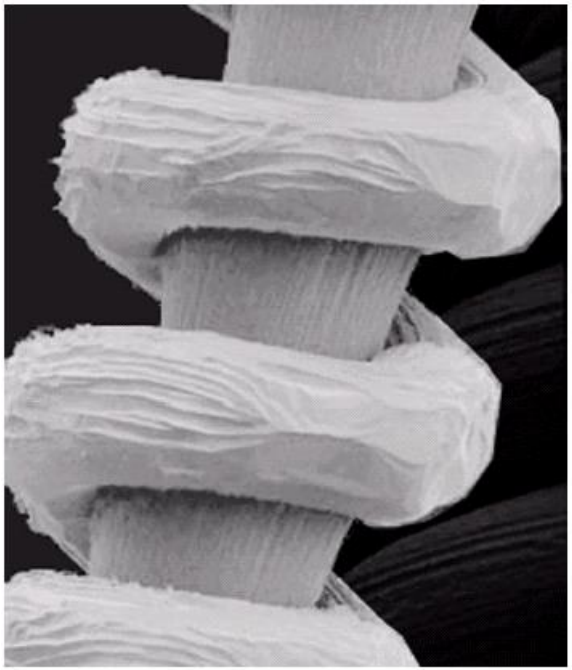


Image of tungsten filament taken using An electron microscope

In the lower right corner, there is a filament in the background which is very dark and we want this to be brighter.

We cannot increase the brightness of the whole image since the white filament will be too bright.

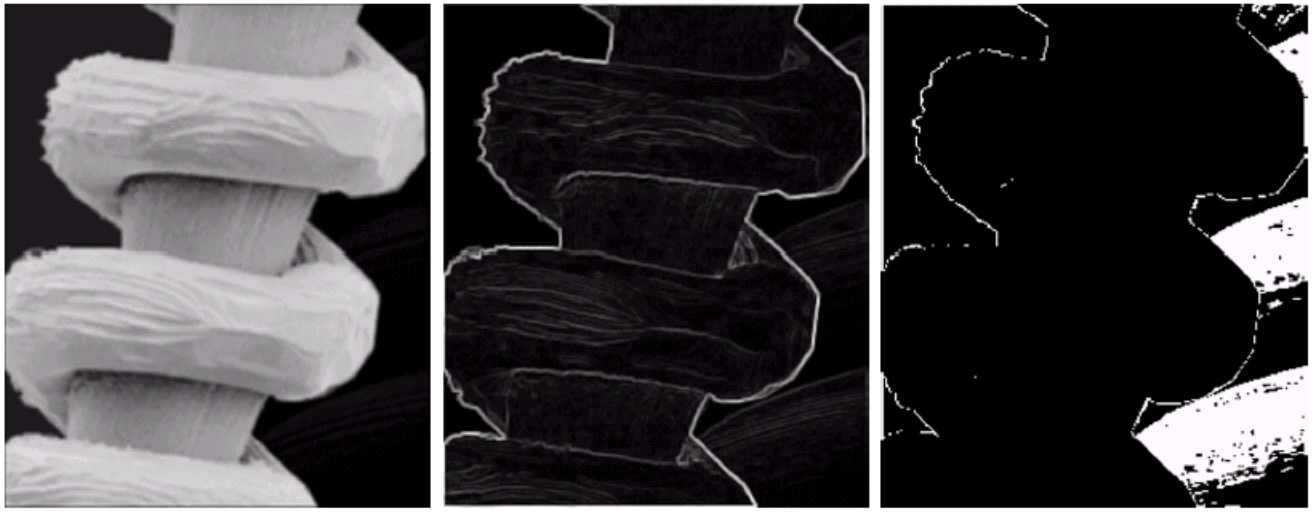
(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Local Enhancement

Example: Local enhancement for this task

$$g(x,y) = \begin{cases} E \cdot f(x,y) & \text{when } m_{s_{xy}} \le k_0 M_G \text{ and } k_1 D_G \le \sigma_{s_{xy}} \le k_2 M_G \\ f(x,y) & \text{otherwise} \end{cases}$$

Original image | Local Variance image | Multiplication factor



a b c

FIGURE 3.25 (a) Image formed from all local means obtained from Fig. 3.24 using Eq. (3.3-21). (b) Image formed from all local standard deviations obtained from Fig. 3.24 using Eq. (3.3-22). (c) Image formed from all multiplication constants used to produce the enhanced image shown in Fig. 3.26.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Local Enhancement

Example: Local enhancement for this task

Global SD

$$g(x,y) = \begin{cases} E \cdot f(x,y) & \text{when } m_{S_{xy}} \le k_0 m_G \text{ and } k_1 \sigma_G \le \sigma_{S_{xy}} \le k_2 \sigma_G \\ f(x,y) & \text{otherwise} \end{cases}$$

global mean

Local standard deviation

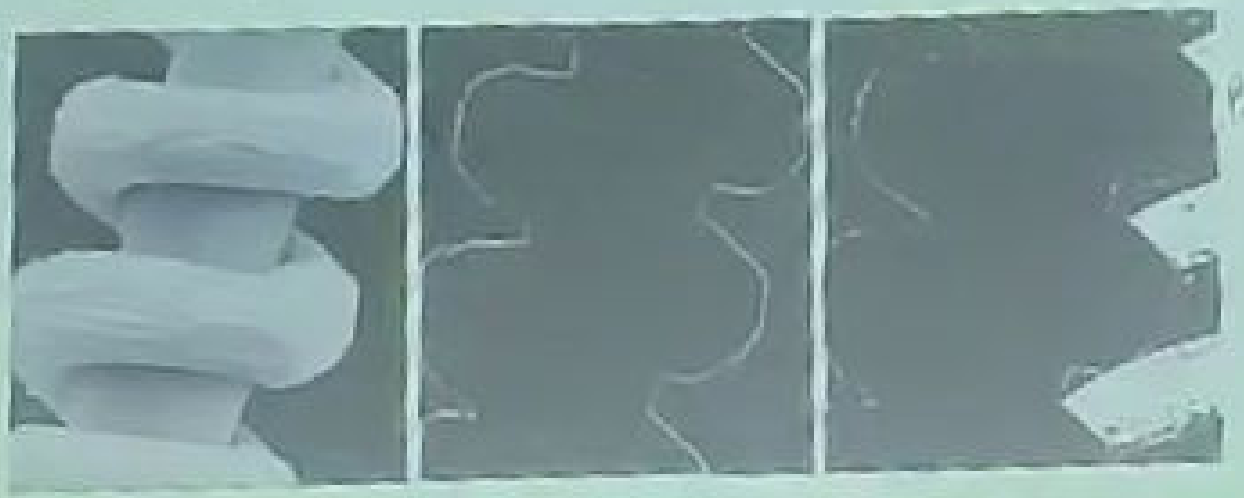
Specified constant

Local mean

Local Variance image

Multiplication factor

Original image



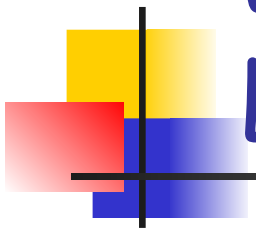
Local Enhancement

FIGURE 3.26
Enhanced SEM image. Compare with Fig. 3.24. Note in particular the enhanced area on the right side of the image.

Output image

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

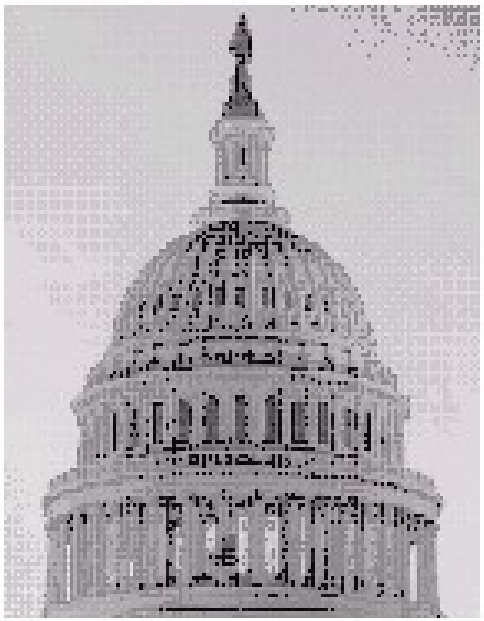
Enhancement using Arithmetic/ Logic Operations



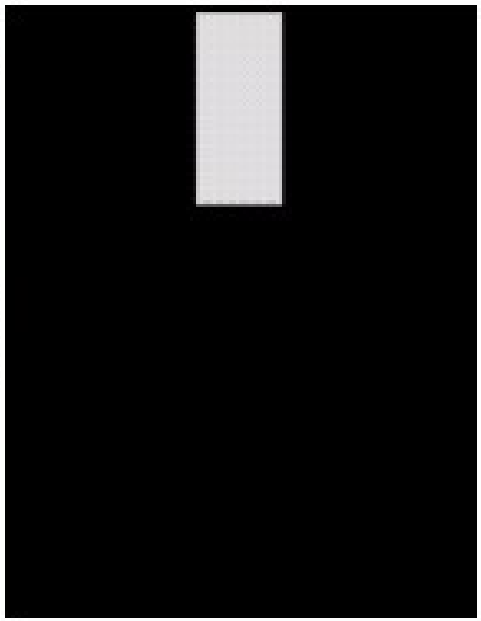
- Arithmetic/Logic operations perform on pixel by pixel basis between two or more images
- except NOT operation which perform only on a single image

Logical AND or OR operations are used for masking

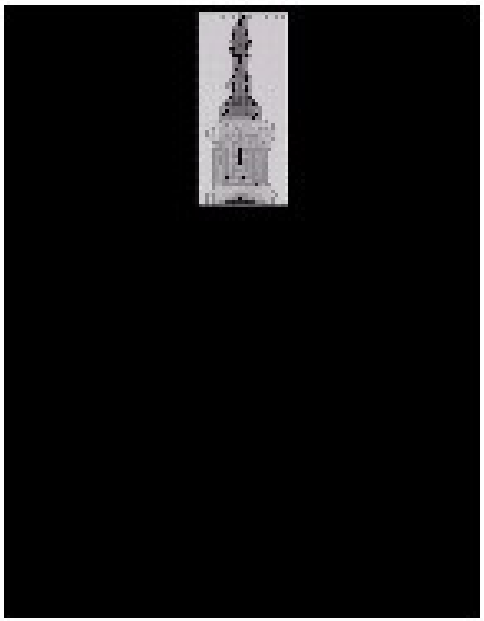
Example of AND Operation



original image



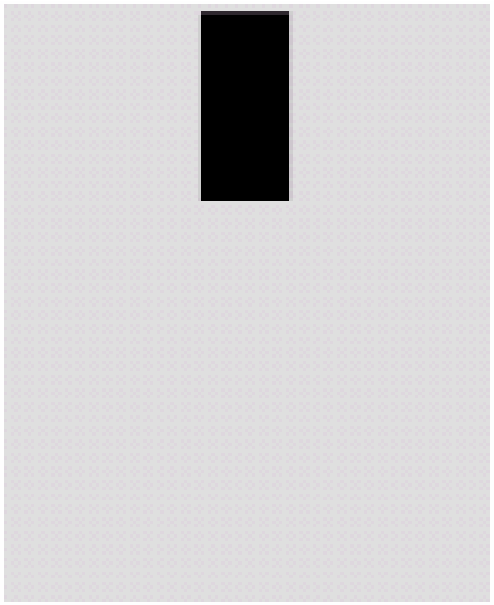
AND image mask



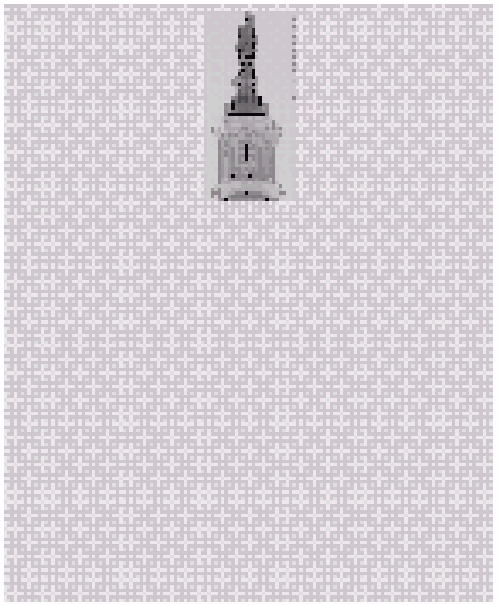
result of AND operation

Example of OR Operation

original image



OR image mask



result of OR operation

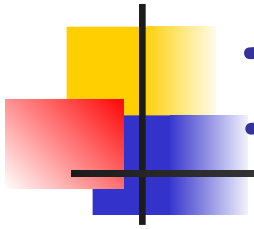


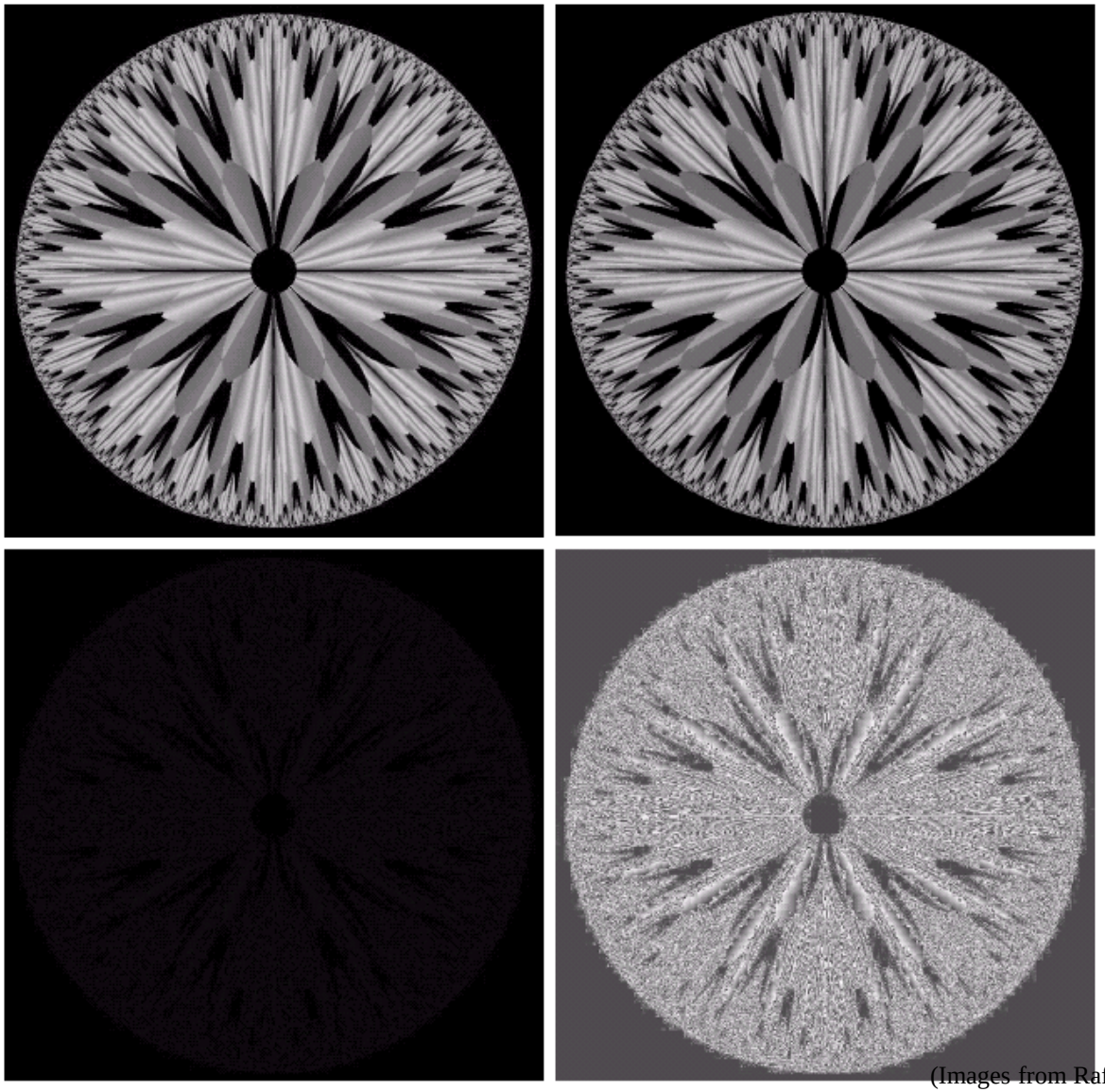
Image Subtraction

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

- enhancement of the differences between images

Arithmetic Operation: Subtraction

Application: Error measurement



a b
c d

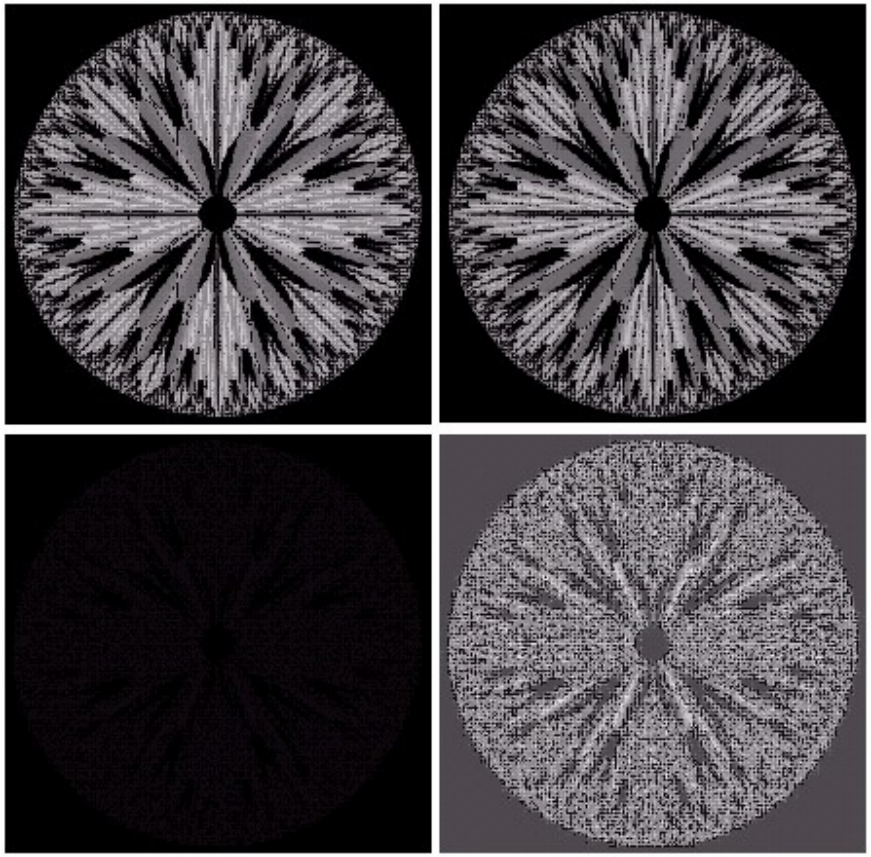
FIGURE 3.28
(a) Original fractal image. (b) Result of setting the four lower-order bit planes to zero. (c) Difference between (a) and (b). (d) Histogram-equalized difference image. (Original image courtesy of Ms. Melissa D. Binde, Swarthmore College, Swarthmore, PA).

Error image

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

a	b
c	d

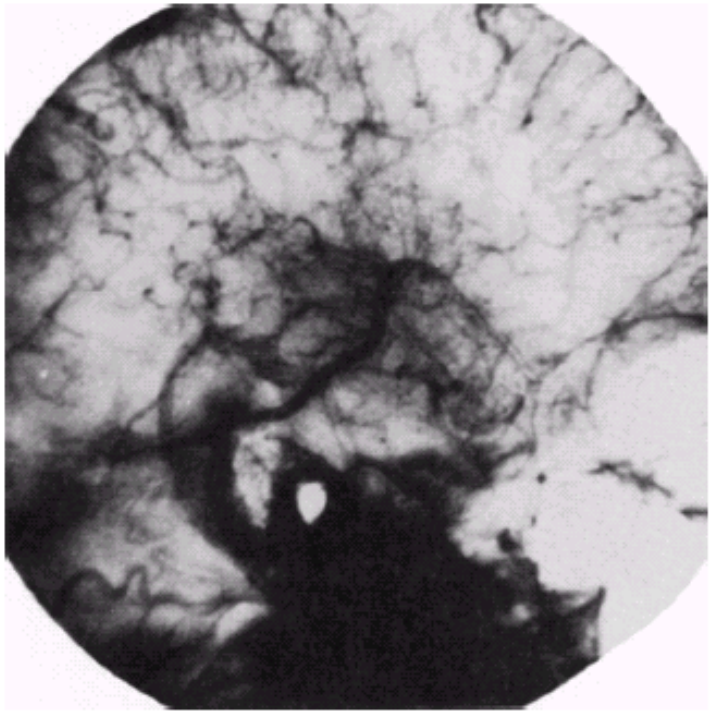
Image Subtraction

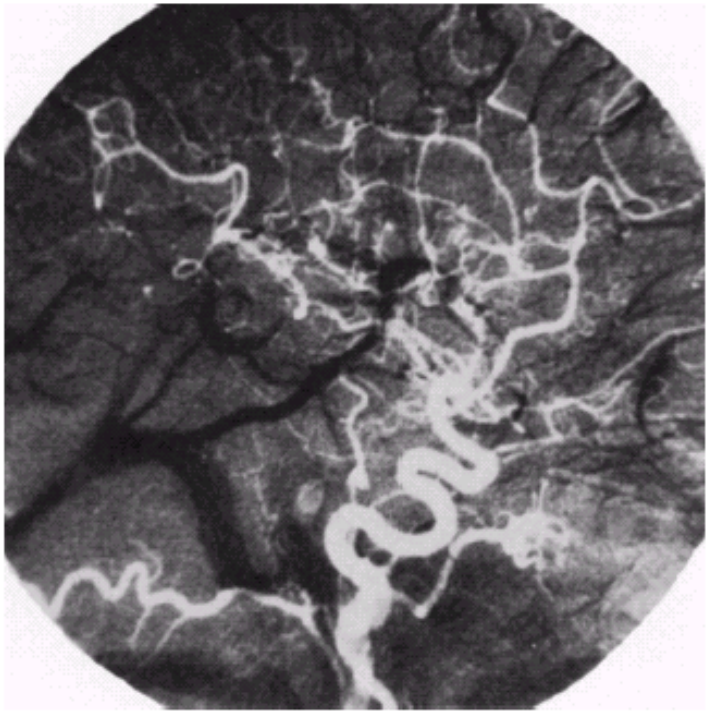


- a). original fractal image
- b). result of setting the four lower-order bit planes to zero
  - refer to the bit-plane slicing
  - the higher planes contribute significant detail
  - the lower planes contribute more to fine detail
  - image b). is nearly identical visually to image a), with a very slightly drop in overall contrast due to less variability of the gray-level values in the image.
- c). difference between a). and b). (nearly black)
- d). histogram equalization of c). (perform contrast stretching transformation)

Arithmetic Operation: Subtraction (cont.)

Application: Mask mode radiography in angiography work

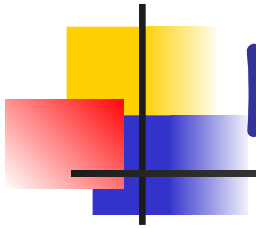




a b

FIGURE 3.29
Enhancement by image subtraction. (a) Mask image. (b) An image (taken after injection of a contrast medium into the bloodstream) with mask subtracted out.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.



Note

- We may have to adjust the gray-scale of the subtracted image to be [0, 255] (if 8-bit is used)
  - first, find the minimum gray value of the subtracted image
  - second, find the maximum gray value of the subtracted image
  - set the minimum value to be zero and the maximum to be 255
  - while the rest are adjusted according to the interval [0, 255], by timing each value with 255/max
- Subtraction is also used in segmentation of moving pictures to track the changes
  - after subtract the sequenced images, what is left should be the moving elements in the image, plus noise

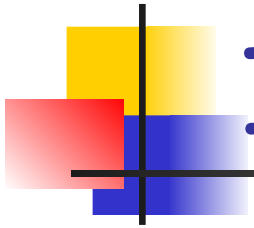


Image Averaging

- consider a noisy image g(x,y) formed by the addition of noise $\eta(x,y)$ to an original image f(x,y)

$$g(x,y) = f(x,y) + \eta(x,y)$$

Image Averaging

- if noise has zero mean and be uncorrelated then it can be shown that if

$\bar{g}(x,y)$ = image formed by averaging K different noisy images

$$\bar{g}(x,y) = \frac{1}{K}\sum_{i=1}^{K} g_i(x,y)$$

Image Averaging

- then

$$\sigma^2_{\bar{g}(x,y)} = \frac{1}{K}\sigma^2_{\eta(x,y)}$$

$\sigma^2_{\bar{g}(x,y)}, \sigma^2_{\eta(x,y)}$ = variances of $\bar{g}$ and η

if K increase, it indicates that the variability (noise) of the pixel at each location (x,y) decreases.

Image Averaging

- thus

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

$E\{\bar{g}(x,y)\}$ = expected value of $\bar{g}$ (output after averaging)

= original image f(x,y)

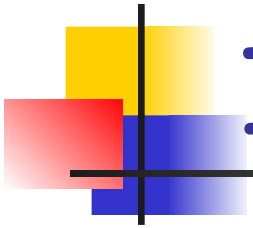
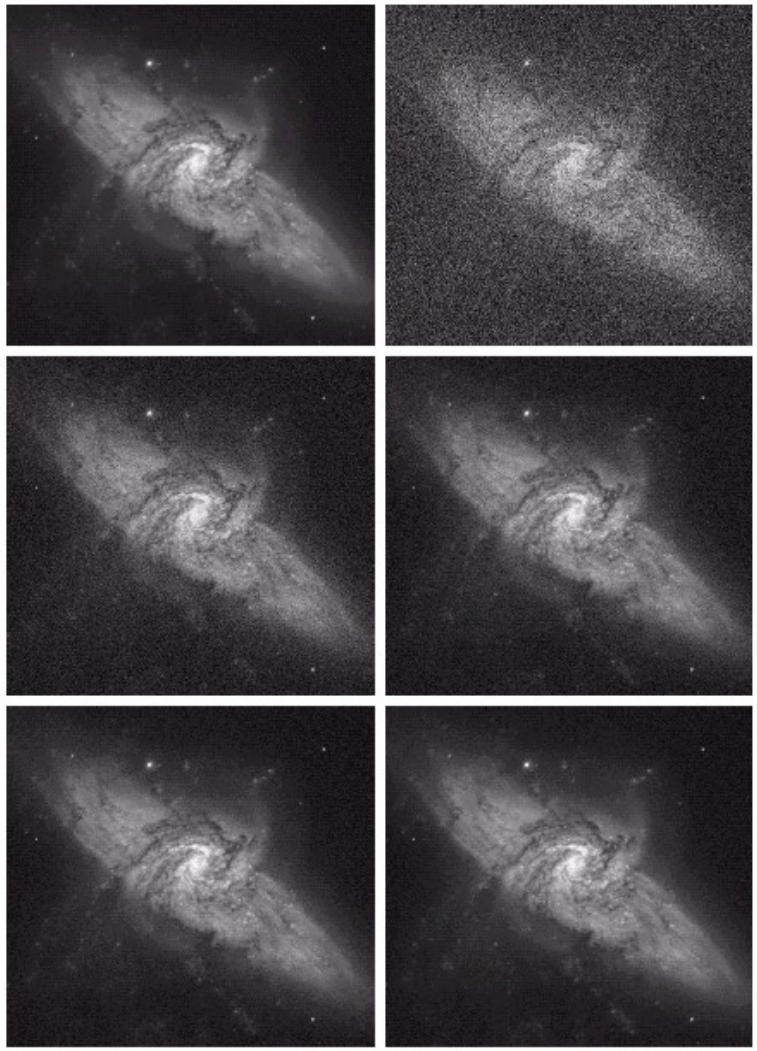


Image Averaging

- Note: the images $g_i(x,y)$ (noisy images) must be registered (aligned) in order to avoid the introduction of blurring and other artifacts in the output image.

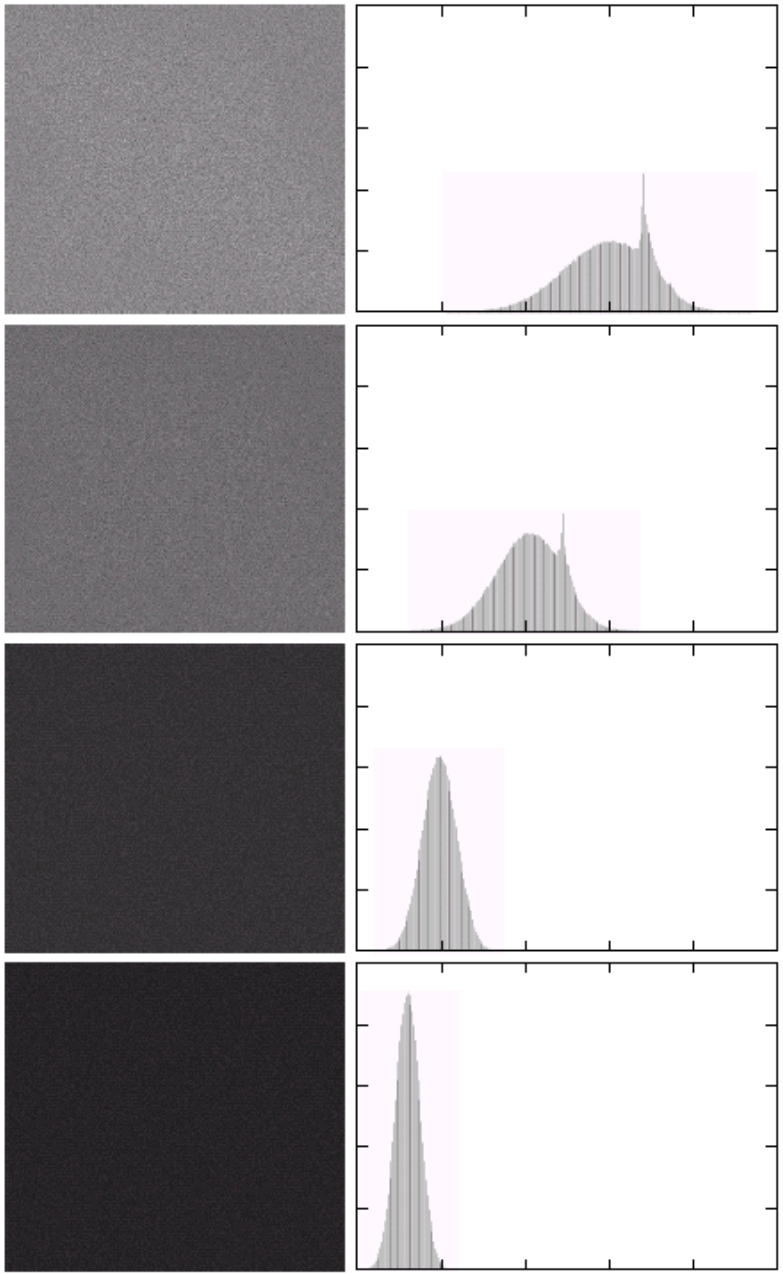
a	b
c	d
e	f

Example



- a) original image
- b) image corrupted by additive Gaussian noise with zero mean and a standard deviation of 64 gray levels.
- c). -f). results of averaging K = 8, 16, 64 and 128 noisy images

Arithmetic Operation: Image Averaging (cont.)



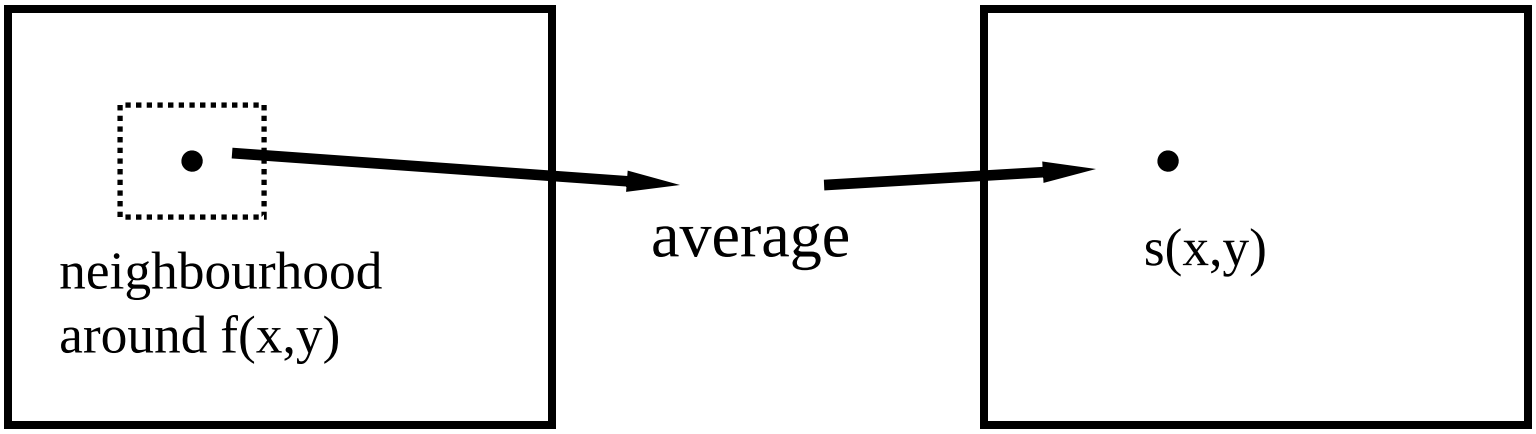
a b

FIGURE 3.31
(a) From top to bottom: Difference images between Fig. 3.30(a) and the four images in Figs. 3.30(c) through (f), respectively. (b) Corresponding histograms.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Spatial Filtering

A common application of spatial filtering is image smoothing using an averaging filter, or averaging mask, or kernel.

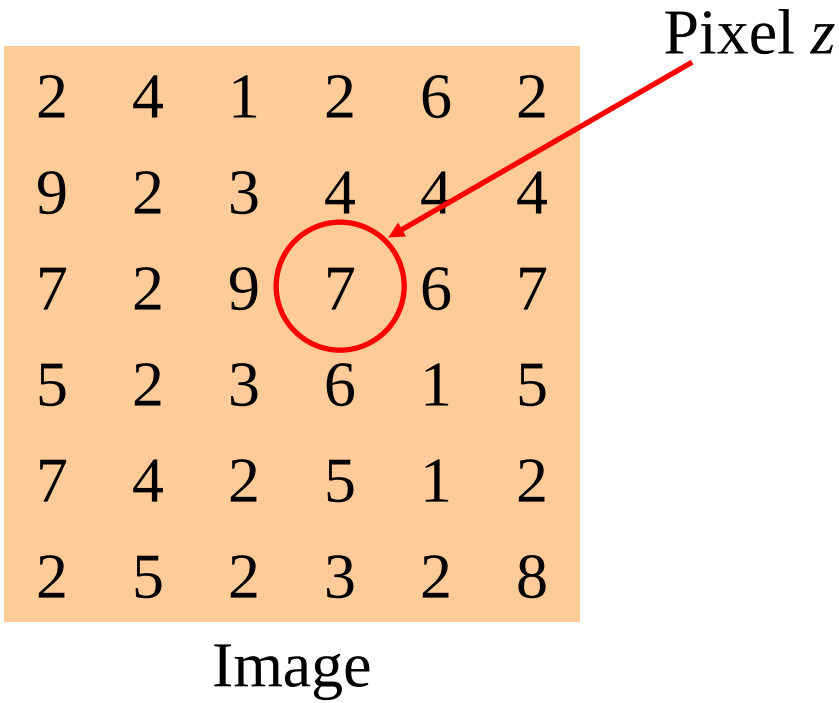


Each point in the smoothed image, g(x,y) is obtained from the average pixel value in a neighbourhood of (x,y) in the input image.

Basics of Spatial Filtering

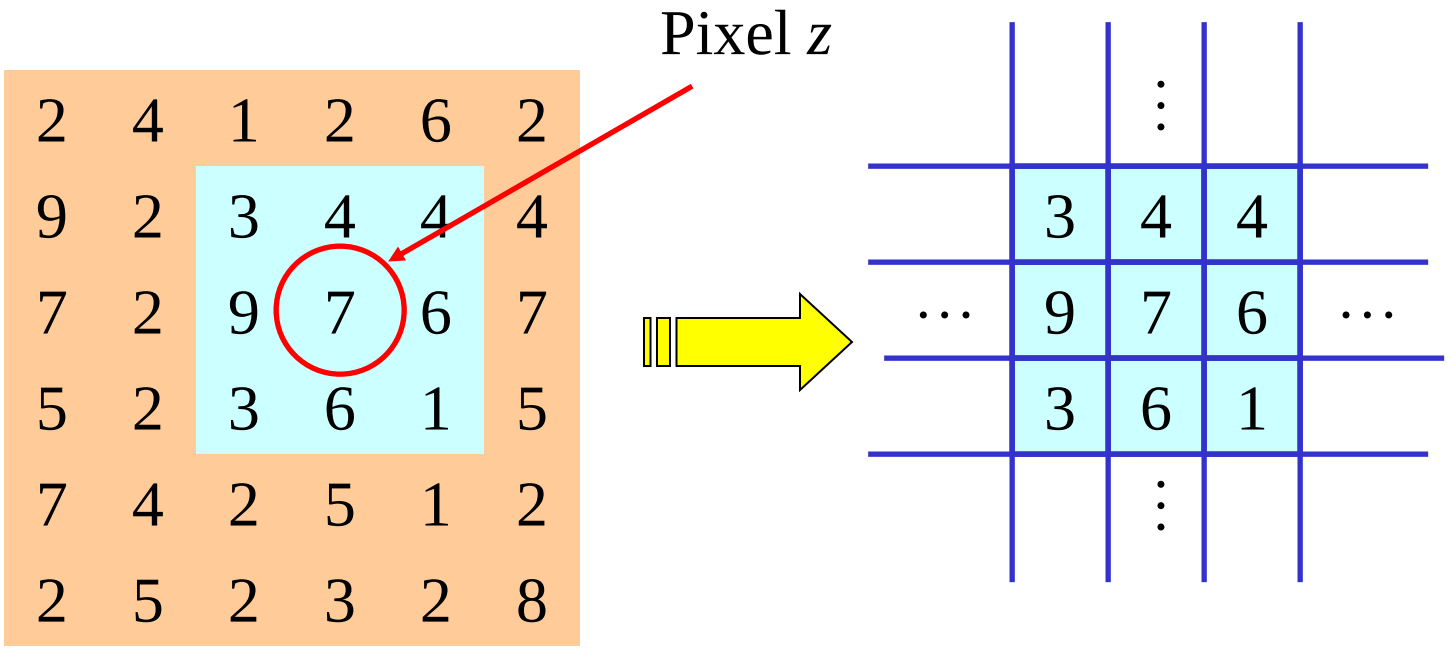
Sometime we need to manipulate values obtained from neighboring pixels

Example: How can we compute an average value of pixels in a 3x3 region center at a pixel *z*?



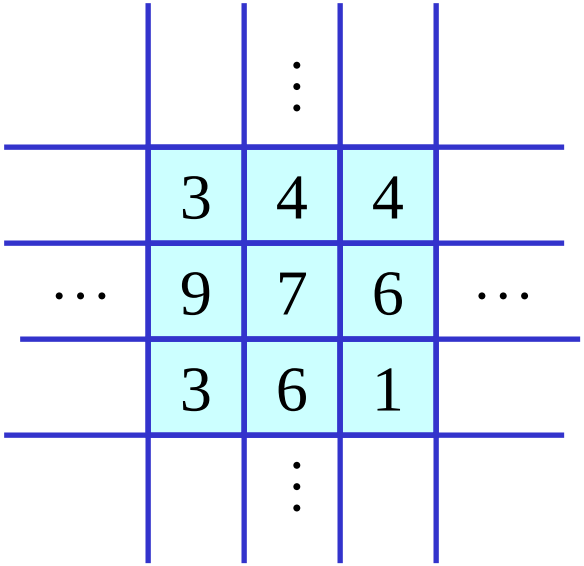
Basics of Spatial Filtering (cont.)

Step 1. Selected only needed pixels



Basics of Spatial Filtering (cont.)

Step 2. Multiply every pixel by 1/9 and then sum up the values



X

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

Mask or
Window or
Template

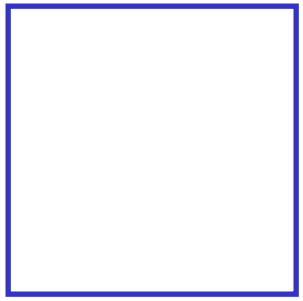
$$y = \frac{1}{9}\cdot 3 + \frac{1}{9}\cdot 4 + \frac{1}{9}\cdot 4 + \frac{1}{9}\cdot 9 + \frac{1}{9}\cdot 7 + \frac{1}{9}\cdot 6 + \frac{1}{9}\cdot 3 + \frac{1}{9}\cdot 6 + \frac{1}{9}\cdot 1$$

Basics of Spatial Filtering (cont.)

Question: How to compute the 3x3 average values at every pixels?

2	4	1	2	6	2
9	2	3	4	4	4
7	2	9	7	6	7
5	2	3	6	1	5
7	4	2	5	1	2

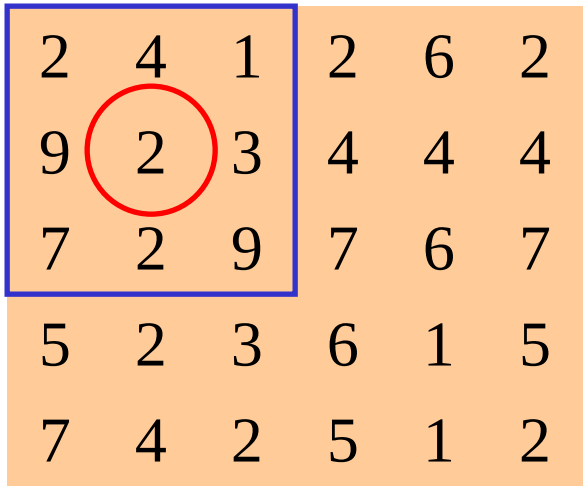
Solution: Imagine that we have a 3x3 window that can be placed everywhere on the image



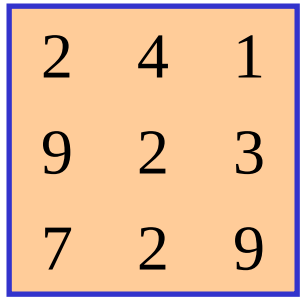
Masking Window

Basics of Spatial Filtering (cont.)

Step 1: Move the window to the first location where we want to compute the average value and then select only pixels inside the window.



Original image

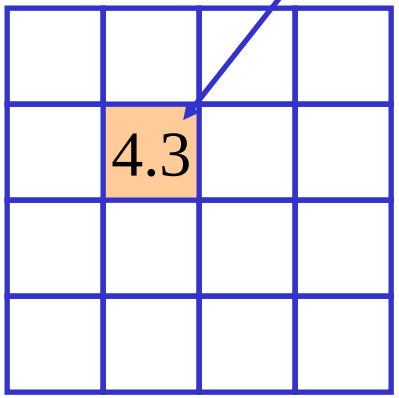


Sub image *p*

Step 2: Compute the average value

$$y = \sum_{i=1}^{3}\sum_{j=1}^{3}\frac{1}{9}\cdot p(i,j)$$

Step 3: Place the result at the pixel in the output image



Output image

Step 4: Move the window to the next location and go to Step 2

Basics of Spatial Filtering (cont.)

The 3x3 averaging method is one example of the ***mask operation*** or ***Spatial filtering*****.**

- The mask operation has the corresponding mask (sometimes called window or template).
- The mask contains coefficients to be multiplied with pixel values.

w(1,1)	w(2,1)	w(3,1)
w(1,2)	w(2,2)	w(3,2)
w(3,1)	w(3,2)	w(3,3)

Mask coefficients

Example : moving averaging

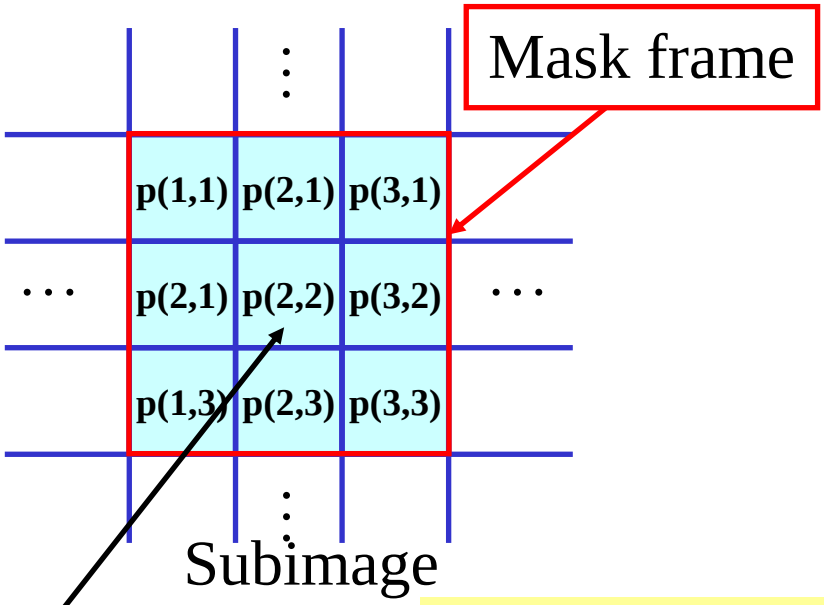
$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

The mask of the 3x3 moving average filter has all coefficients = 1/9

Basics of Spatial Filtering (cont.)

The mask operation at each point is performed by:

1. Move the reference point (center) of mask to the location to be computed
2. Compute ***sum of products*** between mask coefficients and pixels in subimage under the mask.



w(1,1)	w(2,1)	w(3,1)
w(1,2)	w(2,2)	w(3,2)
w(3,1)	w(3,2)	w(3,3)

Mask coefficients

$$y = \sum_{i=1}^{N}\sum_{j=1}^{M} w(i,j)\cdot p(i,j)$$

Basics of Spatial Filtering (cont.)

The spatial filtering on the whole image is given by:

1. Move the mask over the image at each location.
2. Compute sum of products between the mask coefficeints and pixels inside subimage under the mask.
3. Store the results at the corresponding pixels of the output image.
4. Move the mask to the next location and go to step 2 until all pixel locations have been used.

Examples of Spatial Filtering Masks

Examples of the masks

Sobel operators

-1	0	1
-2	0	2
-1	0	1

to compute $\frac{\partial P}{\partial x}$

-1	-2	-1
0	0	0
1	2	1

to compute $\frac{\partial P}{\partial y}$

3x3 moving average filter

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

3x3 sharpening filter

$\frac{1}{9}$

-1	-1	-1
-1	8	-1
-1	-1	-1

Convolutions

Computationally, spatial filtering is implemented in a computer program via a mathematical process known as convolution.

If we assume a 3x3 neighbourhood the smoothing by averaging would correspond to convolving the image with the following filter, or mask.

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

This filter is convolved with the image by placing it over a 3x3 portion of the image, multiplying the overlaying pixel values and adding them all up to find the value that replaces the original central pixel value.

.1	.1	.1
.1	.1	.1
.1	.1	.1

Filter

3 x.1	2 x.1	1 x.1	2	2
4 x.1	2 x.1	5 x.1	3	3
8 x.1	6 x.1	3 x.1	1	1
4	7	2	1	6
6	5	3	2	8

Input image

-	-	-	-	-
-	3.4			-
-				-
-				

Filtered image

.1	.1	.1
.1	.1	.1
.1	.1	.1

Filter

3	2 x.1	1 x.1	2 x.1	2
4	2 x.1	5 x.1	3 x.1	3
8	6 x.1	3 x.1	1 x.1	1
4	7	2	1	6
6	5	3	2	8

Input image

-	-	-	-	-
-	3.4	2.5		-
-				-
-				

Filtered image

.1	.1	.1
.1	.1	.1
.1	.1	.1

Filter

3	2	1 x.1	2 x.1	2 x.1
4	2	5 x.1	3 x.1	3 x.1
8	6	3 x.1	1 x.1	1 x.1
4	7	2	1	6
6	5	3	2	8

Input image

-	-	-	-	-
-	3.4	2.5	2.1	-
-				-
-				

Filtered image

.1	.1	.1
.1	.1	.1
.1	.1	.1

Filter

3	2	1	2	2
4 x.1	2 x.1	5 x.1	3	3
8 x.1	6 x.1	3 x.1	1	1
4 x.1	7 x.1	2 x.1	1	6
6	5	3	2	8

Input image

-	-	-	-	-
-	3.4	2.5	2.1	-
	4.1			-
-				

Filtered image

.1	.1	.1
.1	.1	.1
.1	.1	.1

Filter

3	2	1	2	2
4	2 x.1	5 x.1	3 x.1	3
8	6 x.1	3 x.1	1 x.1	1
4	7 x.1	2 x.1	1 x.1	6
6	5	3	2	8

Input image

-	-	-	-	-
-	3.4	2.5	2.1	-
	4.1	3.0		-
-				

Filtered image

.1	.1	.1
.1	.1	.1
.1	.1	.1

Filter

3	2	1	2	2
4	2	5 x.1	3 x.1	3 x.1
8	6	3 x.1	1 x.1	1 x.1
4	7	2 x.1	1 x.1	6 x.1
6	5	3	2	8

Input image

-	-	-	-	-
-	3.4	2.5	2.1	-
	4.1	3.0	2.5	-
-				

Filtered image

.1	.1	.1
.1	.1	.1
.1	.1	.1

Filter

3	2	1	2	2
4	2	5	3	3
8 x.1	6 x.1	3 x.1	1	1
4 x.1	7 x.1	2 x.1	1	6
6 x.1	5 x.1	3 x.1	2	8

Input image

-	-	-	-	-
-	3.4	2.5	2.1	-
-	4.1	3.0	2.5	-
-	4.4			

Filtered image

.1	.1	.1
.1	.1	.1
.1	.1	.1

Filter

3	2	1	2	2
4	2	5	3	3
8	6 x.1	3 x.1	1 x.1	1
4	7 x.1	2 x.1	1 x.1	6
6	5 x.1	3 x.1	2 x.1	8

Input image

-	-	-	-	-
-	3.4	2.5	2.1	-
-	4.1	3.0	2.5	-
-	4.4	3.0		

Filtered image

.1	.1	.1
.1	.1	.1
.1	.1	.1

Filter

3	2	1	2	2
4	2	5	3	3
8	6	3 x.1	1 x.1	1 x.1
4	7	2 x.1	1 x.1	6 x.1
6	5	3 x.1	2 x.1	8 x.1

Input image

-	-	-	-	-
-	3.4	2.5	2.1	-
-	4.1	3.0	2.5	-
	4.4	3.0	2.7	

Filtered image

The mask is moved across the image until every pixel has been covered.
We convolve the image with the mask.

.1	.1	.1
.1	.1	.1
.1	.1	.1

Filter

3	2	1	2	2
4	2	5	3	3
8	6	3	1	1
4	7	2	1	6
6	5	3	2	8

Input image

-	-	-	-	-
-	3.4	2.5	2.1	-
-	4.1	3.0	2.5	-
-	4.4	3.0	2.7	-
-	-	-	-	-

Filtered image

Correlation

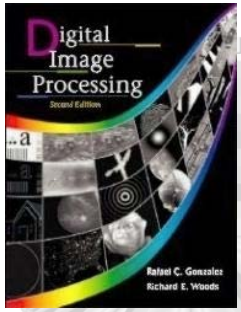
$$G[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} H[u,v]F[i+u, j+v]$$

$$= H \otimes F$$

Convolution

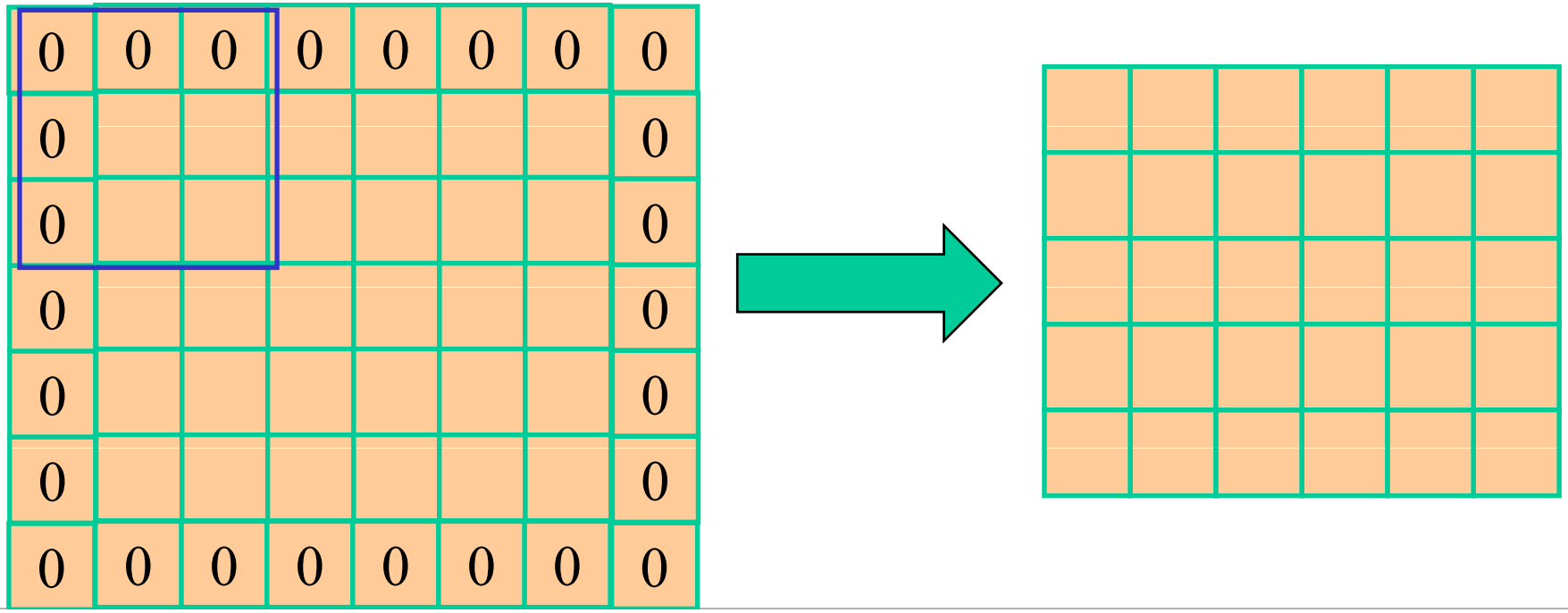
$$G[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} H[u,v]F[i-u, j-v]$$

$$G = H \star F$$



Chapter 3
Basics of Spatial Filtering

- Consider when the mask is moving near the edge of image. How can we keep the size of original image? If the size of the mask is *nxn*.
  - Pad the image by adding *(n-1)/2* rows or columns with 0's
  - Pad with Replicating adding *(n-1)/2* rows or columns.
- What are the effects?
- How to do these with matlab or OpenCV?



Size of padding

- For a gray scale (M x N) image and (m x n) filter/kernel, the dimensions of the image resulting from a convolution operation is **(M − m + 1) x (N − n + 1).**
- Sol– **Padding Input Images**

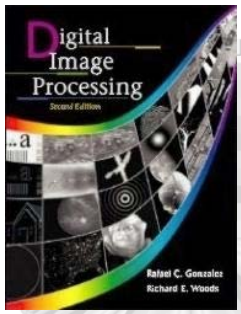
adding layers of zeros to input images so as to avoid the problems mentioned above.

p = number of layers of zeros added to the border of the image,

(M x N) image becomes **(M + 2p) x (N + 2p)**

Applying convolution-operation (with (m x n) filter) **outputs (M + 2p − m + 1) x (N + 2p − n + 1)** images.

p=(m-1)/2 for row p=(n-1)/2 for column



Chapter 3
Basics of Spatial Filtering

- Is this Spatial Filter is linear operation?
  - What are Linear Spatial Filters?

    (Example: Mean Filter)

  - What are Nonlinear Spatial Filters?

    (Example: Median Filter, Maximum and Minimum Filter)

Examples of Spatial Filtering Masks

Examples of the masks

Sobel operators

-1	0	1
-2	0	2
-1	0	1

to compute $\frac{\partial P}{\partial x}$

-1	-2	-1
0	0	0
1	2	1

to compute $\frac{\partial P}{\partial y}$

3x3 moving average filter

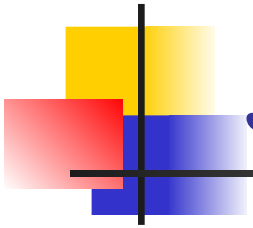
$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

3x3 sharpening filter

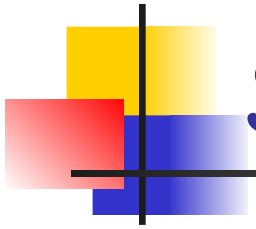
$\frac{1}{9}$

-1	-1	-1
-1	8	-1
-1	-1	-1



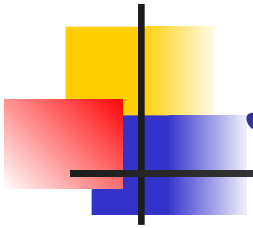
Smoothing Spatial Filters

- used for blurring and for noise reduction
- blurring is used in preprocessing steps, such as
  - removal of small details from an image prior to object extraction
  - bridging of small gaps in lines or curves
- noise reduction can be accomplished by blurring with a linear filter and also by a nonlinear filter



Smoothing Linear Filters

- output is simply the average of the pixels contained in the neighborhood of the filter mask.
- called averaging filters or lowpass filters.



Smoothing Linear Filters

- replacing the value of every pixel in an image by the average of the gray levels in the neighborhood will reduce the "sharp" transitions in gray levels.
- sharp transitions
  - random noise in the image
  - edges of objects in the image
- thus, smoothing can reduce noises (desirable) and blur edges (undesirable)

3x3 Smoothing Linear Filters

$\frac{1}{9} \times$

1	1	1
1	1	1
1	1	1

box filter

$\frac{1}{16} \times$

1	2	1
2	4	2
1	2	1

weighted average

the center is the most important and other pixels are inversely weighted as a function of their distance from the center of the mask

Mean /Average Filter

$\frac{1}{9} \times$

1	1	1
1	1	1
1	1	1

box filter

- output is simply the average of the pixels contained in the neighborhood of the filter mask.
- called averaging filters or lowpass filters.
- replacing the value of every pixel in an image by the average of the gray levels in the neighborhood will reduce the "sharp" transitions in gray levels.
- sharp transitions
  - random noise in the image
  - edges of objects in the image
- thus, smoothing can reduce noises (desirable) and blur edges (undesirable)

Mean Filters: Arithmetic mean filter

- Smooth local variations in an image
- Noise is reduced as a result of blurring
- $g(s,t)$: degraded image
- S_{xy}: set of coordinates in a rectangular subimage window of size $m \times n$

$$\hat{f}(x,y) = \frac{1}{mn} \sum_{(s,t) \in S_{xy}} g(s,t)$$

Causes a certain amount of blurring (proportional to the window size) to the image, thereby reducing the effects of noise.

Can be used to reduce noise of different types, but works best for Gaussian, uniform, or Erlang noise.

Mean Filters: Geometric mean filter

- Comparable to the arithmetic mean filter→ achieve smoothing; lose less image detail in the process

$$\hat{f}(x,y) = \left[\prod_{(s,t)\in S_{xy}} g(s,t) \right]^{\frac{1}{mn}}$$

- *A variation of the arithmetic mean filter*
- *Primarily used on images with Gaussian noise*
- *Retains image detail better than the arithmetic mean*

Weighted average filter

- the basic strategy behind weighting the center point the highest and then reducing the value of the coefficients as a function of increasing distance from the origin is simply an attempt to reduce blurring in the smoothing process.

$\frac{1}{16} \times$

1	2	1
2	4	2
1	2	1

weighted average

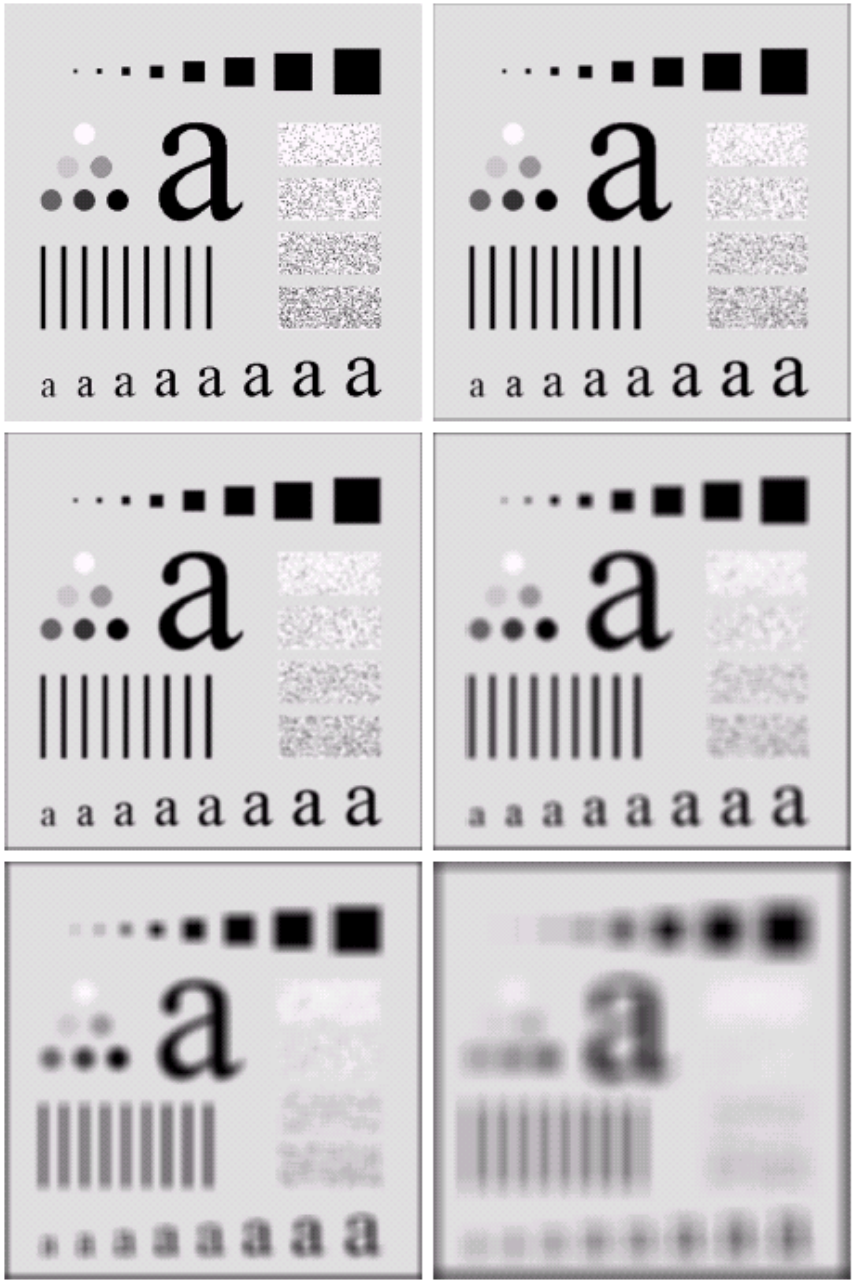
General form : smoothing mask

- filter of size mxn (m and n odd)

$$g(x,y) = \frac{\sum_{s=-a}^{a}\sum_{t=-b}^{b} w(s,t) f(x+s, y+t)}{\sum_{s=-a}^{a}\sum_{t=-b}^{b} w(s,t)}$$

summation of all coefficient of the mask

Smoothing Linear Filter : Moving Average



Application : noise reduction and image smoothing

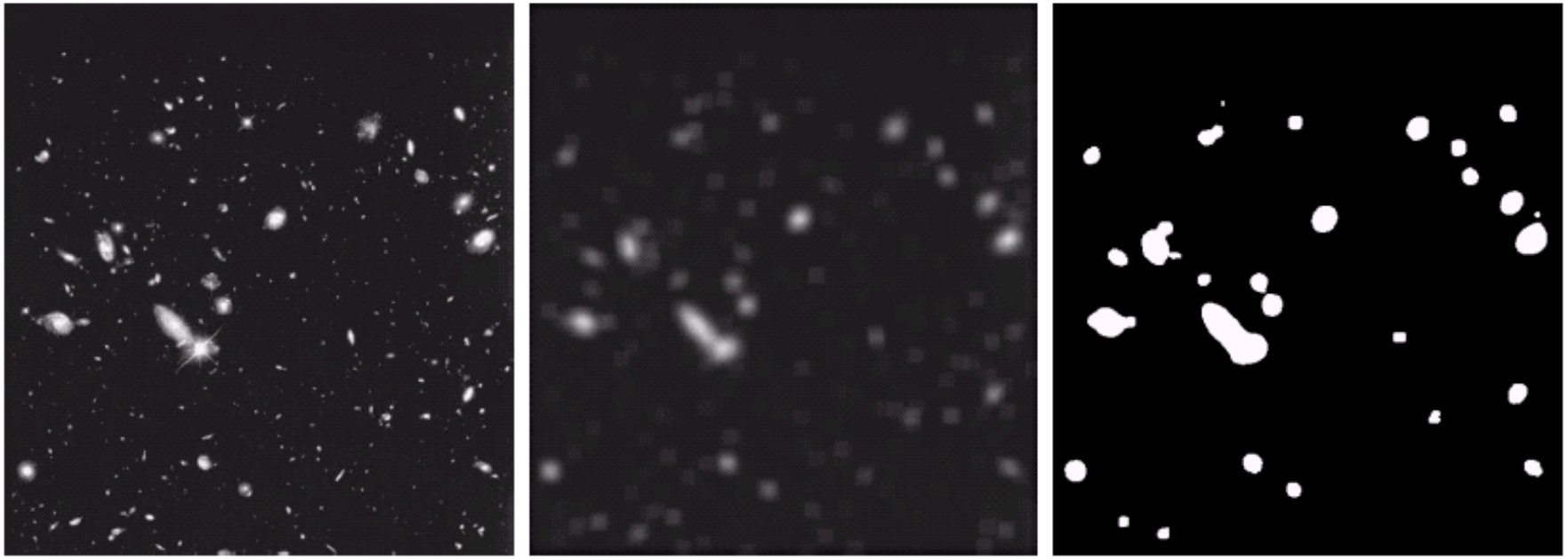
Disadvantage: lose sharp details

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

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.

Smoothing Linear Filter (cont.)



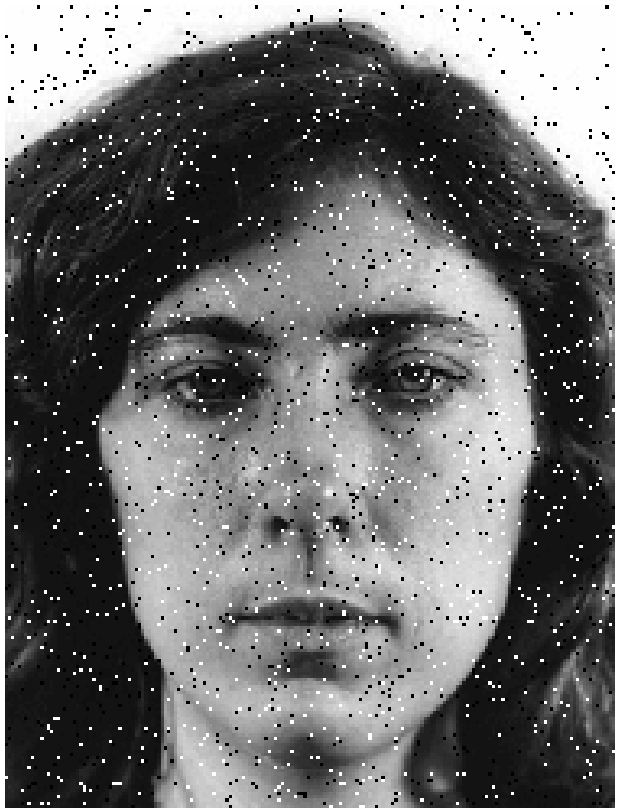
a b c

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.)

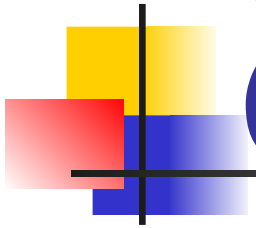
(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Smoothing is useful when we want to eliminate fine detail, particularly if that detail is an artifact of the image and not part of the inherent information. The larger the smoothing neighborhood, the more blurred the result becomes.

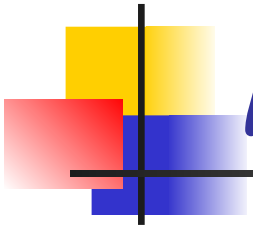
Smoothing can be useful in eliminating "salt and pepper" noise.



Order-Statistics Filters (Nonlinear Filters)

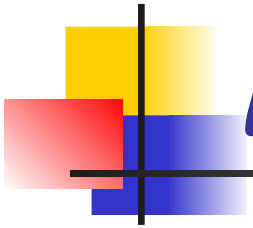


- the response is based on ordering (ranking) the pixels contained in the image area encompassed by the filter
- example
  - median filter : $R = \text{median}\{z_k \mid k = 1,2,\ldots,n \times n\}$
  - max filter : $R = \max\{z_k \mid k = 1,2,\ldots,n \times n\}$
  - min filter : $R = \min\{z_k \mid k = 1,2,\ldots,n \times n\}$
- note: n x nis the size of the mask



Median Filters

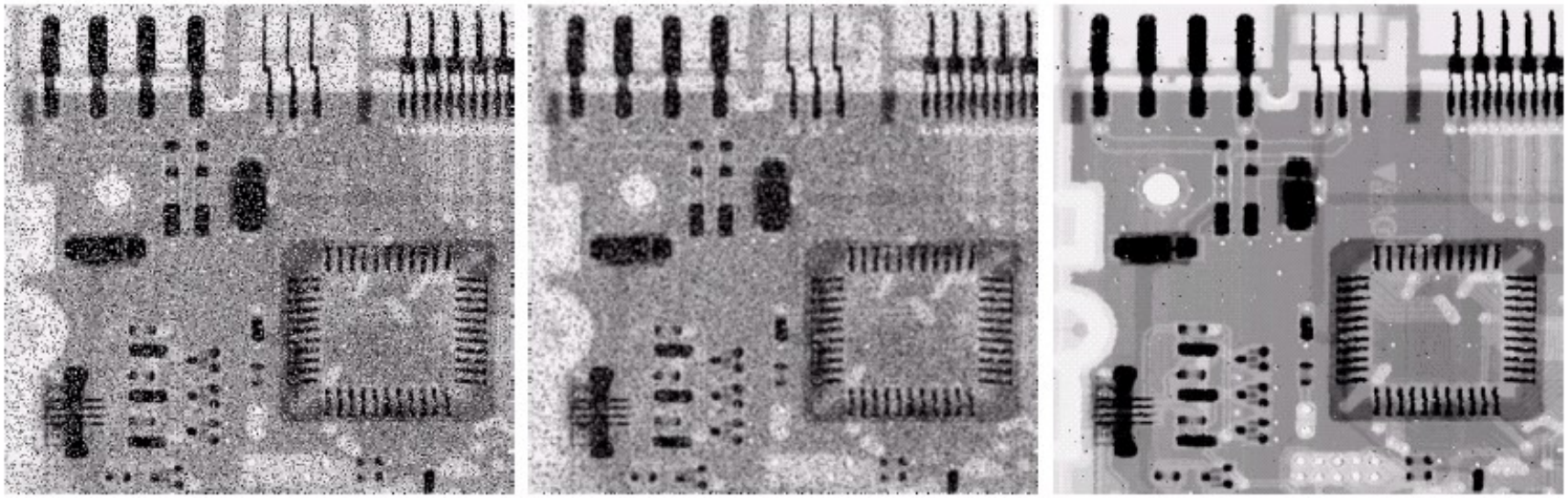
- replaces the value of a pixel by the median of the gray levels in the neighborhood of that pixel (the original value of the pixel is included in the computation of the median)
- quite popular because for certain types of random noise (**impulse noise salt and pepper noise**) , they **provide excellent noise-reduction capabilities**, with considering **less blurring than linear smoothing filters of similar size.**



Median Filters

- forces the points with distinct gray levels to be more like their neighbors.
- isolated clusters of pixels that are light or dark with respect to their neighbors, and whose area is less than $n^2/2$ (one-half the filter area), are eliminated by an n x n median filter.
- eliminated = forced to have the value equal the median intensity of the neighbors.
- larger clusters are affected considerably less

Example : Median Filters

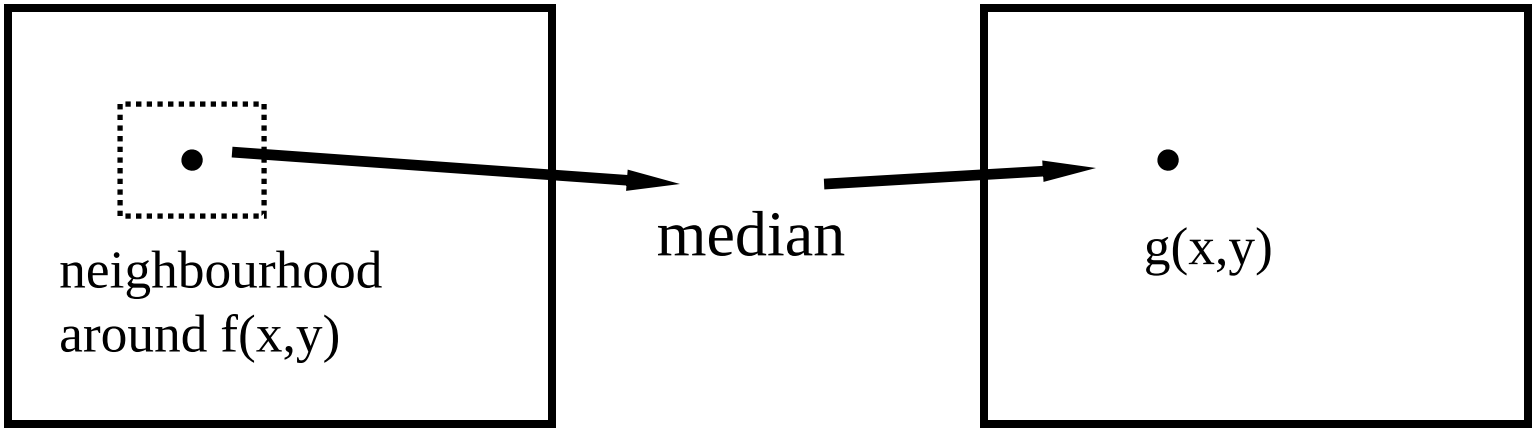


a b c

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.)

Median Filtering

Median filtering is an alternative form of area process transformation where the central pixel value in a neighbourhood is replaced by the median value of those pixels.



input image **f**

output image **g**

The median of a list of numbers is the value such that half the numbers are less than this value and the other half are greater.

For example, the list 1 1 1 1 3 3 3 3 5 has median 3.

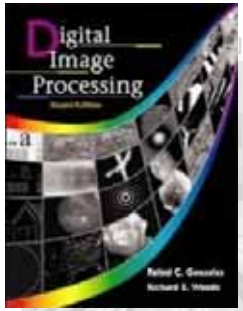
Median filtering is particularly good at removing "salt and pepper" noise whilst preserving edge structure.

original

added noise

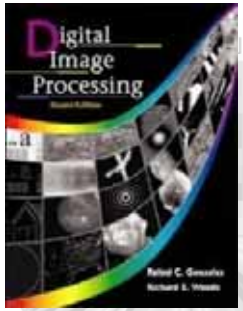
average

median



3.6.2 Smoothing Spatial Non-linear Filters

- Median Filter
  - The response is based on ordering (ranking) the pixels contained in the image area encompassed by the filter, and then replacing the value of the center pixel with the value determined by the ranking result.
- For certain noise, such as ***impulse noise*** or ***salt-and-pepper*** noise, median filter is effective.
- The median, ξ, of a set of values is such that half of the values in the set are less than or equal to ξ, and half are greater than or equal to ξ.



3.6.2 Smoothing Spatial Non-linear Filters

- ***Median filtering***:

1. Sort the values of the target pixel and its neighboring pixels to find the median.
2. Replace the target pixel with the median

For example, the median is the 5^{th} largest value of the 3x3 neighborhood and 15th largest value in the 5x5 neighborhood.

Rank / Order / Order Statistics Filters

- Known as Rank filters, Order filters OR Order Statistics filters
- Operate on a neighborhood around a reference pixel by ordering (ranking) the pixel values and then performing an operation on those ordered values to obtain the new value for the reference pixel
- They perform very well in the presence of salt and pepper noise but are more computationally expensive as compared to mean filters

Rank / Order Statistics Filters: Median filter

- Output is based on ordering (ranking) the pixels in a subimage
- Replace the value of a pixel by the median of the gray levels in the neighborhood of that pixel (a specified mask)
- Excellent for removing both bipolar and unipolar impulse noise

$$\hat{f}(x,y) = \underset{(s,t)\in S_{xy}}{median}\{g(s,t)\}$$

Rank / Order Statistics Filters: Median filter

- *Most popular and useful of the rank filters.*
- *It works by selecting the middle pixel value from the ordered set of values within the $m \times n$ neighborhood (W) around the reference pixel.*
  - If *mn* is an even number, the arithmetic average of the two values closest to the middle of the ordered set is used instead.
- *Many variants, extensions, and optimized implementations in the literature.*