

## Unit-6

### **Spatial domain: Enhancement in the case of a single image**

#### **Spatial masks**

Many image enhancement techniques are based on spatial operations performed on local neighbourhoods of input pixels.

The image is usually convolved with a finite impulse response filter called spatial mask. The use of spatial masks on a digital image is called spatial filtering.

The mechanics of spatial filtering are illustrated in Fig. 3.32. 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. For linear spatial filtering, the response is given by a sum of products of the filter coefficients and the corresponding image pixels in the area spanned by the filter mask. For the 3\*3 mask shown in Fig. 3.32, the result (or response),  $R$ , of linear filtering with the filter

Mask at a point  $(x, y)$  in the image is

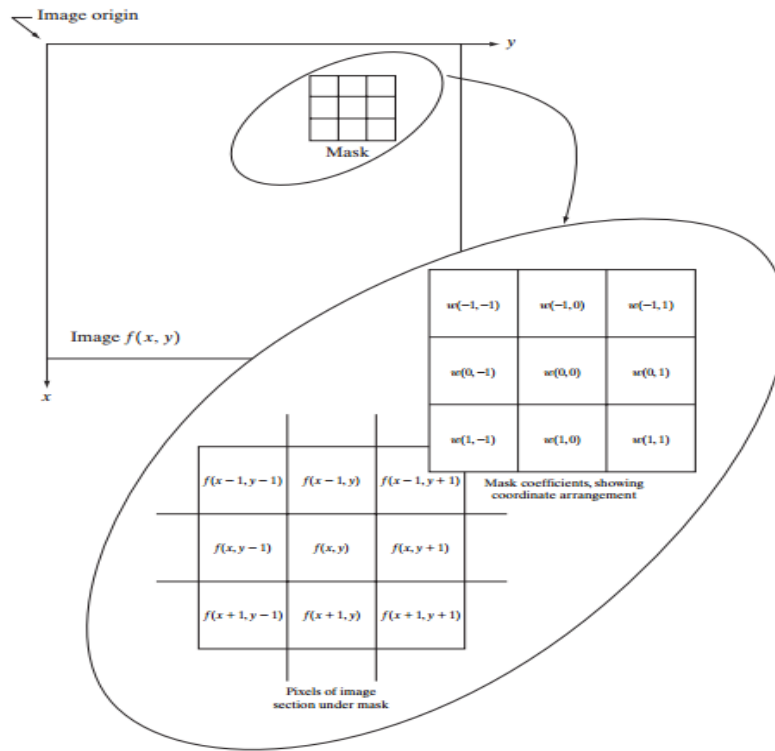
$$R = \omega(-1, -1)f(x - 1, y - 1) + \omega(-1, 0)f(x - 1, y) + \dots \dots + \omega(0, 0)f(x, y) \\ + \dots \dots + \omega(1, 0)f(x + 1, y) + \omega(1, 1)f(x + 1, y + 1)$$

which we see is the sum of products of the mask coefficients with the corresponding pixels directly under the mask. Note in particular that the coefficient  $w(0, 0)$  coincides with image value  $f(x, y)$ , indicating that the mask is centered at  $(x, y)$  when the computation of the sum of products takes place.

In general, linear filtering of an image  $f$  of size of size  $M \times N$  with a filter mask of size  $m \times n$  is given by the expression:

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

Where  $a=(m-1)/2$  and  $b=(n-1)/2$



**FIGURE 3.32** The mechanics of spatial filtering. The magnified drawing shows a  $3 \times 3$  mask and the image section directly under it; the image section is shown displaced out from under the mask for ease of readability.

### 3.6 Smoothing Spatial Filters

Smoothing filters are used for blurring and for noise reduction. Blurring is used in preprocessing steps, such as removal of small details from an image prior to (large) object extraction, and bridging of small gaps in lines or curves. Noise reduction can be accomplished by blurring with a linear filter and also by nonlinear filtering

#### 3.6.1 Smoothing Linear Filters

The output (response) of a smoothing, linear spatial filter is simply the average of the pixels contained in the neighborhood of the filter mask. These filters sometimes are called averaging filters. For reasons explained in Chapter 4, they also are referred to a lowpass filters.

The idea behind smoothing filters is straightforward. By replacing the value of every pixel in an image by the average of the gray levels in the neighborhood defined by the filter mask, this process results in an image with reduced “sharp” transitions in gray levels. Because random noise typically consists of sharp transitions in gray levels, the most obvious application of smoothing is noise reduction. However, edges (which almost always are desirable features of an image) also are characterized by sharp transitions in gray levels, so averaging filters have the undesirable side effect that they blur edges.

$$\frac{1}{9} \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \quad \frac{1}{16} \times \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

**Fig: Two 3x3 smoothing (averaging) filter masks. The constant multiplier in front of each mask is equal to the sum of the values of its coefficients, as is required to compute the average**

Figure shows two 3\*3 smoothing filters. Use of the first filter yields the standard average of the pixels under the mask. This can best be seen by substituting the coefficients of the mask into Eq

$$R = \frac{1}{9} \sum_{i=1}^9 z_i$$

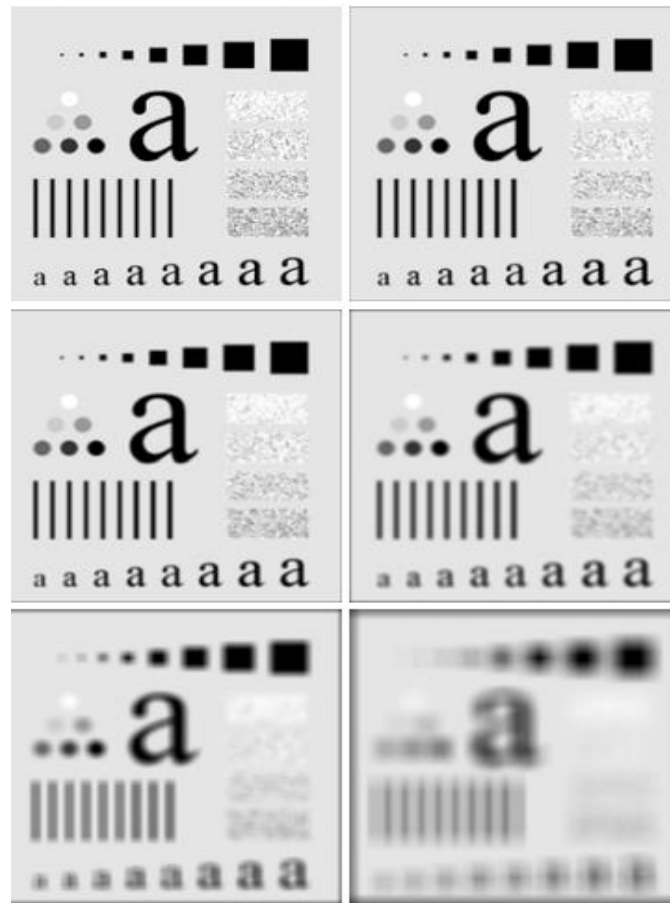
which is the average of the gray levels of the pixels in the 3\*3 neighborhood defined by the mask. A spatial averaging filter in which all coefficients are equal is sometimes called a box filter

The second mask shown in Fig. is a little more interesting. This mask yields a so-called weighted average, terminology used to indicate that pixels are multiplied by different coefficients, thus giving more importance (weight) to some pixels at the expense of others. In the mask shown in Fig.(b) the pixel at the center of the mask is multiplied by a higher value than any other, thus giving this pixel more importance in the calculation of the average. The other pixels are inversely weighted as a function of their distance from the center of the mask. The diagonal terms are further away from the center than the orthogonal neighbors (by a factor of ) and, thus, are weighed less than these immediate neighbors of the center pixel. The basic strategy behind weighing 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.

The general implementation for filtering an  $M \times N$  image with a weighted averaging filter of size  $m \times n$  ( $m$  and  $n$  odd) is given by the expression

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



**FIGURE 3.35** (a) Original image, of size  $500 \times 500$  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  $50 \times 120$  pixels.

### 3.6.2 Order Statistics Filter (Eg: Median Filter)

Median filter replaces the pixel at the center of the filter with the median value of the pixels falling beneath the mask. Median filter does not blur the image but it rounds the corners.

Median filter, which, as its name implies, 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). Median filters are particularly effective in the presence of impulse noise, also called salt-and-pepper noise because of its appearance as white and black dots superimposed on an image

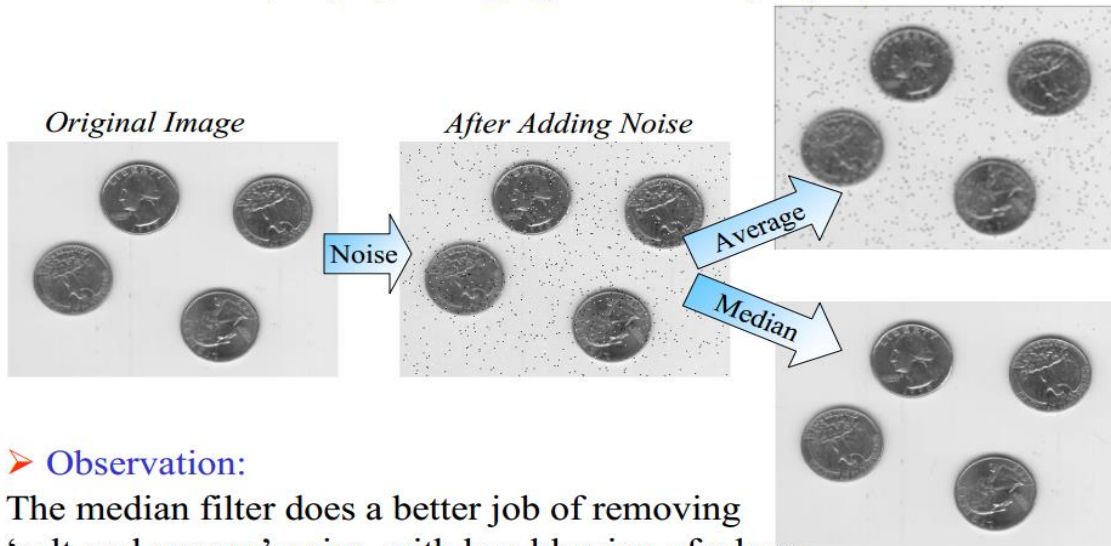
Consider 3x3 original image

<b>91</b>	<b>55</b>	<b>90</b>
<b>77</b>	<b>68</b>	<b>95</b>
<b>115</b>	<b>151</b>	<b>210</b>

Median filtering using the full 3x3 neighborhood:

We sort the original image values on the full 3x3 window: 55 , 68 , 77 , 90 , **91** , 95 , 115 , 151 , 210

median value = 91. We replace the center pixel(i.e 68) by 91. Other pixels are unchanged



➤ **Observation:**  
 The median filter does a better job of removing ‘salt-and-pepper’ noise, with less blurring of edges.

**Questionnaire**

1. What is the value of the yellow box after median filtering?

1	1	1
9	8	7
3	3	3

Answers: a) 7    b) 1    c) 3    d) 9

2. What are the values of yellow boxes after median filtering is done?

1	2	2	2	3
2	5	2	4	3
2	2	2	2	3
4	3	7	5	3
3	3	3	1	1

1	2	2	2	3
2	5	5	4	3
2	7	7	3	3
4	3	7	3	3
3	3	3	1	1

Answers: a)

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

(b)

### Sharpening Spatial Filters (High Pass)

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

High pass filters let the high frequency content of the image pass through the filter and block the low frequency content.

High pass filters can be modelled by first order derivative as:

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

A second order derivative can also be used for extracting high frequency data

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

a b  
c  
**FIGURE 3.38**  
(a) A simple image. (b) 1-D horizontal gray-level profile along the center of the image and including the isolated noise point. (c) Simplified profile (the points are joined by dashed lines to simplify interpretation).

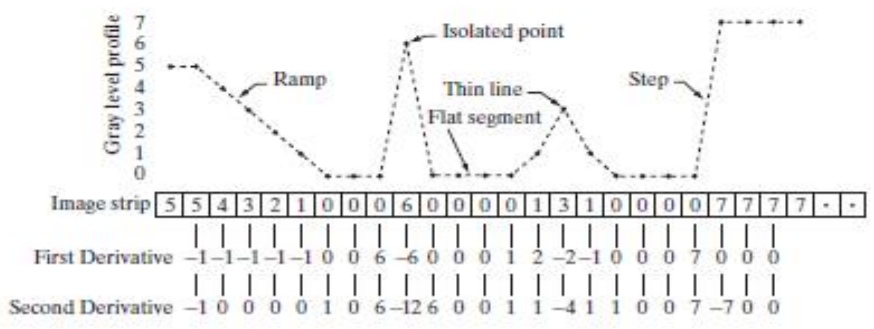
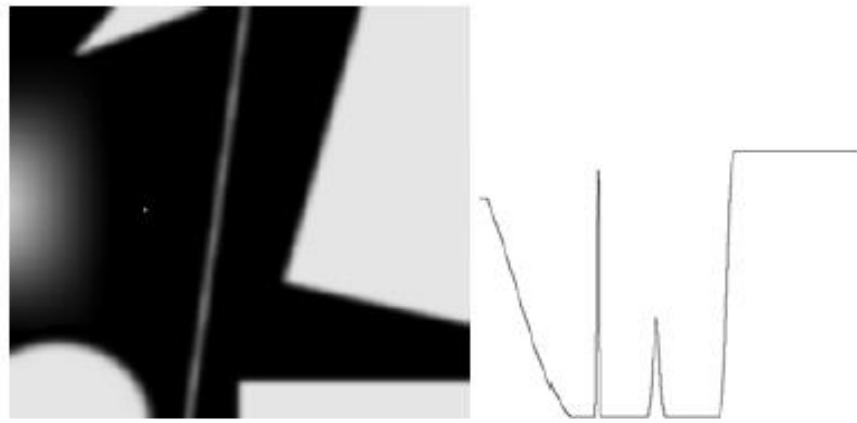


Figure 3.38(a) shows a simple image that contains various solid objects, a line, and a single noise point. Figure 3.38(b) shows a horizontal gray-level profile (scan line) of the image along the center and including the noise point. Figure 3.38(c) shows a simplification of the profile, with just enough numbers to make it possible for us to analyse how the first- and second-order derivatives behave as they encounter a noise point, a line, and then the edge of an object.

Let us consider the properties of the first and second derivatives as we traverse the profile from left to right. First, we note that the first-order derivative is nonzero along the entire ramp, while the second-order derivative is nonzero only at the onset and end of the ramp. Because edges in an image resemble this type of transition, we conclude that first-order derivatives produce “thick” edges and second-order derivatives, much finer ones. Next we encounter the isolated noise point. Here, the response at and around the point is much stronger for the second- than for the first-order derivative. Of course, this is not unexpected. A second-order derivative is much more aggressive than a first-order derivative in enhancing sharp changes. Thus, we can expect a second-order derivative to enhance fine detail (including noise) much more than a first-order derivative. The thin line is a fine detail, and we see essentially the same difference between the two derivatives. If the maximum gray level of the line had been the same as the isolated point, the response of the second derivative would have been stronger for the latter. Finally, in this case, the response of the two derivatives is the same at the gray-level step



(in most cases when the transition into a step is not from zero, the second derivative will be weaker). We also note that the second derivative has a transition from positive back to negative. In an image, this shows as a thin double line. This “double-edge” effect is an issue that will be important in Chapter 10, where we use derivatives for edge detection. It is of interest also to note that if the gray level of the thin line had been the same as the step, the response of the second derivative would have been stronger for the line than for the step.

In summary, comparing the response between first- and second-order derivatives, we arrive at the following conclusions. (1) First-order derivatives generally produce thicker edges in an image. (2) Second-order derivatives have a stronger response to fine detail, such as thin lines and isolated points. (3) First order derivatives generally have a stronger response to a gray-level step. (4) Second- order derivatives produce a double response at step changes in gray level. We also note of second-order derivatives that, for similar changes in gray-level values in an image, their response is stronger to a line than to a step and to a point than to a line.

## Use of Second Derivatives for Enhancement–The Laplacian

### Development of the method

It can be shown that the simplest isotropic derivative operator is the *Laplacian*, which, for a function (image)  $f(x, y)$  of two variables, is defined as

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}.$$

Because derivatives of any order are linear operations, the Laplacian is a linear operator.

Taking into account that we now have two variables, we use the following notation for the partial second-order derivative in the  $x$ -direction:

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

and, similarly in the  $y$ -direction, as

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

The digital implementation of the two-dimensional Laplacian in Eq. (3.7-1) is obtained by summing these two components:

$$\nabla^2 f = [f(x + 1, y) + f(x - 1, y) + f(x, y + 1) + f(x, y - 1)] - 4f(x, y). \quad (3.7-4)$$

This equation can be implemented using the mask shown in figure below

0	1	0	1	1	1
1	-4	1	1	-8	1
0	1	0	1	1	1
0	-1	0	-1	-1	-1
-1	4	-1	-1	8	-1
0	-1	0	-1	-1	-1

a b  
c d

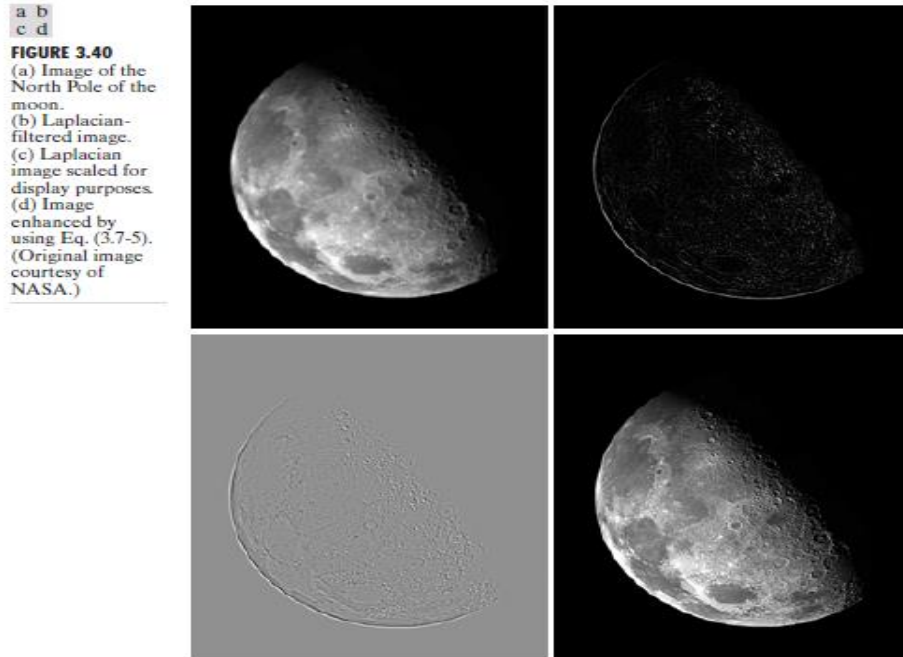
**FIGURE 3.39**  
(a) Filter mask used to implement the digital Laplacian, as defined in Eq. (3.7-4).  
(b) Mask used to implement an extension of this equation that includes the diagonal neighbors. (c) and (d) Two other implementations of the Laplacian.

The diagonal directions can be incorporated in the definition of the digital Laplacian by adding two more terms to Eq. (3.7-4), one for each of the two diagonal directions. Since each diagonal term also contains a  $-2f(x, y)$  term, the total subtracted from the difference terms now would be  $-8f(x, y)$ . The mask used to implement this new definition is shown in Fig. 3.39(b).

Because the Laplacian is a derivative operator, its use highlights gray-level discontinuities in an image and deemphasizes regions with slowly varying gray levels. This will tend to produce images that have grayish edge lines and other discontinuities, all superimposed on a dark, featureless background. Background features can be “recovered” while still preserving the sharpening effect of the Laplacian operation simply by adding the original and Laplacian images. If the definition used has a negative center coefficient, then we *subtract*, rather than add, the Laplacian image to obtain a sharpened result. Thus, the basic way in which we use the Laplacian for image enhancement is as follows:

$$g(x, y) = \begin{cases} f(x, y) - \nabla^2 f(x, y) & \text{if the center coefficient of the Laplacian mask is negative} \\ f(x, y) + \nabla^2 f(x, y) & \text{if the center coefficient of the Laplacian mask is positive.} \end{cases}$$

Figure 3.40(a) shows an image of the North Pole of the moon. Figure 3.40(b) shows the result of filtering this image with the Laplacian mask in Fig. 3.39(b). The image shown in Fig. 3.40(c) was scaled in the manner just described for display purposes. Finally, Fig. 3.40(d) shows the result obtained using Eq. (3.7-5). The detail in this image is unmistakably clearer and sharper than in the original image.



### Simplifications

In the previous example, we implemented Eq. (3.7-5) by first computing the Laplacian-filtered image and then subtracting it from the original image. The coefficients of the single mask are easily obtained by substituting Eq. (3.7-4) for in the first line of Eq. (3.7-5):

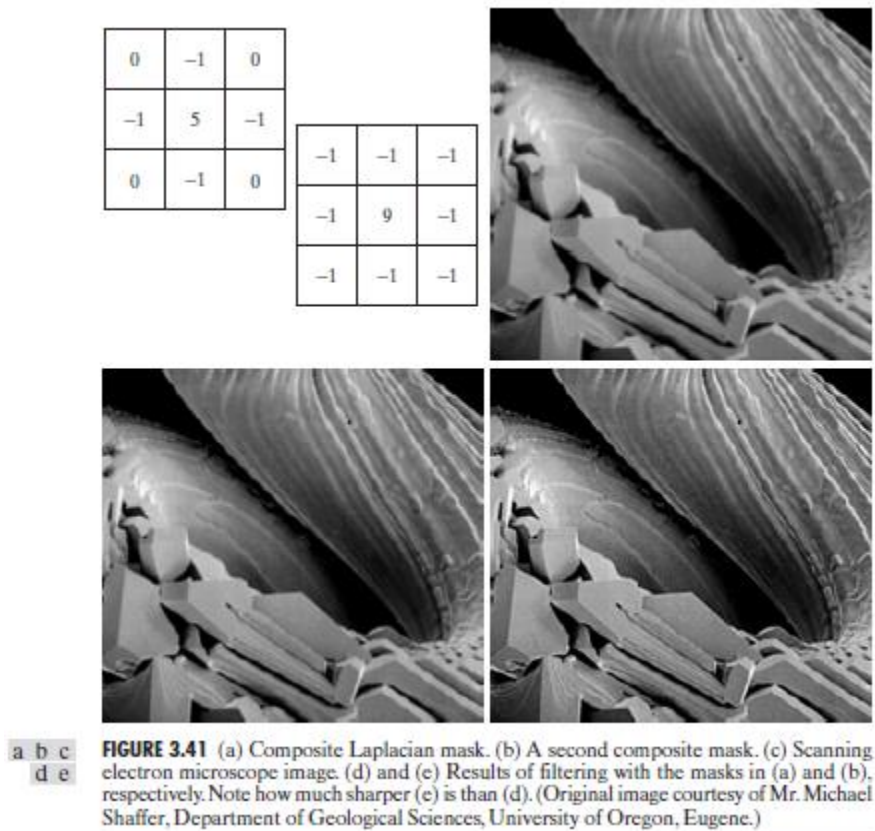
$$\begin{aligned}
 g(x, y) &= f(x, y) - [f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) + 4f(x, y)] \\
 &= 5f(x, y) - [f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1)]
 \end{aligned}$$

This equation can be implemented using the mask shown in Fig. 3.41(a). The mask shown in Fig. 3.41(b) would be used if the diagonal neighbors also were included in the calculation of the Laplacian. Identical masks would have resulted if we had substituted the negative of Eq. (3.7-4) into the second line of Eq. (3.7-5).

The results obtainable with the mask containing the diagonal terms usually are a little sharper than those obtained with the more basic mask of Fig. 3.41(a). This property is illustrated by the Laplacian-filtered images shown in

Figs. 3.41(d) and (e), which were obtained by using the masks in Figs. 3.41(a) and (b), respectively. By comparing the filtered images with the original image shown in Fig. 3.41(c), we note that both masks produced effective enhancement, but the result using the mask in Fig. 3.41(b) is visibly sharper. Figure 3.41(c) is a scanning

electron microscope (SEM) image of a tungsten filament following thermal failure; the magnification is approximately 250\*.)



### Unsharp masking and high-boost filtering

A process used for many years in the publishing industry to sharpen images consists of subtracting a blurred version of an image from the image itself. This process, called *unsharp masking*, is expressed as

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

where  $f_s(x, y)$  denotes the sharpened image obtained by unsharp masking, and  $\bar{f}(x, y)$  is a blurred version of  $f(x, y)$ .

A slight further generalization of unsharp masking is called *high-boost filtering*. A high-boost filtered image,  $f_{hb}$ , is defined at any point  $(x, y)$  as

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

0	-1	0	-1	-1	-1
-1	$A + 4$	-1	-1	$A + 8$	-1
0	-1	0	-1	-1	-1

a b

**FIGURE 3.42** The high-boost filtering technique can be implemented with either one of these masks, with  $A \geq 1$ .

where  $A-1$  and, as before,  $\bar{f}$  is a blurred version of  $f$ . This equation may be written as

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

By using Eq. (3.7-7), we obtain

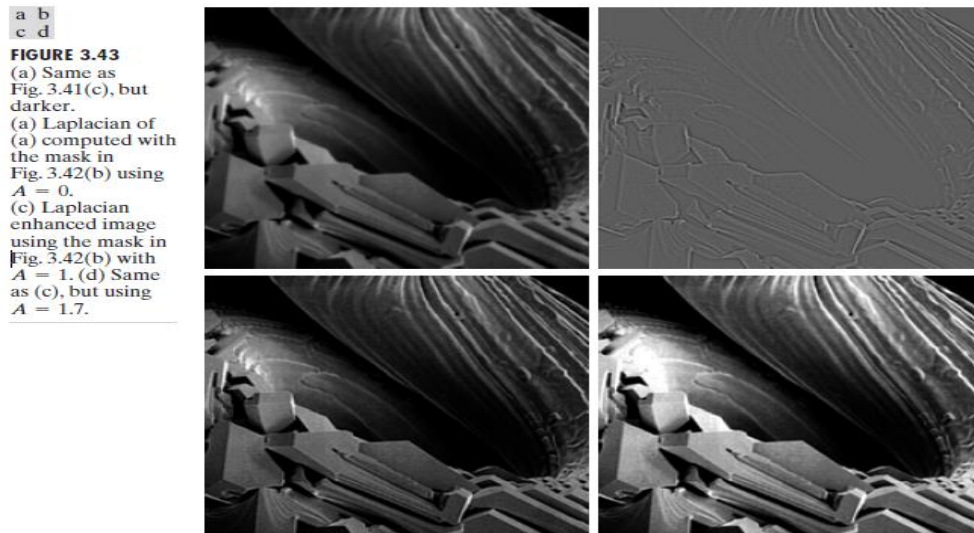
$$f_{hb}(x, y) = (A - 1)f(x, y) + f_s(x, y)$$

as the expression for computing a high-boost-filtered image. Equation (3.7-10) is applicable in general and does not state explicitly how the sharp image is obtained. If we elect to use the Laplacian, then we know that  $f_s(x, y)$  can be obtained using Eq. (3.7-5). In this case, Eq. (3.7-10) becomes

$$f_{hb} = \begin{cases} Af(x, y) - \nabla^2 f(x, y) & \text{if the center coefficient of the} \\ & \text{Laplacian mask is negative} \\ Af(x, y) + \nabla^2 f(x, y) & \text{if the center coefficient of the} \\ & \text{Laplacian mask is positive.} \end{cases}$$

High-boost filtering can be implemented with one pass using either of the two masks shown in Fig. 3.42. Note that, when  $A=1$ , high-boost filtering becomes “standard” Laplacian sharpening. As the value of  $A$  increases past 1, the contribution of the sharpening process becomes less and less important. Eventually, if  $A$  is large enough, the high-boost image will be approximately equal to the original image multiplied by a constant.

One of the principal applications of boost filtering is when the input image is darker than desired. By varying the boost coefficient, it generally is possible to obtain an overall increase in average gray level of the image, thus helping to brighten the final result. Figure 3.43 shows such an application. Part (a) of this figure is a darker version of the image in Fig. 3.41(c). Figure 3.43(b) shows the Laplacian computed using the mask in Fig. 3.42(b), with  $A=0$ . Figure 3.43(c) was obtained using the mask in Fig. 3.42(b) with  $A=1$ . As expected, the image has been sharpened, but it is still as dark as the original. Finally, Fig. 3.43(d) shows the result of using  $A=1.7$ . This is a much more acceptable result, in which the average gray level has increased, thus making the image lighter and more natural.



**FIGURE 3.43**  
 (a) Same as Fig. 3.41(c), but darker.  
 (b) 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  $A = 1.7$ .

### Use of First Derivatives for Enhancement—The Gradient

First derivatives in image processing are implemented using the magnitude of the gradient. For a function  $f(x, y)$ , the gradient of  $f$  at coordinates  $(x, y)$  is defined as the two-dimensional column vector

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}.$$

The magnitude of this vector is given by

$$\begin{aligned} \nabla f &= \text{mag}(\nabla f) \\ &= [G_x^2 + G_y^2]^{1/2} \\ &= \left[ \left( \frac{\partial f}{\partial x} \right)^2 + \left( \frac{\partial f}{\partial y} \right)^2 \right]^{1/2}. \end{aligned}$$

The components of the gradient vector itself are linear operators, but the magnitude of this vector obviously is not because of the squaring and square root operation. Although it is not strictly correct, the magnitude of the gradient vector often is referred to as the *gradient*.

For practical reasons this can be simplified as:  $\nabla f \approx |G_x| + |G_y|$

There is some debate as to how best to calculate these gradients but we will use:

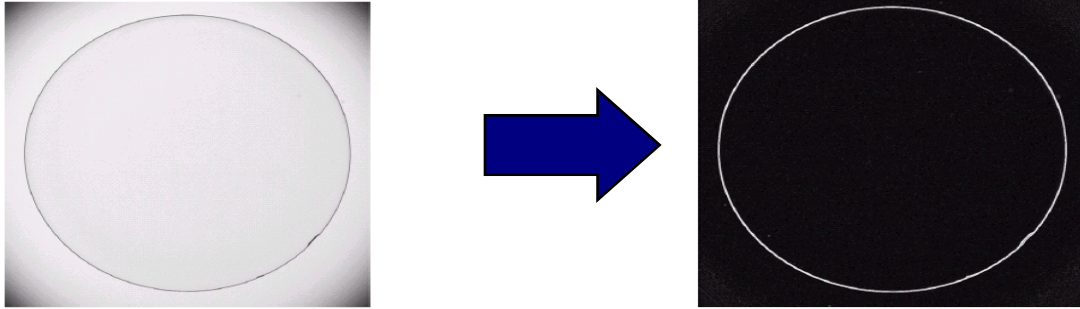
$$\nabla f \approx \left| (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3) \right| + \left| (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7) \right|$$

Based on the previous equations we can derive the *Sobel Operators*

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

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

To filter an image it is filtered using both operators the results of which are added together.

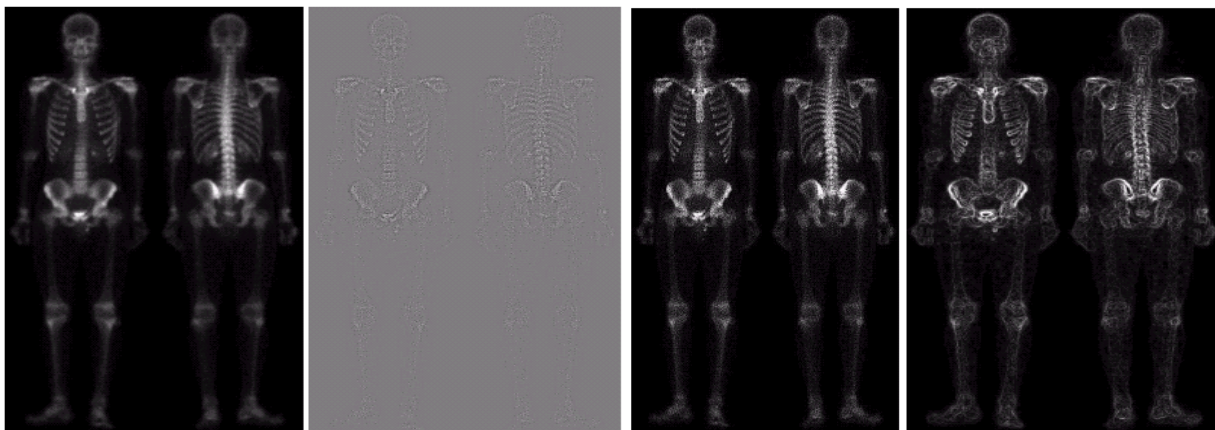


**Fig: Optical image of contact lens (b) Sobel gradient obtained with the two sobel masks using above two masks. The edge defects are quite visible in this image.**

Sobel filters are typically used for edge detection

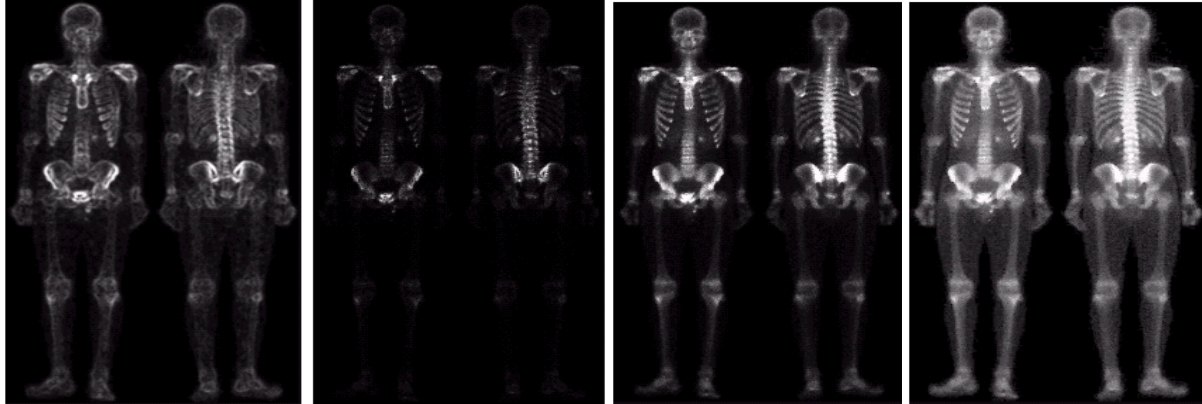
### Combining Spatial Enhancement Methods

Successful image enhancement is typically not achieved using a single operation. Rather we combine a range of techniques in order to achieve a final result. This example will focus on enhancing the bone scan to the right



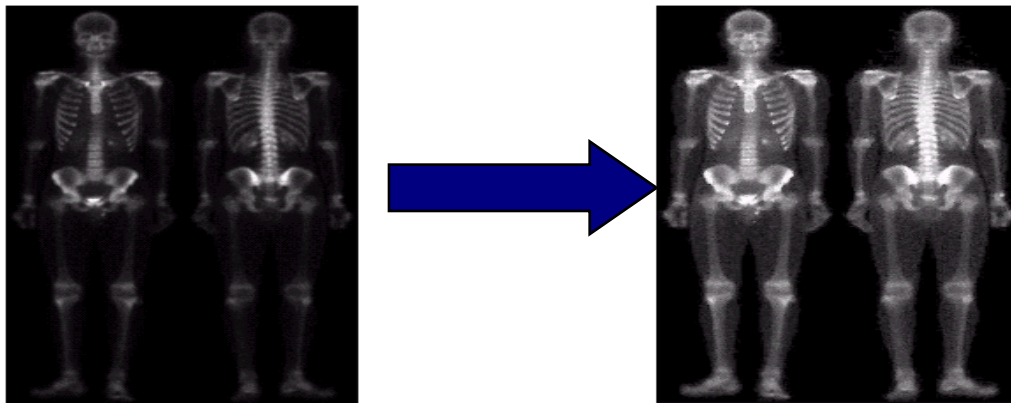
**(a)Original image (b) Laplacian of (a) (c) Sharpened image obtained by adding (a) and (b) (d) Sobel of (a)**





(e) Sobel image smoothed with 5X5 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 image obtained by applying power law transform to (g)

Compare the original and final images



### Image Enhancement in the Frequency Domain

#### Fourier series

Any function that periodically repeats itself can be expressed as the sum of sines and/or cosines

of different frequencies, each multiplied by a different coefficient

### Fourier transform

Even functions that are not periodic (but whose area under the curve is finite) can be expressed as the integral of sines and/or cosines multiplied by a weighting function

The one-dimensional Fourier transform and its inverse Fourier transform (continuous case)

$$F(u) = \int_{-\infty}^{\infty} f(x)e^{-j2\pi ux} dx \quad \text{where } j = \sqrt{-1}$$

Where

$$e^{j\theta} = \cos\theta + j\sin\theta$$

Inverse Fourier transform:

$$f(x) = \int_{-\infty}^{\infty} F(u)e^{j2\pi ux} du$$

The two-dimensional Fourier transform and its inverse Fourier transform (continuous case)

$$F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y)e^{-j2\pi(ux+vy)} dx dy$$

Inverse Fourier transform

$$f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u, v)e^{j2\pi(ux+vy)} dudv$$

The one-dimensional Fourier transform and its inverse (discrete time case)

Fourier transform (DFT)

$$F(u) = \frac{1}{M} \sum_{x=0}^{M-1} f(x)e^{-j2\pi ux/M} \quad \text{for } u = 0, 1, 2, \dots, M-1$$

Inverse Fourier transform (IDFT)

$F(u)$  can be expressed in polar coordinates:

$$f(x) = \sum_{u=0}^{M-1} F(u)e^{j2\pi ux/M} \quad \text{for } x = 0, 1, 2, \dots, M-1$$

$$F(u) = |F(u)|e^{j\phi(u)}$$

where  $|F(u)| = [R^2(u) + I^2(u)]^{1/2}$  (magnitude or spectrum)

$$\phi(u) = \tan^{-1} \left[ \frac{I(u)}{R(u)} \right] \quad \text{(phase angle or phase spectrum)}$$

–  $I(u)$ : the imaginary part of  $F(u)$

Power spectrum:

$$P(u) = |F(u)|^2 = R^2(u) + I^2(u)$$

The two-dimensional Fourier transform and its inverse (discrete time case)

Fourier transform (DFT)

$$F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(ux/M + vy/N)}$$

for  $u = 0, 1, 2, \dots, M-1, v = 0, 1, 2, \dots, N-1$

Inverse Fourier transform (IDFT)

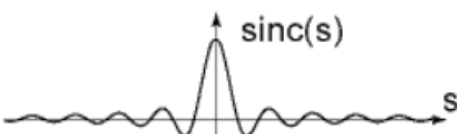
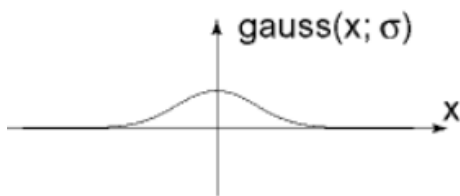
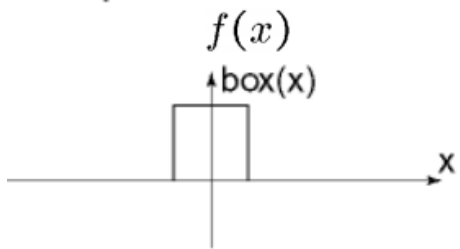
$$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi(ux/M + vy/N)}$$

for  $x = 0, 1, 2, \dots, M-1, y = 0, 1, 2, \dots, N-1$

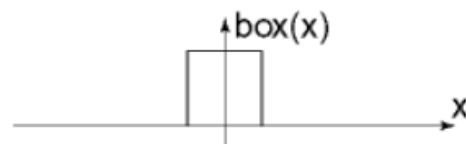
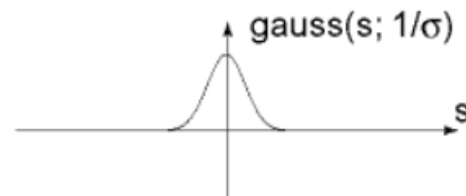
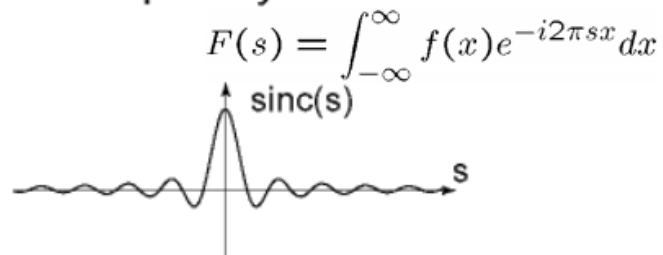
$u, v$  : the transform or frequency variables

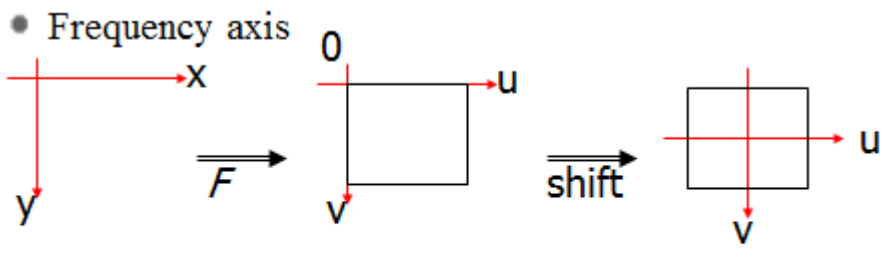
•  $x, y$  : the spatial or image variables

Spatial domain

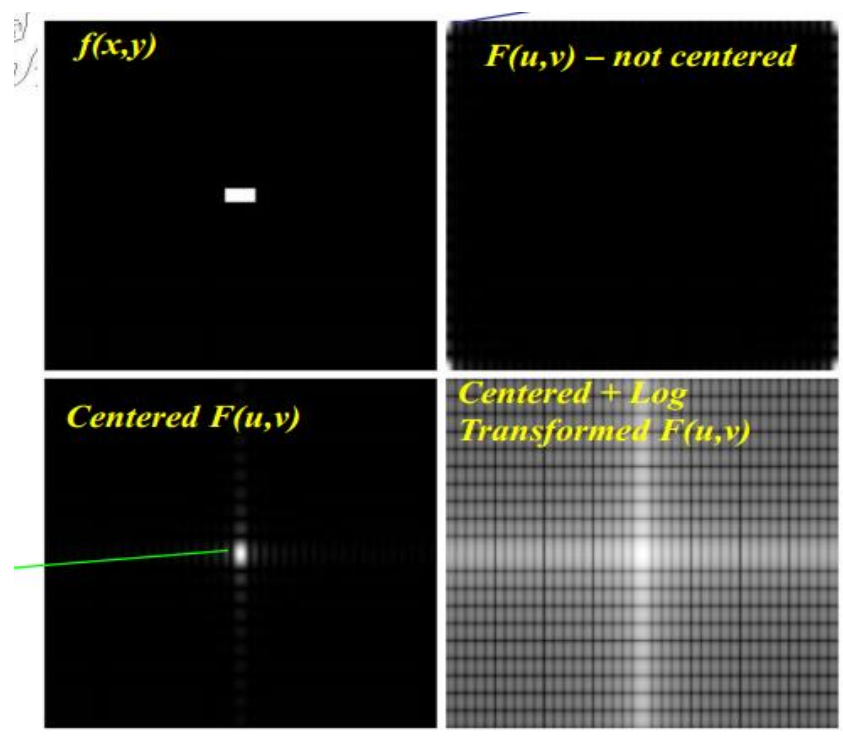


Frequency domain





$$\mathfrak{F}[f(x, y)(-1)^{x+y}] = F(u - M/2, v - N/2)$$

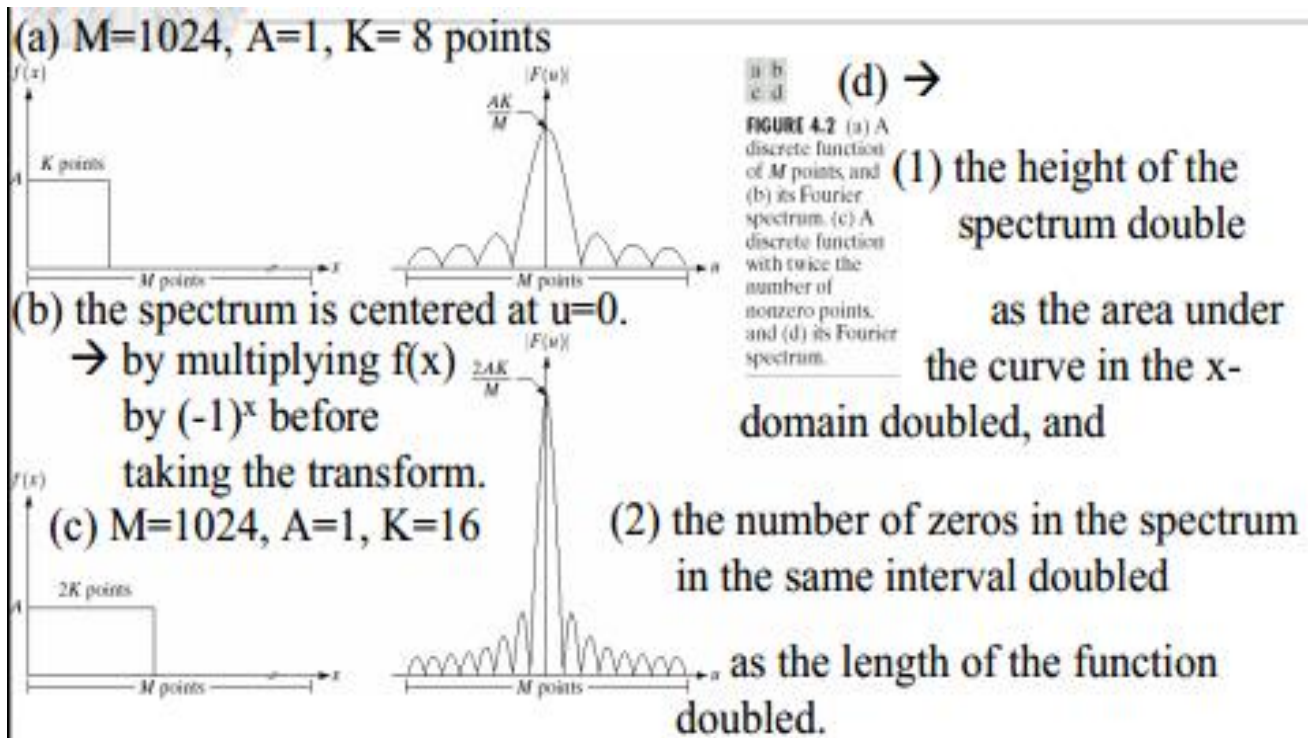


a	b
c	d

**FIGURE 4.3**  
 (a) A simple image.  
 (b) Fourier spectrum.  
 (c) Centered spectrum.  
 (d) Spectrum visually enhanced by a log transformation.

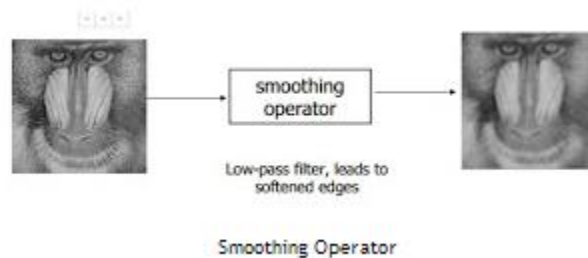
Log(F)

Some One-Dimensional Fourier Transform Examples



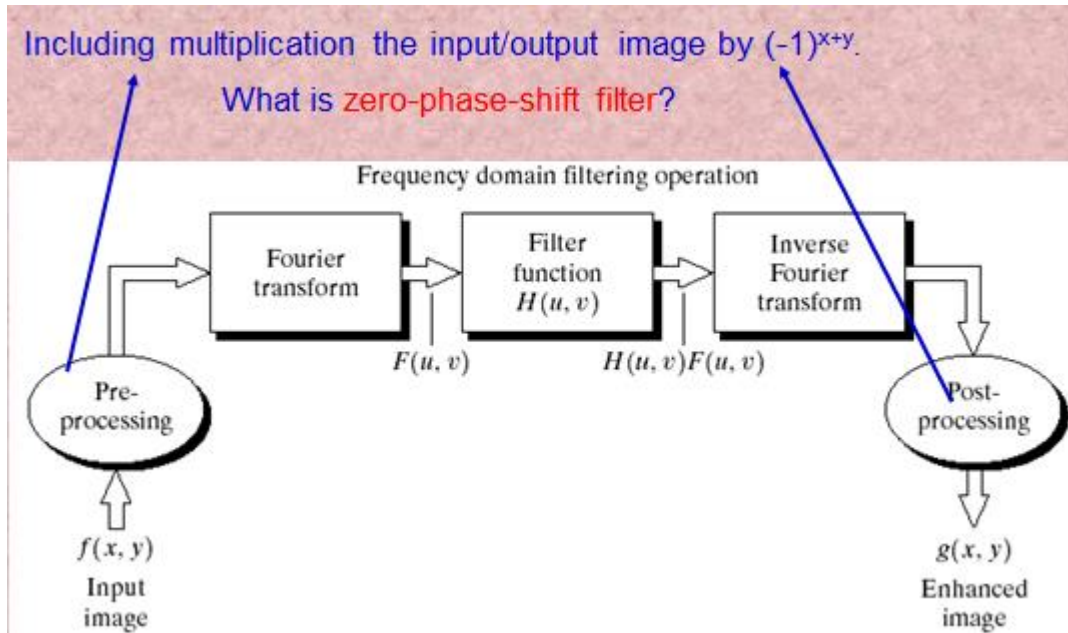
### Correspondence between filtering in spatial and frequency domains

Spatial domain refers to the image plane itself, and approaches in this category are based on direct manipulation of pixels in an image. Used for filtering basics, smoothing filters, sharpening filters, unsharp masking and laplacian.



Frequency domain processing techniques are based on modifying the Fourier transform of the image.

## Steps involved in Frequency domain filtering



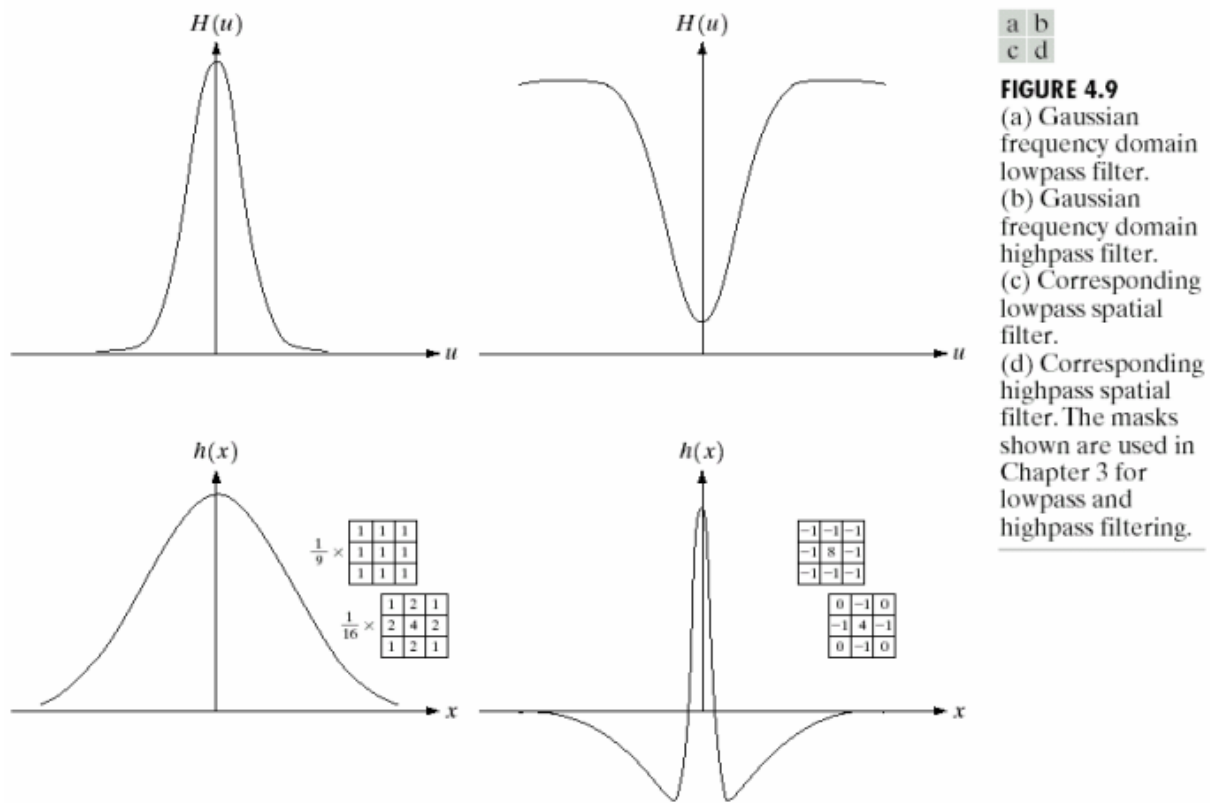
### Basics of filtering in the frequency domain

Filtering in the frequency domain is straightforward. It consists of the following steps:

1. Multiply the input image  $f(x, y)$  by  $(-1)^{x+y}$  to center the transform.
2. Compute  $F(u, v)$ , 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  $(-1)^{x+y}$ .

**Point 3:**  $G(u, v) = H(u, v)F(u, v)$  where  $G(u, v)$  is the Fourier transform of the filtered image and  $F(u, v)$  the Fourier transform of the Input

**Point 4:** Output image =  $F^{-1}(G(u, v))$



**FIGURE 4.9**  
 (a) Gaussian frequency domain lowpass filter.  
 (b) Gaussian frequency domain highpass filter.  
 (c) Corresponding lowpass spatial filter.  
 (d) Corresponding highpass spatial filter. The masks shown are used in Chapter 3 for lowpass and highpass filtering.

Some Basic Filters and their properties

**LPF**  
**Ideal LPF**

The simplest lowpass filter we can imagine is a filter that “cuts off” all high frequency components of the Fourier transform that are at a distance greater than a specified distance  $D_0$  from the origin of the (centered) transform. Such a filter is called a two-dimensional (2-D) *ideal lowpass filter* (ILPF) and has the transfer function given below

$$H(u, v) = \begin{cases} 1, & \text{if } D(u, v) \leq D_0 \\ 0, & \text{if } D(u, v) > D_0 \end{cases}$$

where

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

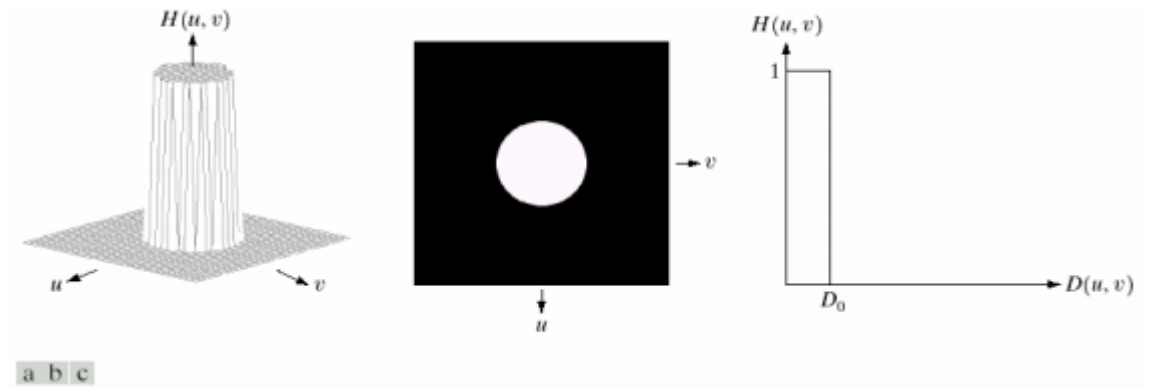


Figure 2 (a) Perspective plot of an ILPF transfer function. (b) filter displayed as an image. (c) Filter radial cross section.

The total image power  $P_T$  is obtained by summing the components of the power spectrum at each point  $(u, v)$ , for  $u=0, 1, 2, \dots, M-1$  and  $v=0, 1, 2, \dots, N-1$  that is

$$P_T = \sum_{u=1}^{M-1} \sum_{v=0}^{N-1} P(u, v)$$

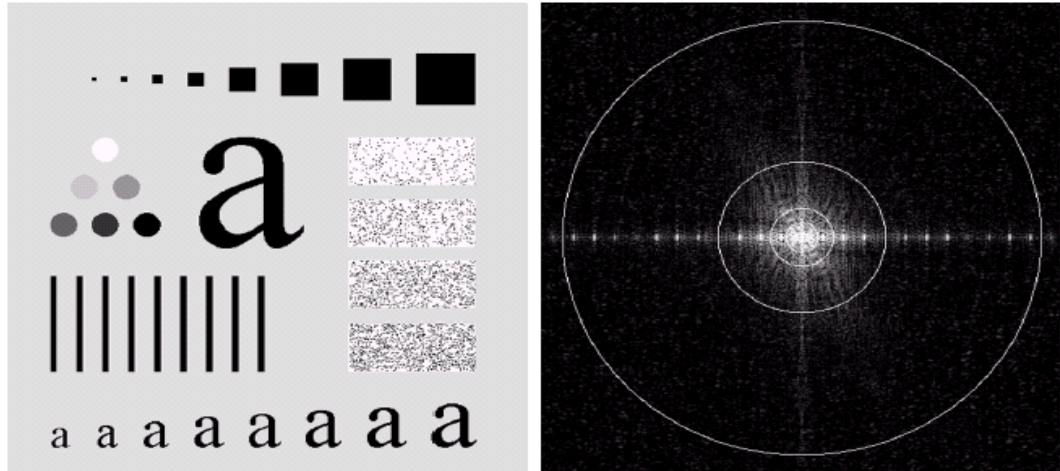
Where

$$P(u) = |F(u)|^2 = R^2(u) + I^2(u)$$

If the transform has been centered, a circle of radius  $r$  with origin at the center of the frequency rectangle encloses  $\alpha$  percentage of the power, where

$$\alpha = 100 \left[ \sum_u \sum_v P(u, v) / P_T \right]$$

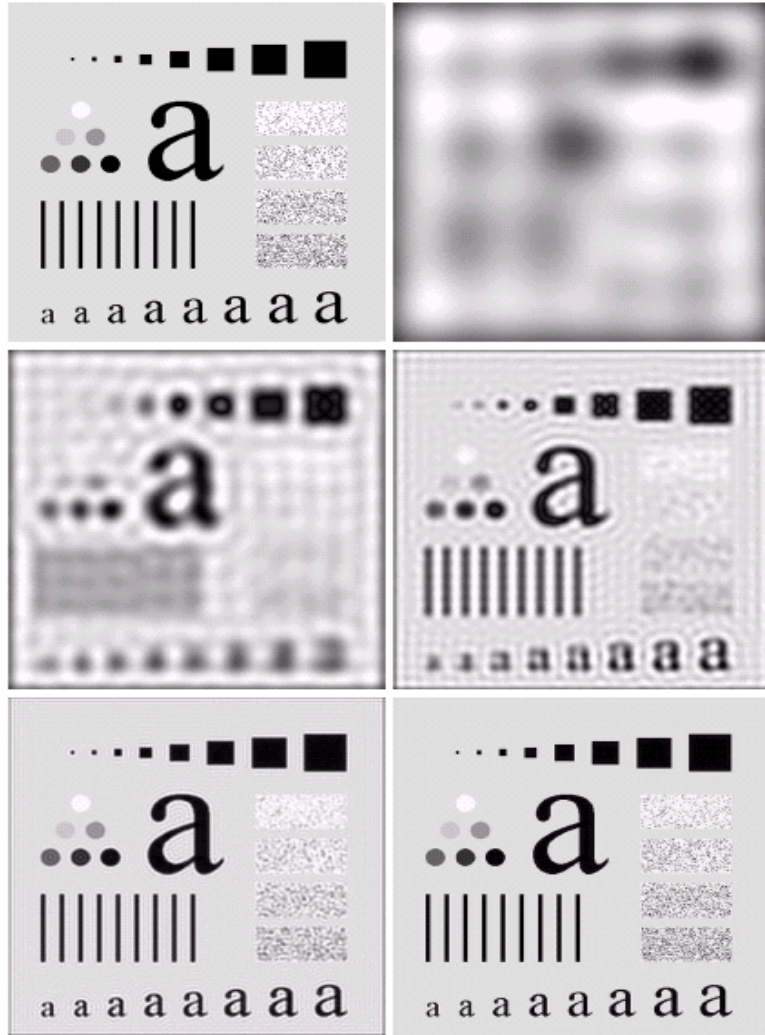




a b

**FIGURE 4.11** (a) An image of size  $500 \times 500$  pixels and (b) its Fourier spectrum. The superimposed circles have radii values of 5, 15, 30, 80, and 230, which enclose 92.0, 94.6, 96.4, 98.0, and 99.5% of the image power, respectively.

Above figure shows the text pattern we used to illustrate spatial blurring. The fourier spectrum is shown in 4.11(b). The circles superimposed on the spectrum have radii of 5,15,30,80 and 230 pixels. These circles enclose 92.0, 94.6, 96.4, 98 and 99.5 % of power respectively.



a	b
c	d
e	f

**FIGURE 4.12** (a) Original image. (b)–(f) Results of ideal lowpass filtering with cutoff frequencies set at radii values of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b). The power removed by these filters was 8, 5.4, 3.6, 2, and 0.5% of the total, respectively.

Fig 4.12 (b) The severe blurring in this image is a clear indication that most of the sharp detail information in this picture is contained in the 8% power removed by the filter. As the filter radius increases, less and less power is removed, resulting in less severe blurring. Figure 4.12 (c) - (e) are characterized by ringing which becomes finer in texture as the amount of high frequency content removed decreases.

### Another example of ILPF

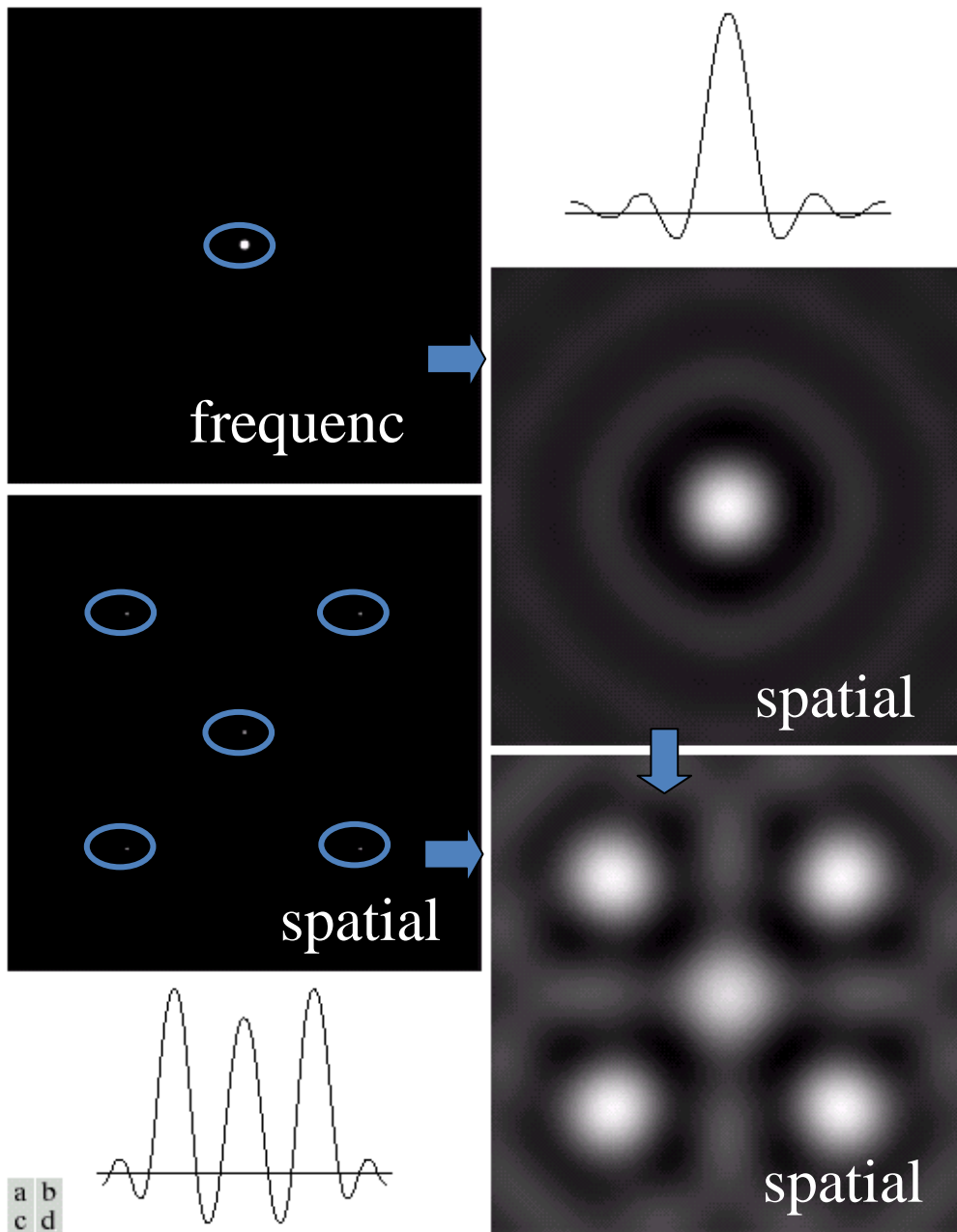


Figure 4.13 (a) A frequency-domain ILPF of radius 5. (b) Corresponding spatial filter. (c) Five impulses in the spatial domain, simulating the values of five pixels. (d) Convolution of (b) and (c) in the spatial domain.

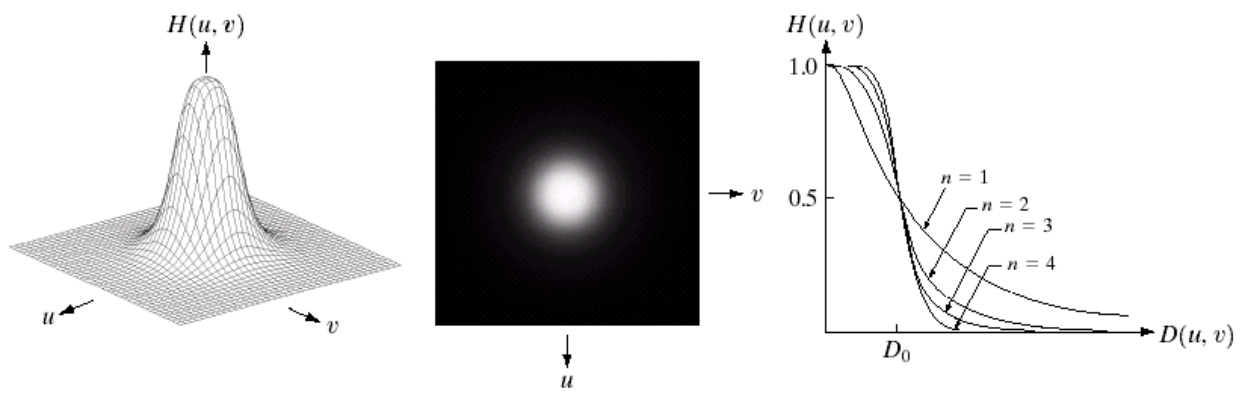
$$f(x, y) * h(x, y) \Leftrightarrow F(u, v)H(u, v)$$

Notation: the radius of center component and the number of circles per unit distance from the origin are inversely proportional to the value of the cutoff frequency.

## Butterworth

The transfer function of a Butterworth lowpass filter (BLPF) of order  $n$ . and with cutoff frequency at a distance  $D_0$  from the origin, is defined as

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



a b c

**FIGURE 4.14** (a) Perspective plot of a Butterworth lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections of orders 1 through 4.



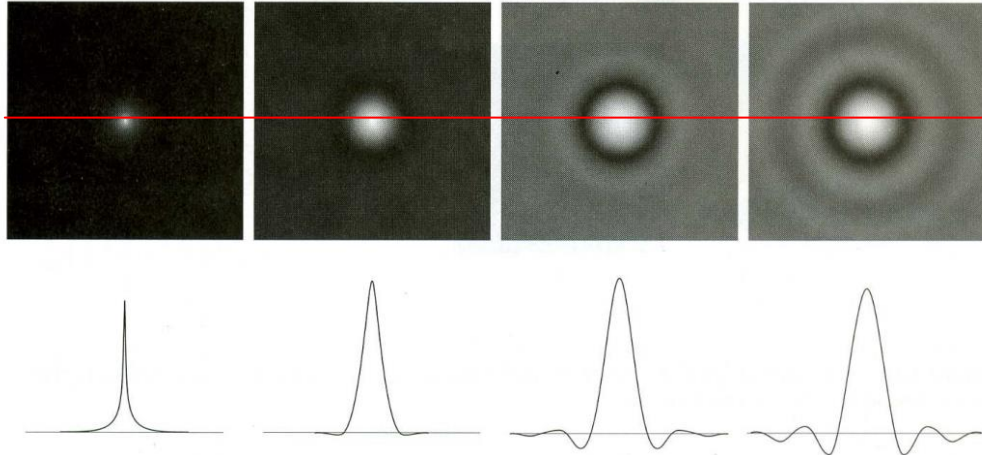


Figure 4 (a)—(d) Spatial representation of BLPFs of order 1,2.5, and 20. and corresponding gray-level profiles through the center of the filters (all filters have a cutoff frequency of 5). Note that ringing increases as a function of filter order.

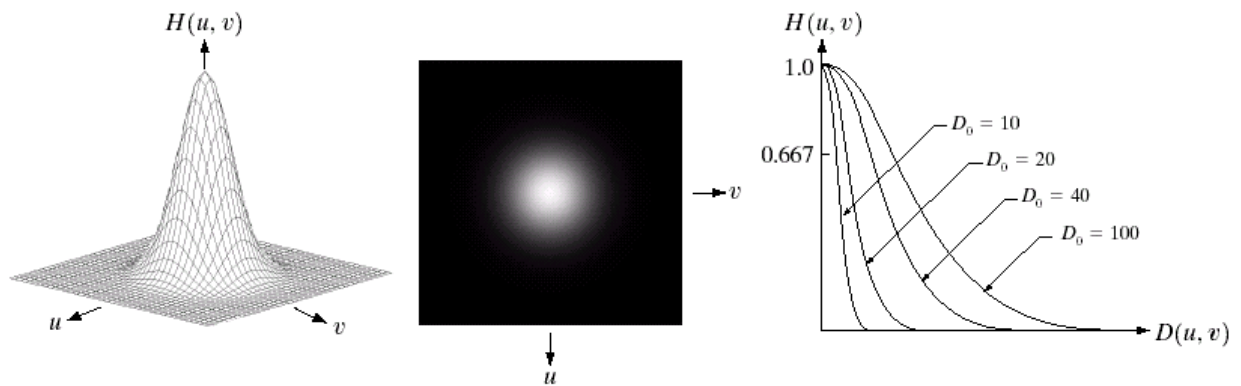
It may be noted that BLPF of order 20 has a filter function almost akin to that of ILPF

### Gaussian

The filter transfer Function of Gaussian Low Pass Filter (GLPF) is given as

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

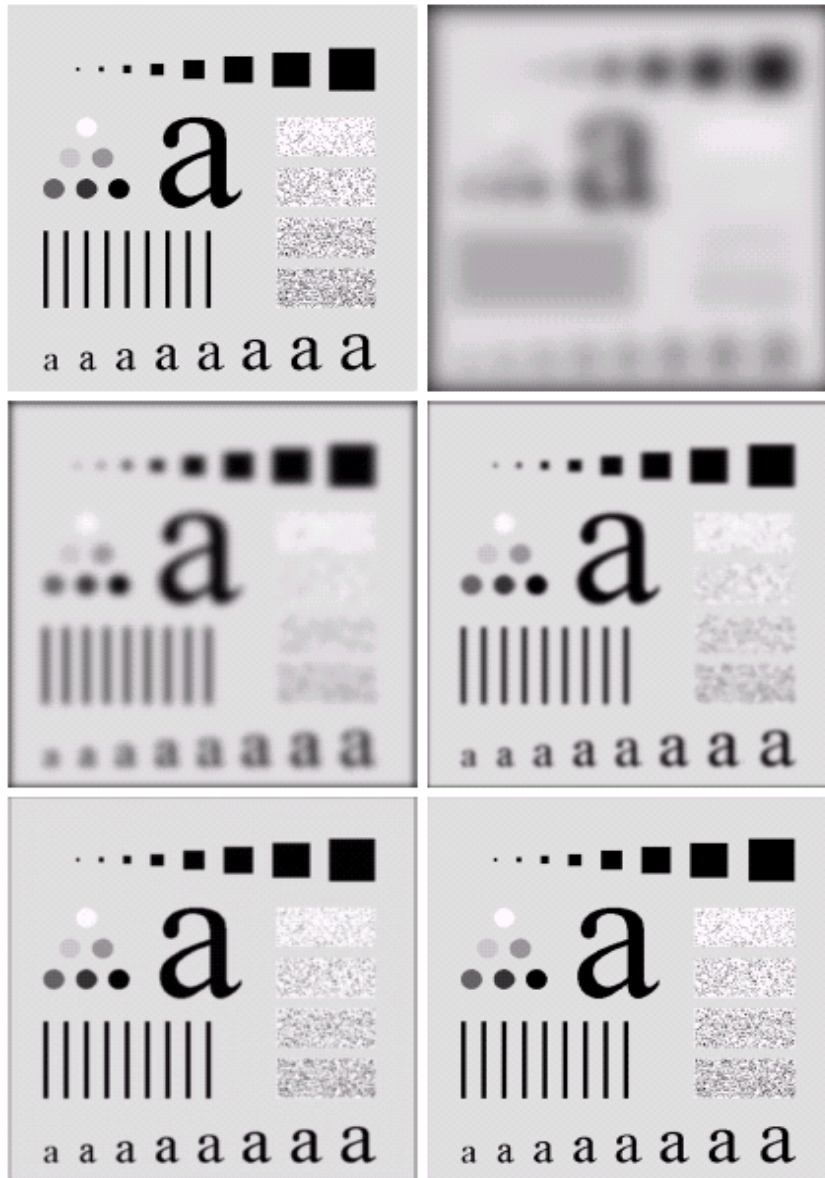
where  $\sigma$  is replaced by  $D_0$  the cutoff



a b c

**FIGURE 4.17** (a) Perspective plot of a GLPF transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections for various values of  $D_0$ .





**FIGURE 4.18** (a) Original image. (b)–(f) Results of filtering with Gaussian lowpass filters with cutoff frequencies set at radii values of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b). Compare with Figs. 4.12 and 4.15.

a	b
c	d
e	f

The GLPF did not achieve as much smoothing as the BLPF of order 2 for the same value of cutoff frequency.

## Additional example of low pass filtering



a b c

**FIGURE 4.20** (a) Original image ( $1028 \times 732$  pixels). (b) Result of filtering with a GLPF with  $D_0 = 100$ . (c) Result of filtering with a GLPF with  $D_0 = 80$ . Note reduction in skin fine lines in the magnified sections of (b) and (c).

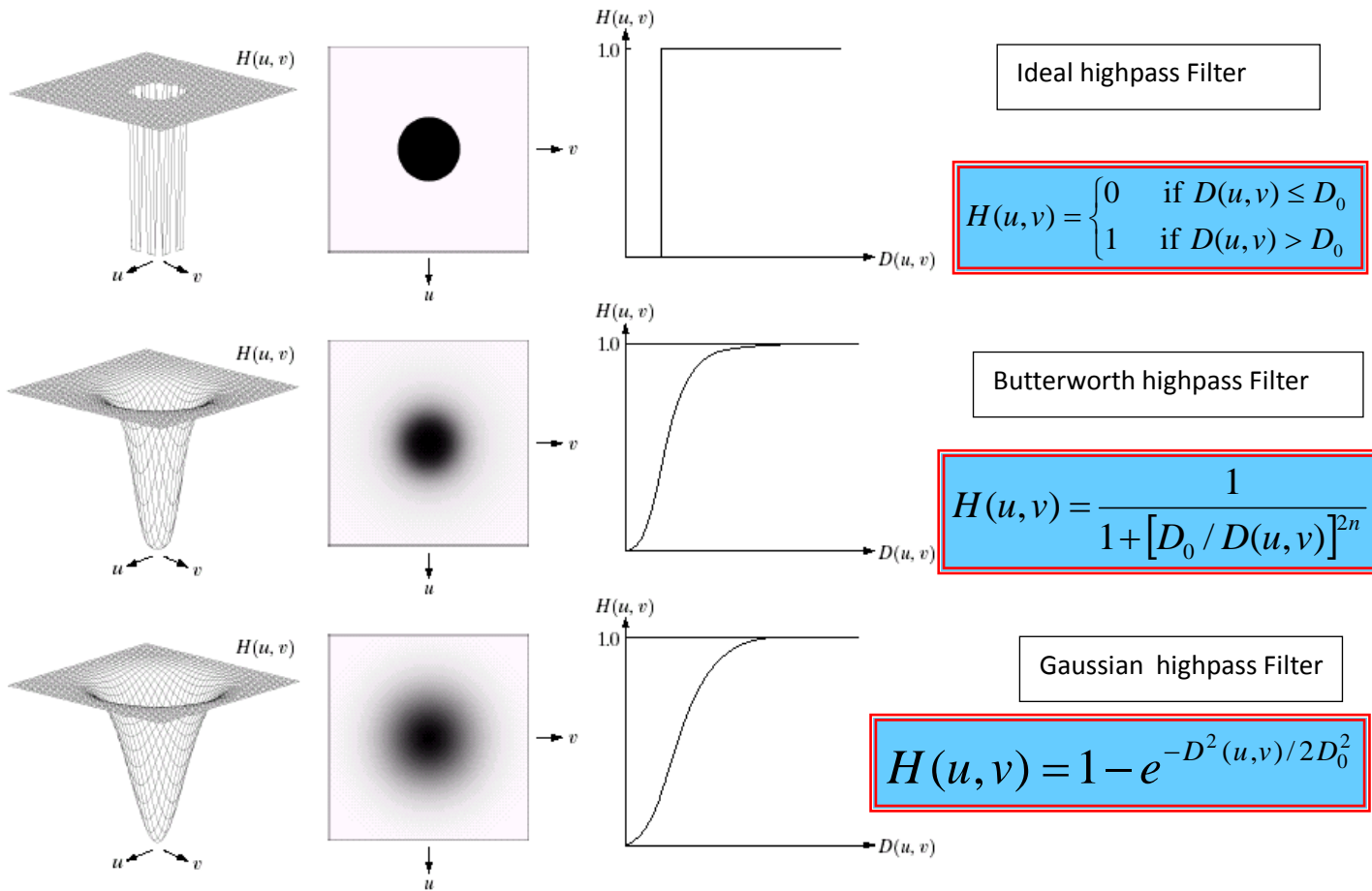
---

## HPF

Sharpening in Frequency domain can be got by applying HPF using the main scheme that

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

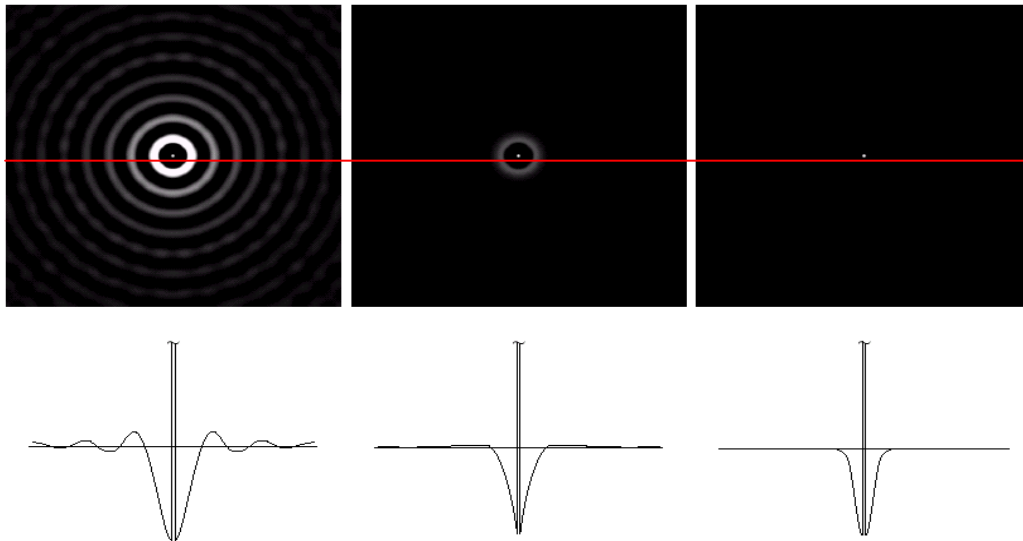




a b c  
d e f  
g h i

**FIGURE 4.22** Top row: Perspective plot, image representation, and cross section of a typical ideal highpass filter. Middle and bottom rows: The same sequence for typical Butterworth and Gaussian highpass filters.

## Highpass Filters Spatial Representations



a b c

**FIGURE 4.23** Spatial representations of typical (a) ideal, (b) Butterworth, and (c) Gaussian frequency domain highpass filters, and corresponding gray-level profiles.

## Ideal HPF

Filter function of an Ideal HPF is given as below

$$H(u, v) = \begin{cases} 0 & \text{if } D(u, v) \leq D_0 \\ 1 & \text{if } D(u, v) \geq D_0 \end{cases}$$

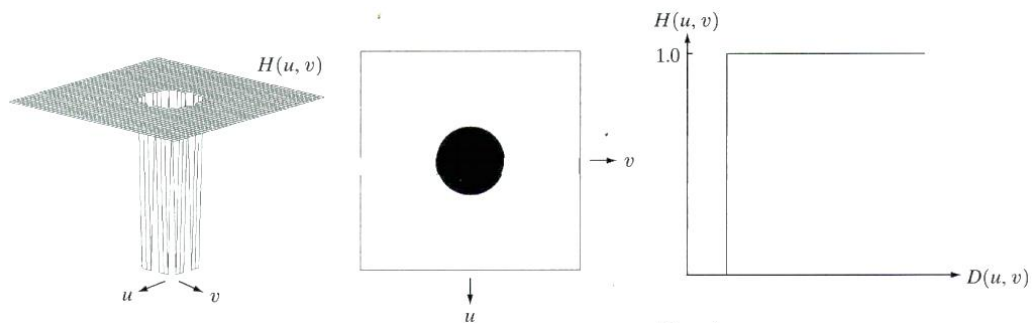


Figure 6 (a) Perspective plot of an IHPF transfer function. (b) filter displayed as an image. (c) Filter radial cross section.



a b c

**FIGURE 4.24** Results of ideal highpass filtering the image in Fig. 4.11(a) with  $D_0 = 15, 30,$  and  $80,$  respectively. Problems with ringing are quite evident in (a) and (b).

## Butterworth

The result is smoother than that of IHPFs and sharper than that of GHPFs

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

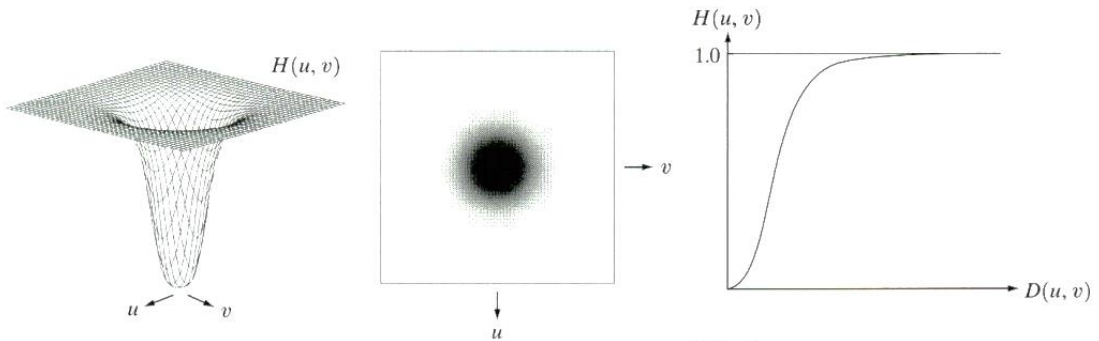


Figure 7 (a)Perspective plot of an BHPF transfer function. (b) filter displayed as an image. (c) Filter radial cross section.

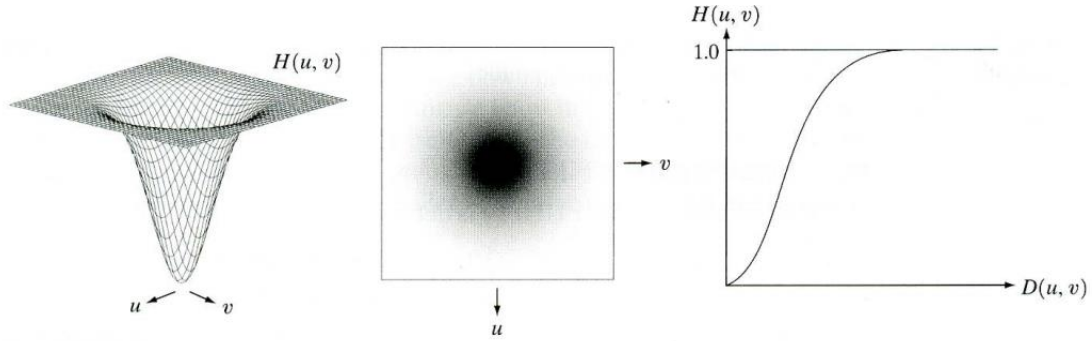


a b c

**FIGURE 4.25** Results of highpass filtering the image in Fig. 4.11(a) using a BHPF of order 2 with  $D_0 = 15, 30,$  and  $80,$  respectively. These results are much smoother than those obtained with an ILPF.

## Gaussian

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



**Figure 8 (a) Perspective plot of an GHPF transfer function. (b) filter displayed as an image. (c) Filter radial cross section.**

## Laplacian

The FT of n-order differential of a function is

$$\mathfrak{T}\left[\frac{d^n f(x)}{dx^n}\right] = (ju)^n F(u)$$

For a two-dimensional function  $f(x, y)$ , it can be shown that

$$\mathfrak{T}[\nabla^2 f(x, y)] = \mathfrak{T}\left[\frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2}\right] = (ju)^2 F(u, v) + (jv)^2 F(u, v) = -(u^2 + v^2)F(u, v)$$

So, Laplacian can be implemented in the frequency domain by using the filter

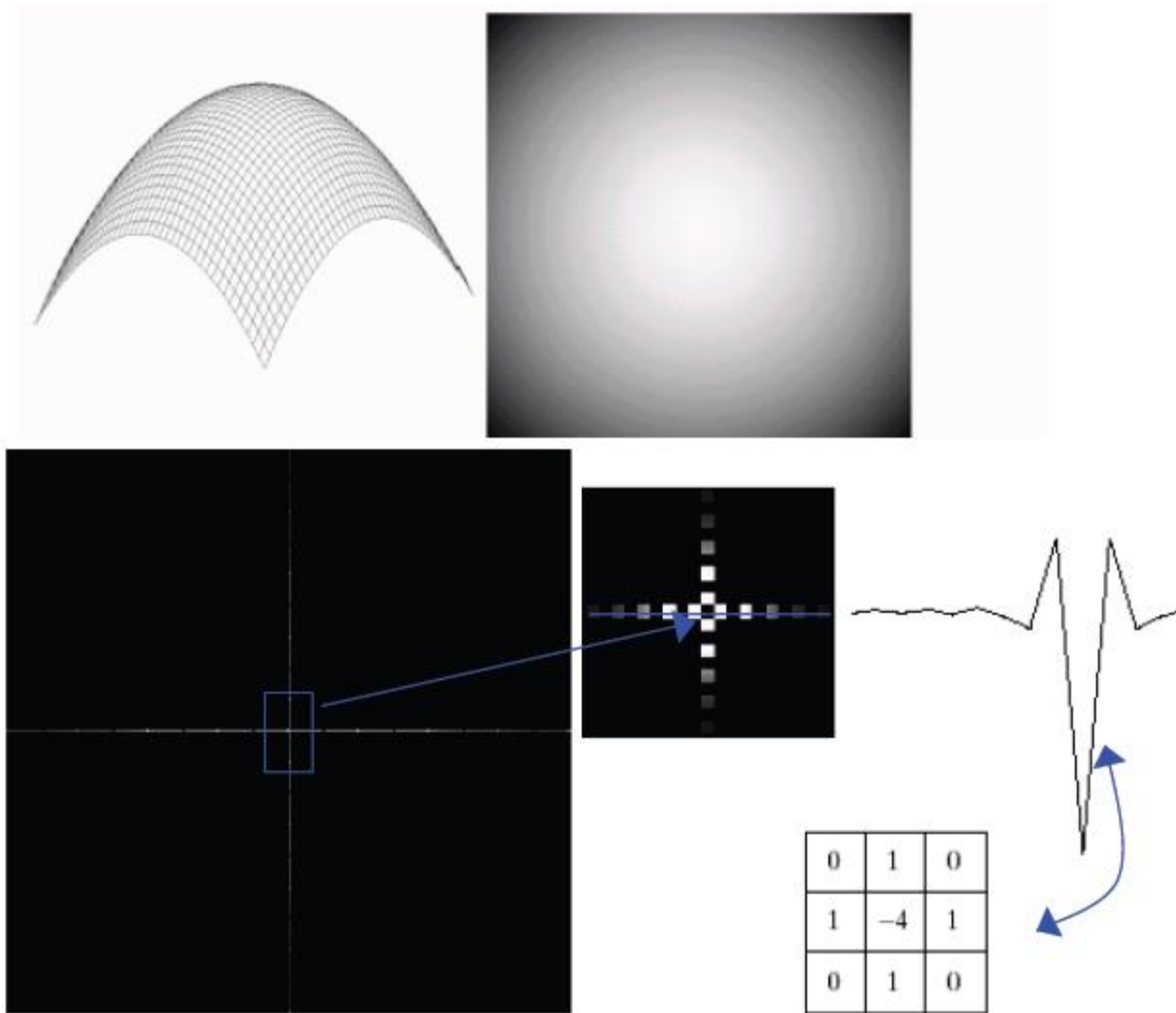
$$H(u, v) = -(u^2 + v^2)$$

Shift the center to  $(M/2, N/2)$  and obtain

$$H(u, v) = -\left[(u - M/2)^2 + (v - N/2)^2\right]$$

We have the following Fourier transform pairs

$$\nabla^2 f(x, y) \Leftrightarrow -\left[(u - M/2)^2 + (v - N/2)^2\right] F(u, v)$$



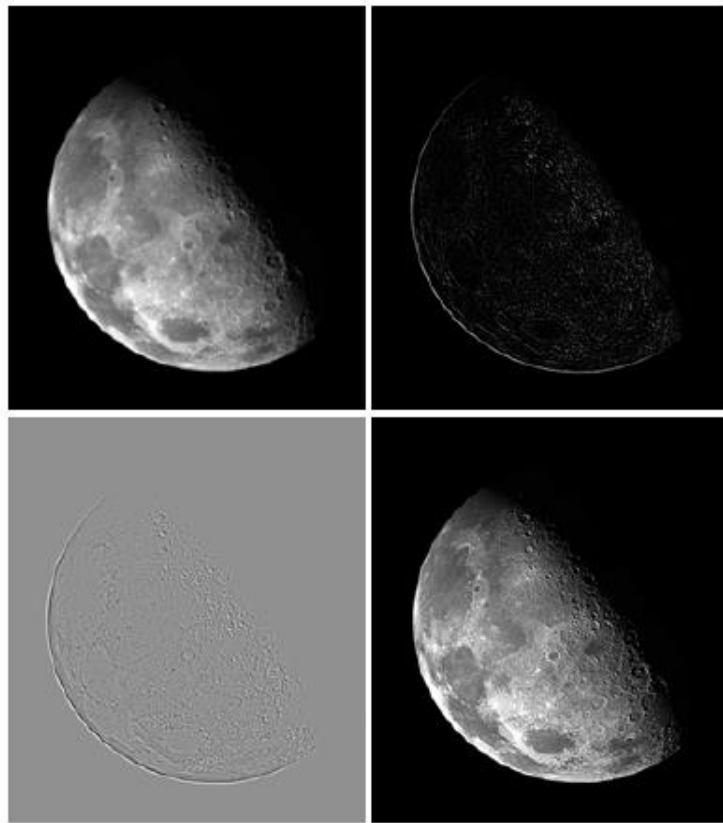
1)The function is centered at  $(M/2, N/2)$ . Value at the top of the dome is zero. All other values are zero. 2)  $H(u,v)$  is also centered. (3) The laplacian in the spatial domain is obtained by (i) multiplying  $h(u,v)$  by  $(-1)^{x+y}$ . (ii) Taking inverse fourier transform (iii) multiplying real part of result by  $(-1)^{x+y}$ . (4) A zoomed section of (3) (5) A horizontal grey level profile passing through center of (4)

Laplacian output is not a enhanced image. To get enhanced image we have to subtract original image from the laplacian image.

$$g(x, y) = f(x, y) - \nabla^2 f(x, y) = \mathfrak{F}^{-1} \left\{ \left[ 1 + \left( \left( u - \frac{M}{2} \right)^2 + \left( v - \frac{N}{2} \right)^2 \right) \right] F(u, v) \right\}$$

a b  
c d

**FIGURE 3.40**  
 (a) Image of the North Pole of the moon.  
 (b) Laplacian-filtered image.  
 (c) Laplacian image scaled for display purposes.  
 (d) Image enhanced by using Eq. (3.7-5).  
 (Original image courtesy of NASA.)



### Unsharp masking and high-boost filtering

A process used for many years in the publishing industry to sharpen images consists of subtracting a blurred version of an image from the image itself. This process, called *unsharp masking*, is expressed as

$$f_{hp}(x, y) = f(x, y) - f_{lp}(x, y) \text{ -----(1)}$$

where  $f_s(x, y)$  denotes the sharpened image obtained by unsharp masking, and is a blurred version of  $f(x, y)$ .

A slight further generalization of unsharp masking is called *high-boost filtering*. A high-boost filtered image,  $f_{hb}$ , is defined at any point  $(x, y)$  as

$$f_{hb}(x, y) = Af(x, y) - f_{lp}(x, y) \quad \text{Where } A \geq 1$$

Above equation can be written as  $f_{hb}(x, y) = (A-1)f(x, y) - f_{lp}(x, y) + f(x, y)$

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

When  $A=1$ , high boost filter reduces to regular high pass filter.

From equation (1)

$$F_{hp}(u, v) = F(u, v) - F_{lp}(u, v)$$

$$F_{lp}(u, v) = F(u, v)H_{lp}(u, v)$$

Therefore unsharp masking can be implemented directly in frequency domain by using composite filter

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

Similar high boost filtering can be implemented with the composite filter ( $A \geq 1$ )

$$H_{hb}(u, v) = (A - 1) + H_{lp}(u, v)$$

a b  
c d

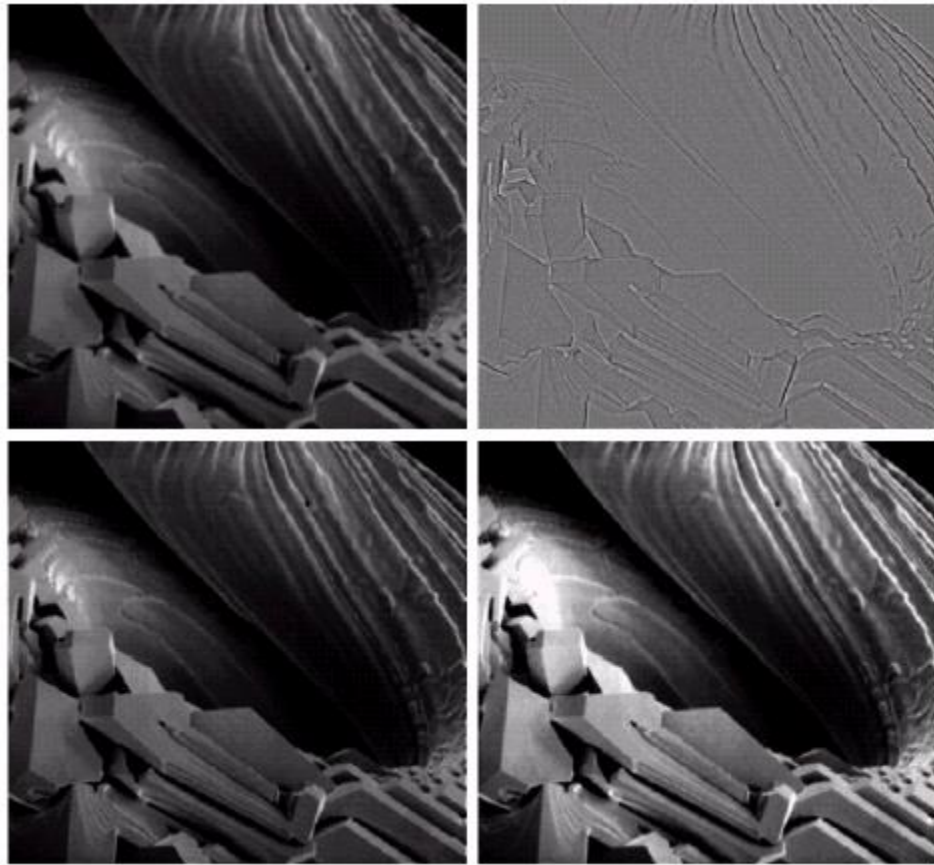
**FIGURE 4.29**

Same as Fig. 3.43, but using frequency domain filtering. (a) Input image.

(b) Laplacian of

(a). (c) Image obtained using Eq. (4.4-17) with  $A = 2$ . (d) Same as (c), but with  $A = 2.7$ . (Original image courtesy of

Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene.)



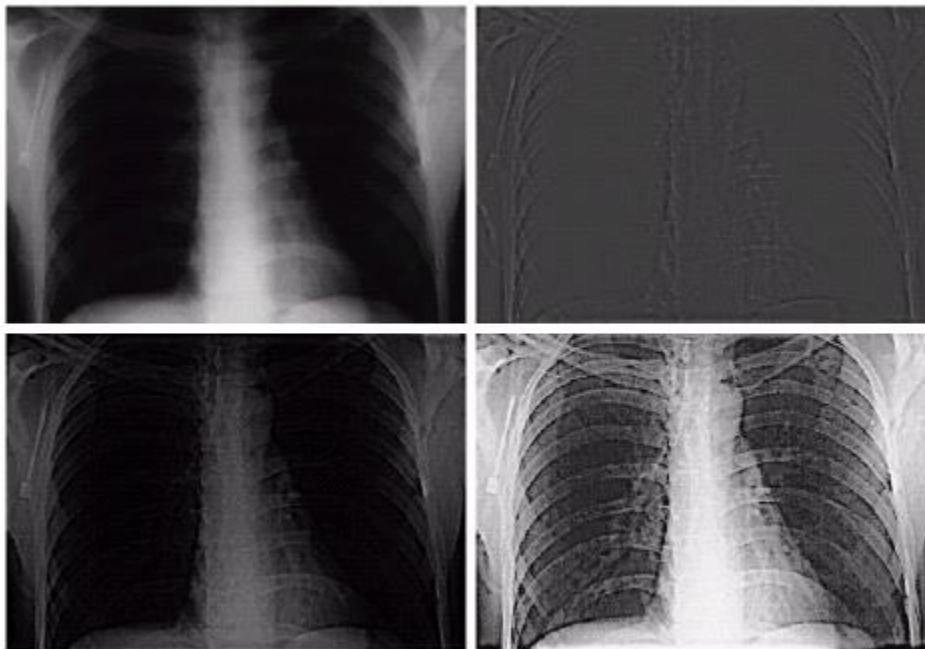


High frequency emphasis has a filter function given by

$$H_{hfe}(u,v) = a + b H_{hp}(u,v)$$

Where  $a \geq 0$  and  $b > a$ .

Typically,  $a = [0.25, 0.5]$  and  $b = [1.5, 2.0]$ .



a b  
c d

**FIGURE 4.30**  
(a) A chest X-ray image. (b) Result of Butterworth highpass filtering. (c) Result of high-frequency emphasis filtering. (d) Result of performing histogram equalization on (c). (Original image courtesy Dr. Thomas R. Gest, Division of Anatomical Sciences, University of Michigan Medical School.)

## Homomorphic Filtering

An image  $f(x,y)$  can be defined as the product of illumination  $i(x,y)$  and reflectance  $r(x,y)$

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

Because the fourier transform of the product of two function is not separable:

$$\mathfrak{F}\{f(x, y)\} \neq \mathfrak{F}\{i(x, y)\}\mathfrak{F}\{r(x, y)\}$$

Therefore taking natural logarithm on both side

$$[\mathfrak{z}(x, y)] = \ln(f(x, y)) = \mathfrak{F}[\ln i(x, y)] + \mathfrak{F}[\ln r(x, y)]$$



Taking Fourier transform on both sides

$$\mathfrak{F}[z(x, y)] = \mathfrak{F}[\ln i(x, y)] + \mathfrak{F}[\ln r(x, y)]$$

$$Z(u, v) = F_i(u, v) + F_r(u, v)$$

Where  $F_i(u, v) = \ln i(x, y)$        $F_r(u, v) = \ln r(x, y)$

If we process  $Z(u, v)$  by means of filter function  $H(u, v)$  then,

$$S(u, v) = H(u, v)F_i(u, v) + H(u, v)F_r(u, v)$$

In spatial domain  $s(x, y) = \mathfrak{F}^{-1}\{S(u, v)\}$

$$s(x, y) = \mathfrak{F}^{-1}\{H(u, v)F_i(u, v) + H(u, v)F_r(u, v)\}$$

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

Where  $i'(x, y) = \mathfrak{F}^{-1}\{H(u, v)F_i(u, v)\}$        $r'(x, y) = \mathfrak{F}^{-1}\{H(u, v)F_r(u, v)\}$

Finally, as  $z(x, y)$  was formed by taking the logarithm of the original image  $f(x, y)$ , the inverse (exponential) operation yields the desired enhanced image, denoted by  $g(x, y)$ , that is

$$g(x, y) = e^{s(x, y)} = e^{i'(x, y)} e^{r'(x, y)} = i_o(x, y) r_o(x, y) \quad \text{where}$$

$$i_o(x, y) = e^{i'(x, y)}$$

$$r_o(x, y) = e^{r'(x, y)}$$

This filtering technique is most famous for removing multiplicative noise and is carried out as shown in fig 10

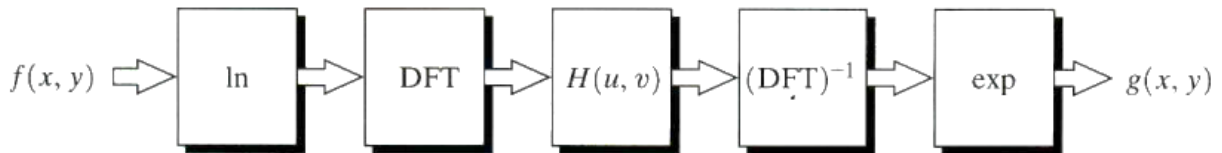


Figure 10 Block diagram of the process of Homomorphic filtering

The illumination components of an image generally characterized by slow spatial variations, while the reflectance component tend to vary abruptly, particularly at the junctions of dissimilar objects.

Low frequency-----→ Illumination  
 High frequency-----→ Reflectance

If the parameter  $\gamma_L$  and  $\gamma_H$  are chosen so that  $\gamma_L < 1$  and  $\gamma_H > 1$ , the filter function shown in figure below tends to decrease the contribution made by the slow frequencies (illumination) and amplify the contribution made by high frequency components (reflectance). The net result is simultaneous dynamic range compression and contrast enhancement.

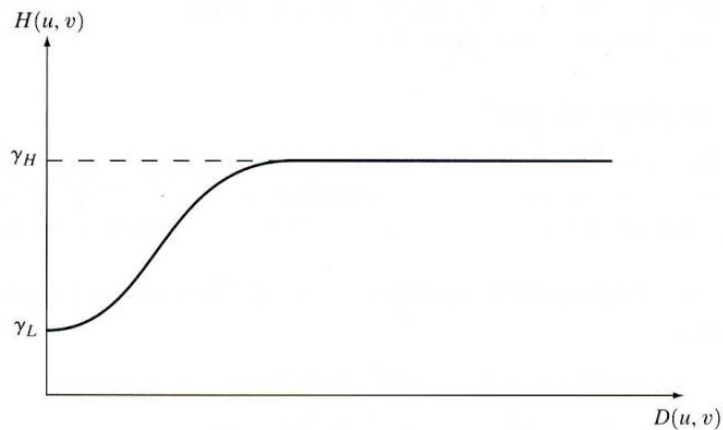


Figure 11 Filter Transfer Function of a Homomorphic Filter

Using a slightly modified form of the Gaussian filter gives us

$$H(u, v) = (\gamma_H - \gamma_L) [1 - e^{-c(D^2(u, v) / D_0^2)}] + \gamma_L$$

Where C is a constant and it controls the sharpness of the slope of the filter function

a b

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

