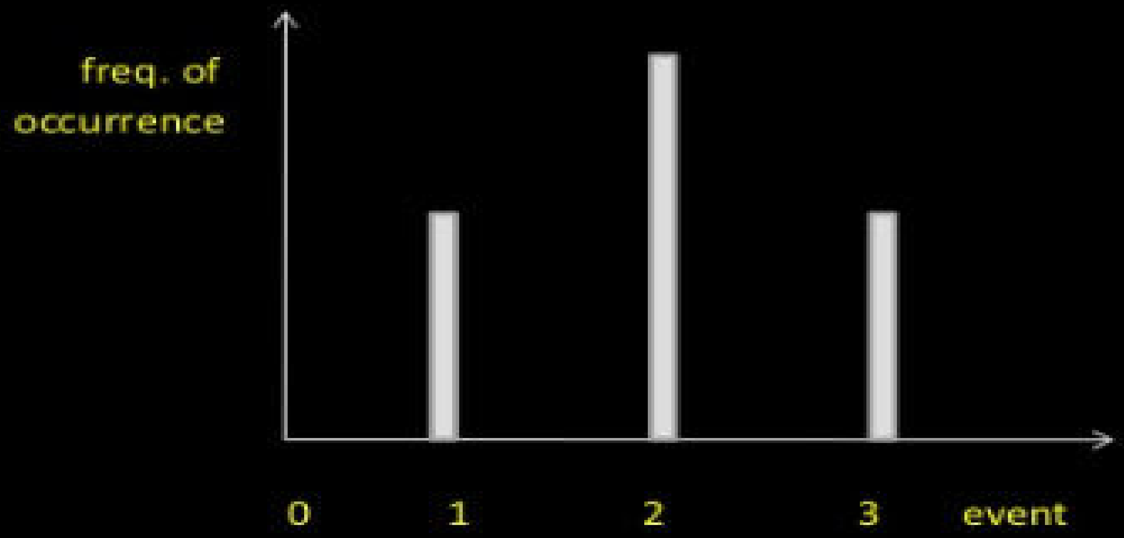
Histogram Processing

Histogram Processing

Histogram:

It is a plot of frequency of occurrence of an event.



Histogram Processing

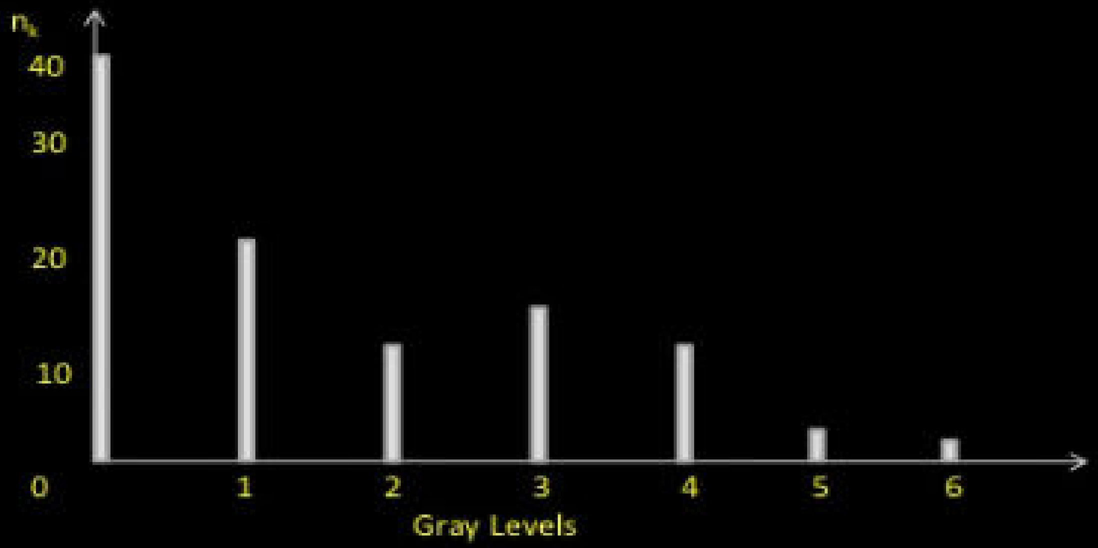
- Histogram of images provide a global description of their appearance.
- Enormous information is obtained.
- It is a spatial domain technique.
- Histogram of an image represents relative frequency of occurrence of various gray levels.
- Histogram can be plotted in two ways:

Histogram Processing

- First Method:

- X-axis has gray levels & Y-axis has No. of pixels in each gray levels.

Gray Level	No. of Pixels (n_k)
0	40
1	20
2	10
3	15
4	10
5	3
6	2



Consider a 5x5 image with integer intensities in the range between zero and seven:

Image

$$\begin{matrix} 0 & 7 & 3 & 2 & 3 \\ 0 & 0 & 0 & 6 & 7 \\ 7 & 7 & 2 & 2 & 0 \\ 1 & 1 & 0 & 4 & 1 \\ 0 & 0 & 7 & 4 & 1 \end{matrix}$$

Image matrix

Gray scale

Consider a 5x5 image with integer intensities in the range between one and Seven

Image

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

Image matrix

Grey scale

Number of pixel with intensity value 0 [h(r0)] = 8

Image

0	7	3	2	3
0	0	0	6	7
7	7	2	2	0
1	**1**	0	4	**1**
0	0	7	4	**1**

Image matrix

Grey scale

Number of pixel with intensity value 0 [h(r0)] = 8

Similarly for **1** h(r1) = 4

Image

$$\begin{matrix} 0 & 7 & 3 & 2 & 3 \\ 0 & 0 & 0 & 6 & 7 \\ 7 & 7 & 2 & 2 & 0 \\ 1 & 1 & 0 & 4 & 1 \\ 0 & 0 & 7 & 4 & 1 \end{matrix}$$

Image matrix

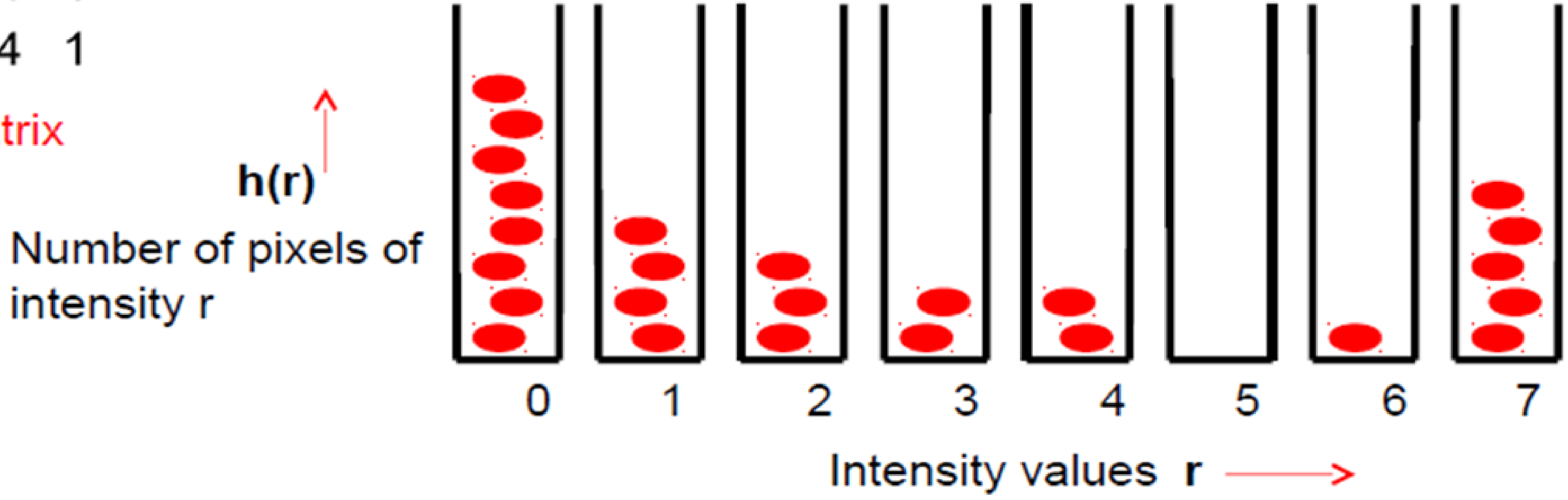
Similarly

INTENSITY r	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
NUMBER of pixels of r h(r)	h(r0)=**8**	h(r1)=**4**	h(r2)=**3**	h(r3)=**2**	h(r4)=**2**	h(r5)=**0**	h(r6)=**1**	h(r7)=**5**

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

Image matrix

r	0	1	2	3	4	5	6	7
h(r)	8	4	3	2	2	0	1	5



HISTOGRAM

Histogram Processing

- **Second Method**:
- X-axis has gray levels & Y-axis has probability of occurrence of gray levels.

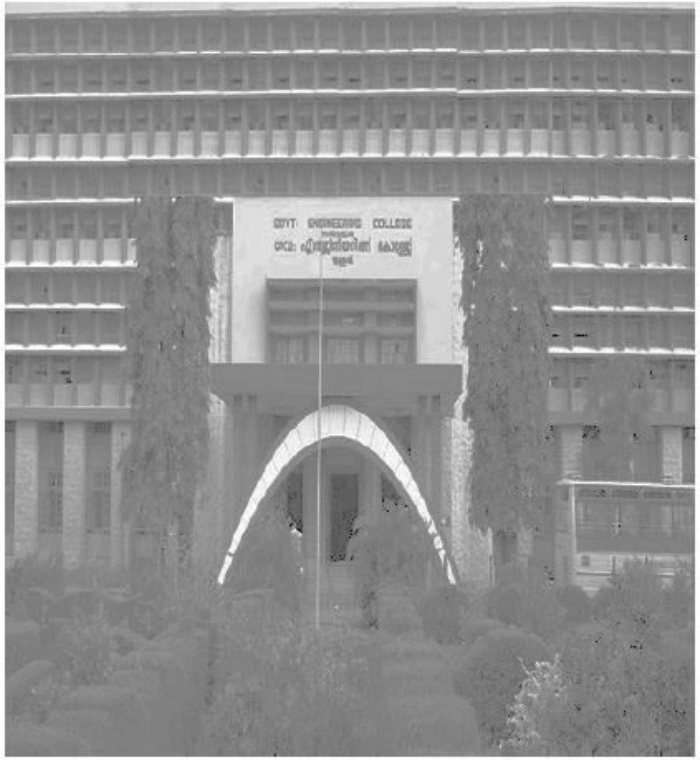
$$P(\mu_k) = n_k / n; \quad \text{where, } \mu_k - \text{gray level}$$

n_k – no. of pixels in k^{th} gray level

n – total number of pixels in an image

Gray Level	No. of Pixels (n_k)	$P(\mu_k)$
0	40	0.4
1	20	0.2
2	10	0.1
3	15	0.15
4	10	0.1
5	3	0.03
6	2	0.02
	n = 100	1

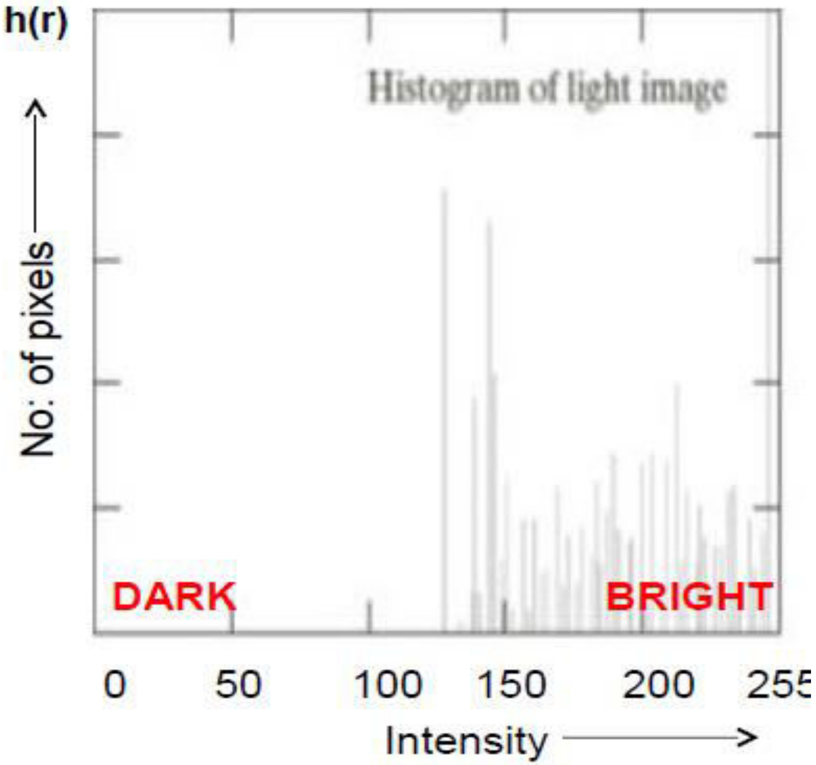
SAMPLE IMAGES AND ITS HISTOGRAM



Bright image
Intensity range 0 - 255

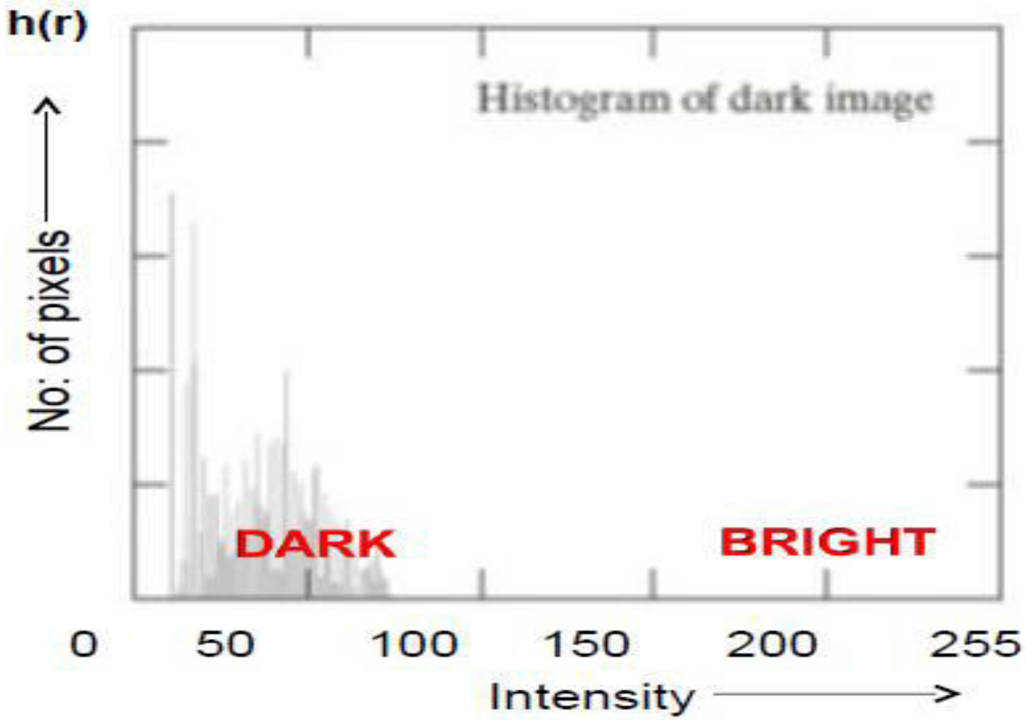
SAMPLE IMAGES AND ITS HISTOGRAM

Bright image
Intensity range 0 - 255



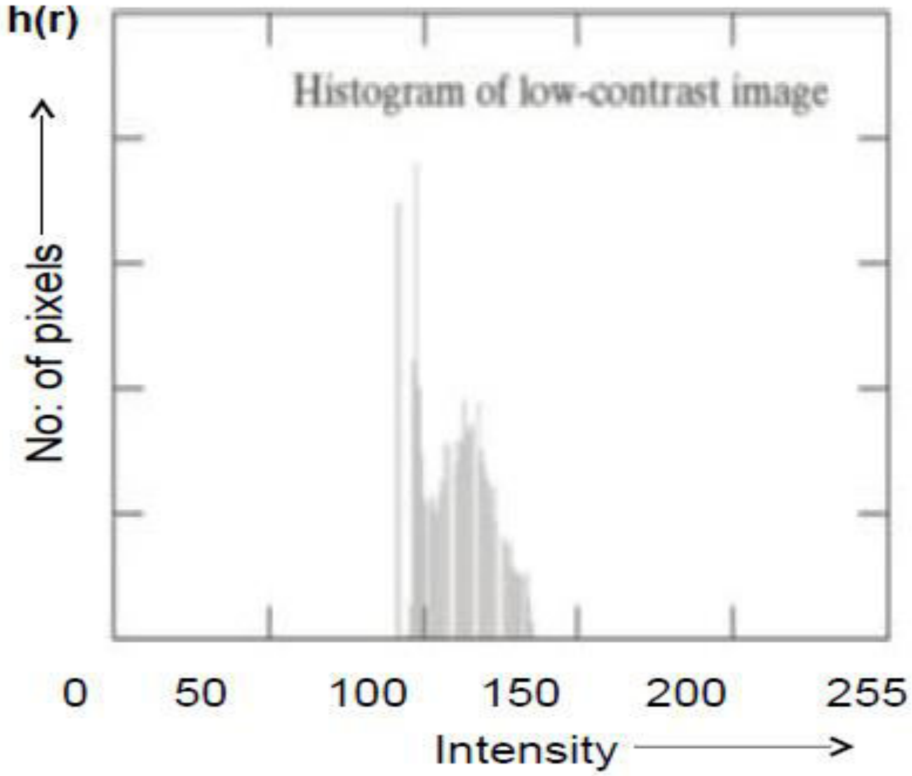
SAMPLE IMAGES AND ITS HISTOGRAM

Dark image
Intensity range 0 - 255



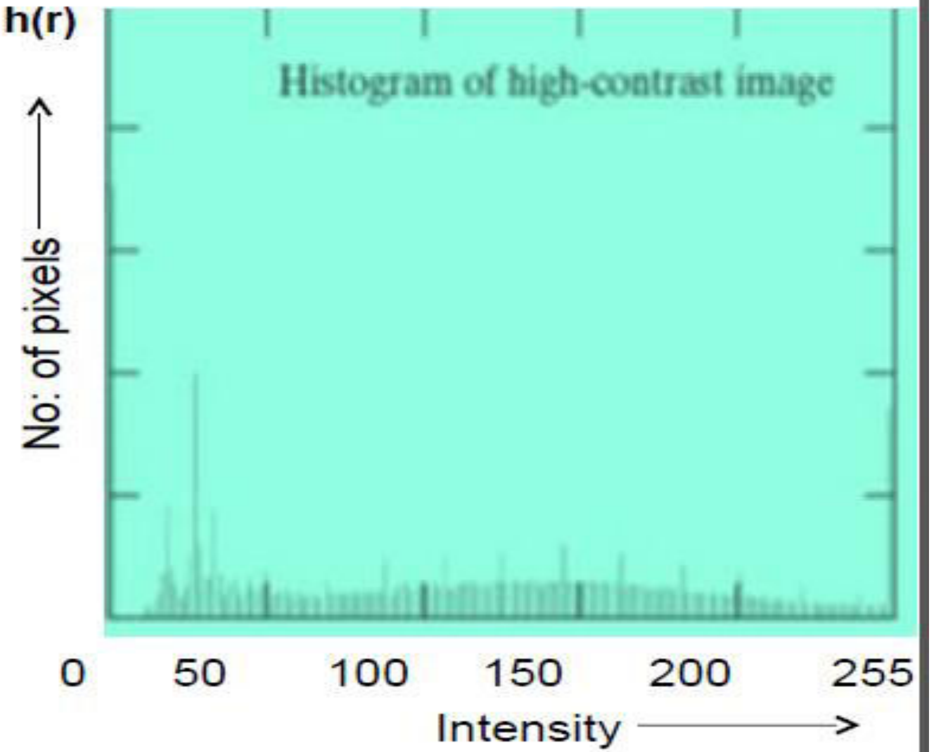
SAMPLE IMAGES AND ITS HISTOGRAM

Light image
Intensity range 0 - 255



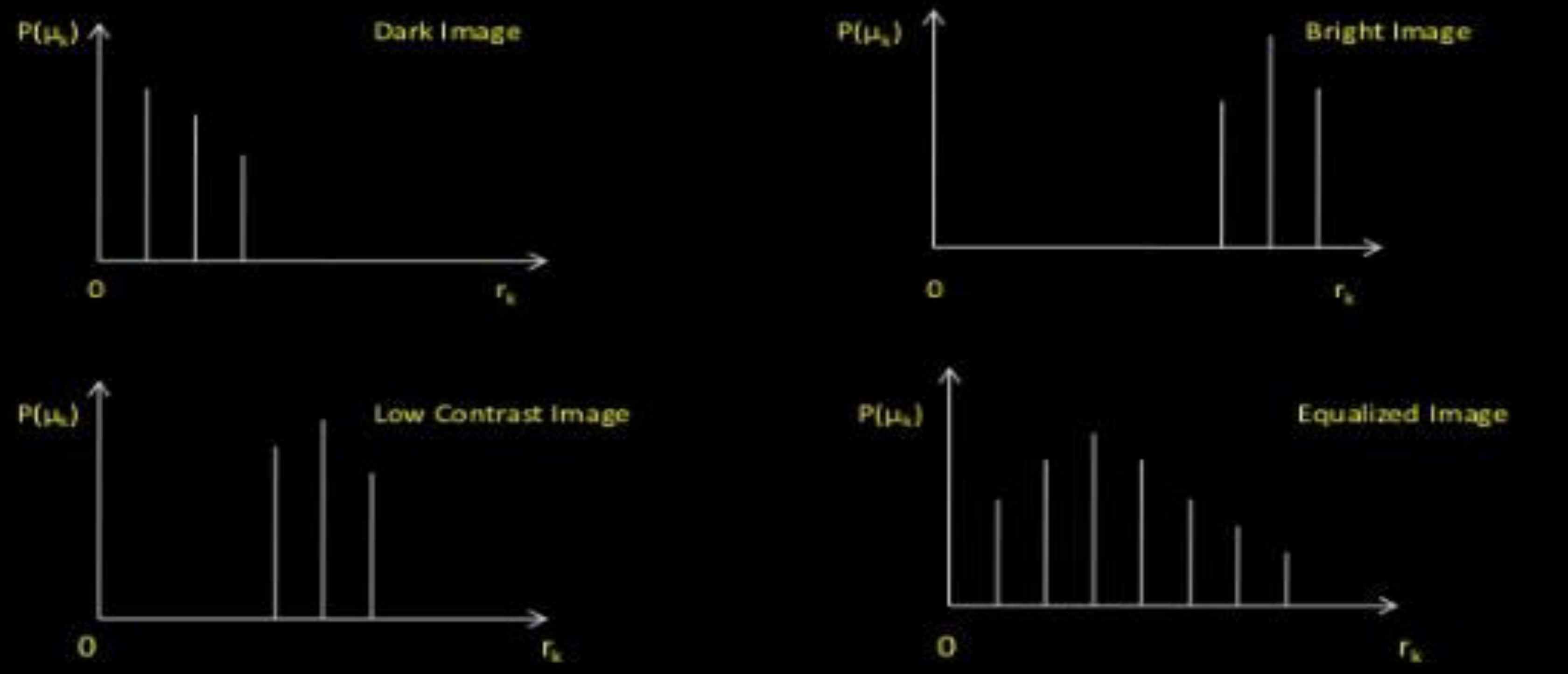
SAMPLE IMAGES AND ITS HISTOGRAM

High contrast image
Intensity range 0 - 255



Histogram Processing

- Advantage of 2nd method: Maximum value plotted will always be 1.
- White – 1, Black – 0.
- Great deal of information can be obtained just by looking at histogram.
- Types of Histograms:



Histogram Processing

- The last graph represent the best image.
- It is a high contrast image.
- Our aim would be to transform the first 3 histograms into the 4^{th} type.
- In other words we try to increase the dynamic range of the image.

Histogram Stretching

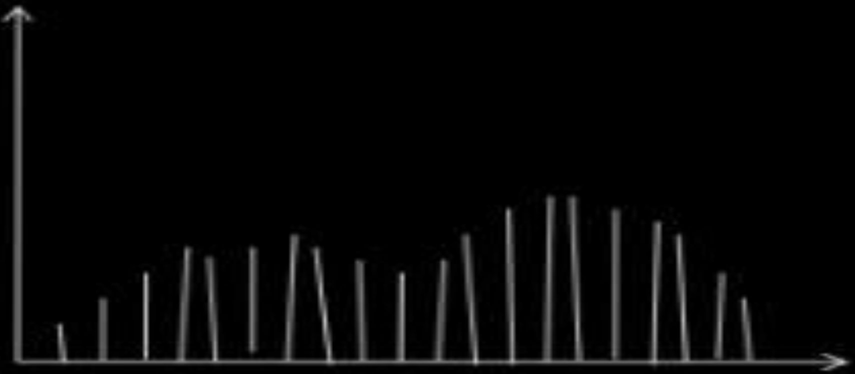
1) <u>Linear stretching:</u>

- Here, we don't alter the basic shape.
- We use basic equation of a straight line having a slope.

$$(S_{max} - S_{min})/(r_{max} - r_{min})$$

Where, Smax – max gray level of output image
Smin – min gray level of output image
rmax – max gray level of input image
rmin – min gray level of input image

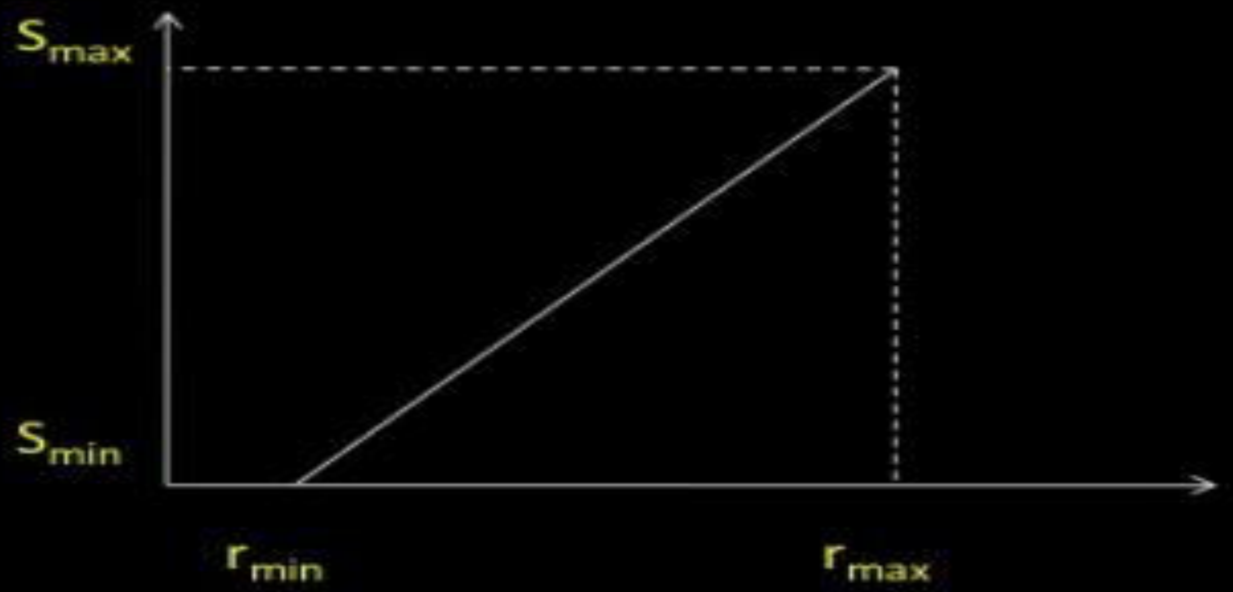
0 min max



0 min max

Histogram Processing

$$S = T(r) = ((S_{max} - S_{min})/(r_{max} - r_{min}))(r - r_{min}) + Smin$$



Histogram Processing

Ex. 1) Perform Histogram Stretching so that the new image has a dynamic range of 0 to 7 [0, 7].

Gray Levels	0	1	2	3	4	5	6	7
No. of Pixels	0	0	50	60	50	20	10	0

Histogram Processing

Ex. 1) Perform Histogram Stretching so that the new image has a dynamic range of 0 to 7 [0, 7].

Gray Levels	0	1	2	3	4	5	6	7
No. of Pixels	0	0	50	60	50	20	10	0

Sol^n:- $r_{min} = 2$; $r_{max} = 6$; $s_{min} = 0$; $s_{max} = 7$;

slope = $((s_{max} - s_{min})/(r_{max} - r_{min})) = ((7 - 0)/(6 - 2)) = 7 / 4 = 1.75$.

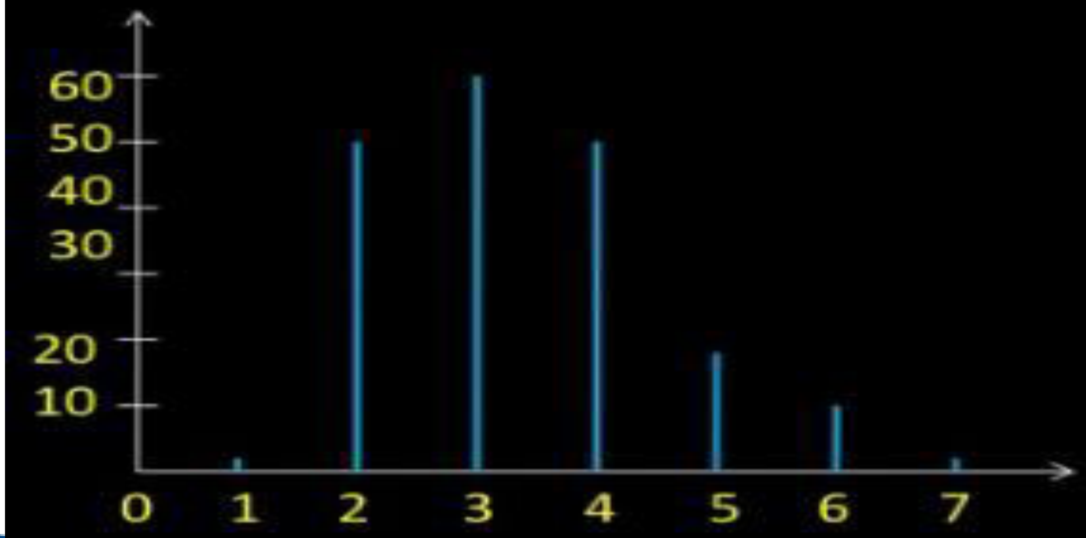
$S = (7 / 4)(r - 2) + 0;$
$S = (7 / 4)(r - 2)$

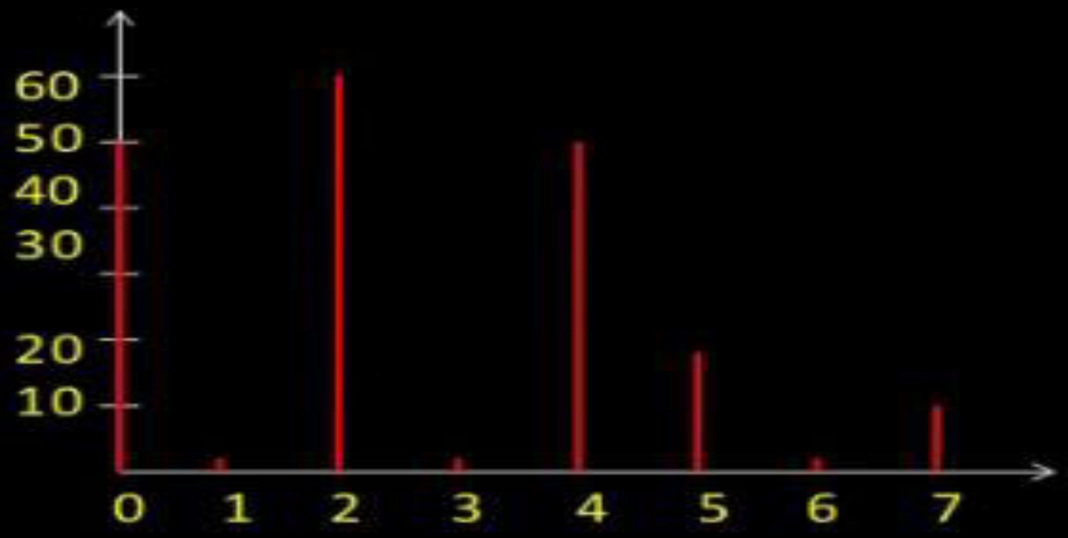
r	(7 / 4)(r - 2) = S
2	0 = 0
3	7/4 = 1.75 = 2
4	7/2 = 3.5 = 4
5	21/4 = 5.25 = 5
6	7 = 7

Histogram Processing

Ex. 1) Perform Histogram Stretching so that the new image has a dynamic range of 0 to 7 [0, 7].

Gray Levels	0	1	2	3	4	5	6	7
No. of Pixels	50	0	60	0	50	20	0	10





Perform histogram Stretching for the new image dynamically ranges form 0 to12

Pixel value	0	1	2	3	4	5	6	7	8	9	10	11	12
No. of occurence	0	0	0	0	20	50	20	40	60	10	0	0	0

Histogram Equalization

Histogram Equalization

- What is the histogram equalization?

- The histogram equalization is an approach to enhance a given image. The approach is to design a transformation *T(.)* such that the gray values in the output is uniformly distributed in [0, 1].

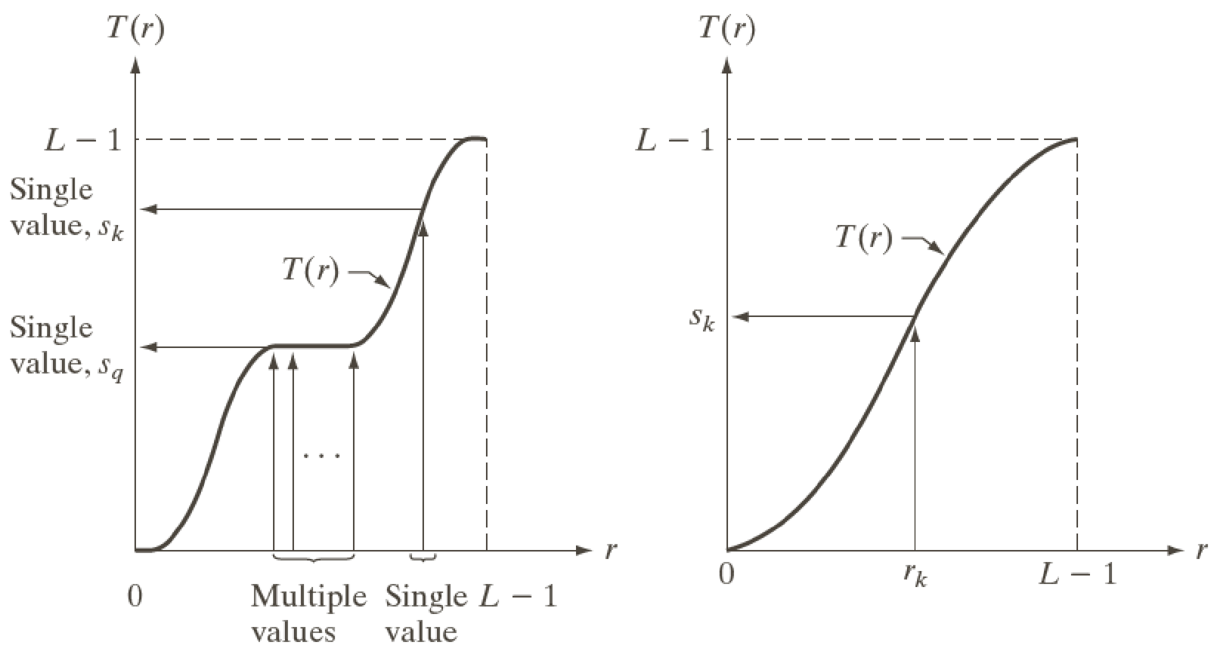
- Let us assume for the moment that the input image to be enhanced has continuous gray values, with $r = 0$ representing black and $r = 1$ representing white.

- We need to design a gray value transformation $s = T(r)$, based on the histogram of the input image, which will enhance the image.

- As before, we assume that:

(1) *T(r)* is a monotonically increasing function for $0 \le r \le 1$(preserves order from black to white).

(2) *T(r)* maps [0,1] into [0,1] (preserves the range of allowed Gray values).



- Let us denote the inverse transformation by $r = T^{-1}(s)$. We assume that the inverse transformation also satisfies the above two conditions.

- We consider the gray values in the input image and output image as random variables in the interval [0, 1].

- Let $p_{in}(r)$ and $p_{out}(s)$ denote the probability density of the Gray values in the input and output images.

$$p_s(s)ds = p_r(r)dr$$

- If $p_{in}(r)$ and T(r) are known, and $r = T^{-1}(s)$ satisfies condition 1, we can write (result from probability theory):

$$p_{out}(s) = \left[p_{in}(r) \frac{dr}{ds} \right]_{r=T^{-1}(s)}$$

- One way to enhance the image is to design a transformation *T(.)* such that the gray values in the output is uniformly distributed in [0, 1], i.e. $p_{out}(s) = 1, \quad 0 \leq s \leq 1$

- In terms of histograms, the output image will have all gray values in "equal proportion" .

- This technique is called **histogram equalization**.

Next we derive the gray values in the output is uniformly distributed in [0, 1].

•Consider the transformation

$$s = T(r) = \int_0^r p_{in}(w)dw, \qquad 0 \le r \le 1$$

• Note that this is the cumulative distribution function (CDF) of $p_{in}(r)$ and satisfies the previous two conditions.

• From the previous equation and using the fundamental theorem of calculus,

$$\frac{ds}{dr} = p_{in}(r)$$

- Therefore, the output histogram is given by

$$p_{out}(s) = \left[p_{in}(r) \cdot \frac{1}{p_{in}(r)} \right]_{r=T^{-1}(s)} = [1]_{r=T^{-1}(s)} = 1, \qquad 0 \leq s \leq 1$$

- The output probability density function is uniform, regardless of the input.

- Thus, using a transformation function equal to the CDF of input gray values r, we can obtain an image with uniform gray values.
- This usually results in an enhanced image, with an increase in the dynamic range of pixel values.

PRESIDENCY UNIVERSITY
GAIN MORE KNOWLEDGE REACH GREATER HEIGHTS
Private University Estd. in Karnataka State by Act No. 41 of 2013
PRESIDENCY GROUP OVER 40 YEARS OF ACADEMIC WISDOM

How to implement histogram equalization?

Step 1:For images with discrete gray values, compute:

$$p_{in}(r_k) = \frac{n_k}{n} \qquad 0 \le r_k \le 1 \qquad 0 \le k \le L-1$$

L: Total number of gray levels

n_k: Number of pixels with gray value r_k

n: Total number of pixels in the image

Step 2: Based on CDF, compute the discrete version of the previous transformation :

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_{in}(r_j) \qquad 0 \le k \le L-1$$

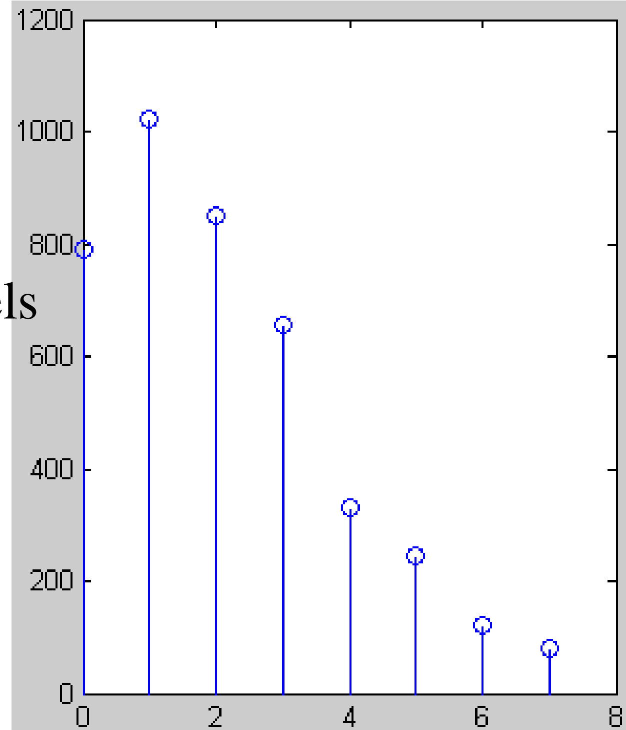
Example:

- Consider an 8-level 64 x 64 image with gray values (0, 1, …, 7). The normalized gray values are (0, 1/7, 2/7, …, 1). The normalized histogram is given below:

k	r_k	n_k	$p(r_k) = n_k/n$
0	0	790	0.19
1	1/7	1023	0.25
2	2/7	850	0.21
3	3/7	656	0.16
4	4/7	329	0.08
5	5/7	245	0.06
6	6/7	122	0.03
7	1	81	0.02

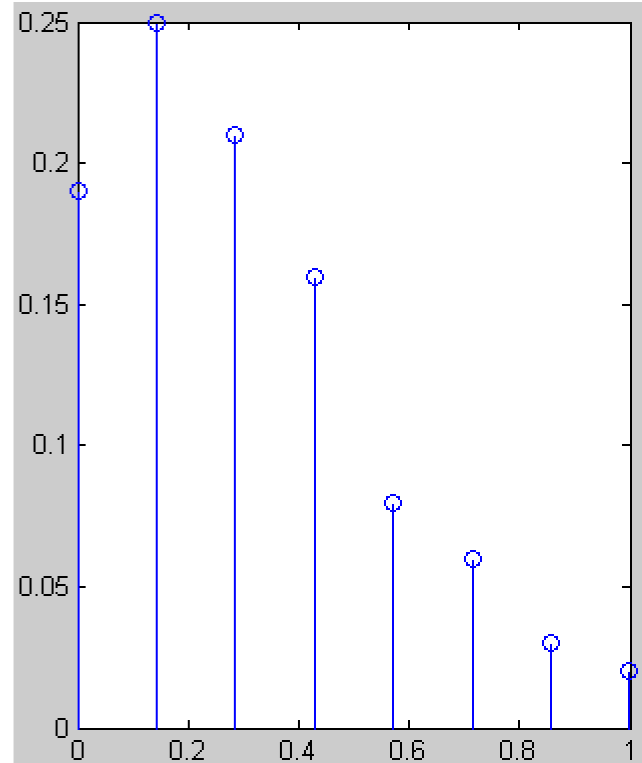
Note: The gray values in output are also (0, 1/7, 2/7, …, 1).

pixels



Gray value

Fraction
of # pixels



Normalized gray value

```
>> clear
>> h=[790 1023 850 656 329 245 122 81];
>> stem(0:7,h)
```

```
>> clear
>> h=[0.19 0.25 0.21 0.16 0.08 0.06 0.03 0.02];
>> stem(0:0.142857:1,h)
```

- Applying the transformation, $s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_{in}(r_j)$ we have

$$s_0 = T(r_0) = 7\sum_{j=0}^{0} p_r(r_j) = 7 \times 0.19 = 1.33 \quad \rightarrow 1$$

$$s_1 = T(r_1) = 7\sum_{j=0}^{1} p_r(r_j) = 7 \times (0.19 + 0.25) = 3.08 \quad \rightarrow 3$$

$$s_2 = 4.55 \quad \rightarrow 5 \qquad s_3 = 5.67 \quad \rightarrow 6$$

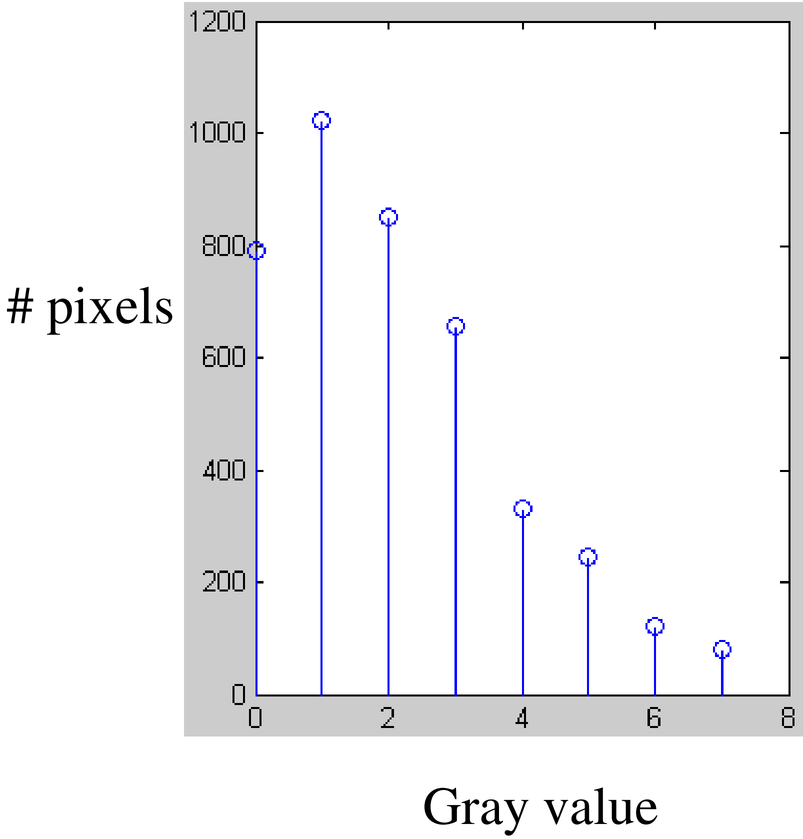
$$s_4 = 6.23 \quad \rightarrow 6 \qquad s_5 = 6.65 \quad \rightarrow 7$$

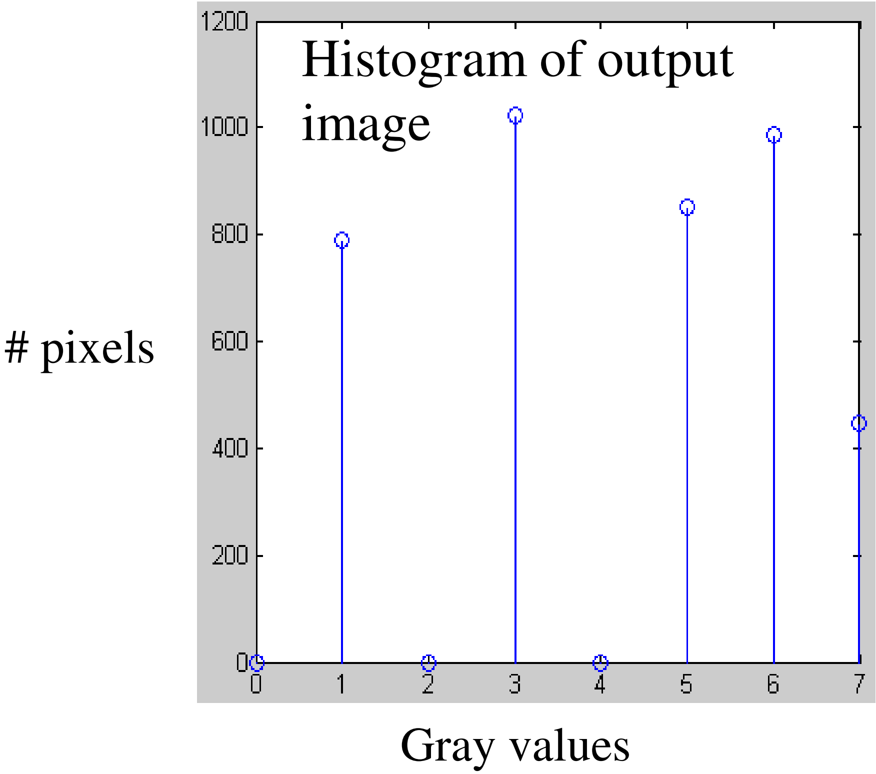
$$s_6 = 6.86 \quad \rightarrow 7 \qquad s_7 = 7.00 \quad \rightarrow 7$$

• Notice that there are only five distinct gray levels --- (1/7, 3/7, 5/7, 6/7, 1) in the output image. We will relabel them as (s_0, s_1, …, s_4).

• With this transformation, the output image will have histogram

k	s_k	n_k	$p(s_k) = n_k/n$
0	1/7	790	0.19
1	3/7	1023	0.25
2	5/7	850	0.21
3	6/7	985	0.24
4	1	448	0.11



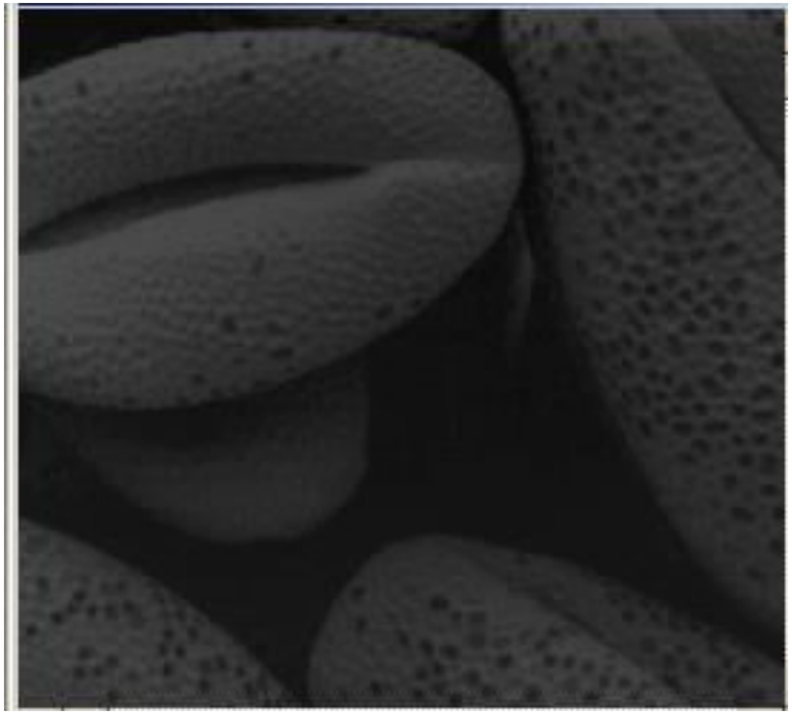


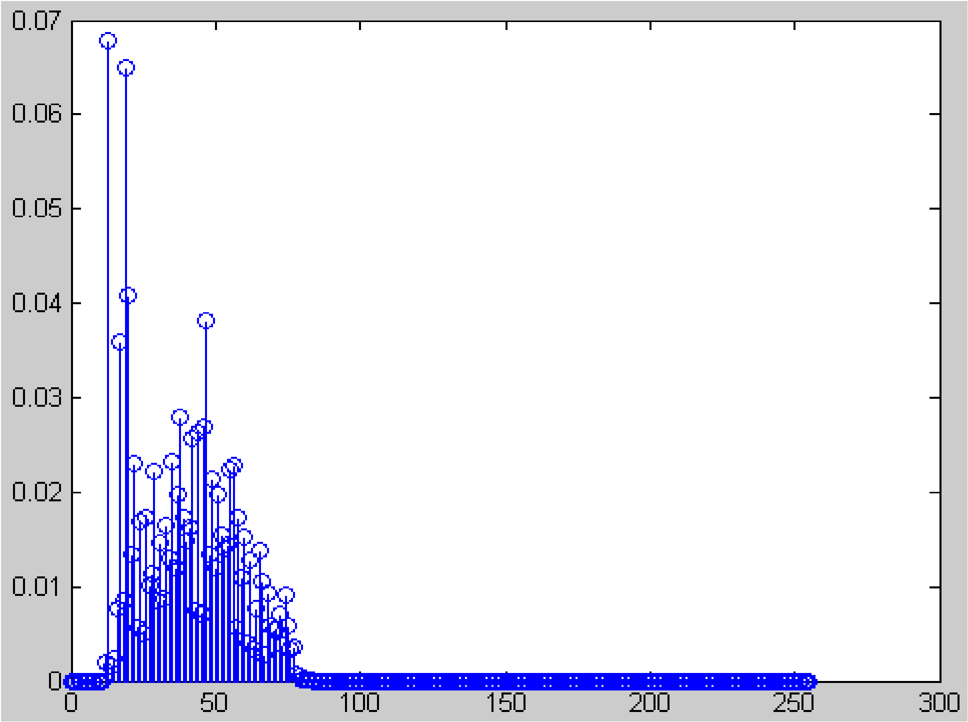

```
>> clear
>> h=[0 790 0 1023 0 850 985 448];
>> stem(0:7,h)
```

• Note that the histogram of output image is only approximately, and not exactly, uniform. This should not be surprising, since there is no result that claims uniformity in the **discrete** case.

Example

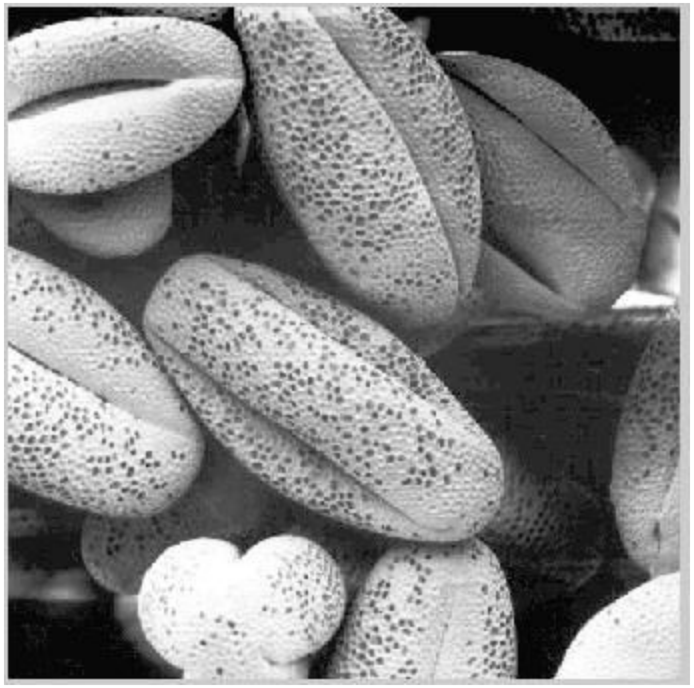
Original image and its histogram

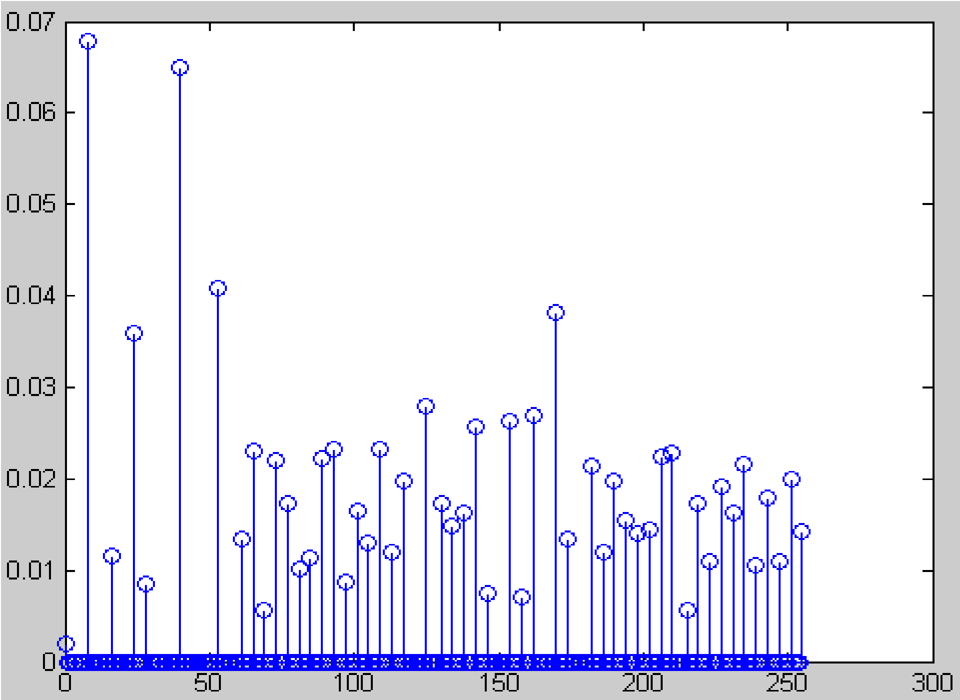





```
>> clear;
[ix,map]=imread('Fig3_15a.jpg');
imshow(ix)
figure;
ix=double(ix);
h=histogram(ix);
stem(0:255,h);
```

Histogram equalized image and its histogram






```
>> clear
>> [ix,map]=imread('Fig3_15a.jpg');
>> imshow(ix);
>> iy=histeq(ix);
>> figure
>> imshow(iy);
>> iy=double(iy);
>> hy=histogram(iy);
>> figure
>> stem(0:255,hy);
```

Module-2
Digital Image Processing

Image Transformation

Arithmetic and Logic Operations for Image processing

Arithmetic and Logic Operations

- Arithmetic and logic operations are often applied a preprocessing steps in image analysis in order to combine images in various way.
- Addition, subtraction, division and multiplication comprise the arithmetic operation, while AND , OR, and NOT make up the logic operations.
- These operation performed on two image , except the NOT logic operation which require only one image, and are done on a pixel by pixel basis.

Image Arithmetic

- For input images f_1 and f_2 and some function Op:

  $$g(x, y) = \text{Op}\left(f_1(x, y), f_2(x, y)\right)$$

  The operator is applied pairwise to each pixel in the images.
- Pseudocode:

  ```
  for all pixel positions x, y:
    out[x,y] = func( image1[x,y] , image2[x,y]
  )
  ```
- Possibilities: addition, subtraction, and, or, ...

Arithmetic operations

- Addition: $g(x,y) = f_1(x,y) + f_2(x,y)$
- Subtraction: $g(x,y) = f_1(x,y) - f_2(x,y)$
- Multiplication: $g(x,y) = f_1(x,y) \cdot f_2(x,y)$
- Division: $g(x,y) = f_1(x,y) / f_2(x,y)$

Addition/Blending

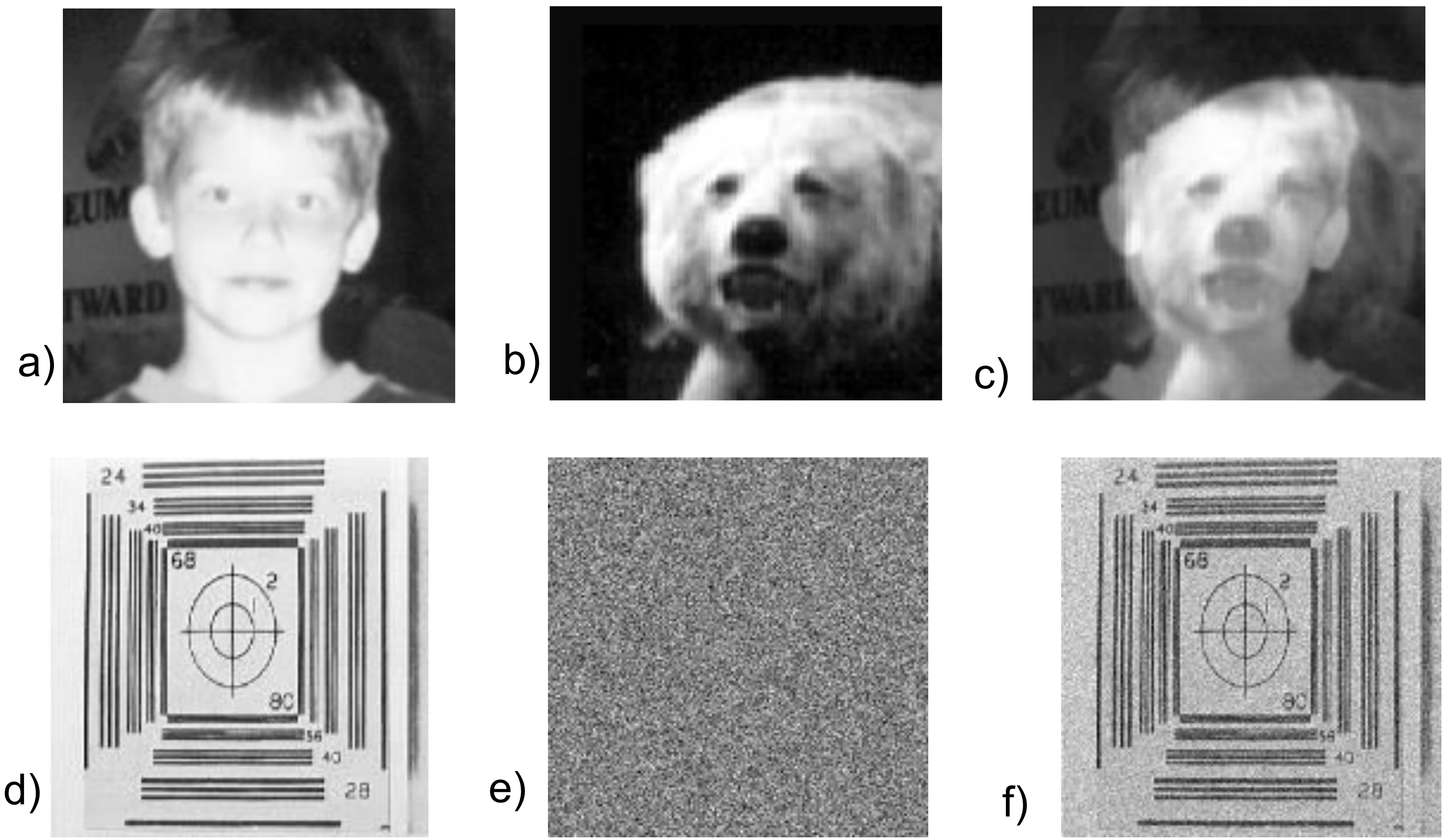
Used to create double-exposures or composites:

$$g(x, y) = f_1(x, y) + f_2(x, y)$$

Can also do a weighted *blend*:

$$g(x, y) = \alpha_1 f_1(x, y) + \alpha_2 f_2(x, y)$$

Figure 3.2-6 Image Addition Examples. This example shows one step in the *image morphing* process where an increasing percentage of the second image is slowly added to the first, and a geometric transformation is usually required to align the images. a) first original, b) second original, c) addition of images (a) and (b). This example shows adding noise to an image which is often useful for developing image restoration models. d) original image, e) Gaussian noise, variance = 400, mean = 0, f) addition of images (d) and (e).



Arithmetic operations

- Addition: $g(x, y) = f_1(x, y) + f_2(x, y)$
- Subtraction: $g(x, y) = f_1(x, y) - f_2(x, y)$
- Multiplication: $g(x, y) = f_1(x, y) \cdot f_2(x, y)$
- Division: $g(x, y) = f_1(x, y) / f_2(x, y)$

Subtraction

- Useful for finding changes between two images

$$g(x, y) = f_1(x, y) - f_2(x, y)$$

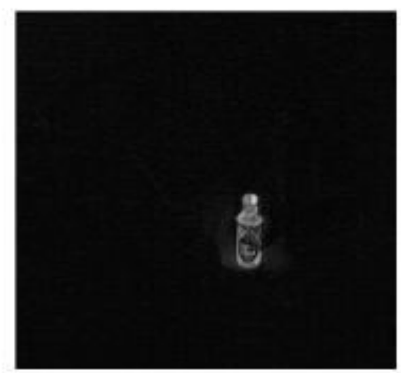
- Sometimes more useful to use *absolute difference*

$$g(x, y) = |f_1(x, y) - f_2(x, y)|$$

- What's changed?

-

=



Subtraction

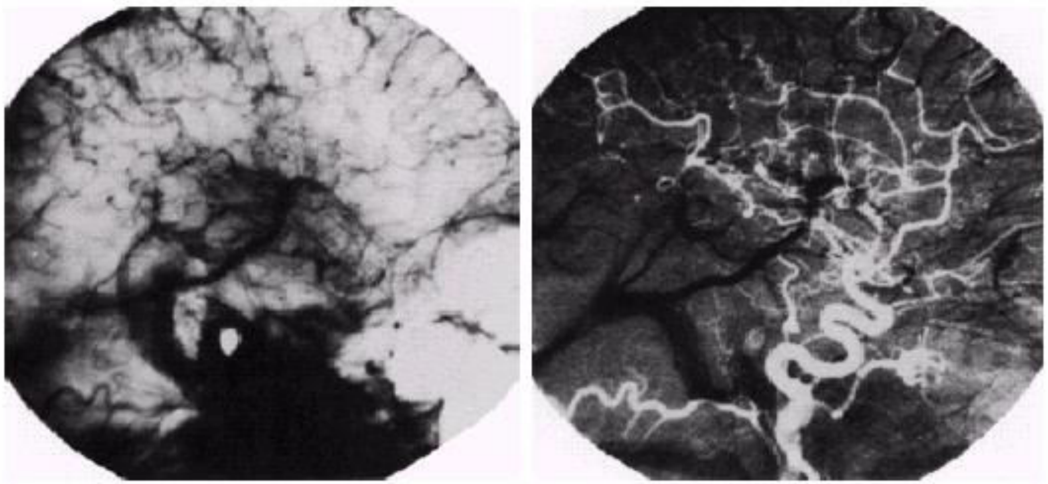
- Subtraction of two image is often used to detect motion.
- Consider the case where nothing has changed in a scene; the image resulting from subtraction of two sequential image is filled with zeros - a black image.
- If something has moved in the scene, subtraction produce a nonzero result at the location of movement.

Subtraction

- Medical imaging often uses this type of operation to allow the doctor to more readily see changes which are helpful in the diagnosis.
- The technique is also used in law enforcement and military applications; for example, to find an individual in a crowd or to detect changes in a military installation.

Digital Subtraction Angiography

1. Take an x-ray
2. Inject patient with a radio-opaque dye (and tell them not to move!)
3. Take another x-ray
4. Subtract the two



a b

FIGURE 3.29 Enhancement b image subtracti (a) Mask image (b) An image (taken after injection of a contrast mediu into the bloodstream) w mask subtracte out.

Arithmetic operations

- Addition: $g(x,y) = f_1(x,y) + f_2(x,y)$
- Subtraction: $g(x,y) = f_1(x,y) - f_2(x,y)$
- Multiplication: $g(x,y) = f_1(x,y) \cdot f_2(x,y)$
- Division: $g(x,y) = f_1(x,y)/f_2(x,y)$

Arithmetic operations

- Addition: $g(x, y) = f_1(x, y) + f_2(x, y)$
- Subtraction: $g(x, y) = f_1(x, y) - f_2(x, y)$
- Multiplication: $g(x, y) = f_1(x, y) \cdot f_2(x, y)$
- Division: $g(x, y) = f_1(x, y)/f_2(x, y)$

Multiplication n Division

- used to adjust the brightness of an image.
  - is done on a pixel by pixel basis and the options are to multiply or divide an image by a constant value, or by another image.
  - Multiplication of the pixel value by a value greater than one will brighten the image (or division by a value less than 1), and division by a factor greater than one will darken the image (or multiplication by a value les than 1).
- Brightness adjustment by a constant is often used as a preprocessing step in image enhancement and is shown in Figure 3.2.8.

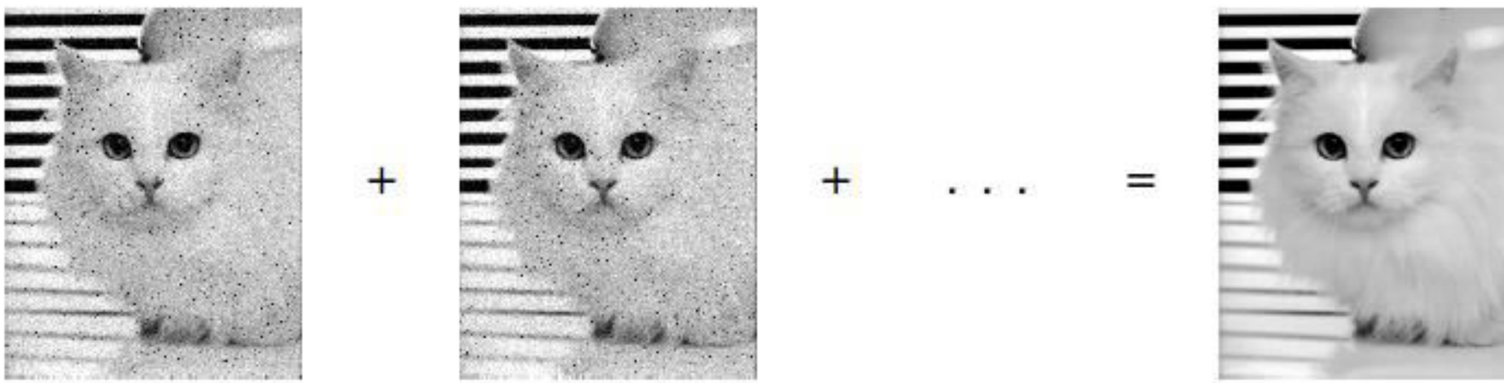
Figure 3.2-8 Image Division. a) original image, b) image divided by a value less than 1 to brighten, c) image divided a value greater than 1 to darken

a) b) c)

Image Averaging

Idea:
Average multiple pictures of the same scene to reduce noise

Similar in principle to acquiring the image for a longer duration.



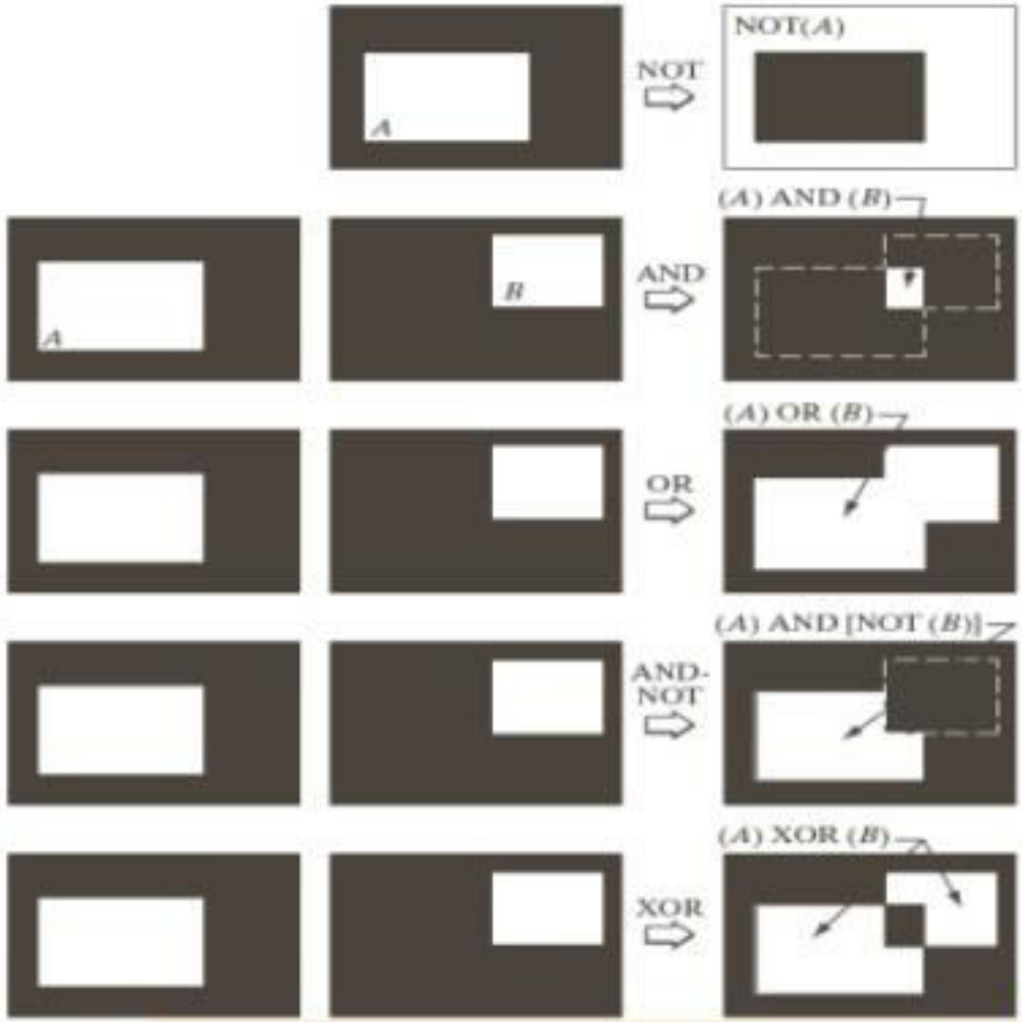
Logic operations

- The logic operations AND, 0R and NOT operate in a bit-wise fashion on pixel data.
- Example
  - performing a logic AND on two images. Two corresponding pixel values are 111_{10} in one image and 88_{10} in the second image. The corresponding bit string are:

$$111_{10} = 01101111_2 \qquad 88 = 01011000_2$$

$$\begin{array}{ll} & 01101111_2 \\ \text{AND} & \underline{01011000}_2 \\ & 01001000_2 \end{array}$$

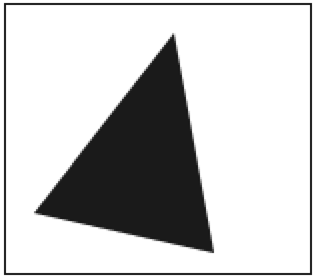
Logical Operations



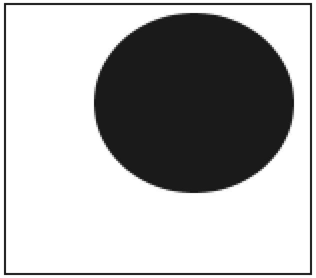
Master Layout 1

Image after logical operation is done

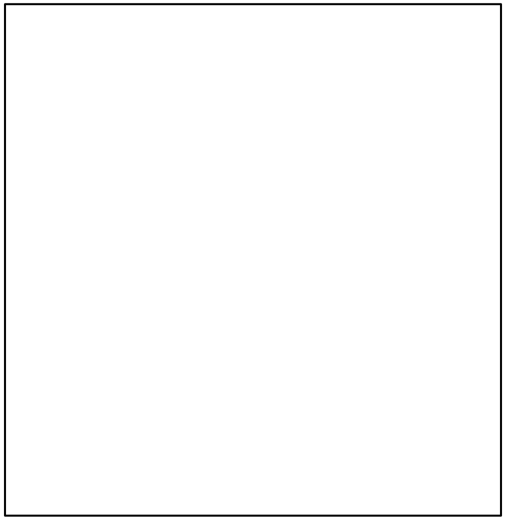
Original Images



IMG_1



IMG_2

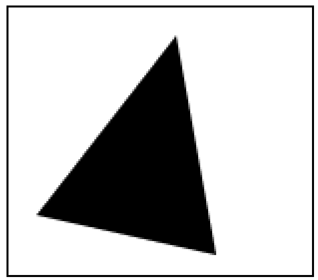


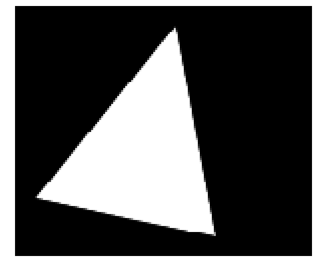
Give a dropdown box to select the operation

- The operations are: NOT, AND, OR, XOR, NOT-AND
- Give Start, Pause, Reset buttons

Step 1:

NOT

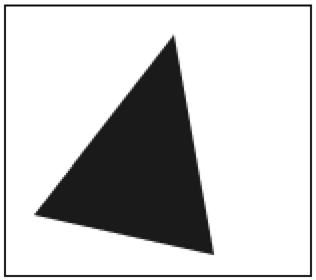




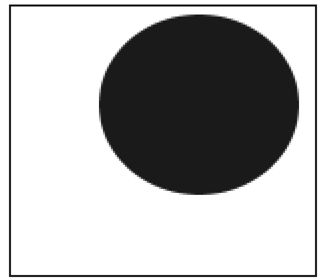
NOT

Step 2:

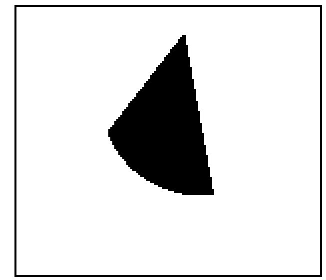
AND



IMG_1



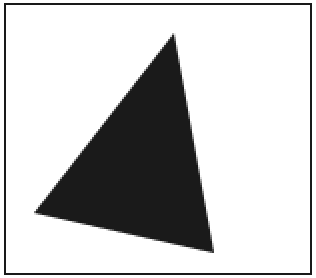
IMG_2



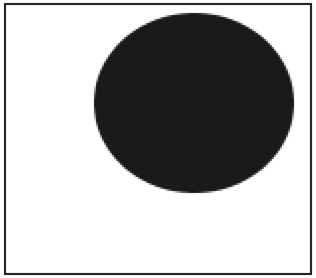
AND

Step 3:

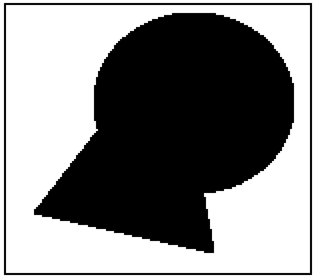
OR



IMG_1



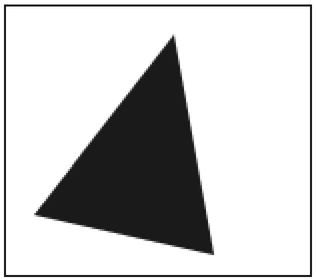
IMG_2



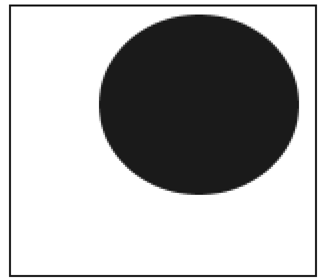
OR

Step 4:

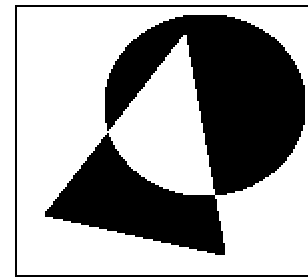
XOR



IMG_1



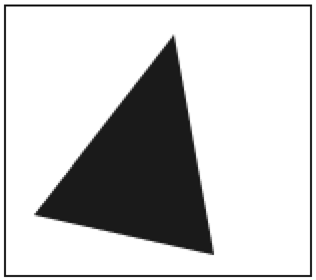
IMG_2



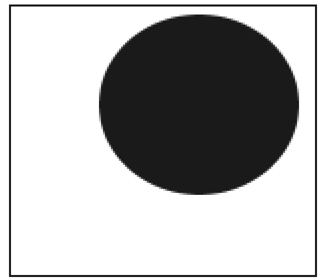
XOR

Step 6:

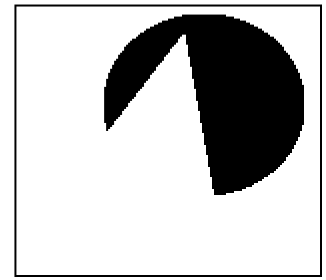
NOT-AND



IMG_1



IMG_2



NOT-AND

- The logic operations AND and OR are used to combine the information in two images.
- This may be done for special effects but a more useful application for image analysis is to perform a masking operation.
- AND and OR can be used as a simple method to extract a ROI from an image.

- For example, a white mask ANDed with an image will allow only the portion of the image coincident with the mask to appear in the output image, with the background turned black; and a black mask ORed with an image will allow only the part f the image corresponding to the black mask to appear in the output image, but will turn the return of the image white.
- This process is called *image masking*

Figure 3.2-10 Image Masking. a) Original image, b) image mask for AND operation, c) Resulting image from (a) AND (b), d) image mask for OR operation, created by performing a NOT on mask (b), e) Resulting image from (a) OR (d).

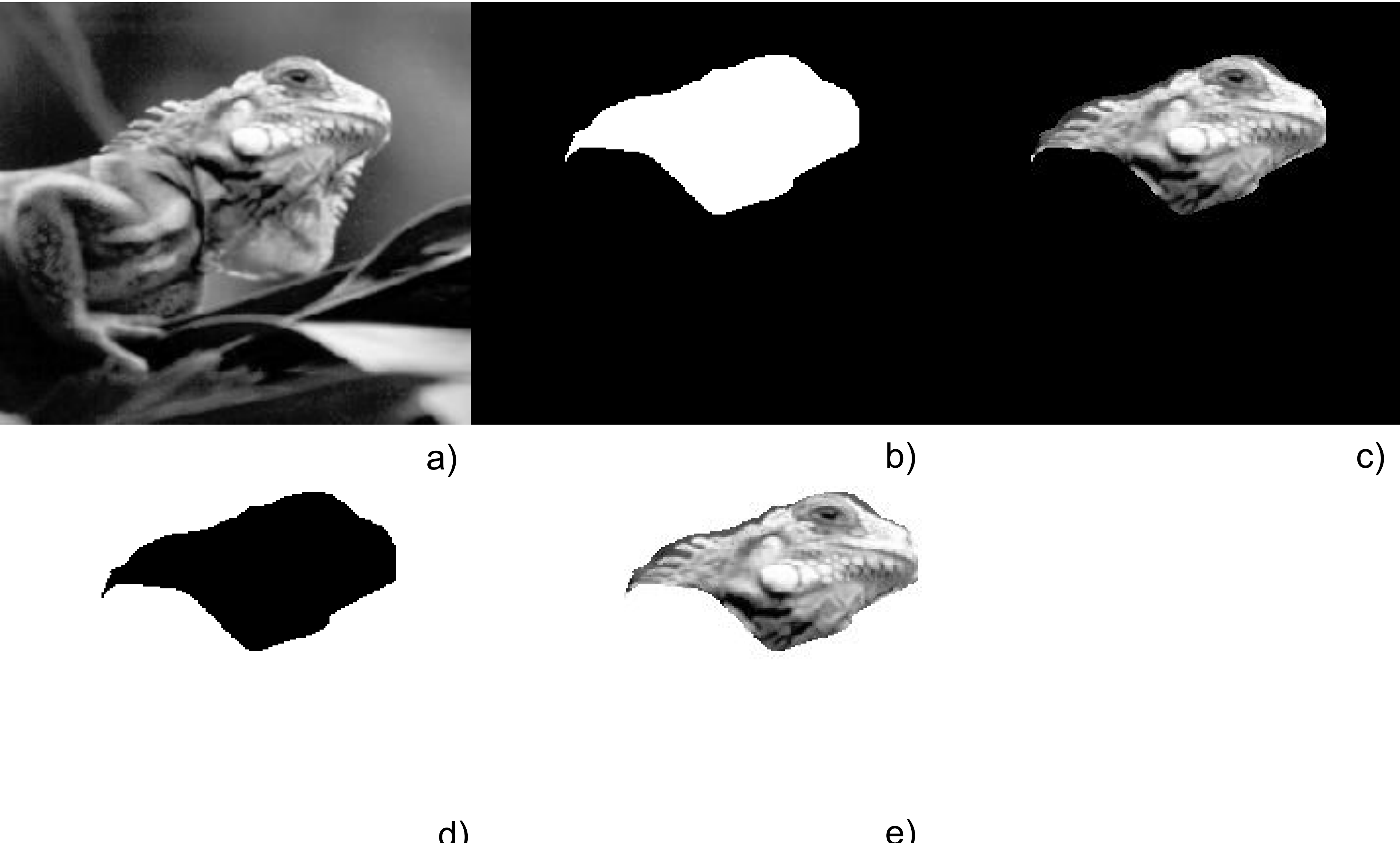
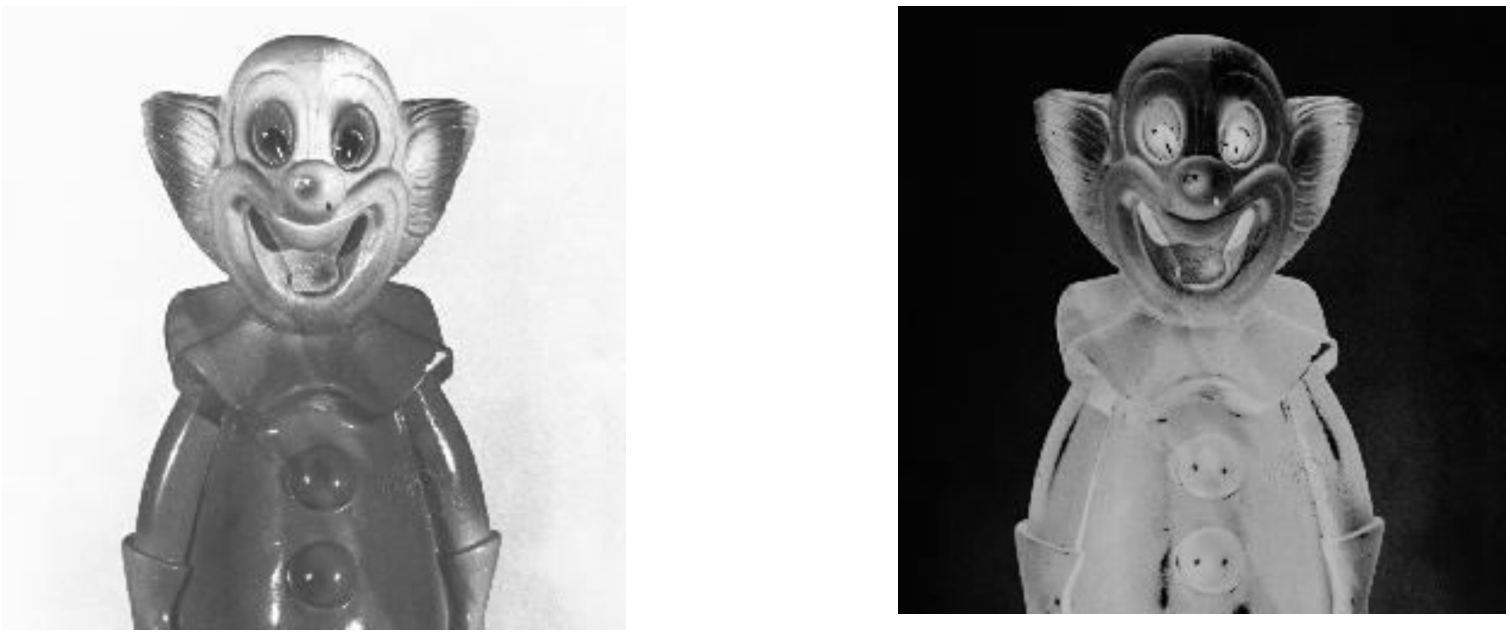


Figure 3.2-11 Complement Image – NOT Operation. a) Original, b) NOT operator applied to the image

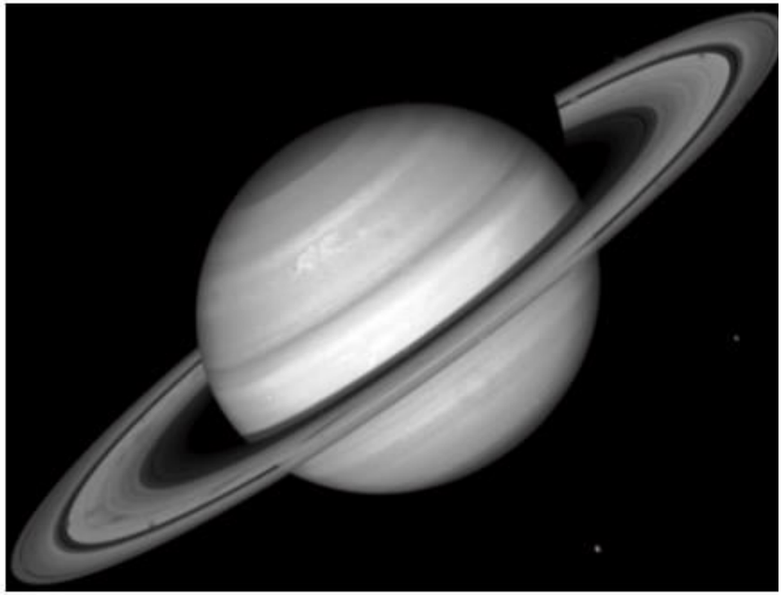


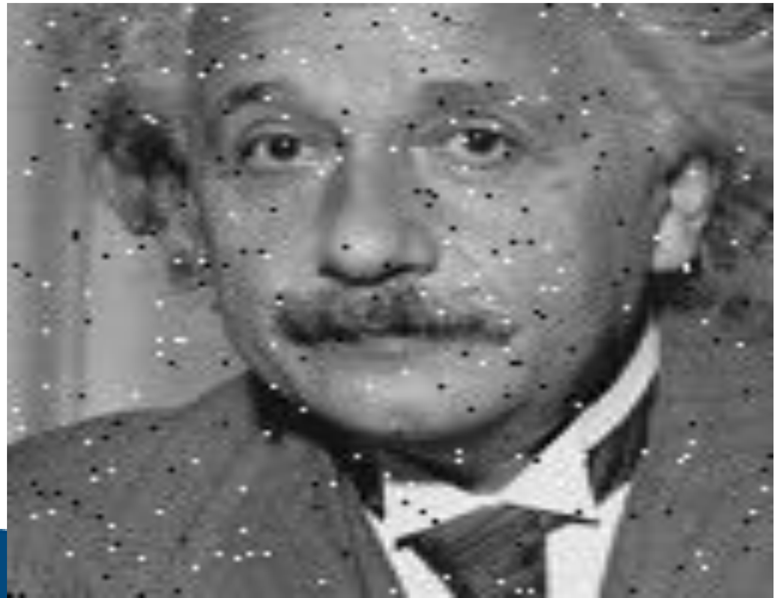
a) b)

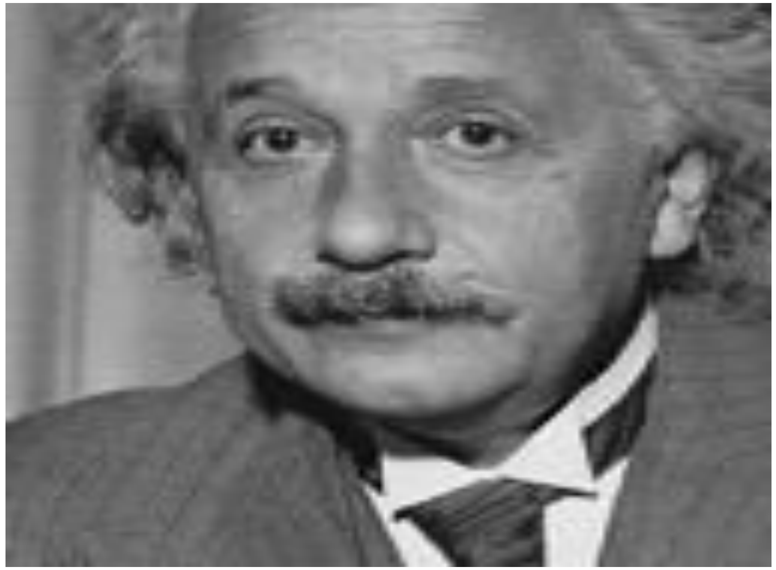
Basics of Spatial Filtering

OUTLINE

- Fundamentals Spatial Filtering
- Smoothing Spatial Filters
  - Smoothing filters are used for blurring and for noise reduction
- Sharpening Spatial Filters
  - Highlight fine detail or enhance detail that has been blurred







Spatial Filtering

- The word "filtering" has been borrowed from the frequency domain.

- Filters are classified as:
  Low-pass (i.e., preserve low frequencies)
  High-pass (i.e., preserve high frequencies)
  Band-pass (i.e., preserve frequencies within a band)
  Band-reject (i.e., reject frequencies within a band)

Spatial Filtering

❖ *Background :*

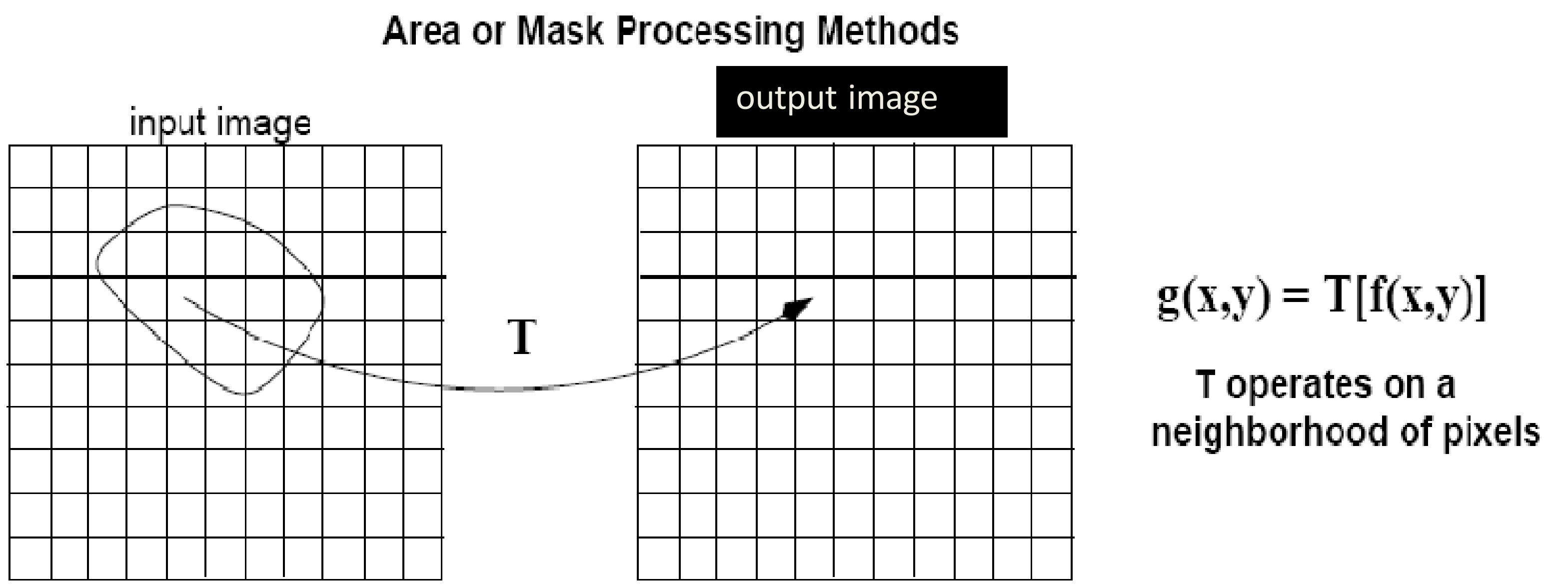
- Filter term in "Digital image processing" is referred to the subimage
- There are others term to call subimage such as mask, kernel, template, or window
- The value in a filter subimage are referred as coefficients, rather than pixels.

❖ *Basics of Spatial Filtering :*

- The concept of filtering has its roots in the use of the Fourier transform for signal processing in the so-called frequency domain.
- Spatial filtering term is the filtering operations that are performed directly on the pixels of an image.

❖ *Definition*

- Spatial filtering are defined by:
  - (1) A **neighborhood**
  - (2) A **predefined operation** that is performed on the pixels inside the neighborhood



Operation with 3x3 Filter

- 3 x 3 Neighborhood / Mask / Window / Template:

	(y - 1)	y	(y + 1)
(x - 1)	w(-1,-1) f(x-1, y-1)	w(-1,0) f(x-1, y)	w(-1,1) f(x-1, y+1)
x	w(0,-1) f(x, y-1)	w(0,0) f(x, y)	w(0,1) f(x, y+1)
(x + 1)	w(1,-1) f(x+1, y-1)	w(1,0) f(x+1, y)	w(1,1) f(x+1, y+1)

❖Spatial Neighborhood

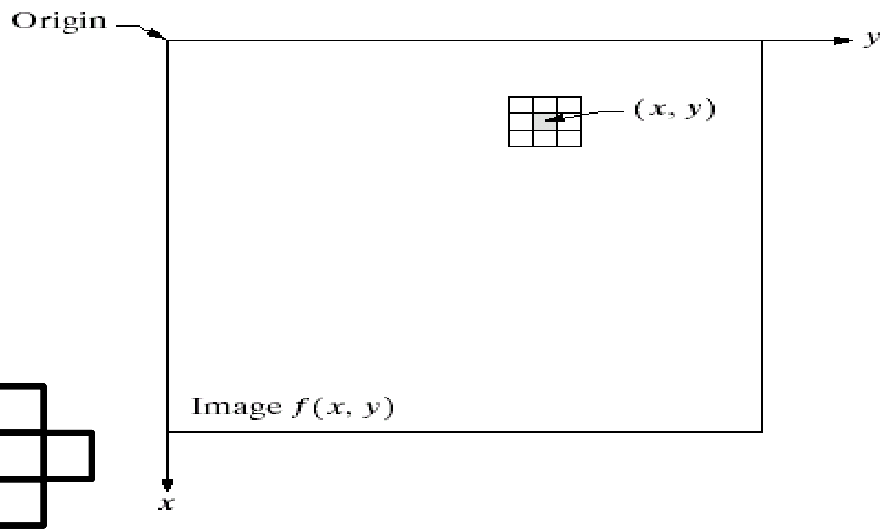
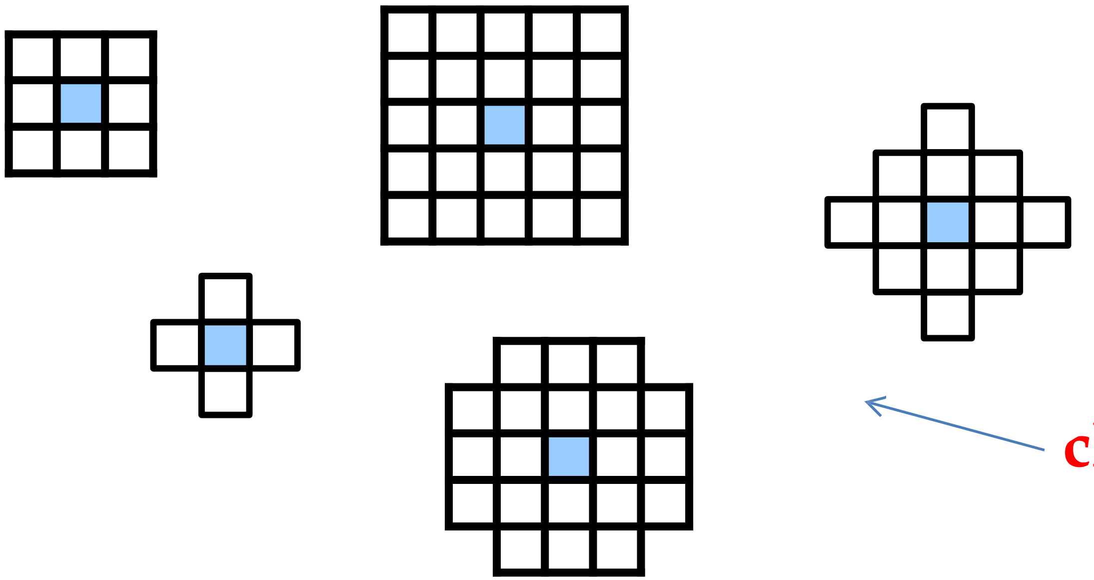


FIGURE 3.1 A 3 × 3 neighborhood about a point (x, y) in an image.



choices of neighborhood

❖ Typically, the neighborhood is **rectangular** and its size is much smaller than that of *f(x,y)* - e.g., 3x3 or 5x5

❖ *Mechanics of spatial filtering*

- The process consists simply of moving the filter mask from point to point in an image.

- At each point (x,y) the response of the filter at that point is calculated using a predefined relationship

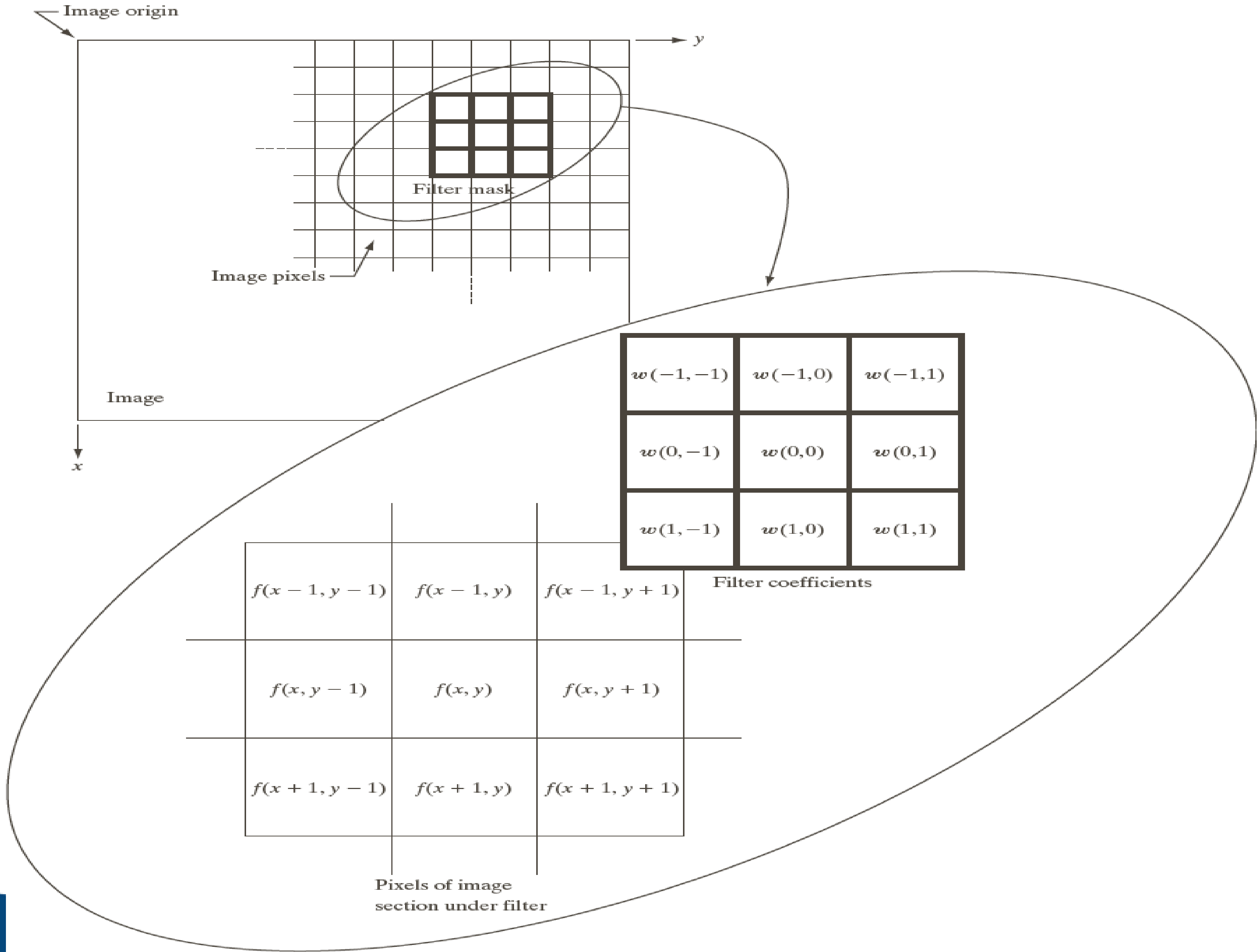


Image origin
y
Filter mask
Image pixels
Image
x
w(−1,−1)
w(−1,0)
w(−1,1)
w(0,−1)
w(0,0)
w(0,1)
w(1,−1)
w(1,0)
w(1,1)
Filter coefficients
f(x − 1, y − 1)
f(x − 1, y)
f(x − 1, y + 1)
f(x, y − 1)
f(x, y)
f(x, y + 1)
f(x + 1, y − 1)
f(x + 1, y)
f(x + 1, y + 1)
Pixels of image
section under filter

❖ *Type*

- Linear spatial filtering
- Nonlinear spatial filtering

➢ A filtering method is linear when the output is a weighted sum of the input pixels.

w1	w2	w3
w4	w5	w6
w7	w8	w9

$$z5' = R = w1z1 + w2z2 + \ldots + z9w9$$

➢ Methods that do not satisfy the above property are called non-linear.

z1	z2	z3
z4	z5	z6
z7	z8	z9

$$z_5' = max(z_k, k = 1, 2, \ldots, 9)$$

Linear Spatial Filtering Methods

- Two main linear spatial filtering methods:
  - Correlation
  - Convolution

The correlation of a filter $w(x, y)$ of size $m \times n$ with an image $f(x, y)$, denoted as $w(x, y) \star f(x, y)$

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

The convolution of a filter $w(x, y)$ of size $m \times n$ with an image $f(x, y)$, denoted as $w(x, y) \bigstar f(x, y)$

$$\boxed{w(x, y) \bigstar f(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x-s, y-t)}$$

Correlation & Convolution

- Correlation & Convolution are two closely related concepts used in linear spatial filtering.
- *Correlation*: It is a process of moving a filter mask over an image & computing the sum of products at each location.
- *Convolution*: Here, the mechanics are same, except that the filter is first rotated by 180˚.
- Correlation & Convolution are function of displacement. Correlation & Convolution are exactly same if the filter mask is symmetric.
- 1D correlation and convolution of a filter with a discrete unit impulse is shown below.

Correlation

Origin f w

(a) 0 0 0 1 0 0 0 0 1 2 3 2 8

(b) 0 0 0 1 0 0 0 0
1 2 3 2 8
Starting position alignment

Zero padding

(c) 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
1 2 3 2 8

(d) 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
1 2 3 2 8
Position after one shift

(e) 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
1 2 3 2 8
Position after four shifts

(f) 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
1 2 3 2 8
Final position

Full correlation result

(g) 0 0 0 8 2 3 2 1 0 0 0 0

Cropped correlation result

(h) 0 8 2 3 2 1 0 0

Convolution

Origin f w rotated 180°

0 0 0 1 0 0 0 0 8 2 3 2 1 (i)

0 0 0 1 0 0 0 0 (j)
8 2 3 2 1

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 (k)
8 2 3 2 1

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 (l)
8 2 3 2 1

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 (m)
8 2 3 2 1

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 (n)
8 2 3 2 1

Full convolution result

0 0 0 1 2 3 2 8 0 0 0 0 (o)

Cropped convolution result

0 1 2 3 2 8 0 0 (p)

Correlation & Convolution

- Correlation is a function of displacement of the filter.
- Correlating a filter w with a function that contains all ‘0’ & single ‘1’ yields a 180˚ rotated copy of w.
- Correlating a function with discrete unit impulse yields a rotated (time inverted) version of the function.
- Convolving a function with a unit impulse yields the same function.
- Thus, to perform convolution all we have to do is rotate one function by 180˚ & perform same operation as in correlation.

Origin $f(x, y)$

0	0	0	0	0
0	0	0	0	0
0	0	1	0	0
0	0	0	0	0
0	0	0	0	0

$w(x, y)$

1	2	3
4	5	6
7	8	9

(a)

Padded f

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

(b)

Initial position for w

1	2	3	0	0	0	0	0	0
4	5	6	0	0	0	0	0	0
7	8	9	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

(c)

Full correlation result

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	9	8	7	0	0	0
0	0	0	6	5	4	0	0	0
0	0	0	3	2	1	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

(d)

Cropped correlation result

0	0	0	0	0
0	9	8	7	0
0	6	5	4	0
0	3	2	1	0
0	0	0	0	0

(e)

Rotated w

9	8	7	0	0	0	0	0	0
6	5	4	0	0	0	0	0	0
3	2	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

(f)

Full convolution result

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	1	2	3	0	0	0
0	0	0	4	5	6	0	0	0
0	0	0	7	8	9	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

(g)

Cropped convolution result

0	0	0	0	0
0	1	2	3	0
0	4	5	6	0
0	7	8	9	0
0	0	0	0	0

(h)

FIGURE 3.30 Correlation (middle row) and convolution (last row) of a 2-D filter with a 2-D discrete, unit impulse. The 0s are shown in gray to simplify visual analysis.

Some fundamental properties of convolution and correlation

Property	Convolution	Correlation
Commutative	$f \star g = g \star f$	-
Associative	$f \star (g \star h) = (f \star g) \star h$	-
Distributive	$f \star (g + h) = (f \star g) + (f \star h)$	$f \,☆\, (g + h) = (f \star g) + (f \star h)$

❖ Linear spatial filtering

Pixels of image

f(x-1.y-1)	f(x-1,y)	f(x-1,y+1)
f(x,y-1)	f(x,y)	f(x,y+1)
f(x+1,y-1)	f(x+1,y)	f(x+1,y+1)

w(-1,-1)	w(-1,0)	w(-1,1)
w(0,-1)	w(0,0)	w(0,1)
w(1,-1)	w(1,0)	w(1,1)

- The result is the sum of products of the mask coefficients with the corresponding pixels directly under the mask

$$g(x,y) = w(-1,-1)f(x-1,y-1) + w(-1,0)f(x-1,y) + w(-1,1)f(x-1,y+1) + w(0,-1)f(x,y-1) + w(0,0)f(x,y) + w(0,1)f(x,y+1) + w(1,-1)f(x+1,y-1) + w(1,0)f(x+1,y) + w(1,1)f(x+1,y+1)$$

❖ Nonlinear spatial filtering

- Nonlinear spatial filters also operate on neighborhoods, and the mechanics of sliding a mask past an image are the same as was just outlined.
- The filtering operation is based conditionally on the values of the pixels in the neighborhood under consideration

  - Ex: Noise reduction can be achieved effectively with a nonlinear filter with a basic function of computing the median gray level value in the neighbourhood

  - Computation of median is a non-linear operation, as is that of variance

- **Motivation: Limitation of Linear Filters**
  - Frequency shaping

    enhance some frequency components and suppress the others
  - For individual frequency component, cannot differentiate its "desirable" and "undesirable" parts

- Nonlinear Filters
  - Cannot be expressed as convolution
  - Cannot be expressed as frequency shaping

- **"Nonlinear" Means Everything** (other than linear)
  - Need to be more specific
  - Often heuristic
  - We will study some "nice" ones

SMOOTHING FILTERS(Low Pass) IN SPATIAL DOMAIN

Smoothing Spatial Filter

Smoothing filters are used for

- blurring
- noise reduction.

Blurring is used in preprocessing steps to removal of small details from an image prior to object extraction and bridging of small gaps in lines or curves

Noise reduction can be accomplished by blurring

Types of Smoothing Filter

- There are 2 way of smoothing spatial filters
  - LinearFilters –operations performed on image pixel
  - Order-Statistics(non-linear)Filters - based on ranking the pixels

Linear Filter

Linear spatial filter is simply the average of the pixels contained in the neighborhood of the filter mask.

The idea is replacing the value of every pixel in an image by the average of the gray levels in the neighborhood defined by the filter mask.

Linear Filter (cont..)

This process result in an image reduce the sharp transitions in intensities.

Two mask

- Averaging filter
- Weighted averaging filter

Averaging Filter

- A major use of averaging filters is in the reduction of irrelevant detail in image.
- mxn mask would have a normalizing constant equal to 1/mn.
- Its also known as low pass filter.
- A spatial averaging filter in which all coefficients are equal is called a box filter.

Averaging Filter - Example

$$\frac{1}{9} \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

Smoothing Filters: Averaging

$$\frac{1}{25} \times \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$\frac{1}{49} \times \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Weighted Average Filter - Example

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

Weighted Averaging Filter

Weighted Averaging Filter

- The general implementation for filtering an MxN image with a weighted averaging filter of size m x n is given by the expression

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

- For complete filtered image apply **x = 0,1,2,3……..m-1** and **y=0,1,2,3,……..n-1** in the above equation.

Ex. 1) 8x8 Pseudo image with a single edge (High Frequency) of 10 & 50. Remove using a 3x3 size averaging mask.

10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50

8x8 Image

10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50

$$\frac{1}{9}\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline\end{array}$$

0	0	0						
0	10	10	10	10	10	10	10	10
0	10	10	10	10	10	10	10	10
	10	10	10	10	10	10	10	10
	10	10	10	10	10	10	10	10
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50

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

0	0	0						
0	4.44	10	10	10	10	10	10	10
0	10	10	10	10	10	10	10	10
	10	10	10	10	10	10	10	10
	10	10	10	10	10	10	10	10
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50

$$\frac{1}{9}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 10 & 10 \\ 0 & 10 & 10 \end{bmatrix}$$

0	0	0	0					
0	4.44	6.66	10	10	10	10	10	10
0	10	10	10	10	10	10	10	10
	10	10	10	10	10	10	10	10
	10	10	10	10	10	10	10	10
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50

$$\frac{1}{9}\begin{bmatrix} 0 & 0 & 0 \\ 10 & 10 & 10 \\ 10 & 10 & 10 \end{bmatrix}$$

0	0	0	0	0				
0	4.44	6.66	6.66	10	10	10	10	10
0	10	10	10	10	10	10	10	10
	10	10	10	10	10	10	10	10
	10	10	10	10	10	10	10	10
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50

$$\frac{1}{9}\begin{bmatrix}0 & 0 & 0\\ 10 & 10 & 10\\ 10 & 10 & 10\end{bmatrix}$$

0	0	0	0	0	0			
0	4.44	6.66	6.66	6.66	10	10	10	10
0	10	10	10	10	10	10	10	10
	10	10	10	10	10	10	10	10
	10	10	10	10	10	10	10	10
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50

$$\frac{1}{9}\begin{bmatrix}0 & 0 & 0\\ 10 & 10 & 10\\ 10 & 10 & 10\end{bmatrix}$$

0	0	0	0	0	0	0	0	0	
0	4.44	6.66	6.66	6.66	6.66	6.66	6.66	4.44	0
0	6.66	10	10	10	10	10	10	10	0
	10	10	10	10	10	10	10	10	
	10	10	10	10	10	10	10	10	
	50	50	50	50	50	50	50	50	
	50	50	50	50	50	50	50	50	
	50	50	50	50	50	50	50	50	
	50	50	50	50	50	50	50	50	

$$\frac{1}{9}\begin{bmatrix} 10 & 10 & 10 \\ 10 & 10 & 10 \\ 10 & 10 & 10 \end{bmatrix}$$

0	0	0	0	0	0	0	0	0	
0	4.44	6.66	6.66	6.66	6.66	6.66	6.66	4.44	0
0	6.66	10	10	10	10	10	10	6.66	0
0	6.66	10	10	10	10	10	10	6.66	0
0	15.55	10	10	10	10	10	10	10	0
0	50	50	50	50	50	50	50	50	0
	50	50	50	50	50	50	50	50	
	50	50	50	50	50	50	50	50	
	50	50	50	50	50	50	50	50	

$$\frac{1}{9}\begin{array}{|c|c|c|}\hline 0 & 10 & 10 \\ \hline 0 & 10 & 10 \\ \hline 0 & 50 & 50 \\ \hline\end{array}$$

0	0	0	0	0	0	0	0	0	
0	4.44	6.66	6.66	6.66	6.66	6.66	6.66	4.44	0
0	6.66	10	10	10	10	10	10	6.66	0
0	6.66	10	10	10	10	10	10	6.66	0
0	15.55	23.33	23.33	23.33	23.33	23.33	23.33	15.55	0
0	24.44	36.66	50	50	50	50	50	50	0
	50	50	50	50	50	50	50	50	
	50	50	50	50	50	50	50	50	
	50	50	50	50	50	50	50	50	

$$\frac{1}{9}\begin{bmatrix} 10 & 10 & 10 \\ 50 & 50 & 50 \\ 50 & 50 & 50 \end{bmatrix}$$

0	0	0	0	0	0	0	0	0	
0	4.44	6.66	6.66	6.66	6.66	6.66	6.66	4.44	0
0	6.66	10	10	10	10	10	10	6.66	0
0	6.66	10	10	10	10	10	10	6.66	0
0	15.55	23.33	23.33	23.33	23.33	23.33	23.33	15.55	0
0	24.44	36.66	36.66	36.66	36.66	36.66	36.66	24.44	0
0	33.33	50	50	50	50	50	50	50	
	50	50	50	50	50	50	50	50	
	50	50	50	50	50	50	50	50	

$$\frac{1}{9} \begin{bmatrix} 50 & 50 & 50 \\ 50 & 50 & 50 \\ 50 & 50 & 50 \end{bmatrix}$$

0	0	0	0	0	0	0	0	0	
0	4.44	6.66	6.66	6.66	6.66	6.66	6.66	4.44	0
0	6.66	10	10	10	10	10	10	6.66	0
0	6.66	10	10	10	10	10	10	6.66	0
0	15.55	23.33	23.33	23.33	23.33	23.33	23.33	15.55	0
0	24.44	36.66	36.66	36.66	36.66	36.66	36.66	24.44	0
0	33.33	50	50	50	50	50	50	33.33	0
0	33.33	50	50	50	50	50	50	33.33	0
0	22.22	33.33	33.33	33.33	33.33	33.33	33.33	22.22	0
	0	0	0	0	0	0	0	0	0

$$\frac{1}{9}\begin{bmatrix} 50 & 50 & 0 \\ 50 & 50 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	23.33	23.33	23.33	23.33	23.33	23.33	10
50	36.66	36.66	36.66	36.66	36.66	36.66	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50

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

- In the resultant image the Low frequency region has remained unchanged.
- Sharp transition between 10 & 50 has changed from 10 to 23.33 to 36.66 and finally to 50.
- Thus, Sharp edges has become blurred.
- Best result when used over image corrupted by Gaussian noise.
- Other types of low pass averaging mask are:

$$\frac{1}{6}\begin{bmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 0 \end{bmatrix} \qquad \frac{1}{10}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Smoothing filters – Example

input image

smoothed image

Order-Statistics Filter

- Order-statistics filters are nonlinear spatial filters.
- It is based on ordering (ranking) the pixels contained in the image area encompassed by the filter,

It replacing the value of the center pixel with the value determined by the ranking result.

PRESIDENCY UNIVERSITY

Order- statics filter

- The filter selects a sample from the window, does not average
- Edges are better preserved than with liner filters
- Best suited for “salt and pepper” noise

Types of order-statics filter

Different types of order-statics filters are

- Minimum filter
- Maximum filter
- Median filter

Minimum Filter

The 0^{th} percentile filter is the min filter.

- Minimum filter selects the smallest value in the window and replace the center by the smallest value

- Using comparison the minimum value can be obtained fast.(not necessary to sort)

- It enhances the dark areas of image

PRESIDENCY UNIVERSITY
GAIN MORE KNOWLEDGE REACH GREATER HEIGHTS
Private University Estd. in Karnataka State by Act No. 41 of 2013
PRESIDENCY GROUP OVER 40 YEARS OF ACADEMIC WISDOM

Minimum Filter - Example

(mask size =3 x 3)

(mask size =7 x 7)

Maximum Filter

- The maximum filter selects the largest value within of pixel values, and replace the center by the largest value.

- Using comparison the maximum value can be obtained fast.(not necessary to sort)
- Using the 100th percentile results in the so-called *max filter*
- it enhances bright areas of image

Maximum Filter

mask (3 x 3)

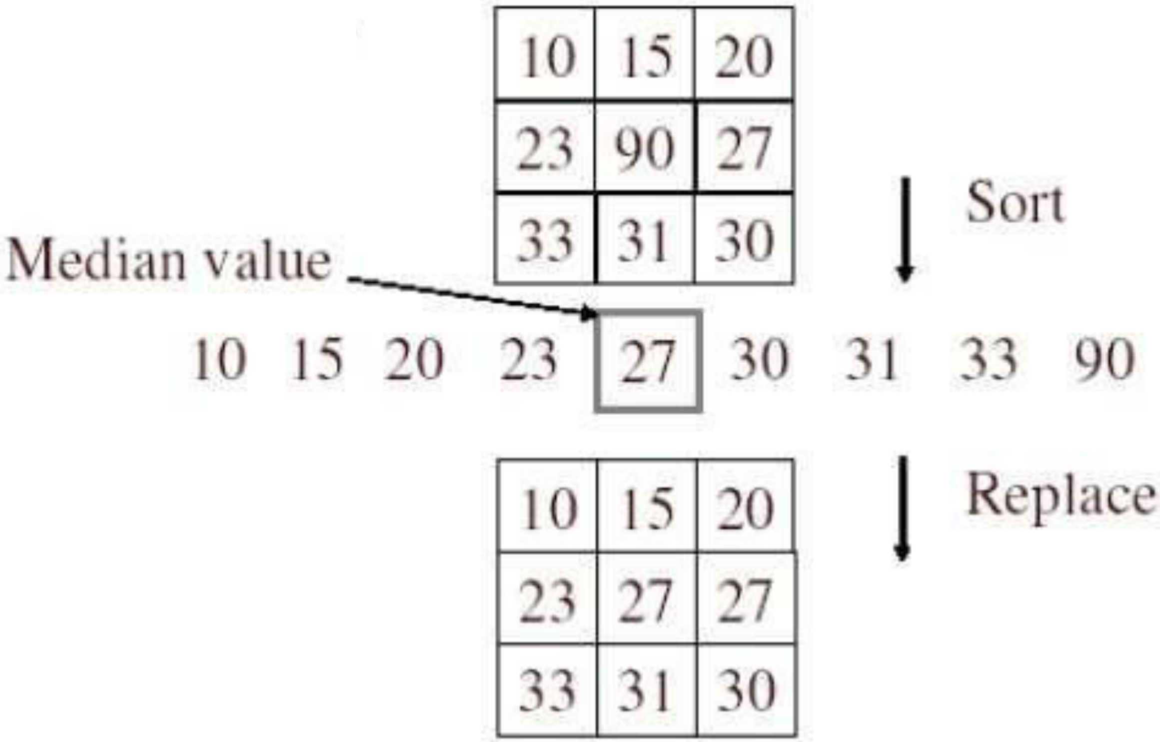
mask (7 x 7)

Median Filter

- Three steps to be followed to run a median filter:

1. Consider each pixel in the image
2. Sort the neighboring pixels into order based upon their intensities
3. Replace the original value of the pixel with the median value from the list.

Median Filter - Process



PRESIDENCY UNIVERSITY

Ex. 2) 8x8 Pseudo image with a single edge (High Frequency) of 10 & 50. Remove using a 3x3 size median filter mask.

10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	250	10	10	10	10	10	10
10	10	10	10	10	10	10	10
50	50	50	50	250	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50

8x8 Image

10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	250	10	10	10	10	10	10
10	10	10	10	10	10	10	10
50	50	50	50	250	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50

8x8 Image with blank mask

10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	250	10	10	10	10	10	10
10	10	10	10	10	10	10	10
50	50	50	50	250	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50

8x8 Image with blank mask

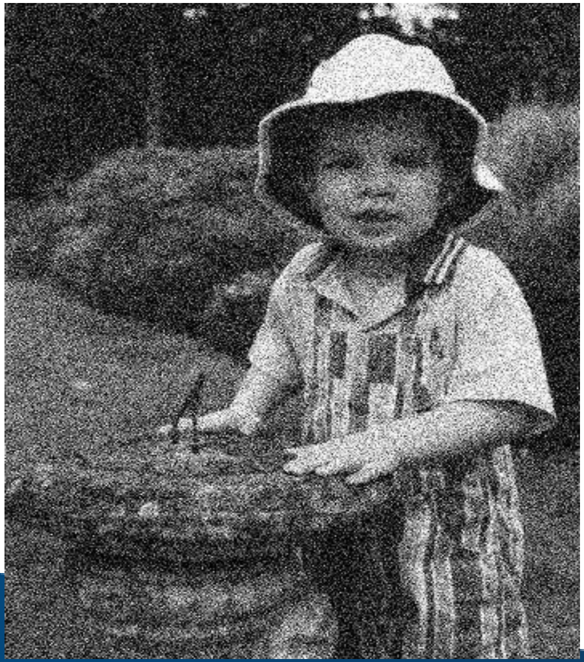
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
50	50	50	50	250	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50

8x8 Image with blank mask

10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50

8x8 Image with blank mask

Median Filter - Example



Median Filter size =3 x 3

Median Filter size =7 x 7

Conclusion

- A linear filter cannot totally eliminate impulse noise, as a single pixel which acts as an intensity spike can contribute significantly to the weighted average of the filter.

- Non-linear filters can be robust to this type of noise because single outlier pixel intensities can be eliminated entirely.

Original image

Mean filter

Median filter

Sharpening Filters(High pass filter)

1. The concept of sharpening filter
2. First and second order derivatives
3. Laplace filter
4. Unsharp mask
5. High boost filter
6. Gradient mask

Sharpening Spatial Filters

- To highlight fine detail in an image or to enhance detail that has been blurred, either in error or as a natural effect of a particular method of image acquisition.

- Blurring vs. Sharpening

  - Blurring/smooth is done in spatial domain by pixel averaging in a neighbors, it is a process of integration

  - Sharpening is an inverse process, to find the difference by the neighborhood, done by spatial differentiation.

PRESIDENCY UNIVERSITY
GAIN MORE KNOWLEDGE REACH GREATER HEIGHTS
Private University Estd. in Karnataka State by Act No. 41 of 2013

Derivative operator

- The strength of the response of a derivative operator is proportional to the degree of discontinuity of the image at the point at which the operator is applied.

- Image differentiation
  - enhances edges and other discontinuities (noise)
  - deemphasizes area with slowly varying gray-level values.

First and second order difference of 1D

- The basic definition of the first-order derivative of a one-dimensional function f(x) is the difference

$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

- The second-order derivative of a one-dimensional function f(x) is the difference

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$$

PRESIDENCY UNIVERSITY
GAIN MORE KNOWLEDGE REACH GREATER HEIGHTS
Private University Estd. in Karnataka State by Act No. 41 of 2013
PRESIDENCY GROUP OVER 40 YEARS OF ACADEMIC WISDOM

First and Second-order derivative of 2D

- when we consider an image function of two variables, f(x, y), at which time we will dealing with partial derivatives along the two spatial axes.

Gradient operator

(linear operator)

$$\nabla \mathrm{f} = \frac{\partial f(x,y)}{\partial x \partial y} = \frac{\partial f(x,y)}{\partial x} + \frac{\partial f(x,y)}{\partial y}$$

Laplacian operator (non-linear)

$$\nabla^2 f = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2}$$

PRESIDENCY UNIVERSITY

GAIN MORE KNOWLEDGE REACH GREATER HEIGHTS

Private University Estd. in Karnataka State by Act No. 41 of 2013

PRESIDENCY GROUP OVER 40 YEARS OF ACADEMIC WISDOM

Discrete form of Laplacian

from $$\frac{\partial^2 f}{\partial x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y)$$

$$\frac{\partial^2 f}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y)$$

$$\nabla^2 f = [f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y)]$$

Result Laplacian mask

0	1	0
1	−4	1
0	1	0

Laplacian mask implemented an extension of diagonal neighbors

1	1	1
1	−8	1
1	1	1

PRESIDENCY UNIVERSITY
GAIN MORE KNOWLEDGE REACH GREATER HEIGHTS
Private University Estd. in Karnataka State by Act No. 41 of 2013
PRESIDENCY GROUP OVER 40 YEARS OF ACADEMIC WISDOM

Other implementation of Laplacian masks

0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

give the same result, but we have to keep in mind that when combining (add / subtract) a Laplacian-filtered image with another image.

Effect of Laplacian Operator

- as it is a derivative operator,
  - it highlights gray-level discontinuities in an image
  - it deemphasizes regions with slowly varying gray levels
- tends to produce images that have
  - grayish edge lines and other discontinuities, all superimposed on a dark,
  - featureless background.

Correct the effect of featureless background

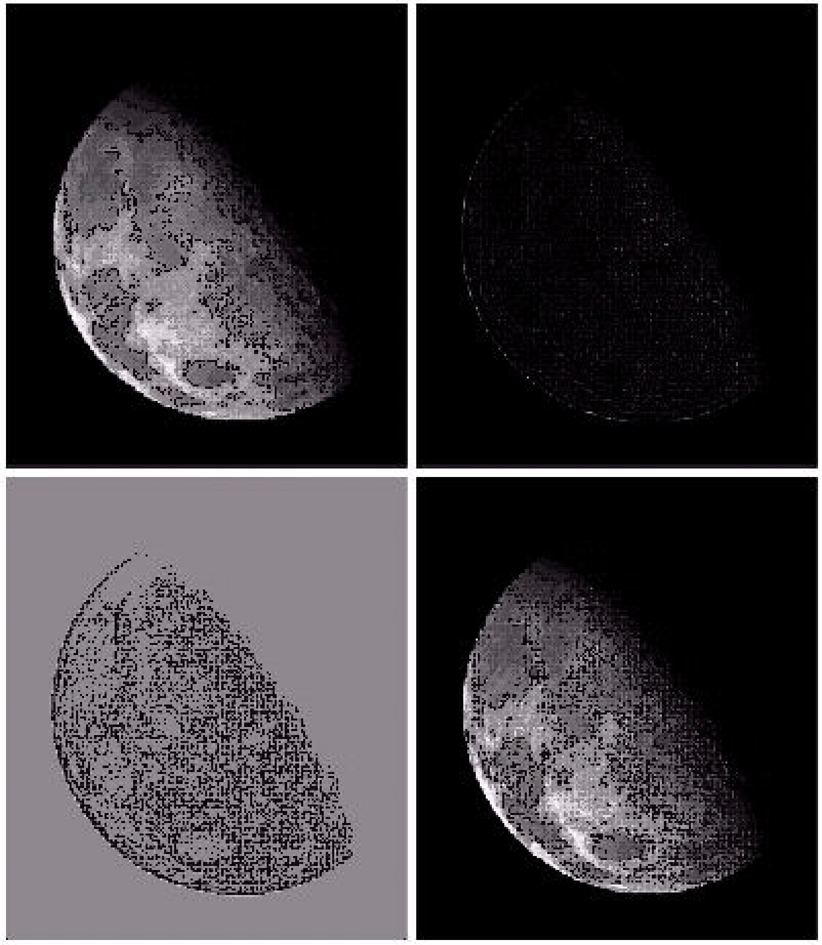
- easily by adding the original and Laplacian image.
- be careful with the Laplacian filter used

$$g(x,y)=\begin{cases} f(x,y)-\nabla^2 f(x,y) \\ f(x,y)+\nabla^2 f(x,y) \end{cases}$$

if the center coefficient of the Laplacian mask is negative

if the center coefficient of the Laplacian mask is positive

Example



- a). image of the North pole of the moon
- b). Laplacian-filtered image with

1	1	1
1	-8	1
1	1	1

- c). Laplacian image scaled for display purposes
- d). image enhanced by addition with original image

Mask of Laplacian + addition

- to simply the computation, we can create a mask which do both operations, Laplacian Filter and Addition the original image.

Mask of Laplacian + addition

$$g(x,y) = f(x,y) - [f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) + 4f(x,y)]$$

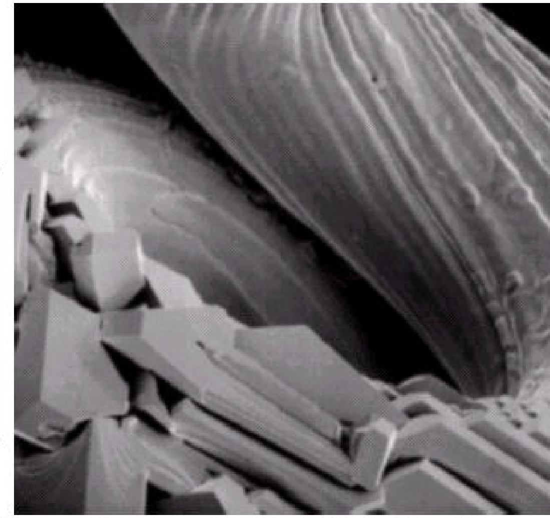
$$= 5f(x,y) - [f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1)]$$

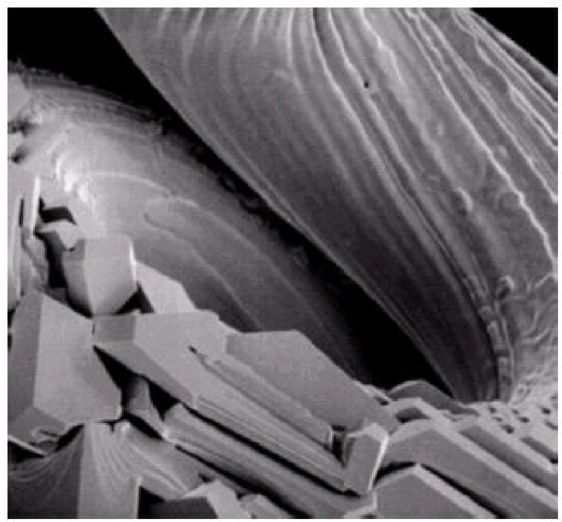
0	-1	0
-1	5	-1
0	-1	0

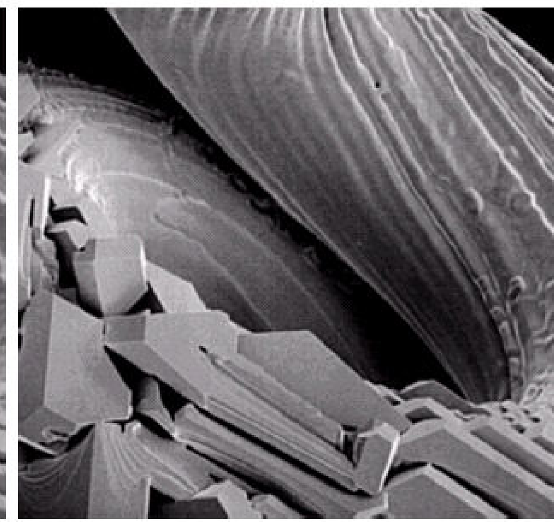
Example

0	–1	0
–1	5	–1
0	–1	0

–1	–1	–1
–1	9	–1
–1	–1	–1







a b c
d e

FIGURE 3.41 (a) Composite Laplacian mask. (b) A second composite mask. (c) Scanning electron microscope image. (d) and (e) Results of filtering with the masks in (a) and (b), respectively. Note how much sharper (e) is than (d). (Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene.)

Note

$$g(x,y)=\begin{cases} f(x,y)-\nabla^2 f(x,y) \\ f(x,y)+\nabla^2 f(x,y) \end{cases}$$

$$\begin{array}{|c|c|c|}\hline 0 & -1 & 0 \\ \hline -1 & 5 & -1 \\ \hline 0 & -1 & 0 \\ \hline \end{array} = \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} + \begin{array}{|c|c|c|}\hline 0 & -1 & 0 \\ \hline -1 & 4 & -1 \\ \hline 0 & -1 & 0 \\ \hline \end{array}$$

$$\begin{array}{|c|c|c|}\hline 0 & -1 & 0 \\ \hline -1 & 9 & -1 \\ \hline 0 & -1 & 0 \\ \hline \end{array} = \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} + \begin{array}{|c|c|c|}\hline 0 & -1 & 0 \\ \hline -1 & 8 & -1 \\ \hline 0 & -1 & 0 \\ \hline \end{array}$$

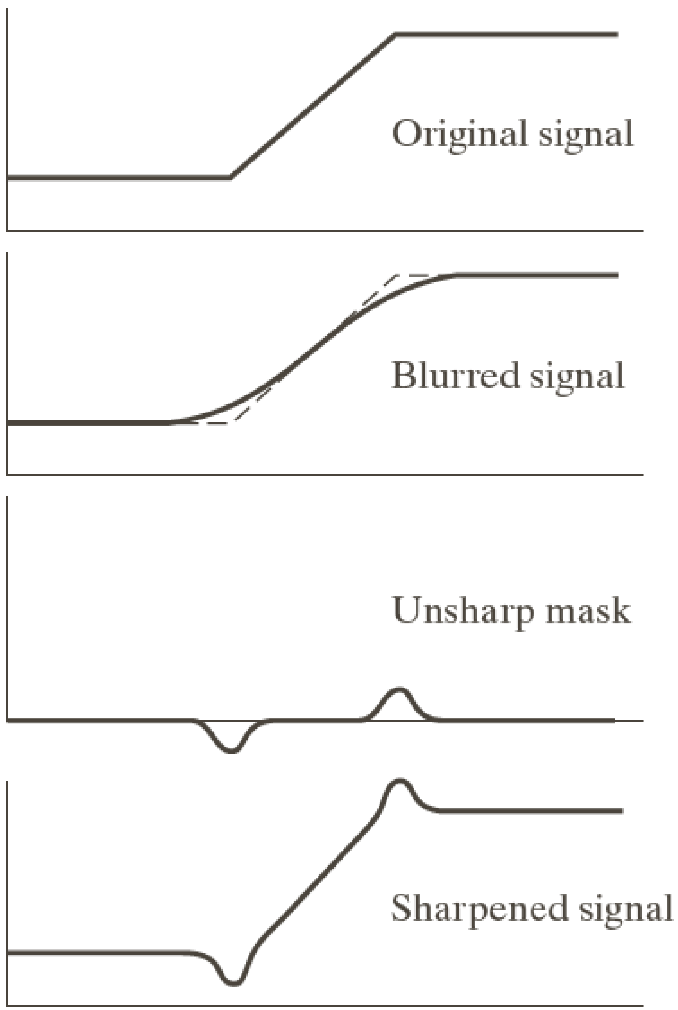
Unsharp masking

$$f_s(x, y) = f(x, y) - \bar{f}(x, y)$$

sharpened image = original image – blurred image

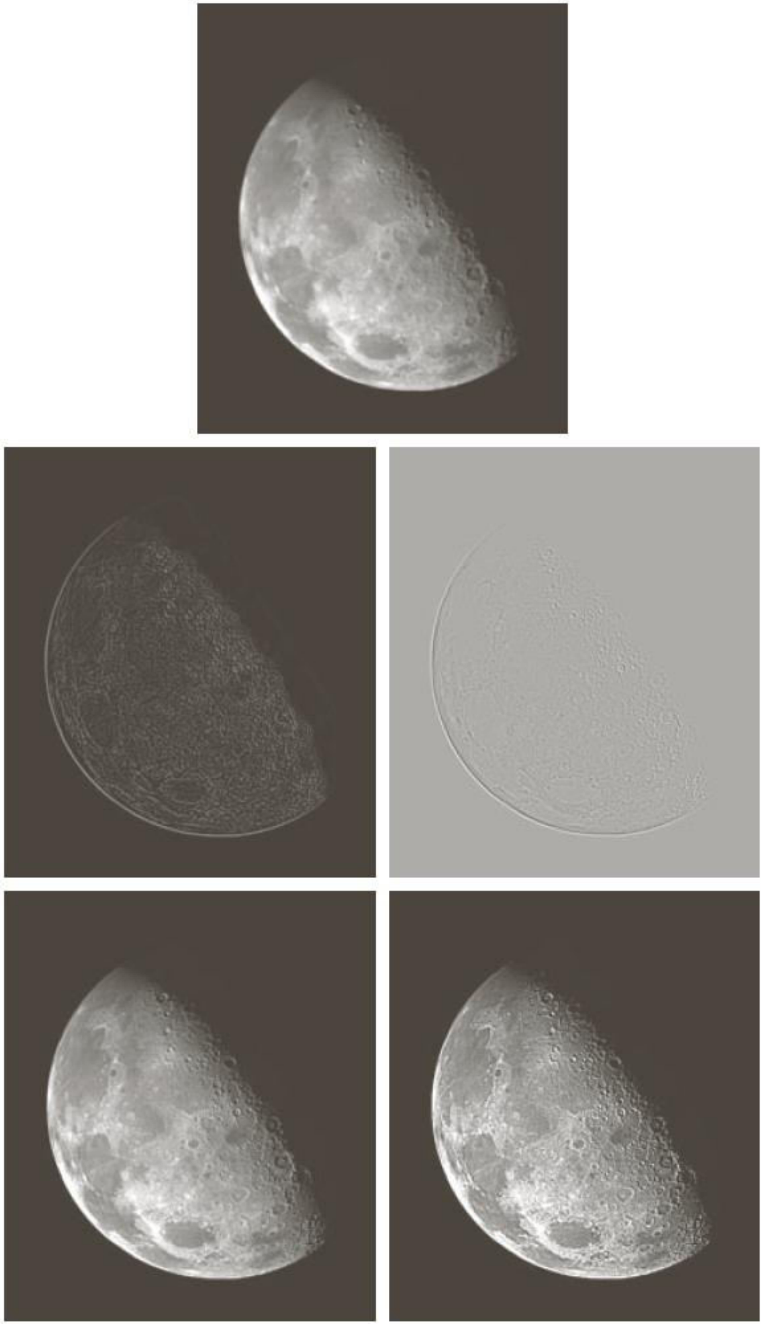
- to subtract a blurred version of an image produces sharpening output image.

Unsharp mask



a
b
c
d

FIGURE 3.39 1-D illustration of the mechanics of unsharp masking. (a) Original signal. (b) Blurred signal with original shown dashed for reference. (c) Unsharp mask. (d) Sharpened signal, obtained by adding (c) to (a).



a	
b	c
d	e

FIGURE 3.38
(a) Blurred image of the North Pole of the moon. (b) Laplacian without scaling. (c) Laplacian with scaling. (d) Image sharpened using the mask in Fig. 3.37(a). (e) Result of using the mask in Fig. 3.37(b). (Original image courtesy of NASA.)

High-boost filtering

$$f_{hb}(x, y) = Af(x, y) - \bar{f}(x, y)$$

$$f_{hb}(x, y) = (A-1)f(x, y) - f(x, y)\bar{f}(x, y)$$

$$= (A-1)f(x, y) - f_s(x, y)$$

- generalized form of Unsharp masking
- A ≥ 1

High-boost filtering

$$f_{hb}(x,y) = (A-1)f(x,y) - f_s(x,y)$$

- if we use Laplacian filter to create sharpen image $f_s(x,y)$ with addition of original image

$$f_s(x,y) = \begin{cases} f(x,y) - \nabla^2 f(x,y) \\ f(x,y) + \nabla^2 f(x,y) \end{cases}$$

$$f_{hb}(x,y) = \begin{cases} Af(x,y) - \nabla^2 f(x,y) \\ Af(x,y) + \nabla^2 f(x,y) \end{cases}$$

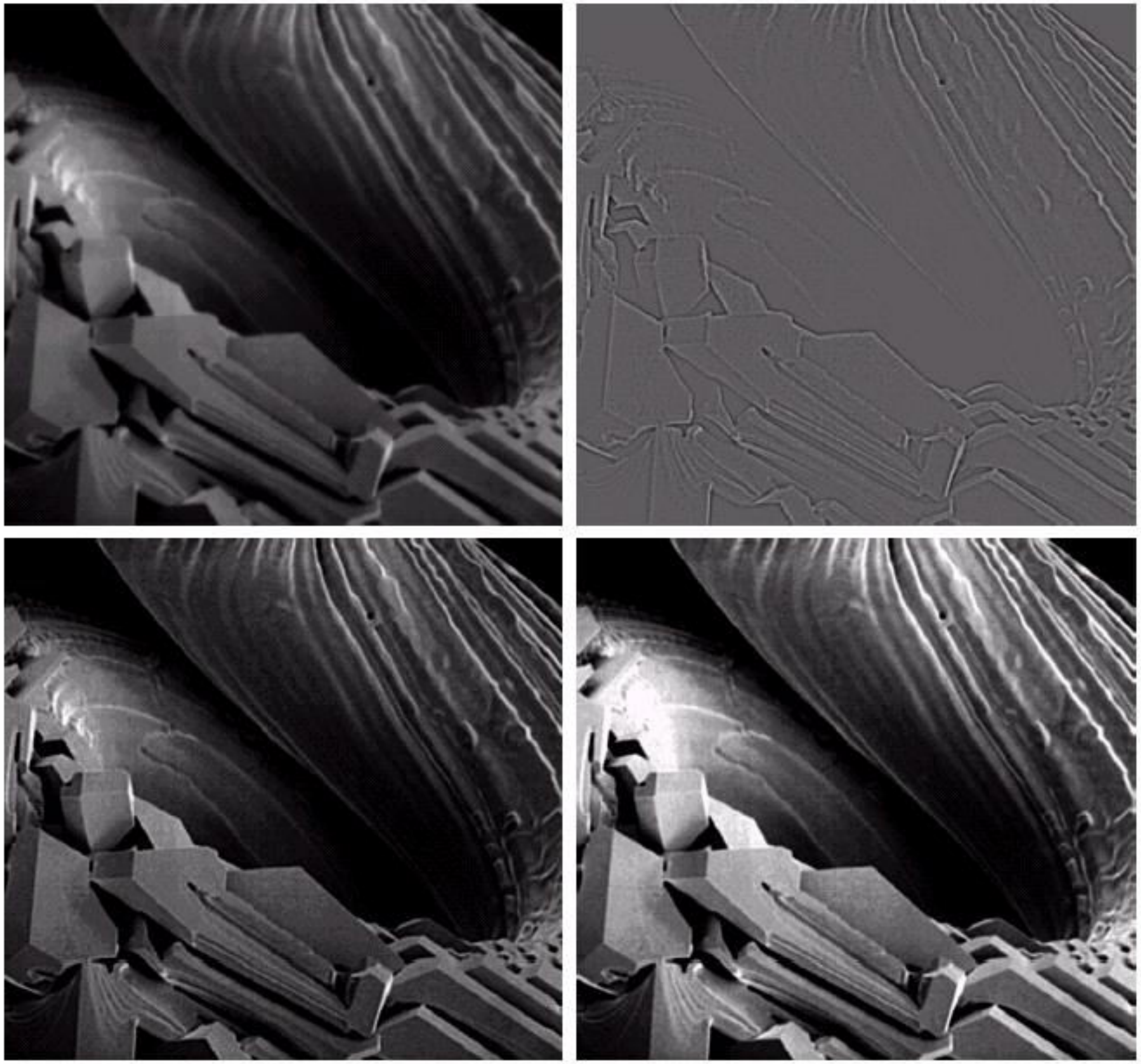
High-boost Masks

0	−1	0
−1	$A + 4$	−1
0	−1	0

−1	−1	−1
−1	$A + 8$	−1
−1	−1	−1

- $A \geq 1$
- if $A = 1$, it becomes “standard” Laplacian sharpening

Example



a b
c d

FIGURE 3.43
(a) Same as Fig. 3.41(c), but darker.
(a) Laplacian of (a) computed with the mask in Fig. 3.42(b) using $A = 0$.
(c) Laplacian enhanced image using the mask in Fig. 3.42(b) with $A = 1$. (d) Same as (c), but using $A = 1.7$.

Gradient Operator

- first derivatives are implemented using the magnitude of the gradient.

$$\nabla \mathrm{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix}$$

$$\nabla f = mag(\nabla \mathrm{f}) = [G_x^2 + G_y^2]^{1/2}$$

$$= \left[\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right]^{1/2}$$

commonly approx.

$$\nabla f \approx |G_x| + |G_y|$$

the magnitude becomes nonlinear

PRESIDENCY UNIVERSITY

GAIN MORE KNOWLEDGE REACH GREATER HEIGHTS

Private University Estd. in Karnataka State by Act No. 41 of 2013

PRESIDENCY GROUP OVER 40 YEARS OF ACADEMIC WISDOM

Gradient Mask

- simplest approximation, 2x2

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

$$G_x = (z_8 - z_5) \quad \text{and} \quad G_y = (z_6 - z_5)$$

$$\nabla f = [G_x^2 + G_y^2]^{1/2} = [(z_8 - z_5)^2 + (z_6 - z_5)^2]^{1/2}$$

$$\nabla f \approx |z_8 - z_5| + |z_6 - z_5|$$

Gradient Mask

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

- Roberts cross-gradient operators, 2x2

$$G_x = (z_9 - z_5) \quad \text{and} \quad G_y = (z_8 - z_6)$$

$$\nabla f = [G_x^2 + G_y^2]^{1/2} = [(z_9 - z_5)^2 + (z_8 - z_6)^2]^{1/2}$$

$$\nabla f \approx |z_9 - z_5| + |z_8 - z_6|$$

−1	0
0	1

0	−1
1	0

PRESIDENCY UNIVERSITY
GAIN MORE KNOWLEDGE REACH GREATER HEIGHTS
Private University Estd. in Karnataka State by Act No. 41 of 2013
PRESIDENCY GROUP OVER 40 YEARS OF ACADEMIC WISDOM

Gradient Mask

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

- Sobel operators, 3x3

$$G_x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$$

$$G_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$$

$$\nabla f \approx |G_x| + |G_y|$$

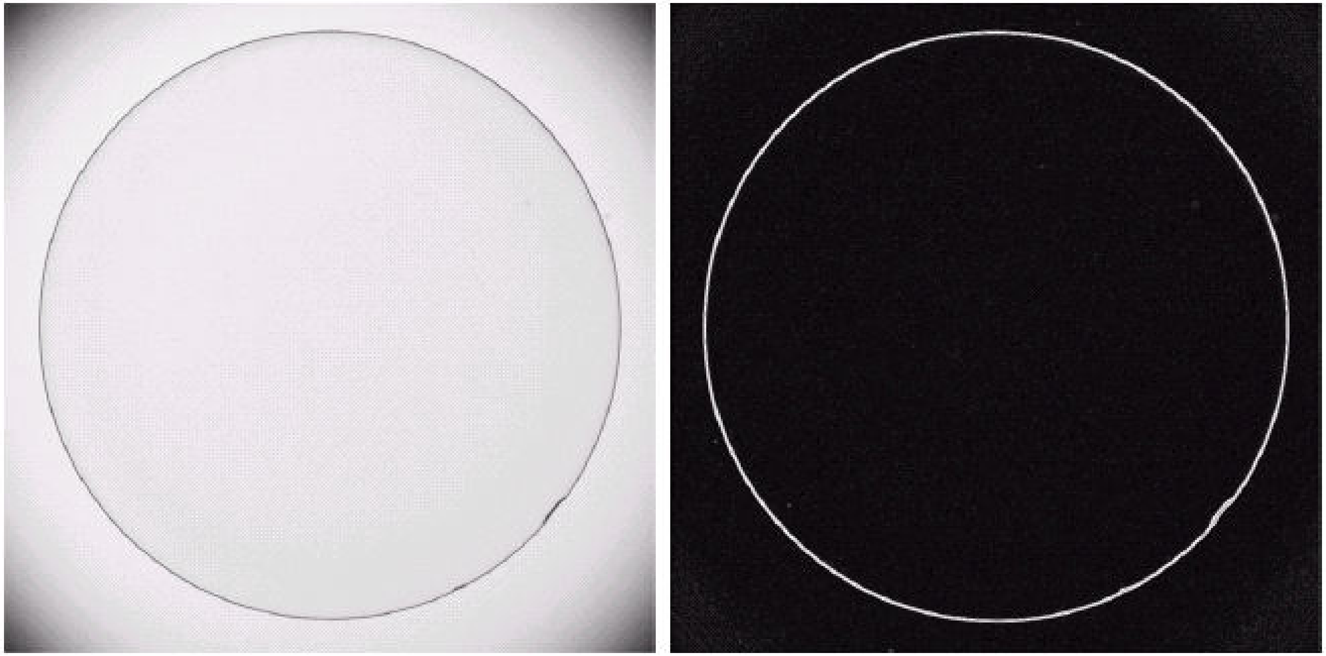
the weight value 2 is to achieve smoothing by giving more important to the center point

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

PRESIDENCY UNIVERSITY
GAIN MORE KNOWLEDGE REACH GREATER HEIGHTS
Private University Estd. in Karnataka State by Act No. 41 of 2013
PRESIDENCY GROUP OVER 40 YEARS OF ACADEMIC WISDOM

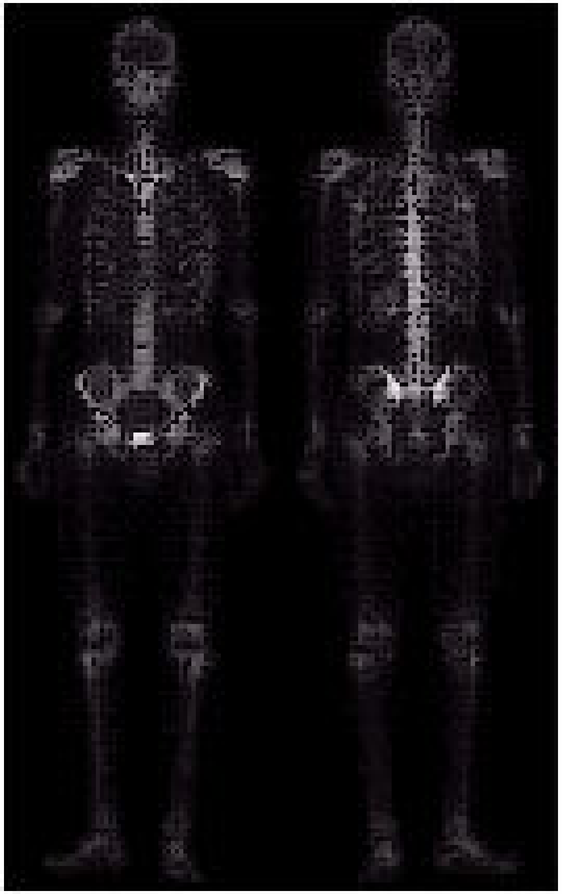
Example



a b

FIGURE 3.45 Optical image of contact lens (note defects on the boundary at 4 and 5 o'clock). (b) Sobel gradient. (Original image courtesy of Mr. Pete Sites, Perceptics Corporation.)

Combining Spatial Enhancement Methods



- want to sharpen the original image and bring out more skeletal detail.
- problems: narrow dynamic range of gray level and high noise content makes the image difficult to enhance

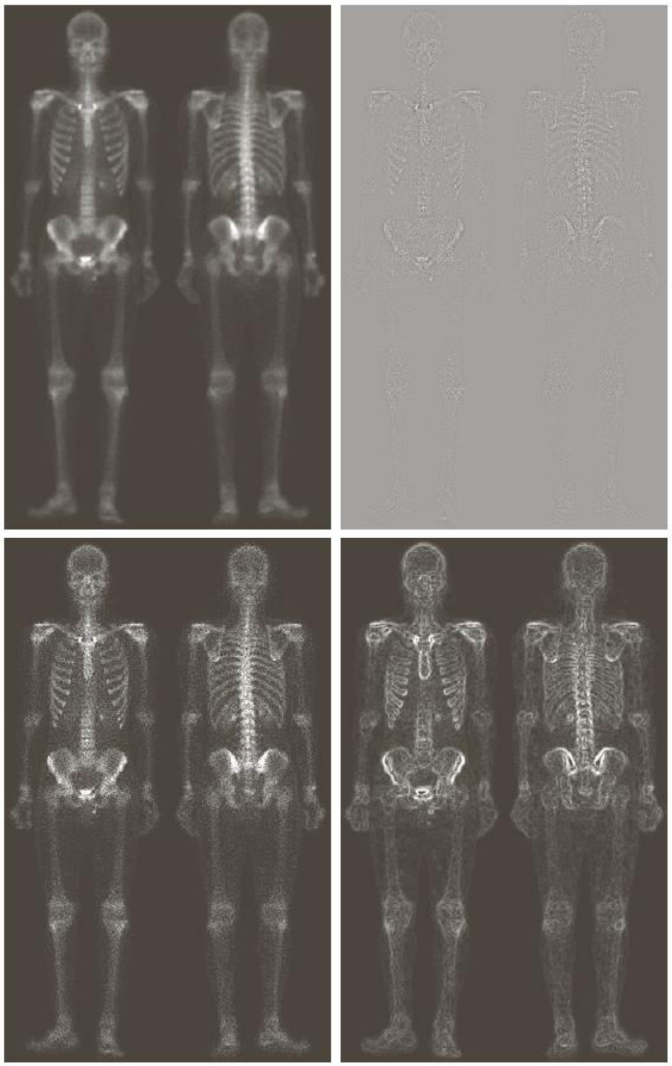
Example of Combining Spatial Enhancement Methods

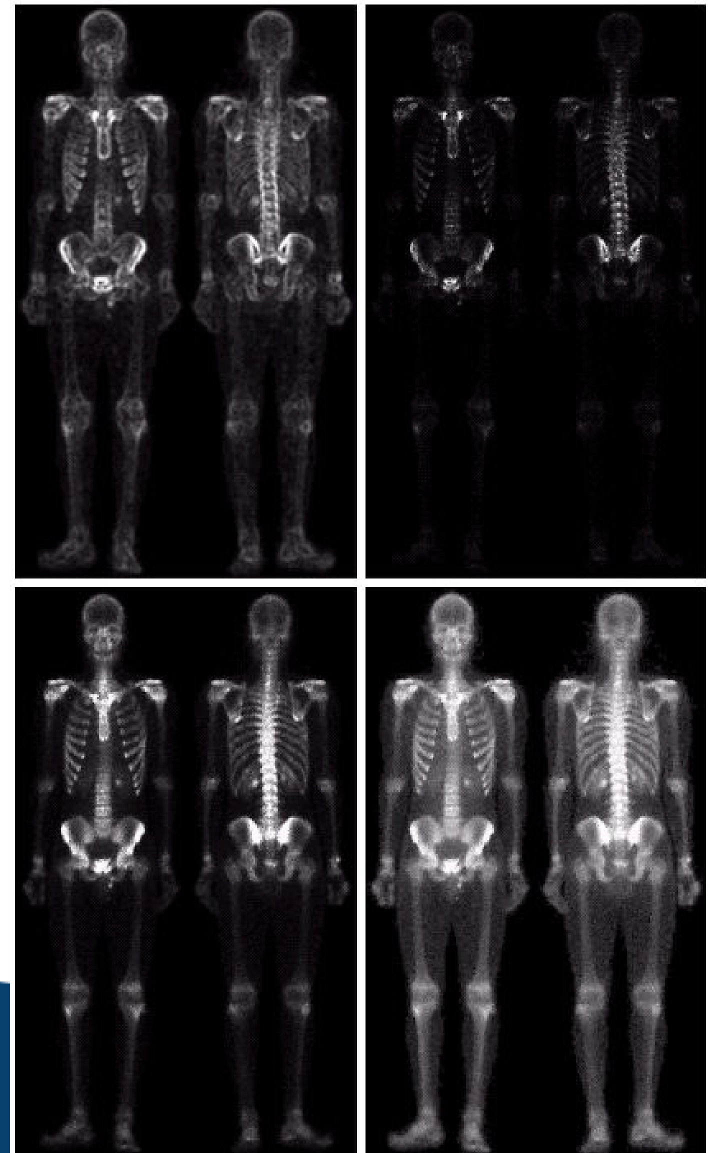
To solve :

1. Laplacian to highlight fine detail
2. gradient to enhance prominent edges
3. gray-level transformation to increase the dynamic range of gray levels

a b
c d

FIGURE 3.43
(a) Image of whole body bone scan.
(b) Laplacian of (a). (c) Sharpened image obtained by adding (a) and (b).
(d) Sobel gradient of (a).





e f
g h

FIGURE 3.46 *(Continued)*
(e) Sobel image smoothed with a 5 × 5 averaging filter. (f) Mask image formed by the product of (c) and (e).
(g) Sharpened image obtained by the sum of (a) and (f). (h) Final result obtained by applying a power-law transformation to (g). Compare (g) and (h) with (a). (Original image courtesy of G.E. Medical Systems.)

FREQUENCY DOMAIN FILTERING

Frequency Domain (F)

– Transform the image to its frequency representation

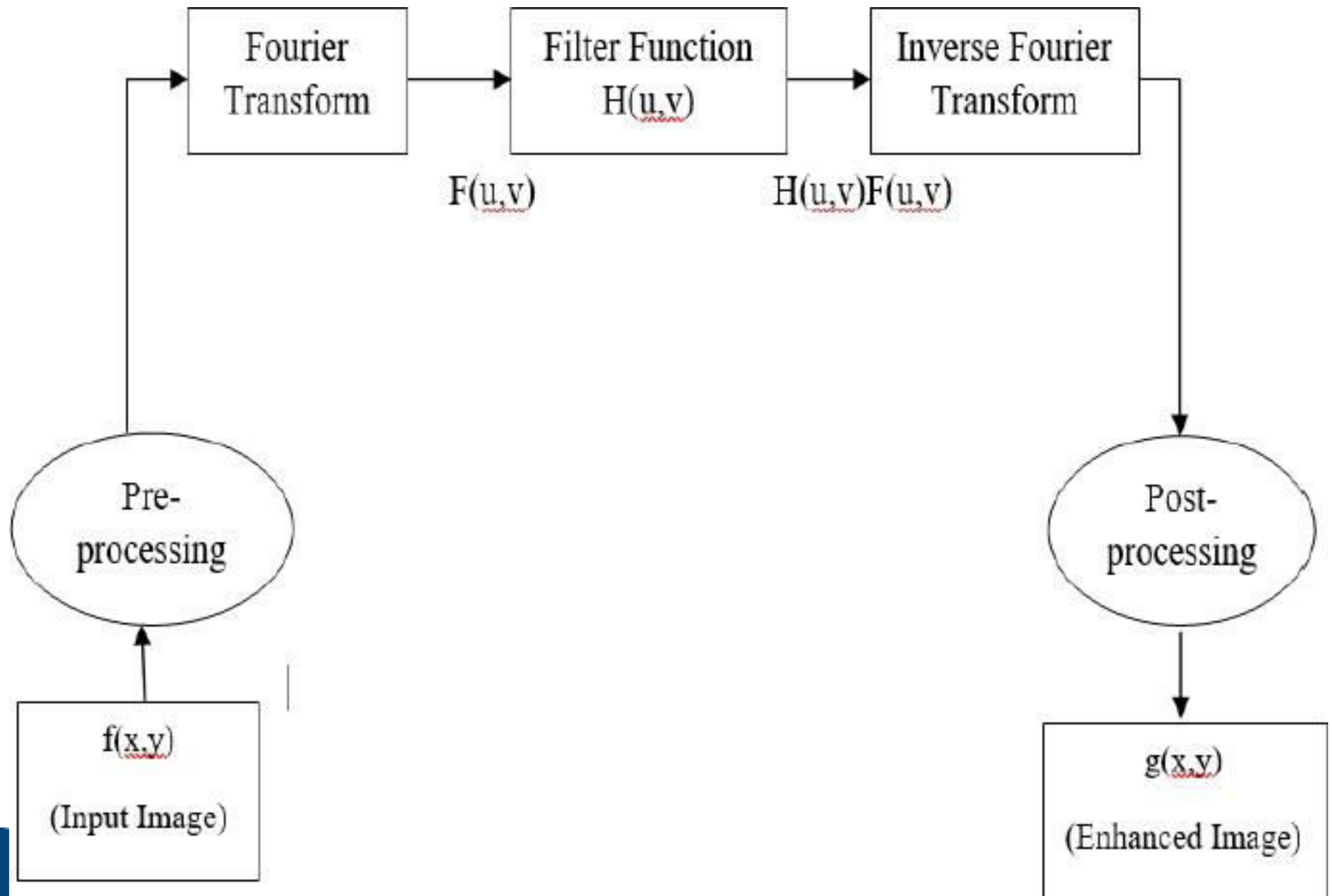
– Perform image processing

– Compute inverse transform back to the spatial domain

Frequencies in an Image

- Any spatial or temporal signal has an equivalent frequency representation
- What do frequencies mean in an image ?
  - – High frequencies correspond to pixel values that change rapidly across the image (e.g. text, texture, leaves, etc.)
  - – Strong low frequency components correspond to large scale features in the image (e.g. a single, homogenous object that dominates the image)
- We will investigate Fourier transformations to obtain frequency representations of an image

Frequency domain filtering steps



Basics of filtering in the frequency domain

1. Multiply the input image by $(-1)^{x+y}$ to center the transform to u = M/2 and v = N/2 (if M and N are even numbers, then the shifted coordinates will be integers)
2. Computer F(u,v), the DFT of the image from (1)
3. Multiply F(u,v) by a filter function H(u,v)
4. Compute the inverse DFT of the result in (3)
5. Obtain the real part of the result in (4)
6. Multiply the result in (5) by $(-1)^{x+y}$ to cancel the multiplication of the input image.

Highpass filters and lowpass filters

The relationship between blurring mask and derivative mask with a high pass filter and low pass filter can be defined simply as.

- **Blurring masks** are also called as **low pass filter/Smoothing**
- **Derivative masks** are also called as **high pass filter/Sharpening**

Blurring masks

Smoothing Filters

A blurring mask has the following properties.

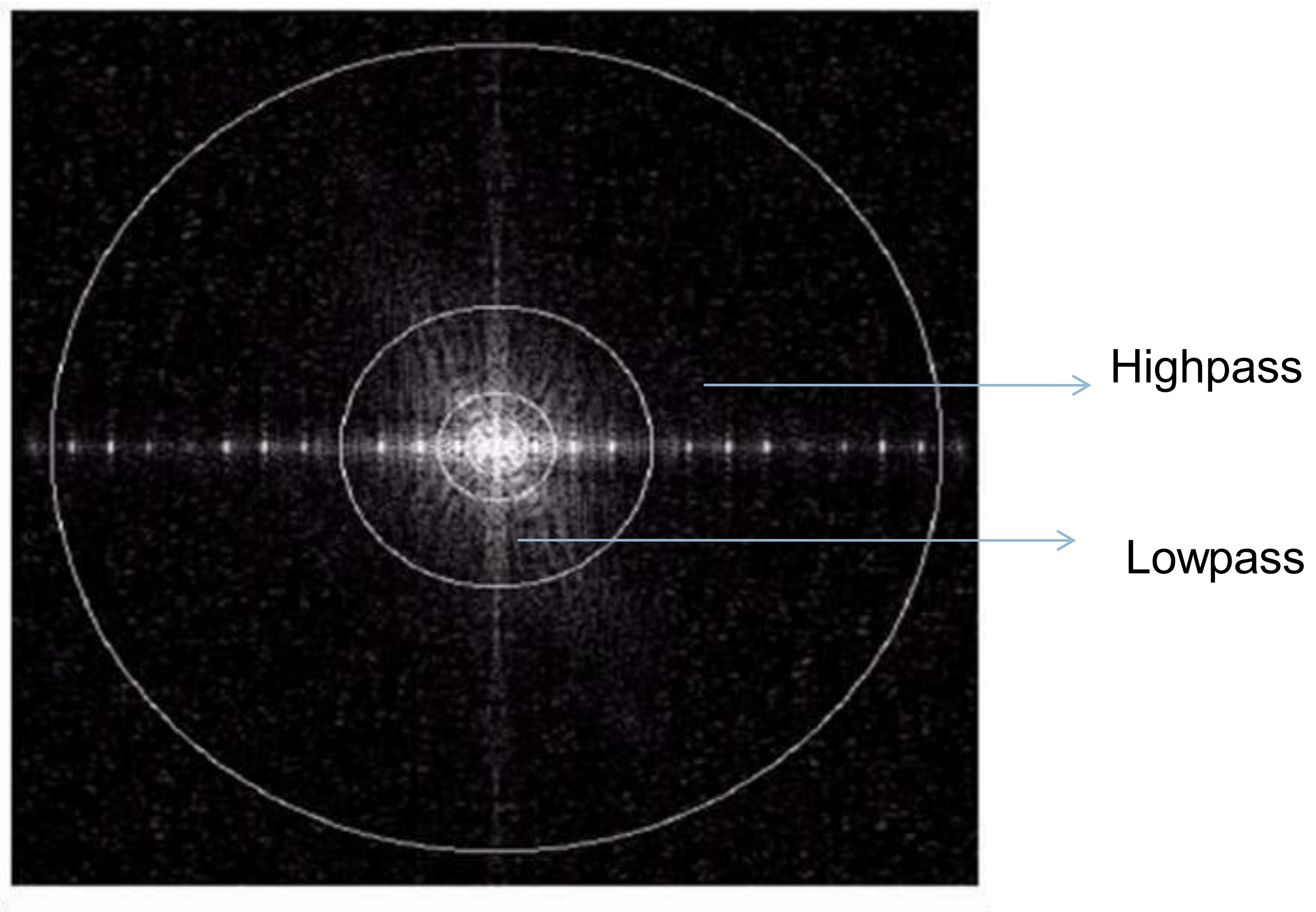
- All the values in blurring masks are positive
- The sum of all the values is equal to 1
- The edge content is reduced by using a blurring mask
- As the size of the mask grow, more smoothing effect will take place

Derivative masks

Sharpening Filters

A derivative mask has the following properties.

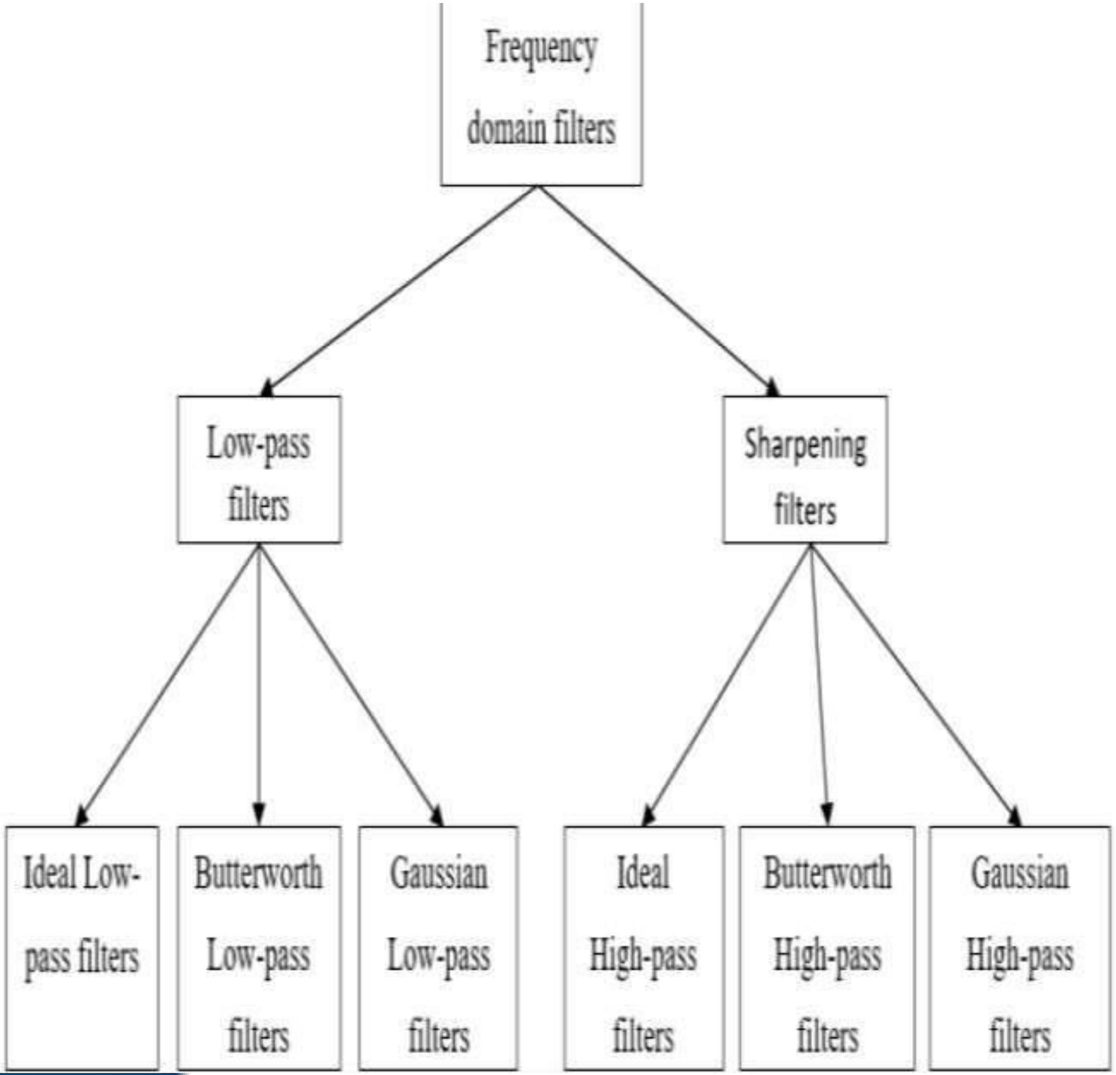
- A derivative mask have positive and as well as negative values
- The sum of all the values in a derivative mask is equal to zero
- The edge content is increased by a derivative mask
- As the size of the mask grows , more edge content is increased



Highpass
Lowpass

Filters used for smoothing and sharpening

S.No	Smoothing	Sharpening
1	Ideal Lowpass	Ideal Highpass
2	Butterworth Lowpass	Butterworth Highpass
3	Gaussian Lowpass	Gaussian Highpass


Frequency domain filters
Low-pass filters
Sharpening filters
Ideal Low-pass filters
Butterworth Low-pass filters
Gaussian Low-pass filters
Ideal High-pass filters
Butterworth High-pass filters
Gaussian High-pass filters

Smoothing

- Smoothing(blurring) is achieved in the frequency domain by high-frequency attenuation; that is, by lowpass filtering.
- Here, we consider 3 types of lowpass filters:
  - Ideal lowpass filters
  - Butterworth lowpass filters
  - Gaussian lowpass filters
- These three categories cover the range from very sharp(ideal), to very smooth(Gaussian) filtering.

Frequency domain filters

- The Butterworth filter has a parameter called the filter order.
- For high order values, the Butterworth filter approaches the ideal filter. For low order values, Butterworth filter is more like a Gaussian filter.
- Thus, the Butterworth filter may be viewed as providing a transition between two "extremes".

Lowpass filters

- The most basic of filtering operations is called “lowpass”.
- A lowpass filter is also called a “blurring” or smoothing filter.
- The simplest lowpass filter just calculates the average of a pixel and all of its eight immediate neighbours.
- Lowpass is also called as blurring mask

Ideal lowpass filters

- A 2-D low pass filter that passes without attenuation all frequencies within a circle of radius D_0 from the origin and "cuts off" all frequencies outside this circle is called an ideal lowpass filter(ILPF); it is specified by the function.

- $$H(u,v)\begin{cases}1 & if \;\; D(u,v) \le D_0 \\ 0 & if \;\; D(u,v) > D_0\end{cases}$$

- D_0 is a positive constant and D(u,v) is the distance between a point (u,v) in the frequency domain and the center of the frequency rectangle; that is,

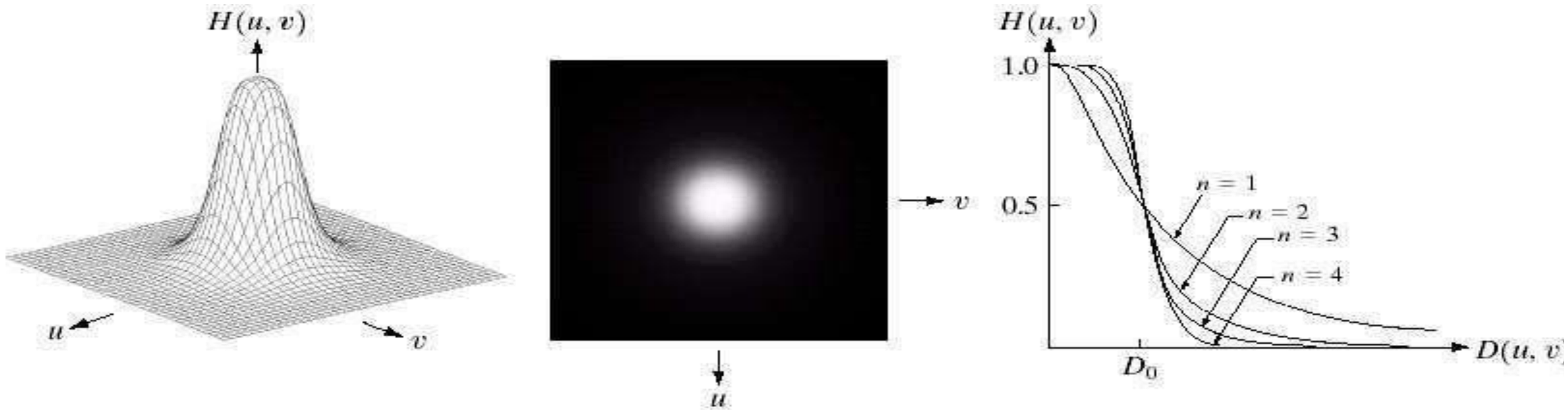
$$D(u, v) = [(u - P/2)^2 + (v - Q/2)^2]^{1/2}$$

Butterworth lowpass filters

- The Butterworth lowpass filter is a type of signal processing filterdesigned to have as flat a frequency response as possible in the passband.
- The transfer function of a Butterworth lowpass filter of order n with cutoff frequency at distance D_0 from the origin is defined as:

$$H(u,v) = \frac{1}{1 + [D(u,v) / D_0]^{2n}}$$

Butterworth lowpass filters



A) Perspective plot of an Butterworth lowpass filter transfer function

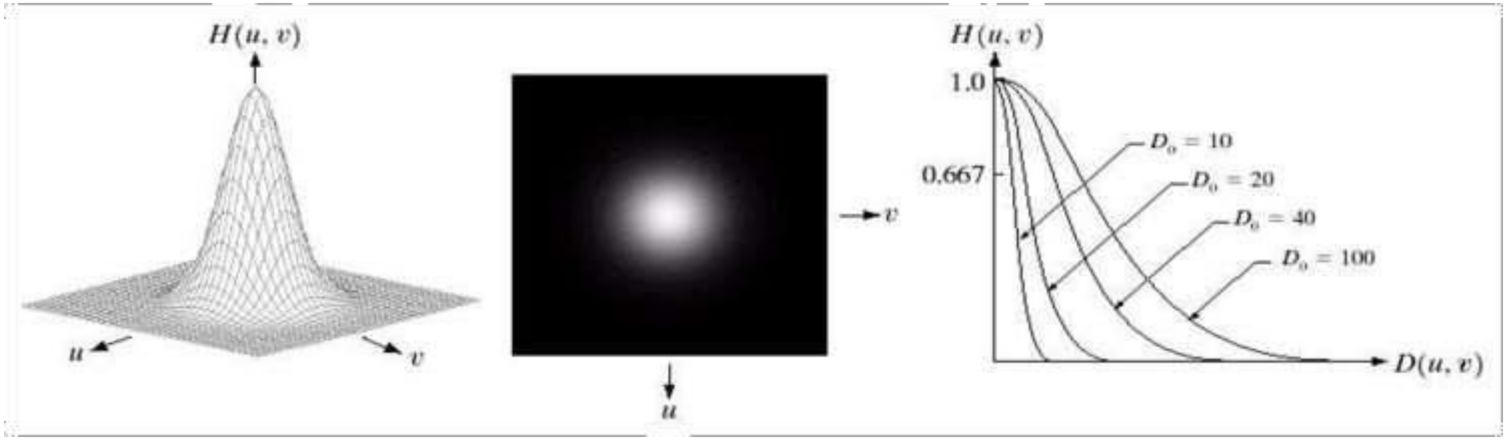
B) Filter displayed as an image

C) Filter radius cross section of orders 1 through 4

Gaussian lowpass filters

The transfer function of a Gaussian lowpass filter is defined as:

$$H(u,v) = e^{-D(u,v)/2\sigma}$$



A)Perspective plot of a GLPF transfer function B)Filter displayed as an image C)Filter radius cross section for various values of D_0

Gaussian lowpass filters

- Main advantage of a Gaussian LPF over a Butterworth LPF is that we are assured that there will be no ringing effects no matter what filter order we choose to work with.

Different Low pass Gaussian filters used to remove blemishes in the image

Sharpening

- Edges and fine detail characterized by sharp transitions in image intensity
- Such transitions contribute significantly to high frequency components of Fourier transform
- Intuitively, attenuating certain low frequency components and preserving high frequency components result in sharpening

Sharpening Filter Transfer Function

- Intended goal is to do the reverse operation of low-pass filters
  - When low-pass filer attenuates frequencies, high-pass filter passes them

$$H_{hp}(u,v) = 1 - H_{lp}(u,v)$$

  - When high-pass filter attenuates frequencies, low-pass filter passes them

High pass frequency components and Low pass frequency components

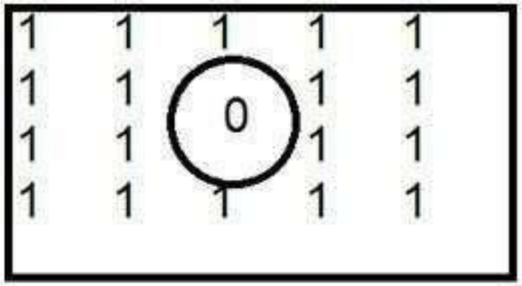
– High pass components pass components frequency and Low frequency

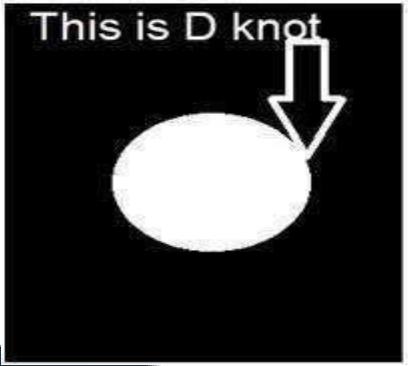
– the low pass frequency components denotes smooth regions.

Ideal High pass filters

This is a common example of high pass filter.

When 0 is placed inside, we get edges, which gives us a sketched image. An ideal low pass filter in frequency domain is given below.



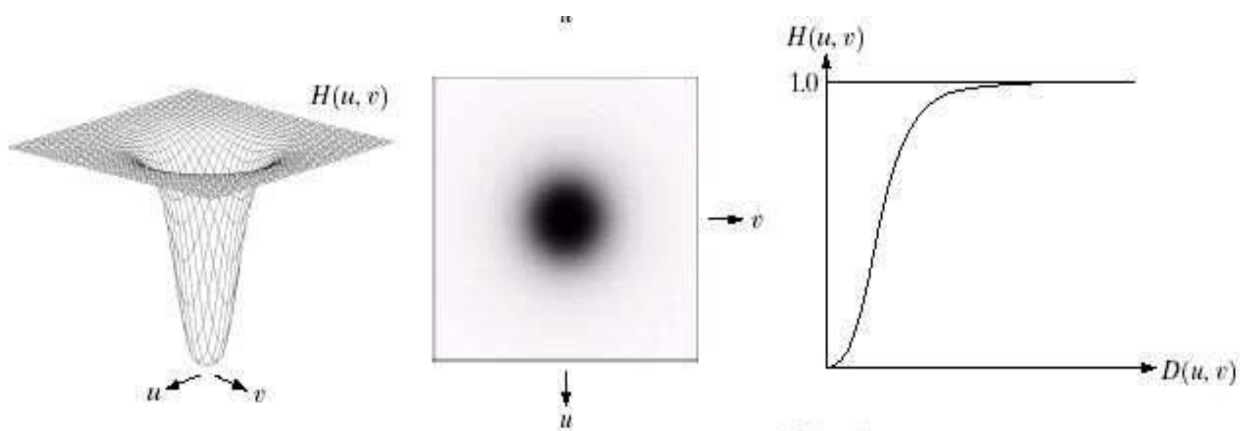


Butterworth High Pass Filters

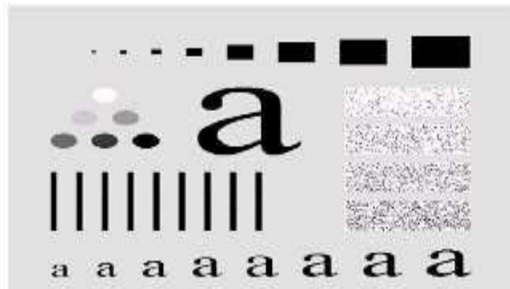
The Butterworth high pass filter is given as:

$$H(u,v) = \frac{1}{1 + [D_0 / D(u,v)]^{2n}}$$

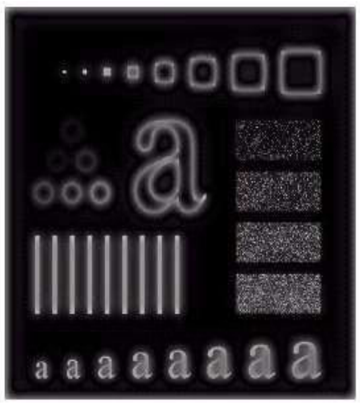
where n is the order and D_0 is the cut off distance as before

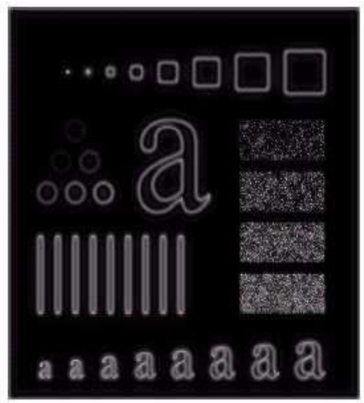


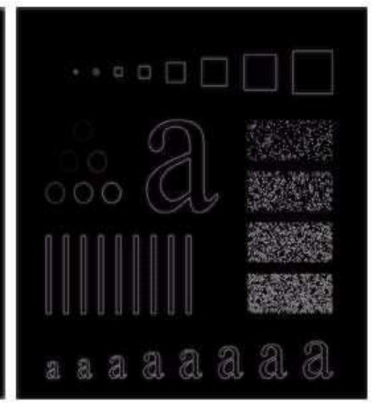
Butterworth High Pass Filters (cont…)



Results of Butterworth high pass filtering of order 2 with D_0= 15







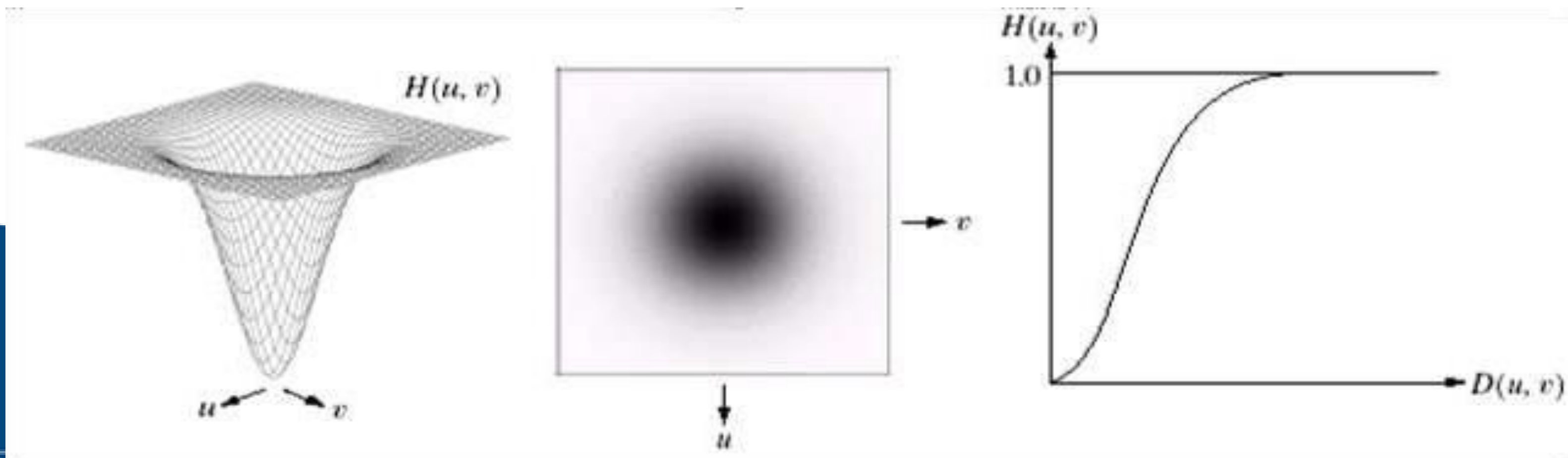
Results of Butterworth high pass filtering of order 2 with D_0= 80

Results of Butterworth high pass filtering of order 2 with D_0= 30

Gaussian high pass filter

– Gaussian high pass filter has the same concept as ideal high pass filter, but again the transition is more smooth as compared to the ideal one.

$$H(u,v) = 1 - e^{-D^2(u,v)/2D_0^2}$$



Sharpening Filters: Laplacian

The Laplacian is defined as:

$$\nabla^2 = \nabla \cdot \nabla = \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} \cdot \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$$

(dot product)

Approximate derivatives:

$$\frac{\partial^2 f}{\partial x^2} = f(i, j+1) - 2f(i, j) + f(i, j-1)$$

$$\frac{\partial^2 f}{\partial y^2} = f(i+1, j) - 2f(i, j) + f(i-1, j)$$

$$\nabla^2 f = -4f(i, j) + f(i, j+1) + f(i, j-1) + f(i+1, j) + f(i-1, j)$$

Sharpening Filters: Laplacian (cont'd)

Laplacian Mask

$$\begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 1 & -2 & 1 \\ \hline 0 & 0 & 0 \\ \hline \end{array} + \begin{array}{|c|c|c|}\hline 0 & 1 & 0 \\ \hline 0 & -2 & 0 \\ \hline 0 & 1 & 0 \\ \hline \end{array} = \begin{array}{|c|c|c|}\hline 0 & 1 & 0 \\ \hline 1 & -4 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array}$$

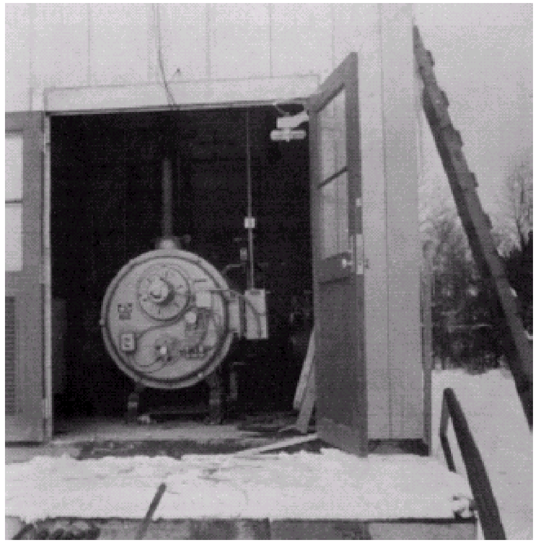
Conclusion

- The aim of image enhancement is to improve the information in images for human viewers, or to provide `better' input for other automated image processing techniques.
- There is no general theory for determining what is `good' image enhancement when it comes to human perception. If it looks good, it is good!

Homomorphic filtering

Homomorphic filtering

- Can be used to remove shading effects in an image (i.e., due to uneven illumination)
  - Enhance high frequencies.
  - Attenuate low frequencies but preserve fine detail.



Homomorphic Filtering (cont'd)

- Consider the following model of image formation:

$$f(x, y) = i(x, y)\, r(x, y)$$

i(x,y): illumination component
r(x,y): reflection component

- Illumination i(x,y) varies slowly and affects low frequencies mostly.

- Reflection r(x,y) varies faster and affects high frequencies mostly.

How are frequencies mixed together?

- Low frequencies from **i(x,y)** and high frequencies from **r(x,y)** are convolved together.

$$f(x, y) = i(x, y)\, r(x, y) \Rightarrow F(u,v) = I(u,v) * R(u,v)$$

- When applying filtering, it is difficult to handle low/high frequencies separately.

$$F(u,v)H(u,v) = [I(u,v) * R(u,v)]H(u,v)$$

Can we separate them?

- Consider the ln(f(x,y)) - why?

$$f(x,y) = i(x,y)\, r(x,y)$$

$$\Downarrow$$

$$ln(f(x,y)) = ln(i(x,y)) + ln(r(x,y))$$

$$\Downarrow$$

$$F(ln(f(x,y))) = F(ln(i(x,y))) + F(ln(r(x,y)))$$

(frequencies have been de-convolved)

Steps of Homomorphic Filtering

(1) Take $ln(f(x, y)) = ln(i(x, y)) + ln(r(x, y))$

(2) Apply FT: $F(ln(f(x, y))) = F(ln(i(x, y))) + F(ln(r(x, y)))$

or $Z(u, v) = Illum(u, v) + Refl(u, v)$

(3) Apply H(u,v)

$$Z(u, v)H(u, v) = Illum(u, v)H(u, v) + Refl(u, v)H(u, v)$$

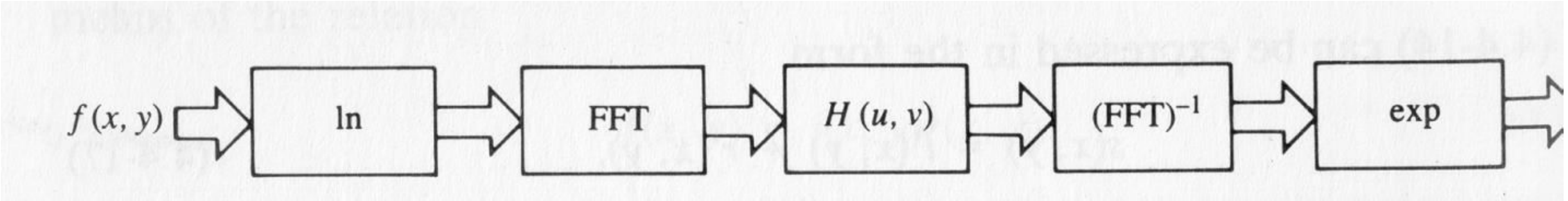
Steps of Homomorphic Filtering (cont'd)

(4) Take Inverse FT:

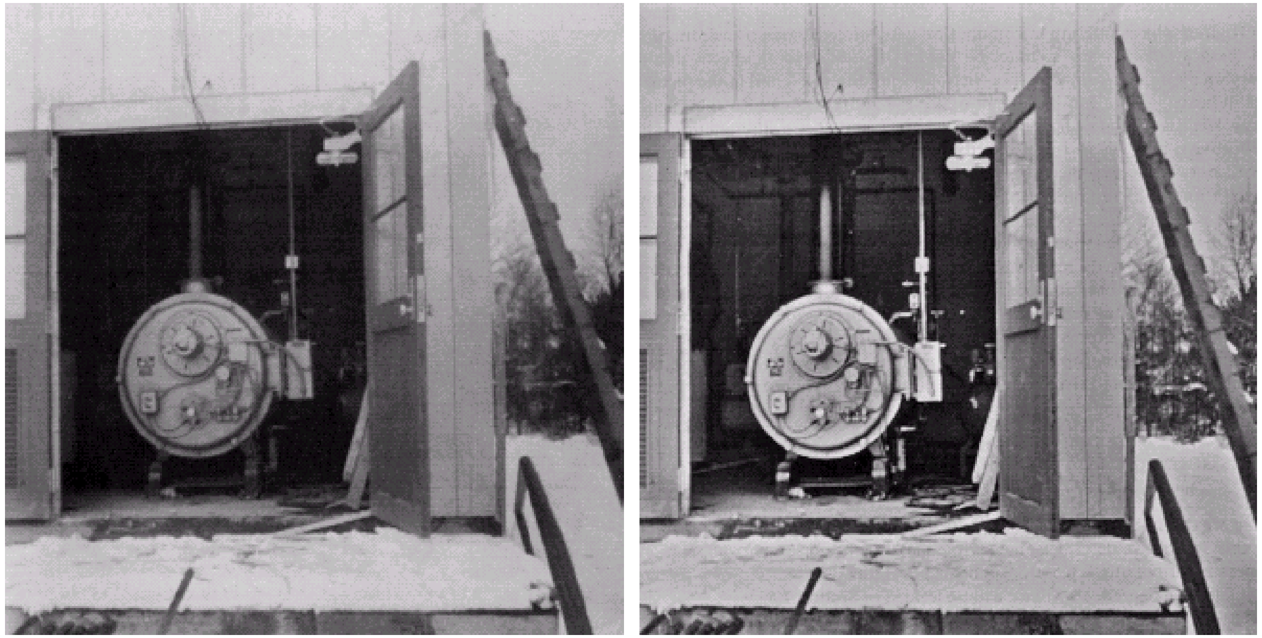
$$F^{-1}(Z(u,v)H(u,v)) = F^{-1}(Illum(u,v)H(u,v)) + F^{-1}(Refl(u,v)H(u,v))$$

or $$s(x,y) = i'(x,y) + r'(x,y)$$

(5) Take exp( ) $e^{s(x,y)} = e^{i'(x,y)} e^{r'(x,y)}$ or $g(x,y) = i_0(x,y) r_0(x,y)$



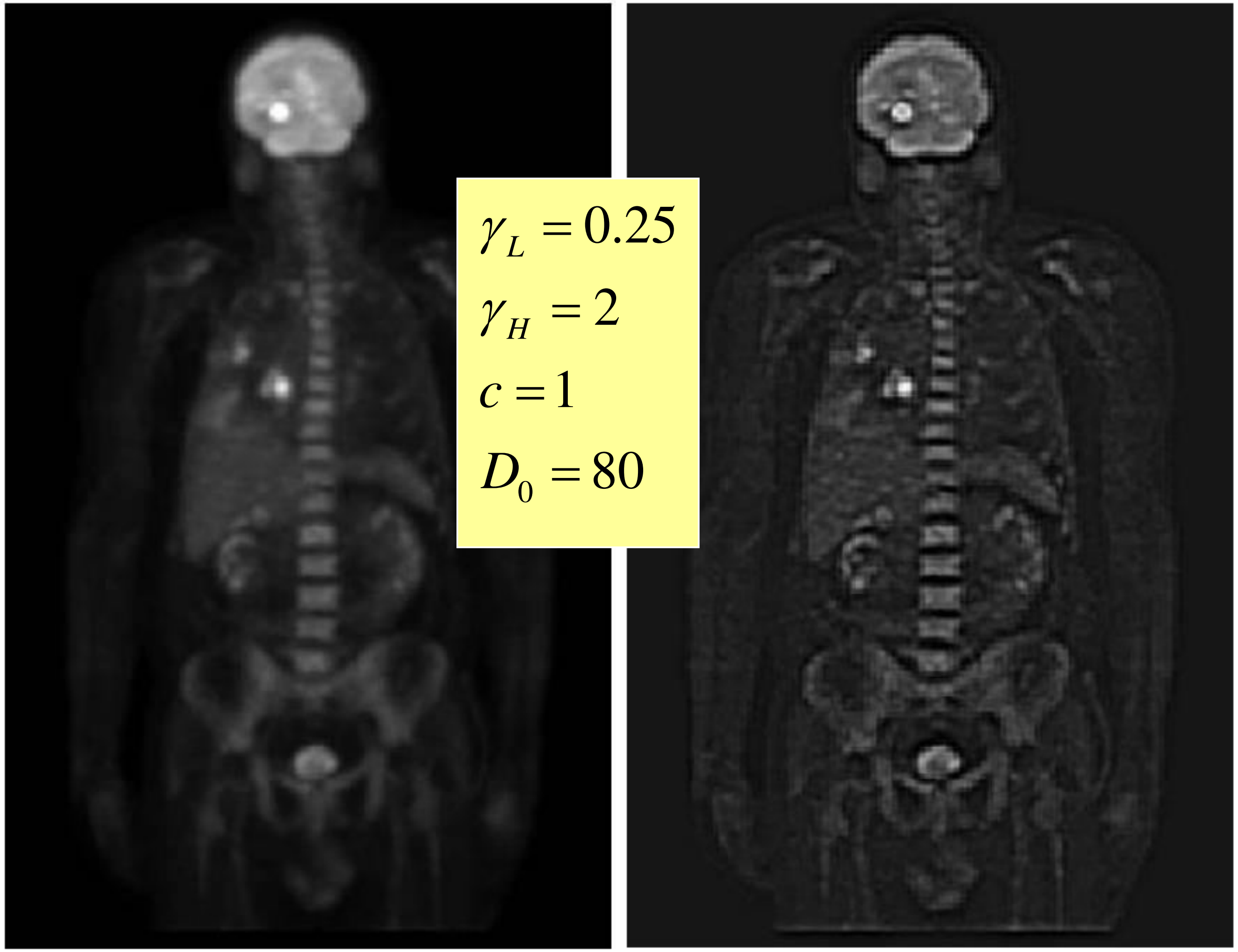
Homomorphic Filtering: Example 1



a b

FIGURE 4.33 (a) Original image. (b) Image processed by homomorphic filtering (note details inside shelter). (Stockham.)

Homomorphic Filtering: Example 2



$$\gamma_L = 0.25$$
$$\gamma_H = 2$$
$$c = 1$$
$$D_0 = 80$$

a b

FIGURE 4.62 (a) Full body PET scan. (b) Image enhanced using homomorphic filtering. (Original image courtesy of Dr. Michael E. Casey, CTI PET Systems.)