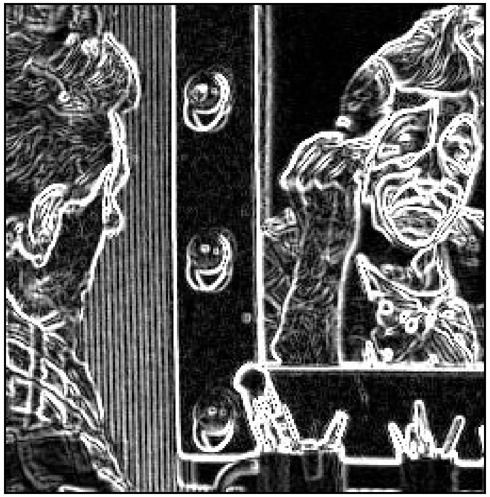
Chapter 2: Image Enhancement and Restoration

2. Image Enhancement: Principle Objective

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

Original Image



High Pass Filtering

2. Image Enhancement: Two Approaches (1)

- **Spatial Domain**: (image plane)
  – Techniques are based on direct manipulation of pixels in an image.

- **Frequency Domain**:
  – Techniques are based on modifying the Fourier transform of an image.

There are some enhancement techniques based on various combinations of methods from these two categories.

2. Image Enhancement: Two Approaches (2)

- **Spatial Domain Methods:** Assumed a digital image represented by a 2-D random field $f(x,y)$. Image processing operators in the spatial domain may be expressed as:

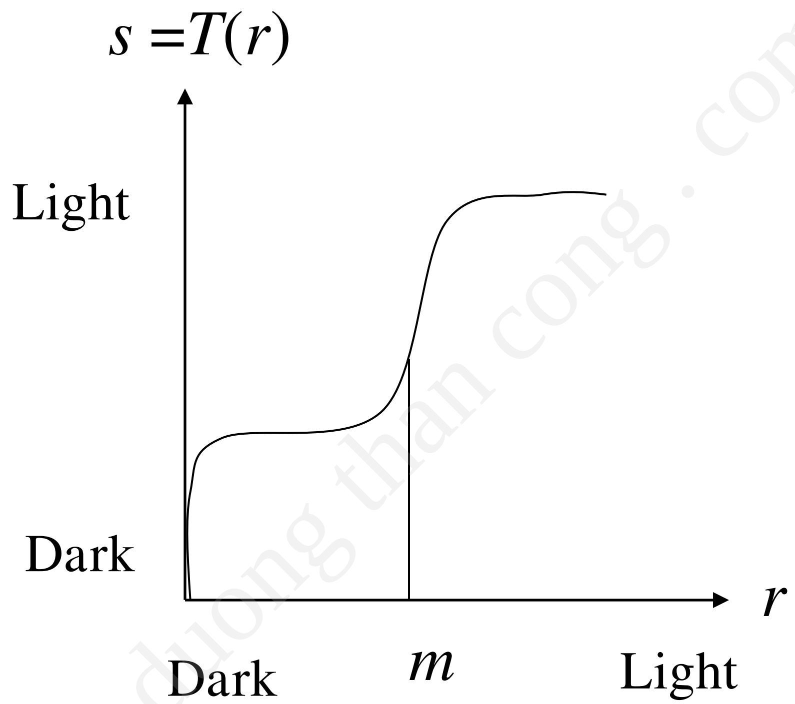
$$g(x,y) = T\left[f(x,y)\right]$$

where T is an operator applied on $f(x,y)$ defined over:

- a **single pixel** (x,y). In this case T is a gray level transformation (or mapping) function. This technique is referred to as **point processing**.
- **some neighbourhood** of (x,y), based on the use of mask (e.g. 3×3 array), window or filter → **mask processing** (or **filtering**).
- T may operate to a set of input images.

2. Image Enhancement: Two Approaches (3)

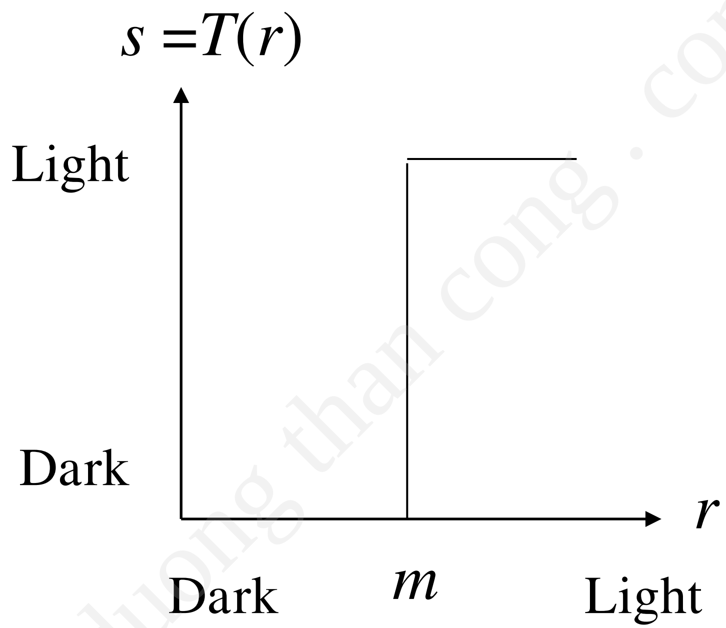
<u>Example</u>: **Contrast stretching**



Producing an image of *higher contrast* than the original, by darkening the levels below *m* and brightening the levels above *m* in the original image.

2. Image Enhancement: Two Approaches (4)

Example: **Producing a binary image**



2. Image Enhancement: Two Approaches (5)

- **Frequency Domain Methods:** Let $g(x,y)$ be a desired image formed by the convolution of an image $f(x,y)$ and a linear, position-invariant operator $h(x,y)$, that is:

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

In frequency domain:

$$G(u,v) = H(u,v)F(u,v)$$

Typically, $f(x,y)$ is given and the goal, after computation of $F(u,v)$, we can select so that the desired image:

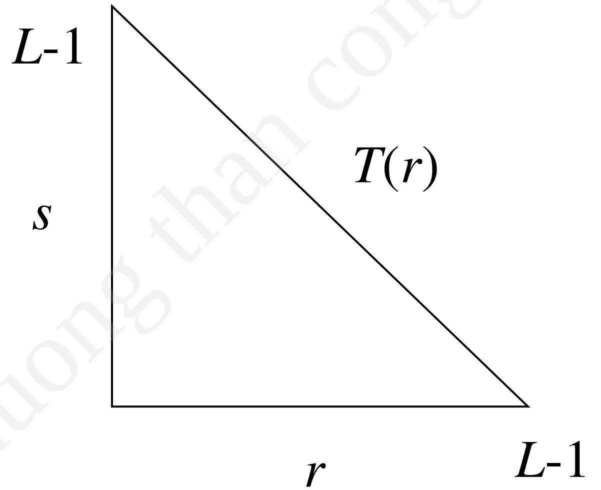
$$g(x,y) = \Im^{-1}\{H(u,v)F(u,v)\}$$

exhibits some highlighted features of $f(x,y)$. For instance, edges in can be enhanced by using $H(u,v)$ as a high pass filter.

Image Enhancement in the Spatial Domain

2. Image Enhancement: Point Processing (1)

- **Intensity transformations:**
  - **Image negatives:** The negative of a digital image is obtained by the transformation function $s = T(r) = L\text{-}1\text{-}r$ where L is the number of grey levels.



Intensity of the output image decreases as the intensity of the input increases. **Applications**: such as displaying medical images.

2. Image Enhancement: Point Processing (2)

Example: Image negative

Original

Negative

2. Image Enhancement: Point Processing (3)

- **Contrast stretching:** Low contrast images occur often due to poor or non uniform lighting conditions, or due to nonlinearity, or small dynamic range of the imaging sensor.

Original

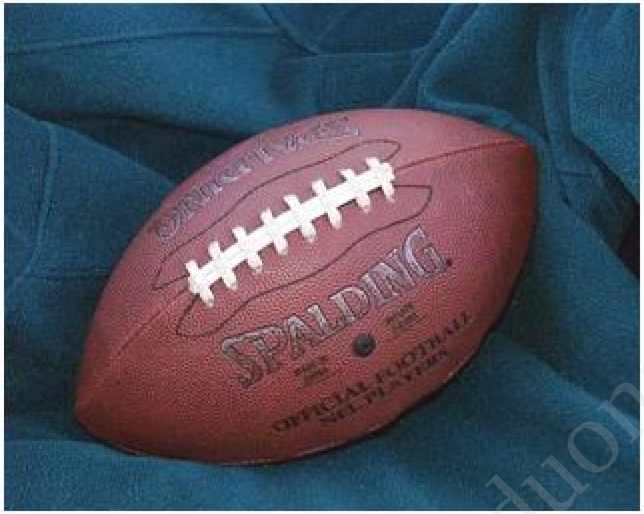
Contrast improvement

2. Image Enhancement: Point Processing (4)

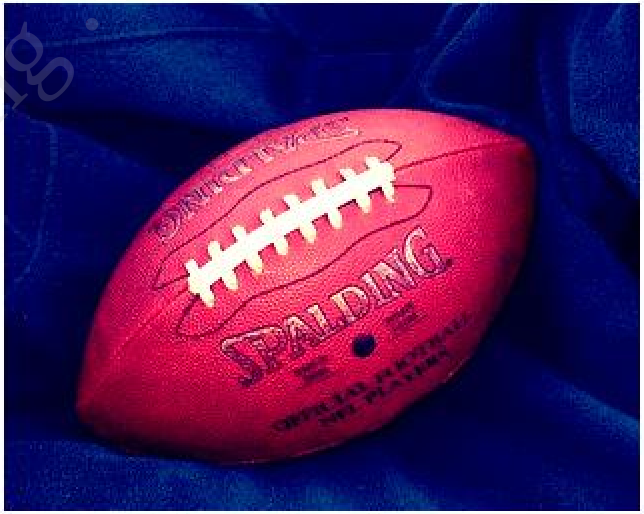
Original

Contrast stretch

2. Image Enhancement: Point Processing (5)



Original



Contrast stretch

2. Image Enhancement: Point Processing (6)

Original

Producing binary image

2. Image Enhancement: Point Processing (7)

❑ <u>Histogram Processing</u>:

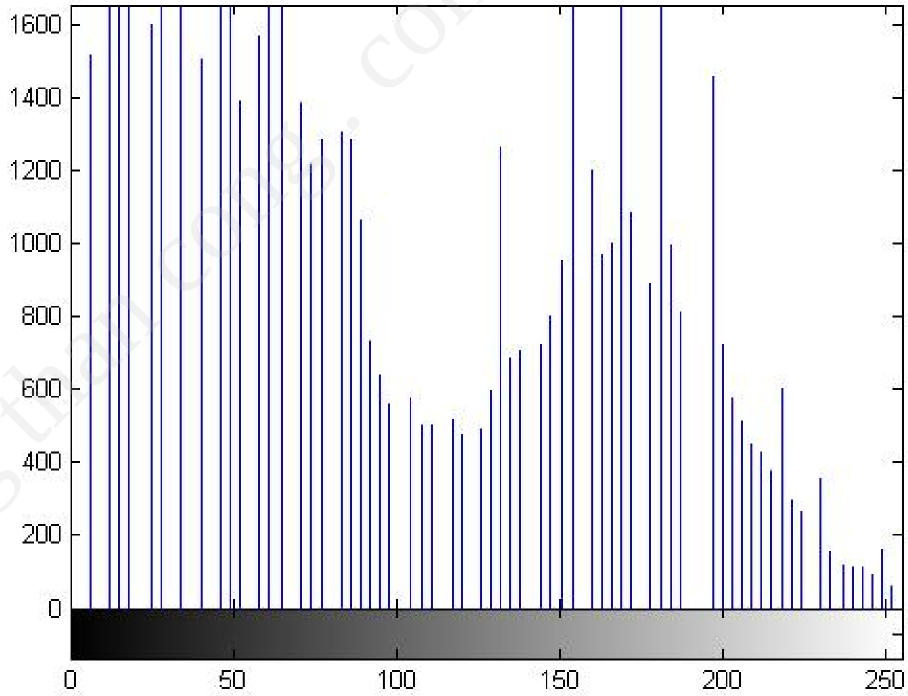
Suppose we have a digital image of size $N \times N$ with gray levels in the range [0, L-1]. The histogram of the image is defined as discrete function:

$$p(r_k) = \frac{n_k}{n}$$

where r_k is the kth gray level, $k = 0,\ldots, L-1$; n_k is the number of pixels in the image with gray level r_k and n is the total number of pixels in the image.

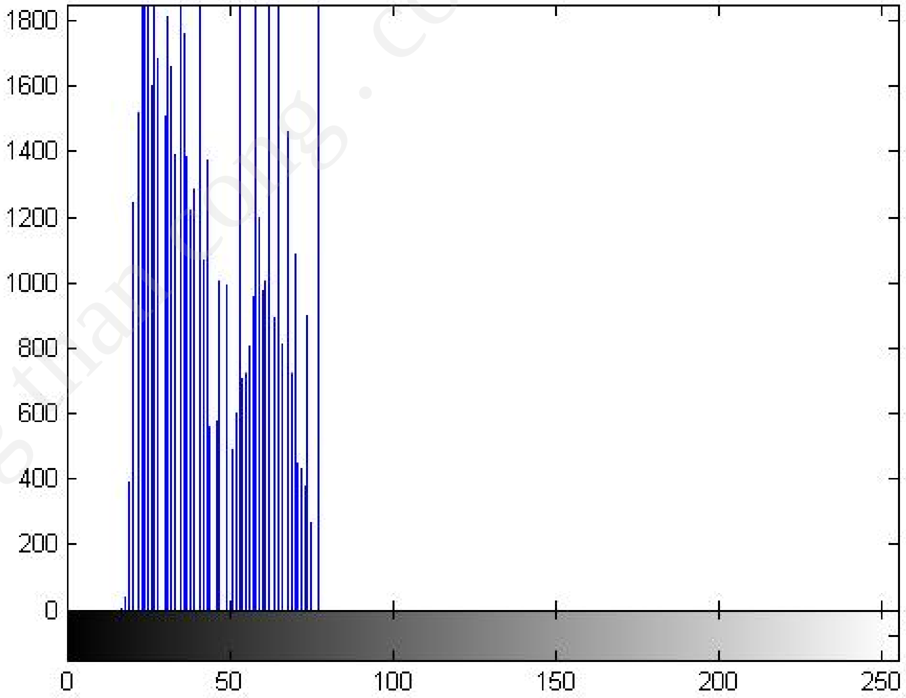
Histogram: Represents the frequency of occurrence of the various gray levels in the image. A plot of this function for all values of k provides a global description of the appearance of the image.

2. Image Enhancement: Point Processing (8)



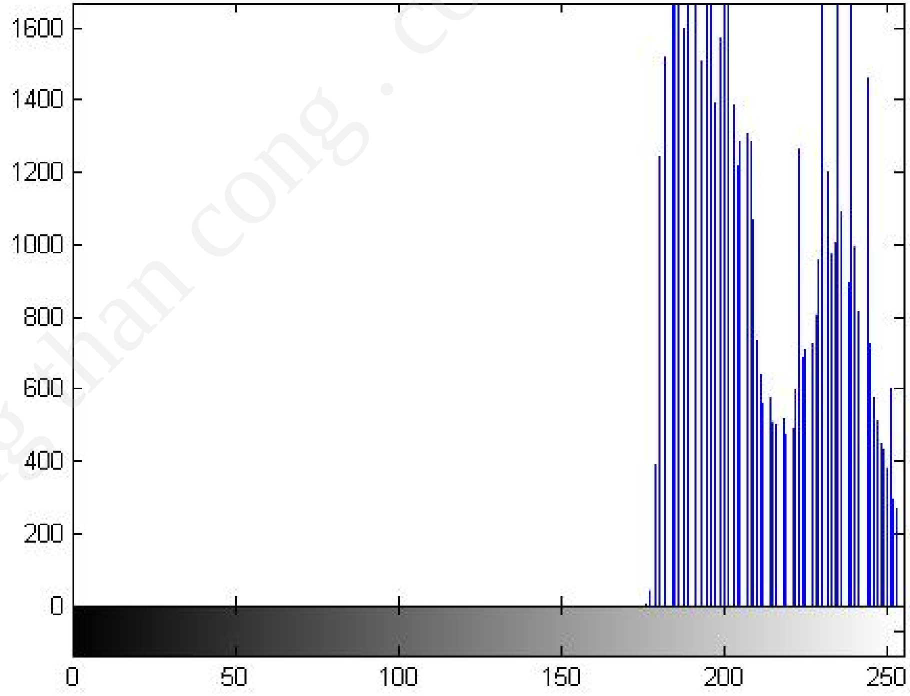
Histogram of high contrast image

2. Image Enhancement: Point Processing (9)



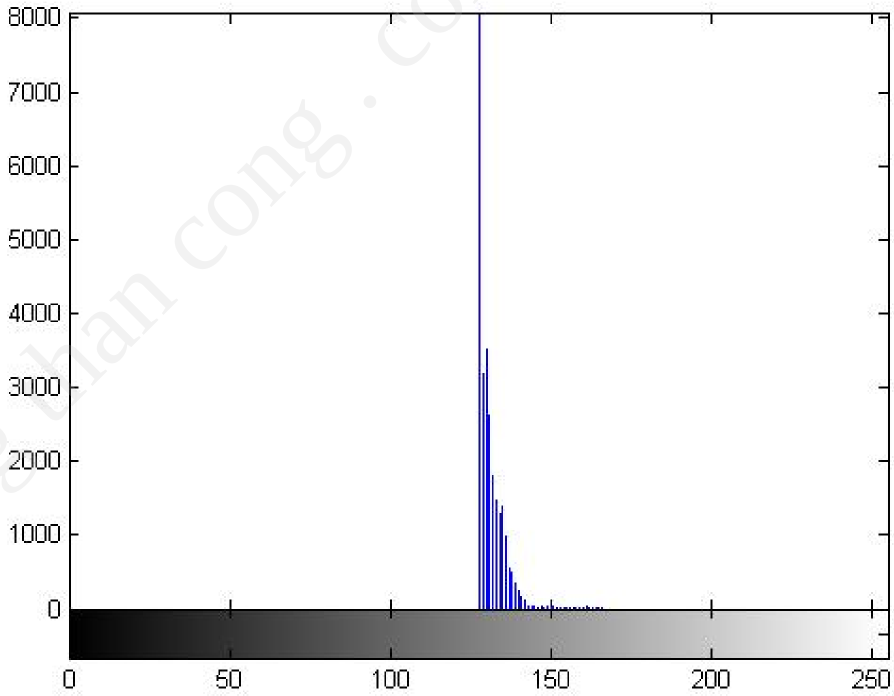
Histogram of dark image

2. Image Enhancement: Point Processing (10)



Histogram of bright image

2. Image Enhancement: Point Processing (11)



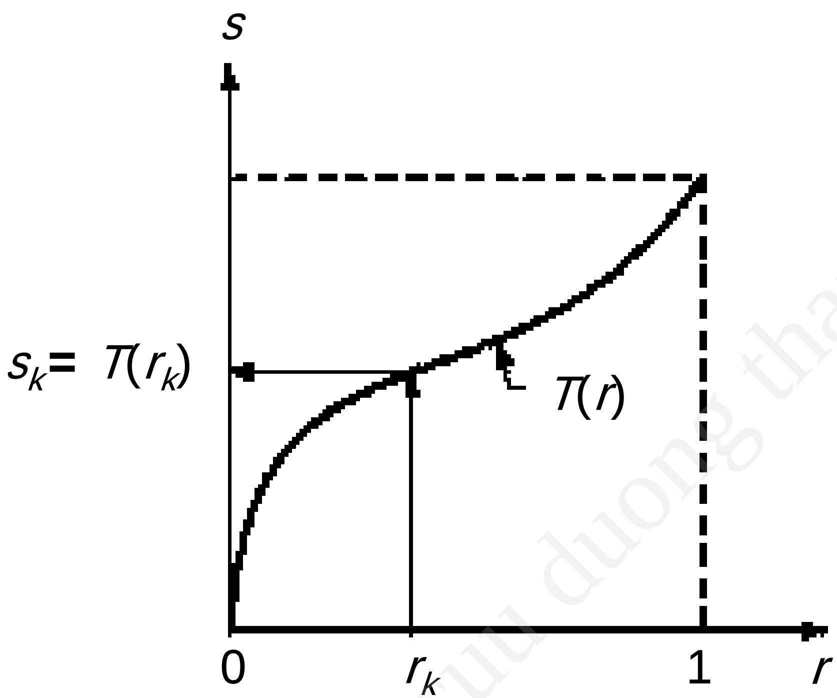
Histogram of low contrast image

2. Image Enhancement: Point Processing (12)

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

2. Image Enhancement: Point Processing (13)

- **Histogram transformation:**



$$s = T(r)$$

where $0 \le r \le 1$;

$T(r)$ satisfies:

- (a). $T(r)$ is single-valued and monotonically increasingly in the interval $0 \le r \le 1$
- (b). $0 \le T(r) \le 1$ for $0 \le r \le 1$

2. Image Enhancement: Point Processing (14)

– Single-valued (one-to-one relationship) guarantees that the inverse transformation will exist.
– Monotonous condition preserves the increasing order from black to white in the output image thus it won't cause a negative image.
– $0 \le T(r) \le 1$ for $0 \le r \le 1$ guarantees that the output gray levels will be in the same range as the input levels.
– The inverse transformation from s back to r is

$$r = T^{-1}(s) \;\; ; \;\; 0 \le s \le 1$$

– The gray levels in an image may be viewed as random variables in the interval [0,1]. PDF (Probability Density Function) is one of the fundamental descriptors of a random variable.

2. Image Enhancement: Point Processing (15)

– If a random variable r is transformed by a monotonic transformation function $T(r)$ to produce a new random variable s, then the PDF of s can be obtained from knowledge of $T(r)$ and the PDF of r, as follows:

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

– Consider a transformation function (also Cumulative Distribution Function (CDF)) of random variable r :

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

2. Image Enhancement: Point Processing (16)

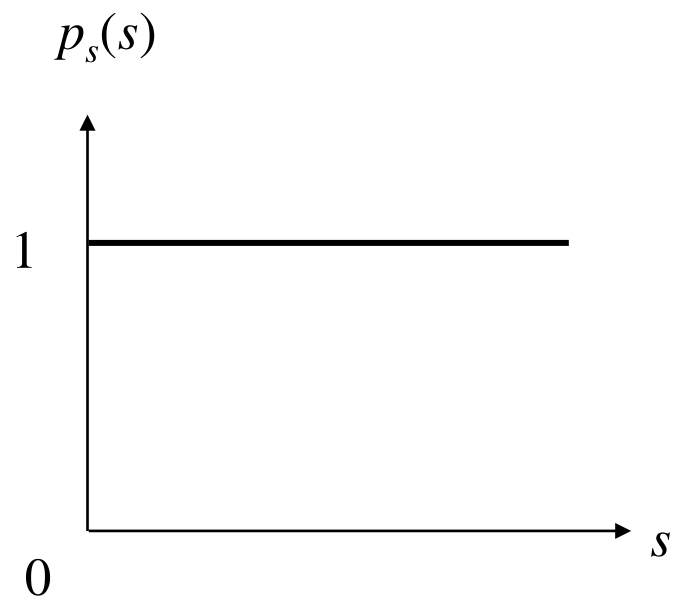
$$\frac{ds}{dr} = \frac{dT(r)}{dr} = \frac{d}{dr}\left[\int_0^r p_r(w)dw\right] = p_r(r)$$

$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right| = p_r(r)\left|\frac{1}{p_r(r)}\right| = 1 \quad \text{where } 0 \le s \le 1$$

- $p_s(s)$ is called as a **uniform probability density function.**
- $p_s(s)$ is always a uniform, independent of the form of $p_r(r)$.

2. Image Enhancement: Point Processing (17)

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



Random variable *s* characterized by a **uniform probability function**

2. Image Enhancement: Point Processing (18)

- **Discrete transformation function:**
  - The probability of occurrence of gray level in an image is approximated by

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

  - The discrete version of transformation

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

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

2. Image Enhancement: Point Processing (19)

- **<u>Histogram Equalization</u>:**
  - Thus, an output image is obtained by mapping each pixel with level r_k in the input image into a corresponding pixel with level s_k in the output image.

$$s_k = T(r_k) = \sum_{j=0}^{k} \frac{n_j}{N^2} = \sum_{j=0}^{k} p_r(r_j),\ 0 \le r_k \le 1,\ k = 0,1,\ldots,L-1$$

  - In discrete space, it cannot be proved in general that this discrete transformation will produce the discrete equivalent of a uniform probability density function, which would be a uniform histogram.

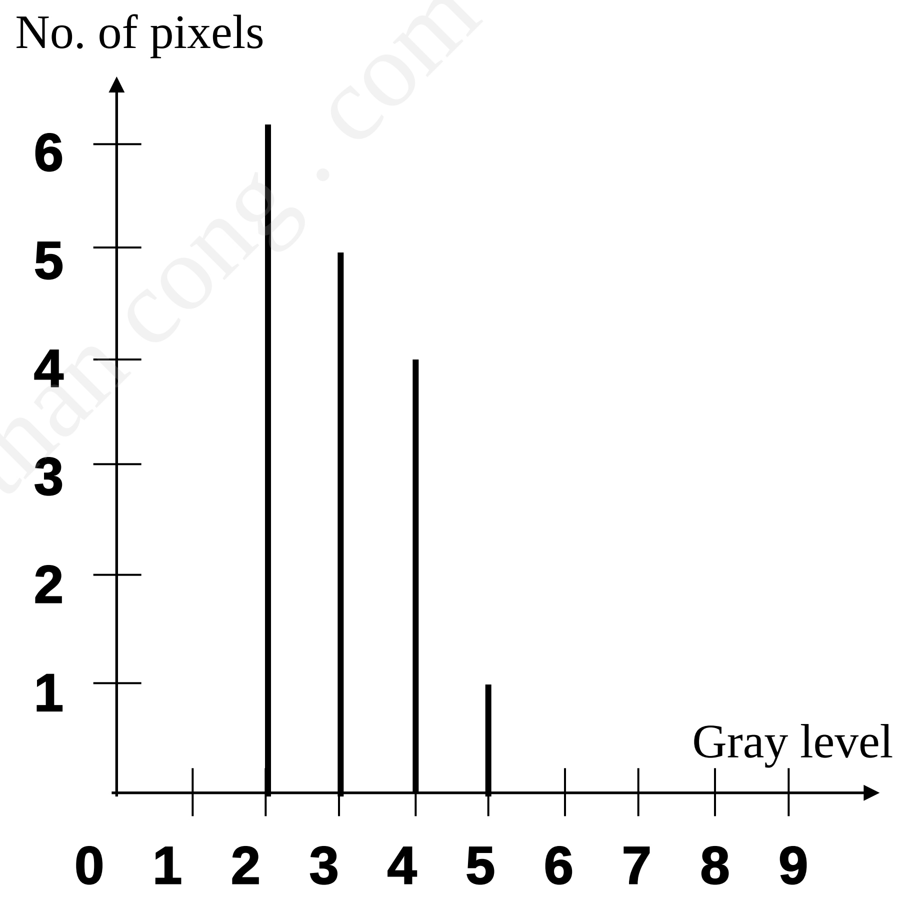
2. Image Enhancement: Point Processing (20)

Example of Histogram Equalization:

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

4x4 image

Gray scale = [0,9]



Histogram

2. Image Enhancement: Point Processing (21)

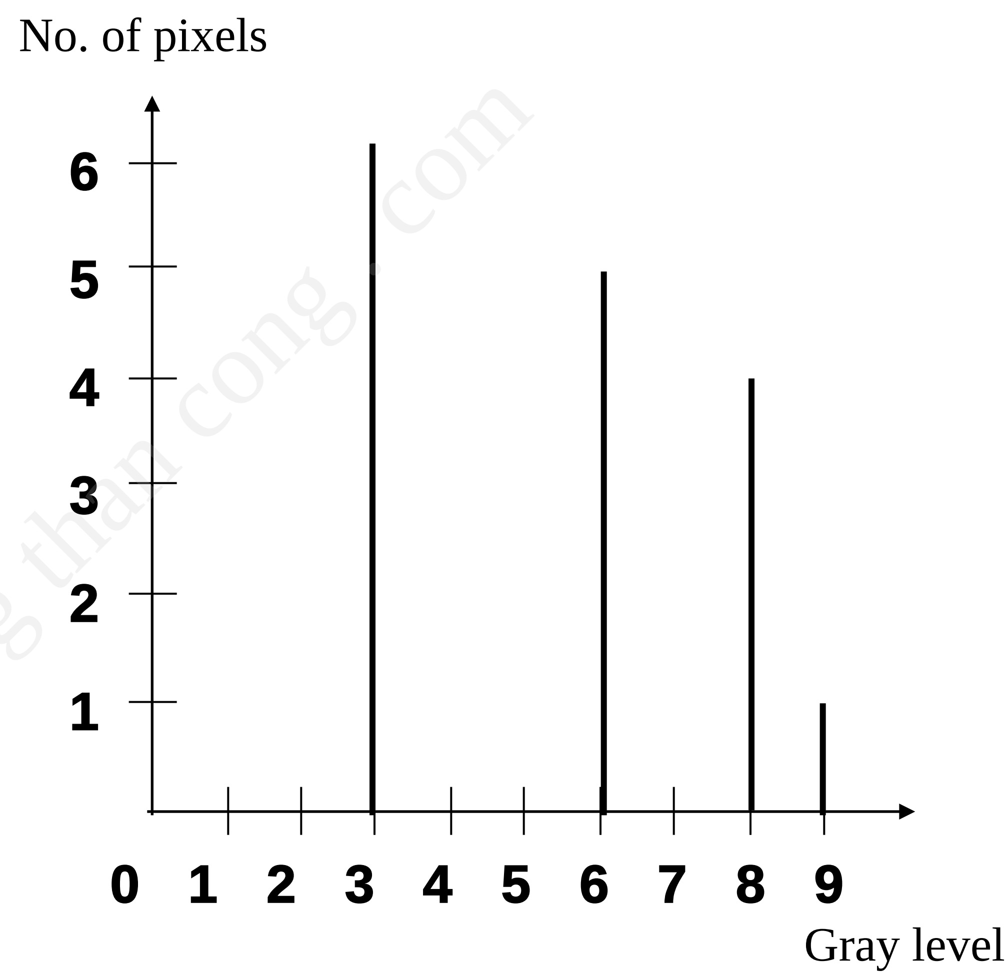
Gray Level (j)	0	1	2	3	4	5	6	7	8	9
No. of pixels	0	0	6	5	4	1	0	0	0	0
$\sum_{j=0}^{k} n_j$	0	0	6	11	15	16	16	16	16	16
$s = \sum_{j=0}^{k} \frac{n_j}{n}$	0	0	6/16	11/ 16	15/ 16	16/ 16	16/ 16	16/ 16	16/ 16	16/ 16
$s \times 9$	0	0	3.3 ≈3	6.1 ≈6	8.4 ≈8	9	9	9	9	9

2. Image Enhancement: Point Processing (22)

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

Output image

Gray scale = [0, 9]



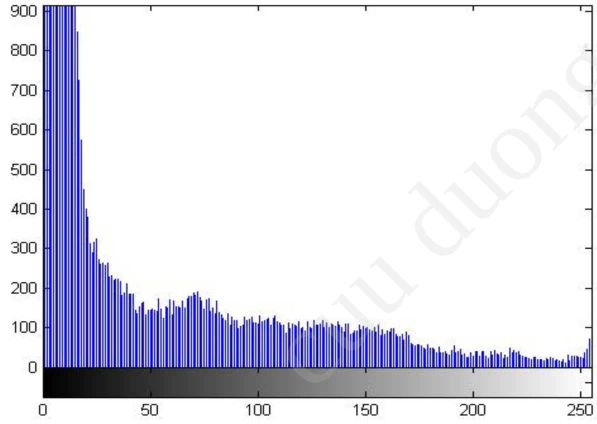
Histogram equalization

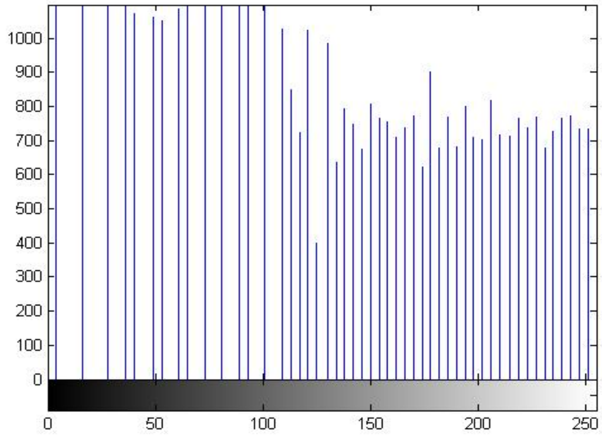
2. Image Enhancement: Point Processing (23)

Comments:

- Histogram equalization distributes the gray level to reach the maximum gray level (white) because the cumulative distribution function equals 1 when $0 \le r \le L$-1. This process **increases the dynamic range** of gray levels and produces an **increase in image contrast.**
- If the cumulative numbers of gray levels are slightly different, they will be mapped to little different or same gray levels as we may have to approximate the processed gray level of the output image to integer number.
- Thus, the discrete transformation function can not guarantee the one-to-one mapping relationship.

2. Image Enhancement: Point Processing (24)





Original

Using histogram equalization

2. Image Enhancement: Point Processing (25)

- **<u>Histogram Specification</u>:** Used to specify a particular histogram shape (not necessarily uniform) which is capable of highlighting certain gray levels in the image.

  Let us suppose that:

  $p_r(r)$ is the original probability density function,

  $p_z(z)$ is the desired probability density function.

  Suppose that the histogram equalisation is first applied on the original image *r*

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

2. Image Enhancement: Point Processing (26)

Suppose that the desired image *r* is available and histogram equalisation is applied as well.

$$v = G(z) = \int_0^z p_z(w)dw$$

and $p_s(s)$, $p_v(v)$ are both uniform densities and they can be considered as identical because the final result of histogram equalisation is independent of the density inside the integral. So in equation

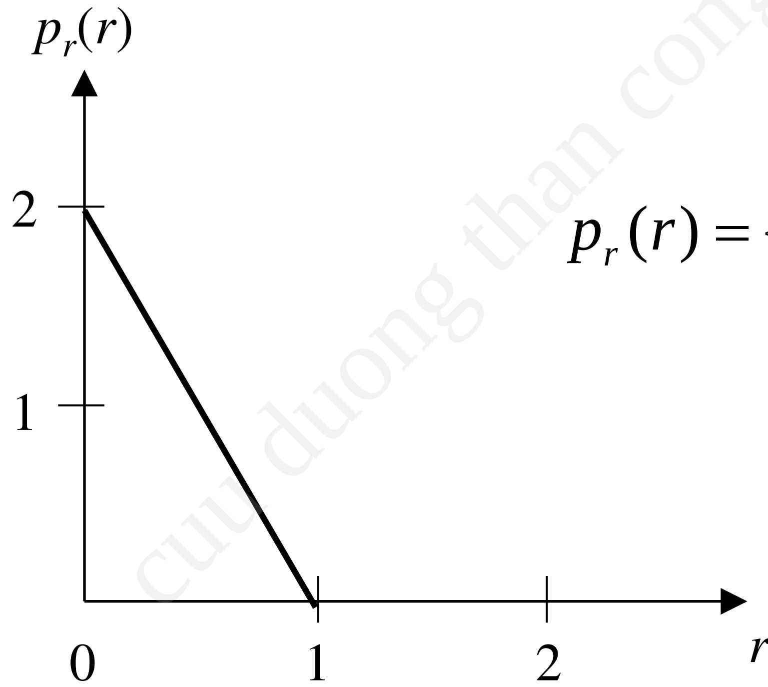
$$v = G(z) = \int_0^z p_z(w)dw$$

we can use the symbol *s* instead of *v*. The inverse process $z = G^{-1}(s)$ will have the desired probability density function.

2. Image Enhancement: Point Processing (27)

Example of Histogram Specification:

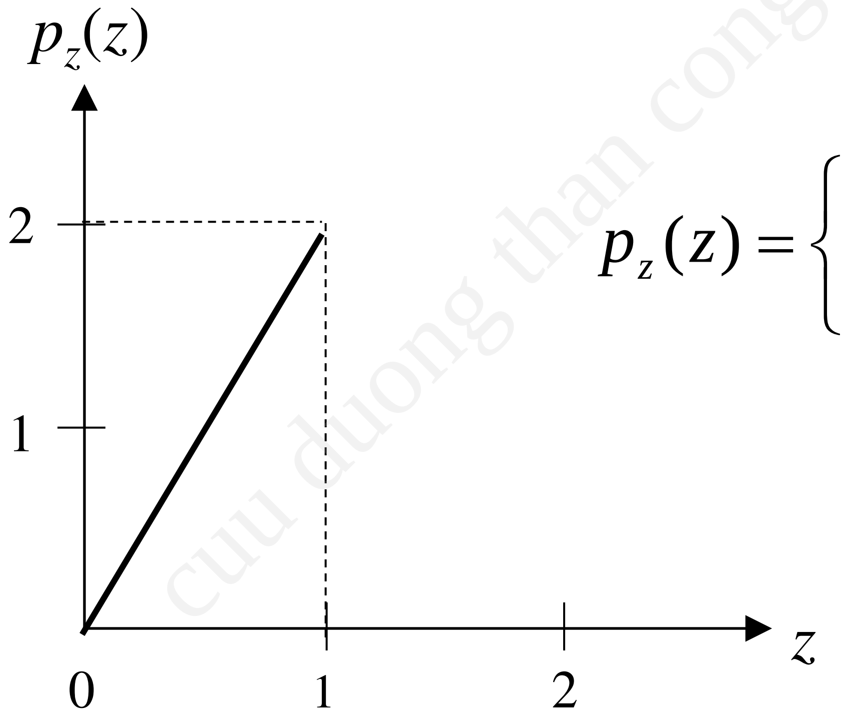
– Assume an image has a gray level probability density function $p_r(r)$ as shown.



$$p_r(r) = \begin{cases} -2r + 2 & ;0 \le r \le 1 \\ 0 & ;\text{elsewhere} \end{cases}$$

2. Image Enhancement: Point Processing (28)

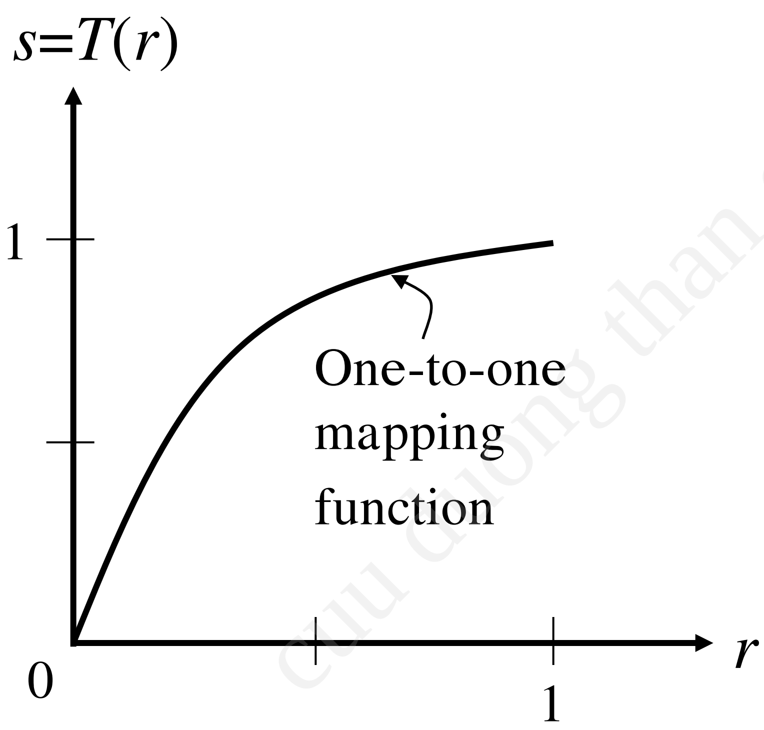
– We would like to apply the histogram specification with the desired probability density function $p_z(z)$ as shown:



$$p_z(z) = \begin{cases} 2z & ; 0 \le z \le 1 \\ 0 & ; \text{elsewhere} \end{cases}$$

2. Image Enhancement: Point Processing (29)

Step 1: Obtain the transformation function $T(r)$



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

$$= \int_0^r (-2w+2)dw$$

$$= -w^2 + 2w\Big|_0^r$$

$$= -r^2 + 2r$$

2. Image Enhancement: Point Processing (30)

Step 2: Obtain the transformation function $G(z)$

$$G(z) = \int_0^z (2w)dw \quad = z^2\Big|_0^z \quad = z^2$$

Step 3: Obtain the inversed transformation function G^{-1}

$$z = G^{-1}(s) = G^{-1}[T(r)]$$

$$G(z) = T(r)$$

$$z^2 = -r^2 + 2r$$

$$z = \sqrt{2r - r^2}$$

We can guarantee that $0 \le z \le 1$ when $0 \le r \le 1$

– Discrete formulation:

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

$$= \sum_{j=0}^{k} \frac{n_j}{n} \qquad k = 0,1,2,\ldots,L-1$$

$$G(z_k) = \sum_{i=0}^{k} p_z(z_i) = s_k \qquad k = 0,1,2,\ldots,L-1$$

$$z_k = G^{-1}\left[T(r_k)\right]$$

$$= G^{-1}\left[s_k\right] \qquad k = 0,1,2,\ldots,L-1$$

2. Image Enhancement: Spatial Filtering (1)

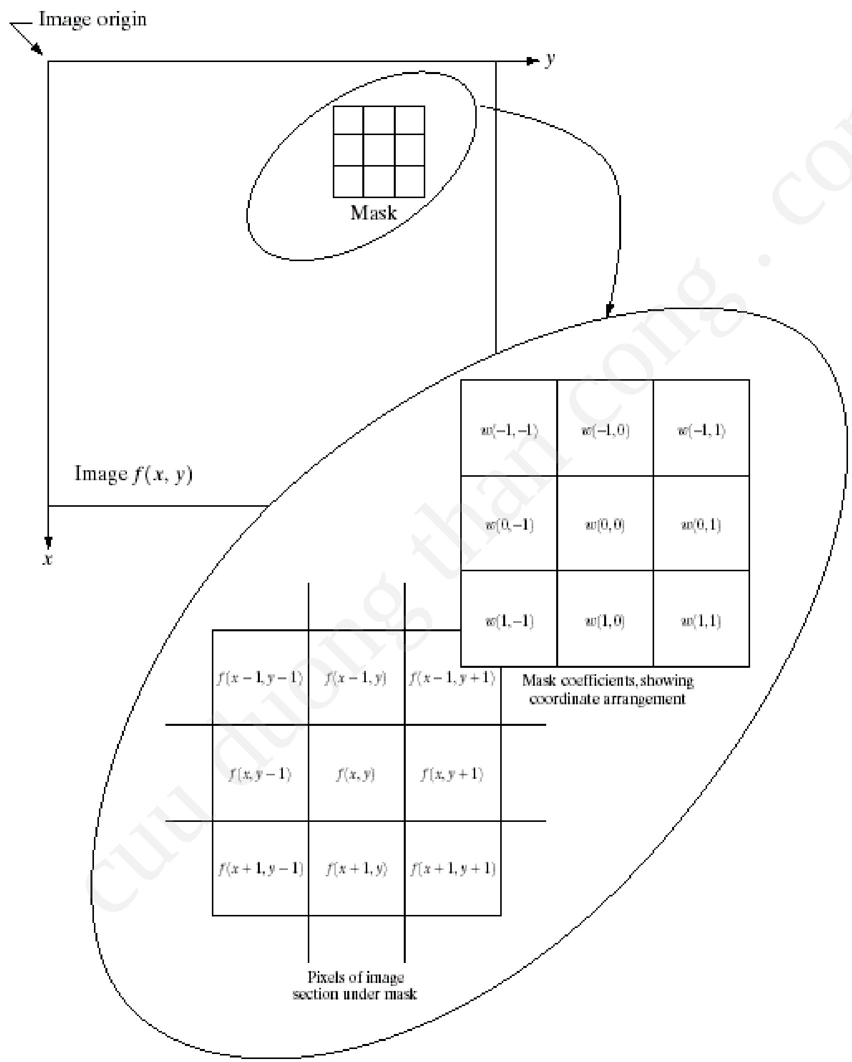


FIGURE 3.32 The mechanics of spatial filtering. The magnified drawing shows a 3 × 3 mask and the image section directly under it; the image section is shown displaced out from under the mask for ease of readability.

2. Image Enhancement: Spatial Filtering (2)

- Using filter (can also be called as mask/ kernel/ template or window)
- The values in a filter subimage are referred to as coefficients, rather than pixel. Our focus will be on masks of odd sizes, e.g. 3×3, 5×5,…
- Simply move 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.

$$R = w_1 z_1 + w_2 z_2 + \ldots + w_{mn} z_{mn} = \sum_{i=i}^{mn} w_i z_i$$

2. Image Enhancement: Spatial Filtering (3)

- **<u>Smoothing Spatial Filters</u>:**
  - Used for blurring and for noise reduction.
  - **Blurring** is used in preprocessing steps, such as:
    - Removal of small details from an image prior to object extraction.
    - Bridging of small gaps in lines or curves.
  - **Noise reduction** can be accomplished by blurring with a linear filter and also by a nonlinear filter.
  - Output is simply the average of the pixels contained in the neighborhood of the filter mask.
  - Called averaging filters or lowpass filters.

2. Image Enhancement: Spatial Filtering (4)

– Replace the value of every pixel in an image by the average of the gray levels in the neighborhood will reduce the “sharp” transitions in gray levels.
– Sharp transitions:
  - Random noise in the image.
  - Edges of objects in the image.
– Thus, smoothing can reduce noises (desirable) and blur edges (undesirable).

2. Image Enhancement: Spatial Filtering (5)

3×3 Smoothing Spatial Filters:

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

1	1	1
1	1	1
1	1	1

Average

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

1	2	1
2	4	2
1	2	1

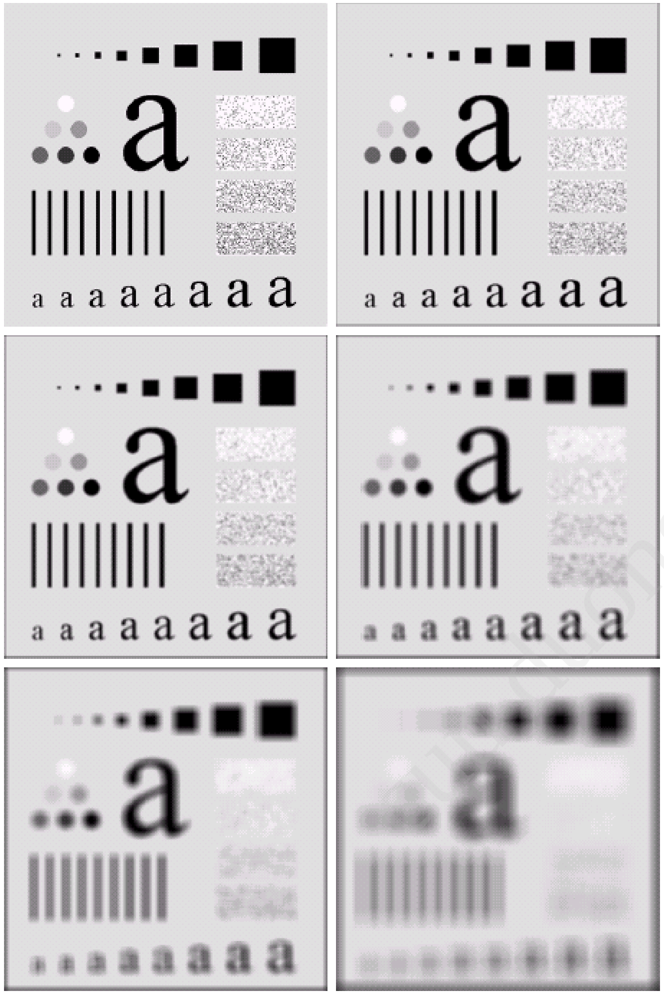
Weighted average

a b

FIGURE 3.34 Two 3 × 3 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 an average.

Weighted average filter: weighting the center point the highest and then reducing the value of the coefficients as a function of increasing distance from the origin is simply **an attempt to reduce blurring in the smoothing process.**

Example of Smoothing Spatial Filters:



a	b
c	d
e	f

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

2. Image Enhancement: Spatial Filtering (7)

- **<u>Order-Statistics Filters (Nonlinear Filters):</u>**
  - The response is based on ordering (ranking) the pixels contained in the image area encompassed by the filter.
  - Example:
    - Median filter: $R = \text{median}\{z_k \,|\, k = 1,2,\ldots,n \times n\}$
    - Max filter: $R = \max\{z_k \,|\, k = 1,2,\ldots,n \times n\}$
    - Min filter: $R = \min\{z_k \,|\, k = 1,2,\ldots,n \times n\}$

  where: $n \times n$ is the size of the mask.

2. Image Enhancement: Spatial Filtering (8)

Median Filter:

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

2. Image Enhancement: Spatial Filtering (9)

Example of Median Filter:

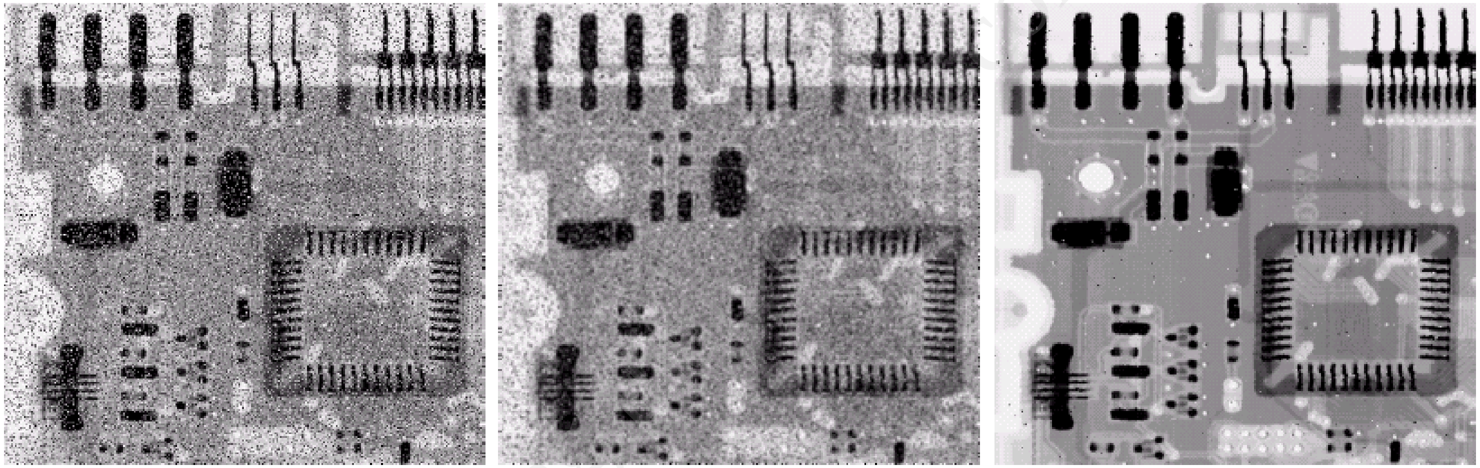


FIGURE 3.37 (a) X-ray image of circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a 3 × 3 averaging mask. (c) Noise reduction with a 3 × 3 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

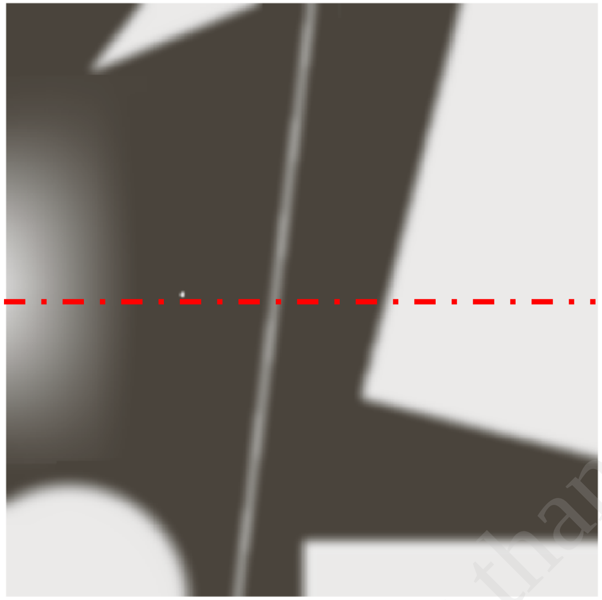
2. Image Enhancement: Spatial Filtering (10)

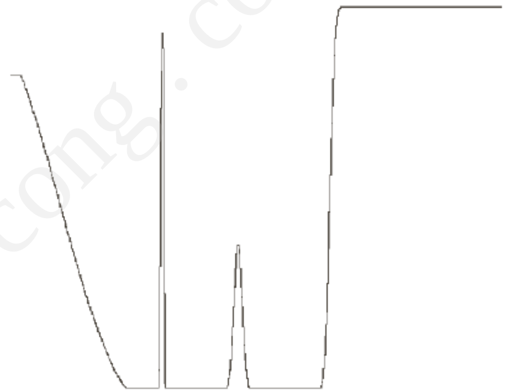
- **Sharpening Spatial Filters:**
  - To highlight fine detail in an image.
  - or to enhance detail that has been blurred, either in error or as a natural effect of a particular method of image acquisition.
  - It is known that **blurring** can be done in spatial domain by pixel **averaging** in a neighbors.
  - Since averaging is analogous to **integration.**
  - Thus, we can guess that the sharpening must be accomplished by **spatial differentiation.**

2. Image Enhancement: Spatial Filtering (11)

- **<u>Derivative operator</u>:**
  - 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.
  - Thus, image differentiation:
    - **enhances edges** and other **discontinuities** (e. g. noise)
    - **deemphasizes** area with **slowly varying gray-level values**.

2. Image Enhancement: Spatial Filtering (12)





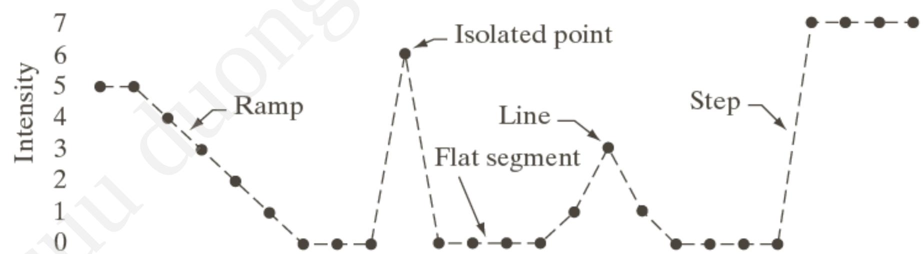


Image strip	5	5	4	3	2	1	0	0	0	6	0	0	0	0	1	3	1	0	0	0	0	7	7	7	7	•	•
First derivative		−1	−1	−1	−1	−1	0	0	6	−6	0	0	0	1	2	−2	−1	0	0	0	7	0	0	0			
Second derivative		−1	0	0	0	0	1	0	6	−12	6	0	0	1	1	−4	1	1	0	0	7	−7	0	0			

2. Image Enhancement: Spatial Filtering (13)

First-order derivative:

– A basic definition of the first-order derivative of a one-dimensional function *f*(*x*) is the difference:

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

Second-order derivative:

– Similarly, we define the second-order derivative of a one-dimensional function *f*(*x*) is the difference:

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

2. Image Enhancement: Spatial Filtering (14)

First and Second-order derivative of *f*(*x*,*y*):

– 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

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

Laplacian operator

(linear operator)

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

2. Image Enhancement: Spatial Filtering (15)

From:

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

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

yield,

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

Result Laplacian mask and its extensions

0	1	0
1	–4	1
0	1	0

1	1	1
1	−8	1
1	1	1

0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−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.

2. Image Enhancement: Spatial Filtering (17)

- **<u>Mask of Laplacian + Addition</u>:** we can create a mask which do both operations, Laplacian Filter and Addition the original image.

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

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

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

2. Image Enhancement: Spatial Filtering (18)

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

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

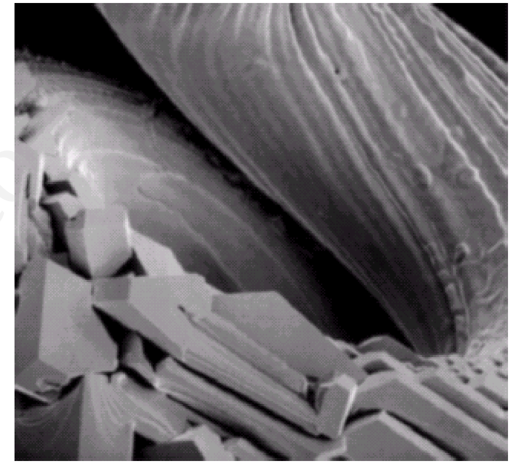
$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 9 & -1 \\ 0 & -1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & -1 & 0 \\ -1 & 8 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

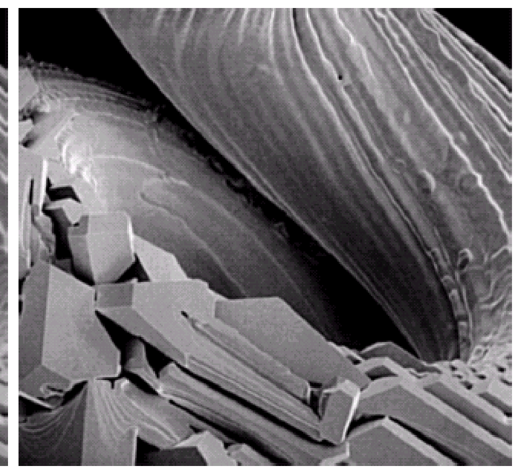
2. Image Enhancement: Spatial Filtering (19)

Example of composite Laplacian mask (Laplacian+ Addition)

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.)

- **Unsharp masking:**

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

Sharpened image = Original image – Blurred image

- **High-boost filtering:**
  - Generalized form of unsharp masking ($A \geq 1$):

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

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

2. Image Enhancement: Spatial Filtering (21)

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

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

yields:

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

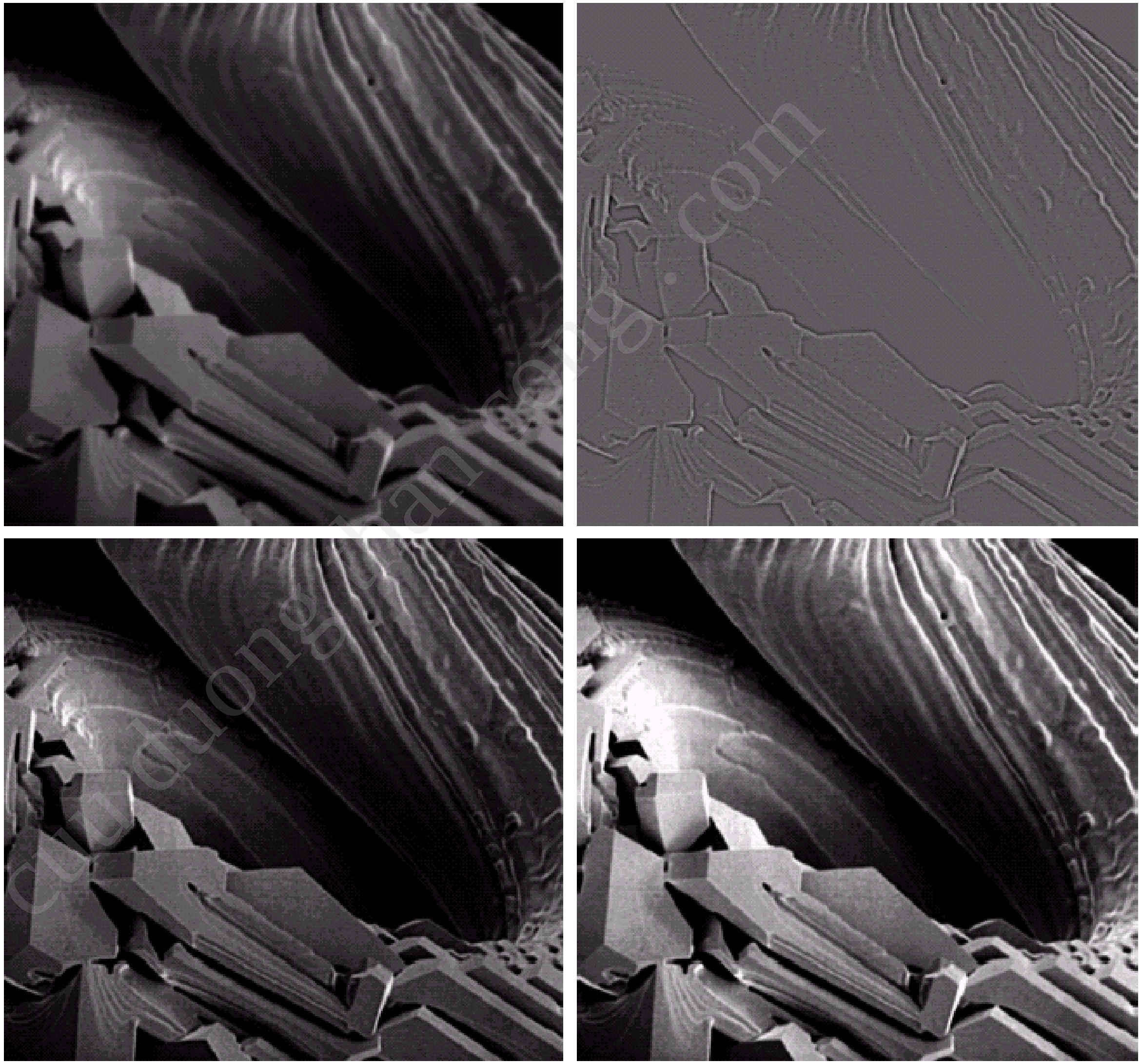
2. Image Enhancement: Spatial Filtering (22)

0	−1	0		−1	−1	−1
−1	$A+4$	−1		−1	$A+8$	−1
0	−1	0		−1	−1	−1

$A \geq 1$

If $A = 1$, it becomes "standard" Laplacian sharpening.

2. Image Enhancement: Spatial Filtering (23)



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 $A = 1.7$.

2. Image Enhancement: Spatial Filtering (24)

- **Gradient operator**:
  - First derivatives are implemented using the **magnitude of the gradient**.

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

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

$$= \left[\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right]^{1/2} \Rightarrow \nabla f \approx |G_x| + |G_y|$$

commonly approx.

the magnitude becomes nonlinear

2. Image Enhancement: Spatial Filtering (25)

Gradient masks:

– **Roberts cross-gradient operators** (2×2):

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

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

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

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

−1	0
0	1

0	−1
1	0

2. Image Enhancement: Spatial Filtering (26)

– **Sobel operators** (3×3):

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

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

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

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

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

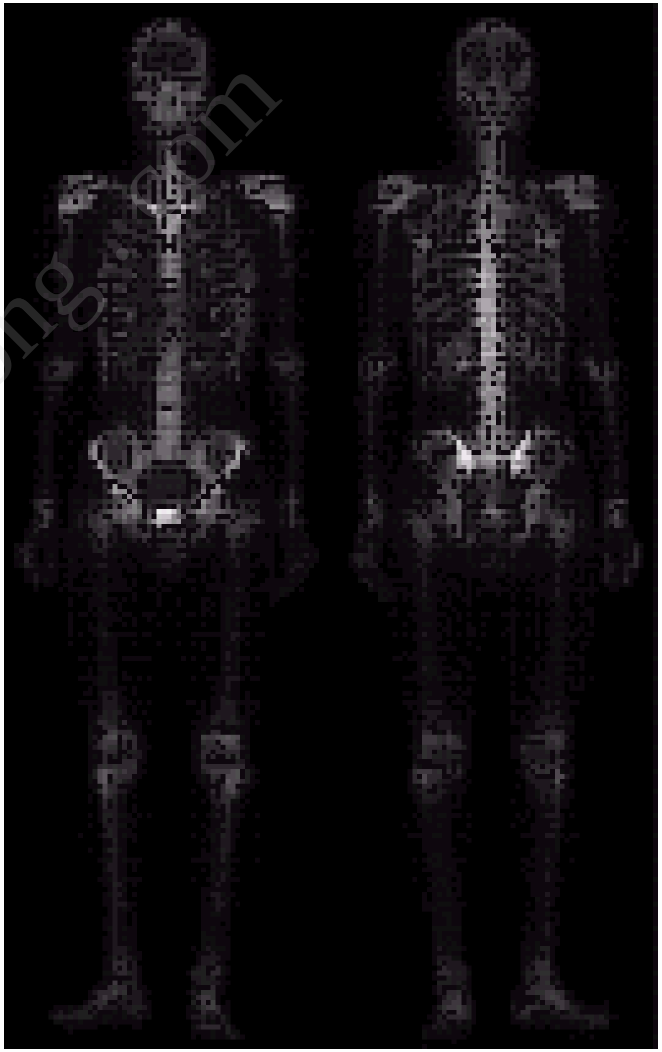
2. Image Enhancement: Spatial Filtering (27)

– **Note**: the **summation of coefficients in all masks equals 0**, indicating that they would give a response of 0 in an area of constant gray level.

2. Image Enhancement: Spatial Filtering (28)

Example of combining spatial enhancement methods:

- We 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.



2. Image Enhancement: Spatial Filtering (29)

– **Solve**:

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

2. Image Enhancement: Spatial Filtering (30)

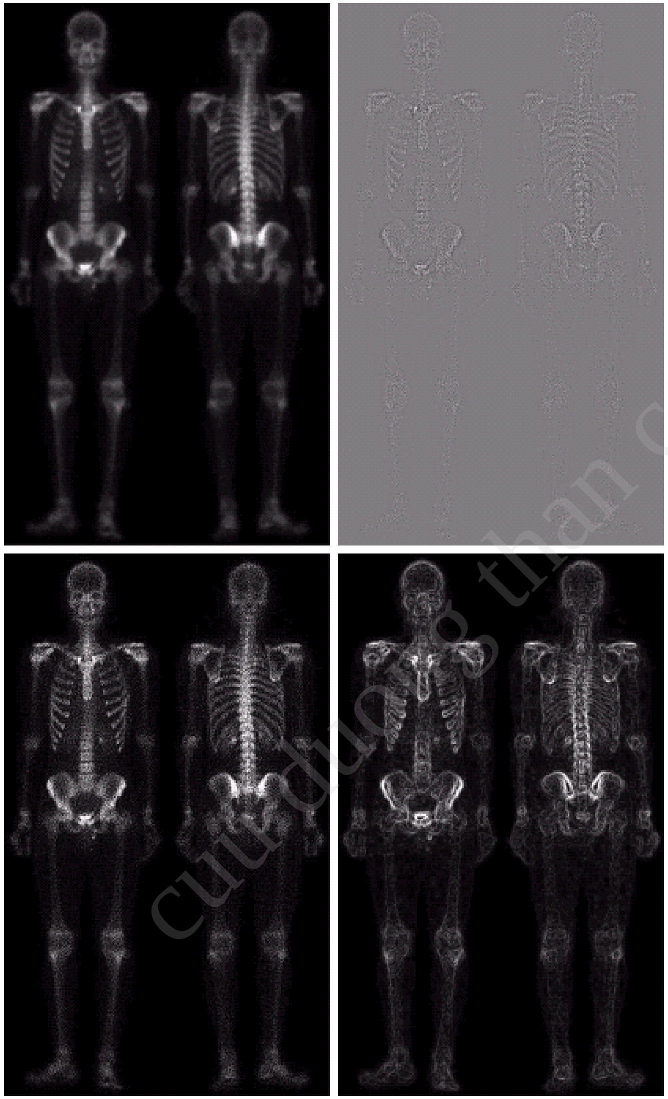
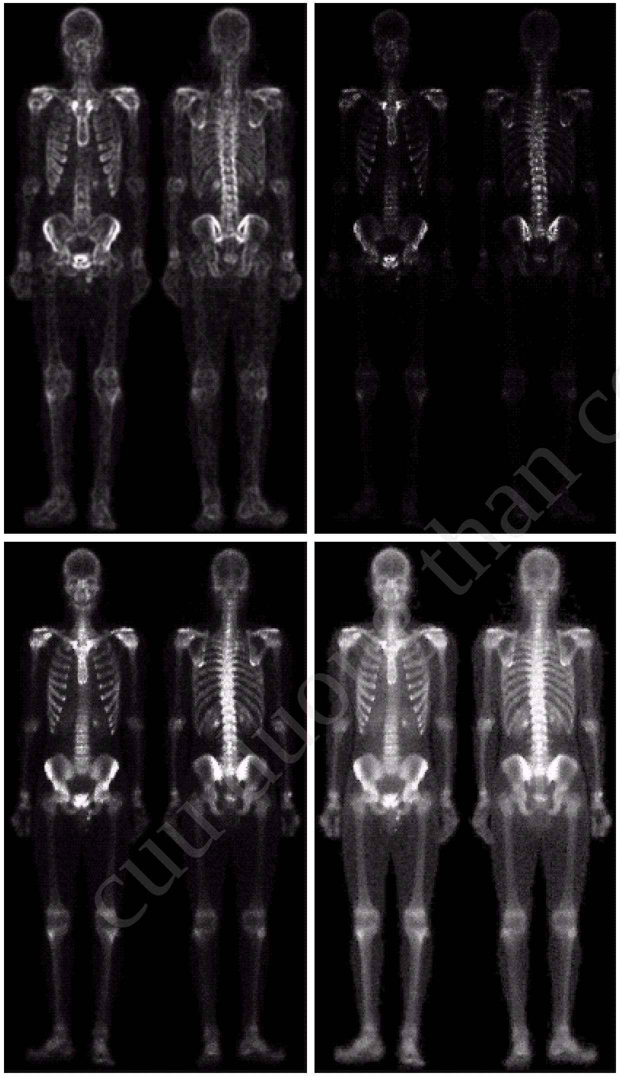


FIGURE 3.46 (a) Image of whole body bone scan. (b) Laplacian of (a). (c) Sharpened image obtained by adding (a) and (b). (d) Sobel of (a).

2. Image Enhancement: Spatial Filtering (31)



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.)

Image Enhancement in the Frequency Domain

2. Image Enhancement: Frequency Domain (1)

Basic steps for filtering in the frequency domain:

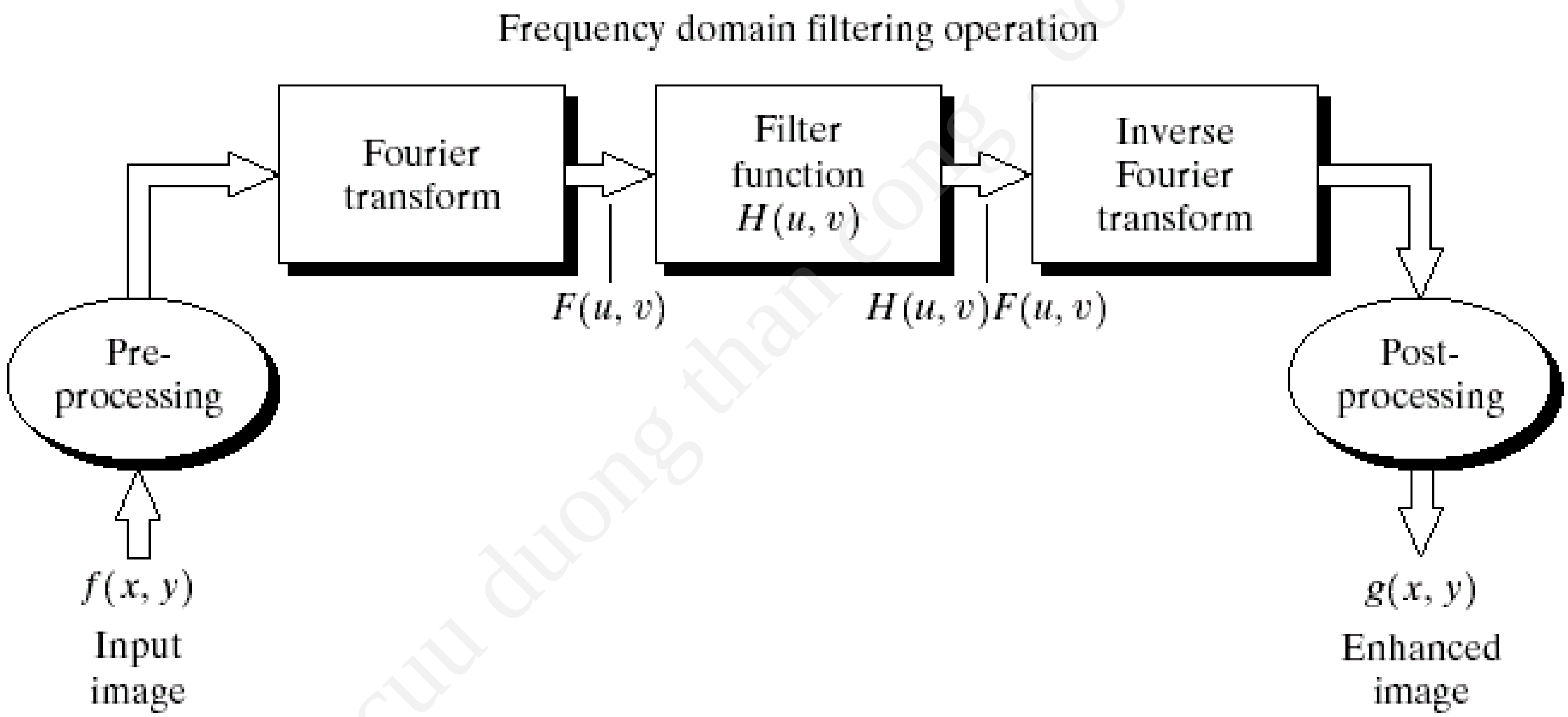


FIGURE 4.5 Basic steps for filtering in the frequency domain.

2. Image Enhancement: Frequency Domain (2)

Basic steps for 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.

2. Image Enhancement: Frequency Domain (3)

Example of Low pass filter and High pass filter:

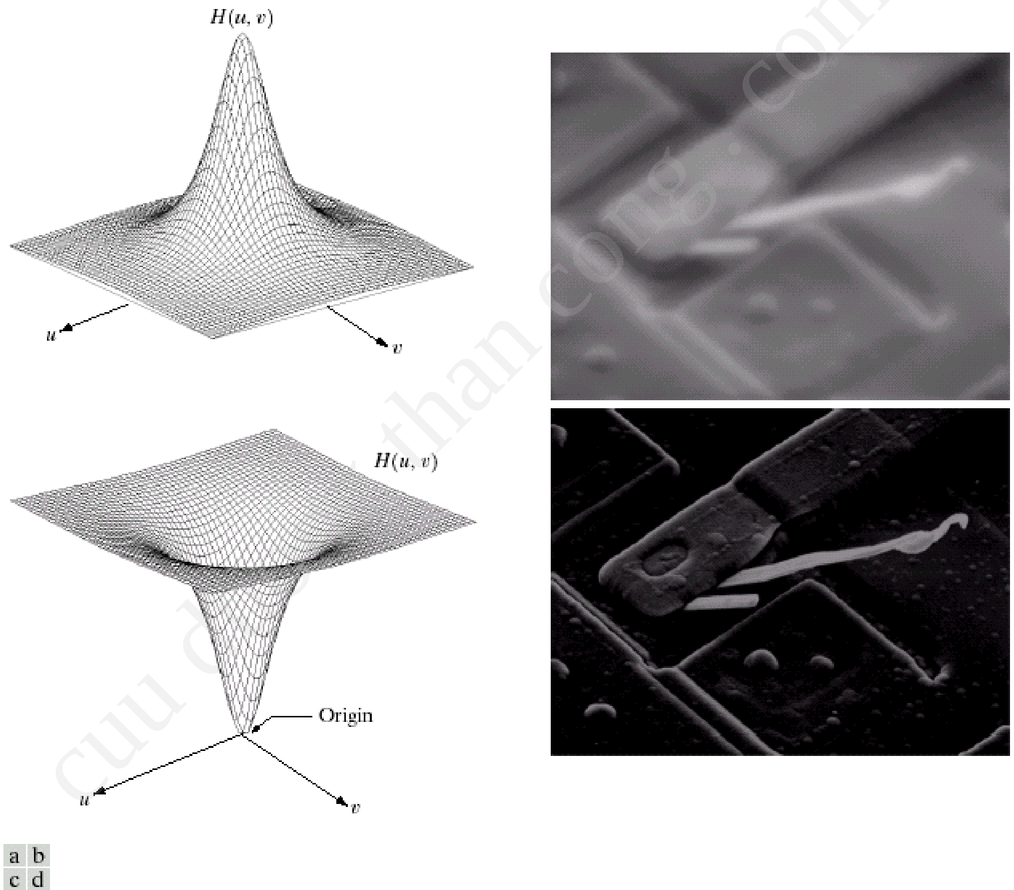
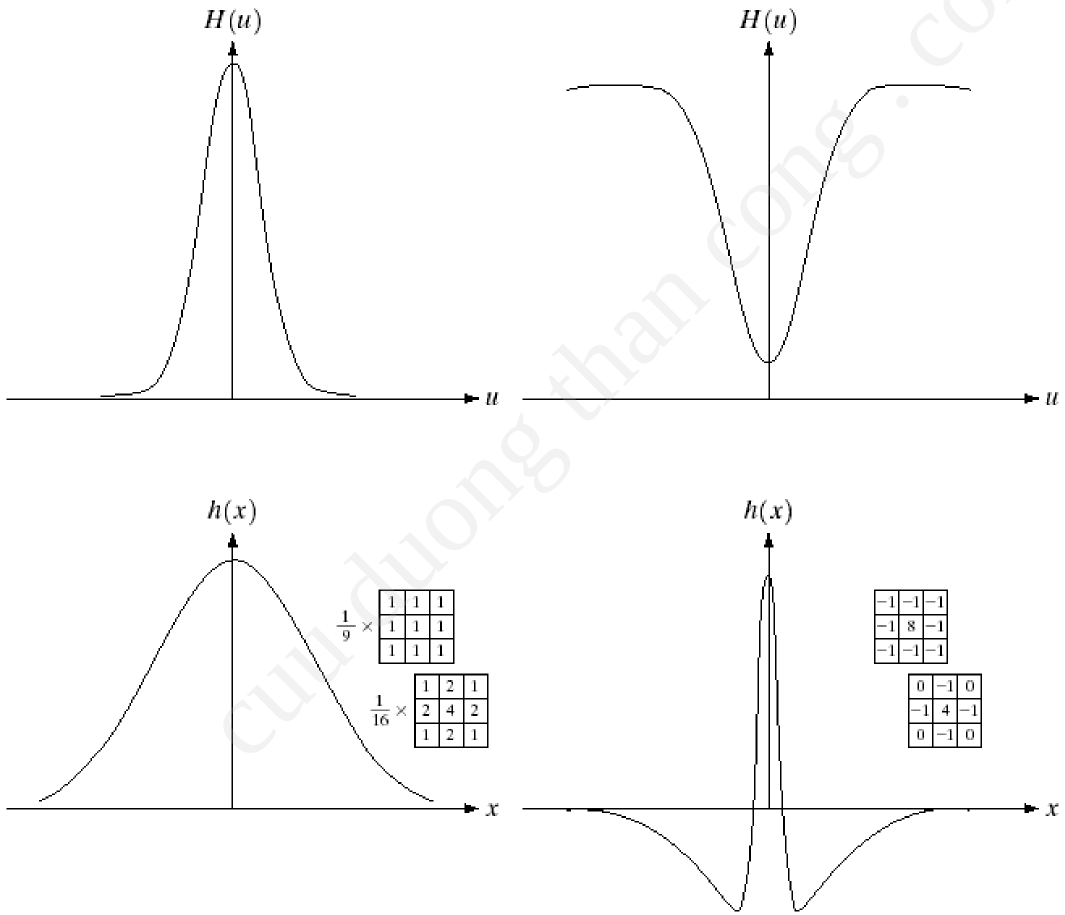


FIGURE 4.7 (a) A two-dimensional lowpass filter function. (b) Result of lowpass filtering the image in Fig. 4.4(a). (c) A two-dimensional highpass filter function. (d) Result of highpass filtering the image in Fig. 4.4(a).

Correspondence between filter in spatial and frequency domains:



a b
c d

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.

Smoothing Frequency-domain filters:
Ideal Lowpass filter

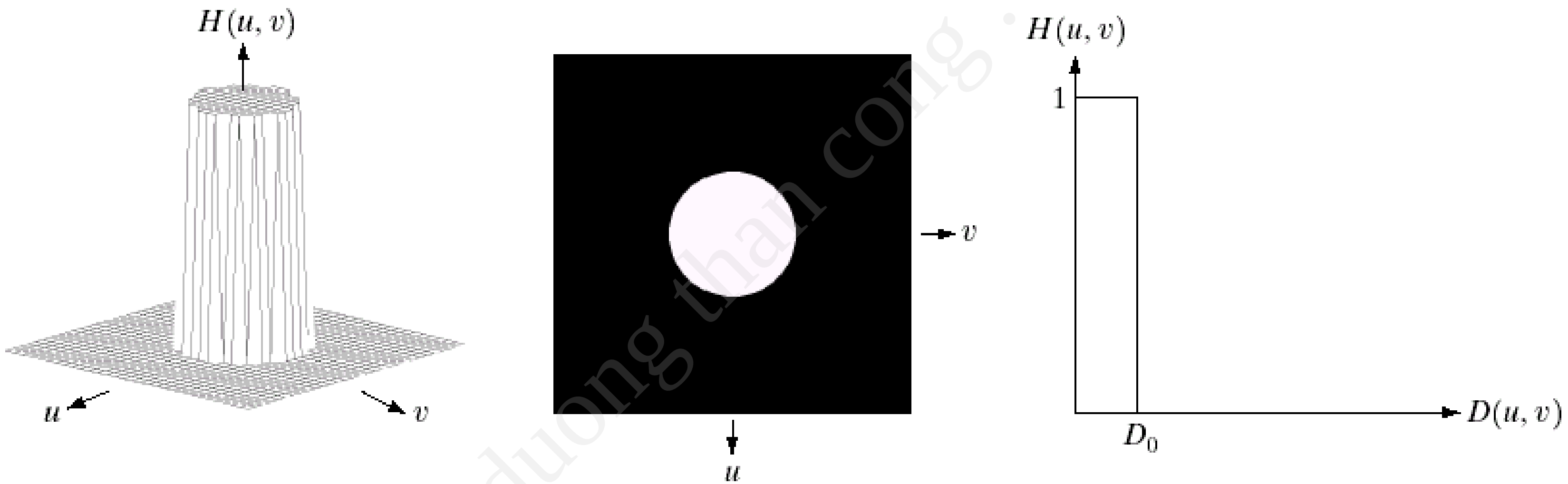
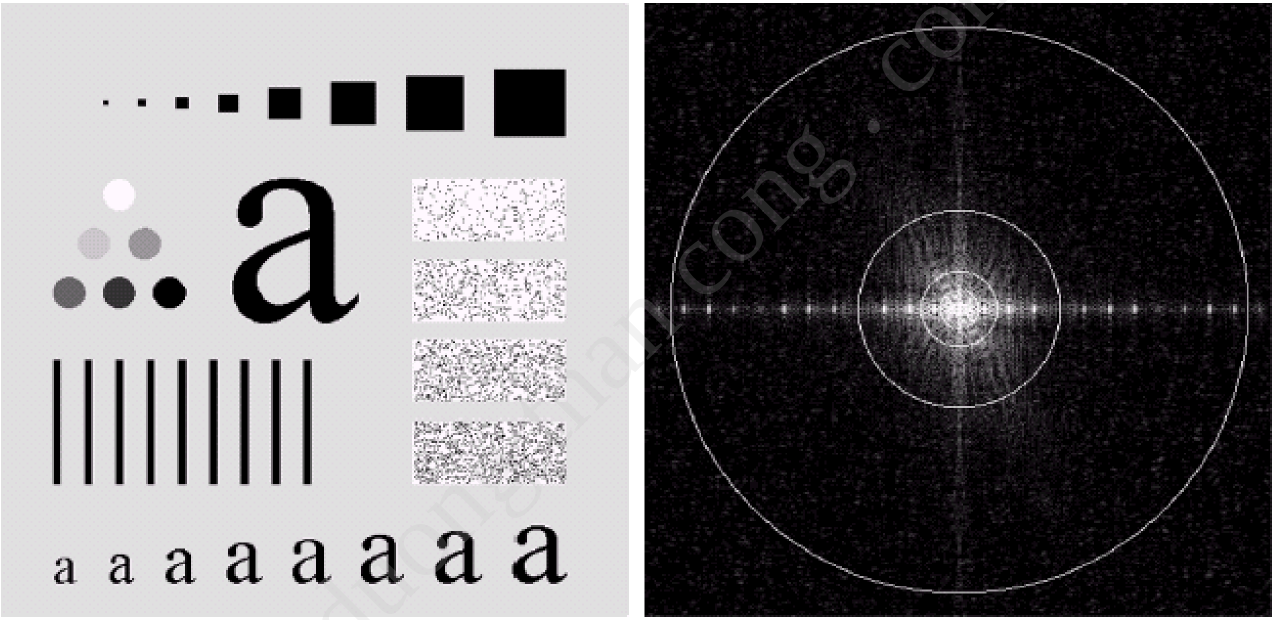


FIGURE 4.10 (a) Perspective plot of an ideal lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross section.

2. Image Enhancement: Frequency Domain (6)

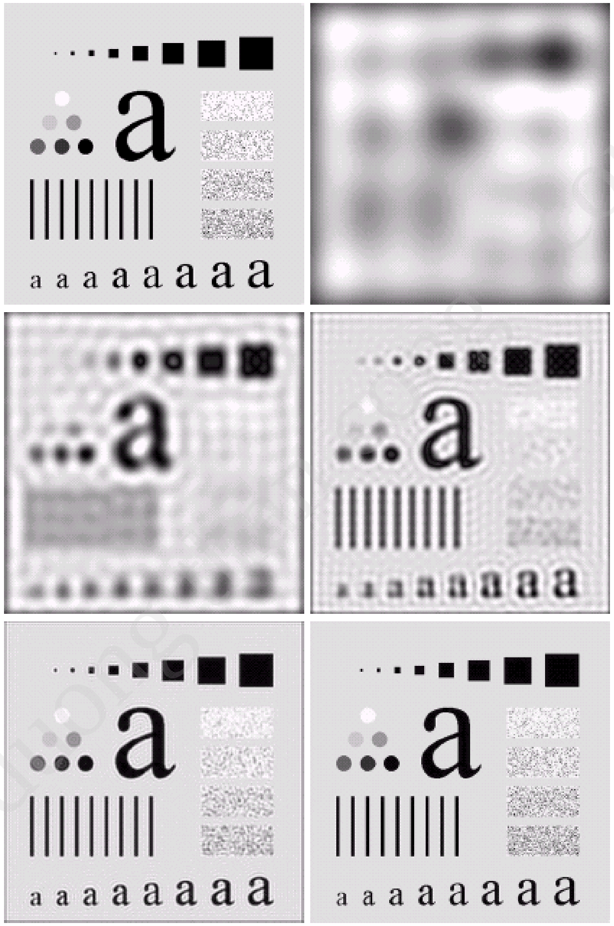
Example of Fourier spectrum:



a b

FIGURE 4.11 (a) An image of size 500 × 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.

2. Image Enhancement: Frequency Domain (7)

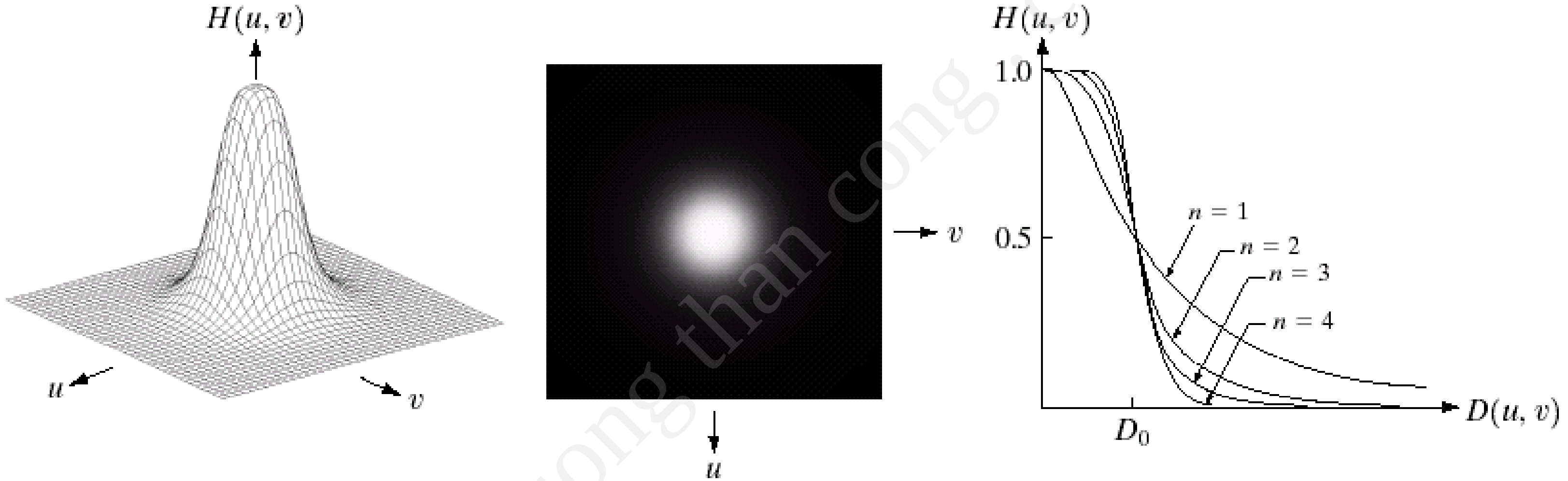


a	b
c	d
e	f

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.

2. Image Enhancement: Frequency Domain (8)

Butterworth Lowpass Filter: BLPF

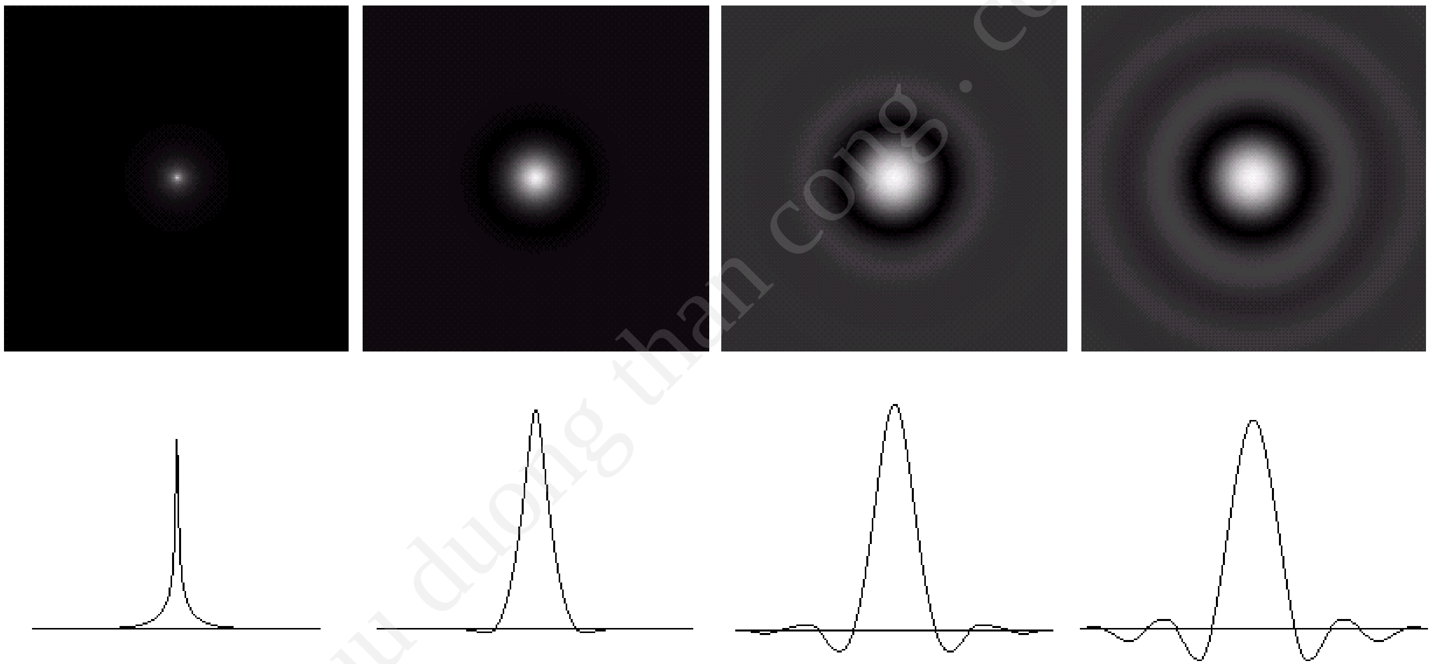


a b c

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.

2. Image Enhancement: Frequency Domain (9)

Spatial representation of BLPFs:



a b c d

FIGURE 4.16 (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.

Gaussian Lowpass Filter: GLPF

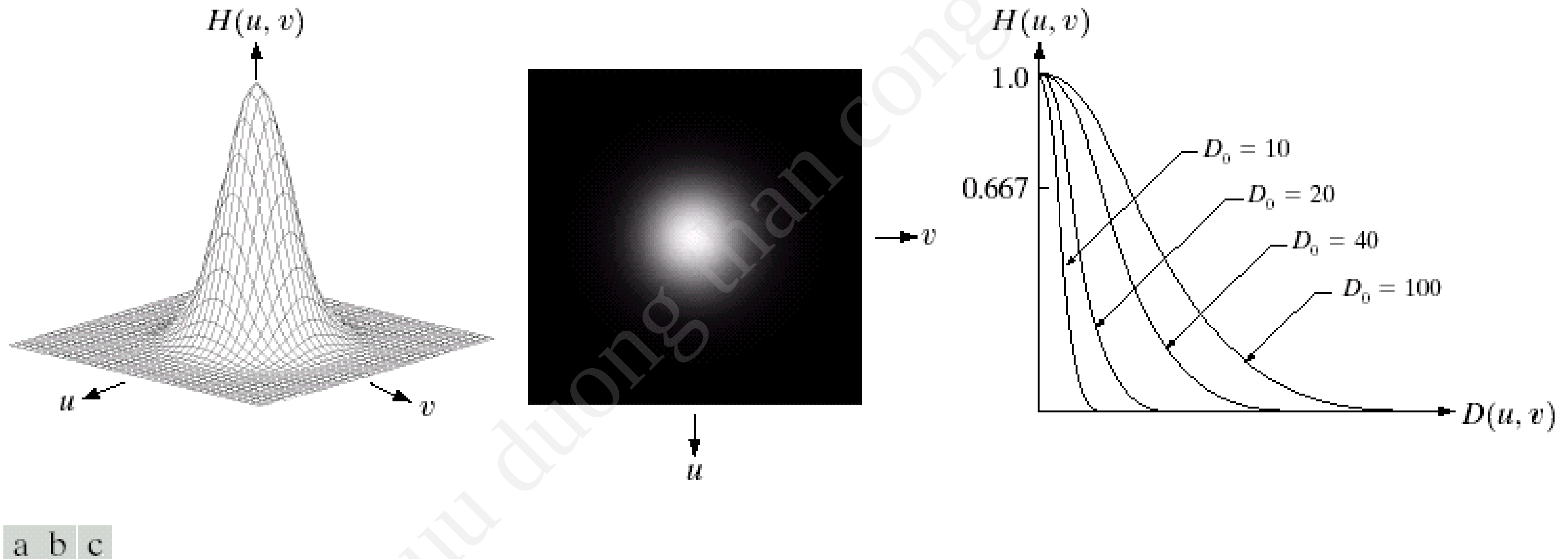


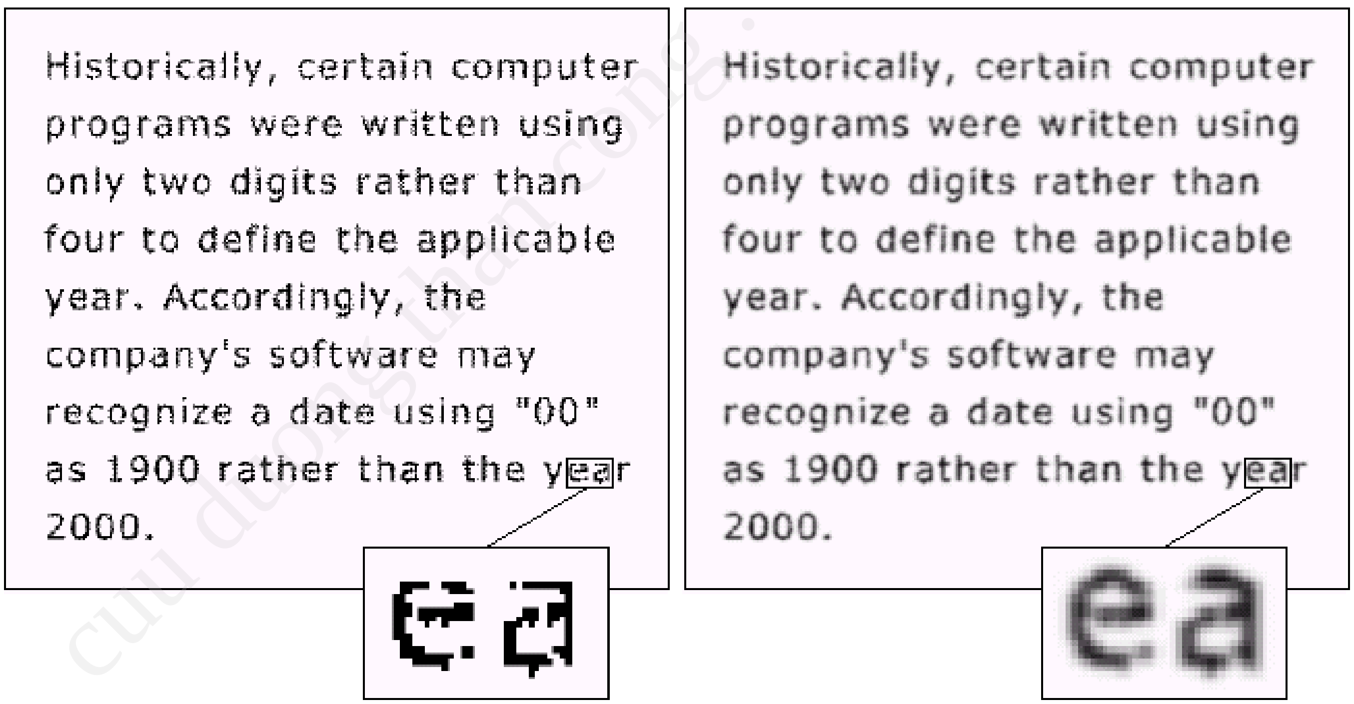
FIGURE 4.17 (a) Perspective plot of a GLPF transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections for various values of D_0.

2. Image Enhancement: Frequency Domain (11)

Example of GLPF (1):

a b

FIGURE 4.19
(a) Sample text of poor resolution (note broken characters in magnified view).
(b) Result of filtering with a GLPF (broken character segments were joined).



2. Image Enhancement: Frequency Domain (12)

Example of GLPF (2):

a b c

FIGURE 4.20 (a) Original image (1028 × 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).

Sharpening Frequency Domain Filter:

Ideal highpass filter:

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

Butterworth highpass filter:

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

Gaussian highpass filter:

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

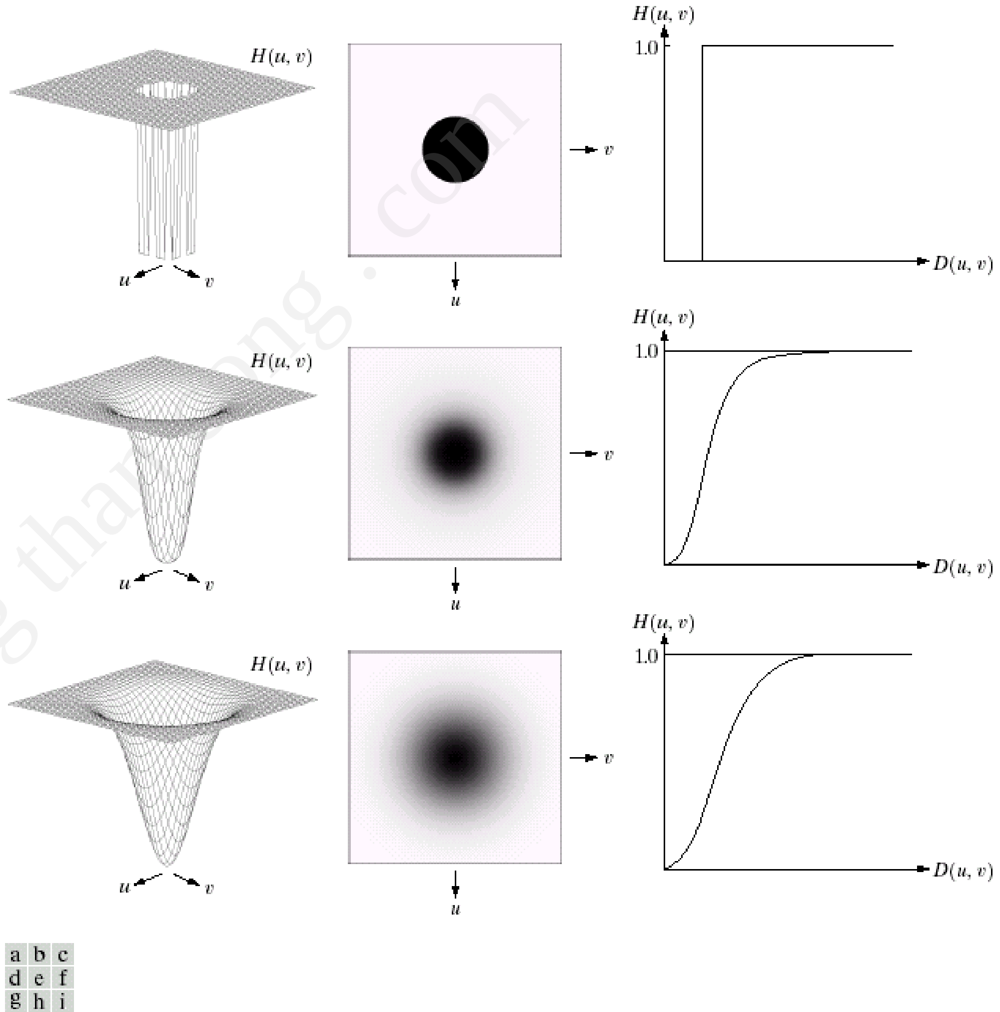
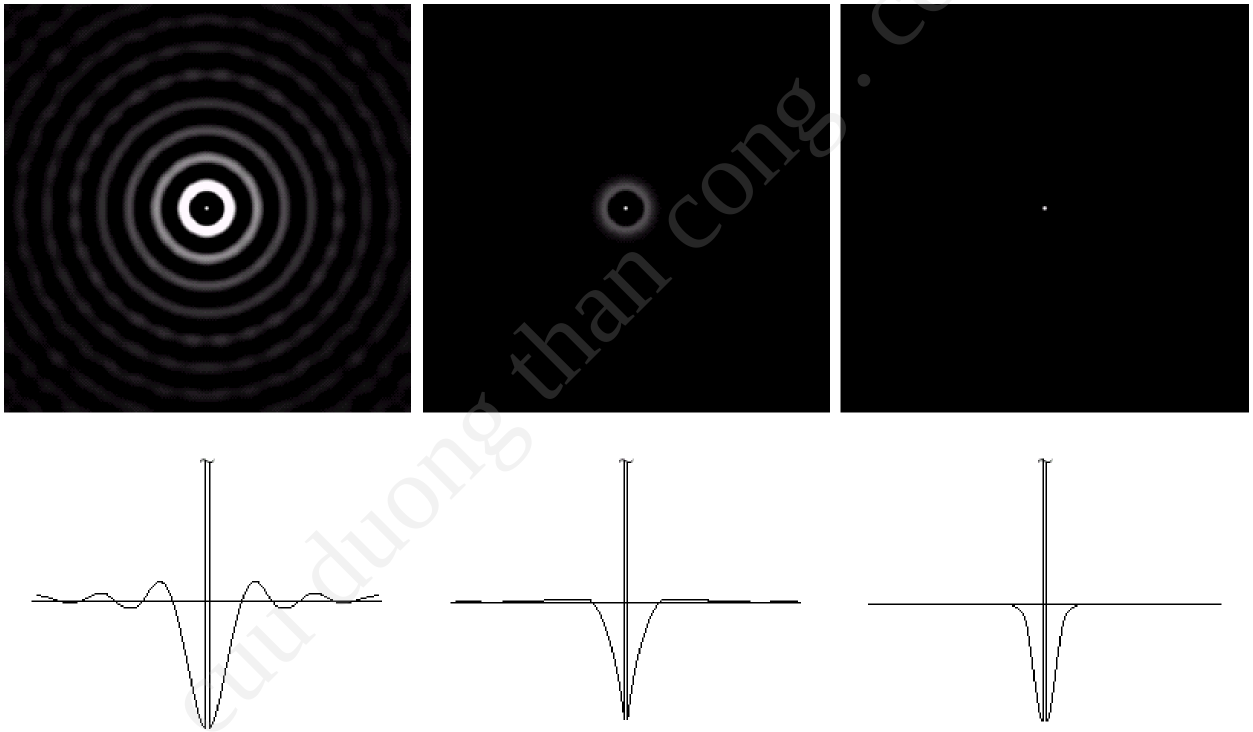


FIGURE 4.22 Top row: Perspective plot, image representation, and cross section of a typical ideal highpass filter. Middle and bottom rows: The same sequence for typical Butterworth and Gaussian highpass filters.

2. Image Enhancement: Frequency Domain (14)

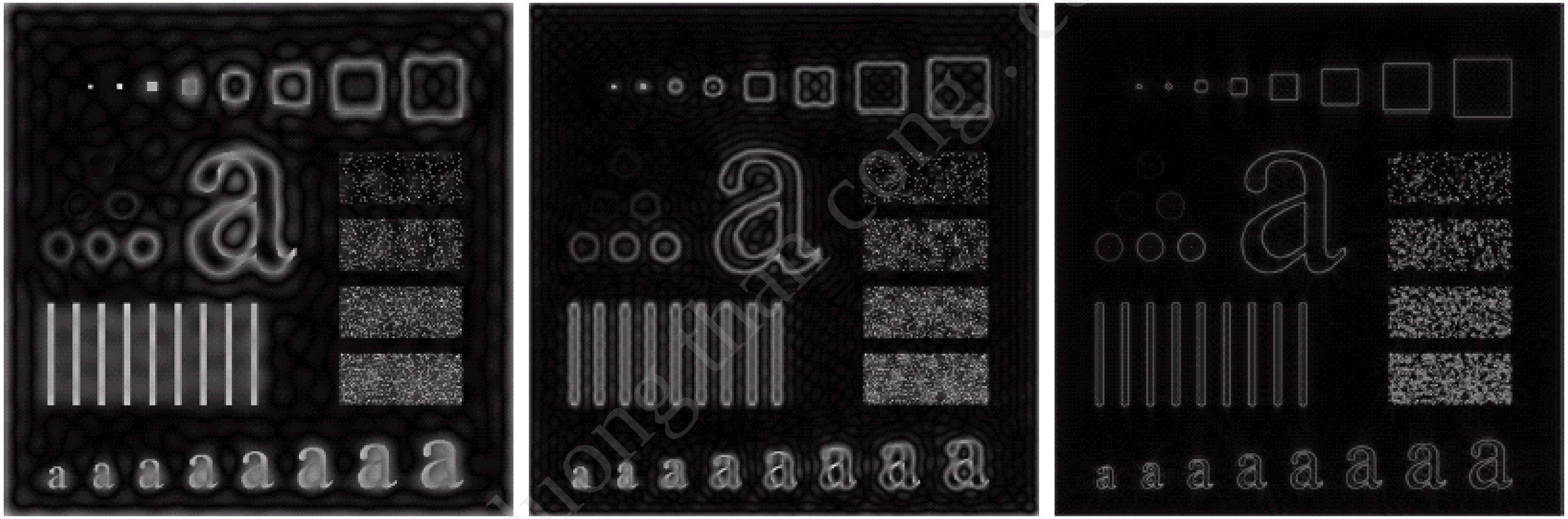
Spatial representation of Ideal, Butterworth and Gaussian highpass filters:



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.

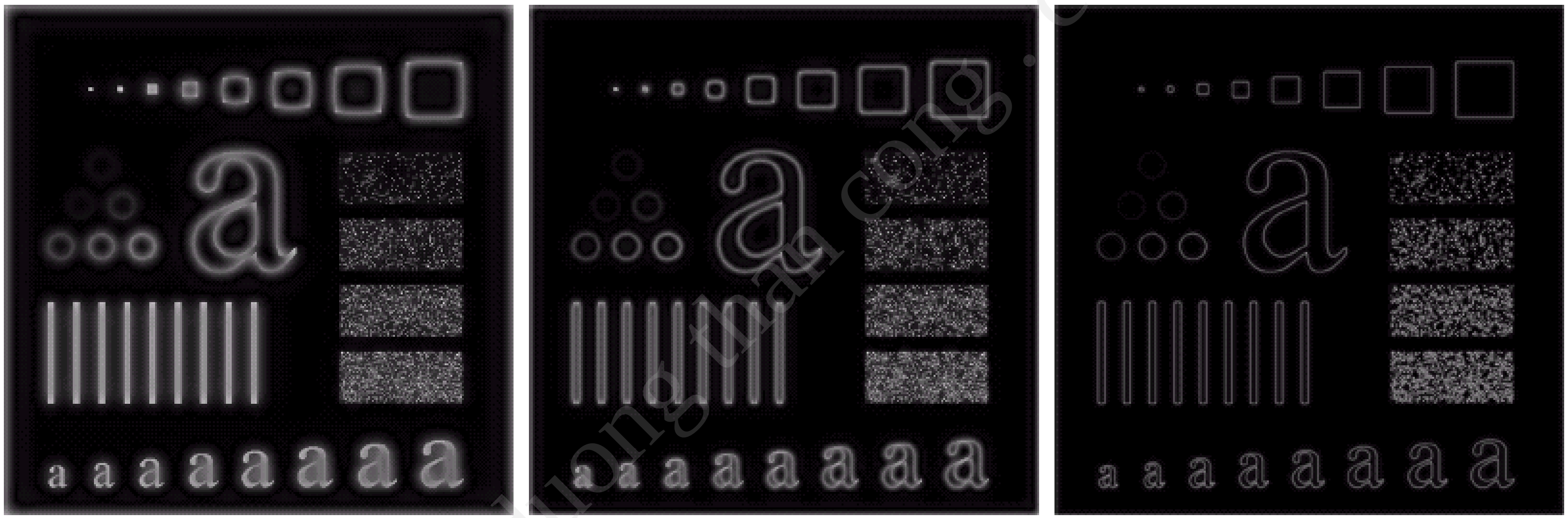
Example: Result of IHPF



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).

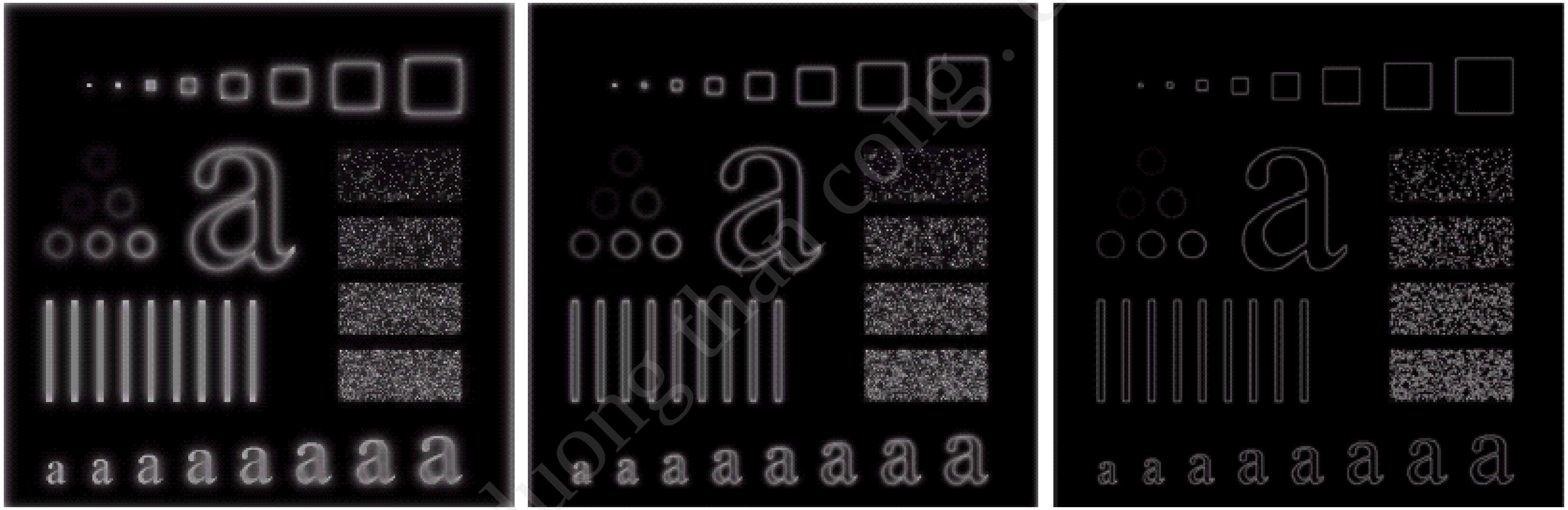
Example: Result of BHPF



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.

Example: Result of GHPF



a b c

FIGURE 4.26 Results of highpass filtering the image of Fig. 4.11(a) using a GHPF of order 2 with $D_0 = 15$, 30, and 80, respectively. Compare with Figs. 4.24 and 4.25.

Homomorphic Filter:

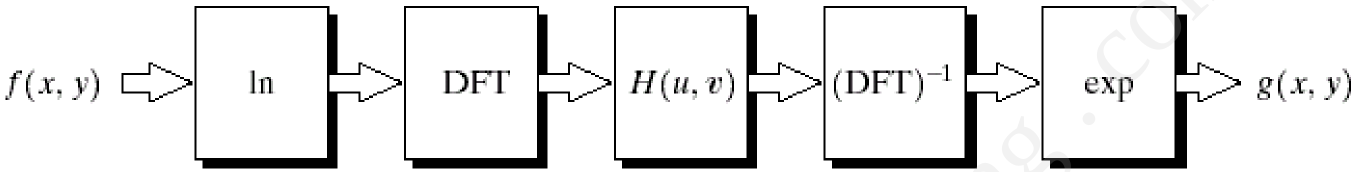


FIGURE 4.31 Homomorphic filtering approach for image enhancement.

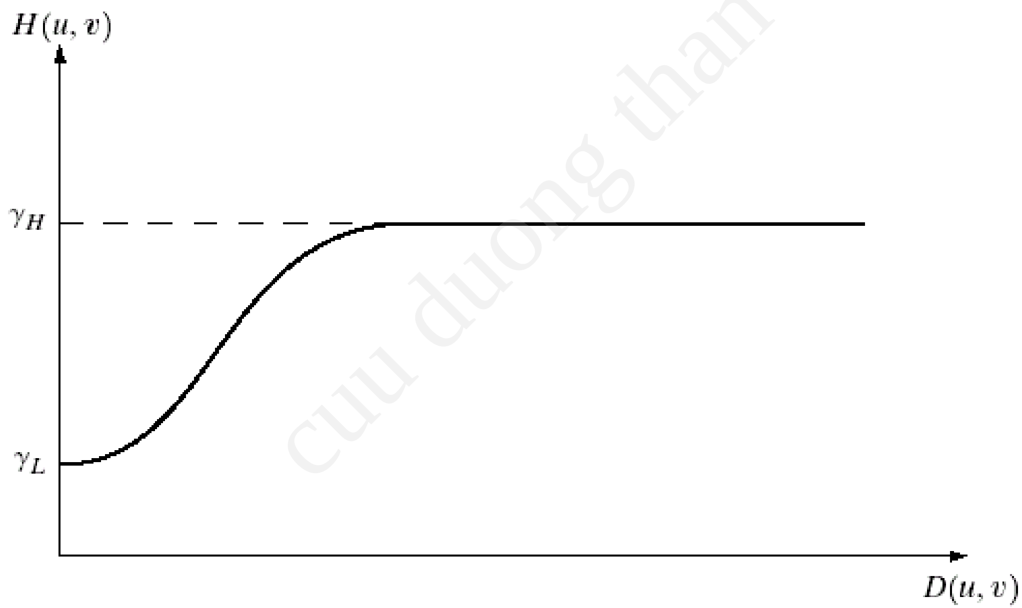


FIGURE 4.32 Cross section of a circularly symmetric filter function. $D(u, v)$ is the distance from the origin of the centered transform.

Example of Homomorphic Filter:

a b

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

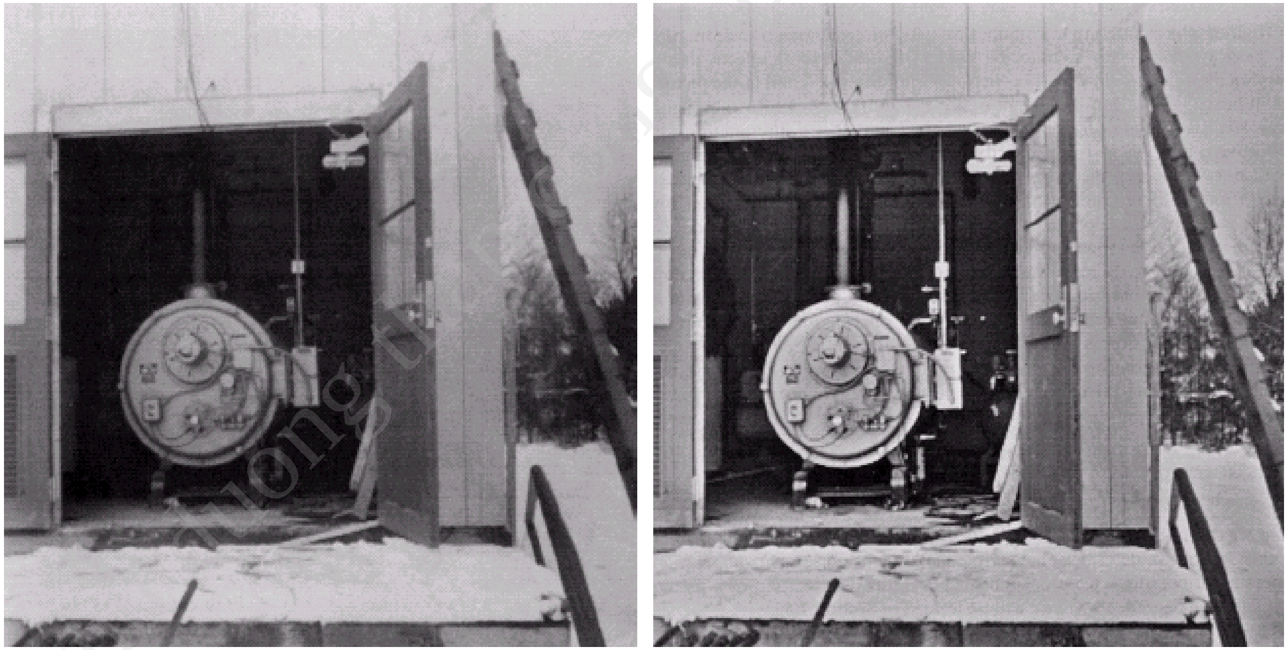


Image Restoration and Reconstruction

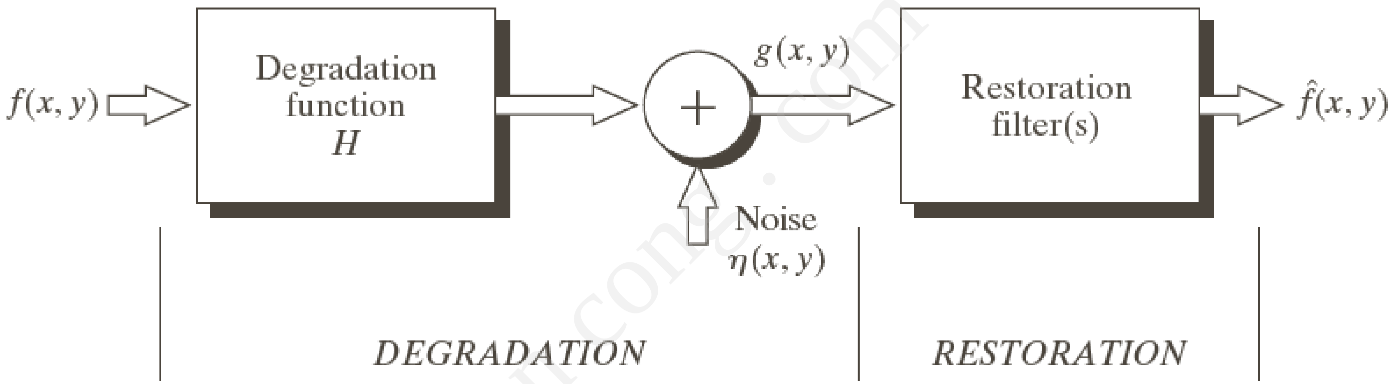
2. Image Restoration (1)

- **Image restoration**: Recover an image that has been degraded by using a prior knowledge of the degradation phenomenon.
- Model the degradation and applying the inverse process in order to recover the original image.

2. Image Restoration (2)

- **A model of image degradation/restoration process:**

FIGURE 5.1 A model of the image degradation/ restoration process.



If H is a linear, position-invariant process, then the degraded image is given in the spatial domain by

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

The model of the degraded image is given in the frequency domain by

$$G(u,v) = H(u,v)F(u,v) + N(u,v)$$

2. IR: Noise Sources

- The principal sources of noise in digital images arise during **image acquisition** and/or **transmission:**
  – Image acquisition, e.g., light levels, sensor temperature, etc.
  – Transmission, e.g., lightning or other atmospheric disturbance in wireless network.

2. IR: Noise Model (1)

- **White noise**
  - The Fourier spectrum of noise is constant.
- With the exception of spatially periodic noise, we assume:
  - Noise is independent of spatial coordinates.
  - Noise is uncorrelated with respect to the image itself.
- **Gaussian noise**

  Electronic circuit noise, sensor noise due to poor illumination and/or high temperature.
- **Erlang (gamma) noise**: Laser imaging.
- **Exponential noise**: Laser imaging.

2. IR: Noise Model (2)

- **Uniform noise**: Least descriptive; basis for numerous random number generators.
- **Impulse noise**: Quick transients, such as faulty switching.
- **Rayleigh noise**: High Dynamic Range Imaging (HDRI).

2. IR: Noise Model (3)

- **High Dynamic Range Imaging** (**HDRI**) is an imaging technique that allows for a greater dynamic range of exposure than would be obtained through any normal imaging process.
- It is now popularly used to refer to the process of **tone mapping** together with **bracketed** exposures of normal digital images, giving the end result a high, often exaggerated dynamic range.
- **Tone mapping** is a technique used in image processing and computer graphics to map a set of colors to another; often to approximate the appearance of HDRI in media with a more limited dynamic range.

2. IR: Noise Model (4)

- **Bracketing** is the general technique of taking several shots of the same subject using different or the same camera settings.
- The intention of HDRI is to accurately represent the wide range of intensity levels found in real scenes ranging from direct sunlight to shadows.
- HDRI, also called HDR (High Dynamic Range) is a feature commonly found in high-end graphics and imaging software.

http://en.wikipedia.org/wiki/High_dynamic_range_imaging
http://www.webopedia.com/TERM/H/High_Dynamic_Range_Imaging.html
http://en.wikipedia.org/wiki/High_dynamic_range_imaging
http://www.webopedia.com/TERM/H/High_Dynamic_Range_Imaging.html

2. IR: Noise Model (5)

Examples:

2. IR: Gaussian Noise

The PDF of Gaussian random variable, z, is given by

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(z-\bar{z})^2/2\sigma^2}$$

where, z represents intensity

$\bar{z}$ is the mean (average) value of z

σ is the standard deviation

- 70% of its values will be in the range:

$$[(\mu-\sigma),(\mu+\sigma)]$$

- 95% of its values will be in the range:

$$[(\mu-2\sigma),(\mu+2\sigma)]$$

2. IR: Rayleigh Noise

The PDF of Rayleigh noise is given by

$$p(z) = \begin{cases} \frac{2}{b}(z-a)e^{-(z-a)^2/b} & \text{for } z \geq a \\ 0 & \text{for } z < a \end{cases}$$

The mean and variance of this density are given by

$$\bar{z} = a + \sqrt{\pi b / 4}$$

$$\sigma^2 = \frac{b(4-\pi)}{4}$$

2. IR: Erlang (Gamma) Noise

The PDF of Erlang noise is given by

$$p(z) = \begin{cases} \dfrac{a^b z^{b-1}}{(b-1)!} e^{-az} & \text{for } z \geq 0 \\ 0 & \text{for } z < a \end{cases}$$

The mean and variance of this density are given by

$$\bar{z} = b / a$$

$$\sigma^2 = b / a^2$$

2. IR: Exponential Noise

The PDF of exponential noise is given by

$$p(z) = \begin{cases} ae^{-az} & \text{for } z \geq 0 \\ 0 & \text{for } z < a \end{cases}$$

The mean and variance of this density are given by

$$\bar{z} = 1/a$$

$$\sigma^2 = 1/a^2$$

2. IR: Uniform Noise

The PDF of uniform noise is given by

$$p(z) = \begin{cases} \dfrac{1}{b-a} & \text{for a} \le z \le b \\ 0 & \text{otherwise} \end{cases}$$

The mean and variance of this density are given by

$$\bar{z} = (a+b)/2$$

$$\sigma^2 = (b-a)^2/12$$

2. IR: Impulse (Salt-and-Pepper) Noise

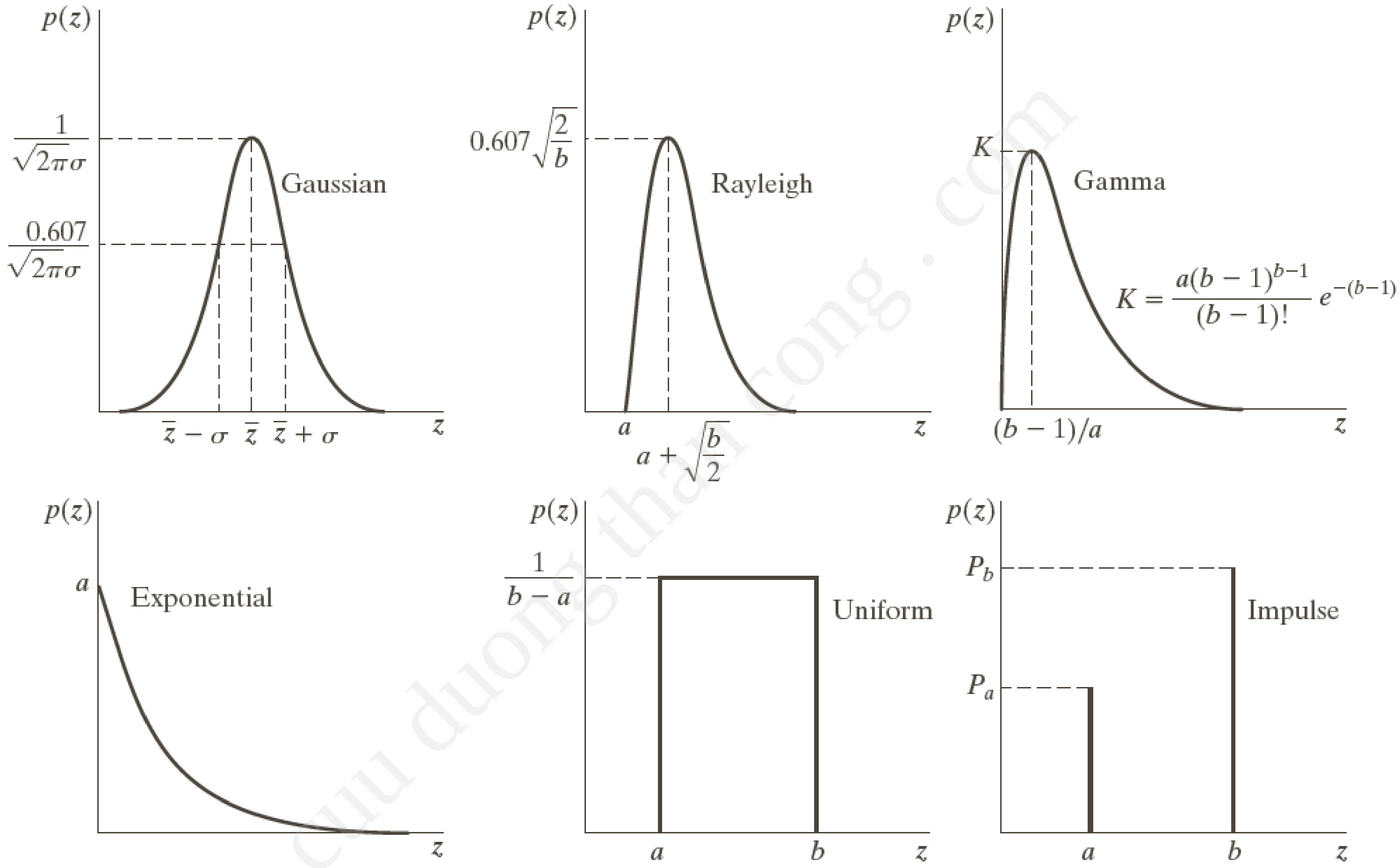
The PDF of (bipolar) impulse noise is given by

$$p(z) = \begin{cases} P_a & \text{for } z = a \\ P_b & \text{for } z = b \\ 0 & \text{otherwise} \end{cases}$$

if $b > a$, gray-level b will appear as a light dot, while level a will appear like a dark dot.

If either P_a or P_b is zero, the impulse noise is called *unipolar*

2. IR: Noise Distributions (1)

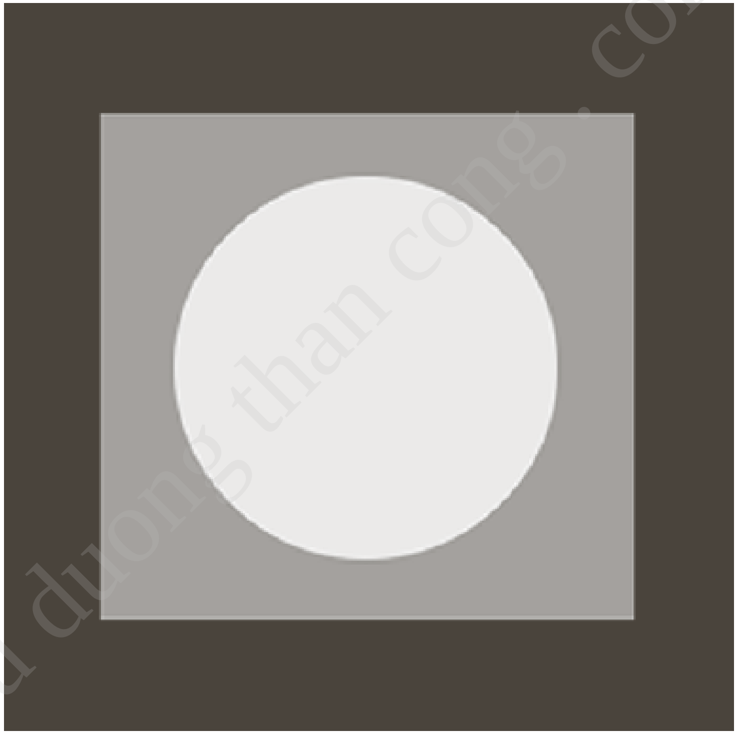


a	b	c
d	e	f

FIGURE 5.2 Some important probability density functions.

2. IR: Noise Distributions (2)

Examples of noise:



Original Image

2. IR: Noise Distributions (3)

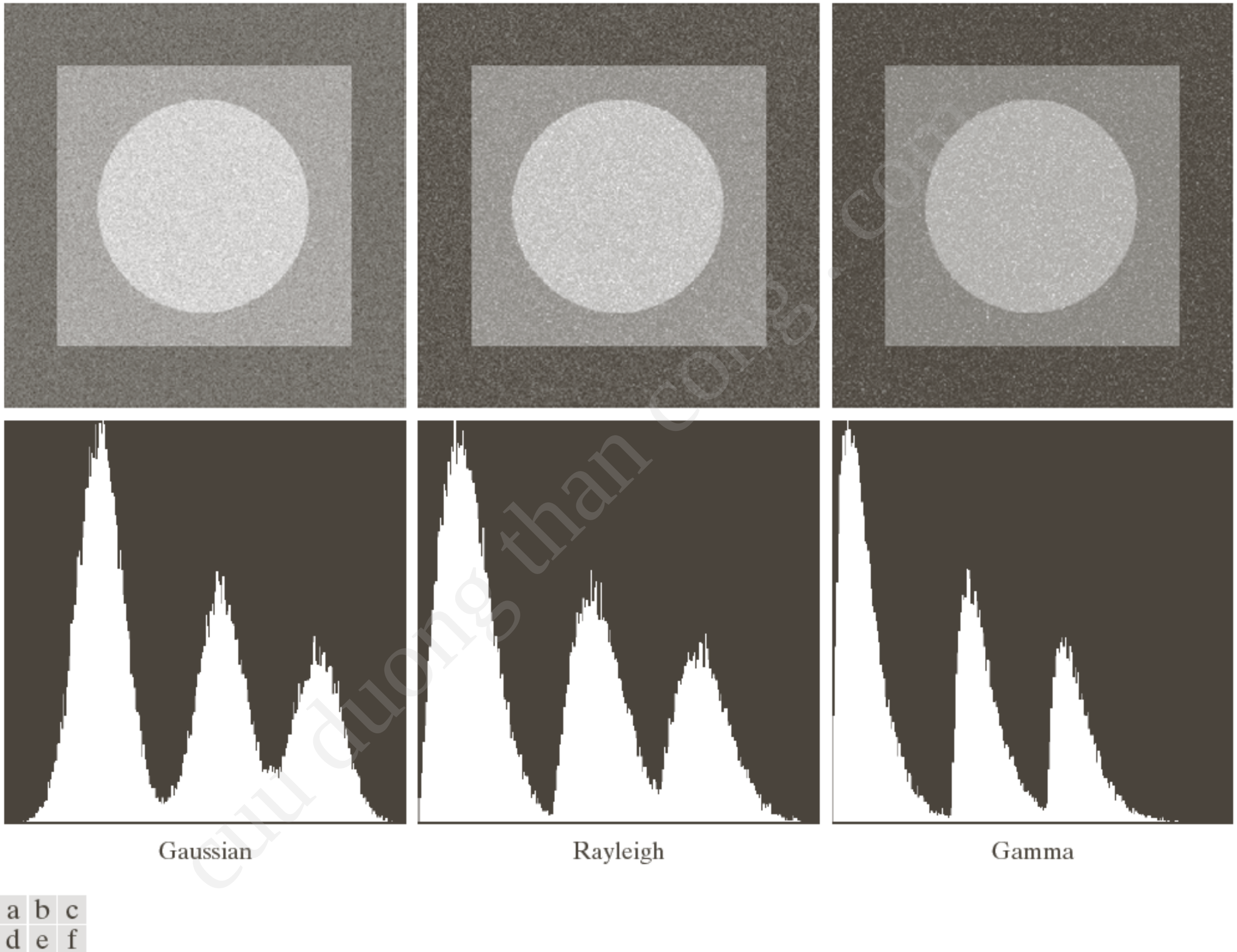


FIGURE 5.4 Images and histograms resulting from adding Gaussian, Rayleigh, and gamma noise to the image in Fig. 5.3.

2. IR: Noise Distributions (4)

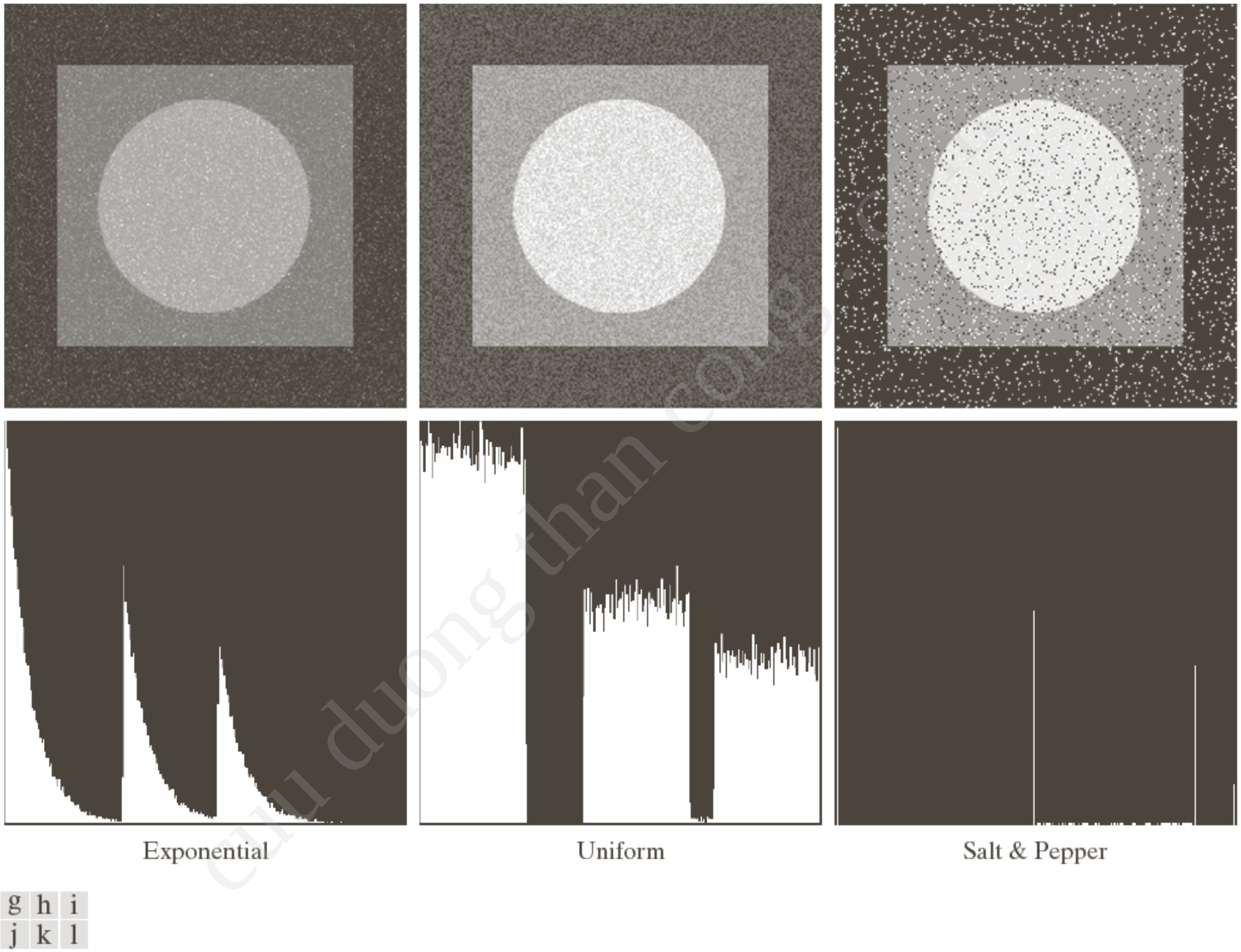


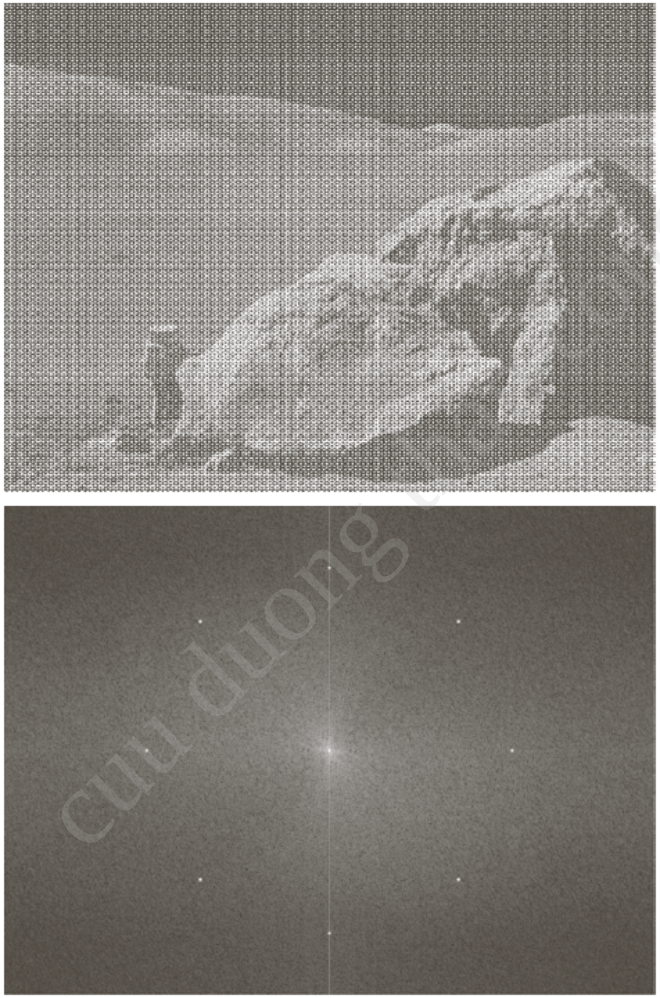
FIGURE 5.4 *(Continued)* Images and histograms resulting from adding exponential, uniform, and salt and pepper noise to the image in Fig. 5.3.

2. IR: Periodic Noise (1)

- Periodic noise in an image arises typically from electrical or electromechanical interference during image acquisition.
- It is a type of spatially dependent noise.
- Periodic noise can be **reduced significantly via frequency domain filtering**.

2. IR: Periodic Noise (2)

Example:

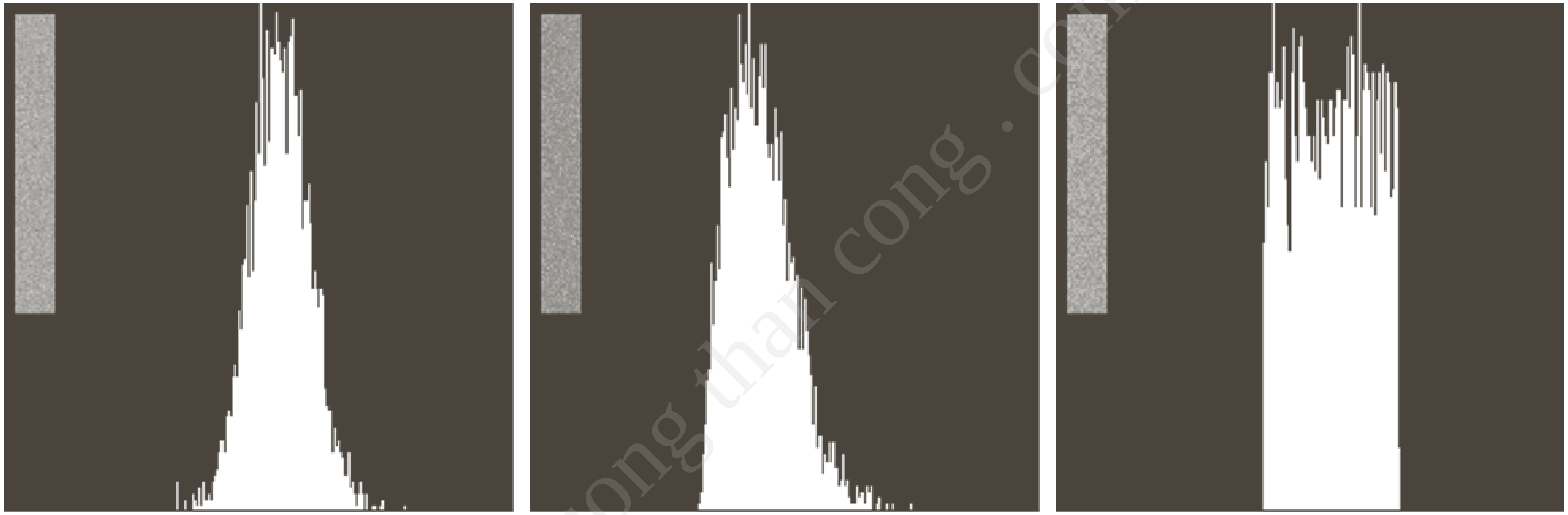


a
b

FIGURE 5.5 (a) Image corrupted by sinusoidal noise. (b) Spectrum (each pair of conjugate impulses corresponds to one sine wave). (Original image courtesy of NASA.)

2. IR: Estimation of Noise Parameters (1)

The shape of the histogram identifies the closest PDF match.



a b c

FIGURE 5.6 Histograms computed using small strips (shown as inserts) from (a) the Gaussian, (b) the Rayleigh, and (c) the uniform noisy images in Fig. 5.4.

2. IR: Estimation of Noise Parameters (2)

Consider a subimage denoted by S, and let $p_s(z_i)$, $i = 0, 1, \ldots, L$ -1, denote the probability estimates of the intensities of the pixels in S. The mean and variance of the pixels in S:

$$\bar{z} = \sum_{i=0}^{L-1} z_i p_s(z_i)$$

and

$$\sigma^2 = \sum_{i=0}^{L-1} (z_i - \bar{z})^2 p_s(z_i)$$

2. IR: Restoration in the Presence of Noise Only (1)

- **Spatial filtering:**

Noise model without degradation

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

and

$$G(u,v) = F(u,v) + N(u,v)$$

- **Mean filters:**

Let S_{xy} represent the set of coordinates in a rectangle subimage window of size $m \times n$, centered at (x, y).

Arithmetic mean filter

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

2. IR: Restoration in the Presence of Noise Only (2)

Geometric mean filter

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

Generally, a geometric mean filter achieves smoothing comparable to the arithmetic mean filter, but it tends to lose less image detail in the process.

2. IR: Restoration in the Presence of Noise Only (3)

Harmonic mean filter

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

It works well for salt noise, but fails for pepper noise. It does well also with other types of noise like Gaussian noise.

2. IR: Restoration in the Presence of Noise Only (4)

Contraharmonic mean filter

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

Q is the order of the filter.

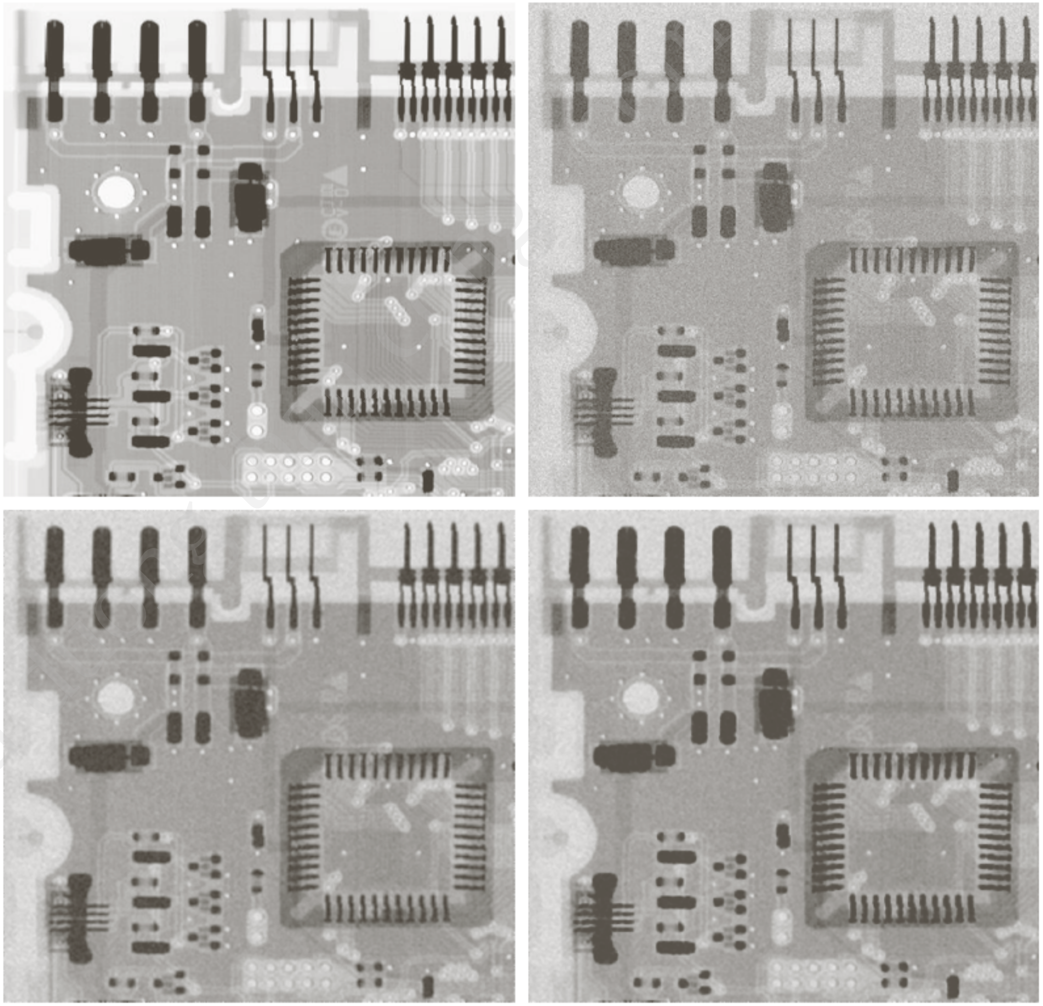
It is well suited for reducing the effects of salt-and-pepper noise. $Q > 0$ for pepper noise and $Q < 0$ for salt noise.

2. IR: Restoration in the Presence of Noise Only (5)

Examples:

a b
c d

FIGURE 5.7
(a) X-ray image. (b) Image corrupted by additive Gaussian noise. (c) Result of filtering with an arithmetic mean filter of size 3 × 3. (d) Result of filtering with a geometric mean filter of the same size. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)



2. IR: Restoration in the Presence of Noise Only (6)

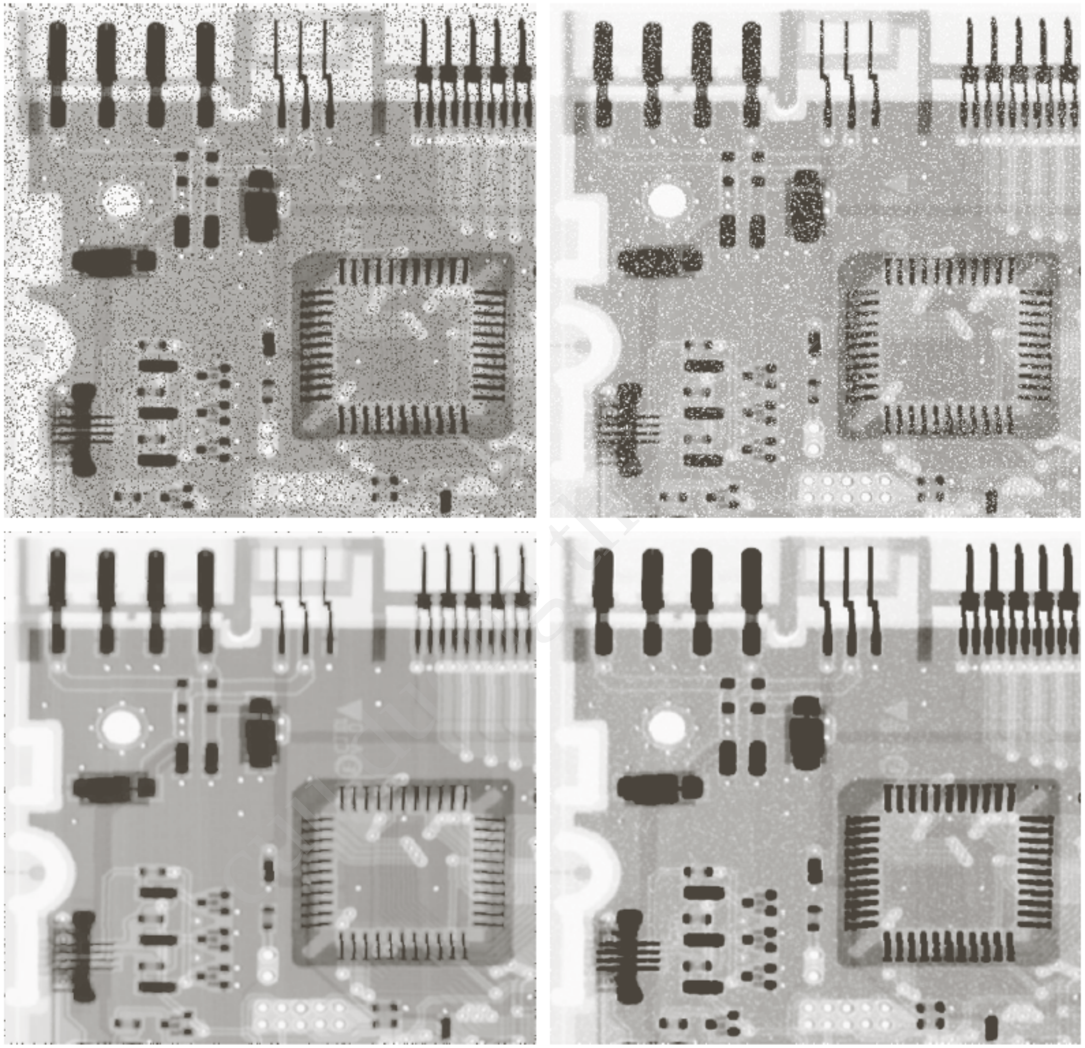
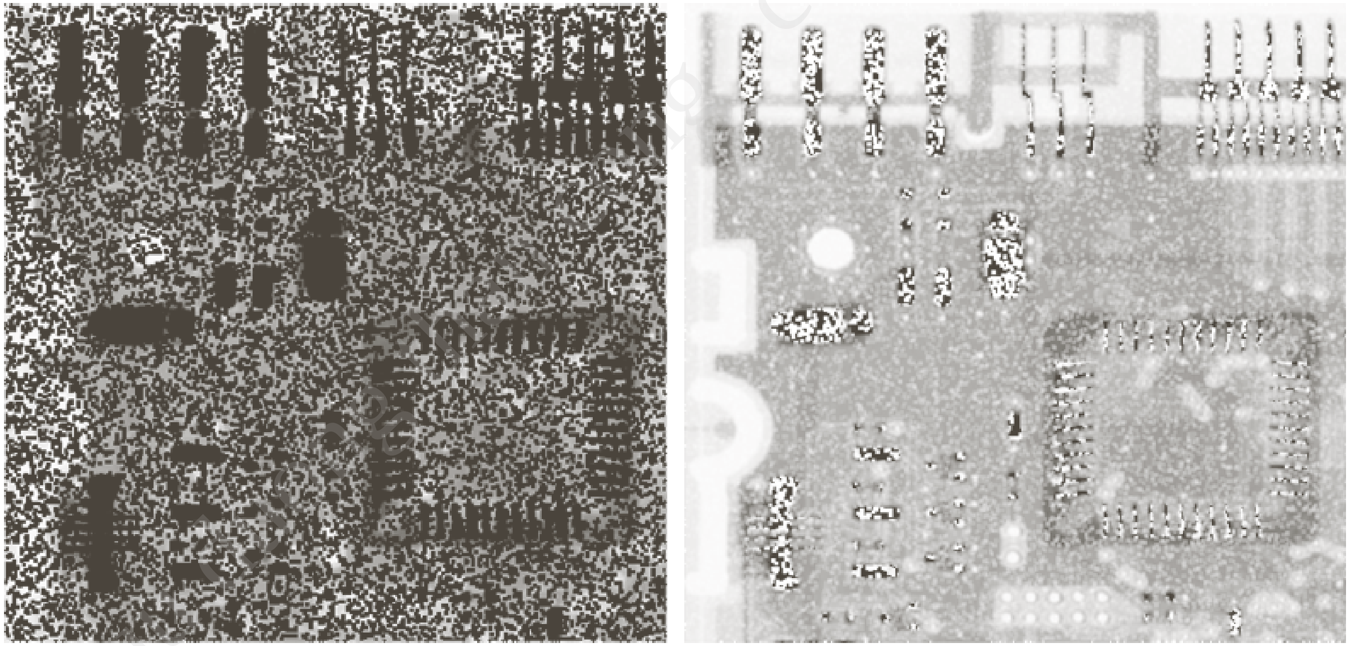


FIGURE 5.8
(a) Image corrupted by pepper noise with a probability of 0.1. (b) Image corrupted by salt noise with the same probability. (c) Result of filtering (a) with a 3×3 contra-harmonic filter of order 1.5. (d) Result of filtering (b) with $Q = -1.5$.

2. IR: Restoration in the Presence of Noise Only (7)



a b

FIGURE 5.9 Results of selecting the wrong sign in contraharmonic filtering. (a) Result of filtering Fig. 5.8(a) with a contraharmonic filter of size 3×3 and $Q = -1.5$. (b) Result of filtering 5.8(b) with $Q = 1.5$.

2. IR: Restoration in the Presence of Noise Only (8)

- **Order-statistic filters:**

Median filter

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

Max filter

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

Min filter

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

Midpoint filter

$$\hat{f}(x,y) = \frac{1}{2}\left[\max_{(s,t)\in S_{xy}}\{g(s,t)\} + \min_{(s,t)\in S_{xy}}\{g(s,t)\}\right]$$

2. IR: Restoration in the Presence of Noise Only (9)

Alpha-trimmed mean filter

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

We delete the $d/2$ lowest and the $d/2$ highest intensity values of $g(s,t)$ in the neighborhood S_{xy}. Let $g_r(s,t)$ represent the remaining $mn-d$ pixels.

2. IR: Restoration in the Presence of Noise Only (10)

a	b
c	d

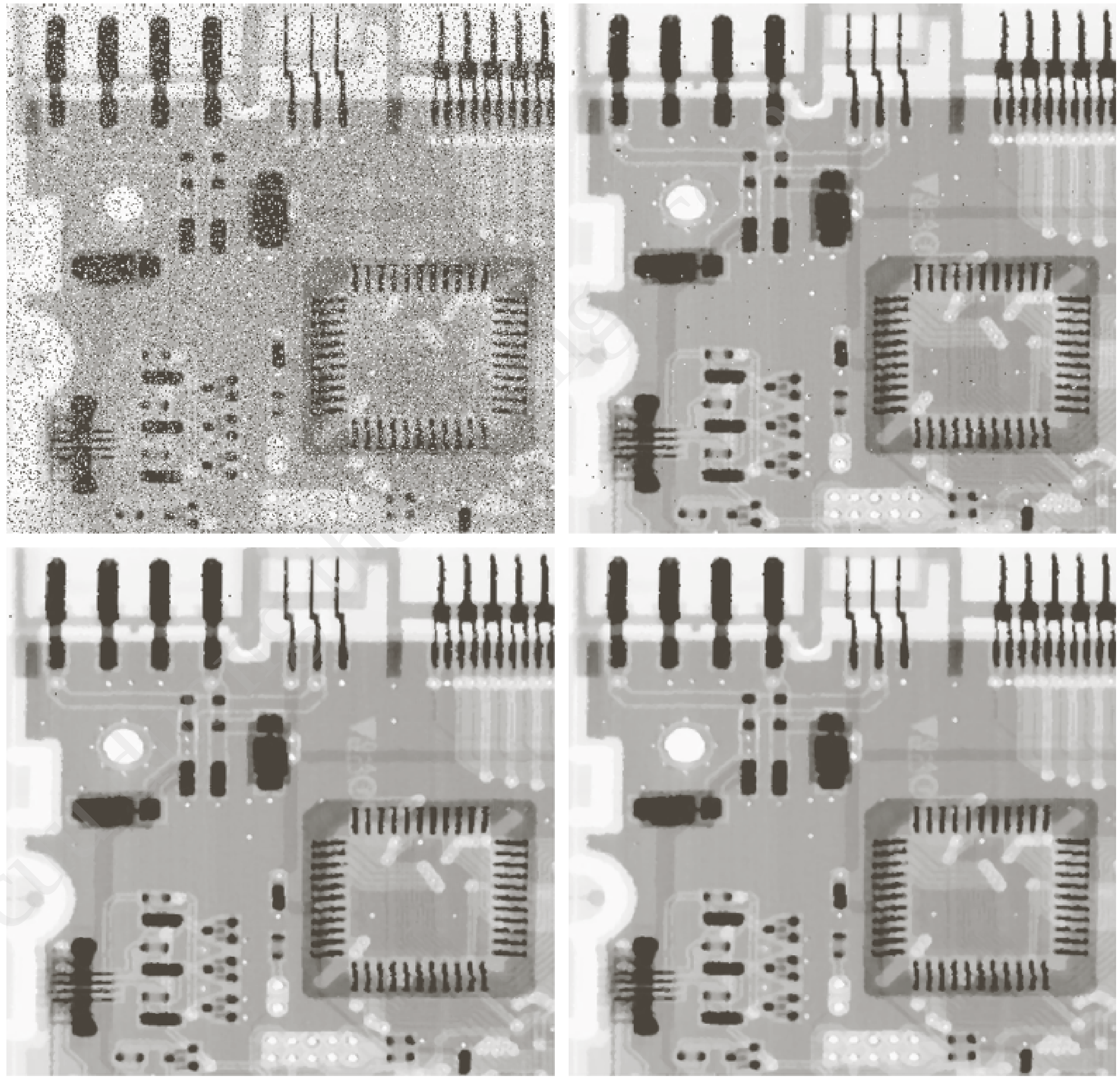
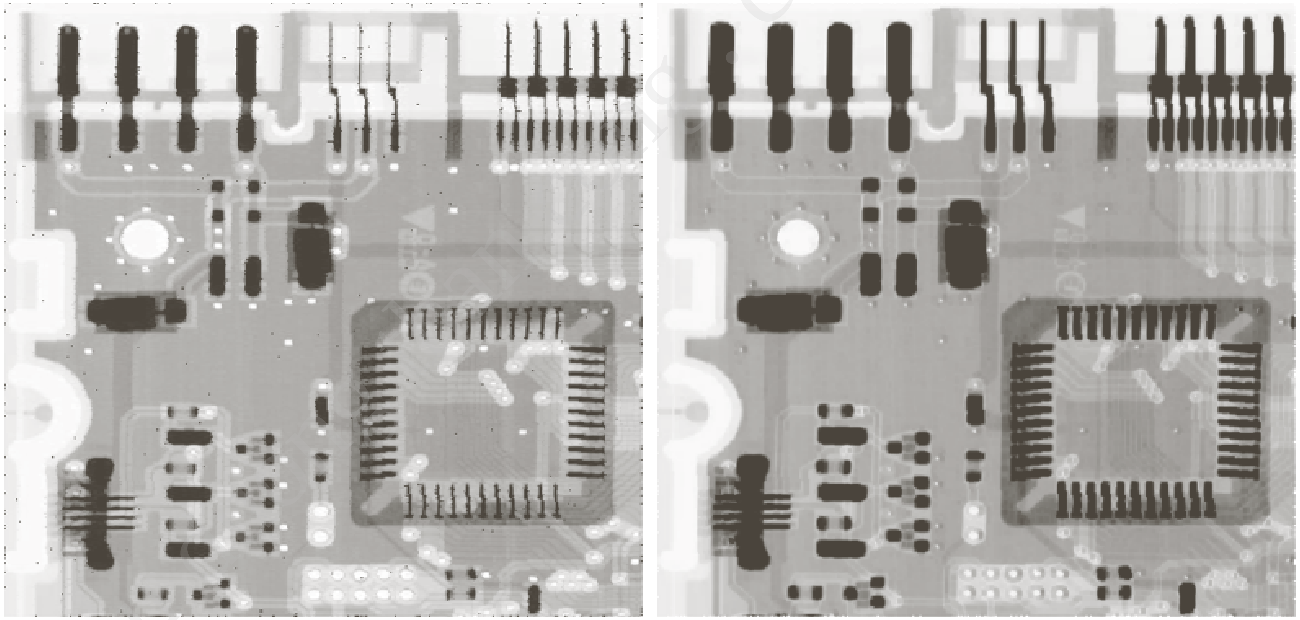


FIGURE 5.10 (a) Image corrupted by salt-and-pepper noise with probabilities $P_a = P_b = 0.1$. (b) Result of one pass with a median filter of size 3×3. (c) Result of processing (b) with this filter. (d) Result of processing (c) with the same filter.

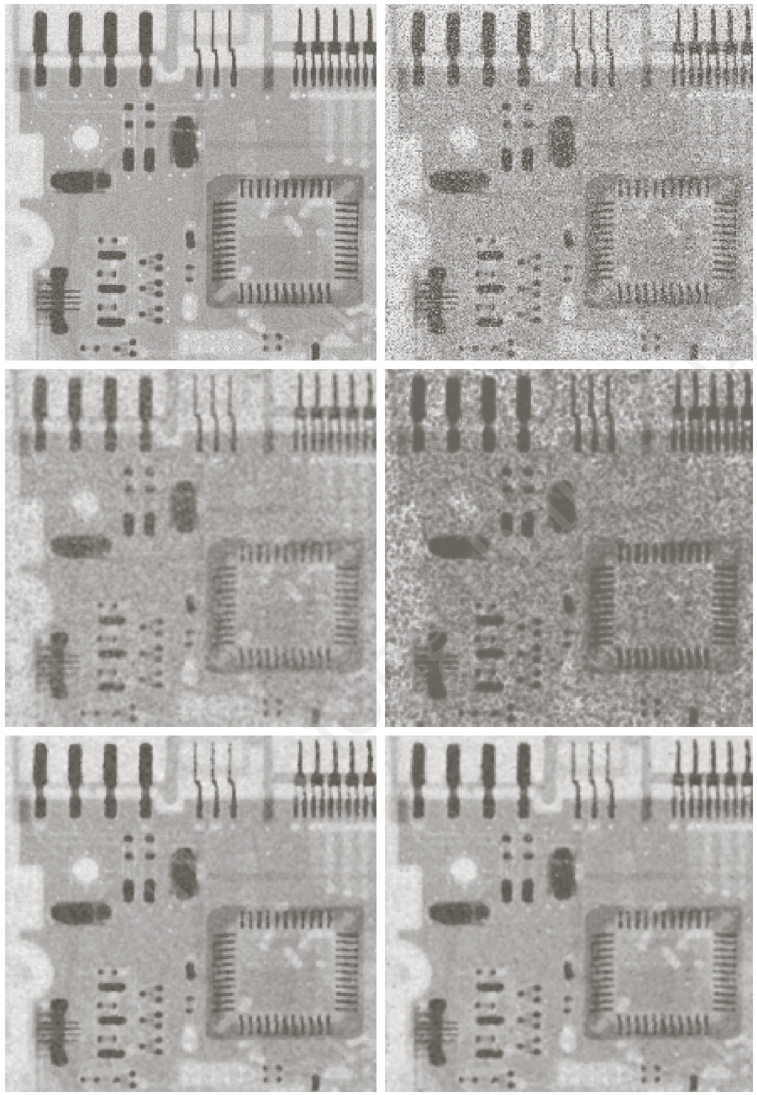
2. IR: Restoration in the Presence of Noise Only (11)



a b

FIGURE 5.11 (a) Result of filtering Fig. 5.8(a) with a max filter of size 3 × 3. (b) Result of filtering 5.8(b) with a min filter of the same size.

2. IR: Restoration in the Presence of Noise Only (12)



a b
c d
e f

FIGURE 5.12 (a) Image corrupted by additive uniform noise. (b) Image additionally corrupted by additive salt-and-pepper noise. Image (b) filtered with a 5 × 5; (c) arithmetic mean filter; (d) geometric mean filter; (e) median filter; and (f) alpha-trimmed mean filter with $d = 5$.

2. IR: Restoration in the Presence of Noise Only (13)

- **Adaptive filters, local noise reduction filters**:

  The behavior changes based on statistical characteristics of the image inside the filter region defined by the $m \times n$ rectangular window.

  The performance is superior to that of the filters discussed.

2. IR: Restoration in the Presence of Noise Only (14)

S_{xy}: local region

The response of the filter at the center point (x,y) of S_{xy} is based on four quantities:

(a) $g(x, y)$, the value of the noisy image at (x, y);

(b) σ_η^2, the variance of the noise corrupting $f(x, y)$ to form $g(x, y)$;

(c) m_L, the local mean of the pixels in S_{xy};

(d) σ_L^2, the local variance of the pixels in S_{xy}.

2. IR: Restoration in the Presence of Noise Only (15)

The behavior of the filter:

(a) if σ_η^2 is zero, the filter should return simply the value of $g(x, y)$.

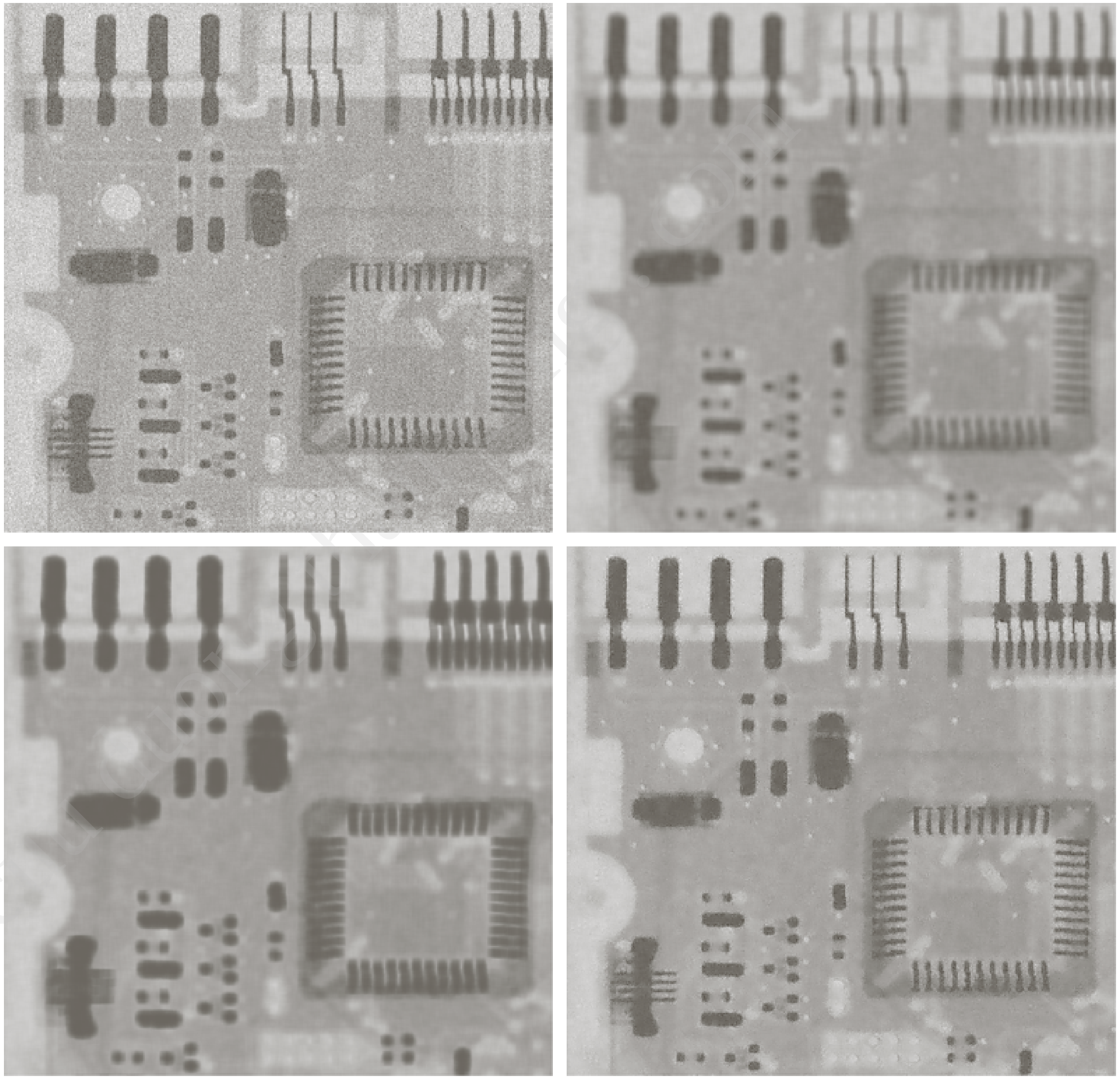
(b) if the local variance is high relative to σ_η^2, the filter should return a value close to $g(x, y)$;

(c) if the two variances are equal, the filter returns the arithmetic mean value of the pixels in S_{xy}.

An adaptive expression for obtaining $\hat{f}(x, y)$ based on the assumptions:

$$\hat{f}(x, y) = g(x, y) - \frac{\sigma_\eta^2}{\sigma_L^2}\left[g(x, y) - m_L\right]$$

2. IR: Restoration in the Presence of Noise Only (16)



a b
c d

FIGURE 5.13
(a) Image corrupted by additive Gaussian noise of zero mean and variance 1000. (b) Result of arithmetic mean filtering. (c) Result of geometric mean filtering. (d) Result of adaptive noise reduction filtering. All filters were of size 7×7.

2. IR: Restoration in the Presence of Noise Only (17)

Adaptive median filters:

The notation:

$z_{\min}$ = minimum intensity value in S_{xy}

$z_{\max}$ = maximum intensity value in S_{xy}

z_{med} = median intensity value in S_{xy}

z_{xy} = intensity value at coordinates (x, y)

$S_{\max}$ = maximum allowed size of S_{xy}

2. IR: Restoration in the Presence of Noise Only (18)

The adaptive median-filtering works in two stages:

Stage A:

$A1 = z_{med} - z_{min}$; $A2 = z_{med} - z_{max}$

if A1>0 and A2<0, go to stage B

Else increase the window size

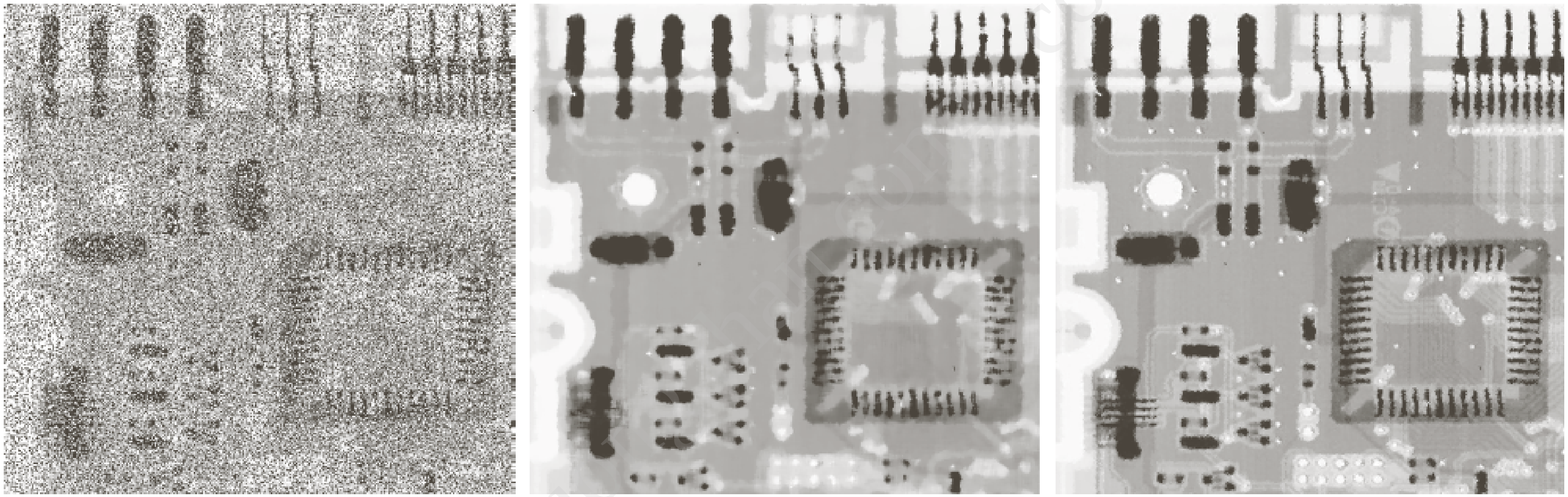
if window size $\leq S_{max}$, repeat stage A; Else output z_{med}

Stage B:

$B1 = z_{xy} - z_{min}$; $B2 = z_{xy} - z_{max}$

if B1>0 and B2<0, output z_{xy}; Else output z_{med}

2. IR: Restoration in the Presence of Noise Only (19)



a b c

FIGURE 5.14 (a) Image corrupted by salt-and-pepper noise with probabilities $P_a = P_b = 0.25$. (b) Result of filtering with a 7×7 median filter. (c) Result of adaptive median filtering with $S_{max} = 7$.

2. IR: Restoration in the Presence of Noise Only (20)

- **<u>Periodic noise reduction by frequency domain filtering:</u>**

  The basic idea: Periodic noise appears as concentrated bursts of energy in the Fourier transform, at locations corresponding to the frequencies of the periodic interference.

  Approach: A selective filter is used to isolate the noise.

2. IR: Restoration in the Presence of Noise Only (21)

Bandreject filter:

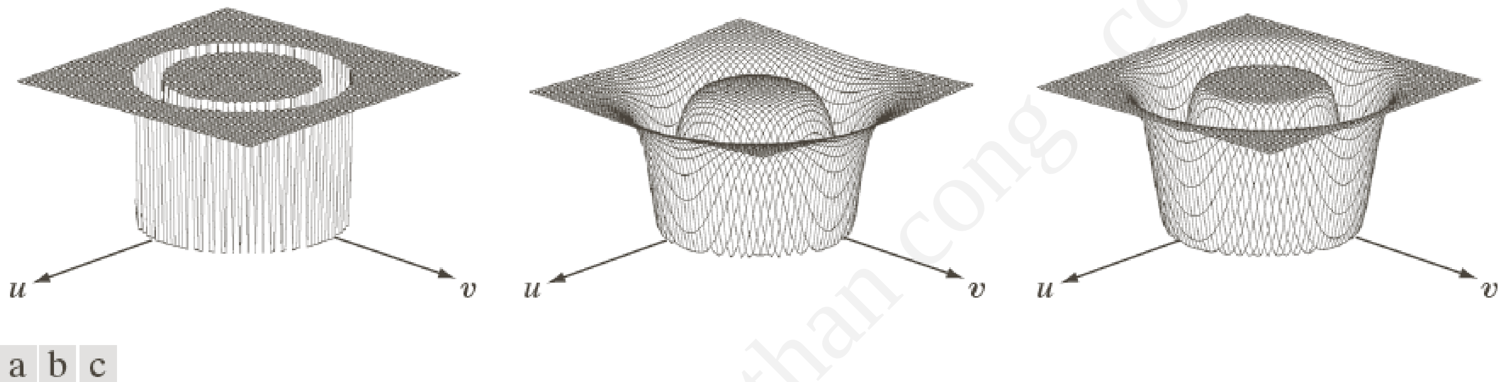


FIGURE 5.15 From left to right, perspective plots of ideal, Butterworth (of order 1), and Gaussian bandreject filters.

2. IR: Restoration in the Presence of Noise Only (22)

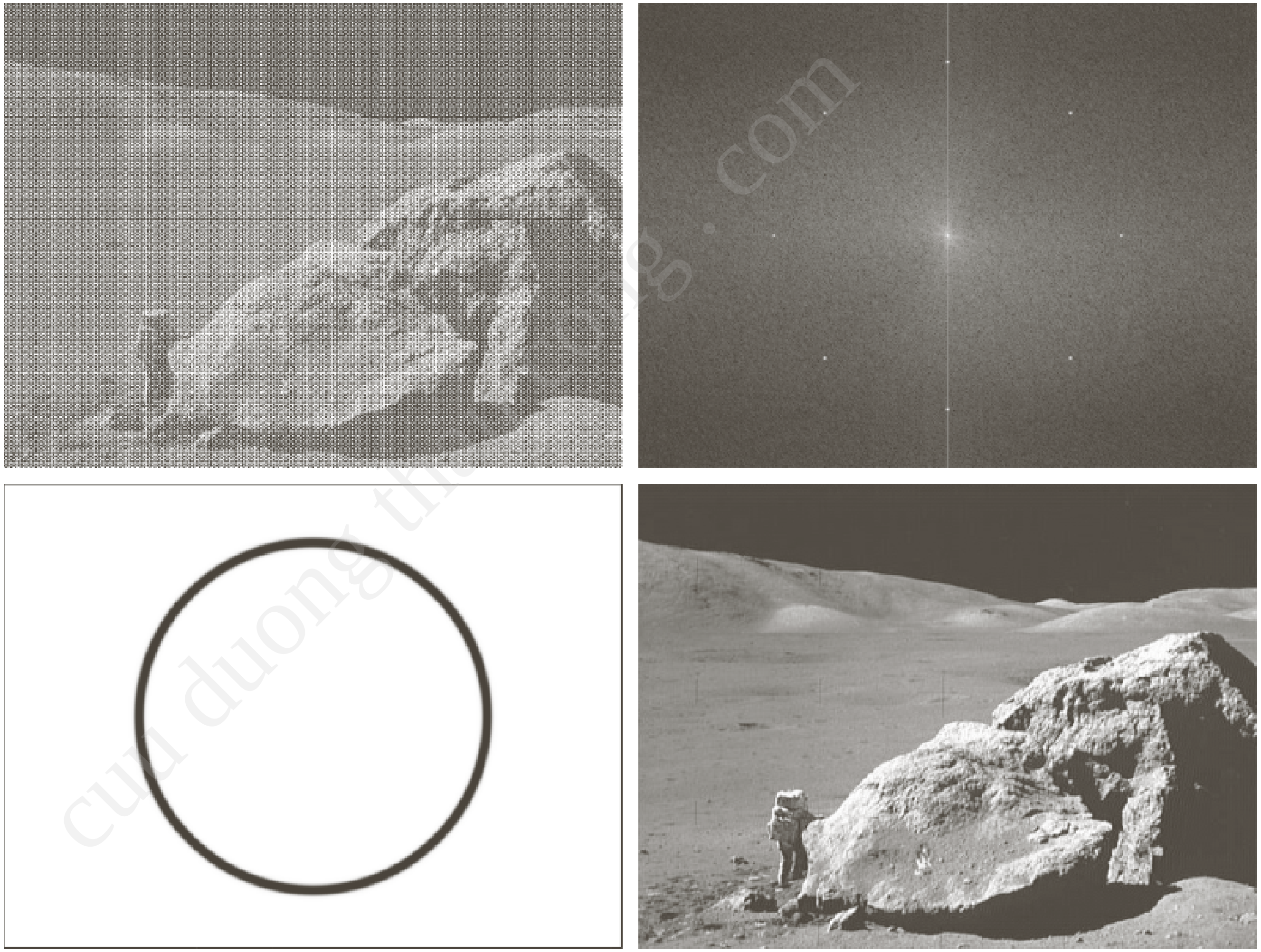


FIGURE 5.16 (a) Image corrupted by sinusoidal noise. (b) Spectrum of (a). (c) Butterworth bandreject filter (white represents 1). (d) Result of filtering. (Original image courtesy of NASA.)

2. IR: Restoration in the Presence of Noise Only (23)

Notch filters:

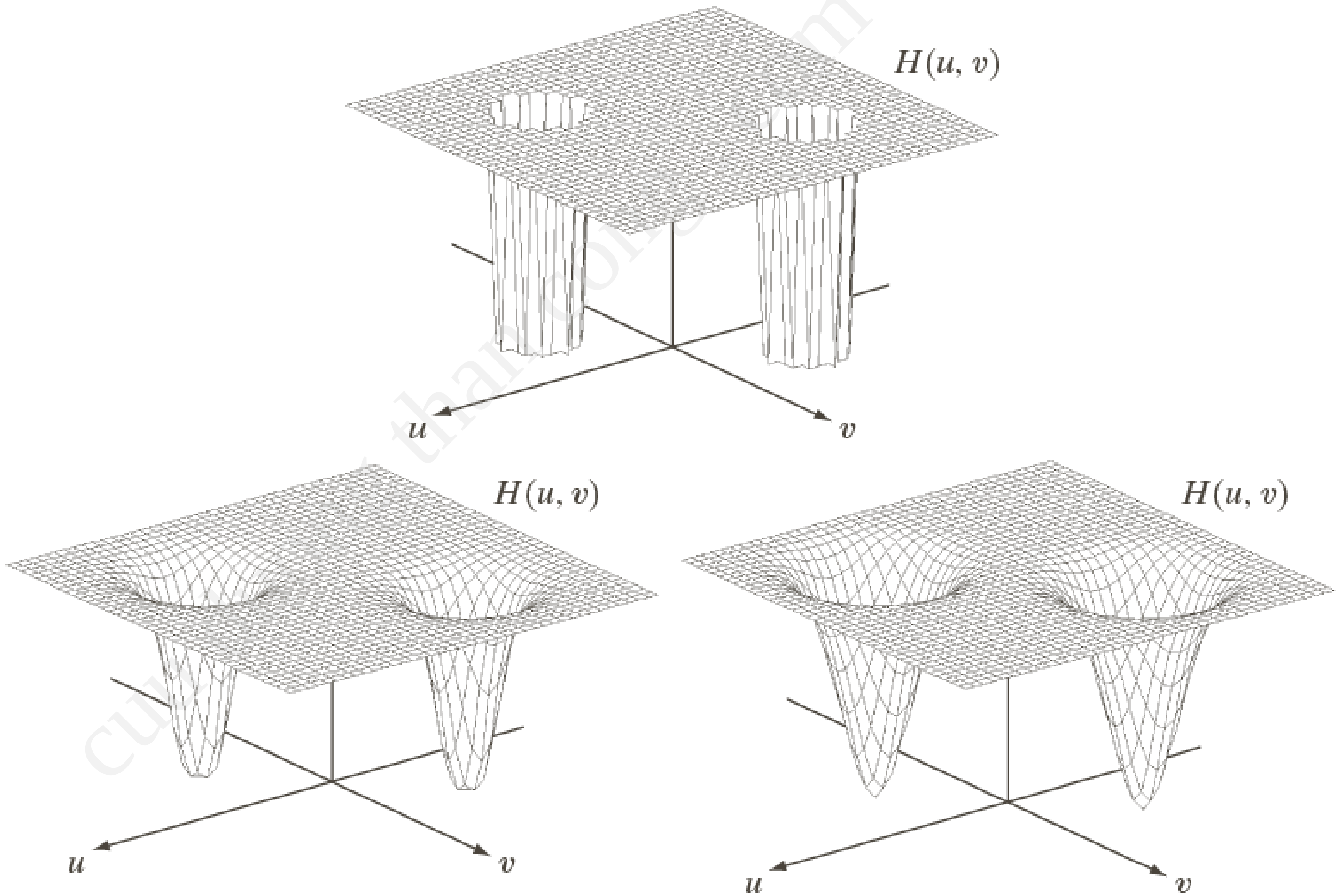


FIGURE 5.18 Perspective plots of (a) ideal, (b) Butterworth (of order 2), and (c) Gaussian notch (reject) filters.

2. IR: Restoration in the Presence of Noise Only (24)

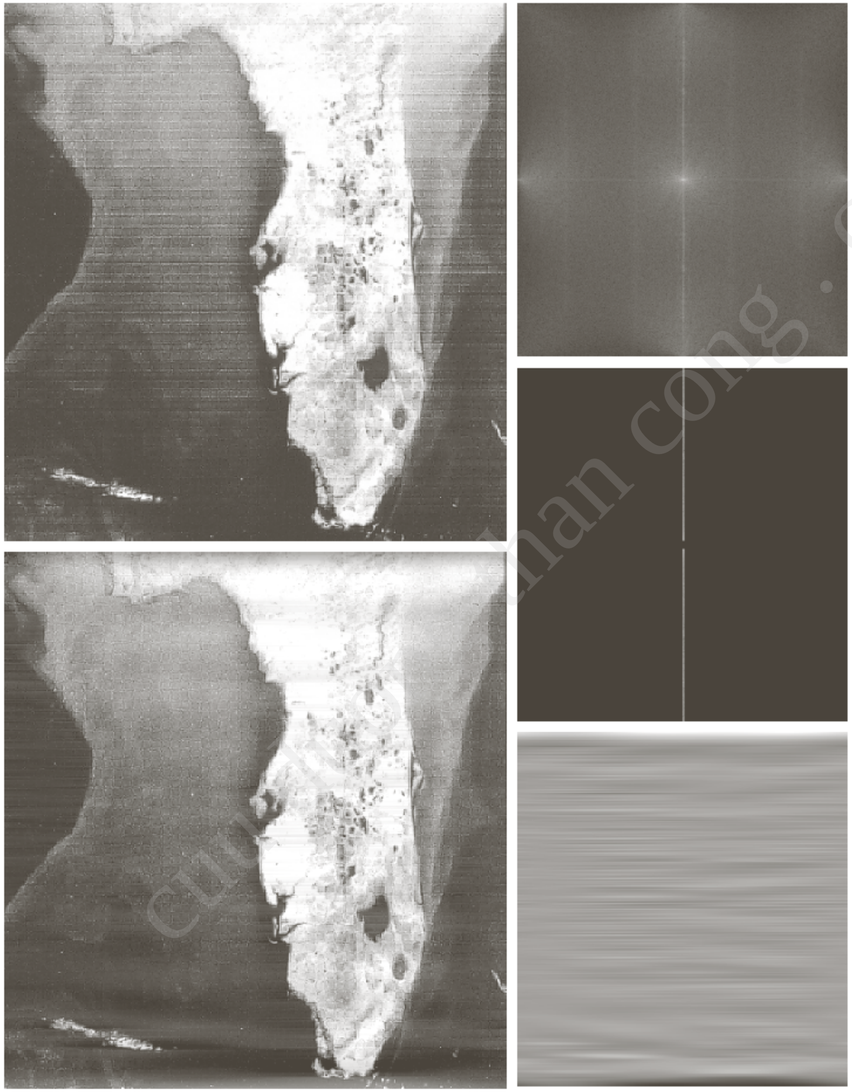


FIGURE 5.19 (a) Satellite image of Florida and the Gulf of Mexico showing horizontal scan lines. (b) Spectrum. (c) Notch pass filter superimposed on (b). (d) Spatial noise pattern. (e) Result of notch reject filtering. (Original image courtesy of NOAA.)

2. IR: Restoration in the Presence of Noise Only (25)

- **<u>Optimum notch filtering</u>**:

  It minimizes local variances of the restored estimated

  $$\hat{f}(x, y)$$

  Procedure for restoration tasks in multiple periodic interference:

  – Isolate the principal contributions of the interference pattern.

  – Subtract a variable, weighted portion of the pattern from the corrupted image.

2. IR: Restoration in the Presence of Noise Only (26)

– Step 1: Extract the principal frequency components of the interference pattern

Place a notch pass filter at the location of each spike.

$$N(u,v) = H_{NP}(u,v)G(u,v)$$

$$\eta(x,y) = \Im^{-1}\left\{H_{NP}(u,v)G(u,v)\right\}$$

2. IR: Restoration in the Presence of Noise Only (27)

– <u>Step 2:</u>

Filtering procedure usually yields only an approximation of the true pattern. The effect of components not present in the estimate of $\eta(x, y)$ can be minimized instead by subtracting from $g(x, y)$ a weighted portion of $\eta(x, y)$ to obtain an estimate of $f(x, y)$:

$$\hat{f}(x, y) = g(x, y) - w(x, y)\eta(x, y)$$

One approach is to select $w(x, y)$ so that the variance of the estimate $\hat{f}(x, y)$ is minimized over a specified neighborhood of every point (x, y).

2. IR: Restoration in the Presence of Noise Only (28)

The local variance of $\hat{f}(x, y)$:

$$\sigma^2(x,y) = \frac{1}{(2a+1)(2b+1)} \sum_{s=-a}^{a} \sum_{t=-b}^{b} \left[\hat{f}(x+s, y+t) - \overline{\hat{f}}(x,y) \right]^2$$

$$= \frac{1}{(2a+1)(2b+1)} \sum_{s=-a}^{a} \sum_{t=-b}^{b} \left\{ \begin{array}{l} \left[g(x+s,y+t) - w(x+s,y+t)\eta(x+s,y+s) \right] \\ - \left[\overline{g}(x,y) - \overline{w(x,y)\eta(x,y)} \right] \end{array} \right\}^2$$

$$= \frac{1}{(2a+1)(2b+1)} \sum_{s=-a}^{a} \sum_{t=-b}^{b} \left\{ \begin{array}{l} \left[g(x+s,y+t) - w(x,y)\eta(x+s,y+s) \right] \\ - \left[\overline{g}(x,y) - w(x,y)\overline{\eta(x,y)} \right] \end{array} \right\}^2$$

2. IR: Restoration in the Presence of Noise Only (29)

To minimize $\sigma^2(x,y)$, $\dfrac{\partial \sigma^2(x,y)}{\partial w(x,y)} = 0$

for $w(x,y)$, the result is

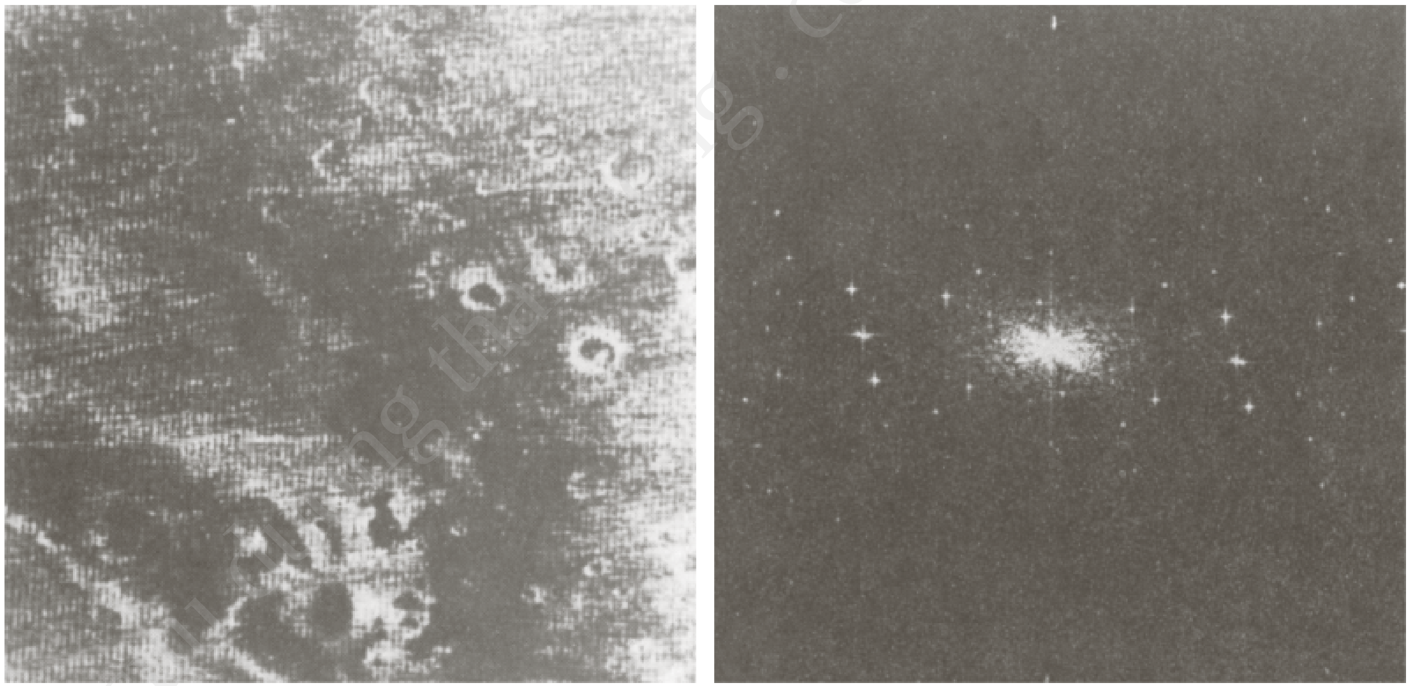
$$w(x,y) = \frac{\overline{g(x,y)\eta(x,y)} - \overline{g}(x,y)\overline{\eta}(x,y)}{\overline{\eta^2}(x,y) - \overline{\eta}^2(x,y)}$$

2. IR: Restoration in the Presence of Noise Only (30)

Example of optimum notch filtering:

a b

FIGURE 5.20 (a) Image of the Martian terrain taken by *Mariner 6*. (b) Fourier spectrum showing periodic interference. (Courtesy of NASA.)



2. IR: Restoration in the Presence of Noise Only (31)

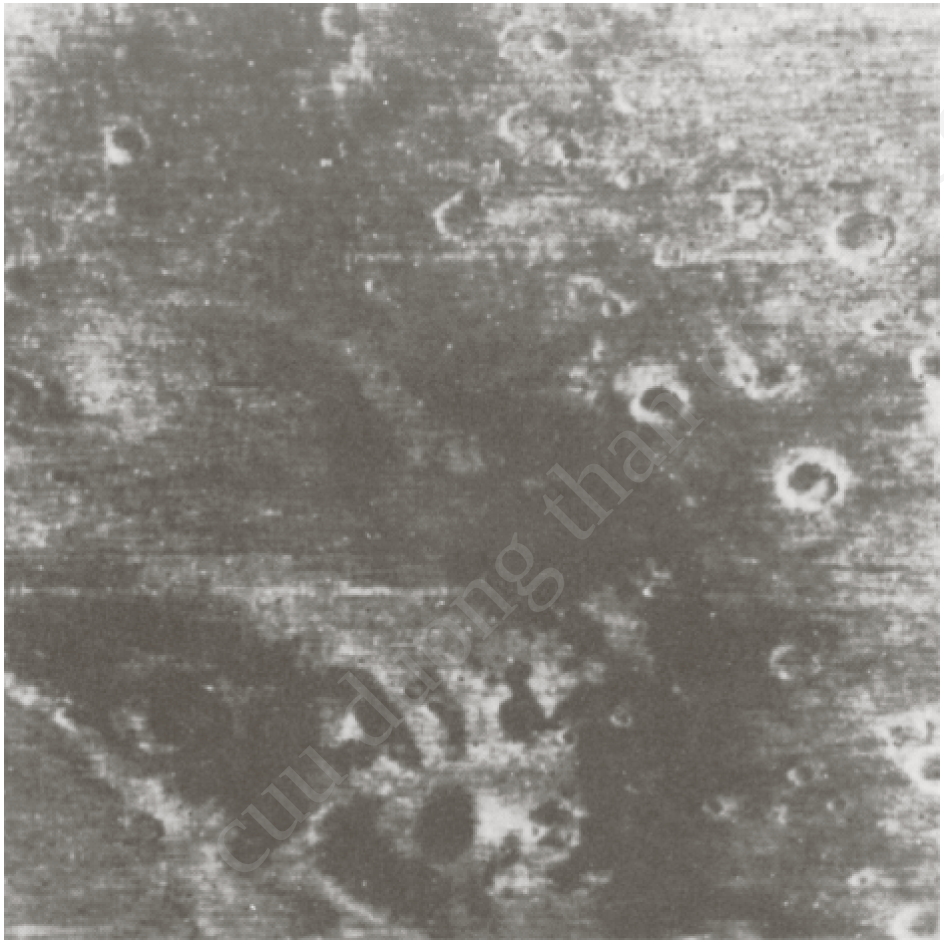
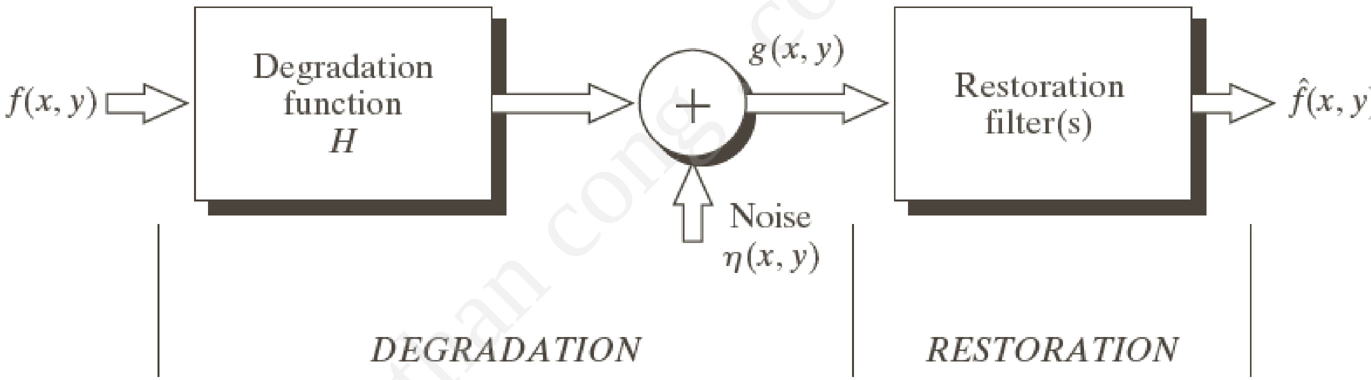


FIGURE 5.23 Processed image. (Courtesy of NASA.)

2. IR: Estimating Degradation Function (1)

❑ Linear, position-invariant degradations

FIGURE 5.1
A model of the image degradation/restoration process.



$$g(x,y) = H\left[f(x,y)\right] + \eta(x,y)$$

2. IR: Estimating Degradation Function (2)

H is linear

$$H\left[af_1(x,y)+bf_2(x,y)\right]=aH\left[f_1(x,y)\right]+bH\left[f_2(x,y)\right]$$

f_1 and f_2 are any two input images.

An operator having the input-output relationship

$g(x,y)=H\left[f(x,y)\right]$ is said to be position invariant

if

$$H\left[f(x-\alpha,y-\beta)\right]=g(x-\alpha,y-\beta)$$

for any $f(x,y)$ and any α and β.

2. IR: Estimating Degradation Function (3)

Assume for a moment that $\eta(x, y) = 0$

if H is a linear operator and position invariant,

$$H\left[\delta(x-\alpha, y-\beta)\right] = h(x-\alpha, y-\beta)$$

$$g(x, y) = H\left[f(x, y)\right]$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha, \beta) H\left[\delta(x-\alpha, y-\beta)\right] d\alpha\, d\beta$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha, \beta) h(x-\alpha, y-\beta) d\alpha\, d\beta$$

Convolution integral in 2-D

2. IR: Estimating Degradation Function (4)

In the presence of additive noise,

if H is a linear operator and position invariant,

$$g(x,y)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha,\beta)h(x-\alpha,y-\beta)d\alpha d\beta+\eta(x,y)$$

$$= h(x,y) \;*\; f(x,y)+\eta(x,y)$$

$$G(u,v)=H(u,v)F(u,v)+N(u,v)$$

2. IR: Estimating Degradation Function (5)

❑ Estimating the degradation function

Three principal ways to estimate the degradation function:

1. Observation
2. Experimentation
3. Mathematical Modeling

2. IR: Estimating Degradation Function (6)

- **Mathematical modeling:**
  - Environmental conditions cause degradation. A model about atmospheric turbulence

$$H(u,v) = e^{-k(u^2+v^2)^{5/6}}$$

k : a constant that depends on the nature of the turbulence.

  - Derive a mathematical model from basic principles.

    Example: An image blurred by uniform linear motion between the image and the sensor during image acquisition.

2. IR: Estimating Degradation Function (7)

a b
c d

FIGURE 5.25 Illustration of the atmospheric turbulence model. (a) Negligible turbulence. (b) Severe turbulence, $k = 0.0025$. (c) Mild turbulence, $k = 0.001$. (d) Low turbulence, $k = 0.00025$. (Original image courtesy of NASA.)

2. IR: Estimating Degradation Function (8)

Suppose that an image $f(x, y)$ undergoes planar motion, $x_0(t)$ and $y_0(t)$ are the time-varying components of motion in the x- and y-directions, respectively.

The optical imaging process is perfect. T is the duration of the exposure. The blurred image $g(x, y)$

$$g(x, y) = \int_0^T f\left[x - x_0(t), y - y_0(t)\right] dt$$

2. IR: Estimating Degradation Function (9)

$$g(x,y)=\int_0^T f\left[x-x_0(t),y-y_0(t)\right]dt$$

$$G(u,v)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x,y)e^{-j2\pi(ux+vy)}dxdy$$

$$=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left[\int_0^T f\left[x-x_0(t),y-y_0(t)\right]dt\right]e^{-j2\pi(ux+vy)}dxdy$$

$$=\int_0^T\left[\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f\left[x-x_0(t),y-y_0(t)\right]e^{-j2\pi(ux+vy)}dxdy\right]dt$$

$$=\int_0^T F(u,v)e^{-j2\pi\left[ux_0(t)+vy_0(t)\right]}dt$$

$$=F(u,v)\int_0^T e^{-j2\pi\left[ux_0(t)+vy_0(t)\right]}dt$$

2. IR: Estimating Degradation Function (10)

$$H(u,v)=\int_0^T e^{-j2\pi[ux_0(t)+vy_0(t)]}dt$$

Suppose that the image undergoes uniform linear motion in the x-direction only, at a rate given by $x_0(t)=at/T$.

$$H(u,v)=\int_0^T e^{-j2\pi ux_0(t)}dt$$

$$=\int_0^T e^{-j2\pi uat/T}dt$$

$$=\frac{T}{\pi ua}\sin(\pi ua)e^{-j\pi ua}$$

2. IR: Estimating Degradation Function (11)

Suppose that the image undergoes uniform linear motion in the x-direction and y-direction, at a rate given by $x_0(t) = at/T$ and $y_0(t) = bt/T$

$$H(u,v) = \int_0^T e^{-j2\pi[ux_0(t)+vy_0(t)]}dt$$

$$= \int_0^T e^{-j2\pi[ua+vb]t/T}dt$$

$$= \frac{T}{\pi(ua+vb)}\sin[\pi(ua+vb)]e^{-j\pi(ua+vb)}$$

2. IR: Estimating Degradation Function (12)

a b

FIGURE 5.26 (a) Original image. (b) Result of blurring using the function in Eq. (5.6-11) with $a = b = 0.1$ and $T = 1$.

2. IR: Inverse Filtering

An estimate of the transform of the original image

$$\hat{F}(u,v) = \frac{G(u,v)}{H(u,v)}$$

$$\hat{F}(u,v) = \frac{F(u,v)H(u,v) + N(u,v)}{H(u,v)}$$

$$= F(u,v) + \frac{N(u,v)}{H(u,v)}$$

1. We can't exactly recover the undegraded image because $N(u,v)$ is not known.
2. If the degradation function has zero or very small values, then the ratio $N(u,v)/H(u,v)$ could easily dominate the estimate $\hat{F}(u,v)$.

2. IR: Wiener Filtering (1)

❑ Objective

Minimum mean-square-error filter (**Wiener filter**): Find an estimate of the uncorrupted image such that the mean square error between them is minimized.

$$e^2 = E\left\{(f - \hat{f})^2\right\}$$

2. IR: Wiener Filtering (2)

The minimum of the error function is given in the frequency domain by the expression

$$\hat{F}(u,v) = \left[\frac{H^*(u,v) S_f(u,v)}{S_f(u,v)\,|H(u,v)|^2 + S_\eta(u,v)} \right] G(u,v)$$

$$= \left[\frac{H^*(u,v)}{|H(u,v)|^2 + S_\eta(u,v) / S_f(u,v)} \right] G(u,v)$$

$$= \left[\frac{1}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + S_\eta(u,v) / S_f(u,v)} \right] G(u,v)$$

2. IR: Wiener Filtering (3)

$$\hat{F}(u,v) = \left[\frac{1}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + S_\eta(u,v)/S_f(u,v)} \right] G(u,v)$$

$H(u,v)$: degradation function

$H^*(u,v)$: complex conjugate of $H(u,v)$

$|H(u,v)|^2 = H^*(u,v)H(u,v)$

$S_\eta(u,v) = |N(u,v)|^2 =$ power spectrum of the noise

$S_f(u,v) = |F(u,v)|^2 =$ power spectrum of the undegraded image

2. IR: Wiener Filtering (4)

$$\hat{F}(u,v) = \left[\frac{1}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + K} \right] G(u,v)$$

K is a specified constant. Generally, the value of K is chosen interactively to yield the best visual results.

2. IR: Wiener Filtering (5)

a b c

FIGURE 5.28 Comparison of inverse and Wiener filtering. (a) Result of full inverse filtering of Fig. 5.25(b). (b) Radially limited inverse filter result. (c) Wiener filter result.

2. IR: Wiener Filtering (6)

Left: Degraded image

Middle: Inverse filtering

Right: Wiener filtering

2. IR: Measures (1)

Singal - to - Noise Ratio (SNR)

$$SNR = \frac{\sum_{u=0}^{M-1}\sum_{v=0}^{N-1} |F(u,v)|^2}{\sum_{u=0}^{M-1}\sum_{v=0}^{N-1} |N(u,v)|^2}$$

This ratio gives a measure of the level of information bearing singal power to the level of noise power.

2. IR: Measures (2)

Mean Square Error (MSE)

$$\text{MSE}=\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left[f(x,y)-\hat{f}(x,y)\right]^2$$

Root - Mean - Square - Error (RMSE)

$$\text{RMSE}=\frac{\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}\hat{f}(x,y)^2}{\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}|f(x,y)-\hat{f}(x,y)|^2}$$

2. IR: Constrained Least Squares Filtering (1)

- In **Wiener filter**, the **power spectra of the undegraded image and noise must be known**. Although a constant estimate is sometimes useful, it is not always suitable.
- **Constrained least squares filtering** just requires the **mean** and **variance of the noise**.

2. IR: Constrained Least Squares Filtering (2)

$$F(u,v) = \left[\frac{H^*(u,v)}{|H(u,v)|^2 + \gamma |P(u,v)|^2} \right] G(u,v)$$

$P(u,v)$ is the Fourier transform of the function

$$p(x,y) = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

γ is a parameter.

2. IR: Constrained Least Squares Filtering (3)

a b c

FIGURE 5.30 Results of constrained least squares filtering. Compare (a), (b), and (c) with the Wiener filtering results in Figs. 5.29(c), (f), and (i), respectively.

2. IR: Geometric Mean Filter

$$\hat{F}(u,v) = \left[\frac{H^*(u,v)}{|H(u,v)|^2}\right]^{\alpha} \left[\frac{|H(u,v)|^2}{|H(u,v)|^2 + \beta\left[S_\eta(u,v)/S_f(u,v)\right]}\right]^{1-\alpha} G(u,v)$$

$\alpha = 1$: inverse filter

$\alpha = 0$: parametric Wiener filter

$\alpha = 1/2$: geometric mean filter

2. IR: Image Reconstruction from Projection (1)

- Reconstruct an image from a series of projections X-ray **computed tomography** (**CT**).

"Computed tomography is a medical imaging method employing tomography where digital geometry processing is used to generate a three-dimensional image of the internals of an object from a large series of two-dimensional X-ray images taken around a single axis of rotation."

http://en.wikipedia.org/wiki/Computed_tomography

2. IR: Image Reconstruction from Projection (2)

“ In computed tomography or other imaging techniques requiring reconstruction from multiple projections, an algorithm for calculating the contribution of each voxel of the structure to the measured ray data, to generate an image; the oldest and simplest method of image reconstruction.“

http://www.medilexicon.com/medicaldictionary.php?t=9165

2. IR: Image Reconstruction from Projection (3)

❑ Image reconstruction: Introduction

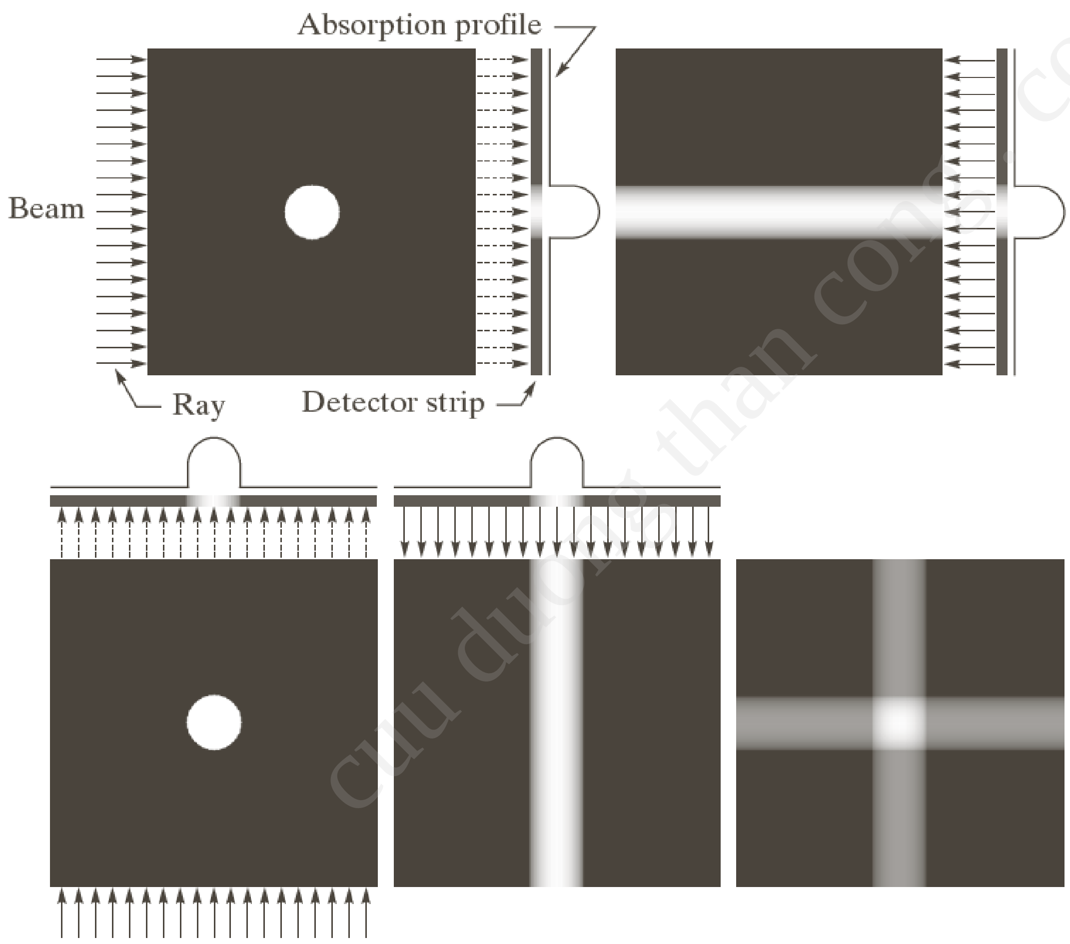
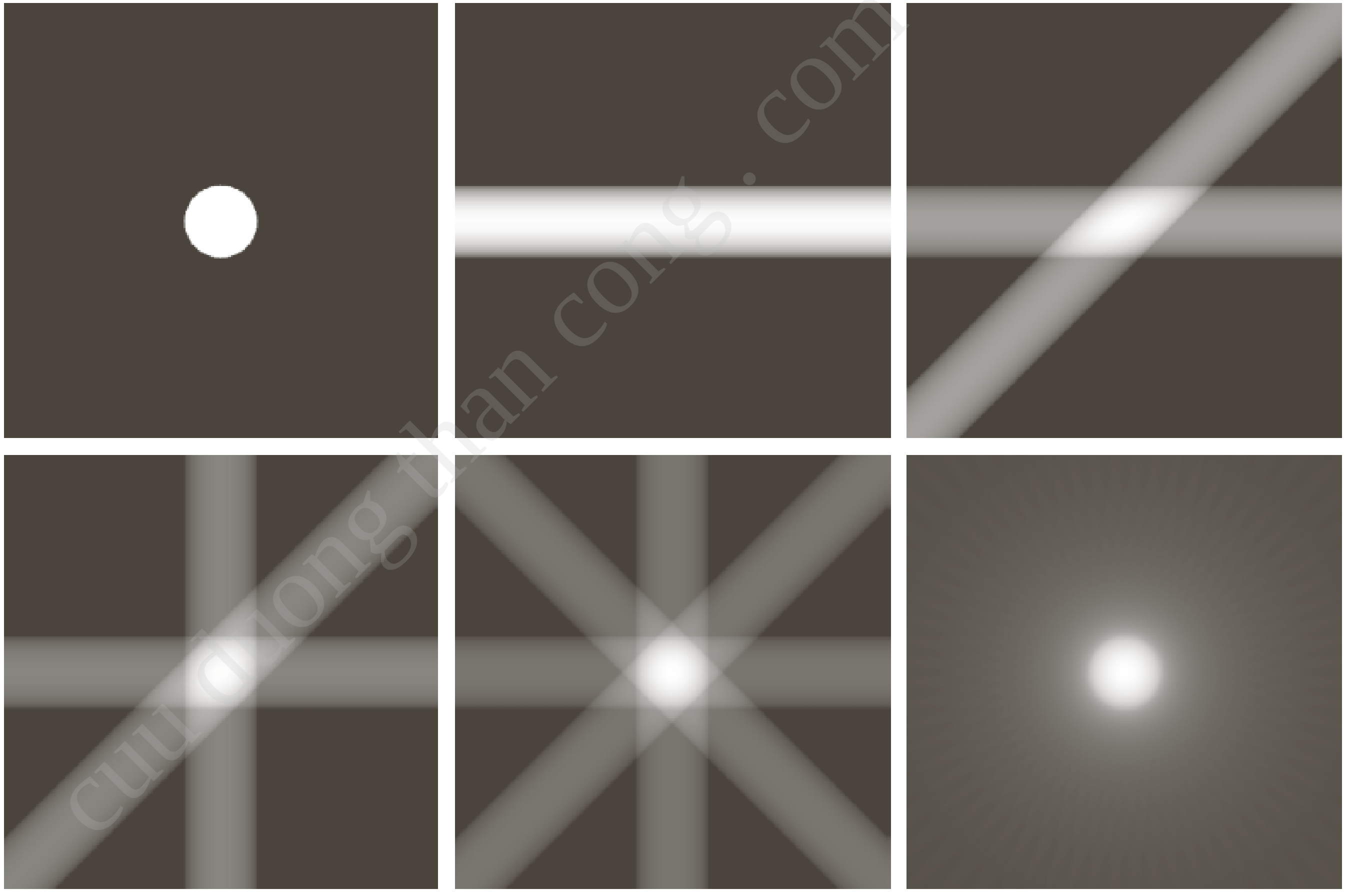


FIGURE 5.32
(a) Flat region showing a simple object, an input parallel beam, and a detector strip.
(b) Result of back-projecting the sensed strip data (i.e., the 1-D absorption profile). (c) The beam and detectors rotated by 90°.
(d) Back-projection.
(e) The sum of (b) and (d). The intensity where the back-projections intersect is twice the intensity of the individual back-projections.

2. IR: Image Reconstruction from Projection (4)

a	b	c
d	e	f

FIGURE 5.33
(a) Same as Fig. 5.32(a).
(b)–(e) Reconstruction using 1, 2, 3, and 4 backprojections 45° apart.
(f) Reconstruction with 32 backprojections 5.625° apart (note the blurring).



2. IR: Image Reconstruction from Projection (5)

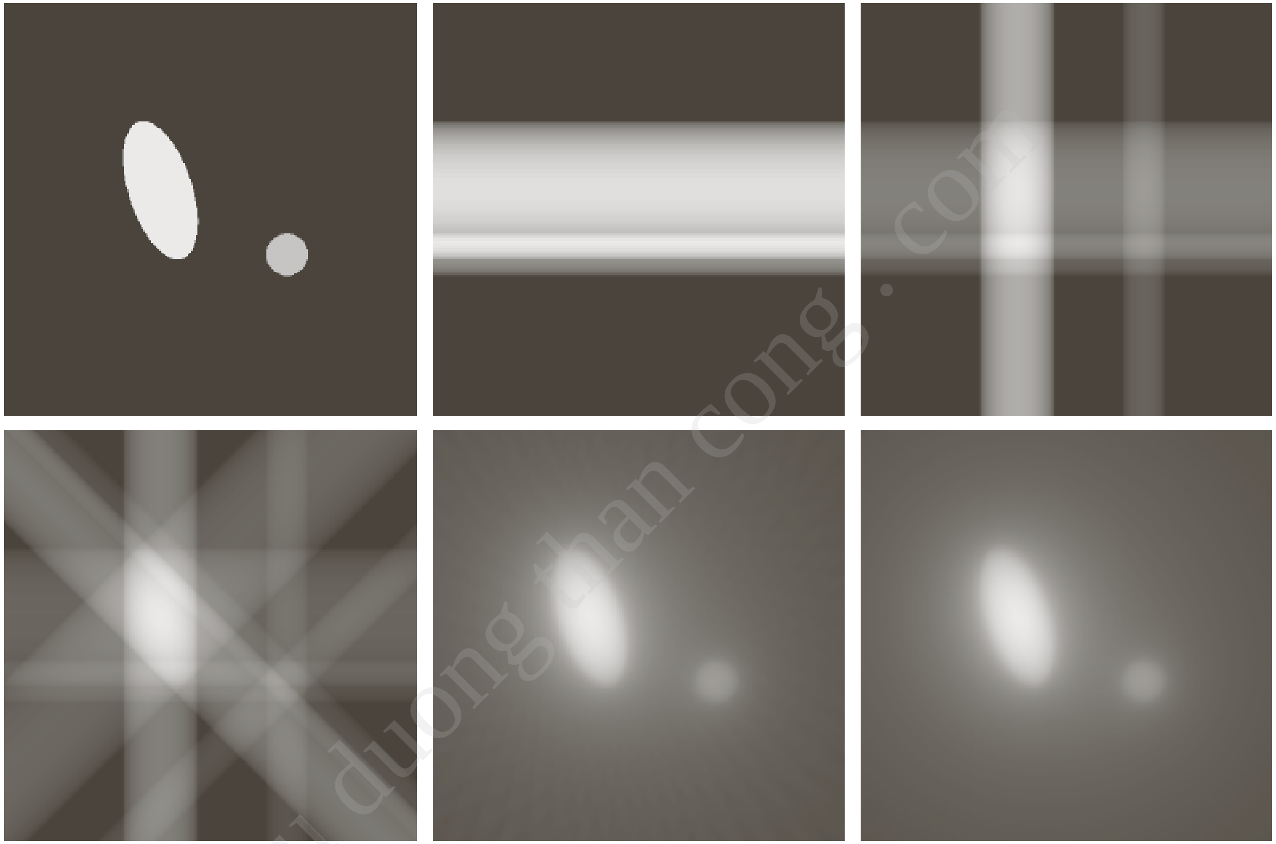
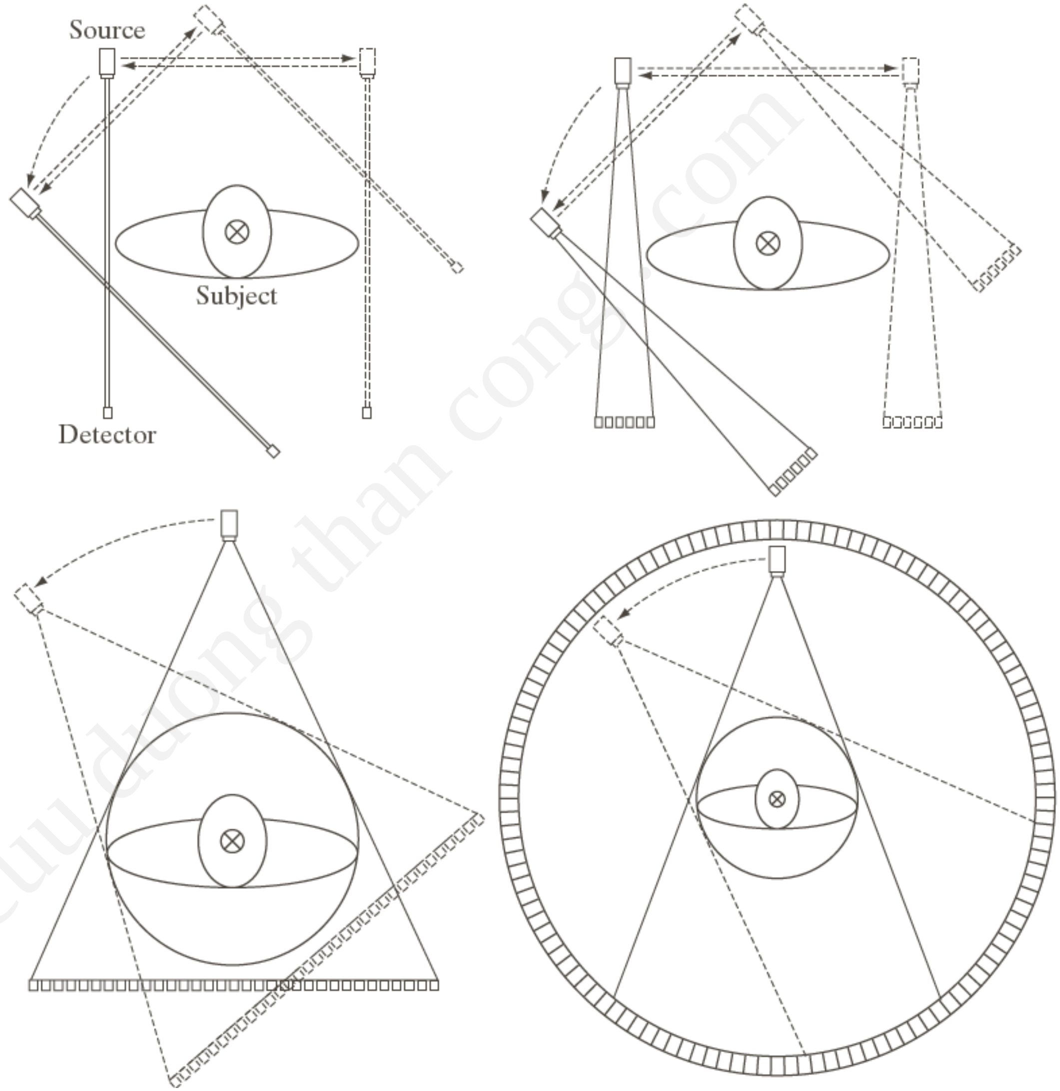


FIGURE 5.34 (a) A region with two objects. (b)–(d) Reconstruction using 1, 2, and 4 backprojections 45° apart. (e) Reconstruction with 32 backprojections 5.625° apart. (f) Reconstruction with 64 backprojections 2.8125° apart.

2. IR: Image Reconstruction from Projection (6)



a b
c d

FIGURE 5.35 Four generations of CT scanners. The dotted arrow lines indicate incremental linear motion. The dotted arrow arcs indicate incremental rotation. The cross-mark on the subject's head indicates linear motion perpendicular to the plane of the paper. The double arrows in (a) and (b) indicate that the source/detector unit is translated and then brought back into its original position.

❑ Projections and the Radon transform

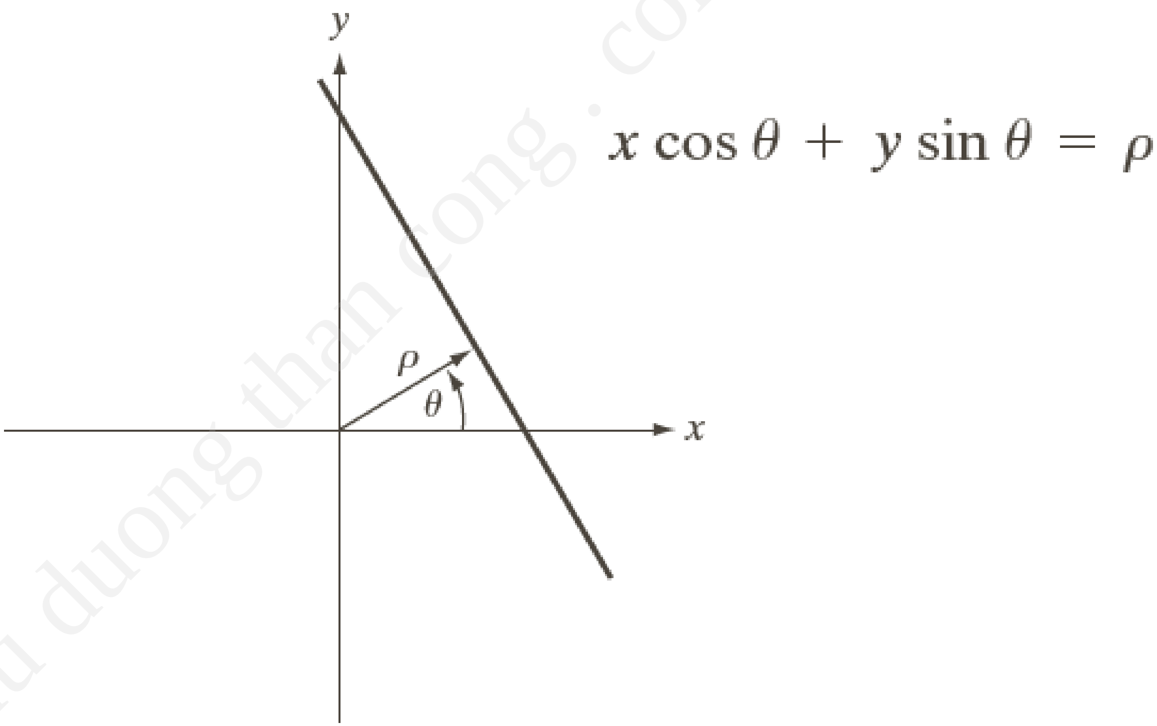
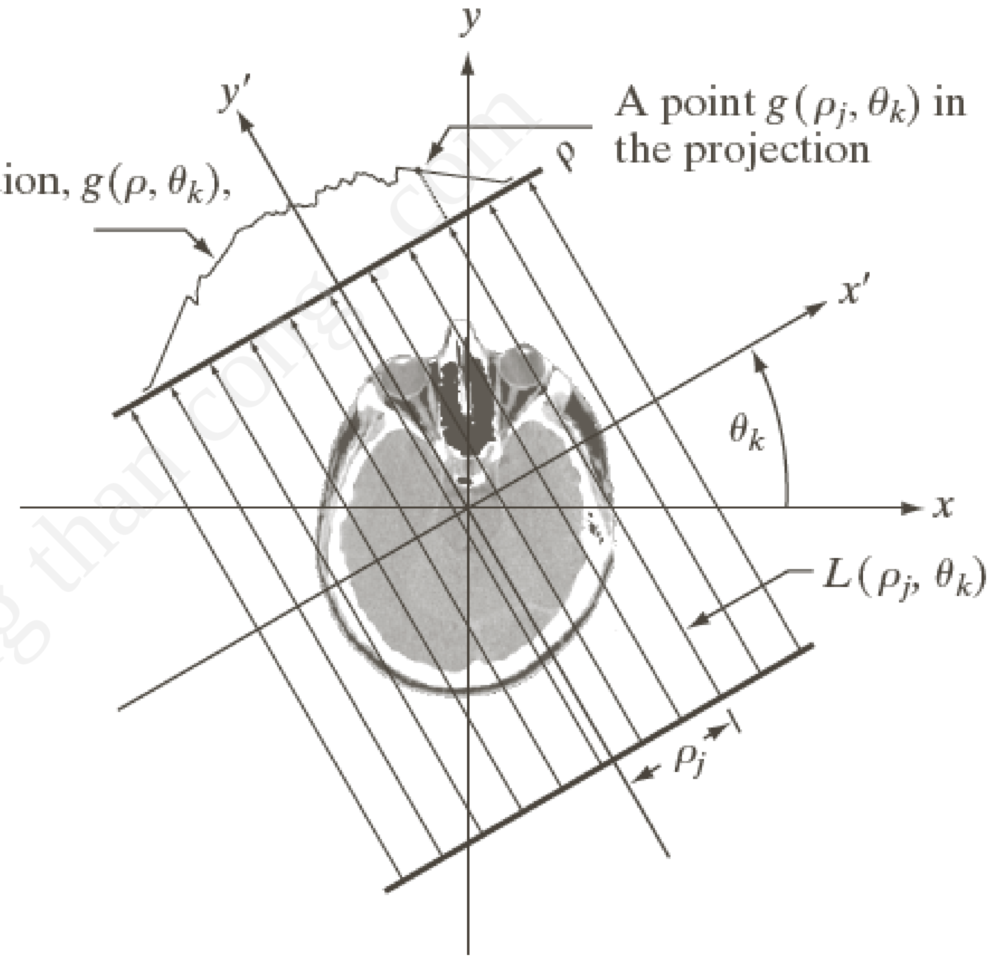


FIGURE 5.36 Normal representation of a straight line.

2. IR: Image Reconstruction from Projection (8)

FIGURE 5.37
Geometry of a parallel-ray beam.



$$g(\rho_j, \theta_k) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x, y)\delta(x\cos\theta_k + y\sin\theta_k - \rho_j)dxdy$$

2. IR: Image Reconstruction from Projection (9)

Radon transform gives the projection (line integral) of $f(x,y)$ along an arbitrary line in the xy-plane

$$\Re\{f\} = g(\rho,\theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\delta(x\cos\theta + y\sin\theta - \rho)dxdy$$

$$\Re\{f\} = g(\rho,\theta) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)\delta(x\cos\theta + y\sin\theta - \rho)$$

2. IR: Image Reconstruction from Projection (10)

Example: Using the Radon transform to obtain the projection of a circular region.

Assume that the circle is centered on the origin of the *xy*-plane. Because the object is circularly symmetric, its projections are the same for all angles, so we just check the projection for $\theta = 0^\circ$

$$f(x, y) = \begin{cases} A & x^2 + y^2 \le r^2 \\ 0 & \text{otherwise} \end{cases}$$

2. IR: Image Reconstruction from Projection (11)

$$g(\rho,\theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\delta(x\cos\theta + y\sin\theta - \rho)dxdy$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\delta(x-\rho)dxdy$$

$$= \int_{-\infty}^{\infty} f(\rho,y)dy$$

$$= \int_{-\sqrt{r^2-\rho^2}}^{\sqrt{r^2-\rho^2}} f(\rho,y)dy$$

$$= \int_{-\sqrt{r^2-\rho^2}}^{\sqrt{r^2-\rho^2}} A dy$$

$$= \begin{cases} 2A\sqrt{r^2-\rho^2} & |\rho| \le \mathrm{r} \\ 0 & \text{otherwise} \end{cases}$$

2. IR: Image Reconstruction from Projection (12)

$$g(\rho) = \begin{cases} 2A\sqrt{r^2 - \rho^2} & |\rho| \leq \text{r} \\ 0 & \text{otherwise} \end{cases}$$

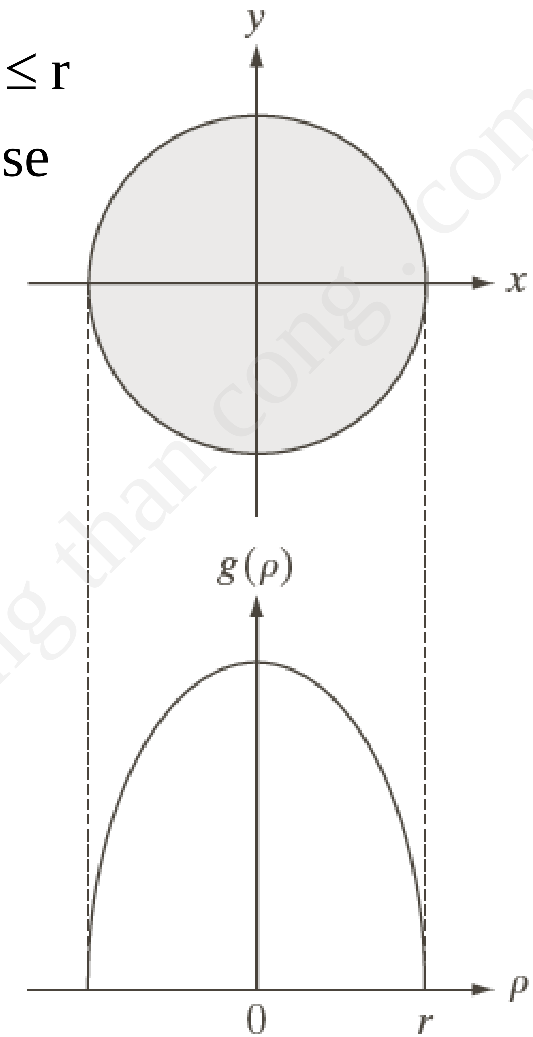


FIGURE 5.38 A disk and a plot of its Radon transform, derived analytically. Here we were able to plot the transform because it depends only on one variable. When g depends on both ρ and θ, the Radon transform becomes an image whose axes are ρ and θ, and the intensity of a pixel is proportional to the value of g at the location of that pixel.

2. IR: Image Reconstruction from Projection (13)

Sinogram: The result of Radon transform

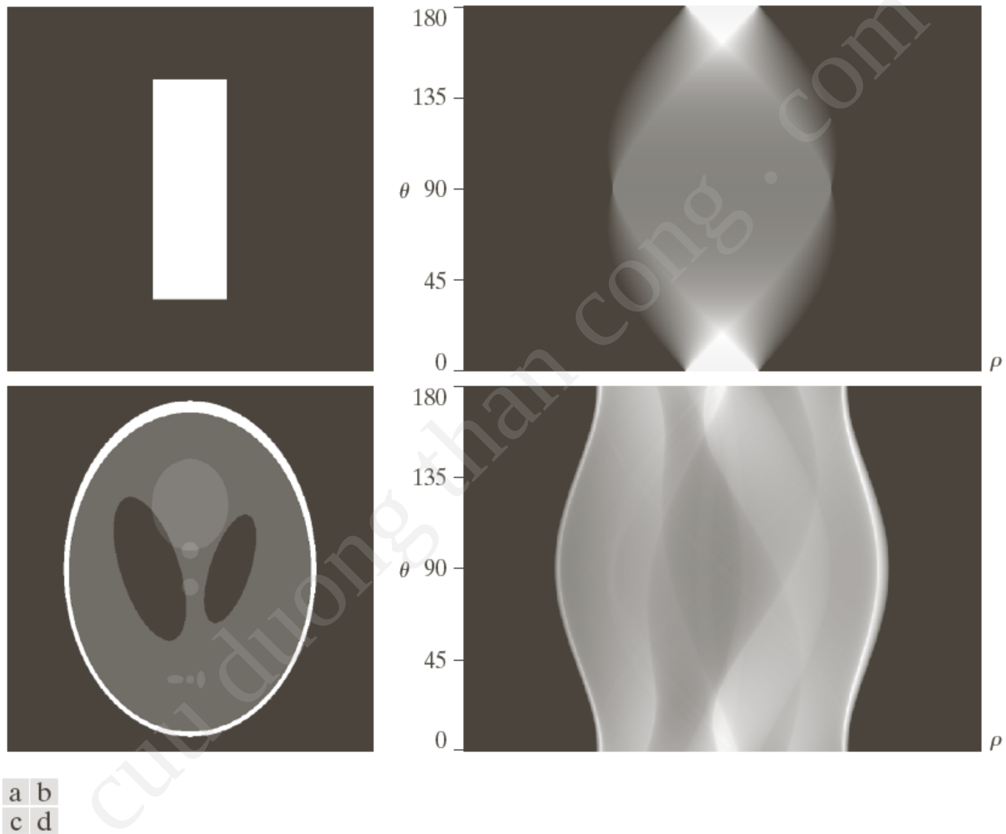


FIGURE 5.39 Two images and their sinograms (Radon transforms). Each row of a sinogram is a projection along the corresponding angle on the vertical axis. Image (c) is called the *Shepp-Logan phantom*. In its original form, the contrast of the phantom is quite low. It is shown enhanced here to facilitate viewing.

2. IR: Image Reconstruction from Projection (14)

$$f_\theta(x,y) = g(x\cos\theta + y\sin\theta, \theta)$$

$$f(x,y) = \int_0^\pi f_\theta(x,y)d\theta$$

$$f(x,y) = \sum_{\theta=0}^{\pi} f_\theta(x,y)$$

A back-projected image formed is referred to as a **laminogram**.

2. IR: Image Reconstruction from Projection (15)

Example: Laminogram

a b

FIGURE 5.40 Backprojections of the sinograms in Fig. 5.39.

2. IR: Image Reconstruction from Projection (16)

❑ The Fourier-slice theorem

For a given value of θ, the 1-D Fourier transform of a projection with respect to ρ is

$$G(w,\theta) = \int_{-\infty}^{\infty} g(\rho,\theta) e^{-j2\pi\omega\rho} d\rho$$

$$G(\omega,\theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\delta(x\cos\theta + y\sin\theta - \rho) e^{-j2\pi\omega\rho} d\rho dxdy$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\left[\int_{-\infty}^{\infty} \delta(x\cos\theta + y\sin\theta - \rho) e^{-j2\pi\omega\rho} d\rho\right] dxdy$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y) e^{-j2\pi\omega(x\cos\theta + y\sin\theta)} dxdy$$

2. IR: Image Reconstruction from Projection (17)

$$G(\omega,\theta)=\int_{-\infty}^{\infty} g(\rho,\theta)e^{-j2\pi\omega\rho}d\rho$$

$$G(\omega,\theta)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-j2\pi\omega(x\cos\theta+y\sin\theta)}dxdy$$

$$=\left[\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-j2\pi(ux+vy)}dxdy\right]_{u=\omega\cos\theta,v=\omega\sin\theta}$$

$$=\left[F(u,v)\right]_{u=\omega\cos\theta,v=\omega\sin\theta}$$

$$=F(\omega\cos\theta,\omega\sin\theta)$$

Fourier - slice theorem: The Fourier transform of a projection is a slice of the 2-D Fourier transform of the region from which the projection was obtained.

2. IR: Image Reconstruction from Projection (18)

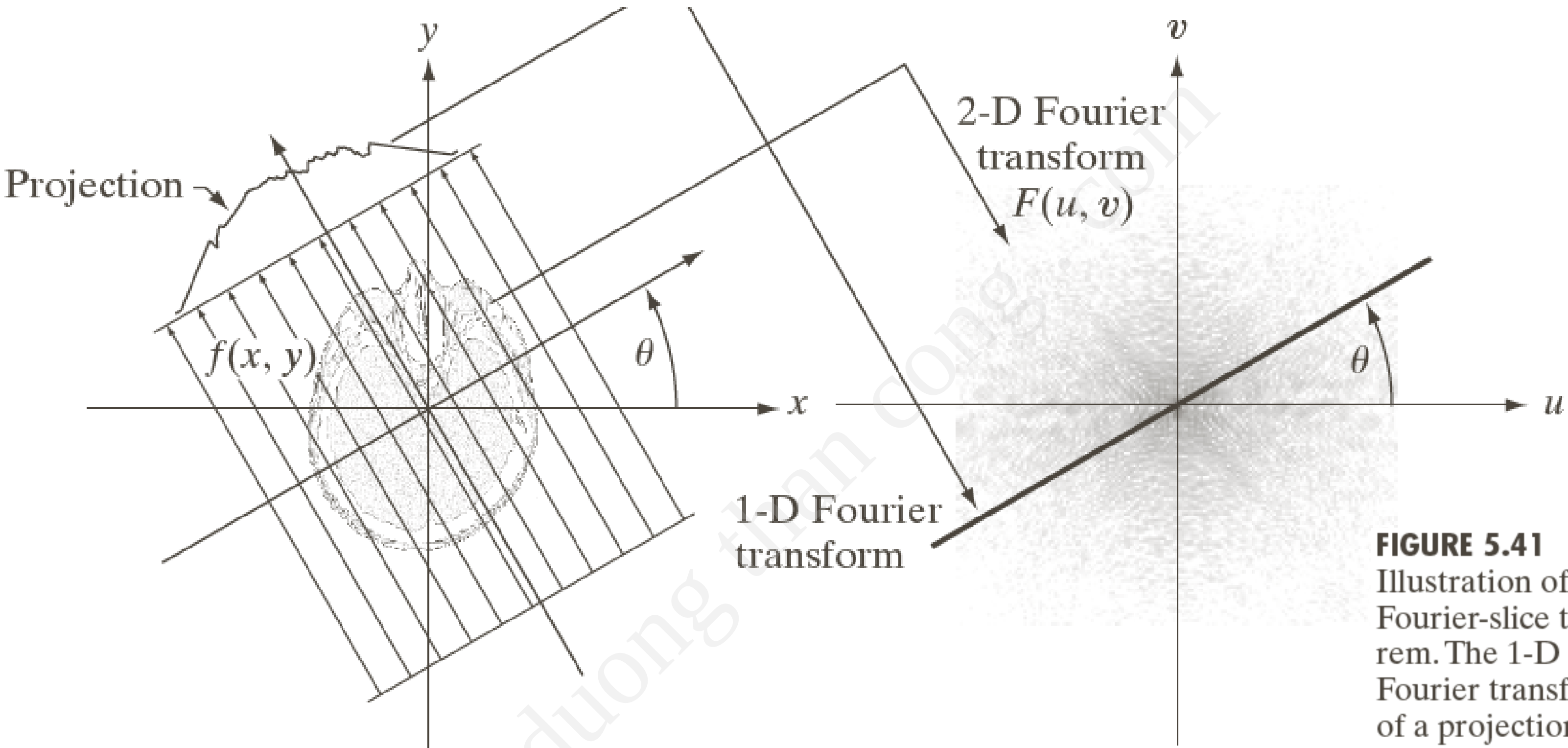


FIGURE 5.41 Illustration of the Fourier-slice theorem. The 1-D Fourier transform of a projection is a slice of the 2-D Fourier transform of the region from which the projection was obtained. Note the correspondence of the angle θ.

2. IR: Image Reconstruction from Projection (19)

- **<u>Reconstruction using parallel-beam filtered back-projections</u>**

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

Let $u=\omega\cos\theta, v=\omega\sin\theta$, then $dudv=\omega d\omega d\theta$,

$$f(x,y)=\int_{0}^{2\pi}\int_{0}^{\infty}F(\omega\cos\theta,\omega\sin\theta)e^{j2\pi\omega(x\cos\theta+y\sin\theta)}\omega d\omega d\theta$$

$$=\int_{0}^{2\pi}\int_{0}^{\infty}G(\omega,\theta)e^{j2\pi\omega(x\cos\theta+y\sin\theta)}\omega d\omega d\theta$$

$$G(\omega,\theta+180^\circ)=G(-\omega,\theta)$$

$$f(x,y)=\int_{0}^{\pi}\int_{-\infty}^{\infty}|\omega|G(\omega,\theta)e^{j2\pi\omega(x\cos\theta+y\sin\theta)}d\omega d\theta$$

2. IR: Image Reconstruction from Projection (20)

$$f(x,y)=\int_0^{\pi}\int_{-\infty}^{\infty}|\omega|G(\omega,\theta)e^{j2\pi\omega(x\cos\theta+y\sin\theta)}d\omega d\theta$$

$$=\int_0^{\pi}\left[\int_{-\infty}^{\infty}|\omega|G(\omega,\theta)e^{j2\pi\omega\rho}d\omega\right]_{\rho=x\cos\theta+y\sin\theta}d\theta$$

Approach: Window the ramp filter (|ω|) so it becomes zero outside of a defined frequency interval. That is, a **window band-limits the ramp filter**.

2. IR: Image Reconstruction from Projection (21)

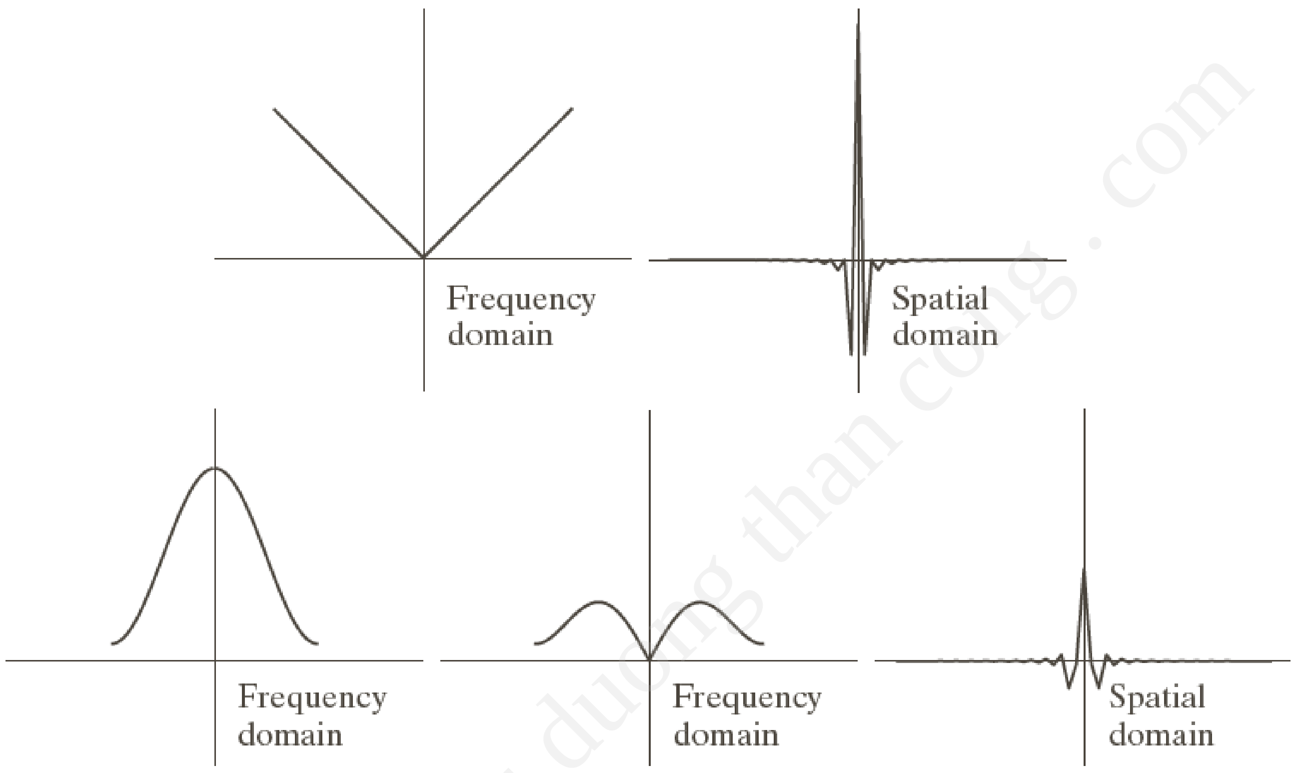
- **Hamming / Hann Windows:**

$$h(w) = \begin{cases} c + (c-1)\cos\dfrac{2\pi w}{M-1} & 0 \le w \le (M-1) \\ 0 & \text{otherwise} \end{cases}$$

$c = 0.54,$ the function is called the <u>Hamming window</u>

$c = 0.5,$ the function is called the <u>Han window</u>

2. IR: Image Reconstruction from Projection (22)



a b
c d e

FIGURE 5.42
(a) Frequency domain plot of the filter $|\omega|$ after band-limiting it with a box filter. (b) Spatial domain representation. (c) Hamming windowing function. (d) Windowed ramp filter, formed as the product of (a) and (c). (e) Spatial representation of the product (note the decrease in ringing).

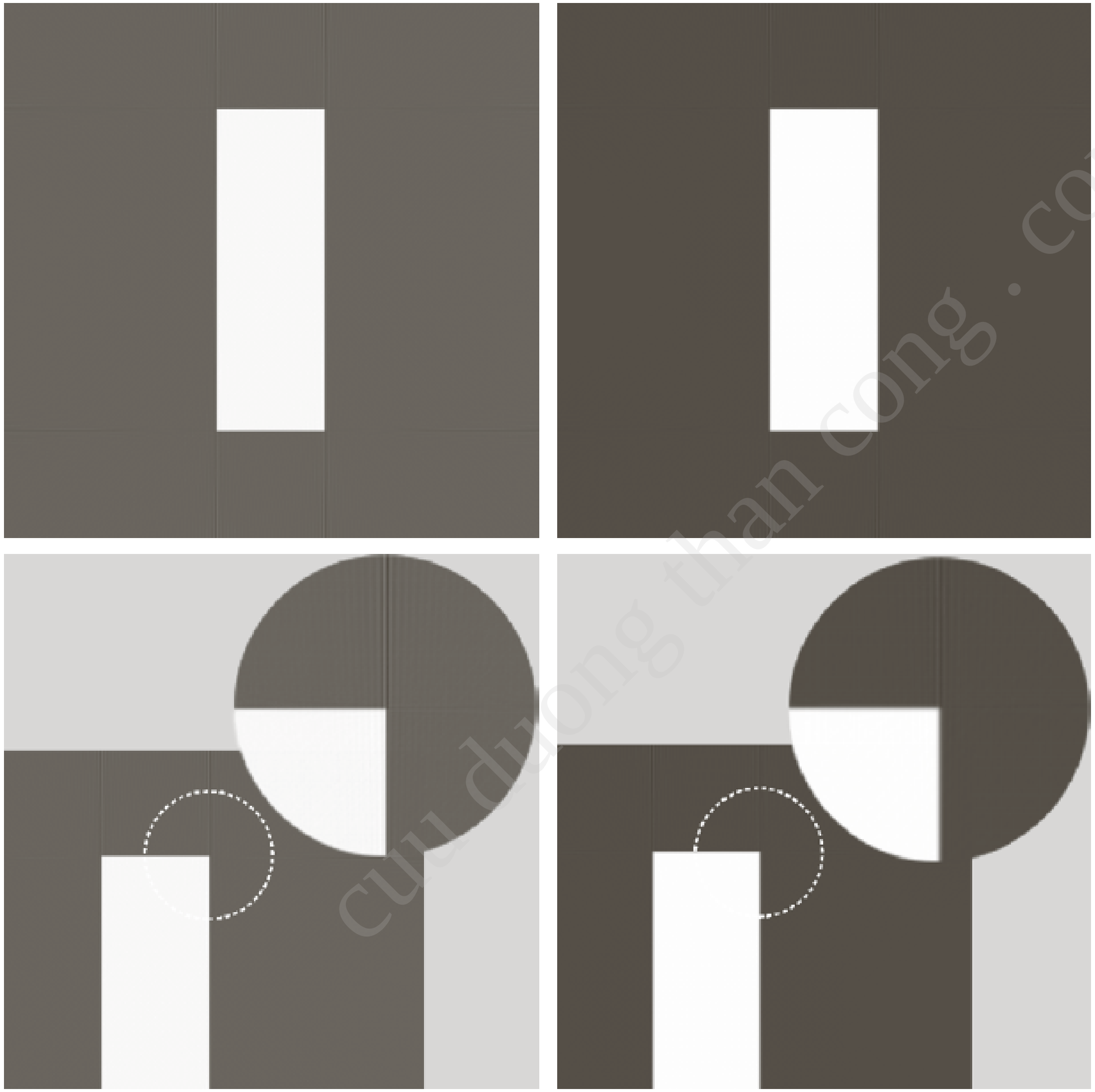
2. IR: Image Reconstruction from Projection (23)

- **Filtered back-projection**

  The complete, filtered back-projection (to obtain the reconstructed image $f(x,y)$) is described as follows:

  1. Compute the 1-D Fourier transform of each projection.
  2. Multiply each Fourier transform by the filter function |w| which has been multiplied by a suitable (e.g., Hamming) window.
  3. Obtain the inverse 1-D Fourier transform of each resulting filtered transform.
  4. Integrate (sum) all the 1-D inverse transforms from step 3.

2. IR: Image Reconstruction from Projection (24)



a b
c d

FIGURE 5.43 Filtered back-projections of the rectangle using (a) a ramp filter, and (b) a Hamming-windowed ramp filter. The second row shows zoomed details of the images in the first row. Compare with Fig. 5.40(a).

2. IR: Image Reconstruction from Projection (25)

a b

FIGURE 5.44 Filtered backprojections of the head phantom using (a) a ramp filter, and (b) a Hamming-windowed ramp filter. Compare with Fig. 5.40(b).

2. IR: Image Reconstruction from Projection (26)

- **Implementation of filtered back-projection in spatial domain**

  Fourier transform of the product of two frequency domain functions is equal to the convolution of the spatial representation.

  Let $s(\rho)$ denote the inverse Fourier transform of $|\omega|$

$$f(x,y) = \int_0^{\pi} \left[\int_{-\infty}^{\infty} |\omega| G(\omega,\theta) e^{j2\pi\omega\rho} d\omega \right]_{\rho = x\cos\theta + y\sin\theta} d\theta$$

$$= \int_0^{\pi} \left[s(\rho) * g(\rho,\theta) \right]_{\rho = x\cos\theta + y\sin\theta} d\theta$$

$$= \int_0^{\pi} \left[\int_{-\infty}^{\infty} g(\rho,\theta) s(x\cos\theta + y\sin\theta - \rho) d\rho \right] d\theta$$

❑ Reconstruction using fan-beam filtered back-projections (for modern CT systems)

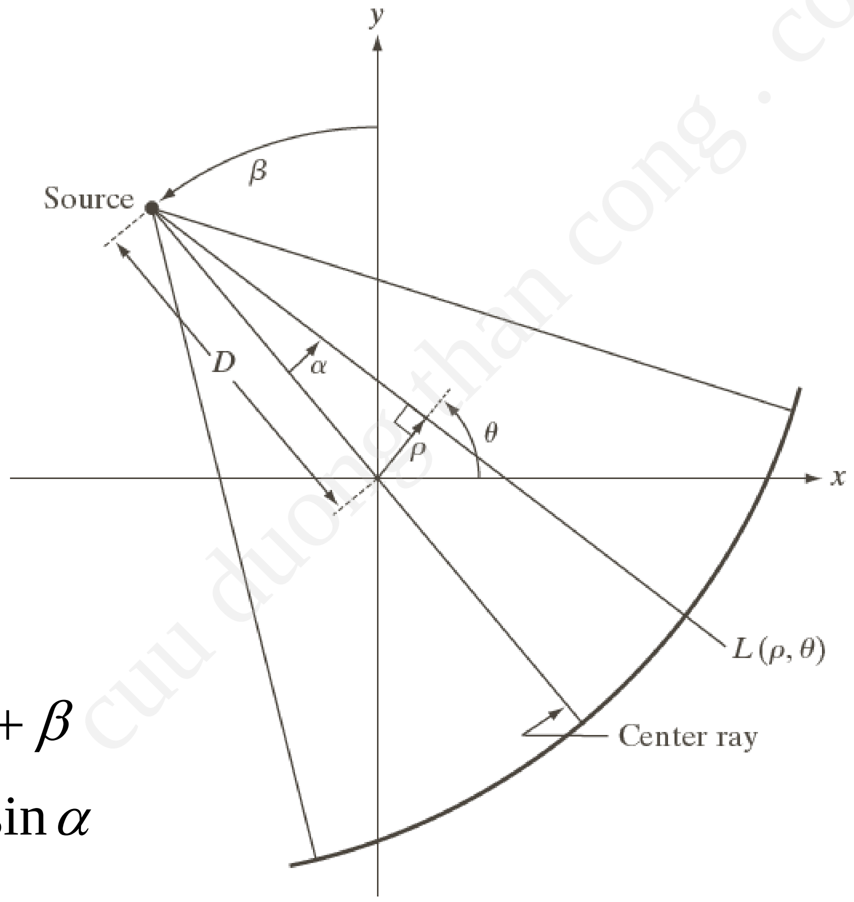


FIGURE 5.45 Basic fan-beam geometry. The line passing through the center of the source and the origin (assumed here to be the center of rotation of the source) is called the *center ray*.

$$\theta = \alpha + \beta$$

$$\rho = D \sin \alpha$$

2. IR: Image Reconstruction from Projection (28)

Objects are encompassed within a circular area of radius T about the origin of the plane, or $g(\rho,\theta) = 0$ for $|\rho| > T$

$$f(x,y) = \int_0^{\pi}\left[\int_{-\infty}^{\infty} g(\rho,\theta)s(x\cos\theta + y\sin\theta - \rho)d\rho\right]d\theta$$

$$= \frac{1}{2}\int_0^{2\pi}\int_{-T}^{T} g(\rho,\theta)s(x\cos\theta + y\sin\theta - \rho)d\rho d\theta$$

$$x = r\cos\varphi;\ y = r\sin\varphi$$

$$x\cos\theta + y\sin\theta = r\cos\varphi\cos\theta + r\sin\varphi\sin\theta$$

$$= r\cos(\varphi - \theta)$$

2. IR: Image Reconstruction from Projection (29)

$$x = r\cos\varphi; y = r\sin\varphi$$

$$x\cos\theta + y\sin\theta = r\cos\varphi\cos\theta + r\sin\varphi\sin\theta$$

$$= r\cos(\varphi - \theta)$$

$$f(x,y) = \frac{1}{2}\int_0^{2\pi}\int_{-T}^{T} g(\rho,\theta)s\left[r\cos(\varphi-\theta)-\rho\right]d\rho d\theta$$

$$\theta = \alpha + \beta \qquad \rho = D\sin\alpha$$

$$d\rho d\theta = D\cos\alpha d\alpha d\beta$$

$$f(x,y) = \frac{1}{2}\int_0^{2\pi}\int_{-T}^{T} g(\rho,\theta)s\left[r\cos(\varphi-\theta)-\rho\right]d\rho d\theta$$

$$= \frac{1}{2}\int_{-\alpha}^{2\pi-\alpha}\int_{-\sin^{-1}(-T/D)}^{\sin^{-1}(T/D)} g(D\sin\alpha, \alpha+\beta)s\left[r\cos(\alpha+\beta-\varphi)-D\sin\alpha\right]$$

$$D\cos\alpha d\alpha d\beta$$

2. IR: Image Reconstruction from Projection (30)

$$f(r,\varphi)=\frac{1}{2}\int_0^{2\pi}\int_{-\alpha_m}^{\alpha_m} p(\alpha,\beta)s\left[R\sin\left(\alpha'-\alpha\right)\right]D\cos\alpha\, d\alpha\, d\beta$$

$$s(R\sin\alpha)=\left(\frac{\alpha}{R\sin\alpha}\right)^2 s(\alpha)$$

$$f(r,\varphi)=\int_0^{2\pi}\frac{1}{R^2}\left[\int_{-\alpha_m}^{\alpha_m} q(\alpha,\beta)h\left(\alpha'-\alpha\right)d\alpha\right]d\beta$$

$$h(\alpha)=\frac{1}{2}\left(\frac{\alpha}{\sin\alpha}\right)^2 s(\alpha),\quad q(\alpha,\beta)=p(\alpha,\beta)D\cos\alpha$$

2. IR: Image Reconstruction from Projection (31)

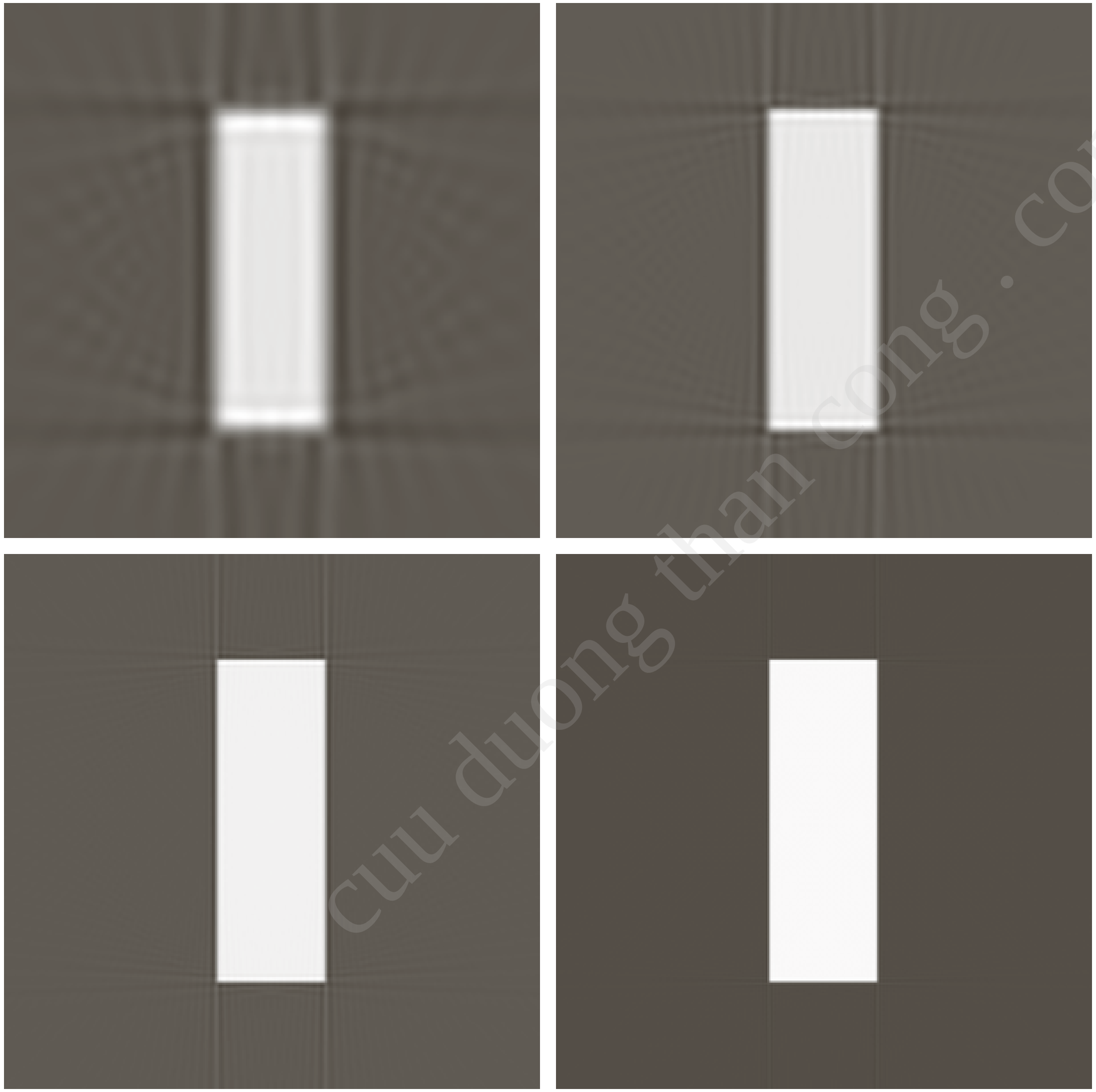
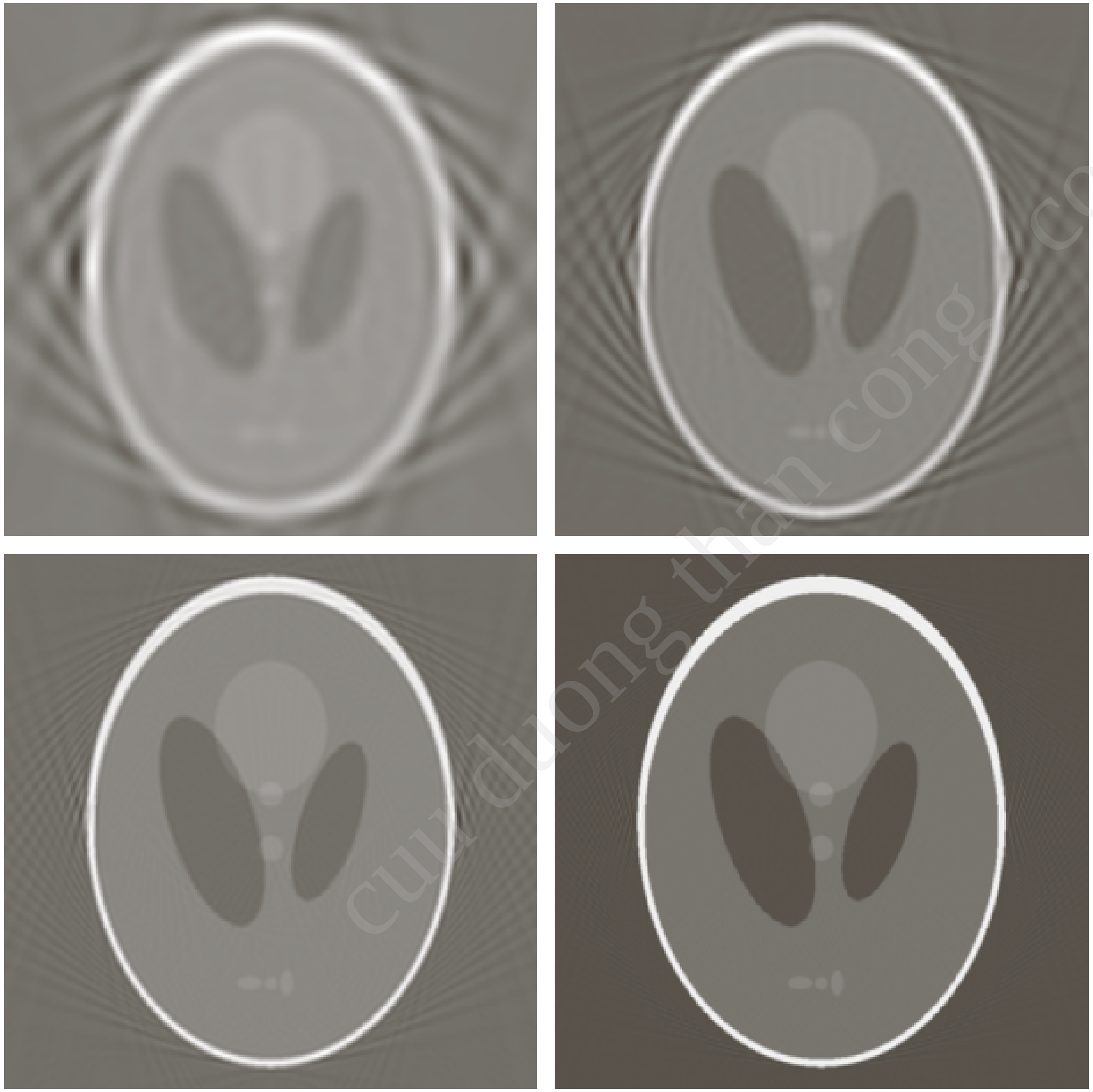


FIGURE 5.48 Reconstruction of the rectangle image from filtered fan backprojections. (a) 1° increments of α and β. (b) 0.5° increments. (c) 0.25° increments. (d) 0.125° increments. Compare (d) with Fig. 5.43(b).

2. IR: Image Reconstruction from Projection (32)



a b
c d

FIGURE 5.49 Reconstruction of the head phantom image from filtered fan backprojections. (a) 1° increments of α and β. (b) 0.5° increments. (c) 0.25° increments. (d) 0.125° increments. Compare (d) with Fig. 5.44(b).