

Solution of VTU Examination Dec-2018

"Digital Image Processing" – 15EC72

CBCS SCHEME

USN 1 C R I S E C O 4 8

15EC72

Seventh Semester B.E. Degree Examination, Dec.2018/Jan.2019

Digital Image Processing

Time: 3 hrs.

Max. Marks: 80

Note: Answer FIVE full questions, choosing ONE full question from each module.

Module-1

- 1 a. What is digital image? Explain the fundamental steps of digital image processing. (08 Marks)
- b. Explain the concept of sampling and quantization of an image. (06 Marks)
- c. Mention any four fields that use digital image processing. (02 Marks)

OR

- 2 a. Explain with neat diagram, how image is acquired using sensor strips? (08 Marks)
- b. Define 4-, 8- and m-adjacency. Compute the lengths of the shortest 4-, 8- and m-path between p and q in the image segment shown in Fig. Q2 (b) by considering $V = \{2, 3, 4\}$ (06 Marks)

	3	4	1	2	0
	0	1	0	4	2
	2	2	3	1	4
(p)	3	0	4	2	1
	1	2	0	3	4

Fig. Q2 (b)

- c. A common measure of transmission for digital data is the baud rate defined as the number of bits transmitted per second. Generally, transmission is accomplished in packets consisting of a start bit, a byte (8 bits) of information and a stop bit. Using these facts find how many minutes would it take to transmit a 2048×2048 image with 256 intensity levels using a 33.6 K baud modem? (02 Marks)

Module-2

- 3 a. For a given 4×4 image having gray scales between $[0, 9]$ perform histogram equalization and draw the histogram of image before and after equalization. 4×4 image is shown in Fig. Q3 (a). (08 Marks)

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

Fig. Q3 (a)

- b. Explain smoothing of images in frequency domain using ideal, Butterworth and Gaussian Low pass filter. (08 Marks)

OR

- 4 a. Define 2D DFT- with respect to 2D DFT of an image and state the following properties: (i) Translation (ii) Rotation (iii) Periodicity (iv) Convolution theorem. (05 Marks)
- b. With necessary graphs, explain the log and power law transformation used for spatial image enhancement. (05 Marks)
- c. Explain image sharpening in spatial domain using second order Laplacian derivative. (06 Marks)

Module-3

- 5 a. With necessary equations and graph, explain any four noise probability density functions. (08 Marks)
 b. Explain minimum mean square error filtering method of restoring images. (08 Marks)

OR

- 6 a. Explain how image degradation is estimated using,
 (i) Observation (ii) Mathematical modeling. (08 Marks)
 b. Explain the order statistics filters used for restoring images in the presence of noise. (08 Marks)

Module-4

- 7 a. Write the equations for converting colors from HSI to RGB. (06 Marks)
 b. Write H matrix for Haar transform for $N = 4$ and explain how it is constructed. (04 Marks)
 c. Explain the following morphological algorithms:
 (i) Thinning (ii) Thickening. (06 Marks)

OR

- 8 a. What is Pseudo color image processing? Explain intensity slicing as applied to pseudo color image processing. (07 Marks)
 b. Explain Erosion and Dilation operations used for morphological processing. (07 Marks)
 c. Define wavelet function. (02 Marks)

Module-5

- 9 a. Explain Marr-Hildreth edge detector. (10 Marks)
 b. Write short note on Boundary segments. (06 Marks)

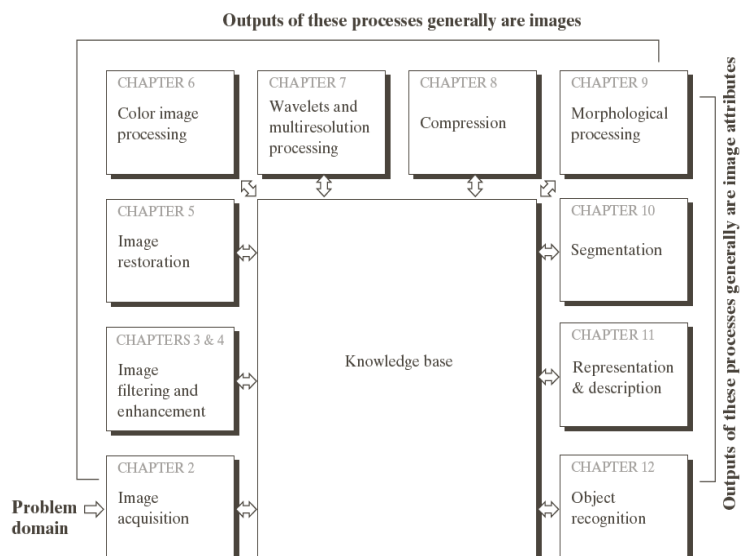
OR

- 10 a. Explain the following boundary descriptors: (i) Shape numbers (ii) Fourier descriptors. (08 Marks)
 b. Explain Global Thresholding using Otsu's method. (08 Marks)

1a.

An image may be defined as a 2D function $f(x,y)$, where x and y are the spatial coordinates and f is the intensity value at x and y .

If x,y and f are all discrete, then the image is called as Digital Image.

Fundamental Steps in Image Processing

1. Image Acquisition: This is the first step or process of the fundamental steps of digital image processing. Image acquisition could be as simple as being given an image that is already in digital form. Generally, the image acquisition stage involves pre-processing, such as scaling etc.

2. Image Enhancement: Image enhancement is among the simplest and most appealing areas of digital image processing. Basically, the idea behind enhancement techniques is to bring out detail that is obscured, or simply to highlight certain features of interest in an image. Such as, changing brightness & contrast etc.

3. Image Restoration: Image restoration is an area that also deals with improving the appearance of an image. However, unlike enhancement, which is subjective, image restoration is objective, in the sense that restoration techniques tend to be based on mathematical or probabilistic models of image degradation.

4. Color Image Processing: Color image processing is an area that has been gaining its importance because of the significant increase in the use of digital images over the Internet. This may include color modeling and processing in a digital domain etc.

5. Wavelets and Multi-Resolution Processing: Wavelets are the foundation for representing images in various degrees of resolution. Images subdivision successively into smaller regions for data compression and for pyramidal representation.

6. Compression: Compression deals with techniques for reducing the storage required to save an image or the bandwidth to transmit it. Particularly in the uses of internet it is very much necessary to compress data.

8. Segmentation: Segmentation procedures partition an image into its constituent parts or objects. In general, autonomous segmentation is one of the most difficult tasks in digital image processing. A rugged segmentation procedure brings the process a long way toward successful solution of imaging problems that require objects to be identified individually.

9. Representation and Description: Representation and description almost always follow the output of a segmentation stage, which usually is raw pixel data, constituting either the boundary of a region or all the points in the region itself. Choosing a representation is only part of the solution for transforming raw data into a form suitable for subsequent computer processing. Description deals with extracting attributes that result in some quantitative information of interest or are basic for differentiating one class of objects from another.

10. Object recognition: Recognition is the process that assigns a label, such as, “vehicle” to an object based on its descriptors.

11. Knowledge Base: Knowledge may be as simple as detailing regions of an image where the information of interest is known to be located, thus limiting the search that has to be conducted in seeking that information. The knowledge base also can be quite complex, such as an interrelated list of all major possible defects in a materials inspection problem or an image database containing high-resolution satellite images of a region in connection with change-detection applications.

1b. Image Sampling and Quantization: The output of most sensors is a continuous voltage waveform whose amplitude and spatial behavior are related to the physical phenomenon being sensed. To create a digital image, we need to convert the continuous sensed data into digital form. This involves two processes: sampling and quantization.

1b.

Basic Concepts in Sampling and Quantization: The basic idea behind sampling and quantization is illustrated in Fig.6.1. Figure 6.1(a) shows a continuous image, $f(x, y)$, that we want to convert to digital form. An image may be continuous with respect to the x - and y -coordinates, and also in amplitude. To convert it to digital form, we have to sample the function in both coordinates and in amplitude. Digitizing the coordinate values is called sampling. Digitizing the amplitude values is called quantization.

The one-dimensional function shown in Fig.6.1 (b) is a plot of amplitude (gray level) values of the continuous image along the line segment AB in Fig. 6.1(a). The random variations are due to image noise.

To sample this function, we take equally spaced samples along line AB, as shown in Fig.6.1 (c). The location of each sample is given by a vertical tick mark in the bottom part of the figure. The samples are shown as small white squares superimposed on the function. The set of these discrete locations gives the sampled function. However, the values of the samples still span (vertically) a continuous range of gray-level values. In order to form a digital function, the gray-level values also must be converted (quantized) into discrete quantities.

The right side of Fig. 6.1 (c) shows the gray-level scale divided into eight discrete levels, ranging from black to white. The vertical tick marks indicate the specific value assigned to each of the eight gray levels. The continuous gray levels are quantized simply by assigning one of the eight discrete gray levels to each sample. The assignment is made depending on the vertical proximity of a sample to a vertical tick mark. The digital samples resulting from both sampling and quantization are shown in Fig.6.1 (d). Starting at the top of the image and carrying out this procedure line by line produces a two-dimensional digital image.

Sampling in the manner just described assumes that we have a continuous image in both coordinate directions as well as in amplitude. In practice, the method of sampling is determined by the sensor arrangement used to generate the image. When an image is generated by a single sensing element combined with mechanical motion, as in Fig. 2.13, the output of the sensor is quantized in the manner described above. However, sampling is accomplished by selecting the number of individual mechanical increments at which we activate the sensor to collect data. Mechanical motion can be made very exact so, in principle; there is almost no limit as to how fine we can sample an image. However, practical limits are established by imperfections in the optics used to focus on the sensor an illumination spot that is inconsistent with the fine resolution achievable with mechanical displacements. When a sensing strip is used for image acquisition, the number of sensors in the strip establishes the sampling limitations in one image direction. Mechanical motion in the other direction can be controlled more accurately, but it makes little sense to try to achieve sampling density in one direction that exceeds the sampling limits established by the number of sensors in the other. Quantization of the sensor outputs completes the process of generating a digital image.

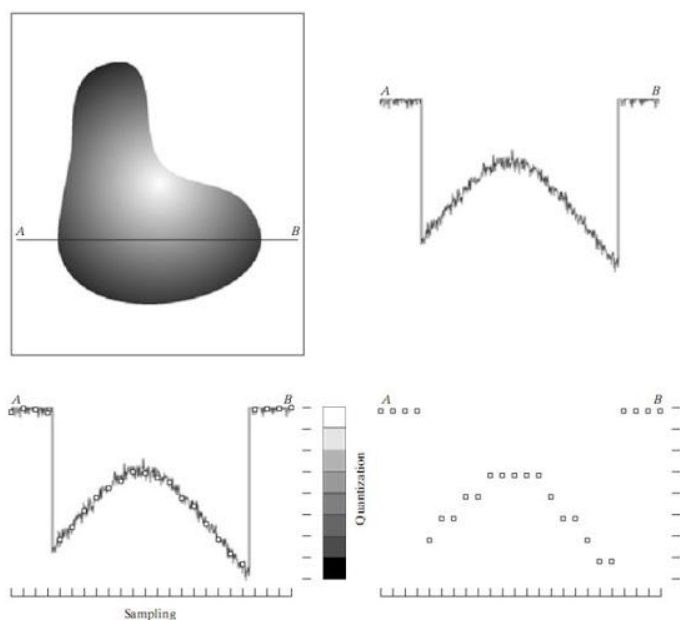


Fig.6.1. Generating a digital image (a) Continuous image (b) A scan line from A to B in the continuous image, used to illustrate the concepts of sampling and quantization (c) Sampling and quantization. (d) Digital scan line

When a sensing array is used for image acquisition, there is no motion and the number of sensors in the array establishes the limits of sampling in both directions. Figure 6.2 illustrates this

concept. Figure 6.2 (a) shows a continuous image projected onto the plane of an array sensor. Figure 6.2 (b) shows the image after sampling and quantization. Clearly, the quality of a digital image is determined to a large degree by the number of samples and discrete gray levels used in sampling and quantization.

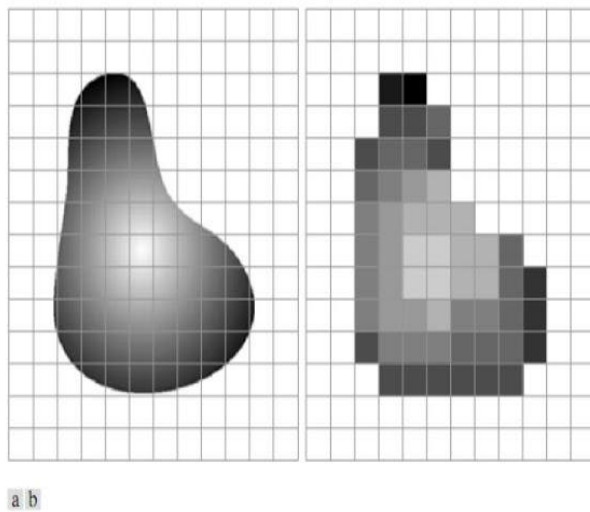


Fig.6.2. (a) Continuous image projected onto a sensor array (b) Result of image sampling and quantization.

1c.

Examples of Fields that use Digital Image processing.

- i. Gamma-Ray Imaging
- ii. X-Ray Imaging
- iii. Imaging in the Microwave Band:
- iv. Imaging in the Radio Band
- v. Imaging in the Visible band and Infrared band:

2a.

Image Acquisition Using Sensor Strips:

A geometry that is used much more frequently than single sensors consists of an in-line arrangement of sensors in the form of a sensor strip, as Fig. 5.1 (b) shows. The strip provides imaging elements in one direction. Motion perpendicular to the strip provides imaging in the other direction, as shown in Fig. 5.3 (a). This is the type of arrangement used in most flat bed scanners. Sensing devices with 4000 or more in-line sensors are possible. In-line sensors are used routinely in airborne imaging applications, in which the imaging system is mounted on an aircraft that flies at a constant altitude and speed over the geographical area to be imaged. Onedimensional imaging sensor strips that respond to various bands of the electromagnetic spectrum are mounted perpendicular to the direction of flight. The imaging strip gives one line of an image at a time, and the motion of the strip completes the other dimension of a two-dimensional image. Lenses or other focusing schemes are used to project the area to be scanned onto the sensors.

Sensor strips mounted in a ring configuration are used in medical and industrial imaging to obtain cross-sectional ("slice") images of 3-D objects, as Fig. 5.3 (b) shows. A rotating X-ray source provides illumination and the portion of the sensors opposite the source collect the X-ray energy that pass through the object (the sensors obviously have to be sensitive to X-ray energy). This is the basis for medical and industrial computerized axial tomography (CAT). It is important to note that the output of the sensors must be processed by reconstruction algorithms whose objective is to transform the sensed data into meaningful cross-sectional images.

In other words, images are not obtained directly from the sensors by motion alone; they require extensive processing. A 3-D digital volume consisting of stacked images is generated as the object is moved in a direction perpendicular to the sensor ring. Other modalities of imaging based on the CAT principle include magnetic resonance imaging (MRI) and positron emission tomography (PET). The illumination sources, sensors, and types of images are different, but conceptually they are very similar to the basic imaging approach shown in Fig. 5.3 (b).

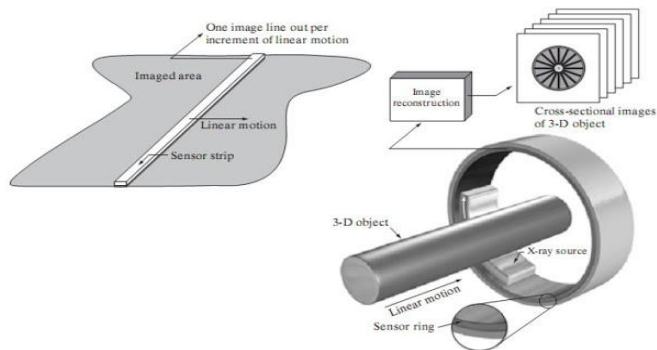


Fig.5.3 (a) Image acquisition using a linear sensor strip (b) Image acquisition using a circular sensor strip.

2b.

Adjacency The concept of adjacency has a slightly different meaning from neighborhood. Adjacency takes into account not just spatial neighborhood but also intensity groups. Suppose we define a set $S=\{0,L-1\}$ of intensities which are considered to belong to the same group. Two pixels p and q will be termed adjacent if both of them have intensities from set S and both also conform to some definition of neighborhood. 4 Adjacency: Two pixels p and q are termed as 4-adjacent if they have intensities from set S and q belongs to $N_4(p)$. 8 Adjacency Two pixels p and q are termed as 4-adjacent if they have intensities from set S and q belongs to $N_8(p)$. Mixed adjacency or m-adjacency : (there shouldn't be closed path)

Two pixels with intensity values from set S are m-adjacent if

- $q \in N_4(p)$,

OR

- $q \in N_D(p)$ and the set $N_4(p) \cap N_4(q)$ has no pixels whose intensity values are from set S .

Mixed adjacency is a modification of 8-adjacency and is used to eliminate the multiple path connections that often arise when 8-adjacency is used.



Fig: (a) Arrangement of pixels; (b) pixels that are 8-adjacent to the center pixel (c) m-adjacency

$$V = \{2, 3, 4\}$$

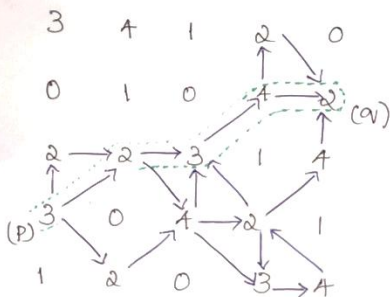
3	4	1	2	0
0	1	0	4	2(q)
2	2	3	1	4
(p) 3	0	4	2	1
1	2	0	3	4

4-path

3	4	1	2	0
0	1	0	4	2(q)
2	2	3	1	4
(p) 3	0	4	2	1
1	2	0	3	4

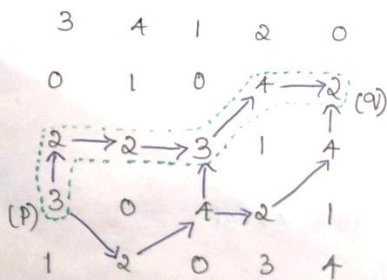
8-path

$V = \{2, 3, 4\}$



The length of the shortest 8-path between p and q is 4

m-path



The length of the shortest m-path between p and q is 5

$$t = \frac{2048 \times 2048 \times (2+8)}{33.6 \times 10^3} \text{ sec}$$

$$= \frac{41943040}{33.6 \times 1000} = 33600 \text{ sec.}$$

$$t = 560 \text{ mins}$$

3a.

3a) 4×4 image, $[0, 9] \Rightarrow L-1=9$
 $\Rightarrow M=N=4$

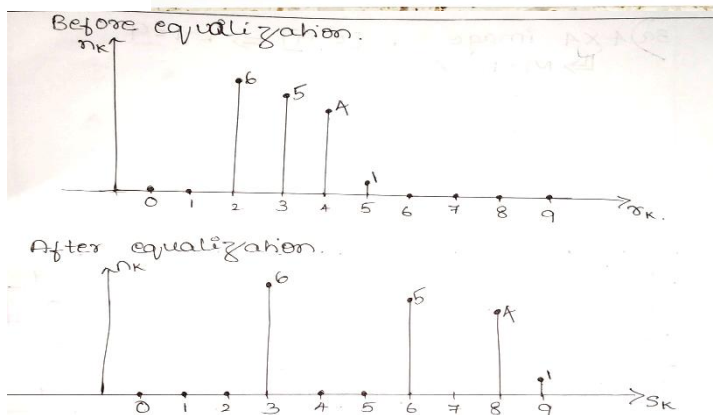
2 3 3 2
 4 2 4 3
 3 2 3 5
 2 4 2 4

r_k	0	1	2	3	4	5	6	7	8	9
n_k	0	0	6	5	4	1	0	0	0	0
$P_r(r_k)$	0	0	0.375	0.3125	0.25	0.0625	0	0	0	0
S_k	0	0	3.3	6.1	8.4	9	9	9	9	9
rand	0	0	3	6	8	9	9	9	9	9

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

$$P_r(r_k) = \frac{n_k}{MN}$$

$$S_k = (L-1) \sum_{j=0}^K P_r(r_j)$$



3b.

IDEAL LOW PASS FILTERS [ILPF]

- It is the simplest low pass filter that cuts-off all the high frequency components of the Fourier Transform that are at a distance greater than specified distance D_0 from the origin of the (centered) Transform.
- Such a filter is called a two dimensional Ideal Lowpass filter (2D – ILPF) and has a transfer function:

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

where; D_0 = Positive integer

$D(u, v)$ = Distance from point (u, v) to the origin of the frequency rectangle

- Due to the fact that the transform has been centered, if the image is of the size $M \times N$; w.k.t its transform will also be of the same size.
- Therefore, center of the frequency rectangle is at $(u, v) = (M/2, N/2)$
- The distance from any point (u, v) to the center of the Fourier Transform is given by:

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

- Fig 4.10(a) shows the 3D – perspective plot of $H(u, v)$ as a function of u and v . Fig.4.10(b) shows $H(u, v)$ displayed as an image.
- Fig. 4.10 (c) shows the filter radial cross section. The filter is radially symmetric about the origin. Therefore, the cross section as a function of the distance from the origin along a radial line is sufficient to specify the filter.

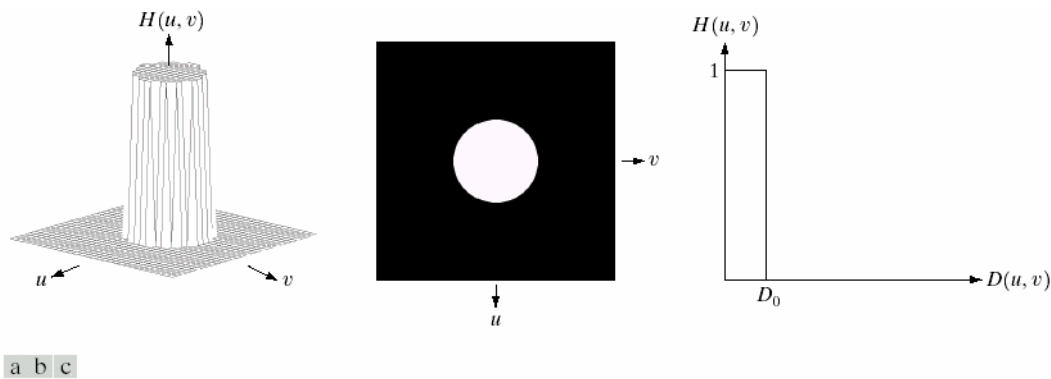


FIGURE 4.10 (a) Perspective plot of an ideal lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross section.

- For an ILPF cross section, the point of transition between $H(u, v)=1$ and $H(u, v)=0$ is called the cut-off frequency. In 4.10(c), D_0 is the cut-off frequency.
- Due to sharp cut-off frequencies ideal low-pass filters cannot be realized using electronic components, they can be implemented in a computer.
- One way to establish a set of standard cut-off frequency loci is to compute circles that enclose specified amounts of total power P_T .
- P_T is obtained by summing the components of the power spectrum at each point (u, v) for $u = 0, 1, 2, \dots, M-1$ and $v = 0, 1, 2, \dots, N-1$ as given:

$$P_T = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} P(u, v)$$
- If the transform has been centered, a circle of radius r with its origin at the center of the frequency rectangle encloses α percent of power where;

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

and the summation is taken over the values of (u, v) that lie inside the circle or on its boundary.

BUTTERWORTH LOWPASS FILTERS [BLPF]

- The transfer function of a Butterworth lowpass filter of order n and with a cut-off frequency at a distance D_0 from the origin is defined as ;

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

Where; $D(u, v)$ is given by:

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

- A perspective plot, image display and radial cross section of the BLPF function are shown in Fig 4.14

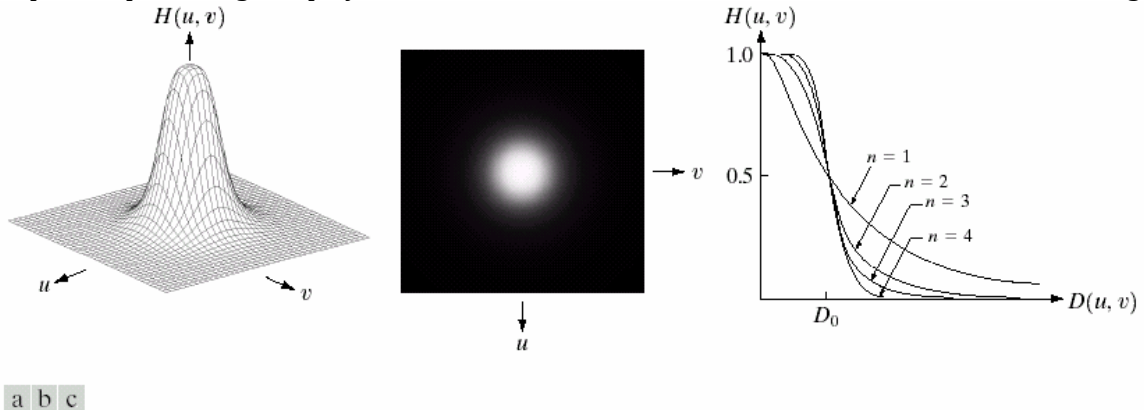


FIGURE 4.14 (a) Perspective plot of a Butterworth lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections of orders 1 through 4.

- Unlike ILPF, BLPF does not have a sharp discontinuity that establishes a clear cut-off between passed and filtered frequencies.
- When $D(u, v) = D_0$, $H(u, v) = 0.5$. That is, the filter is down to 50% of its maximum value 1.

GAUSSIAN LOWPASS FILTERS [GLPF]

- The transfer function for Gaussian lowpass filters for two dimensions is of the form:

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

where; σ is the measure of spread of the Gaussian curve.

- By letting $\sigma = D_0$; the filter can be expressed as:

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

where; D_0 is the cut-off frequency and $D(u, v)$ is given by:

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

- When $D(u, v) = D_0$, $H(u, v) = 0.607$. That is, the filter is down to 60.7% of its maximum value 1.
- A perspective plot, image display and radial cross section of the GLPF function are shown in Fig 4.17
- The spatial domain of the Gaussian filter obtained by computing Inverse DFT of Gaussian filter transfer function $H(u, v)$ has no ringing.
- A Gaussian filter does not achieve as much smoothing as an BLPF of order n as it does not have a tight control over transition.
- Thus, in cases where tight control of transition between high to low frequencies about cut-off

frequency are needed, then the BLPF presents a more suitable choice, but there is a possibility of ringing.

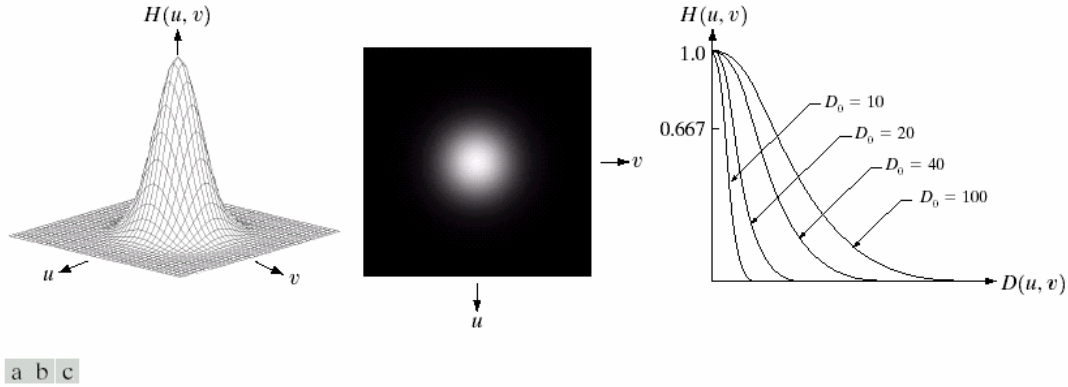


FIGURE 4.17 (a) Perspective plot of a GLPF transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections for various values of D_0 .

4a

The 2D DFT is given by

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

Where $f(x, y)$ is a digital image of size $M \times N$, u and v are discrete variables in the range $u = 0, 1, 2, \dots, M - 1$ and $v = 0, 1, 2, \dots, N - 1$

i. Translation Property

$$f(x, y) e^{j2\pi(\frac{u_0 x}{M} + \frac{v_0 y}{N})} \leftrightarrow F(u - u_0, v - v_0)$$

And

$$f(x - x_0, y - y_0) \leftrightarrow F(u, v) e^{-j2\pi(\frac{x_0 u}{M} + \frac{y_0 v}{N})}$$

ii. Rotation Property

$$f(r, \theta + \theta_0) \leftrightarrow F(w, \varphi + \theta_0)$$

iii. Periodicity Property

$$F(u, v) = F(u + k_1 M, v) = F(u, v + k_2 N) = F(u + k_1 M, v + k_2 N)$$

iv. Convolution Theorem

$$f(x, y) * h(x, y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) h(x - m, y - n), \text{ for } x = 0, 1, 2, \dots, M - 1 \text{ and } y = 0, 1, 2, \dots, N - 1$$

4b.

LOG TRANSFORMATION

- The general form of the log transformation :
 $s = c \log(1+r)$, where c is a constant and r is assumed as: $r \geq 0$.
- The details that are **hidden in dark values are highlighted**.
- It is a *monotonic* and *reversible* process.
- The transformation leads to only one curve.

- The amount of expansion and compression to be done is fixed and cannot be changed.
- It maps a narrow range of low intensity values in the input into a wider range of output values.
- *It compresses the dynamic range of image with large variation in grey level values.*

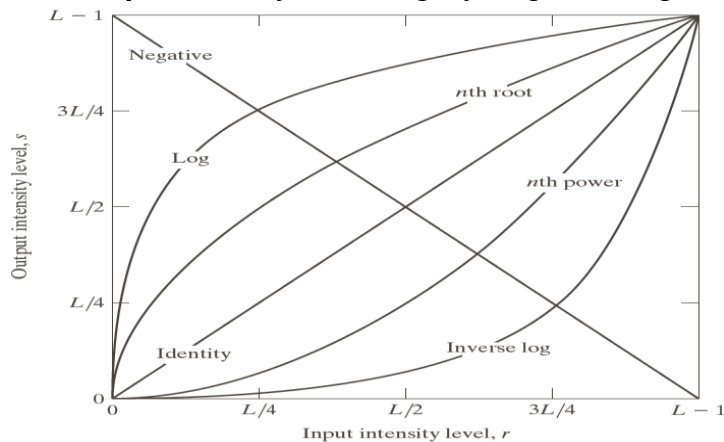
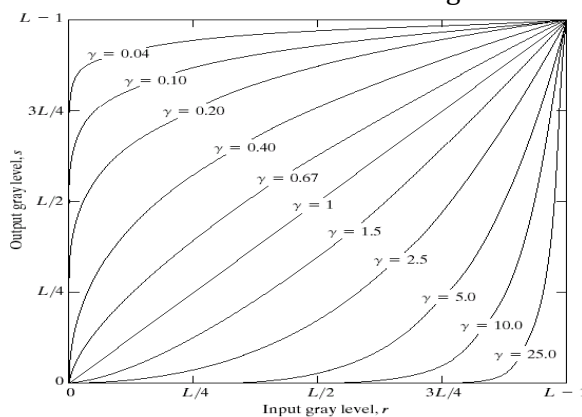


FIGURE 3.3 Some basic intensity transformation functions. All curves were scaled to fit in the range shown.

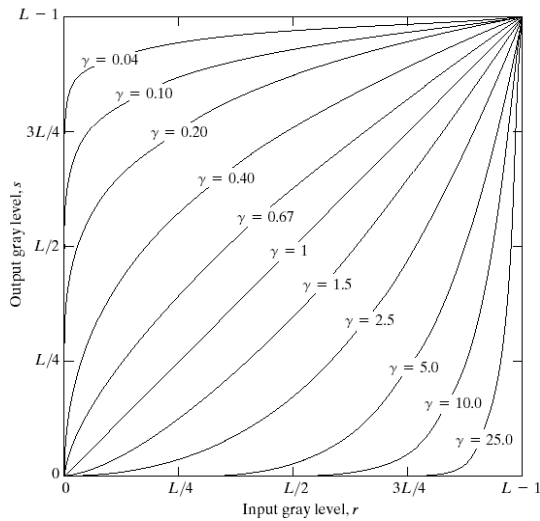
POWER LAW TRANSFORMATION

$$s = cr^\gamma$$

- Power Law Transformation is given by $s = cr^\gamma$ where c and γ are positive constants.
- Variety of devices used for image capture, printing and display respond according to a power law.
- The process used to correct this power-law response phenomena is called **gamma correction**.
- Plots of s versus r for various values of γ are shown in Fig.



- Power-law curves with fractional values of γ map a narrow range of dark input values into a wider range of output values, with the opposite being true for higher values of input levels.
- Unlike the log function, here a family of possible transformation curves can be obtained simply by varying γ .
- Curves generated with values of $\gamma > 1$ have exactly the opposite effect as those generated with values of $\gamma < 1$.
- The above equation reduces to the identity transformation when $c = 1$ and $\gamma = 1$.



4c.

The derivative operator Laplacian for an Image is defined as

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

Fig A

For X-direction,

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

Fig B

For Y-direction,

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

Fig C

By substituting, Equations in Fig.B and Fig.C in Fig.A, we obtain the following equation

$$\Delta^2 f(x, y) = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y)$$

The equation represented in terms of Mask:

0	1	0
1	-4	1
0	1	0

When the diagonals also considered then the equation becomes,

$$\Delta^2 f(x, y) = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) + f(x+1, y-1) + f(x-1, y+1) - 8f(x, y)$$

Fig D

The Mask representation of the above equation,

1	1	1
1	-8	1
1	1	1

Now let's discuss further how image sharpening is done using Laplacian.

Equation:

$$g(x,y) = f(x,y) + c[\Delta^2 f(x,y)]$$

Where $f(x,y)$ is the input image

$g(x,y)$ is the sharpened image and

$c = -1$ for the above mentioned filter masks.(fig.D and fig.E)

5a.

The Gaussian distribution is often used to describe, at least approximately, any variable that tends to cluster around the mean \bar{z} . Gaussian distribution can be completely characterized by its mean \bar{z} and the standard deviation σ .

The Gaussian function has certain very useful mathematical properties. It is symmetric around the point $z = \bar{z}$. The PDF of a 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 the intensity, \bar{z} is the mean(average) value of z , σ is its standard deviation. The standard deviation squared, σ^2 is called the variance of z . approximately 70% of $p(z)$ values will be in the range $[(\bar{z} - \sigma), (\bar{z} + \sigma)]$ and 95% will be in the range $[(\bar{z} - 2\sigma), (\bar{z} + 2\sigma)]$.

The Gaussian model is suitable to model the electronic circuitry noise in image acquisition systems. It is also useful to characterize the sensor noise which can be due to factors like poor illumination or high temperature.

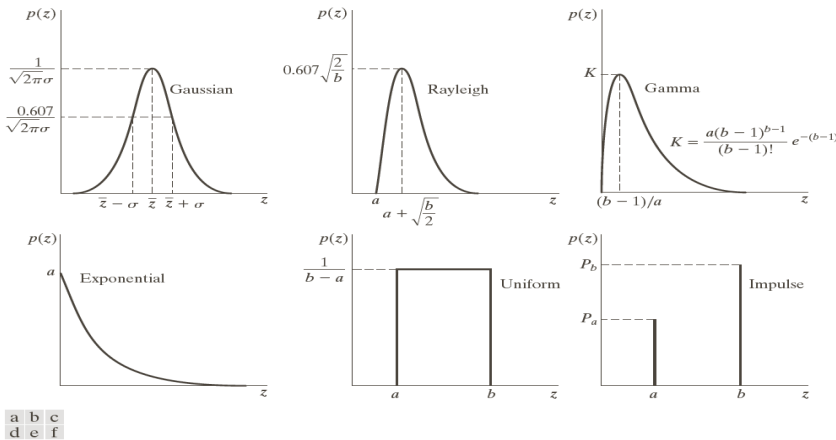


FIGURE 5.2 Some important probability density functions.

The Rayleigh Distribution is left skewed distribution with light tails. An attractive feature of the Rayleigh distribution is that the mode can be estimated from the mean. The range is determined by the scale parameter b . Its skewness is constant.

The PDF of Rayleigh noise is given by

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

The mean and variance of this density are given by

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

And

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

The formula for Rayleigh distribution has 2 factors, the first one $(z-a)$ is a linearly increasing term and the second one $e^{-\frac{(z-a)^2}{b}}$ is an exponentially decaying term like the one in Gaussian. The second term also indicates that the parameter b plays a role similar to the variance.

The Rayleigh distribution is useful for modeling skewed distributions. The Rayleigh distribution is suitable for characterizing noise in range imaging.

The Erlang distribution is also skewed like the Rayleigh distribution. Similar to the Rayleigh distribution, its formula shows two factors, the term z^{b-1} is responsible for the increase and the other term e^{-az} is responsible for the exponential decay. The exponential decay in Erlang distribution is slower compared to Rayleigh because Rayleigh has a quadratic decay term.

- The Erlang distribution is suitable for characterizing noise in range imaging. The PDF of Erlang noise is given by

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

Where $a > 0$, b is positive integer and “!” indicates factorial.

The mean and variance of this density are given by

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

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

Exponential noise is a special case of Erlang distribution with the parameter $b = 1$

The PDF is given by

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

Where $a > 0$. The mean and variance of this density function are given by

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

$$\text{And } \sigma^2 = \frac{1}{a^2}$$

The uniform noise has a PDF which remains constant for specified bounds $a \leq z \leq b$ of the noise amplitude. The constant value of probability is pegged at $\frac{1}{b-a}$ because the total area under the pdf curve is 1. This noise is less practical and is used for random number generator.

The PDF is given by

$$p(z) = \begin{cases} \frac{1}{b-a} & \text{if } a \leq z \leq b \\ 0 & \text{otherwise} \end{cases}$$

The mean and variance of this density function are given by

$$\bar{z} = \frac{a+b}{2}$$

$$\text{And } \sigma^2 = \frac{(b-a)^2}{12}$$

5b.

The Wiener filter solves the signal estimation problem for stationary signals. The filter was introduced by Norbert Wiener in the 1940's. A major contribution was the use of a statistical model for the estimated signal. The noise present in the signal is reduced by comparison with an estimation of the desired noiseless signal. The filter is optimal in the sense of the minimum mean square error. The Wiener filtering approach takes into account both the degradation function and the noise characteristics for estimating the undegraded image. The Wiener filter computes an optimal estimate of the undegraded image.

Assumptions:

1. The power spectrum $S_{\eta}(u, v)$ of the noise is available.
2. The power spectrum $S_f(u, v)$ of the original image is available.

3. The image signal and the noise signal are uncorrelated.
4. Either the image signal or the noise must have zero mean.
5. The intensity levels in the restored image are a linear function of the levels in the degraded image.

Optimality Criteria

The Wiener filter is a minimum mean square error filter. The optimality criteria is to minimize the expected value (i.e. the mean) of the square error between the original image f and the estimate of the un-

degraded image \hat{f} .
$$e^2 = E\{(f - \hat{f})^2\}$$

Here the $E\{\circ\}$ denotes the expected value.

The Wiener Filter expression

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

where $H(u, v)$ = 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 noise

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

- If the $H(u, v)$ is zero, the denominator will remain non-zero unless the noise power spectrum is also zero. This is an advantage over the inverse filter.
- The problem with the Wiener filter is that it requires an estimate of $S_\eta(u, v)$ and $S_f(u, v)$. The latter quantity is difficult to guess because we don't have access to the original image $f(x, y)$.
- In a simplified expression of the Wiener filter we assume that the ratio $\frac{S_\eta(u, v)}{S_f(u, v)}$ is a constant K .

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

•

- When restoring a degraded image using the Wiener filter, we can interactively adjust the value of K as per our visual assessment and obtain the most satisfactory restored image.

6a.

There are 3 principal ways to estimate the degradation function: 1) Observation. 2) Experimentation 3) Mathematical Modeling. The process of restoring an image by using degradation function that has been estimated in some way is called blind deconvolution.

Estimation by Image Observation: This method of estimating the degradation function is used when we have absolutely no clue of what caused the image degradation. We just have the degraded image given to

us. In order to restore the image we must have some idea of what the original image could be looking like. On the given degraded image we select a small patch which has relatively less noise and has good contrast. Following our guesswork, we attempt to restore this patch by applying image operations like sharpening, contrast or brightness adjustment, etc. Our objective here is to get the restored patch. It does not depend on what operations we apply and in what sequence. Let the Fourier transform of the degraded patch be $G_s(u, v)$ and that of the restored patch be $\hat{F}_s(u, v)$. Then the Fourier transform of the degradation function $H_s(u, v)$ can be estimated as:

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

Following our assumption that $H(u, v)$ is position invariant, the degradation function $H(u, v)$ will have the same basic shape as $H_s(u, v)$. However the scale of $H(u, v)$ will be larger compared to that of $H_s(u, v)$.

Estimation by Experimentation: If the image acquisition system which was used to acquire the degraded image is available to us, then we can tune the system settings so that we get an image (not necessarily of the same scene/object) of similar degradation. The idea is to recover the same system settings which were responsible for producing the degradation which we want to estimate. Once we are able to achieve those system settings we need to know the response of the system to an impulse signal. An impulse can be simulated using a small bright dot of light. We record the system's response for this impulse as G_δ in frequency domain. Since the Fourier transform of an impulse is a constant say A the frequency domain representation of the system transfer function, i.e. the degradation $H(u, v)$ is given as:

$$H(u, v) = \frac{G_\delta(u, v)}{A}$$

Estimation by Modeling: We can mathematically model the physical phenomena or the imaging conditions which lead to degraded images. This requires extensive research. For example, it has been possible to model the different type of blurring effects (low-pass filtering) due to various degrees of severity of atmospheric turbulence conditions. For simple cases of blurring due to image motion, it is possible to mathematically derive the degradation function

Estimating Degradation due to motion blur

When we acquire the image of a moving object we generally get a blurred image because of the relative motion between the sensor and the object. In this section we consider how to mathematically model the blur due to image motion. To simplify the modeling we assume that the image moves along a plane and the time varying displacement in x and y directions for every pixel is given as $x_0(t)$ and $y_0(t)$ respectively. If T is the duration for which the camera shutter is open, the intensity at each pixel on the blurred image $g(x, y)$ is computed as an integration of the (true, unblurred) image $f(x, y)$ intensities over the period T .

Relation between Fourier transform of $f(x, y)$ and $g(x, y)$

The Fourier transform of $g(x, y)$ can be written as:

$$\begin{aligned}
 G(u, v) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \textcircled{g(x, y)} e^{-j2\pi(ux+vy)} dx dy \\
 G(u, v) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \textcircled{g(x, y)} e^{-j2\pi(ux+vy)} dx dy \\
 &\quad \downarrow \\
 G(u, v) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left[\int_0^T f[x - x_0(t), y - y_0(t)] dt \right] e^{-j2\pi(ux+vy)} dx dy \\
 &= \int_0^T \left[\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f[x - x_0(t), y - y_0(t)] e^{-j2\pi(ux+vy)} dx dy \right] dt
 \end{aligned}$$

Using the Fourier transform shift property $\mathcal{F}[f(x-a, y-b)] = F(u, v) e^{-j2\pi(ua+vb)}$ we

$$\begin{aligned}
 G(u, v) &= \int_0^T F(u, v) e^{-j2\pi[ux_0(t)+vy_0(t)]} dt \\
 &= F(u, v) \int_0^T e^{-j2\pi[ux_0(t)+vy_0(t)]} dt \\
 &= F(u, v) \left[\int_0^T e^{-j2\pi[ux_0(t)+vy_0(t)]} dt \right] \\
 &\quad \downarrow \\
 &= F(u, v) \textcircled{H(u, v)}
 \end{aligned}$$

have

since $F(u, v)$ does not depend on t

We find that the Fourier transform of degradation due to motion blurring can be formulated as:

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

As we notice from this formulation, the degradation function can be estimated only when the image motion is planar and the time-varying displacements $x_0(t)$ and $y_0(t)$ are known. For general objects which can have articulated motion, it is very difficult to estimate the values of $x_0(t)$ and $y_0(t)$ for the different parts of the objects.

Once we have estimated the degradation function $H(u, v)$ the next step is to use it to restore the image. This process is called as filtering the degraded image so as to get the restored image as the output.

6b.

Order-statistic filters are used when the noise is non-Gaussian. Order statistic involve some kind of ranking or ordering of the intensity values of the pixels within the filter mask. The filter output is then determined based on this rank order. Order statistic filters lead to considerably less blurring compared to the mean filters. All the order statistic filters are non-linear filters.

Median filter

- For this filter the intensity value of a pixel is assigned as the median of the intensity values within the spatial neighborhood (i.e. the filter mask). Median filters are effective for removing salt and pepper noise (impulse noise).

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

- An important issue here is the selection of the size of the filter mask S_{xy} . If the impulse density can be high in particular areas of the image, it is better to choose a larger mask. A larger mask increases the probability that the median value will not correspond to a noise spike.

Max and Min filter

- The max filter is useful for locating image pixels which have the maximum brightness. The maximum brightness value will get copied to the neighboring pixels thereby increasing the brightness level of the image. Likewise the min filter can locate image pixels which have the minimum brightness. The overall image becomes darker as a result of min filtering.

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

Midpoint filter

- The midpoint filter computes the average (midpoint) of the maximum and the minimum values within the filter mask.

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

- It is used for reducing the Gaussian noise or uniform noise.

Adaptive Filters for noise removal

Adaptive filters adapt their response depending on the local characteristics of the image. The filter adapts based on the statistical characteristics of the pixel intensities within the filter mask. Adaptive filters perform better than the non-adaptive filters in terms of preserving the image details while removing the noise.

Adapting to statistical characteristics

The commonly used statistical characteristics are the mean and the variance computed over the local region S_{xy} . The mean m_L or the average intensity within the mask region gives a measure of the local

brightness of the image. The variance σ_L^2 of the pixel intensities within the mask region give a measure of the local image contrast. The local variance value by itself is not much informative but it

can be compared with the variance of the noise σ_η^2 . We assume that the noise will corrupt all regions

in the image equally, and therefore $\sigma_L^2 \geq \sigma_\eta^2$. We consider the following situations in order to define the behavior of the filter: If the local variance is high compared to the noise variance it means that the local spatial region has a high contrast possibly due to edges and other details. We want to preserve such edges and do not want to smoothen them out. Thus the filter should return a

value close to $g(x,y)$. If the local variance is nearly same as the noise variance $\sigma_L^2 \approx \sigma_\eta^2$ it means that the image region was smooth and that the additive noise has corrupted the intensity values thus leading to a variance in intensities. The following expression for the filter response caters to our requirements of adaptive filter behavior.

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

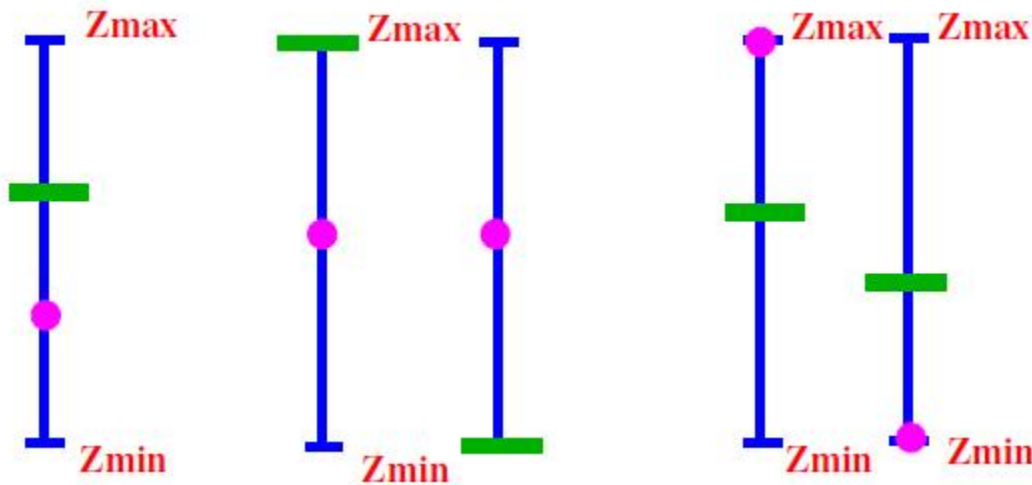
- Practically it is difficult to estimate the correct value for σ_{η}^2 . We use the techniques for noise parameter estimation as given in section 3
- **Adaptive Median Filtering**
- Traditional median filtering works well if the density of salt and pepper noise is less. If the density of the spikes is more (i.e. the spikes occur with a probability > 0.2) then there is a possibility that the median value computed over a spatial neighborhood gets corrupted due to the spikes.

To overcome this problem, the median filter can be made adaptive so that it can remove the impulse noise of high spatial density and also provide smoothing of non-impulse noise.

How to make the median filter adaptive?

- The median filter can be made adaptive by allowing it to vary its size, i.e. the spatial neighborhood window S_{xy} depending on certain conditions.

If we find that the median value coincides with an extreme value in the neighborhood then we do not use the median value as the output intensity of the center pixel. Instead we increase the size of the neighborhood S_{xy} in which the median is computed in the hope that the computed median value would be less likely to coincide with an extreme value. We go on increasing the size of the



- **Figure 1:** In this figure the green bar indicates the computed median value. The pink dot indicates the pixel intensity at $g(x,y)$. The leftmost bar indicates the case when both the center pixel intensity and the computed median value lie between z_{min} and z_{max} . In this case we choose the filter response as z_{xy} . The 2nd and the 3rd bar indicates the case when the median coincides with the extreme intensities in the neighborhood. In this case we increase the size of S_{xy} . The 4th and the 5th bars indicate the case when the center pixel value coincides with the minimum or the maximum intensity in the neighborhood, but the median does not. This is a case where the center pixel is likely to be a noisy spike. The filter response is the median value which replaces z_{xy} .
- neighborhood till the median value is found to be different from the maximum intensity z_{max} or minimum intensity z_{min} within the neighborhood. But if this does not happen and the size of the neighborhood reaches a specified limit, then we accept the median value as the value of the center pixel.

The second modification made to the traditional median filtering is that if the center pixel z_{xy} has a value which is not the maximum or minimum intensity (z_{max} or z_{min}) in the neighborhood then we do not replace it by the median. We preserve the same value z_{xy} in the output image. However if the center pixel has a value which coincides with the maximum or minimum in the neighborhood then we replace it by the median intensity in the neighborhood, provided the median is not the maximum or the minimum.

7a.

Converting colors from HSI to RGB

• **RG sector :** $0^\circ \leq H < 120^\circ$

$$B = I(1 - S)$$

$$R = I \left[1 + \frac{S \cos H}{\cos(60^\circ - H)} \right]$$

$$G = 3I - (R + B)$$

• **GB sector :** $120^\circ \leq H < 240^\circ$

$$H = H - 120^\circ$$

$$R = I(1 - S)$$

$$G = I \left[1 + \frac{S \cos H}{\cos(60^\circ - H)} \right]$$

$$B = 3I - (R + G)$$

• **BR sector :** $240^\circ \leq H < 360^\circ$

$$H = H - 240^\circ$$

$$G = I(1 - S)$$

$$B = I \left[1 + \frac{S \cos H}{\cos(60^\circ - H)} \right]$$

$$R = 3I - (G + B)$$

7b.

The Haar Transform

- Its basis functions are the simplest known orthonormal wavelets.
- The Haar transform is both separable and symmetric:
- $\mathbf{T} = \mathbf{H}\mathbf{F}\mathbf{H}^T$
- F is a NxN image and H is the NxN transformation matrix and T is the NxN transformed image.
- Matrix H contains the Haar basis functions.
- The Haar basis functions $h_k(z)$ are defined for in $0 \leq z \leq 1$, for $k=0,1,..., N-1$, where $N=2^n$.
- To generate **H**:
 - we define the integer $k=2^p+q-1$, with $0 \leq p \leq N-1$.
 - if $p=0$, then $q=0$ or $q=1$.
 - if $p \neq 0$, $1 \leq q \leq 2^p$
 - For the above pairs of p and q , a value for k is determined and the Haar basis functions are computed.

$$h_0(z) = h_{00}(z) = \frac{1}{\sqrt{N}}, \quad z \in [0,1] \quad h_k(z) = h_{pq}(z) = \frac{1}{\sqrt{N}}, \quad \begin{cases} 2^{p/2} & (q-1)/2^p \leq z \leq (q-0.5)/2^p \\ -2^{p/2} & (q-0.5)/2^p \leq z \leq q/2^p \\ 0 & \text{otherwise, } z \in [0,1] \end{cases}$$

The i th row of a NxN Haar transformation matrix contains the elements of $h_k(z)$ for $z=0/N, 1/N, 2/N, ..., (N-1)/N$.

For instance, for $N=4$, p, q and k have the following values:

K	p	q
0	0	0

1	0	1
2	1	1
3	1	2

and the 4x4 transformation matrix is:

$$\mathbf{H}_4 = \frac{1}{\sqrt{4}} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ \sqrt{2} & -\sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & -\sqrt{2} \end{bmatrix}$$

Similarly, for $N=2$, the 2x2 transformation matrix is:

$$\mathbf{H}_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

The rows of \mathbf{H}_2 are the simplest filters of length 2 that may be used as analysis filters $h_0(n)$ and $h_1(n)$ of a perfect reconstruction filter bank.

Moreover, they can be used as scaling and wavelet vectors (defined in what follows) of the simplest and oldest wavelet transform.

7c. Thinning

- The thinning of a set A by a structuring element B , can be defined by terms of the hit-and-miss transform: $A \otimes B = A - (A \circledast B) = A \cap (A \circledast B)^c$
- A more useful expression for thinning A symmetrically is based on a sequence of structuring elements: $A \otimes \{B\} = ((\dots ((A \otimes B^1) \otimes B^2) \dots) \otimes B^n)$
- $\{B\} = \{B^1, B^2, B^3, \dots, B^n\}$
- Where B^i is a rotated version of B^{i-1} . Using this concept we define thinning by a sequence of structuring elements:
- The process is to thin by one pass with B^1 , then thin the result with one pass with B^2 , and so on until A is thinned with one pass with B^n .
- The entire process is repeated until no further changes occur.
- Each pass is preformed using the equation: $A \otimes B = A - (A \circledast B) = A \cap (A \circledast B)^c$
- Thinning is similar to performing a repeated erosion which stops where the shape has been reduced down to a single line. Once it finishes, it leaves a line one pixel thick that runs through the center of where the the shape used to be, along the shape's longer axis.
- The drawback is there is no provision for keeping the skeletons connected.

Thickening

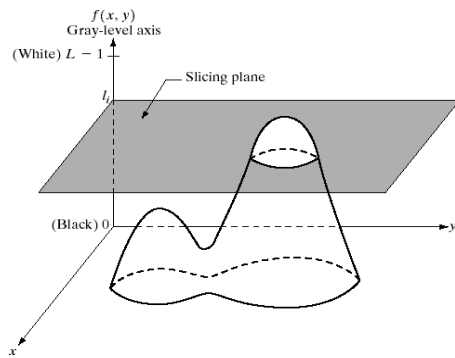
- Thickening is a morphological dual of thinning.
- Definition of thickening
- $A \odot B = A \cup (A \circledast B)$
- As in thinning, thickening can be defined as a sequential operation:

- $A \odot \{B\} = ((\dots((A \odot B^1) \odot B^2) \dots) \odot B^n)$
- the structuring elements used for thickening have the same form as in thinning, but with all 1's and 0's interchanged.
- A separate algorithm for thickening is often used in practice, Instead the usual procedure is to thin the background of the set in question and then complement the result.
- In other words, to thicken a set A, we form $C=A^c$, thin C and then form C^c .
- Depending on the nature of A, this procedure may result in some disconnected points. Therefore thickening by this procedure usually require a simple post-processing step to remove disconnected points.

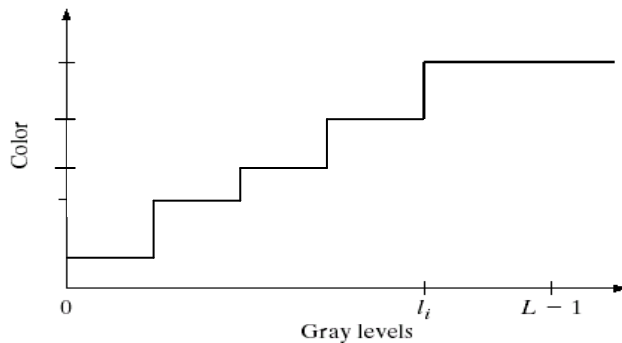
8a.

Pseudocolor image processing

- Also known as indexed color
- Assign colors to gray values based on a fixed criteria
 - 216 index entries from 8-bit RGB color system as a $6 \times 6 \times 6$ cube in a direct color system
 - Gives an integer in the range 0 to 5 for each component of RGB
 - Requires less data to encode an image
 - Some graphics file formats, such as GIF and TIFF add an index colormap to the image with gamma-corrected RGB entries
- Used as an aid to human visualization and interpretation of gray-scale events in an image or sequence of images, such as visualizing population density in different areas on a map
- May have nothing to do with processing of true color images
- Intensity slicing
 - Also called density slicing or color coding
 - Slicing planes parallel to horizontal plane in 3D space, with the intensity of image providing the third dimension on image plane



- Plane at $f(x, y) = l_i$ to slice the image function into two levels *
- Assign different colors to area on different sides of the slicing plane
- Relative appearance of the resulting image manipulated by moving the slicing plane up and down the gray-level axis
- Technique summary
 - * Gray scale representation – $[0, L - 1]$
 - * Black represented by l_0 , $[f(x, y) = 0]$
 - * White represented by $[L-1]$, $[f(x, y) = L - 1]$
 - * Define P planes perpendicular to intensity axis at levels l_1, l_2, \dots, l_P
 - * $0 < P < L - 1$ * P planes partition the gray scale into P + 1 intervals as V_1, V_2, \dots, V_{P+1}
 - * Make gray-level to color assignment as $f(x, y) = c_k$ if $f(x, y) \in V_k$ where c_k is the color associated with kth intensity interval V_k defined by partitioning planes at $l = k - 1$ and $l = k$



Gray lev

- Separate independent transformation of gray level inputs to three colors
- Figure 6.23
- Composite image with color content modulated by nature of transformation function
- Piecewise linear functions of gray levels

Assigning colors to gray levels based on specific mapping functions

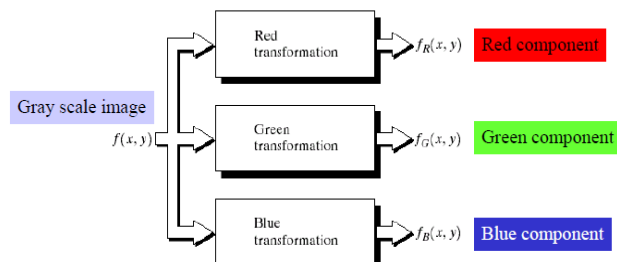


FIGURE 6.23 Functional block diagram for pseudocolor image processing. f_R , f_G , and f_B are fed into the corresponding red, green, and blue inputs of an RGB color monitor.

Combining several monochrome images into a single color composite

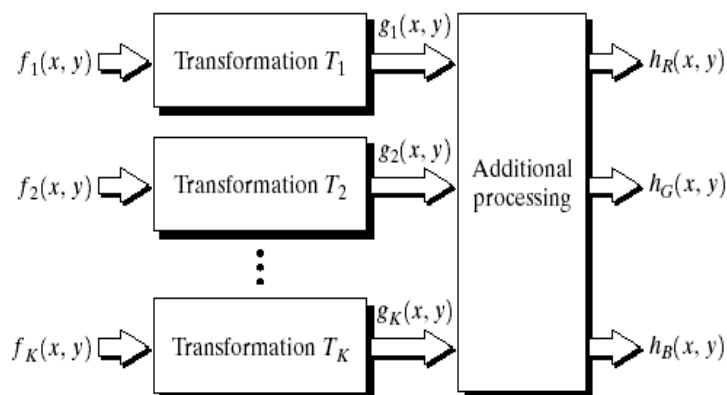


Fig: pseudo color approach used when several monochrome images are available.

Used in multispectral image processing, with different sensors producing individual monochrome images in different spectral bands

8b.

Dilation of A by B and is defined by the following equation:

$$A \oplus B = \{z | (\hat{B})_z \cap A \neq \emptyset\} \quad (9.2 - 1)$$

- This equation is based on obtaining the reflection of B about its origin and shifting this reflection by z.
- The dilation of A by B is the set of all displacements z, such that \hat{B} and A overlap by at least one element. Based On this interpretation the equation of (9.2-1) can be rewritten as:
- $A \oplus B = \{z | [(\hat{B})_z \cap A] \neq \emptyset\}$ (9.2 – 2)
- **Dilation** is the set of all points in the image, where the structuring element “touches” the foreground.
- It adds pixels to the boundaries of objects in an image. It increases the size of object & fills gap.
- Dilation “grows” or “thickens” objects in an binary image.
- The extent of thickening is controlled by the shape of the structuring element used.
- Dilation is used for expanding an element A by using structuring element B.

Erosion

- With A and B as sets in Z^2 , the erosion of A by B, denoted A, is defined as
- $A \ominus B = \{z | (B)_z \subseteq A\}$ (9.2 – 3)
- This equation indicates that the erosion of A by B is the set of all points z such that B, translated by z, is contained in A where B is the structuring element.
- The statement that B has to be contained in A equivalent to B not sharing any common elements with background. The equivalent expression is
- $A \ominus B = \{z | (B)_z \cap A^c = \emptyset\}$
- Erosion is used for shrinking of element A by using element B.
- **Erosion** is the set of all points in the image, where the structuring element “fits into”.
- Good for, e.g.,
- Noise removal in background
- Removal of holes in foreground / background
- One of the simplest uses of erosion is for eliminating irrelevant details (in terms of size) from a binary image.

8c.

7.2.3 Wavelet Functions

Given a scaling function that meets the MRA requirements of the previous section, we can define a *wavelet function* $\psi(x)$ that, together with its integer translates and binary scalings, spans the difference between any two adjacent scaling subspaces, V_j and V_{j+1} . The situation is illustrated graphically in Fig. 7.13. We define the set $\{\psi_{j,k}(x)\}$ of wavelets

$$\psi_{j,k}(x) = 2^{j/2} \psi(2^j x - k) \quad (7.2-19)$$

for all $k \in \mathbf{Z}$ that span the W_j spaces in the figure. As with scaling functions, we write

$$W_j = \overline{\text{Span}\{\psi_{j,k}(x)\}} \quad (7.2-20)$$

and note that if $f(x) \in W_j$,

$$f(x) = \sum_k a_k \psi_{j,k}(x) \quad (7.2-21)$$

The scaling and wavelet function subspaces in Fig. 7.13 are related by

$$V_{j+1} = V_j \oplus W_j \quad (7.2-22)$$

where \oplus denotes the union of spaces (like the union of sets). The orthogonal complement of V_j in V_{j+1} is W_j , and all members of V_j are orthogonal to the members of W_j . Thus,

$$\langle \psi_{j,k}(x), \psi_{j,l}(x) \rangle = 0 \quad (7.2-23)$$

for all appropriate $j, k, l \in \mathbf{Z}$.

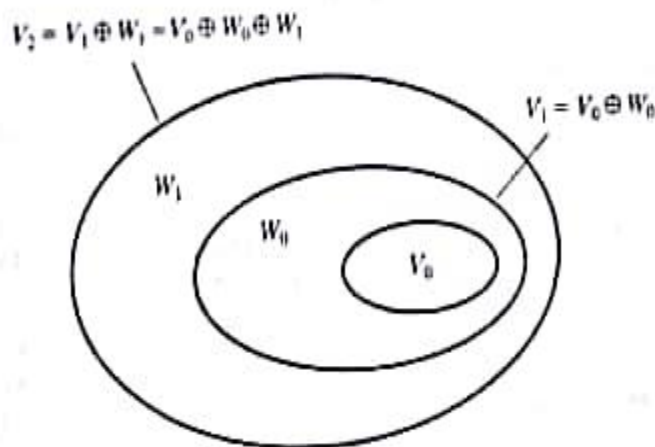


FIGURE 7.13
The relationship
between scaling
and wavelet
function spaces.

We can now express the space of all measurable, square-integrable functions as

$$L^2(\mathbf{R}) = V_0 \oplus W_0 \oplus W_1 \oplus \dots \quad (7.2-24)$$

or

$$L^2(\mathbf{R}) = V_1 \oplus W_1 \oplus W_2 \oplus \dots \quad (7.2-25)$$

or even

$$L^2(\mathbf{R}) = \dots \oplus W_{-2} \oplus W_{-1} \oplus W_0 \oplus W_1 \oplus W_2 \oplus \dots \quad (7.2-26)$$

which eliminates the scaling function, and represents a function in terms of wavelets alone [i.e., there are only wavelet function spaces in Eq. (7.2-26)]. Note that if $f(x)$ is an element of V_1 , but not V_0 , an expansion using Eq. (7.2-24) contains an *approximation* of $f(x)$ using V_0 scaling functions. Wavelets from W_0 would encode the *difference* between this approximation and the actual function. Equations (7.2-24) through (7.2-26) can be generalized to yield

$$L^2(\mathbf{R}) = V_{j_0} \oplus W_{j_0} \oplus W_{j_0+1} \oplus \dots \quad (7.2-27)$$

where j_0 is an arbitrary starting scale.

Since wavelet spaces reside within the spaces spanned by the next higher resolution scaling functions (see Fig. 7.13), any wavelet function—like its scaling function counterpart of Eq. (7.2-18)—can be expressed as a weighted sum of shifted, double-resolution scaling functions. That is, we can write

$$\psi(x) = \sum_n h_\psi(n) \sqrt{2} \varphi(2x - n) \quad (7.2-28)$$

where the $h_\psi(n)$ are called the *wavelet function coefficients* and h_ψ is the *wavelet vector*. Using the condition that wavelets span the orthogonal complement spaces in Fig. 7.13 and that integer wavelet translates are orthogonal, it can be shown that $h_\psi(n)$ is related to $h_\varphi(n)$ by (see, for example, Burrus, Gopinath, and Guo [1998])

$$h_\psi(n) = (-1)^n h_\varphi(1 - n) \quad (7.2-29)$$

Note the similarity of this result and Eq. (7.1-14), the relationship governing the impulse responses of orthonormal subband coding and decoding filters.

The Marr-Hildreth edge detector

One of the earliest successful attempts at incorporating more sophisticated analysis into the edge-finding process is attributed to Marr and Hildreth [1980]. Edge-detection methods in use at the time were based on using small operators (such as the Sobel masks), as discussed in the previous section. Marr and Hildreth argued (1) that intensity changes are not independent of image scale and so their detection requires the use of operators of different sizes; and (2) that a sudden intensity change will give rise to a peak or trough in the first derivative or, equivalently, to a zero crossing in the second derivative (as we saw in Fig. 10.10).

These ideas suggest that an operator used for edge detection should have two salient features. First and foremost, it should be a differential operator capable of computing a digital approximation of the first or second derivative at every point in the image. Second, it should be capable of being "tuned" to act at any desired scale, so that large operators can be used to detect blurry edges and small operators to detect sharply focused fine detail.

Marr and Hildreth argued that the most satisfactory operator fulfilling these conditions is the filter $\nabla^2 G$ where, as defined in Section 3.6.2, ∇^2 is the Laplacian operator, $(\partial^2/\partial x^2 + \partial^2/\partial y^2)$, and G is the 2-D Gaussian function

$$G(x, y) = e^{-\frac{x^2 + y^2}{2\sigma^2}} \quad (10.2-21)$$

with standard deviation σ (sometimes σ is called the *space constant*). To find an expression for $\nabla^2 G$ we perform the following differentiations:

$$\begin{aligned} \nabla^2 G(x, y) &= \frac{\partial^2 G(x, y)}{\partial x^2} + \frac{\partial^2 G(x, y)}{\partial y^2} \\ &= \frac{\partial}{\partial x} \left[\frac{-x}{\sigma^2} e^{-\frac{x^2 + y^2}{2\sigma^2}} \right] + \frac{\partial}{\partial y} \left[\frac{-y}{\sigma^2} e^{-\frac{x^2 + y^2}{2\sigma^2}} \right] \\ &= \left[\frac{x^2}{\sigma^4} - \frac{1}{\sigma^2} \right] e^{-\frac{x^2 + y^2}{2\sigma^2}} + \left[\frac{y^2}{\sigma^4} - \frac{1}{\sigma^2} \right] e^{-\frac{x^2 + y^2}{2\sigma^2}} \end{aligned} \quad (10.2-22)$$

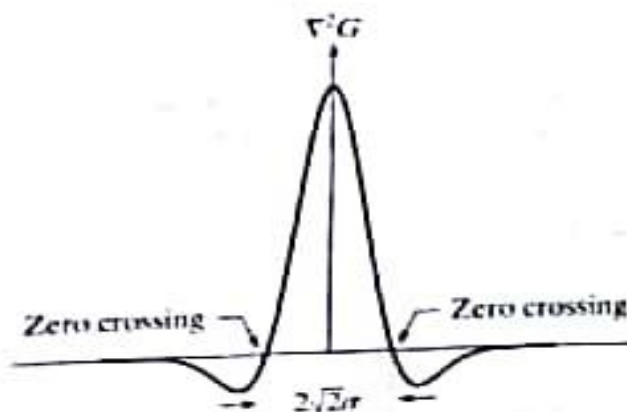
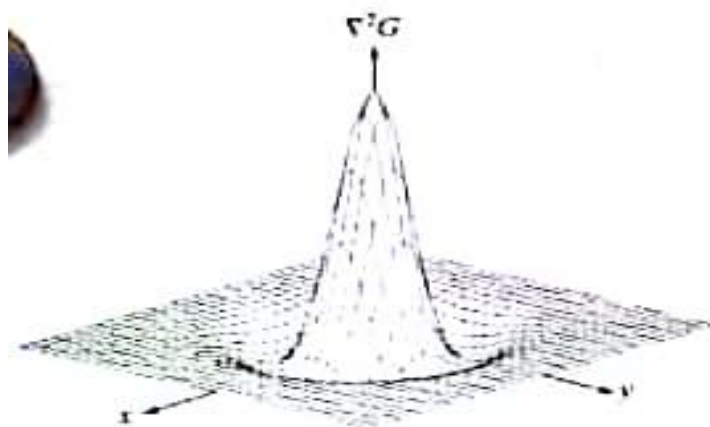
Collecting terms gives the final expression:

$$\nabla^2 G(x, y) = \left[\frac{x^2 + y^2 - 2\sigma^2}{\sigma^4} \right] e^{-\frac{x^2 + y^2}{2\sigma^2}} \quad (10.2-23)$$

This expression is called the *Laplacian of a Gaussian* (LoG).

Figures 10.21(a) through (c) show a 3-D plot, image, and cross section of the *negative* of the LoG function (note that the zero crossings of the LoG occur at $x^2 + y^2 = 2\sigma^2$, which defines a circle of radius $\sqrt{2}\sigma$ centered on the origin). Because of the shape illustrated in Fig. 10.21(a), the LoG function sometimes is called the *Mexican hat* operator. Figure 10.21(d) shows a 5×5 mask that approximates the shape in Fig. 10.21(a) (in practice we would use the *negative* of this mask). This approximation is not unique. Its purpose is to capture the essential *shape* of the LoG function; in terms of Fig. 10.21(a), this means a positive, central term surrounded by an adjacent, negative region whose values increase as a function of distance from the origin, and a zero outer region. The coefficients must sum to zero so that the response of the mask is zero in areas of constant intensity.

Masks of arbitrary size can be generated by sampling Eq. (10.2-23) and scaling the coefficients so that they sum to zero. A more effective approach for generating a LoG filter is to sample Eq. (10.2-21) to the desired $n \times n$ size and



0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

then convolve¹ the resulting array with a Laplacian mask, such as the mask in Fig. 10.4(a). Because convolving an image array with a mask whose coefficients sum to zero yields a result whose elements also sum to zero (see Problems 3.16 and 10.14), this approach automatically satisfies the requirement that the sum of the LoG filter coefficients be zero. We discuss the issue of selecting the size of LoG filter later in this section.

There are two fundamental ideas behind the selection of the operator $\nabla^2 G$. First, the Gaussian part of the operator blurs the image, thus reducing the intensity of structures (including noise) at scales much smaller than σ . Unlike averaging of the form discussed in Section 3.5 and used in Fig. 10.18, the Gaussian function is smooth in both the spatial and frequency domains (see Section 4.8.3), and is thus less likely to introduce artifacts (e.g., ringing) not present in the original image. The other idea concerns ∇^2 , the second derivative part of the filter. Although first derivatives can be used for detecting abrupt changes in intensity, they are directional operators. The Laplacian, on the other hand, has the important advantage of being isotropic (invariant to rotation), which not only corresponds to characteristics of the human visual system (Marr [1982]) but also responds equally to changes in intensity in any mask direction, thus avoiding having to use multiple masks to calculate the strongest response at any point in the image.

The Marr-Hildreth algorithm consists of convolving the LoG filter with an input image, $f(x, y)$,

$$g(x, y) = [\nabla^2 G(x, y)] \star f(x, y) \quad (10.2-24)$$

and then finding the zero crossings of $g(x, y)$ to determine the locations of edges in $f(x, y)$. Because these are linear processes, Eq. (10.2-24) can be written also as

$$g(x, y) = \nabla^2 [G(x, y) \star f(x, y)] \quad (10.2-25)$$

indicating that we can smooth the image first with a Gaussian filter and then compute the Laplacian of the result. These two equations give identical results.

The Marr-Hildreth edge-detection algorithm may be summarized as follows:

1. Filter the input image with an $n \times n$ Gaussian lowpass filter obtained by sampling Eq. (10.2-21).
2. Compute the Laplacian of the image resulting from Step 1 using, for example, the 3×3 mask in Fig. 10.4(a). [Steps 1 and 2 implement Eq. (10.2-25).]
3. Find the zero crossings of the image from Step 2.

To specify the size of the Gaussian filter, recall that about 99.7% of the volume under a 2-D Gaussian surface lies between $\pm 3\sigma$ about the mean. Thus, as a rule

of thumb, the size of an $n \times n$ LoG discrete filter should be such that n is the smallest odd integer greater than or equal to 6σ . Choosing a filter mask smaller than this will tend to “truncate” the LoG function, with the degree of truncation being inversely proportional to the size of the mask; using a larger mask would make little difference in the result.

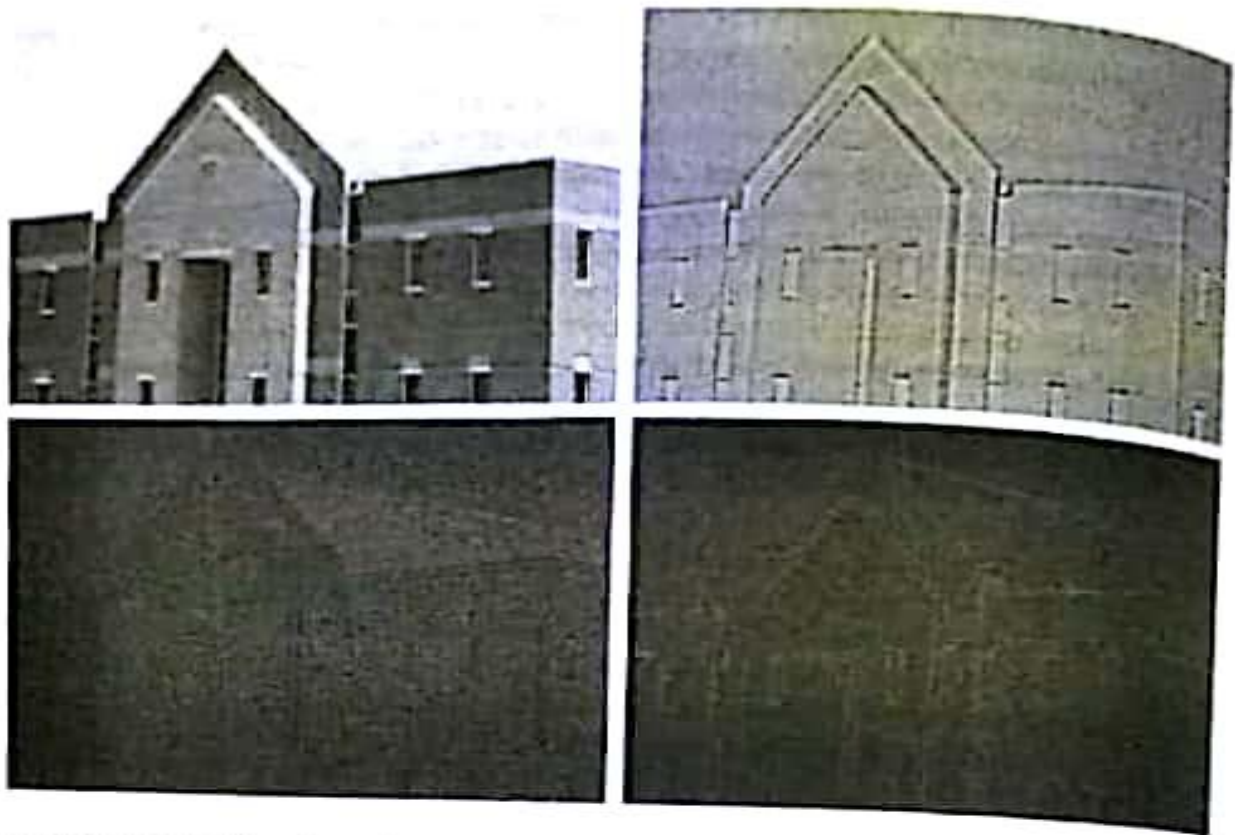
One approach for finding the zero crossings at any pixel, p , of the filtered image, $g(x, y)$, is based on using a 3×3 neighborhood centered at p . A zero crossing at p implies that the signs of at least two of its opposing neighboring pixels must differ. There are four cases to test: left/right, up/down, and the two diagonals. If the values of $g(x, y)$ are being compared against a threshold (a common approach), then not only must the signs of opposing neighbors be different, but the absolute value of their numerical difference must also exceed the threshold before we can call p a zero-crossing pixel. We illustrate this approach in Example 10.7 below.

Zero crossings are the key feature of the Marr-Hildreth edge-detection method. The approach discussed in the previous paragraph is attractive because of its simplicity of implementation and because it generally gives good results. If the accuracy of the zero-crossing locations found using this method is inadequate in a particular application, then a technique proposed by Huertas and Medioni [1986] for finding zero crossings with subpixel accuracy can be employed.

■ Figure 10.22(a) shows the original building image used earlier and Fig. 10.22(b) is the result of Steps 1 and 2 of the Marr-Hildreth algorithm, using $\sigma = 4$ (approximately 0.5% of the short dimension of the image) and $n = 25$ (the smallest odd integer greater than or equal to 6σ , as discussed earlier). As in Fig. 10.5, the gray tones in this image are due to scaling. Figure 10.22(c) shows the zero crossings obtained using the 3×3 neighborhood approach discussed above with a threshold of zero. Note that all the edges form closed loops. This so-called “spaghetti” effect is a serious drawback of this method when a threshold value of zero is used (Problem 10.15). We avoid closed-loop edges by using a positive threshold.

Figure 10.22(d) shows the result of using a threshold approximately equal to 4% of the maximum value of the LoG image. Note that the majority of the principal edges were readily detected and “irrelevant” features, such as the edges due to the bricks and the tile roof, were filtered out. As we show in the next section, this type of performance is virtually impossible to obtain using the gradient-based edge-detection techniques discussed in the previous section. Another important consequence of using zero crossings for edge detection is that the resulting edges are 1 pixel thick. This property simplifies subsequent stages of processing, such as edge linking. ■

A procedure used sometimes to take into account the fact mentioned earlier that intensity changes are scale dependent is to filter an image with various values of σ . The resulting zero-crossings edge maps are then combined by keeping only the edges that are common to all maps. This approach can yield



useful information, but, due to its complexity, it is used in practice mostly as a design tool for selecting an appropriate value of σ to use with a single filter.

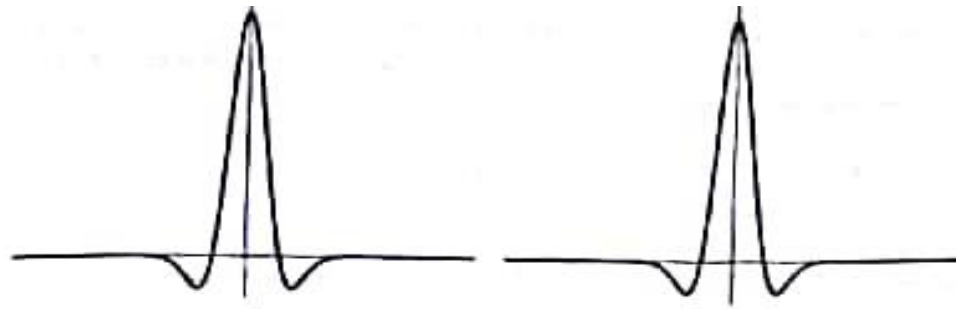
Marr and Hildreth [1980] noted that it is possible to approximate the LoG filter in Eq. (10.2-23) by a difference of Gaussians (DoG):

$$\text{DoG}(x, y) = \frac{1}{2\pi\sigma_1^2} e^{-\frac{x^2+y^2}{2\sigma_1^2}} - \frac{1}{2\pi\sigma_2^2} e^{-\frac{x^2+y^2}{2\sigma_2^2}} \quad (10.2-26)$$

with $\sigma_1 > \sigma_2$. Experimental results suggest that certain “channels” in the human vision system are selective with respect to orientation and frequency, and can be modeled using Eq. (10.2-26) with a ratio of standard deviations of 1.75:1. Marr and Hildreth suggested that using the ratio 1.6:1 preserves the basic characteristics of these observations and also provides a closer “engineering” approximation to the LoG function. To make meaningful comparisons between the LoG and DoG, the value of σ for the LoG must be selected as in the following equation so that the LoG and DoG have the same zero crossings (Problem 10.17):

$$\sigma^2 = \frac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 - \sigma_2^2} \ln \left[\frac{\sigma_1^2}{\sigma_2^2} \right] \quad (10.2-27)$$

Although the zero crossings of the LoG and DoG will be the same when this value of σ is used, their amplitude scales will be different. We can make them compatible by scaling both functions so that they have the same value at the origin.



The profiles in Figs. 10.23(a) and (b) were generated with standard deviation ratios of 1:1.75 and 1:1.6, respectively (by convention, the curves shown are inverted, as in Fig. 10.21). The LoG profiles are shown as solid lines while the DoG profiles are dotted. The curves shown are intensity profiles through the center of LoG and DoG arrays generated by sampling Eq. (10.2-23) (with the constant in $1/2\pi\sigma^2$ in front) and Eq. (10.2-26), respectively. The amplitude of all curves at the origin were normalized to 1. As Fig. 10.23(b) shows, the ratio 1:1.6 yielded a closer approximation between the LoG and DoG functions.

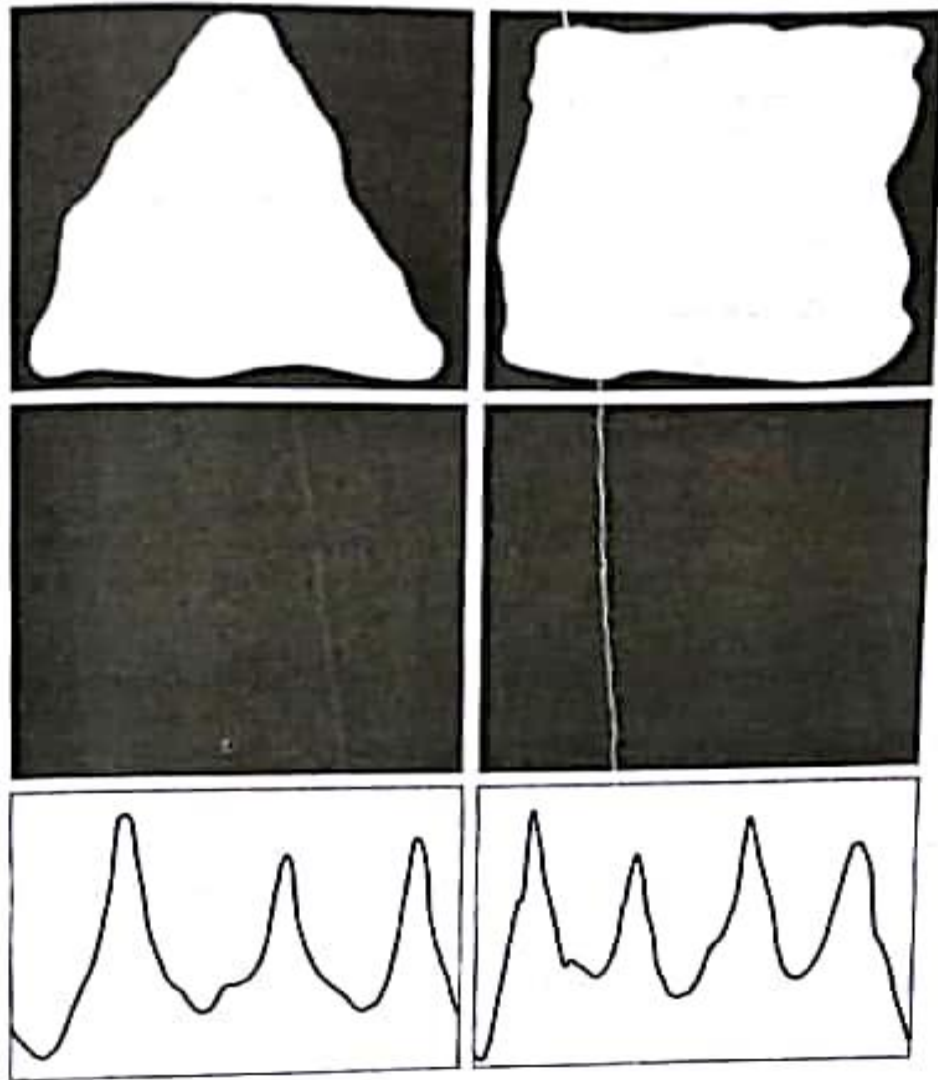
Both the LoG and the DoG filtering operations can be implemented with 1-D convolutions instead of using 2-D convolutions directly (Problem 10.19). For an image of size $M \times N$ and a filter of size $n \times n$, doing so reduces the number of multiplications and additions for each convolution from being proportional to n^2MN for 2-D convolutions to being proportional to nMN for 1-D convolutions. This implementation difference is significant. For example, if $n = 25$, a 1-D implementation will require on the order of 12 times fewer multiplication and addition operations than using 2-D convolution.

11.1.6 Boundary Segments

Decomposing a boundary into segments is often useful. Decomposition reduces the boundary's complexity and thus simplifies the description process. This approach is particularly attractive when the boundary contains one or more significant concavities that carry shape information. In this case, use of the convex hull of the region enclosed by the boundary is a powerful tool for robust decomposition of the boundary.

As defined in Section 9.5.4, the *convex hull* H of an arbitrary set S is the smallest convex set containing S . The set difference $H - S$ is called the *convex deficiency* D of the set S . To see how these concepts might be used to partition a boundary into meaningful segments, consider Fig. 11.12(a), which shows an object (set S) and its convex deficiency (shaded regions). The region boundary can be partitioned by following the contour of S and marking the points at which a transition is made into or out of a component of the convex deficiency. Figure 11.12(b) shows the result in this case. Note that, in principle, this scheme is independent of region size and orientation.

In practice, digital boundaries tend to be irregular because of digitization, noise, and variations in segmentation. These effects usually result in convex deficiencies that have small, meaningless components scattered randomly throughout the boundary. Rather than attempt to sort out these irregularities by postprocessing, a common approach is to smooth a boundary prior to partitioning. There are a number of ways to do so. One way is to traverse the boundary and replace the coordinates of each pixel by the average coordinates of k of its neighbors along the boundary. This approach works for small irregularities, but it is time-consuming and difficult to control. Large values of k can result in excessive smoothing, whereas small values of k might not be sufficient in some




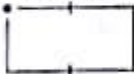
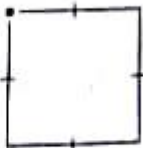

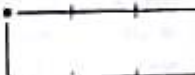
segments of the boundary. A more rugged technique is to use a polygonal approximation prior to finding the convex deficiency of a region. Most digital boundaries of interest are simple polygons (recall from Section 11.1.3 that these are polygons without self-intersection). Graham and Yao [1983] give an algorithm for finding the convex hull of such polygons.



11.2.2 Shape Numbers

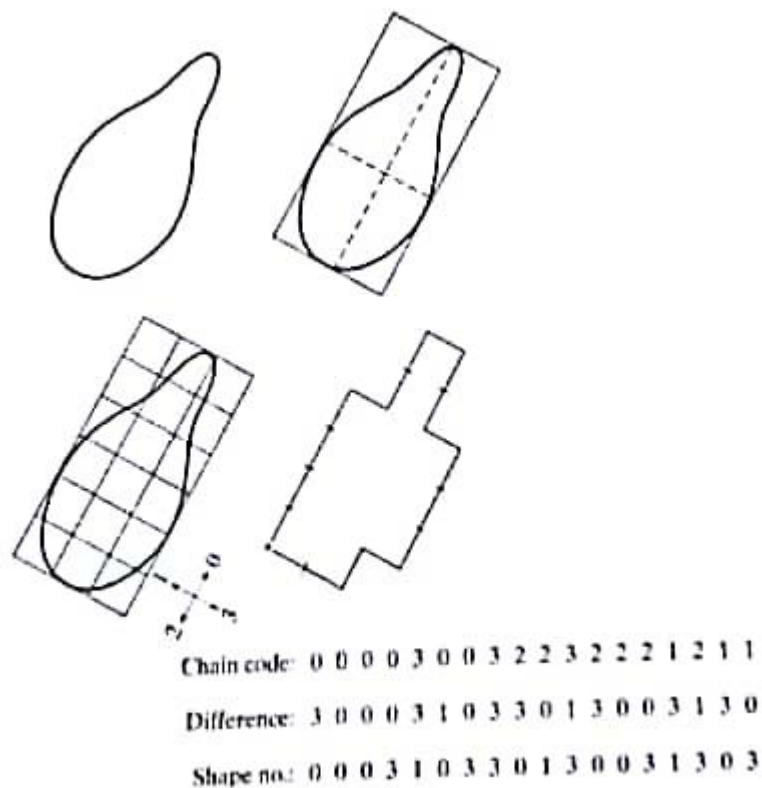
As explained in Section 11.1.2, the first difference of a chain-coded boundary depends on the starting point. The *shape number* of such a boundary, based on the 4-directional code of Fig. 11.3(a), is defined as the first difference of smallest magnitude. The *order* n of a shape number is defined as the number of digits in its representation. Moreover, n is even for a closed boundary, and its value limits the number of possible different shapes. Figure 11.17 shows all the shapes of order 4, 6, and 8, along with their chain-code representations, first differences, and corresponding shape numbers. Note that the first difference is computed by treating the chain code as a circular sequence, as discussed in Section 11.1.2. Although the first difference of a chain code is independent of rotation, in general the coded boundary depends on the orientation of the grid. One way to normalize the grid orientation is by aligning the chain-code grid with the sides of the basic rectangle defined in the previous section.

In practice, for a desired shape order, we find the rectangle of order n whose eccentricity (defined in the previous section) best approximates that of the basic rectangle and use this new rectangle to establish the grid size. For

Order 4		Order 6	
			
Chain code: 0 3 2 1		0 0 3 2 2 1	
Difference: 3 3 3 3		3 0 3 3 0 3	
Shape no.: 3 3 3 3		0 3 3 0 3 3	
Order 8		Order 8	
			
Chain code: 0 0 3 3 2 2 1 1	0 3 0 3 2 2 1 1		0 0 0 3 2 2 2 1
Difference: 3 0 3 0 3 0 3 0	3 3 1 3 3 0 3 0		3 0 0 3 3 0 0 3
Shape no.: 0 3 0 3 0 3 0 3	0 3 0 3 3 1 3 3		0 0 3 3 0 0 3 3

example, if $n = 12$, all the rectangles of order 12 (that is, those whose perimeter length is 12) are 2×4 , 3×3 , and 1×5 . If the eccentricity of the 2×4 rectangle best matches the eccentricity of the basic rectangle for a given boundary, we establish a 2×4 grid centered on the basic rectangle and use the procedure outlined in Section 11.1.2 to obtain the chain code. The shape number follows from the first difference of this code. Although the order of the resulting shape number usually equals n because of the way the grid spacing was selected, boundaries with depressions comparable to this spacing sometimes yield shape numbers of order greater than n . In this case, we specify a rectangle of order lower than n and repeat the procedure until the resulting shape number is of order n .

■ Suppose that $n = 18$ is specified for the boundary in Fig. 11.18(a). To obtain a shape number of this order requires following the steps just discussed. The first step is to find the basic rectangle, as shown in Fig. 11.18(b). The closest rectangle of order 18 is a 3×6 rectangle, requiring subdivision of the basic rectangle as shown in Fig. 11.18(c), where the chain-code directions are aligned with the resulting grid. The final step is to obtain the chain code and use its first difference to compute the shape number, as shown in Fig. 11.18(d). ■



11.2.3 Fourier Descriptors

Figure 11.19 shows a K -point digital boundary in the xy -plane. Starting at an arbitrary point (x_0, y_0) , coordinate pairs $(x_0, y_0), (x_1, y_1), (x_2, y_2), \dots, (x_{K-1}, y_{K-1})$ are encountered in traversing the boundary, say, in the counter-clockwise direction. These coordinates can be expressed in the form $x(k) = x_k$ and $y(k) = y_k$. With this notation, the boundary itself can be represented as the sequence of coordinates $s(k) = [x(k), y(k)]$, for $k = 0, 1, 2, \dots, K-1$. Moreover, each coordinate pair can be treated as a complex number so that

$$s(k) = x(k) + jy(k) \quad (11.2-2)$$

for $k = 0, 1, 2, \dots, K-1$. That is, the x -axis is treated as the real axis and the y -axis as the imaginary axis of a sequence of complex numbers. Although the interpretation of the sequence was recast, the nature of the boundary itself was not changed. Of course, this representation has one great advantage: it reduces a 2-D to a 1-D problem.

From Eq. (4.4-6), the discrete Fourier transform (DFT) of $s(k)$ is

$$a(u) = \sum_{k=0}^{K-1} s(k) e^{-j2\pi uk/K} \quad (11.2-3)$$

for $u = 0, 1, 2, \dots, K-1$. The complex coefficients $a(u)$ are called the *Fourier descriptors* of the boundary. The inverse Fourier transform of these coefficients restores $s(k)$. That is, from Eq. (4.4-7),

$$s(k) = \frac{1}{K} \sum_{u=0}^{K-1} a(u) e^{j2\pi uk/K} \quad (11.2-4)$$

for $k = 0, 1, 2, \dots, K-1$. Suppose, however, that instead of all the Fourier coefficients, only the first P coefficients are used. This is equivalent to setting $a(u) = 0$ for $u > P-1$ in Eq. (11.2-4). The result is the following approximation to $s(k)$:

$$\hat{s}(k) = \frac{1}{P} \sum_{u=0}^{P-1} a(u) e^{j2\pi uk/P} \quad (11.2-5)$$

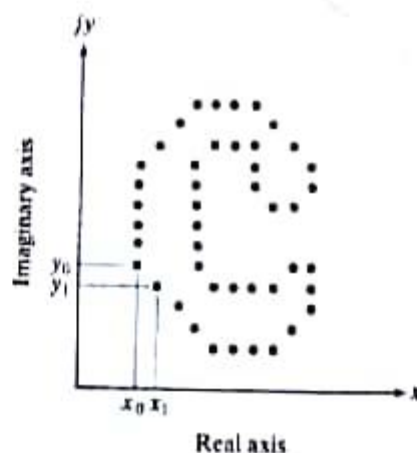
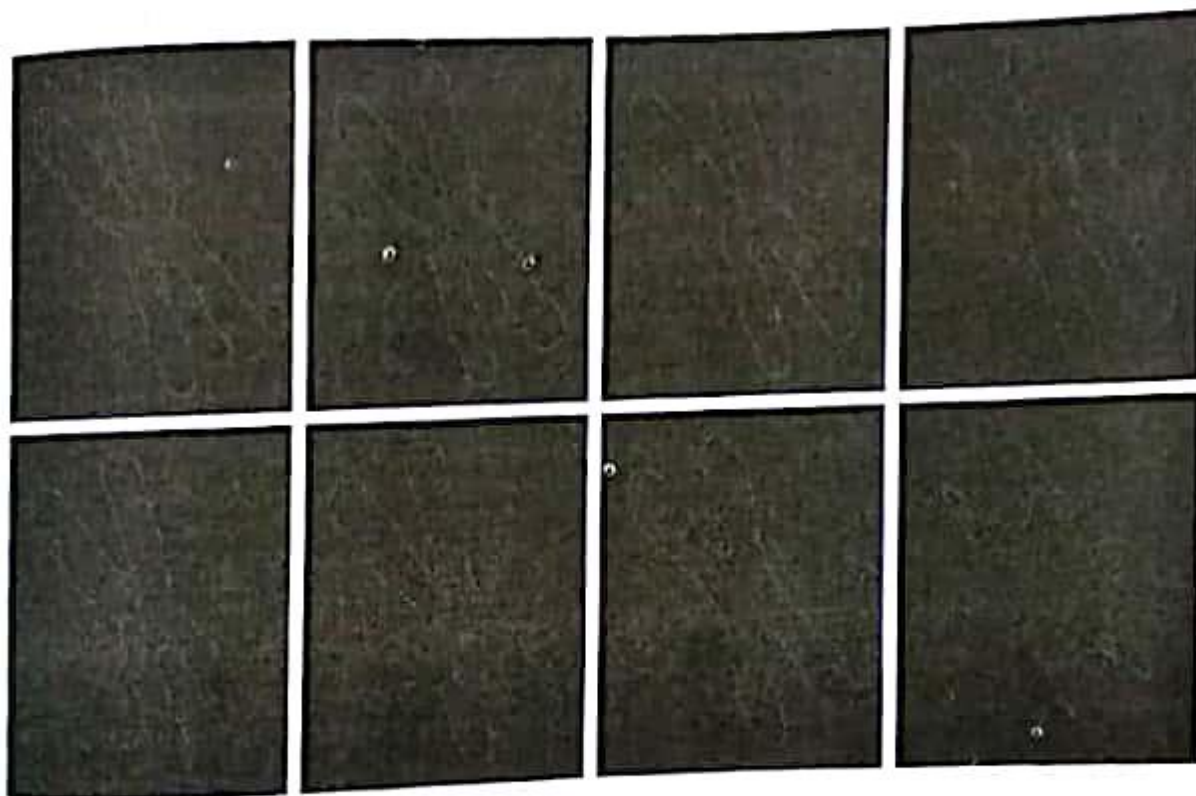


Figure 11.19

for $k = 0, 1, 2, \dots, K - 1$. Although only P terms are used to obtain each component of $\hat{s}(k)$, k still ranges from 0 to $K - 1$. That is, the *same* number of points exists in the approximate boundary, but not as many terms are used in the reconstruction of each point. Recall from discussions of the Fourier transform in Chapter 4 that high-frequency components account for fine detail, and low-frequency components determine global shape. Thus, the smaller P becomes, the more detail that is lost on the boundary, as the following example demonstrates.

■ Figure 11.20(a) shows the boundary of a human chromosome, consisting of 2868 points. The corresponding 2868 Fourier descriptors were obtained for this boundary using Eq. (11.2-3). The objective of this example is to examine the effects of reconstructing the boundary based on decreasing the number of Fourier descriptors. Figure 11.20(b) shows the boundary reconstructed using one-half of the 2868 descriptors. It is interesting to note that there is no perceptible difference between this boundary and the original. Figures 11.20(c) through (h) show the boundaries reconstructed with the number of Fourier

EXAMPLE 11.7:
Using Fourier
descriptors.



a b c d
e f g h

FIGURE 11.20 (a) Boundary of human chromosome (2868 points). (b)–(h) Boundaries reconstructed using 1434, 286, 144, 72, 36, 18, and 8 Fourier descriptors, respectively. These numbers are approximately 50%, 10%, 5%, 2.5%, 1.25%, 0.63%, and 0.28% of 2868, respectively.

10.3.3 Optimum Global Thresholding Using Otsu's Method

Thresholding may be viewed as a statistical-decision theory problem whose objective is to minimize the average error incurred in assigning pixels to two or more groups (also called *classes*). This problem is known to have an elegant closed-form solution known as the *Bayes decision rule* (see Section 12.2.2). The solution is based on only two parameters: the probability density function (PDF) of the intensity levels of each class and the probability that each class occurs in a given application. Unfortunately, estimating PDFs is not a trivial

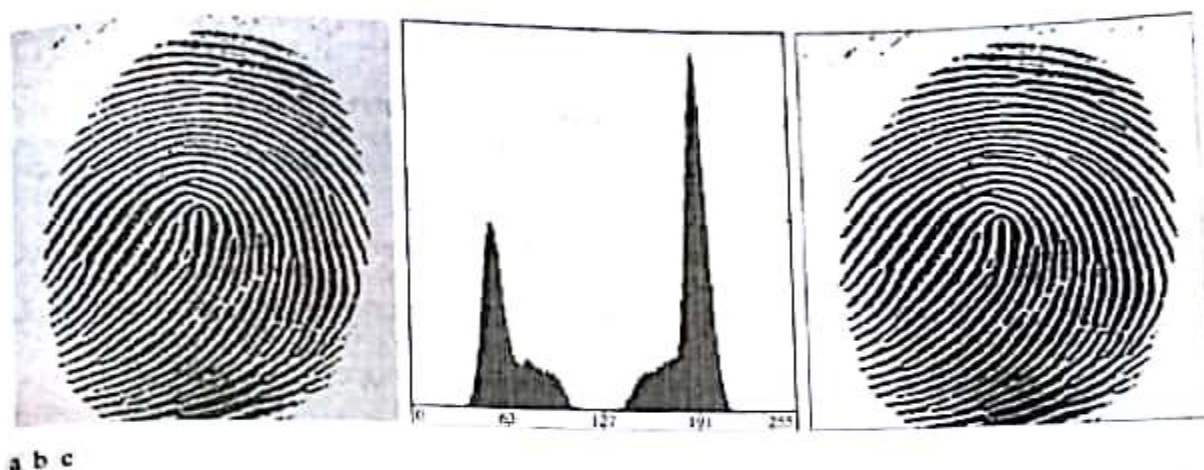


FIGURE 10.38 (a) Noisy fingerprint. (b) Histogram. (c) Segmented result using a global threshold (the border was added for clarity). (Original courtesy of the National Institute of Standards and Technology.)

matter, so the problem usually is simplified by making workable assumptions about the form of the PDFs, such as assuming that they are Gaussian functions. Even with simplifications, the process of implementing solutions using these assumptions can be complex and not always well-suited for practical applications.

The approach discussed in this section, called *Otsu's method* (Otsu [1979]), is an attractive alternative. The method is optimum in the sense that it maximizes the *between-class variance*, a well-known measure used in statistical discriminant analysis. The basic idea is that well-thresholded classes should be distinct with respect to the intensity values of their pixels and, conversely, that a threshold giving the best separation between classes in terms of their intensity values would be the best (optimum) threshold. In addition to its optimality, Otsu's method has the important property that it is based entirely on computations performed on the histogram of an image, an easily obtainable 1-D array.

Let $\{0, 1, 2, \dots, L - 1\}$ denote the L distinct intensity levels in a digital image of size $M \times N$ pixels, and let n_i denote the number of pixels with intensity i . The total number, MN , of pixels in the image is $MN = n_0 + n_1 + n_2 + \dots + n_{L-1}$. The normalized histogram (see Section 3.3) has components $p_i = n_i/MN$, from which it follows that

$$\sum_{i=0}^{L-1} p_i = 1, \quad p_i \geq 0 \quad (10.3-3)$$

Now, suppose that we select a threshold $T(k) = k$, $0 < k < L - 1$, and use it to threshold the input image into two classes, C_1 and C_2 , where C_1 consists of all the pixels in the image with intensity values in the range $[0, k]$ and C_2 consists of the pixels with values in the range $[k + 1, L - 1]$. Using this threshold, the probability, $P_1(k)$, that a pixel is assigned to (i.e., thresholded into) class C_1 is given by the cumulative sum

The validity of the following two equations can be verified by direct substitution of the preceding results:

$$P_1 m_1 + P_2 m_2 = m_G \quad (10.3-10)$$

and

$$P_1 + P_2 = 1 \quad (10.3-11)$$

where we have omitted the k s temporarily in favor of notational clarity.

In order to evaluate the "goodness" of the threshold at level k we use the normalized, dimensionless metric

$$\eta = \frac{\sigma_B^2}{\sigma_G^2} \quad (10.3-12)$$

where σ_G^2 is the *global variance* [i.e., the intensity variance of all the pixels in the image, as given in Eq. (3.3-19)],

$$\sigma_G^2 = \sum_{i=0}^{L-1} (i - m_G)^2 p_i \quad (10.3-13)$$

and σ_B^2 is the *between-class variance*, defined as

$$\sigma_B^2 = P_1(m_1 - m_G)^2 + P_2(m_2 - m_G)^2 \quad (10.3-14)$$

This expression can be written also as

$$\begin{aligned} \sigma_B^2 &= P_1 P_2 (m_1 - m_2)^2 \\ &= \frac{(m_G P_1 - m)^2}{P_1(1 - P_1)} \end{aligned} \quad (10.3-15)$$

where m_G and m are as stated earlier. The first line of this equation follows from Eqs. (10.3-14), (10.3-10), and (10.3-11). The second line follows from Eqs. (10.3-5) through (10.3-9). This form is slightly more efficient computationally because the global mean, m_G , is computed only once, so only two parameters, m and P_1 , need to be computed for any value of k .

We see from the first line in Eq. (10.3-15) that the farther the two means m_1 and m_2 are from each other the larger σ_B^2 will be, indicating that the between-class variance is a measure of *separability* between classes. Because σ_G^2 is a constant, it follows that η also is a measure of separability, and maximizing this metric is equivalent to maximizing σ_B^2 . The objective, then, is to determine the threshold value, k , that maximizes the between-class variance, as stated at the beginning of this section. Note that Eq. (10.3-12) assumes implicitly that $\sigma_G^2 > 0$. This variance can be zero only when all the intensity levels in the image are the same, which implies the existence of only one class of pixels. This in turn means that $\eta = 0$ for a constant image since the separability of a single class from itself is zero.

The second step in Eq. (10.3-15) makes sense only if P_1 is greater than 0 and less than 1, which, in view of Eq. (10.3-11), implies that P_2 must satisfy the same condition.

$$P_1(k) = \sum_{i=0}^k p_i \quad (10.3-4)$$

Viewed another way, this is the probability of class C_1 occurring. For example, if we set $k = 0$, the probability of class C_1 having any pixels assigned to it is zero. Similarly, the probability of class C_2 occurring is

$$P_2(k) = \sum_{i=k+1}^{L-1} p_i = 1 - P_1(k) \quad (10.3-5)$$

From Eq. (3.3-18), the mean intensity value of the pixels assigned to class C_1 is

$$\begin{aligned} m_1(k) &= \sum_{i=0}^k iP(i/C_1) \\ &= \sum_{i=0}^k iP(C_1/i)P(i)/P(C_1) \\ &= \frac{1}{P_1(k)} \sum_{i=0}^k ip_i \end{aligned} \quad (10.3-6)$$

where $P_1(k)$ is given in Eq. (10.3-4). The term $P(i/C_1)$ in the first line of Eq. (10.3-6) is the probability of value i , given that i comes from class C_1 . The second line in the equation follows from Bayes' formula:

$$P(A/B) = P(B/A)P(A)/P(B)$$

The third line follows from the fact that $P(C_1/i)$, the probability of C_1 given i , is 1 because we are dealing only with values of i from class C_1 . Also, $P(i)$ is the probability of the i th value, which is simply the i th component of the histogram, p_i . Finally, $P(C_1)$ is the probability of class C_1 , which we know from Eq. (10.3-4) is equal to $P_1(k)$.

Similarly, the mean intensity value of the pixels assigned to class C_2 is

$$\begin{aligned} m_2(k) &= \sum_{i=k+1}^{L-1} iP(i/C_2) \\ &= \frac{1}{P_2(k)} \sum_{i=k+1}^{L-1} ip_i \end{aligned} \quad (10.3-7)$$

The cumulative mean (average intensity) up to level k is given by

$$m(k) = \sum_{i=0}^k ip_i \quad (10.3-8)$$

and the average intensity of the entire image (i.e., the *global* mean) is given by

$$m_G = \sum_{i=0}^{L-1} ip_i \quad (10.3-9)$$

Reintroducing k , we have the final results:

$$\eta(k) = \frac{\sigma_B^2(k)}{\sigma_G^2} \quad (10.3-16)$$

and

$$\sigma_B^2(k) = \frac{[m_G P_1(k) - m(k)]^2}{P_1(k)[1 - P_1(k)]} \quad (10.3-17)$$

Then, the optimum threshold is the value, k^* , that maximizes $\sigma_B^2(k)$:

$$\sigma_B^2(k^*) = \max_{0 \leq k \leq L-1} \sigma_B^2(k) \quad (10.3-18)$$

In other words, to find k^* we simply evaluate Eq. (10.3-18) for all *integer* values of k (such that the condition $0 < P_1(k) < 1$ holds) and select that value of k that yielded the maximum $\sigma_B^2(k)$. If the maximum exists for more than one value of k , it is customary to average the various values of k for which $\sigma_B^2(k)$ is maximum. It can be shown (Problem 10.33) that a maximum always exists, subject to the condition that $0 < P_1(k) < 1$. Evaluating Eqs. (10.3-17) and (10.3-18) for all values of k is a relatively inexpensive computational procedure, because the maximum number of integer values that k can have is L .

Once k^* has been obtained, the input image $f(x, y)$ is segmented as before:

$$g(x, y) = \begin{cases} 1 & \text{if } f(x, y) > k^* \\ 0 & \text{if } f(x, y) \leq k^* \end{cases} \quad (10.3-19)$$

for $x = 0, 1, 2, \dots, M - 1$ and $y = 0, 1, 2, \dots, N - 1$. Note that all the quantities needed to evaluate Eq. (10.3-17) are obtained using only the histogram of $f(x, y)$. In addition to the optimum threshold, other information regarding the segmented image can be extracted from the histogram. For example, $P_1(k^*)$ and $P_2(k^*)$, the class probabilities evaluated at the optimum threshold, indicate the portions of the areas occupied by the classes (groups of pixels) in the thresholded image. Similarly, the means $m_1(k^*)$ and $m_2(k^*)$ are estimates of the average intensity of the classes in the original image.

The normalized metric η , evaluated at the optimum threshold value, $\eta(k^*)$, can be used to obtain a quantitative estimate of the separability of classes, which in turn gives an idea of the ease of thresholding a given image. This measure has values in the range

$$0 \leq \eta(k^*) \leq 1 \quad (10.3-20)$$

The lower bound is attainable only by images with a single, constant intensity level, as mentioned earlier. The upper bound is attainable only by 2-valued images with intensities equal to 0 and $L - 1$ (Problem 10.34).

Otsu's algorithm may be summarized as follows:

1. Compute the normalized histogram of the input image. Denote the components of the histogram by p_i , $i = 0, 1, 2, \dots, L - 1$.
2. Compute the cumulative sums, $P_1(k)$, for $k = 0, 1, 2, \dots, L - 1$, using Eq. (10.3-4).
3. Compute the cumulative means, $m(k)$, for $k = 0, 1, 2, \dots, L - 1$, using Eq. (10.3-8).
4. Compute the global intensity mean, m_G , using (10.3-9).
5. Compute the between-class variance, $\sigma_B^2(k)$, for $k = 0, 1, 2, \dots, L - 1$, using Eq. (10.3-17).
6. Obtain the Otsu threshold, k^* , as the value of k for which $\sigma_B^2(k)$ is maximum. If the maximum is not unique, obtain k^* by averaging the values of k corresponding to the various maxima detected.
7. Obtain the separability measure, η^* , by evaluating Eq. (10.3-16) at $k = k^*$.