USN | | | | | | | | | | |

Internal Assessment Test 2 – November 2024

Sub: **DIGITAL IMAGE PROCESSING**		Sub Code: **21CS732**		Branch: **AIML**	
Date: **20.11.24**	Duration: **90 min**	Max Marks: **50**	Sem/Sec: **VII - A**		**OBE**
Answer any FIVE FULL Questions			**MARKS**	**CO**	**RBT**
1	Explain the implementation of first order derivatives for image sharpening - The gradient.		10	CO2	L2

Ans:

Using first-order derivatives for (Non-linear) Image sharpening - The Gradient

* 1st derivatives in image processing are implemented using the magnitude of the gradient

* for a function $f(x,y)$, the gradient of "f" at co-ordinates (x,y) is defined as the 2-dimensional column vector

$$\nabla f \equiv grad(f) \equiv \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \rightarrow (10)$$

* The vector has important geometrical property that it points in the direction of the greatest rate of change of 'f' at location (x,y)

* magnitude (length) of vector ∇f, denoted as $M(x,y)$, where

$$\boxed{M(x,y) = mag(\nabla f) = \sqrt{g_x^2 + g_y^2}} \rightarrow (11)$$

is the value at (x,y) of the rate of change in the direction of the gradient vector.

$$M(x,y) \approx |g_x| + |g_y| \rightarrow (12)$$

8

partial derivatives of eq (16) are not rotation invariant (isotropic) but the magnitude of the gradient vector is

* we define discrete approximation to the preceding eqn & from there formulate the appropriate filter mask

* 3x3 region of image (z s are intensity value]

fig (a)

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

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

(1) Roberts cross-gradient operators

-1	0
0	1

0	-1
1	0

fig (b)

$g_x = (z_8 - z_5)$ & $g_y (z_6 - z_5)$

* 2 other defns proposed by Roberts in the early development of digital image processing use cross differences

Scanned with CamScanner

$g_x = (z_9 - z_5)$ & $g_y = (z_8 - z_6) \rightarrow (13)$

use eq 11 & 13 we can compute gradient image as

$M(x,y) = \sqrt{g_x^2 + g_y^2}$

$$M(x,y) = \left[(z_9 - z_5)^2 + (z_8 - z_6)^2 \right]^{1/2} \rightarrow (14)$$

If we use eq (12) & (13)

$M(x,y) \approx |g_x| + |g_y|$

$$M(x,y) \approx |z_9 - z_5| + |z_8 - z_6| \rightarrow (15)$$

* The partial derivatives terms in eq (13) ~~it~~ can be implemented using 2 linear filter as shown in fig (6)

* These masks are referred as Roberts-cross-gradient operator

(ii) Sobel operator

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

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

fig (7)

(9)

* masks of even sizes don't have a center of symmetry
* The smallest filter mask is 3 X 3

$$g_x = \frac{\partial f}{\partial x} = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3) \rightarrow (16)$$

&

$$g_y = \frac{\partial f}{\partial y} = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7) \quad (17)$$

* These eqns can be implemented using masks of fig (c).
* Substituting g_x & g_y in (12)

$$M(x,y) \approx \lvert (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3) \rvert + \lvert (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7) \rvert \quad (18)$$

* The masks are called Sobel operators

125

2	Explain Spatial Correlation and convolution. Perform correlation and convolution on the following function: **f = 0 0 0 1 1 0 1** **w = 8 2 4** (Add the padding required) Ans:	10	CO2	L3

0 0 0 0 | 0 0 0 1 0 0 0 0 | 0 0 0 0   o/p

1 2 3 2 8 = 0

(d) → 1 2 3 2 8 = 0

after 1 shift 1 2 3 2 8 → 0

1 2 3 2 8 → 8

(e) after 4 shift → 1 2 3 2 8 → 2

1 2 3 2 8 → 3

1 2 3 2 8 → 2

1 2 3 2 8 → 1

1 2 3 2 8 → 0

1 2 3 2 8 → 0

1 2 3 2 8 → 0

(f) final position → 1 2 3 2 8 → 0

cropped correlation result

(g) Full correlation result

0 0 0 8 2 3 2 1 0 0 0 0

(h) cropped correlation result (the size should be same as f)

0 8 2 3 2 1 0 0

Convolution

(a) f: 0 0 0 1 0 0 0 0   w rotated 180°: 8 2 3 2 1

(b) 0 0 0 1 0 0 0 0
8 2 3 2 1

(c) 0 0 0 0 | 0 0 0 1 0 0 0 0 | 0 0 0 0   zero padding
8 3 2 1

(g) Full convolution result
0 0 0 1 2 3 2 8 0 0 0 0

(e) 0 1 2 3 2 8 0 0 → cropped convolution result

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

8 2 3 2 1 — 0
8 2 3 2 1 — 0
8 2 3 2 1 — 0
8 2 3 2 1 — 1
8 2 3 2 1 — 2
8 2 3 2 1 — 3 Cropped
8 2 3 2 1 — 2
8 2 3 2 1 — 8
8 2 3 2 1 — 0
8 2 3 2 1 — 0
8 2 3 2 1 — 0
8 2 3 2 1 — 0
8 2 3 2 1 — 0

* Two important points to note from the above discussion.

(1) * correlation is a function of displacement of the filter.

* 1st value of correlation corresponds to zero displacement of the filter
* 2nd corresponds to one unit displacement & so on . .

(2) The correlating a filter 'w' with a function that contains all 0's & a single '1' yields a result that is copy of 'w' but roated by 180°

Linear spatial filtering

(1) Correlation (2) convolution

(1) Correlation :- is the process of moving a filter mask over the image & computing the sum of products at each location. [as explained in Linear spatial filtering]

(2) convolution :- the mechanism is same except the filter is first rotated by 180°

* Let us explain the above concept using 1-D illustration.

$\boxed{\text{correlation}}$

	origin f	w	
(a)	0 0 0 1 0 0 0 0	1 2 3 2 8	Length of image = 8 length of filter (size) = m = 5

(b) 0 0 0 1 0 0 0 0
1 2 3 2 8 ← starting position alignment

zero padding

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

(m-1) 0's are padded on ~~other~~ either side of 'f'

<table>
<tr><td></td><td></td><td>zero padding
f - 0 0 0 0 1 1 0 1 0
w - 8 2 4
8 2 4
8 2 4
8 2 4
8 2 4
8 2 4
8 2 4
· Result:
0 0 0 4 6 10 12 2 0
(Note: Reverse result for Convolution)</td><td></td><td></td><td></td></tr>
<tr><td>3</td><td>a</td><td>What is Unsharp masking and Highboost filtering?

Ans:</td><td>6</td><td>CO2</td><td>L1</td></tr>
</table>

* $\bar{f}(x,y) \Rightarrow$ denote blurred image

* unsharp masking is expressed in eq^n form as follows

$$\boxed{g_{mask}(x,y) = f(x,y) - \bar{f}(x,y)} \rightarrow (8)$$

* Then we add a weighted portion of the mask back to the original image

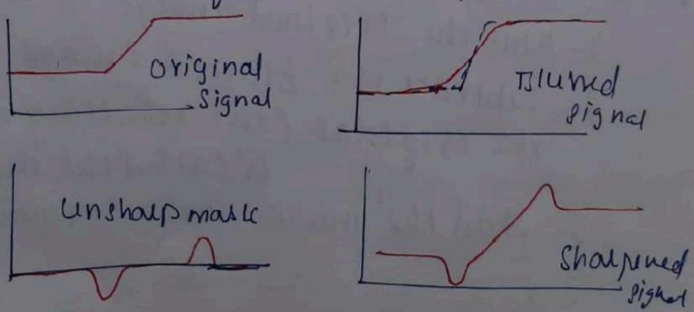
$$\boxed{g(x,y) = f(x,y) + K * g_{mask}(x,y)} \rightarrow (9)$$

$K \geq 0$ for generality

$K = 1$ we have unsharp masking

$K > 1$, process is referred as highboost filtering

$K < 1$, de-emphasizes the contribution of the un-sharp mask



* Laplacian for image sharpening

$$\boxed{g(x,y) = f(x,y) + c\left[\nabla^2 f(x,y)\right]}$$

$f(x,y) \rightarrow$ i/p image

$g(x,y) \rightarrow$ sharpened image

$c = -1$ = constant (sub)

1 if other filter are used (add)

Unsharp masking & Highboost filtering

* A process that has been used by the printing & publishing industry for many years is to sharpen images consists of subtracting an unsharp (smoothed) version of an image from the original image
* This process is called unsharp masking
* & consists of foll steps
  1. Blur the original image
  2. subtract the blurred image from the original (the resulting differenc is called the mask).
  3. Add the mask to the original

(2)

b	Explain the mechanics of Linear Spatial Filtering? Ans:	4	CO2	L2

THE MECHANICS OF LINEAR SPATIAL FILTERING

A linear spatial filter performs a sum-of-products operation between an image f and a *filter kernel*, w. The kernel is an array whose size defines the neighborhood of operation, and whose coefficients determine the nature of the filter. Other terms used to refer to a spatial filter kernel are *mask*, *template*, and *window*. We use the term *filter kernel* or simply *kernel*.

Figure 3.28 illustrates the mechanics of linear spatial filtering using a 3×3 kernel. At any point (x, y) in the image, the response, $g(x, y)$, of the filter is the sum of products of the kernel coefficients and the image pixels encompassed by the kernel:

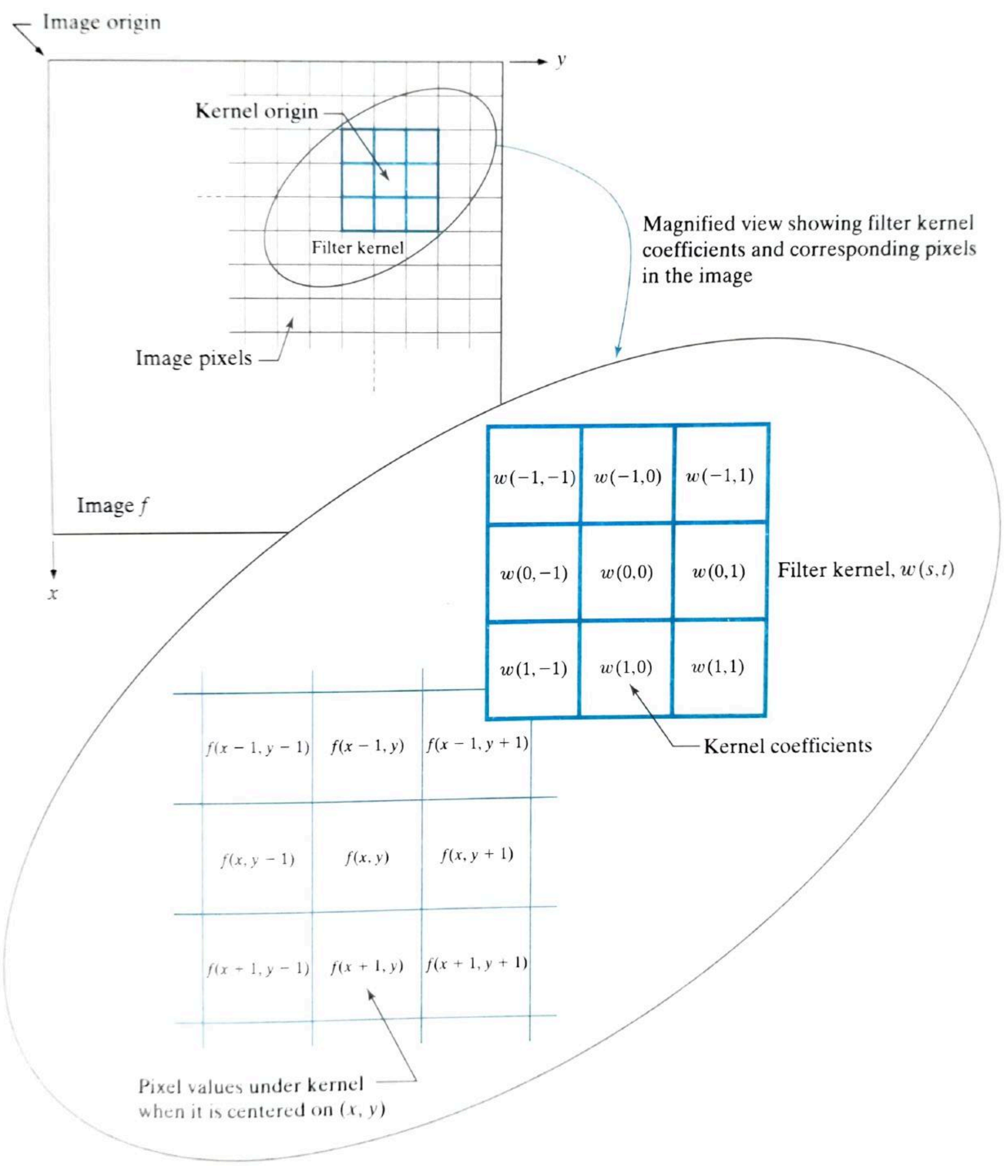
$$\begin{aligned} g(x,y) = {} & w(-1,-1)f(x-1,y-1) + w(-1,0)f(x-1,y) + \dots \\ & + w(0,0)f(x,y) + \dots + w(1,1)f(x+1,y+1) \end{aligned} \tag{3-30}$$

As coordinates x and y are varied, the center of the kernel moves from pixel to pixel, generating the filtered image, g, in the process.[†]

Observe that the center coefficient of the kernel, $w(0,0)$, aligns with the pixel at location (x, y). For a kernel of size $m \times n$, we assume that $m = 2a + 1$ and $n = 2b + 1$, where a and b are nonnegative integers. This means that our focus is on kernels of odd size in both coordinate directions. In general, linear spatial filtering of an image of size $M \times N$ with a kernel of size $m \times n$ is given by the expression

$$g(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t) \tag{3-31}$$

where x and y are varied so that the center (origin) of the kernel visits every pixel in f once. For a fixed value of (x, y), Eq. (3-31) implements the *sum of products* of the form shown in Eq. (3-30), but for a kernel of arbitrary odd size. As you will learn in the following section, this equation is a central tool in linear filtering.



4	a	Briefly explain the model of Image degradation and restoration with a neat diagram. Ans:	4	CO3	L2

5.1 A MODEL OF THE IMAGE DEGRADATION/RESTORATION PROCESS

In this chapter, we model image degradation as an operator $\mathcal{H}$ that, together with an additive noise term, operates on an input image $f(x, y)$ to produce a degraded image $g(x, y)$ (see Fig. 5.1). Given $g(x, y)$, some knowledge about $\mathcal{H}$, and some knowledge about the additive noise term $\eta(x, y)$, the objective of restoration is to obtain an estimate $\hat{f}(x, y)$ of the original image. We want the estimate to be as close as possible to the original image and, in general, the more we know about $\mathcal{H}$ and η, the closer $\hat{f}(x, y)$ will be to $f(x, y)$.

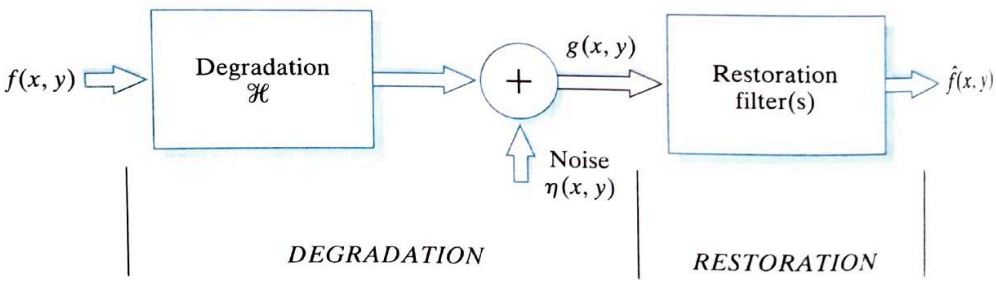
We will show in Section 5.5 that, if $\mathcal{H}$ is a linear, position-invariant operator, then the degraded image is given in the spatial domain by

$$g(x,y) = (h \star f)(x,y) + \eta(x,y) \tag{5-1}$$

where $h(x, y)$ is the spatial representation of the degradation function. As in Chapters 3 and 4, the symbol "$\star$" indicates convolution. It follows from the convolution theorem that the equivalent of Eq. (5-1) in the frequency domain is

$$G(u,v) = H(u,v)F(u,v) + N(u,v) \tag{5-2}$$

where the terms in capital letters are the Fourier transforms of the corresponding terms in Eq. (5-1). These two equations are the foundation for most of the restoration material in this chapter.



	Question	Marks	CO	Level
b	Briefly mention 3 important noise PDFs.	6	CO3	L1

Ans:

1. <u>Gaussian Noise</u>

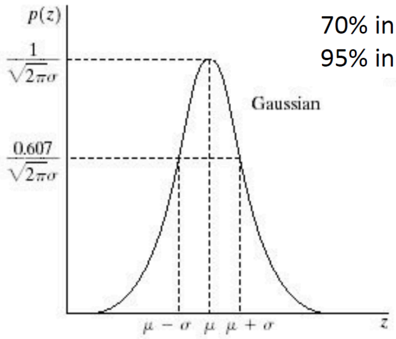
   The PDF of a Guassian random variable, z, is defined by the following familiar expression:

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

mean variance

Note: $\int_{-\infty}^{\infty} p(z)dz = 1$

Where z represents intensity, $\bar{z}$ is the mean value of z, and σ is its standard deviation.



70% in [(μ−σ), (μ+σ)]
95% in [(μ−2σ), (μ+2σ)]

2. <u>Rayleigh Noise</u>
   The PDF is given by

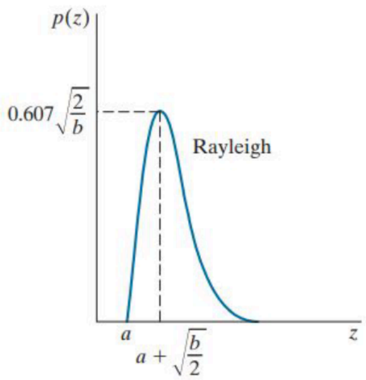
$$p(z) = \begin{cases} \frac{2}{b}(z - a)e^{-(z-a)^2/b} & z \geq a \\ 0 & z < a \end{cases}$$

The mean and variance of z when this random variable is characterized by a Rayleigh PDF are

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

and

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



$\bar{z}$

3. Gamma Noise
   The PDF of Erlang or Gamma noise is

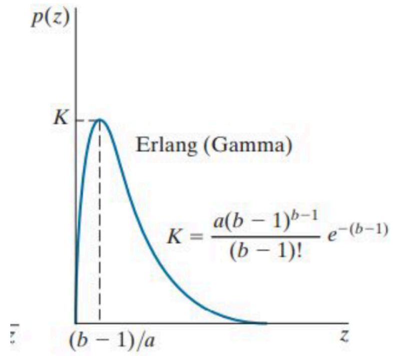
$$p(z) = \begin{cases} \frac{a^b z^{b-1}}{(b - 1)!}e^{-az} & z \geq 0 \\ 0 & z < 0 \end{cases}$$

where the parameters are such that a>b, b is a positive integer, and “!” indicates factorial. The mean and variance of z are

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

and

$$\sigma^2 = \frac{b}{a^2}$$

				
5	What are the ways to estimate the degradation function in image restoration?	10	CO3	L2

Ans:

ESTIMATION BY IMAGE OBSERVATION

Suppose that we are given a degraded image without any knowledge about the degradation function $\mathcal{H}$. Based on the assumption that the image was degraded by a linear, position-invariant process, one way to estimate $\mathcal{H}$ is to gather information from the image itself. For example, if the image is blurred, we can look at a small rectangular section of the image containing sample structures, like part of an object and the background. In order to reduce the effect of noise, we would look for an area in which the signal content is strong (e.g., an area of high contrast). The next step would be to process the subimage to arrive at a result that is as unblurred as possible.

Let the observed subimage be denoted by $g_s(x, y)$, and let the processed subimage (which in reality is our estimate of the original image in that area) be denoted by $\hat{f}_s(x, y)$. Then, assuming that the effect of noise is negligible because of our choice of a strong-signal area, it follows from Eq. (5-65) that

$$H_s(u,v) = \frac{G_s(u,v)}{\hat{F}_s(u,v)} \tag{5-66}$$

From the characteristics of this function, we then deduce the complete degradation function $H(u,v)$ based on our assumption of position invariance. For example, suppose that a radial plot of $H_s(u,v)$ has the approximate shape of a Gaussian curve. We can use that information to construct a function $H(u,v)$ on a larger scale, but having the same basic shape. We then use $H(u,v)$ in one of the restoration approaches to be discussed in the following sections. Clearly, this is a laborious process used only in very specific circumstances, such as restoring an old photograph of historical value.

ESTIMATION BY EXPERIMENTATION

If equipment similar to the equipment used to acquire the degraded image is available, it is possible in principle to obtain an accurate estimate of the degradation. Images similar to the degraded image can be acquired with various system settings until they are degraded as closely as possible to the image we wish to restore. Then the idea is to obtain the impulse response of the degradation by imaging an impulse (small dot of light) using the same system settings. As noted in Section 5.5, a linear, space-invariant system is characterized completely by its impulse response.

An impulse is simulated by a bright dot of light, as bright as possible to reduce the effect of noise to negligible values. Then, recalling that the Fourier transform of an impulse is a constant, it follows from Eq. (5-65) that

$$H(u,v) = \frac{G(u,v)}{A} \tag{5-67}$$

where, as before, $G(u,v)$ is the Fourier transform of the observed image, and A is a constant describing the strength of the impulse. Figure 5.24 shows an example.

ESTIMATION BY MODELING

Degradation modeling has been used for many years because of the insight it affords into the image restoration problem. In some cases, the model can even take into account environmental conditions that cause degradations. For example, a degradation model proposed by Hufnagel and Stanley [1964] is based on the physical characteristics of atmospheric turbulence. This model has a familiar form:

$$H(u,v) = e^{-k(u^2 + v^2)^{5/6}} \tag{5-68}$$

where k is a constant that depends on the nature of the turbulence. With the exception of the 5/6 power in the exponent, this equation has the same form as the Gaussian lowpass filter transfer function discussed in Section 4.8. In fact, the Gaussian LPF is used sometimes to model mild, uniform blurring. Figure 5.25 shows examples obtained by simulating blurring an image using Eq. (5-68) with values $k = 0.0025$

(severe turbulence), $k = 0.001$ (mild turbulence), and $k = 0.00025$ (low turbulence). We restore these images using various methods later in this chapter.

Another approach used frequently in modeling is to derive a mathematical model starting from basic principles. We illustrate this procedure by treating in some detail the case in which an image has been blurred by uniform linear motion between the image and the sensor during image acquisition. Suppose that an image $f(x,y)$ undergoes planar motion and that $x_0(t)$ and $y_0(t)$ are the time-varying components of motion in the x- and y-directions, respectively. We obtain the total exposure at any point of the recording medium (say, film or digital memory) by integrating the instantaneous exposure over the time interval during which the imaging system shutter is open.

6 | Explain in detail the Order Statistic Filters. | 10 | CO3 | L3

Perform any two Order statistic filtering on the following:

20	**50**	**30**
40	**40**	**20**
10	**10**	**20**

(Add the padding required)

	Ans: **ORDER-STATISTIC FILTERS** We introduced order-statistic filters in Section 3.6. We now expand the discussion in that section and introduce some additional order-statistic filters. As noted in Section 3.6, order-statistic filters are spatial filters whose response is based on ordering (ranking) the values of the pixels contained in the neighborhood encompassed by the filter. The ranking result determines the response of the filter. **Median Filter** The best-known order-statistic filter in image processing is the *median filter*, which, as its name implies, replaces the value of a pixel by the median of the intensity levels in a predefined neighborhood of that pixel: $$\hat{f}(x,y) = \underset{(r,c)\in S_{xy}}{\text{median}}\{g(r,c)\} \qquad (5\text{-}27)$$ where, as before, S_{xy} is a subimage (neighborhood) centered on point (x, y). The value of the pixel at (x, y) is included in the computation of the median. Median filters			

are quite popular because, for certain types of random noise, they provide excellent noise-reduction capabilities, with considerably less blurring than linear smoothing filters of similar size. Median filters are particularly effective in the presence of both bipolar and unipolar impulse noise, as Example 5.3 below shows. Computation of the median and implementation of this filter are discussed in Section 3.6.

Max and Min Filters

Although the median filter is by far the order-statistic filter most used in image processing, it is by no means the only one. The median represents the 50th percentile of a ranked set of numbers, but you will recall from basic statistics that ranking lends itself to many other possibilities. For example, using the 100th percentile results in the so-called *max filter*, given by

$$\hat{f}(x,y) = \max_{(r,c)\in S_{xy}} \{g(r,c)\} \tag{5-28}$$

This filter is useful for finding the brightest points in an image or for eroding dark regions adjacent to bright areas. Also, because pepper noise has very low values, it is reduced by this filter as a result of the max selection process in the subimage area S_{xy}.

The 0th percentile filter is the *min filter*:

$$\hat{f}(x,y) = \min_{(r,c)\in S_{xy}} \{g(r,c)\} \tag{5-29}$$

This filter is useful for finding the darkest points in an image or for eroding light regions adjacent to dark areas. Also, it reduces salt noise as a result of the min operation.

Midpoint Filter

The *midpoint filter* computes the midpoint between the maximum and minimum values in the area encompassed by the filter:

$$\hat{f}(x,y) = \frac{1}{2}\left[\max_{(r,c)\in S_{xy}} \{g(r,c)\} + \min_{(r,c)\in S_{xy}} \{g(r,c)\}\right] \tag{5-30}$$

Note that this filter combines order statistics and averaging. It works best for randomly distributed noise, like Gaussian or uniform noise.

Alpha-Trimmed Mean Filter

Suppose that we delete the $d/2$ lowest and the $d/2$ highest intensity values of $g(r,c)$ in the neighborhood S_{xy}. Let $g_R(r,c)$ represent the remaining $mn - d$ pixels in S_{xy}. A filter formed by averaging these remaining pixels is called an *alpha-trimmed mean filter*. The form of this filter is

$$\hat{f}(x,y) = \frac{1}{mn-d}\sum_{(r,c)\in S_{xy}} g_R(r,c) \tag{5-31}$$

where the value of d can range from 0 to $mn-1$. When $d=0$ the alpha-trimmed filter reduces to the arithmetic mean filter discussed earlier. If we choose $d = mn - 1$, the filter becomes a median filter. For other values of d, the alpha-trimmed filter is useful in situations involving multiple types of noise, such as a combination of salt-and-pepper and Gaussian noise.

20	50	30
40	40	20
10	10	20

After padding

Kernel (3×3)

0	0	0	0	0
0	20	50	30	0
0	40	40	20	0
0	10	10	20	0
0	0	0	0	0

Using Median Filter (3×3)

0	0	0	0	0
0	20	50	30	0
0	40	40	20	0
0	10	10	20	0
0	0	0	0	0

⇒

20	50	30
40	40	20
10	10	20

Using Max Filter

⇒

50	50	50
50	50	50
40	40	40

CI **CCI** **HOD**

---All the Best--