Internal Assessment Test II – Dec. 2021

Sub:	Digital Image Processing			Sub Code:	17EC72/15 EC72	Branch:	ECE
Date:	17-12-2021	Duration:	90 Minutes	Max Marks:	50	Sem / Sec:	7E
							OBE

	Answer any FIVE FULL questions	MARKS	CO	RBT
1	Explain image smoothing in frequency domain using ideal low pass filter, Butterworth low pass filter and Gaussian low pass filtering	[10]	CO3	L1,L2
2	The following table gives the number of pixels at each of the gray levels 0 to 7 in an image: (table below) Draw the corresponding histogram. Perform histogram equalization and draw the resulting histogram	[10]	CO3	L2
3 (a)	Compute median value of the marked pixel using 3X3 mask (table below)	[04]	CO3	L2
3 (b)	Explain the basic steps for filtering in frequency domain.	[06]	CO3	L1
4	Explain homomorphic filters for image enhancement with necessary equations, block diagram and transfer function.	[10]	CO3	L1
5 (a)	Define 2D DFT with respect to a 2D DFT of an image, and state the following properties a) Translation, b) Rotation, c) Periodicity, and d) Convolution theorem	[06]	CO3	L1
5 (b)	Explain Gradient filtering in images.	[04]	CO3	L1
6	Histogram of a 64X64 image is given below: (table below) It is desired to transform this histogram to a new histogram given below: (table below)	[10]	CO3	L2

Question 2:

r_k	0	1	2	3	4	5	6	7
n_k	123	78	281	417	639	1054	816	688

Question 3 (a):

20	18	33	12	120	122
64	㉒	0	⑩	(100)	90
6	3	⑨	11	110	0
10	23	22	12	1	1

Question 6:

r_k	0	1	2	3	4	5	6	7
n_k	81	122	245	329	656	850	1023	790

z_k	0	1	2	3	4	5	6	7
p(z_k)	0.15	0.20	0.30	0.20	0.15	0	0	0

Sub:	Digital Image Processing					Sub Code:	17EC72/15EC72	Branch:
Date:	17-12-2021	Duration:	90 Minutes	Max Marks:	50	Sem / Sec:	7E	

Answer any FIVE FULL questions

MARKS

1 Explain image smoothing in frequency domain using ideal low pass filter, Butterworth low pass filter and Gaussian low pass filtering [10]

The edges and other sharp transitions (such as noise) in the gray levels of an image contribute significantly to the high-frequency content of its Fourier transform. Hence blurring (smoothing)is achieved in the frequency domain by attenuating high frequencies in the transform of a given image.

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

where F (u, v) is the Fourier transform of an image to be smoothed. The problem is to select a filter transfer function H (u, v) that yields G (u, v) by attenuating the high-frequency components of F (u, v). The inverse transform then will yield the desired smoothed image g (x, y).

Ideal Filter:

A 2-D ideal lowpass filter (ILPF) is one whose transfer function satisfies the relation

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

where D is a specified nonnegative quantity, and D(u, v) is the distance from point (u, v) to the origin of the frequency plane; that is,

$$D(u, v) = (u^2 + v^2)^{1/2}.$$

Figure 1(a) shows a 3-D perspective plot of H (u, v) as a function of u and v. The name ideal filter indicates that all frequencies inside a circle of radius Do are passed with no attenuation, whereas all frequencies outside this circle are completely attenuated.

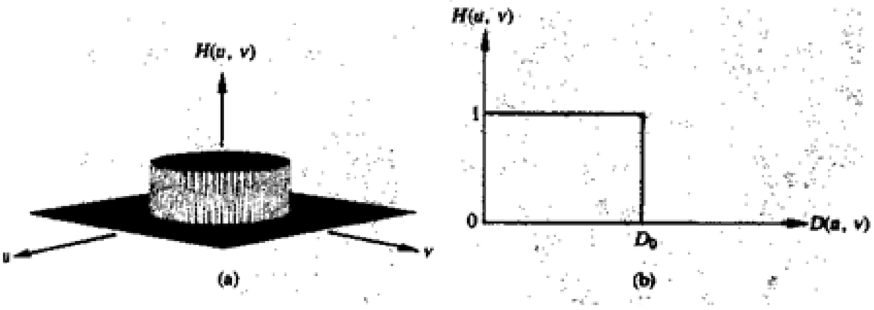


Fig. 1a) Perspective plot of an ideal lowpass filter transfer function; (b) filter crosssection.

The lowpass filters are radially symmetric about the origin. For this type of filter, specifying a cross section extending as a function of distance from the origin along a radial line is sufficient, as Fig. 1 (b) shows. The complete filter transfer function can then be generated by rotating the cross section 360 about the origin. Specification of radially symmetric filters centered on the N x N frequency square is based on the assumption that the origin of the Fourier transform has been centered on the square.

For an ideal lowpass filter cross section, the point of transition between H(u, v) =1 and H(u, v) = 0 is often called the cutoff frequency. In the case of Fig.1 (b), for example, the cutoff frequency is Do. As the cross section is rotated about the origin, the point Do traces a circle giving a locus of cutoff frequencies, all of which are a distance Do from the origin.

Butterworth low pass filter

The transfer function of the Butterworth lowpass (BLPF) of order n and with cutoff frequency locus at a distance Do, from the origin is defined by the relation

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

A perspective plot and cross section of the BLPF function are shown in figure 2.

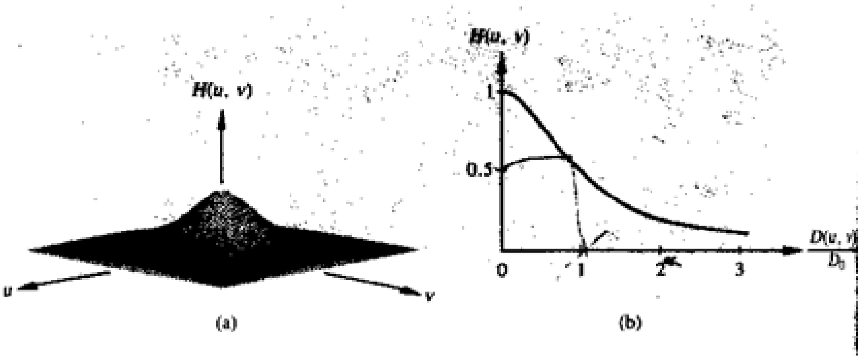


Fig.2 (a) A Butterworth lowpass filter (b) radial cross section for n = 1.

Unlike the ILPF, the BLPF transfer function does not have a sharp discontinuity that establishes a clear cutoff between passed and filtered frequencies. For filters with smooth transfer functions, defining a cutoff frequency locus at points for which H (u, v) is down to a certain fraction of its maximum value is customary. In the case of above Eq. H (u, v) = 0.5 (down 50 percent from its maximum value of 1) when D (u, v) = Do. Another value commonly used is 1/√2 of the maximum value of H (u, v). The following simple modification yields the desired value when D(u, v) = Do:

$$H(u, v) = \frac{1}{1 + [\sqrt{2} - 1][D(u, v)/D_0]^{2n}} = \frac{1}{1 + 0.414[D(u, v)/D_0]^{2n}}$$

It also serves as a common base for comparing the behavior of different types of filters.

The sharp cutoff frequencies of an ideal lowpass filter cannot be realized with electronic components, although they can certainly be simulated in a computer.

Gaussian Lowpass Filters:

The form of these filters in two dimensions is given by

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

where, D(u, v) is the distance from the origin of the Fourier transform.

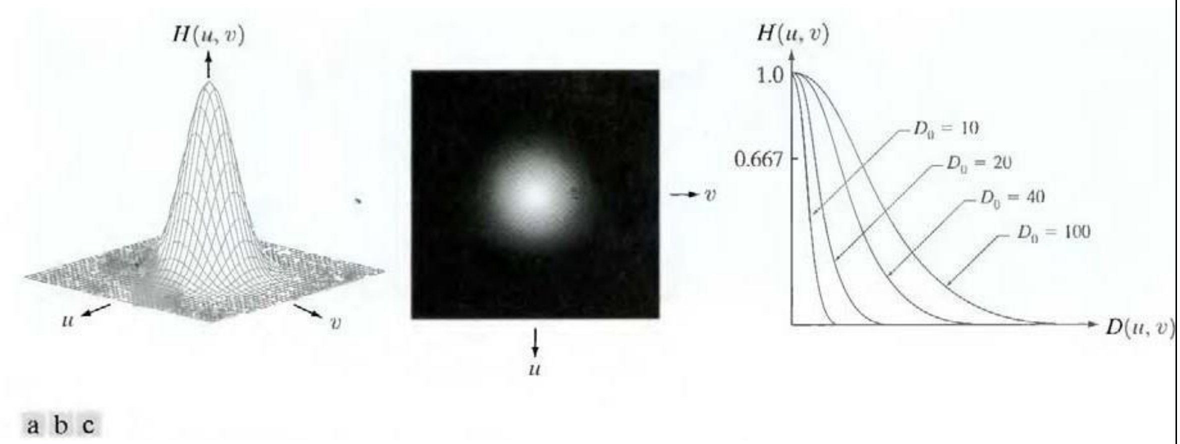


Fig.3 (a) Perspective plot of a GLPF transfer function, (b) Filter displayed as an image, (c) Filter radial cross sections for various values of Do.

σ is a measure of the spread of the Gaussian curve. By letting σ = Du, we can express the filter ina more familiar form in terms of the notation:

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

where Do is the cutoff frequency. When D (u, v) = Do, the filter is down to 0.607 of itsmaximum value.

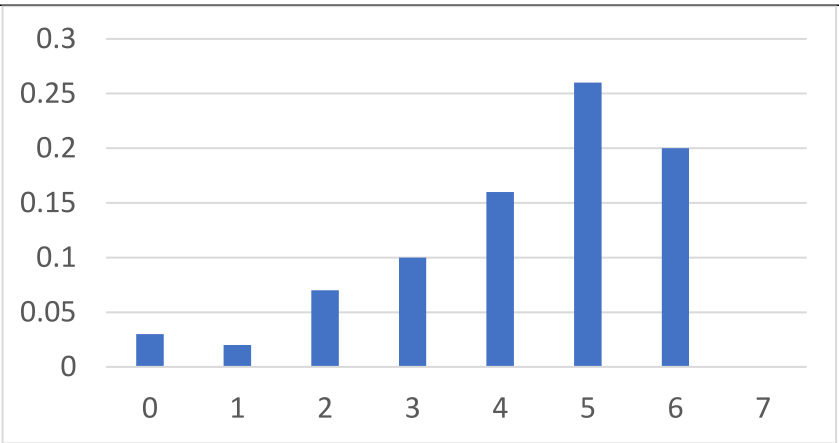
2 The following table gives the number of pixels at each of the gray levels 0 to 7 in an image: [10]

r_k	0	1	2	3	4	5	6	7
n_k	123	78	281	417	639	1054	816	688

Draw the corresponding histogram. Perform histogram equalization and draw the resulting histogram

Total no. of pixels: sum of all n_k=$\sum_{k=0}^{7} n_k$=(123+78+281+417+639+1054+816+688)= 4096

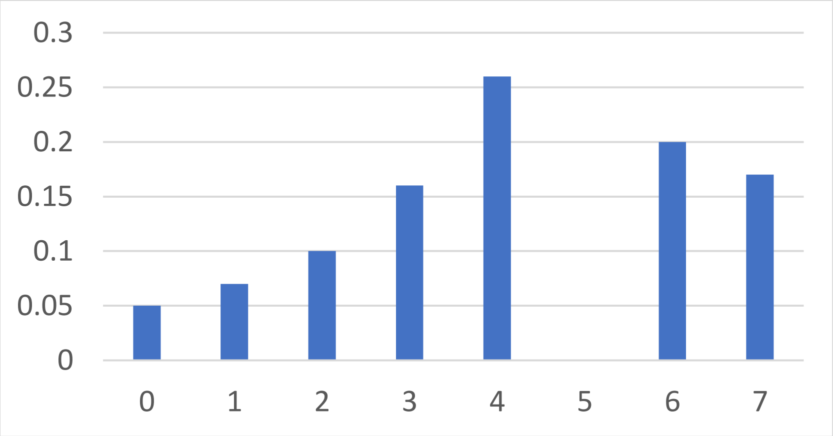
Normalized histogram



r_k	n_k	$p_r(r_k)$	cdf	Cdf*(L-1)	s_k
0	123	123/4096=0.03	0.03	0.03*7=0.21	0
1	78	78/4096=0.02	0.05	0.05*7=0.35	0
2	281	281/4096=0.07	0.12	0.12*7=0.84	1
3	417	417/4096=0.10	0.22	0.22*7=1.54	2
4	639	639/4096=0.16	0.38	0.38*7=2.66	3
5	1054	1054/4096=0.26	0.64	0.64*7=4.48	4
6	816	816/4096=0.20	0.84	0.84*7=5.88	6
7	688	688/4096=0.17	1	1*7=7	7

s_k	$p_s(s_k)$
0	0.03+0.02=0.05
1	0.07
2	0.10
3	0.16
4	0.26
5	0
6	0.20
7	0.17

Resulting histogram:



3 (a) Compute median value of the marked pixel using 3X3 mask [04]

20	18	33	12	120	122
64	22	0	10	100	90
6	3	9	11	110	0

10	23	22	12	1	1

Using a 3X3 mask for pixel (2,2):

20	18	33	12	120	122
64	22	0	10	100	90
6	3	9	11	110	0
10	23	22	12	1	1

Ordering the pixel values in increasing order:

[0
3
6
9
18
20
22
33
64]

Where 18 is the median value

Using a 3X3 mask for pixel (4,2):

20	18	33	12	120	122
64	22	0	10	100	90
6	3	9	11	110	0
10	23	22	12	1	1

Ordering the pixel values in increasing order:

[0
9
10
11
12
33
100
110
120]

Where 12 is the median value

Using a 3X3 mask for pixel (5,2):

20	18	33	12	120	122
64	22	0	10	100	90
6	3	9	11	110	0
10	23	22	12	1	1

Ordering the pixel values in increasing order:

[0
10
11
12
90
100
110
120
122]

Where 90 is the median value

Using a 3X3 mask for pixel (3,3):

20	18	33	12	120	122
64	22	0	10	100	90
6	3	9	11	110	0
10	23	22	12	1	1

Ordering the pixel values in increasing order:

[0
3
9
10
11
12
22
22
23]

Where 11 is the median value

3 (b) Explain the basic steps for filtering in frequency domain. [06]

1. Given an input image $f(x,y)$ of size MXN, obtain the padding parameters P and Q using the following equations:

$$P \geq M + C - 1 \; and \quad Q \geq N + D - 1$$

where the mask used for filtering is of the size CXD.

Typically, we select P=2Mand Q=2N.

2. Form a padded image $f_p(x,y)$ of size PXQ by appending the necessary number of zeros to $f(x,y)$

3. Multiply $f_p(x,y)$ by $(-1)^{x+y}$ to center its transform.

4. Compute the DFT, F(u,v), of the image from step 3.

5. Generate a real, symmetric filter function H(u,v) of size PXQ with senter at co-ordinates (P/2,Q/2). Form the product:

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

Using array multiplication, i.e. G(i,k)=F(i,k)H(i,k)

6. Obtain the processed image:

$$g_p(x,y) = \{real[\Im^{-1}[G(u,v)]]\}(-1)^{x+y}$$

Where the real part is selected in order to ignore parasitic complex components resulting from the computational inaccuracies and the subscript p indicates that we are dealing with padded arrays.

7. Obtain the final processes result $g(x,y)$, by extracting the MXN region from the top, left quadrant of $g_p(x,y)$

4 Explain homomorphic filters for image enhancement with necessary equations, block diagram and transfer function. [10]

An image f(x,y) can be expressed as the product of its illumination i(x,y) and reflectance r(x,y), components:

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

This equation cannot be used directly to operate on the frequency components of illumination and reflectance because the Fourier transform of a product is not the product of the transforms:

$$\Im[f(x,y)] \neq \Im[i(x,y)]\Im[r(x,y)]$$

However, we can define:

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

Then,

$$\Im[z(x,y)] = \Im[\ln\,(f(x,y))]$$

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

Or

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

Where $F_i(u,v)$ and $F_r(u,v)$ are the Fourier transform of $\ln\{i(x,y)\}\, and \ln\{r(x,y)\}$, respectively.

We can filter Z(u,v) using a filter H(u,v), such that:

$$S(u,v) = H(u,v)Z9u,v) = H(u,v)F_i(u,v) + H(u,v)F_r(u,v)$$

The filtered image in the spatial domain is

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

By defining

$i'(x,y) = \Im^{-1}\{H(u,v)F_i(u,v)\}$, and

$r'(x,y) = \Im^{-1}\{H(u,v)F_r(u,v)\}$,

Therefore

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

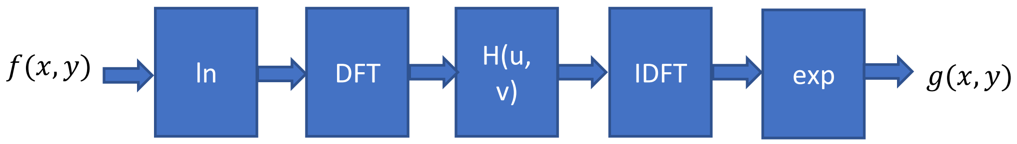
Finally because z(x,y) was formed by taking natural logarithm of the input image, we reverse the process by taking the exponential of the filtered result to form the output image:

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

Where,

$$i_0(x,y) = e^{i'(x,y)}\; and\; r_0(x,y) = e^{r'(x,y)}$$

Are the illumination and reflectance components of the output (processed) image.



The filter H(u,v) is called the homomorphic filter. The key approach is the separation of the illumination and reflectance components. The illumination component of an image generally is characterized by the slow spatial variation

While the reflectance component tends to vary abruptly particularly at the junctions of dissimilar objects. These characteristics lead to associating the low frequencies of the Fourier transform of the logarithmic of an image with illumination and high frequencies with reflectance. Better control can be gained over the illumination and reflectance components with a homomorphic filter. This control requires specification of a filter function H(u,v) that affects the low- and high- frequency components of the Fourier transform in different, controllable ways

5 (a) Define 2D DFT with respect to a 2D DFT of an image, and state the following properties b) Translation, b) Rotation, c) Periodicity, and d) Convolution theorem [06]

For an M × N 2D image f(x,y), 2D DFT is defined as follows:

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

Where u=0,1,2,…,M-1, and v=0,1,2,….,N-1

Inverse 2D-DFT :

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

a) Translation in frequency domain:

FT{ $f(x,y)e^{j2\pi\left(\frac{u_0x}{M}+\frac{v_0y}{N}\right)}$} = $F(u-u_0,v-v_0)$

Translation in time domain:

FT{f(x $-\, x_0, y-y_0$)}= $F(u,v)e^{-j2\pi\left(\frac{ux_0}{M}+\frac{y_0v}{N}\right)}$

b) Rotation:

FT{$f(r,\Theta+\Theta_0)$}=$F(\omega,\Phi+\Theta_0)$

Where, $x=rcos\Theta,\ y=rsin\Theta, u=\omega cos\ \Phi,\ v=\omega sin\ \Phi$

c) Periodicity:

$f(x,y)=f(x+k_1M,y)=f(x,y+k_2N)=f(x+k_1M,y+k_2N)$

F$(u,v)=F(u+k_1M,v)=F(u,v+k_2N)=F(u+k_1M,v+k_2N)$

d) Convolution Theorem

FT{$f(x,y)*h(x,y)$}= $F(u,v)H(u,v)$

5 (b) Explain Gradient filtering in images. [04]

First-order derivatives of a digital image are based on various approximations of the 2-Dgradient. The gradient of an image f (x, y) at location (x, y) is defined as the vector

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

It is well known from vector analysis that the gradient vector points in the direction of maximumrate of change of f at coordinates (x, y). An important quantity in edge detection is the magnitude

of this vector, denoted by Af, where

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

This quantity gives the maximum rate of increase of f (x, y) per unit distance in the direction of Af. It is a common (although not strictly correct) practice to refer to Af also as the gradient. The direction of the gradient vector also is an important quantity. Let α (x, y) represent the direction angle of the vector Af at (x, y). Then, from vector analysis,

$$\alpha(x,y)=\tan^{-1}\left(\frac{G_y}{G_x}\right)$$

where the angle is measured with respect to the x-axis. The direction of an edge at (x, y) is perpendicular to the direction of the gradient vector at that point. Computation of the gradient of an image is based on obtaining the partial derivatives &f/&x and &f/&y at every pixel location.Let the 3x3 area shown in Fig. 1.1 (a) represent the gray levels in a neighborhood of an image. One of the simplest ways to implement a first-order partial derivative at point z_5 is to use the

following Roberts cross-gradient operators:

$$G_x = (z_9 - z_5)$$

and

$$G_y = (z_8 - z_6).$$

These derivatives can be implemented for an entire image by using the masks shown in Fig. 1.1(b). Masks of size 2 X 2 are awkward to implement because they do not have a clear center.An approach using masks of size 3 X 3 is given by

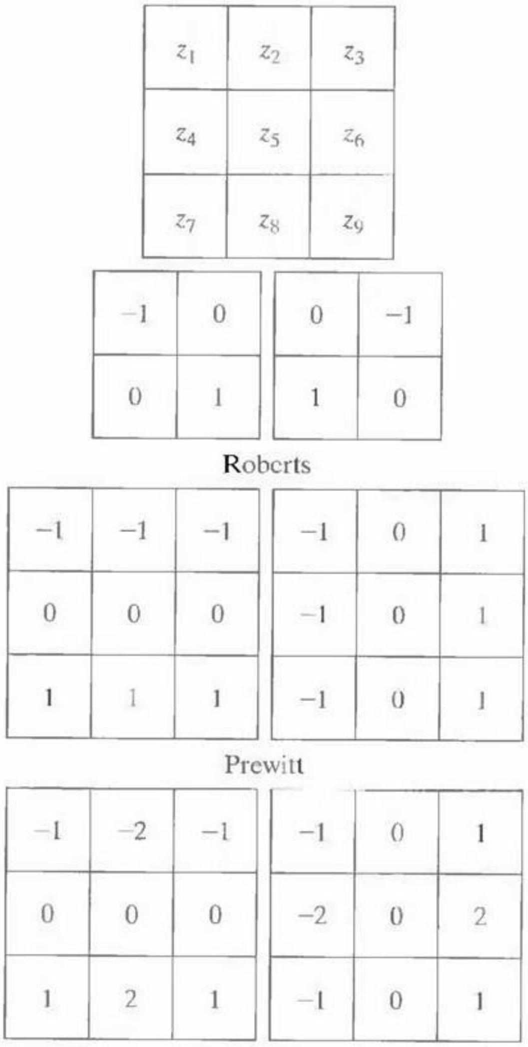
$$G_x = (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)$$

and

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

A weight value of 2 is used to achieve some smoothing by giving more importance to the center point. Figures 1.1(f) and (g), called the Sobel operators, and are used to implement these two equations. The Prewitt and Sobel operators are among the most used in practice for computing digital gradients. The Prewitt masks are simpler to implement than the Sobel masks, but the latter have slightly superior noise-suppression characteristics, an important issue when dealing with derivatives. Note that the coefficients in all the masks shown in Fig. 1.1 sum to 0, indicating that they give a response of 0 in areas of constant gray level, as expected of a derivative operator.

The masks just discussed are used to obtain the gradient components G_x and G_y. Computation of the gradient requires that these two components be combined. However, this implementation is not always desirable because of the computational burden required by squares and square roots. An approach used frequently is to approximate the gradient by



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

Roberts

−1	−1	−1
0	0	0
1	1	1

−1	0	1
−1	0	1
−1	0	1

Prewitt

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

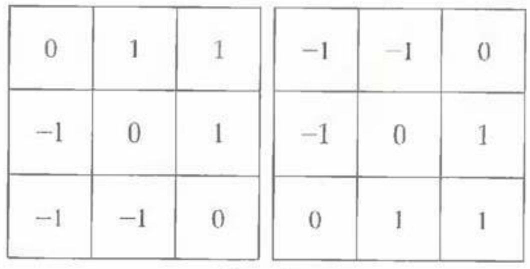
Sobel

absolute values:

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

This equation is much more attractive computationally, and it still preserves relative changes in gray levels. However, this is not an issue when masks such as the Prewitt and Sobel masks are used to compute Gx and Gy.

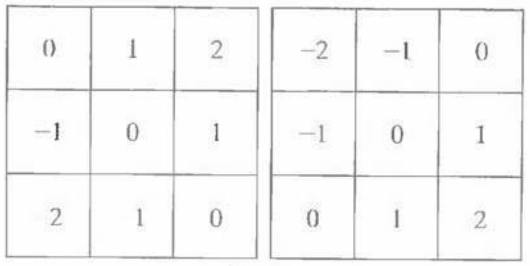
It is possible to modify the 3 X 3 masks in Fig. 1.1 so that they have their strongest responses along the diagonal directions. The two additional Prewitt and Sobel masks for detecting discontinuities in the diagonal directions are shown in Fig. 1.2.



0	1	1
−1	0	1
−1	−1	0

−1	−1	0
−1	0	1
0	1	1

Prewitt



0	1	2
−1	0	1
−2	−1	0

−2	−1	0
−1	0	1
0	1	2

Sobel

6 Histogram of a 64X64 image is given below: [10]

r_k	0	1	2	3	4	5	6	7
n_k	81	122	245	329	656	850	1023	790

It is desired to transform this histogram to a new histogram given below:

z_k	0	1	2	3	4	5	6	7
p(z_k)	0.15	0.20	0.30	0.20	0.15	0	0	0

r_k	n_k	$p_r(r_k)$	cdf	Cdf*(L-1)	s_k
0	81	81/4096=0.02	0.02	0.02*7=0.14	0
1	122	122/4096=0.03	0.05	0.05*7=0.35	0
2	245	245/4096=0.06	0.11	0.11*7=0.77	1
3	329	329/4096=0.08	0.19	0.19*7=1.33	1
4	656	656/4096=0.16	0.35	0.35*7=2.45	2
5	850	850/4096=0.21	0.56	0.56*7=3.92	4
6	1023	1023/4096=0.25	0.81	0.81*7=5.67	6
7	790	790/4096=0.19	1	1*7=7	7

z_k	$p_r(r_k)$	cdf	Cdf*(L-1)	s_k
0	0.15	0.15	0.15*7=1.05	1

1	0.20	0.35	0.35*7=2.45	2
2	0.30	0.65	0.65*7=4.55	5
3	0.20	0.85	0.85*7=5.95	6
4	0.15	1	1*7=7	7
5	0	1	1*7=7	7
6	**0**	1	1*7=7	7
7	**0**	1	1*7=7	7

r_k	s_k	z_k
0	0	0
1	0	0
2	1	0
3	1	0
4	2	1
5	4	2
6	6	3
7	7	4