E-CONTENT

SUBJECT NAME: IMAGE PROCESSING

SUB CODE : 18K3P3CSELCS3:A

STAFF NAME: M.IDA ROSE
R.VALARMATHY
J.SHANMUGAPRIYA

Semester – III
MBE – 3

Hours - 6
Credit - 4

Image Processing (18KP3CSELCS3:A)

Objective: To apply knowledge in image processing applications.

UNIT I: Introduction : What is Image Processing – The Origins of Digital Image processing - Examples of Fields that Use DIP - Fundamental step in DIP – Components of an image processing System. Digital Image Fundamentals : Elements of Visual Perception – Image Sensing and Acquisition - Image Sampling and Quantization – Some Basic Relationships between Pixels.

UNIT II: Intensity Transformations and Spatial Filtering : Background - Some Basic Intensity Transformation Functions – Histogram Processing - Funtamentals of Spatial Filtering - Smoothing Spatial Filters – Sharpening Spatial Filters - Combining Spatial Enchancement Methods – Using Fuzzy Techniques for Intensity Transformations Spatial Filtering.

UNIT III: Image Restoration and Reconstruction: A Model of Image Degradation/Restoration Process – Noise Models – Restoration in the Presence of Noise only-Spatial Filtering – Periodic Noise Reduction by Freqency Domain Filtering – Periodic Noise Reduction by Frequency Domain Filtering- Linear, Position- Invariant Degradations – Estimating the Degradation Function.

UNIT IV: Image Compression: Fundamentals – Some Basic Compression Methods: Huffman coding- Golomb Coding-Arithmetic Coding - LZW Coding – Run Length Coding– Symbol – Based Coding – Bit- Plane Coding – Block Transform Coding - Predictive Coding - Wavelet Coding.

UNIT V: Morphological Image Processing : Preliminaries –Erosion and Dilation –Opening and Closing - The Hit-or-Miss Transformation – Some Basic Morphological Algorithms – Gray- Scal Morphology- Image Segmentation : Funtamentals- Point, Line and Edge Detection-Thresholding– Object Recognition: Patterns and Pattern Classes – Recoginition Based On Decision – Theoretic Methods.

Text : "Digital Image Processing", Third Edition, First Impression Rafel C.Gonzalez and Richard E. Woods, Pearson Education.

Chapters: 1, 2.1, 2.3 - 2.5, 3.1 – 3.8, 5.1 - 5.6, 8.1 – 8.2, 9, 10.1 - 10.3, 12.1, 12.2

Reference:

1. "Fundamentals of Digital Image Processing" - Anil K. Jain, PHI, Pvt, Ltd, Sixth printing 2001
2. "Digital Image Processing and Analysis", B. Chandra and D. Dutta Majumder, PHI, New Delhi, 2006
3. "Fundamentals of Digital Image Processing" – S.Annadurai – Pearson Education India - 2007

Head, Dept. of Computer Science,
Kunthavai Naachiaar Govt. Arts College (W),
THANJAVUR- 7.

Image Processing

Unit III

Chapter 5 – Image Restoration and Reconstruction

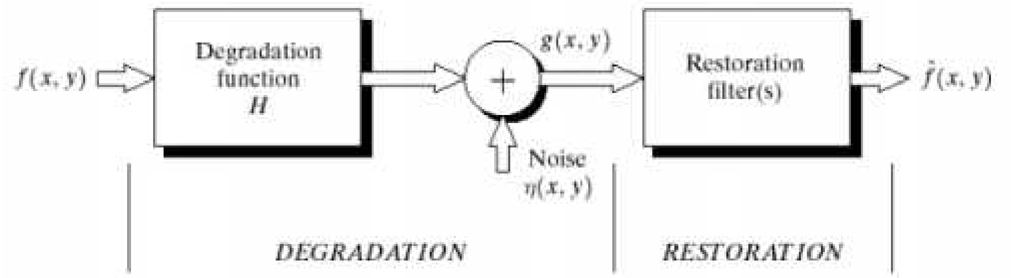


FIGURE 5.1 A model of the image degradation/restoration process.

Noise Models:

- The principal source of noise in digital images arise during image acquision and/or

transmission.

- The performance of imaging sensors is affected by a variety of factors, such as environmental conditions during image acquistion and by the quality of the sensing elements themselves.
- For instance,in acquring images with a CCD camera,light levels and sensor temperature are major factors affecting the amount of noise in the resulting image.
- Images are corrupted during transmission principally due to interference in the

channel used for transmission.

- For example., an image transmitted using a wireless network might be corrupted as a result of lighting or other atmospheric disturbance.
- That define the spatial characteristics of noise, and whether the noise is correlated with

the image.

- Frequency properties refer to the frequency content of noise in the fourier sense.(i.e., as

opposed to frequencies of the electromagnetic spectrum)

- for exmple, when the fourier spectrum of noise is constant, the noise usually is called

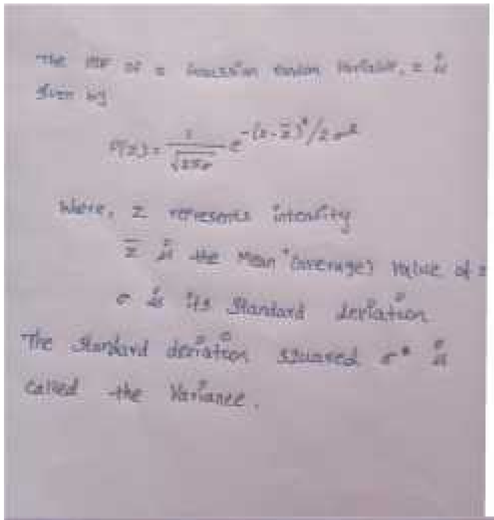
“white noise”.

- This terminology is a carryover from the physical properties of white light,which contains nearly all frequencies in the visible spectrum in equal properties.
- With the exception of spatially periodic noise.That noise is independent of spatial coordinates and that it is uncorrelated with respect to the image itself. (i.e there is no correlation between pixel values and the values of noise component)
- Although these assumptions are atleast partially invalid in some applications quantum_limited imaging,such as x-ray and nuclear-medicine imageing is a good example.
- PDF:Probability Density Functions

The most common PDF’s found in image processing applications:

GAUSSIAN NOISE

- Because of its mathematical trackability in both the spatial and frequency domains.
- Gaussian are also called normal noise model.
- Noise model are used to frequently in practice.
- In fact, this tractability is so convenient that it often results in gaussian models being used in situation which they are marginally applicable at best.



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

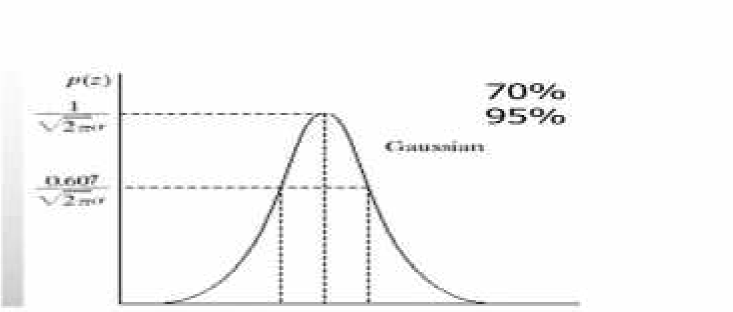
$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(z-\bar{z})^2/2\sigma^2}$$

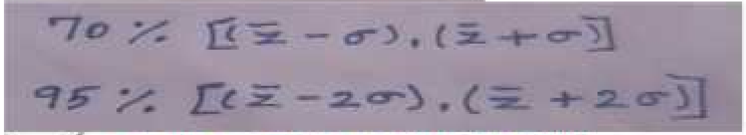
Where, z represents intensity

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

σ is its standard deviation

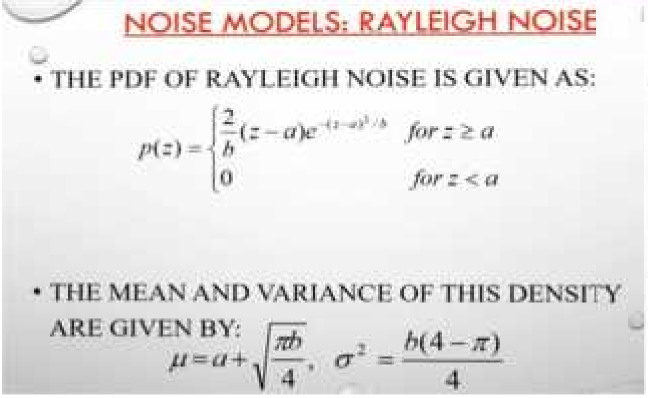
The standard deviation squared σ^2 is called the variance.





70% $[(\bar{z}-\sigma), (\bar{z}+\sigma)]$

95% $[(\bar{z}-2\sigma), (\bar{z}+2\sigma)]$



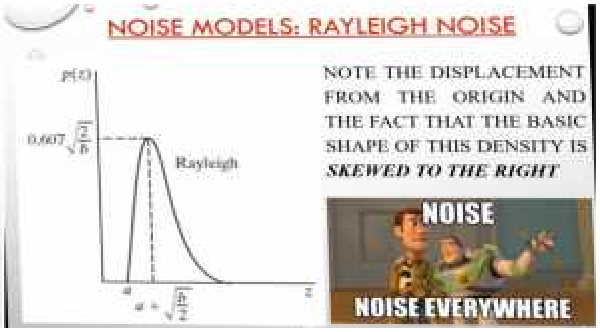
NOISE MODELS: RAYLEIGH NOISE

- THE PDF OF RAYLEIGH NOISE IS GIVEN AS:

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

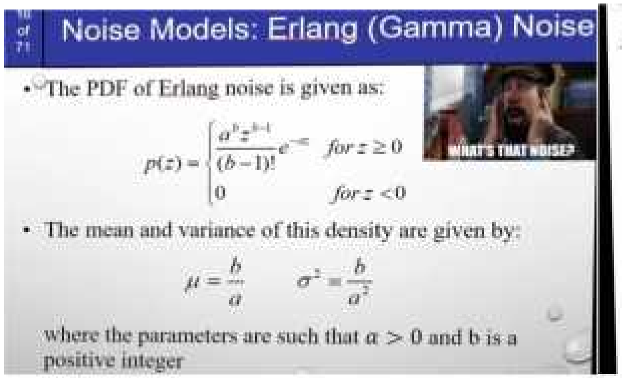
- THE MEAN AND VARIANCE OF THIS DENSITY ARE GIVEN BY:

$$\mu = a + \sqrt{\frac{\pi b}{4}}, \quad \sigma^2 = \frac{b(4-\pi)}{4}$$



NOISE MODELS: RAYLEIGH NOISE

NOTE THE DISPLACEMENT FROM THE ORIGIN AND THE FACT THAT THE BASIC SHAPE OF THIS DENSITY IS ***SKEWED TO THE RIGHT***



Noise Models: Erlang (Gamma) Noise

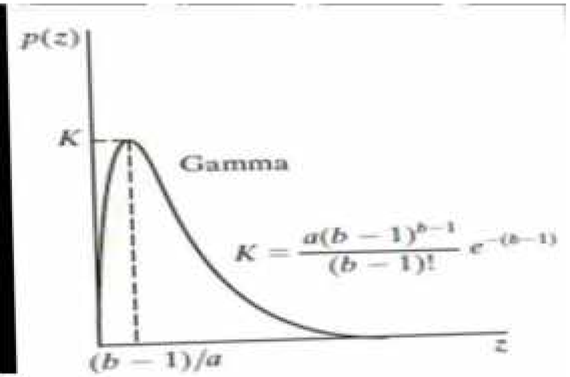
- The PDF of Erlang noise is given as:

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

- The mean and variance of this density are given by:

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

where the parameters are such that $a > 0$ and b is a positive integer



Exponential noise

The PDF of *exponential* noise is given by

$$p(z) = \begin{cases} ae^{-az} & \text{for } z \geq 0 \\ 0 & \text{for } z < 0 \end{cases} \qquad (5.2\text{-}8)$$

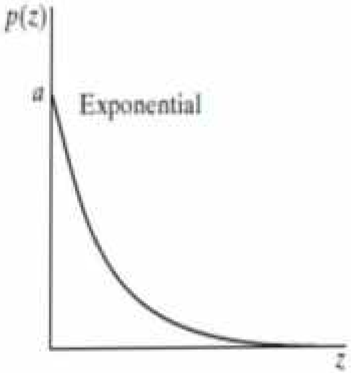
where $a > 0$. The mean and variance of this density function are

$$\bar{z} = \frac{1}{a} \qquad (5.2\text{-}9)$$

and

$$\sigma^2 = \frac{1}{a^2} \qquad (5.2\text{-}10)$$

Note that this PDF is a special case of the Erlang PDF, with $b = 1$. Figure 5.2(d) shows a plot of this density function.



Uniform noise

The PDF of *uniform* noise is given by

$$p(z) = \begin{cases} \frac{1}{b-a} & \text{if } a \leq z \leq b \\ 0 & \text{otherwise} \end{cases} \qquad (5.2\text{-}11)$$

The mean of this density function is given by

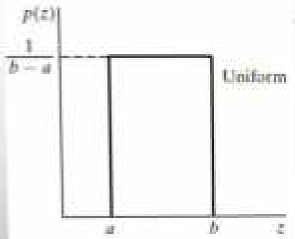
$$\bar{z} = \frac{a+b}{2} \qquad (5.2\text{-}12)$$

and its variance by

$$\sigma^2 = \frac{(b-a)^2}{12} \qquad (5.2\text{-}13)$$

Figure 5.2(e) shows a plot of the uniform density.

NOISE MODELS: UNIFORM NOISE



Impulse (salt-and-pepper) noise

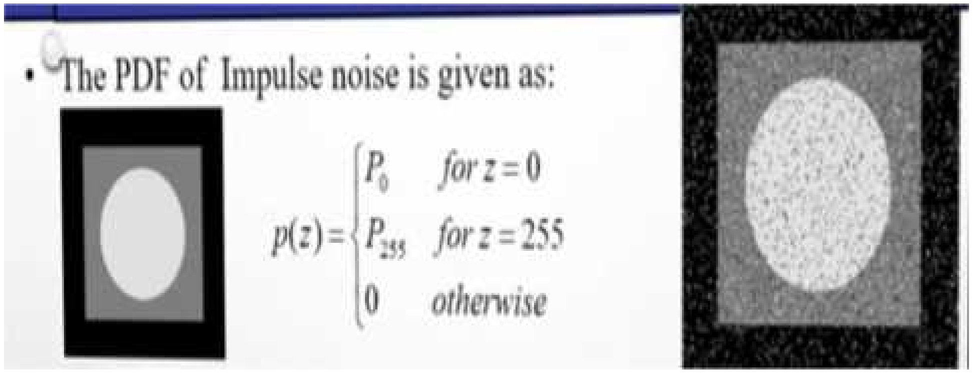
The PDF of (*bipolar*) *impulse* noise is given by

$$p(z) = \begin{cases} P_a & \text{for } z = a \\ P_b & \text{for } z = b \\ 0 & \text{otherwise} \end{cases} \qquad (5.2\text{-}14)$$

- If b>a, intensity b will appear as a light dot in the image.
- Conversely, level a will appear like a dark dot.
- If either is zero, the impulse noise is called unipolar.
- If neither probability is zero, and especially when they are approximately equal, impulse noise values will resemble salt-and-pepper granules (black and white pixels) randomly distributed over the image.
- For this reason, bipolar impulse noise is also called **salt-and-pepper** or **data-drop-out** or **spike noise**.
- Noise impulses can be negative or positive.
- Scaling usually is part of the image digitizing process.
- Because impulse corruption usually is large compared with the strength.
- Thus, the assumption usually is that a and b are

"saturated" values, in the sense that they are equal to the maximum and minimum allowed values in the digitized image.

- As a result, negative impulses appears as black(pepper)points in an image.
- Positive impulses appears as white (salt)noise.



- The PDF of Impulse noise is given as:

$$p(z)=\begin{cases} P_0 & for\ z=0 \\ P_{255} & for\ z=255 \\ 0 & otherwise \end{cases}$$

The preceding PDF's provide useful tools for modeling a broad range of noise corruption situation found in practice.

For example:

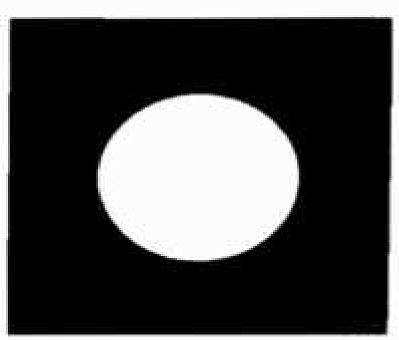
Gaussian noise arises in an image due to factors such as electronic circuit and sensor noise due to poor illumination and/or high temperature. The rayleigh density is helpful in characterizing noise phenomena in range imaging.

The exponential &gamma densities find application in laser imaging.

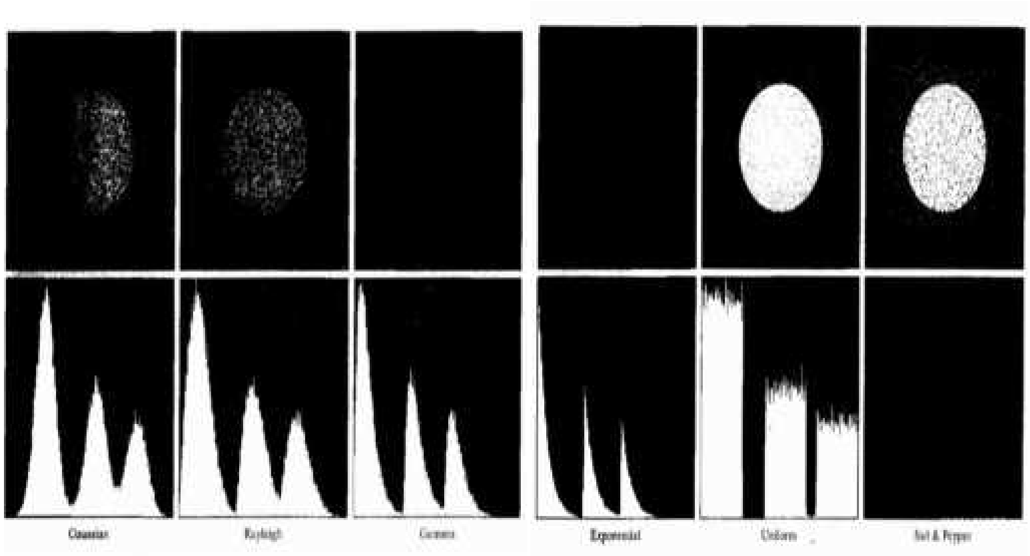
Impulse noise is found in situations where quick transients, such as faulty switching, take place during imageing, as mentioned in the previous paragraph.

The uniform density perhaps the least descriptive of

practical situation.however,the uniform density is quite useful as the basis of numerous random number generator that are used in



- Shows a test pattern well suited for illustrating the noise model is just discussed.
- This is suitable pattern to use because it is composed of simple, constant areas, that span the gray scale from black to near white in only three increments.
- This facilities visual analysis of the characteristics of the various noise compound added to the image.

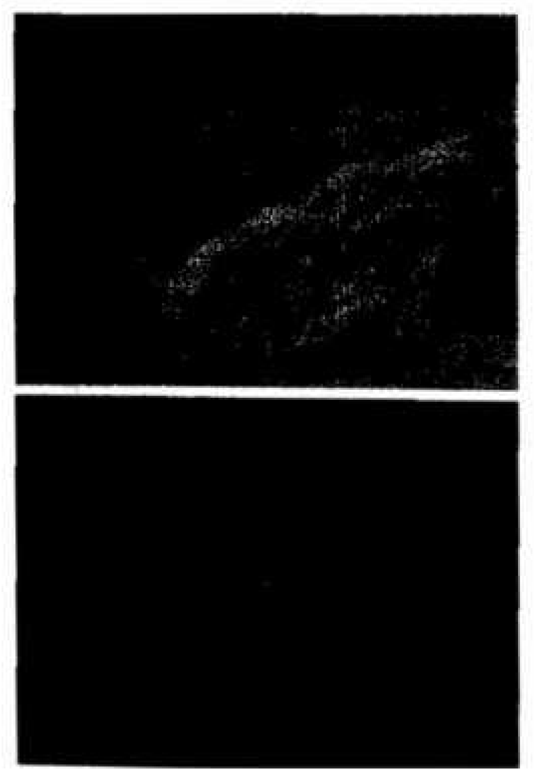
1. 

Shows the test pattern after addition of the six types of noise discussed thus far in this section.

2. Shown below each image is the histogram computed directly from the image.
3. The parameters of the noise were chosen in each case so that the histogram corresponding to the three intensity levels in the test pattern would start to merge.This made the noise quite visible.
4. Without obsuring the basis sructure of the underlying image.we see a close corresponding in comparing the histograms in diagrams
5. The histogram for the salt-and-pepper example has an extra peak at the white end of the intensity scale.
6. Because the noise components were pure black&white and the lightest component of the test pattern(circle) is light gray.

7. With the exceptiin of slightly different overall intensity, it is difficult to differentiate visually between the first five images even througth their histograms are significantly different.
8. The salt&pepper apperance of the images corrupted by impulse noise is the only one that is visually indicative of the type of noise causing the degradation.

Periodic Noise:

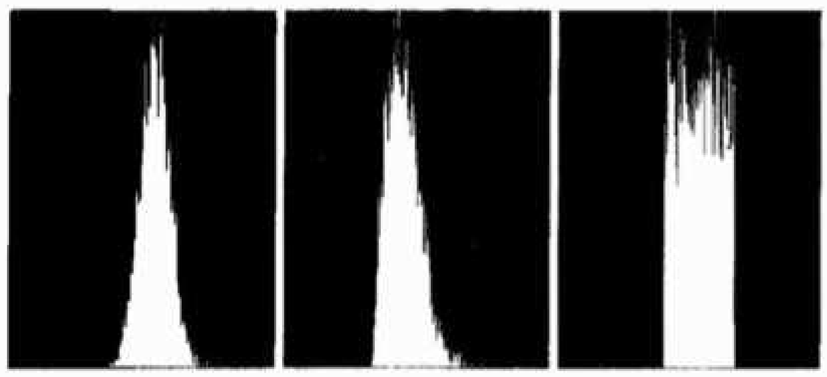
- Periodic noise in an image arises typically from electrical or electromechanical interference during image acquistion.
- This is the type of spatially dependent noise that will be considered in this chapter
- Periodic noise can be reduced significantly via frequency domain filtering.
- 

- Fig a) This image is severely corrupted by(spatial) sinusoidal noise of various frequencies.
- The fourier transform of a pure-sinusoid is a pair of conjugate impulses^{+} located at the conjugate frequencies of the sine wave.(table4.3)
- Thus,if the amplitude of a sine wave in the spatial domain is strong enough.
- Fig b) This is indeed the case, with the impulses appearing in an approximate circle.
- Because the frequency values in this particular case are so arranged.

Estimation of Noise Parameters:

- The parameters of periodic noise typically are estimated by inspection of the fourier spectrum of the image.
- As noted the preview section, periodic noise tends to produce frequency spikes that often can be detected even by visual analysis.
- Another approach is to attempt to infer the periodicity of noise components directly from the image.
- Automated analysis is possible in situation in which the noise spikes are either exceptionally pronouced (or) when knowledge is available about the general location of the frequency components of the interference.
- The parameters of noise PDF's may be known partially from sensor specification, but it is often necessary to estimate them for a particular imageing arrangement.
- If the imageing system is available, one simple way to study the characteristics of system noise is to capture a set of images of "flat" environment.
- For example., in case of an optical sensor, this is as simple as imageing a solid gray board that is illuminated uniformly.
- The resulting images typically are good indicates of system noise.
- When only images already generated by a sensor are availabe.

Frequently, it is possible to estimate the parameters of the PDF from small patches of reasonably constant background intensity.



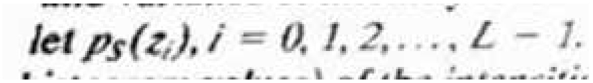
a b c

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

- For example., the vertical strips(150*20 pixels) in the figure were cropped from the gaussian,rayleigh and uniform images in figure
- Figure The ones in the middle of the group of three in figure d,e&k. We see that the shapes of these histgrams correspond quite closely to the shape of the histogram in figure
- Their heights are different due to scaling, but the shapes are unmistakably similar.
- The simplest use of the data from the image strips is for mean&varience

of intensity levels.

- Consider a strip(sub image) denoted by s



let $p_S(z_i),\ i = 0, 1, 2, \ldots, L-1.$

- Denote the probability estimates(normalized histogram values) of the intensitives of the pixels in s
- Where,L is the number of possible intensities in the entire image.

We estimate the mean and varience of the pixels in s as follows:

$$\bar{z} = \sum_{i=0}^{L-1} z_i p_S(z_i) \qquad (5.2\text{-}15)$$

$$\sigma^2 = \sum_{i=0}^{L-1} (z_i - \bar{z})^2 p_S(z_i) \qquad (5.2\text{-}16)$$

- The shape of the histogram identifies the closeset PDF match.
- If the shape is approximately gaussian, then the mean and varience are all we need.
- Because the gaussian PDF is completely specified by these two parameters.
- For the other,shaped discussed in fugure are use the mean & varience to solve for the parameters a and b.
- Impulse noise is handle differently because the estimate needed is of the actual probability of occurance of white&black pixels.
- Obtaining this estimate rquires that both black&white pixels be visible,so a midgray,relatively constant area is needed in the image in order to be able to compute a histogram.
- The heights of the peaks corresponding to black&white pixels are the estimate of p_a and p_b in

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

PERIODIC NOISE REDUCTION BY FREQUENCY DOMAIN FILTERING:

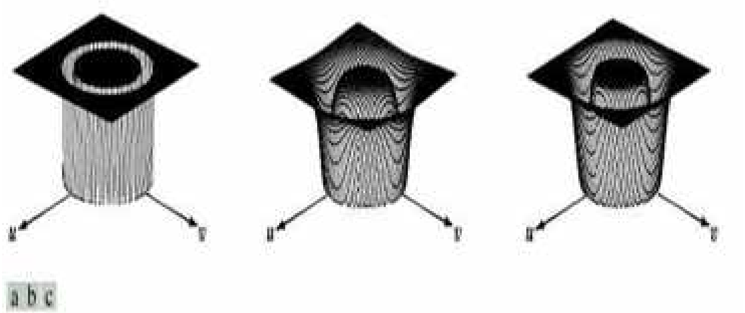
- Periodic noise can be analyzed and filtered quite effectively using frequency domain techniques. The basic idea is that periodic noise appears as concentrated bursts of energy in the Fourier transform, at locations corresponding to the frequencies of the periodic interference.

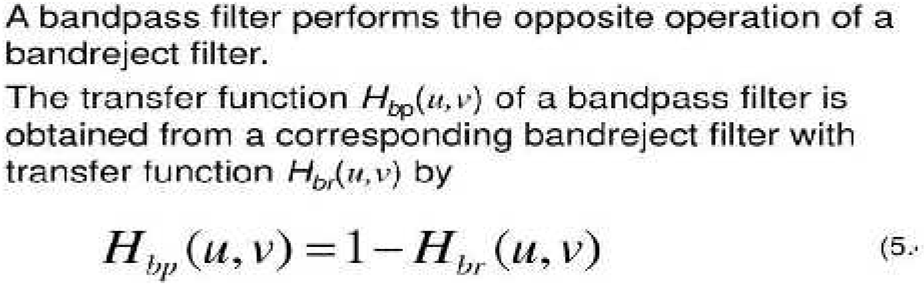
Frequency Domain Filtering:

- Bandreject filters
- Bandpass filters

- Notch filters

Bandreject Filters:
The transfer functions of ideal, Butterworth, and Gaussian band reject filters, are summarized in example illustrates using a band reject filter for reducing the effects of periodic noise.

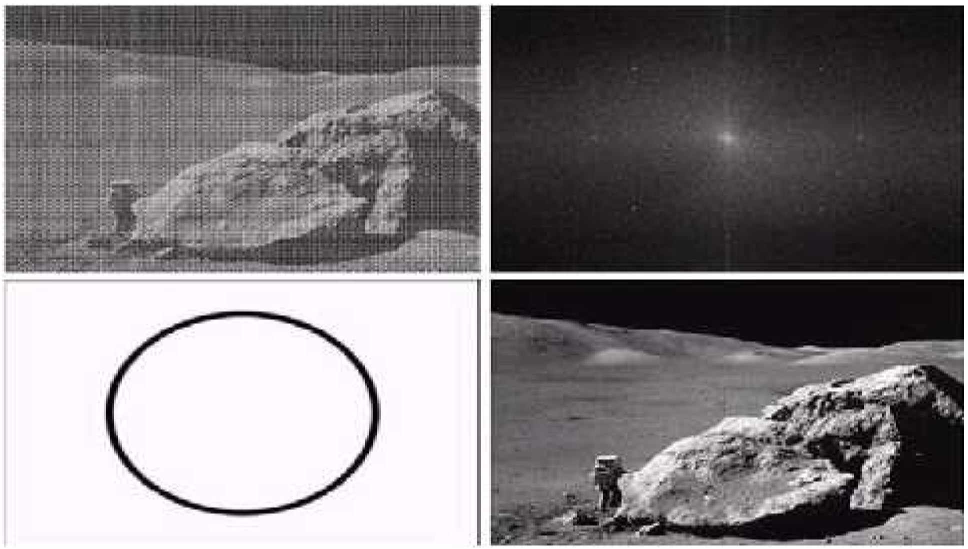




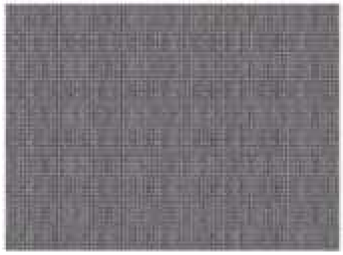
A bandpass filter performs the opposite operation of a bandreject filter.

The transfer function $H_{bp}(u,v)$ of a bandpass filter is obtained from a corresponding bandreject filter with transfer function $H_{br}(u,v)$ by

$$H_{bp}(u,v) = 1 - H_{br}(u,v) \qquad (5.$$

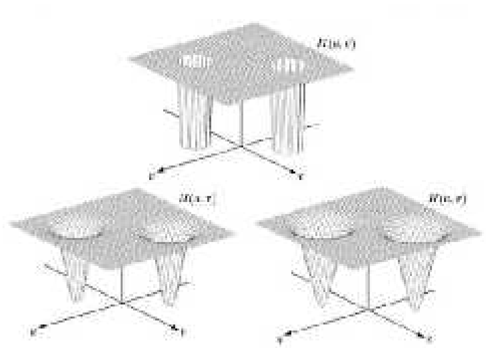


(a) Image corrupted by sinusoidal noise. (b) Spectrum of (a). (c) Butterworth band reject filter (white represents 1). (d) Result of filtering. (Original image courtesy of NASA.)


Noise pattern of the image band pass filtering

Notch Filters:
A notch filter rejects (or passes) frequencies in predefined neighborhoods about a center frequency. The one exception to this rule is if the notch filter is located at the origin, in which case it appears by itself. The shape of the notch areas also can be arbitrary (e.g., rectangular).

$$H_{\mathrm{NP}}(u, v) = 1 - H_{\mathrm{NR}}(u, v)$$

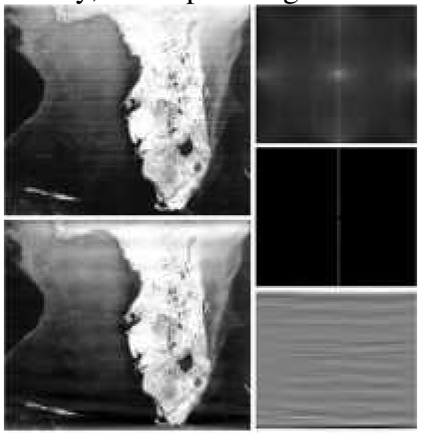


Perspective plots of (a) ideal, (b) Butterworth (of order 2), and (c) Gaussian notch (reject) filters

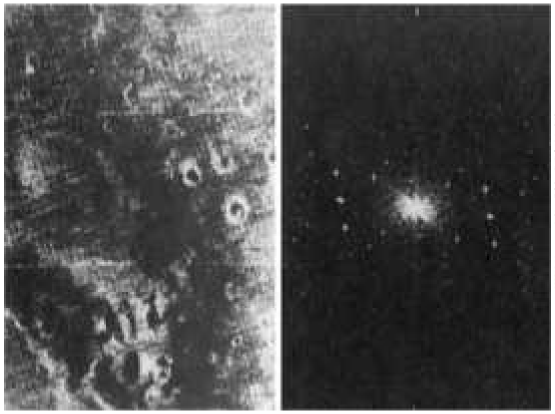
Optimum Notch Filtering:

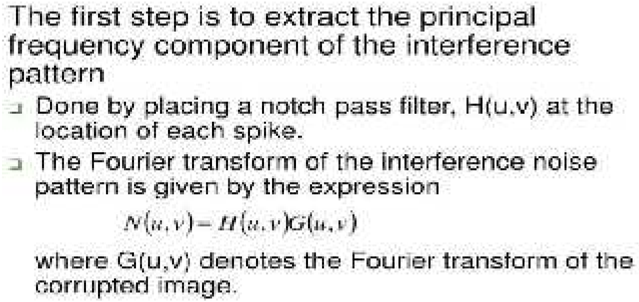
Then an adaptive **optimum notch filter** is proposed. In the proposed method, the regions of noise frequencies are determined by analyzing the spectral of noisy image. ...

Finally, an output image with reduced periodic noise is restored.



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





The first step is to extract the principal frequency component of the interference pattern

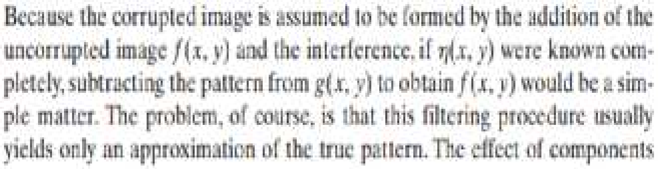
- Done by placing a notch pass filter, H(u,v) at the location of each spike.
- The Fourier transform of the interference noise pattern is given by the expression

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

where G(u,v) denotes the Fourier transform of the corrupted image.

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

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



Because the corrupted image is assumed to be formed by the addition of the uncorrupted image $f(x, y)$ and the interference, if $\eta(x, y)$ were known completely, subtracting the pattern from $g(x, y)$ to obtain $f(x, y)$ would be a simple matter. The problem, of course, is that this filtering procedure usually yields only an approximation of the true pattern. The effect of components

$$\hat{f}(x, y) = g(x, y) - w(x, y)\eta(x, y) \tag{5.4-5}$$

where, as before, $\hat{f}(x, y)$ is the estimate of $f(x, y)$ and $w(x, y)$ is to be determined. The function $w(x, y)$ is called a *weighting* or *modulation* function, and the objective of the procedure is to select this function so that the result is optimized in some meaningful way. One approach is to select $w(x, y)$ so that the variance of the estimate $\hat{f}(x, y)$ is minimized over a specified neighborhood of every point (x, y).

Consider a neighborhood of size $(2a + 1)$ by $(2b + 1)$ about a point (x, y). The "local" variance of $\hat{f}(x, y)$ at coordinates (x, y) can be estimated from the samples, as follows:

where $\overline{\hat{f}}(x, y)$ is the average value of $\hat{f}$ in the neighborhood; that is,

$$\overline{\hat{f}}(x, y) = \frac{1}{(2a + 1)(2b + 1)} \sum_{s=-a}^{a} \sum_{t=-b}^{b} \hat{f}(x + s, y + t) \tag{5.4-7}$$

Points on or near the edge of the image can be treated by considering partial neighborhoods or by padding the border with 0s.

Substituting Eq. (5.4-5) into Eq. (5.4-6) yields

$$\begin{aligned}\sigma^2(x, y) = \frac{1}{(2a + 1)(2b + 1)} \sum_{s=-a}^{a} \sum_{t=-b}^{b} \{[g(x + s, y + t) \\ - w(x + s, y + t)\eta(x + s, y + t)] \\ - [\overline{g}(x, y) - \overline{w(x, y)\eta(x, y)}]\}^2\end{aligned} \tag{5.4-8}$$

Assuming that $w(x, y)$ remains essentially constant over the neighborhood gives the approximation

$$w(x + s, y + t) = w(x, y) \tag{5.4-9}$$

for $-a \leq s \leq a$ and $-b \leq t \leq b$. This assumption also results in the expression

$$\overline{w(x, y)\eta(x, y)} = w(x, y)\overline{\eta}(x, y) \tag{5.4-10}$$

in the neighborhood. With these approximations, Eq. (5.4-8) becomes

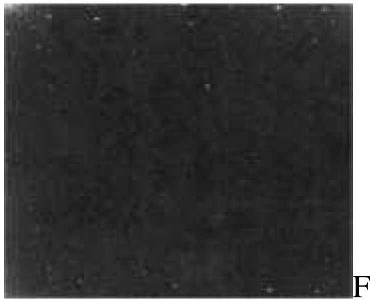
$$\begin{aligned}\sigma^2(x, y) = \frac{1}{(2a + 1)(2b + 1)} \sum_{s=-a}^{a} \sum_{t=-b}^{b} \{ & [g(x + s, y + t) \\ & - w(x, y)\eta(x + s, y + t)] \\ & - [\overline{g}(x, y) - w(x, y)\overline{\eta}(x, y)]\}^2\end{aligned} \tag{5.4-11}$$

To minimize $\sigma^2(x, y)$, we solve

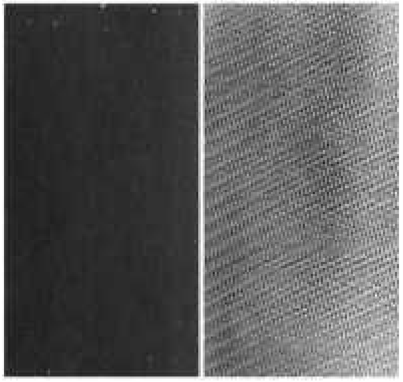
$$\frac{\partial\sigma^2(x, y)}{\partial w(x, y)} = 0$$

for $w(x, y)$. The result is

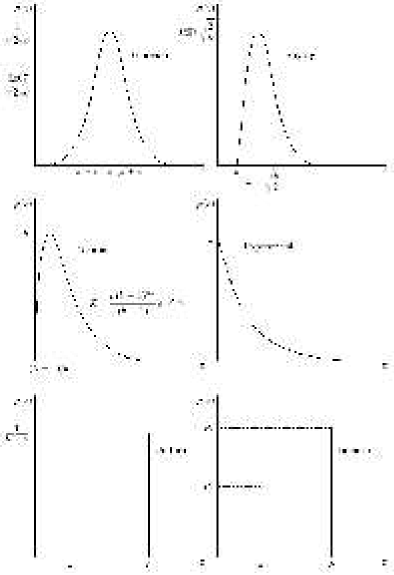
$$w(x, y) = \frac{\overline{g(x, y)\eta(x, y)} - \overline{g}(x, y)\overline{\eta}(x, y)}{\overline{\eta^2}(x, y) - \overline{\eta}^2(x, y)}$$



Fourier spectrum (without shifting) of the image (Courtesy of NASA)



(a) Fourier spectrum of $N(u, v)$, and (b) corresponding noise interference pattern (Courtesy of NASA.) $h(x, y)$.



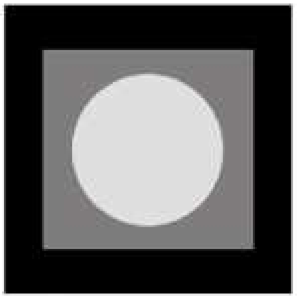
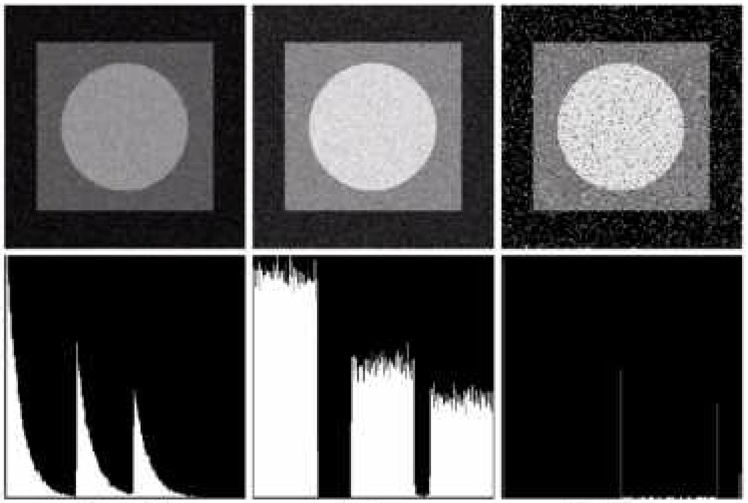


FIGURE 5.3 Test pattern used to illustrate the characteristics of the noise PDFs shown in Fig. 5.2.



Exponential

Uniform

Salt & Pepper

g h i
j k l

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

The size of typical still image (1200x1600)

$$1200 \times 1600 \times 3 byte = 5760000 byte$$

$$= 5{,}760 Kbyte = 5.76 Mbyte$$

The size of two hours standard television (720x480) movies

$$30 \frac{frame}{\sec} \times (760 \times 480) \frac{pixels}{frame} \times 3 \frac{bytes}{pixel} = 31{,}104{,}000 bytes/\sec$$

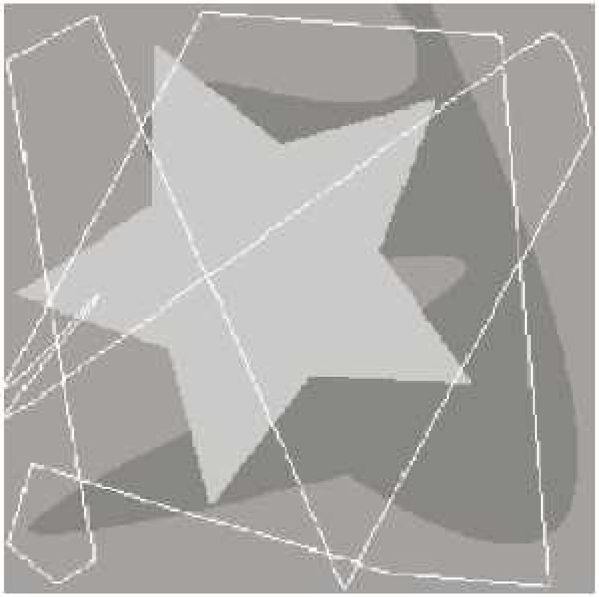
$$31{,}104{,}000 \times \frac{bytes}{\sec} \times (60 \times 60) \frac{\sec}{hour} \times 2 hours = 2.24 \times 10^{11} bytes$$

$$= 224 GByte.$$

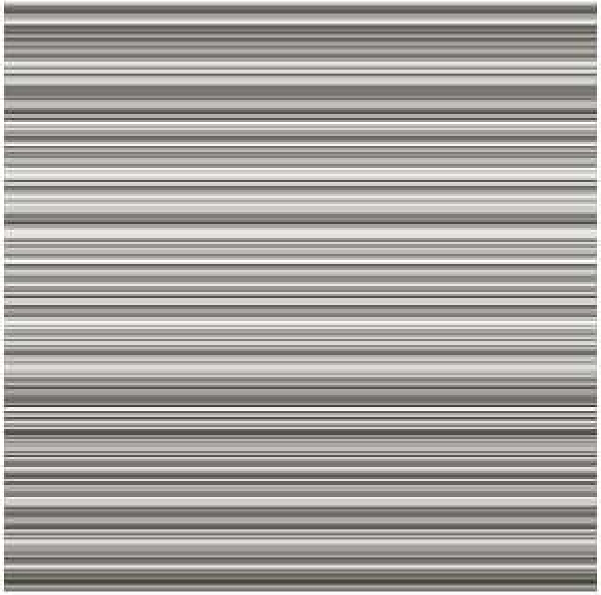
Data, Information, and Redundancy

- **Information**
- **Data** is used to represent information
- **Redundancy** in data representation of an information provides no relevant information or repeats a stated information
- Let n1, and n2 are data represents the same information. Then, the relative data redundancy R of the n1 is defined as
  R = 1 – 1/C where C = n1/n2

- Redundancy in Digital Images
  - Coding redundancy
    usually appear as results of the uniform representation of each pixel
  - Spatial/Temopral redundancy
    because the adjacent pixels tend to have similarity in practical.
  - Irrelevant Information
    Image contain information which are ignored by the human visual system.



Coding Redundancy



Spatial Redundancy

Irrelevant Information

Coding Redundancy

- Assume the discrete random variable for r_k in the interval [0,1] that represent the gray levels. Each r_k occurs with probability p_k

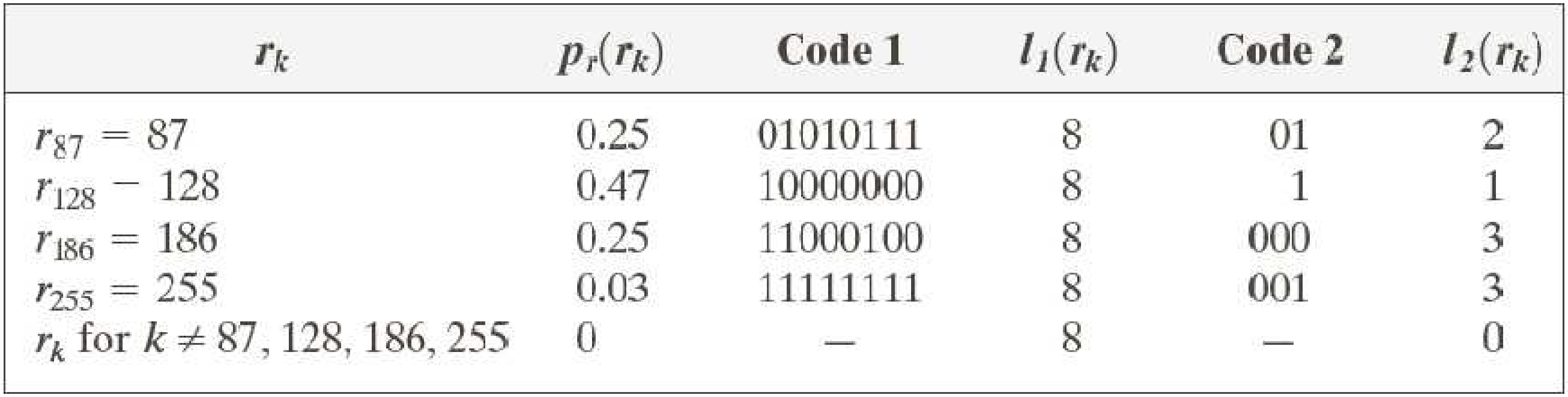
- If the number of bits used to represent each value of r_k by

  $l(r_k)$ then $$L_{avg} = \sum_{k=0}^{L-1} l(r_k)\, p(r_k)$$

- The average code bits assigned to the gray level values.

- The length of the code should be inverse proportional to its probability (occurrence).

Examples of variable length encoding



r_k	$p_r(r_k)$	Code 1	$l_1(r_k)$	Code 2	$l_2(r_k)$
$r_{87} = 87$	0.25	01010111	8	01	2
$r_{128} - 128$	0.47	10000000	8	1	1
$r_{186} = 186$	0.25	11000100	8	000	3
$r_{255} = 255$	0.03	11111111	8	001	3
r_k for $k \neq 87, 128, 186, 255$	0	—	8	—	0

Spatial/Temopral Redundancy

- Internal Correlation between the pixel result from

  Respective Autocorrelation

  - Structural Relationship

  - Geometric Relation ship

- The value of a pixel can be reasonablypredicted from the values of its neighbors.
- To reduce the inter-pixel redundancies in animage the 2D array is transformed (*mapped*) into more efficient format (Frequency Domain etc.)

Irrelevant information and Psycho-Visual Redundancy

- The brightness of a region depend on other factors that the light reflection
- The perceived intensity of the eye is limited an non linear
- Certain information has less relative importance that other information in normal visual processing
- In general, observer searches for distinguishing features such as edges and textural regions.

Measuring Information

- A random even E that occurs with probability P(E) is said to contain I(E) information where I(E) is defined as I(E) = log(1/P(E)) = -log(P(E))
- P(E) = 1 contain no information
- P(E) = ½ requires one bit of information.

Measuring Information

- For a source of events a_0, a_1,a_2, .., a_k with

  associated probability P(a_0), P(a_1), P(a_2), .., P(a_k).
- The average information per source (entropy) is

$$H = -\sum_{j=0}^{k} P(a_j)\log(P(a_j))$$

For image, we use the normalized histogram to generate the source probability, which leas to the entropy

$$\tilde{H} = -\sum_{i=0}^{L-1} p_r(r_i)\log(p_r(r_i))$$

Fidelity Criteria

- Objective Fidelity Criteria

  - The information loss can be expressed as a function of the encoded and decoded images.

  - For image I (x,y) and its decoded approximation I'(x,y)

  - For any value of x and y, the error e(x,y) could be defined as

    $e(x, y)$ □ $I'(x,$ $y)$ □ $I(x,y)$

  - For the entire Image

    M □1 N □1

    $I(x,y)$

    □□ $I'(x, y)$ □

    x□0 y □0

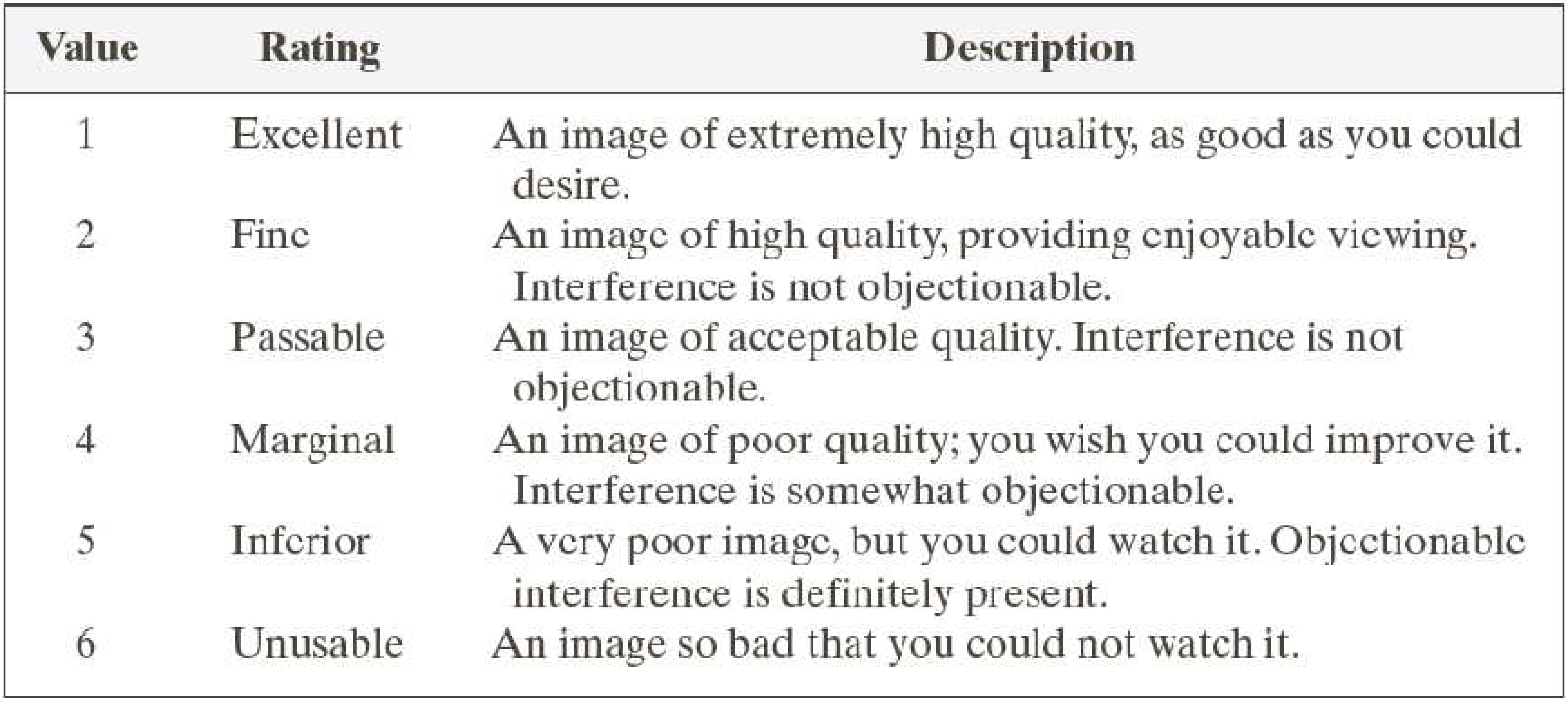
Fidelity Criteria

- The mean-square-error, e$_{rms}$ is

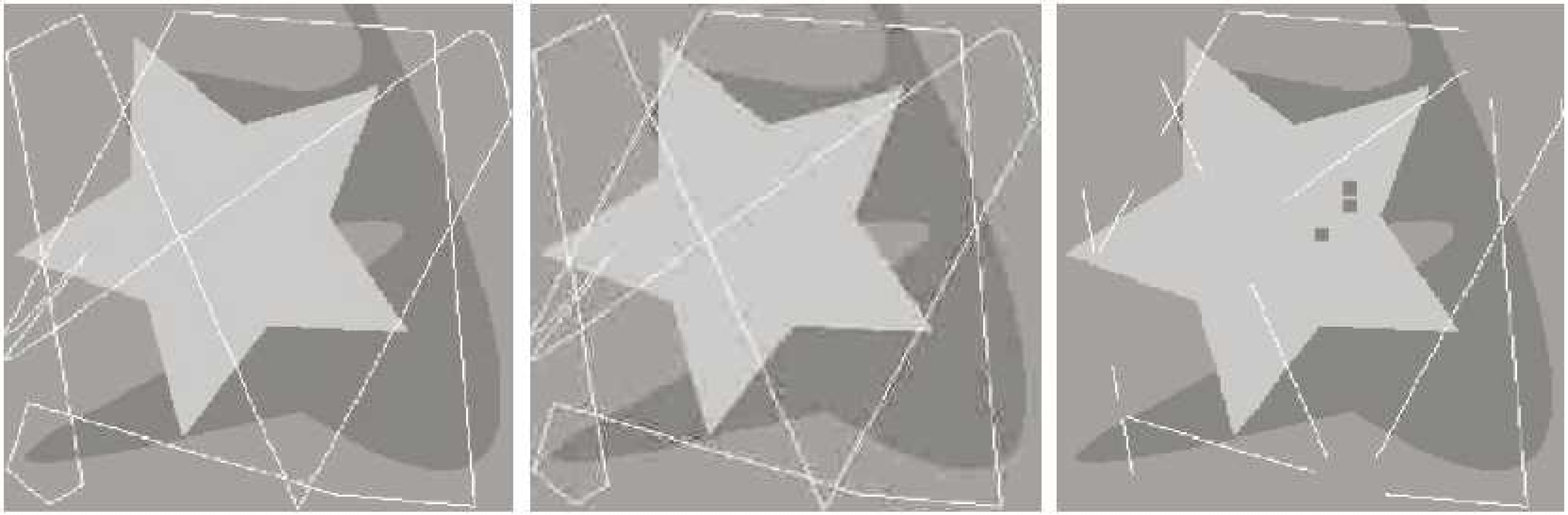
$$e_{rms} = \sqrt{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left[I'(x, y) - I(x, y)\right]^2}$$

The mean-square-error signal-to-noise ratio SNR$_{ms}$ is

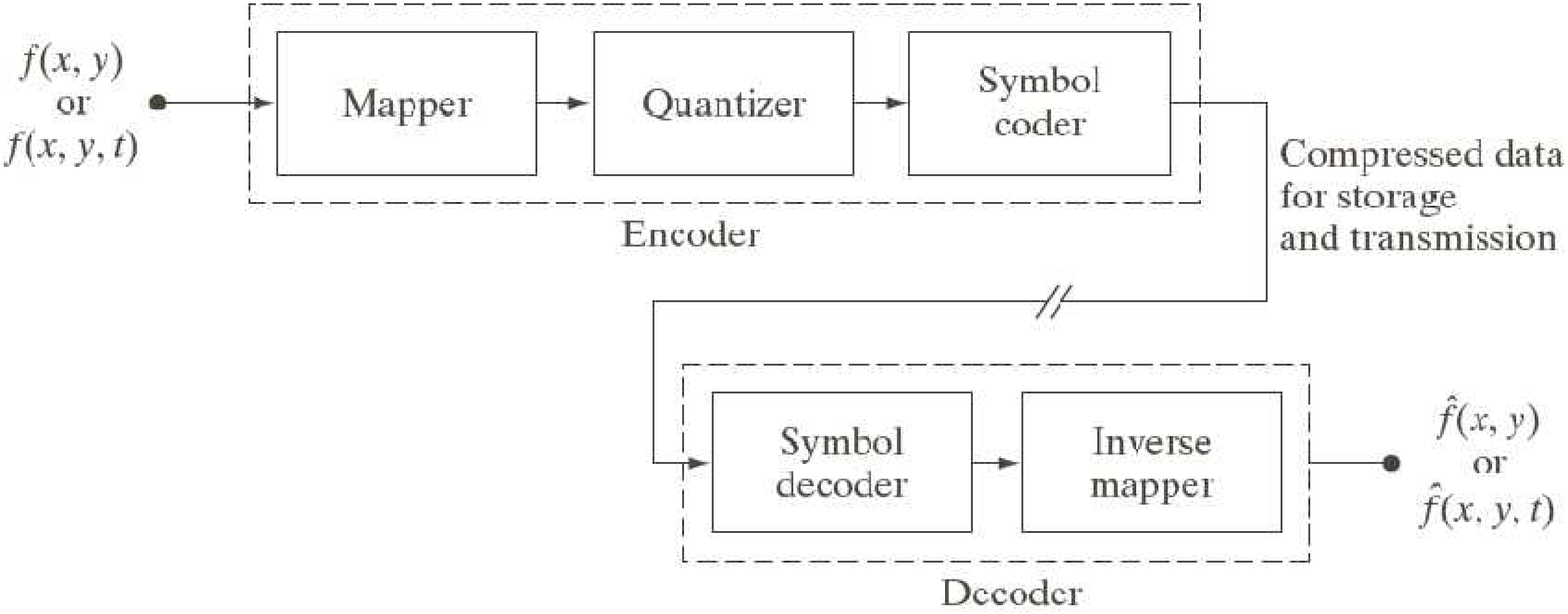
$$SNR_{ms} = \frac{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} I'(x, y)^2}{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left[I'(x, y) - I(x, y)\right]^2}$$



Value	Rating	Description
1	Excellent	An image of extremely high quality, as good as you could desire.
2	Finc	An imagc of high quality, providing cnjoyablc vicwing. Interference is not objectionable.
3	Passable	An image of acceptable quality. Interference is not objectionable.
4	Marginal	An image of poor quality; you wish you could improve it. Interference is somewhat objectionable.
5	Infcrior	A vcry poor imagc, but you could watch it. Objcctionablc interference is definitely present.
6	Unusable	An image so bad that you could not watch it.



Three approximations of the same image


$f(x, y)$
or
$f(x, y, t)$
Mapper
Quantizer
Symbol coder
Encoder
Compressed data for storage and transmission
Symbol decoder
Inverse mapper
Decoder
$\hat{f}(x, y)$
or
$\hat{f}(x, y, t)$

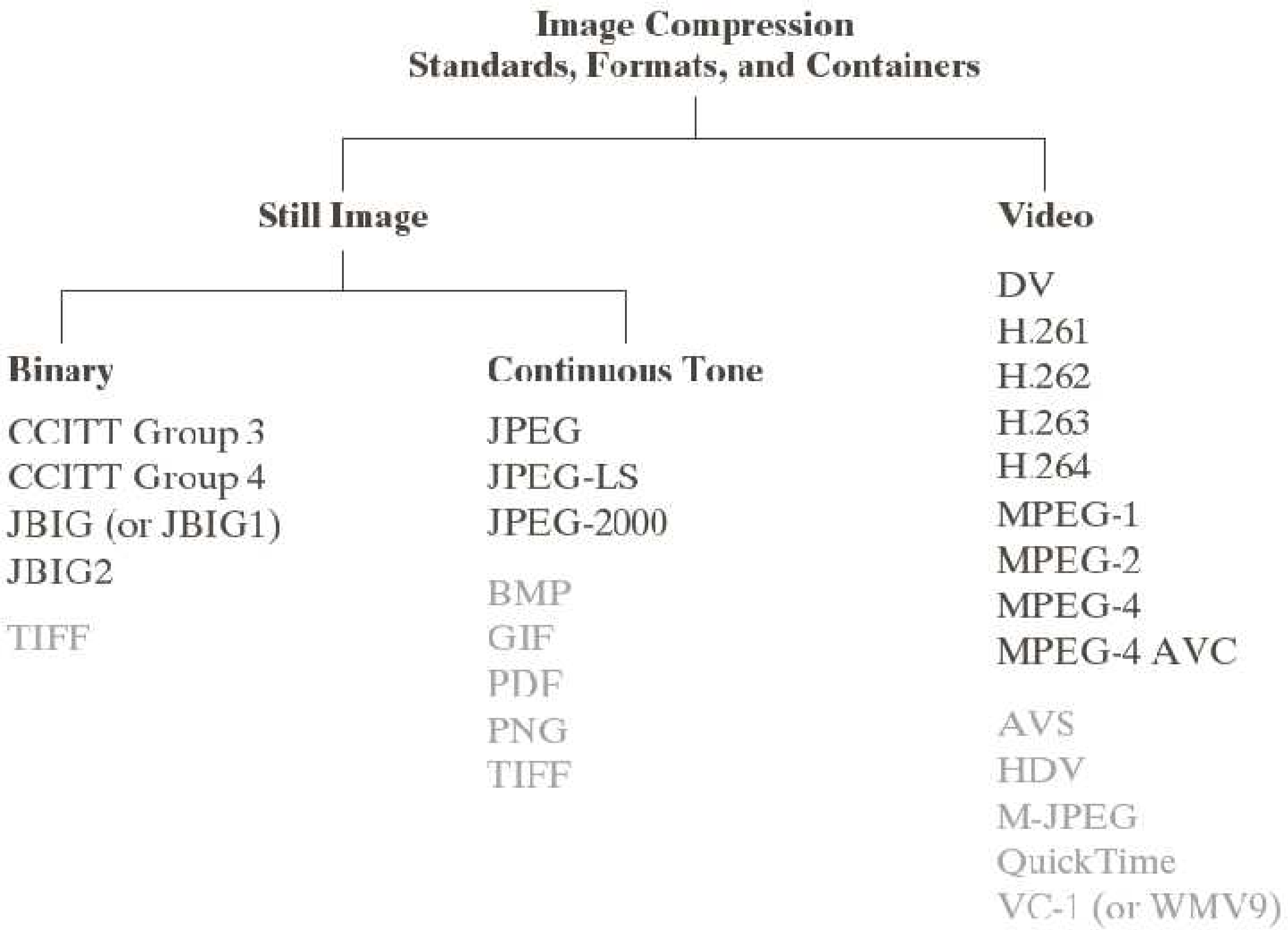


Image Compression
Standards, Formats, and Containers

Still Image

Binary

CCITT Group 3
CCITT Group 4
JBIG (or JBIG1)
JBIG2

TIFF

Continuous Tone

JPEG
JPEG-LS
JPEG-2000

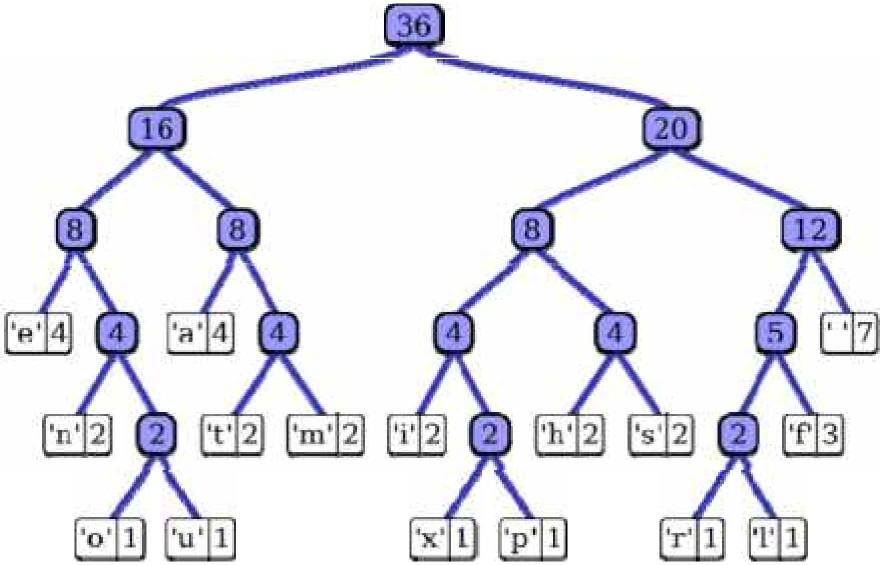
BMP
GIF
PDF
PNG
TIFF

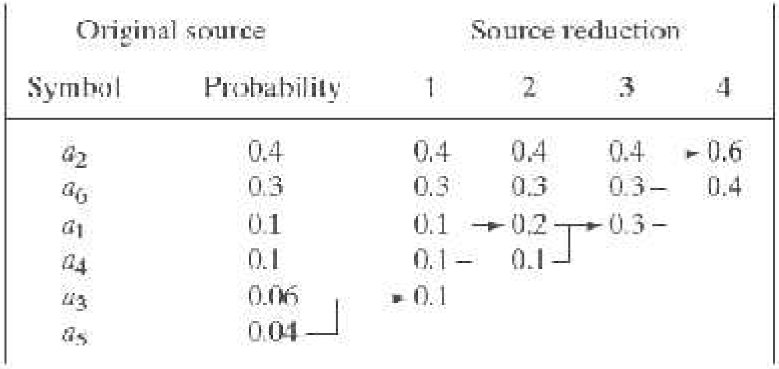
Video

DV
H.261
H.262
H.263
H.264
MPEG-1
MPEG-2
MPEG-4
MPEG-4 AVC

AVS
HDV
M-JPEG
QuickTime
VC-1 (or WMV9)

Huffman coding is an entropy encoding algorithm used for lossless data compression. The term refers to the use of a variable- length code table for encoding a source symbol (such as a character in a file) where the variable-length code table has been derived in a particular way based on the estimated probability of occurrence for each possible value of the source symbol.

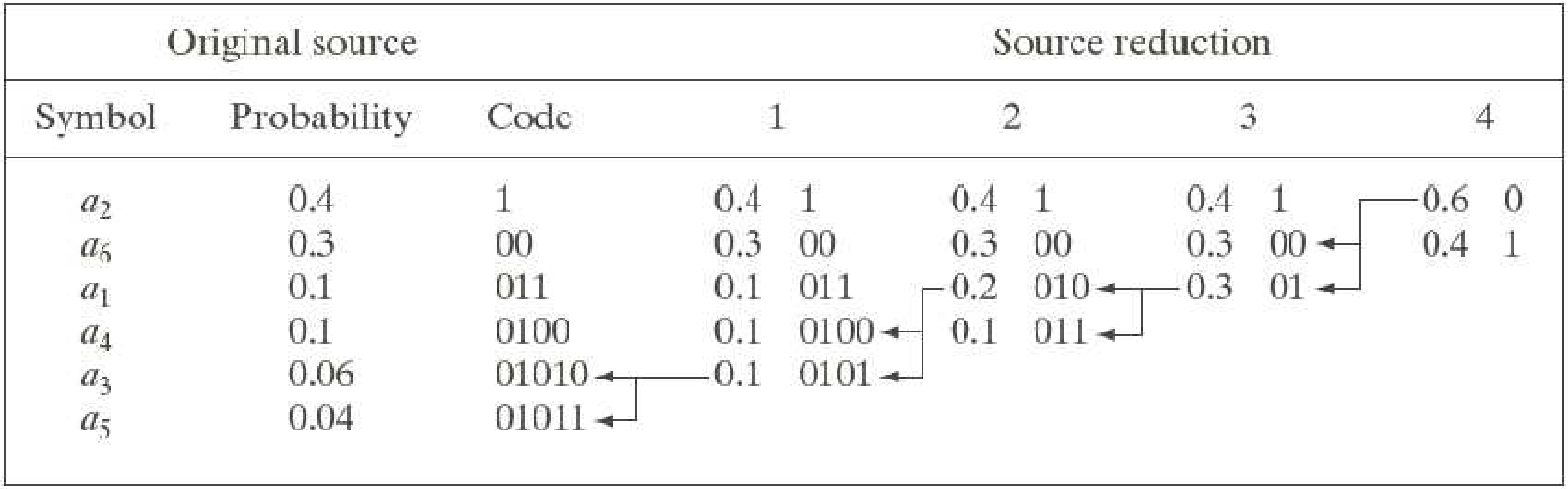




Original source		Source reduction			
Symbol	Probability	1	2	3	4
a_2	0.4	0.4	0.4	0.4	0.6
a_6	0.3	0.3	0.3	0.3	0.4
a_1	0.1	0.1	0.2	0.3	
a_4	0.1	0.1	0.1		
a_3	0.06	0.1			
a_5	0.04				

Huffman coding

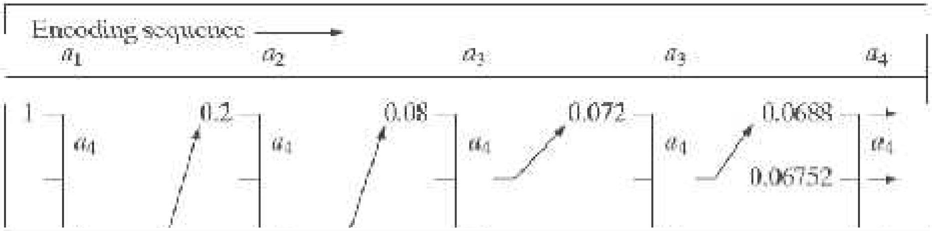
Assignment procedure

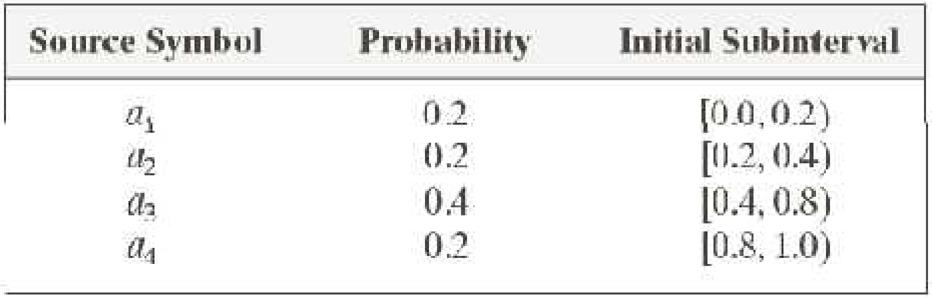


Original source			Source reduction							
Symbol	Probability	Codc	1		2		3		4	
a_2	0.4	1	0.4	1	0.4	1	0.4	1	0.6	0
a_6	0.3	00	0.3	00	0.3	00	0.3	00	0.4	1
a_1	0.1	011	0.1	011	0.2	010	0.3	01		
a_4	0.1	0100	0.1	0100	0.1	011				
a_3	0.06	01010	0.1	0101						
a_5	0.04	01011								

Arithmetic coding is a form of variable-length entropy encoding. A string is converted to arithmetic encoding, usually characters are stored with fewer bits

Arithmetic coding encodes the entire message into a single number, a fraction *n* where (0.0 $\leq n < 1.0$).





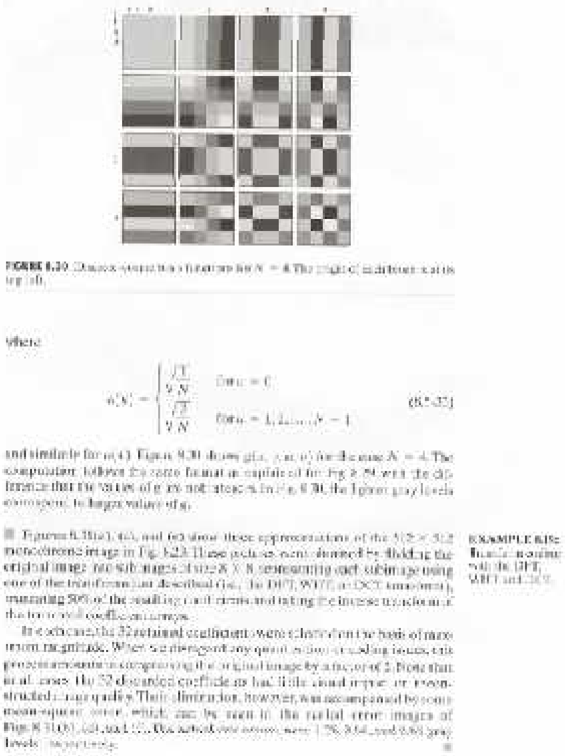
Source Symbol	Probability	Initial Subinterval
a_1	0.2	[0.0, 0.2)
a_2	0.2	[0.2, 0.4)
a_3	0.4	[0.4, 0.8)
a_4	0.2	[0.8, 1.0)

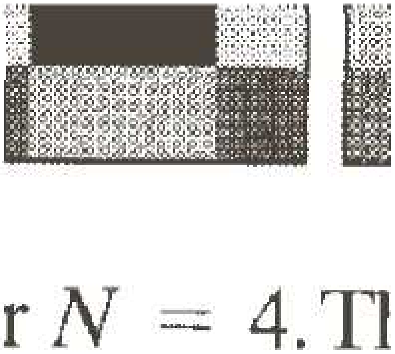
Compression Algorithms

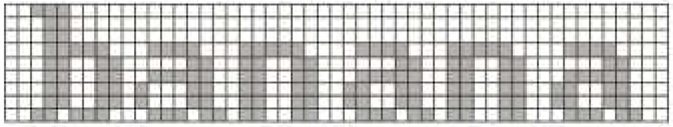
Symbol compression

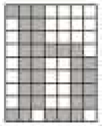
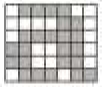
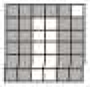
This approaches determine a set of symbols that constitute the image, and take advantage of their multiple appearance. It convert each symbol into token, generate a token table and represent the compressed image as a list of tokens.

This approach is good for document images.







Token	Symbol
0	
1	
2	

Triplet
(0, 2, 0)
(3, 10, 1)
(3, 18, 2)
(3, 26, 1)
(3, 34, 2)
(3, 42, 1)

a b c

FIGURE 8.17 (a) A bi-level document, (b) symbol dictionary, and (c) the triplets used to locate the symbols in the document.

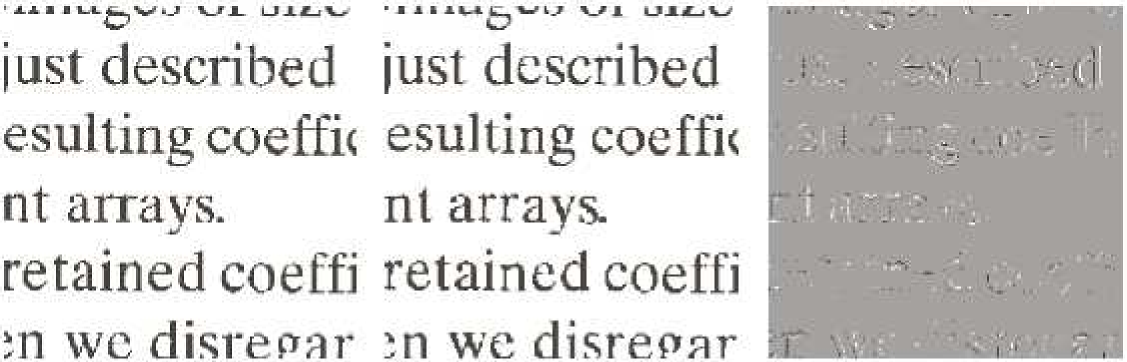
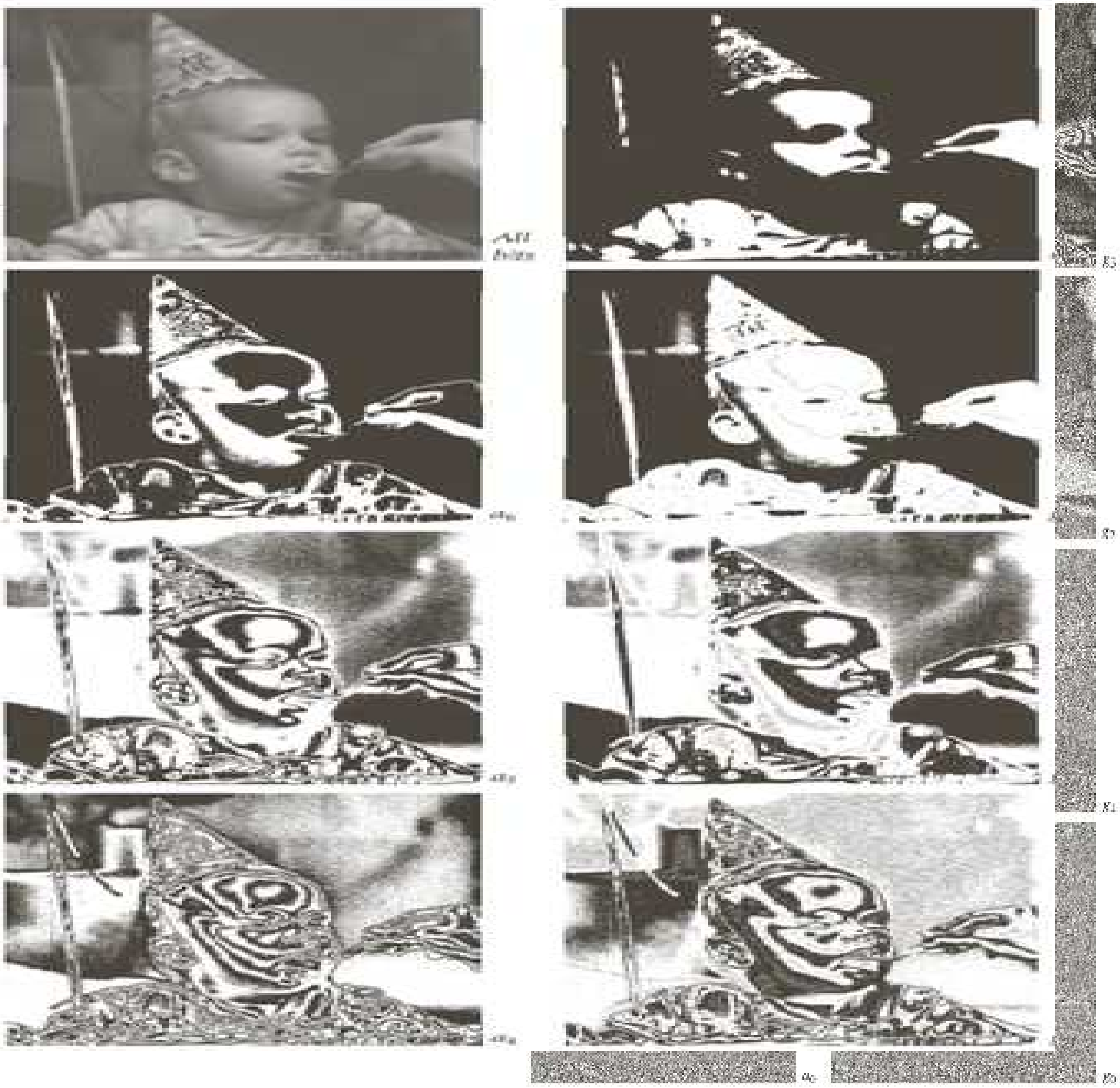
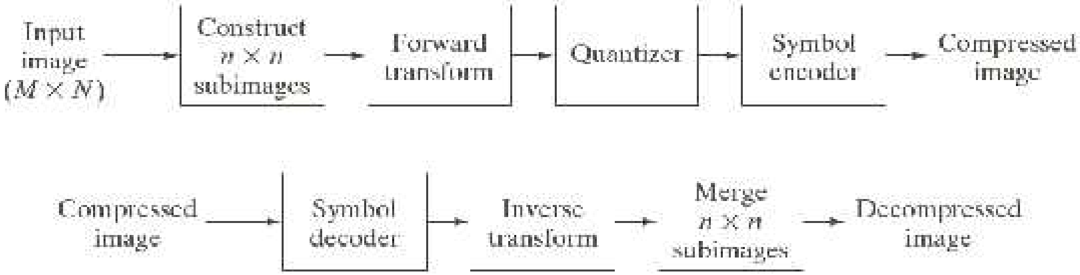


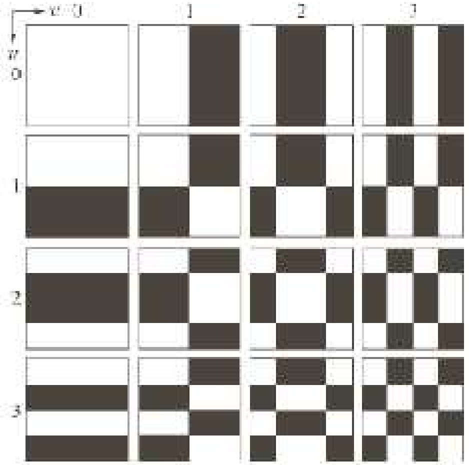
FIGURE 8.18 JBIG2 compression comparison: (a) lossless compression and reconstruction; (b) perceptually lossless; and (c) the scaled difference between the two.


g3
g2
g1
a0
g0



a
b

FIGURE 8.21 A block transform coding system: (a) encoder; (b) decoder.



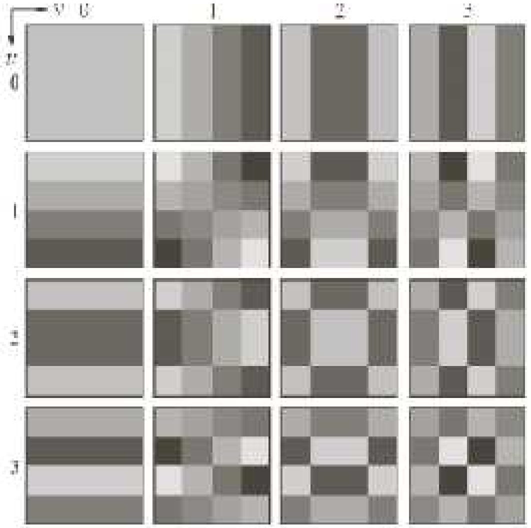


FIGURE 8.23 Discrete-cosine basis functions for $n - 4$. The origin of each block is at its top left.

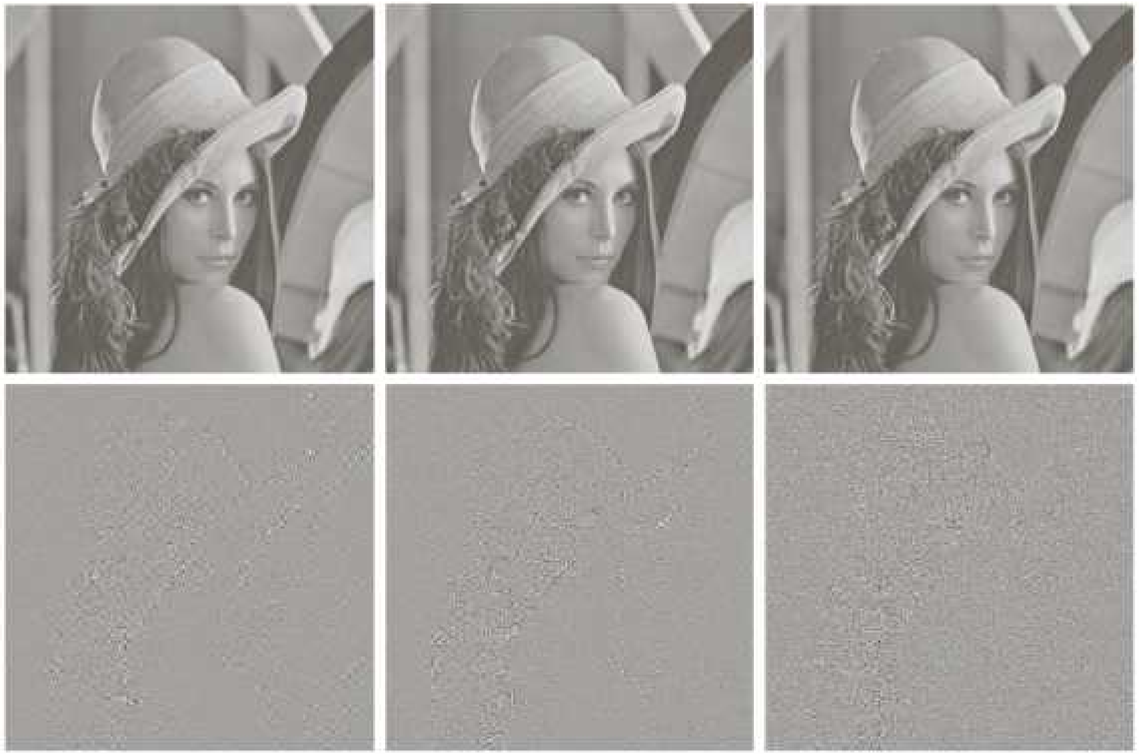
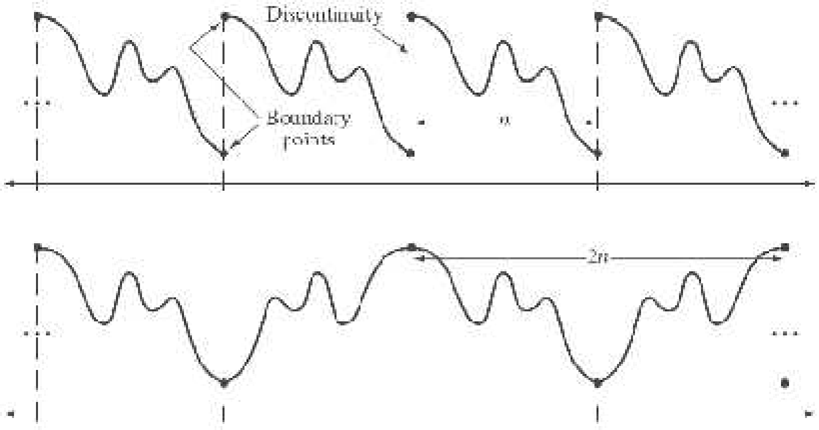


FIGURE 8.24 Approximations of Fig. 8.9(a) using the (a) Fourier, (b) Walsh-Hadamard, and (c) cosine transforms, together with the corresponding scaled error images in (d)–(f).

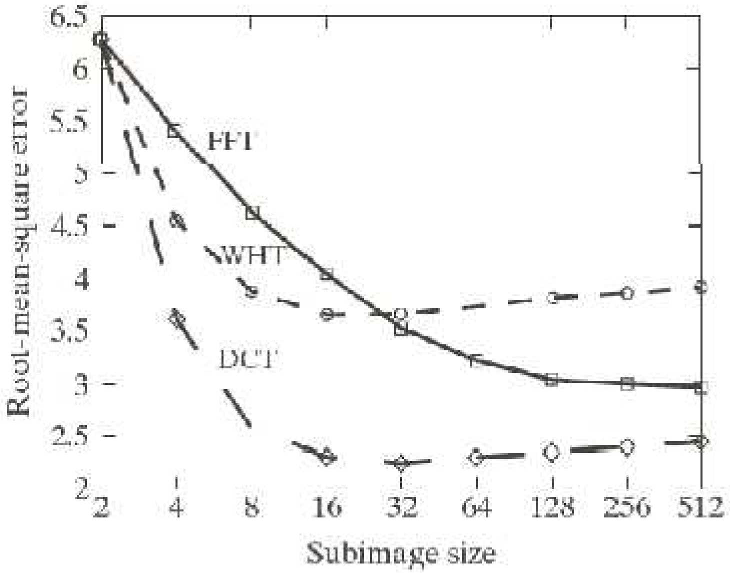
DFT and DCT

The periodicity implicit in the 1-D DFT and DCT. The DCT provide better continuity that the general DFT.



Bock Size vs. Reconstruction Error

The DCT provide the least error at almost any sub-image size.
The error takes its minimum at sub-images of sizes between 16 and 32.



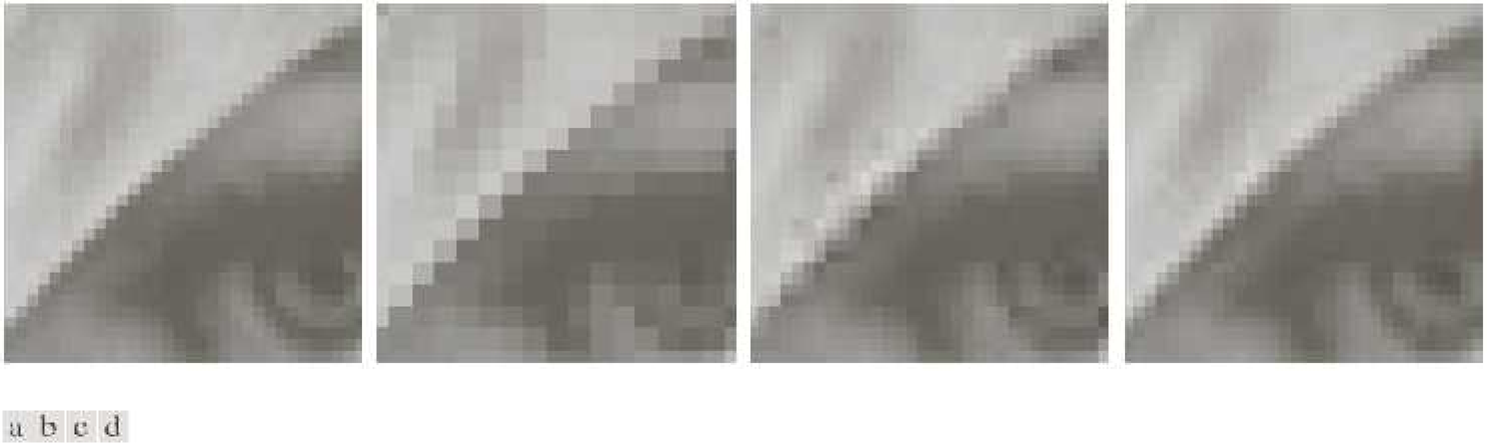


FIGURE 8.27 Approximations of Fig. 8.27(a) using 25% of the DCT coefficients and (b) 2 × 2 subimages, (c) 4 × 4 subimages, and (d) 8 × 8 subimages. The original image in (a) is a zoomed section of Fig. 8.9(a).

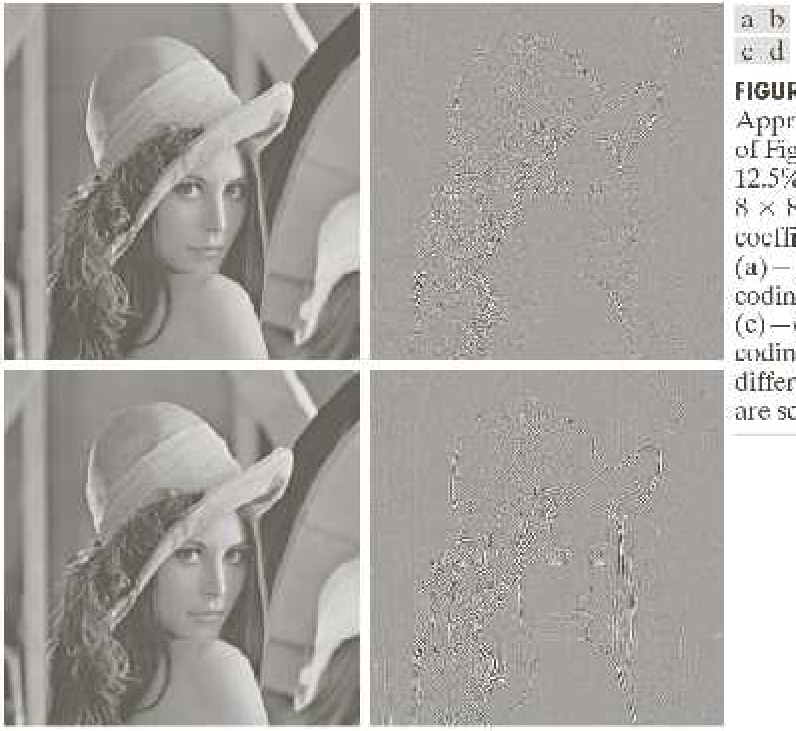


FIGURE 8.28 Approximations of Fig. 8.9(a) using 12.5% of the 8 × 8 DCT coefficients: (a)—(b) threshold coding results; (c)—(d) zonal coding results. The difference images are scaled by 4.

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

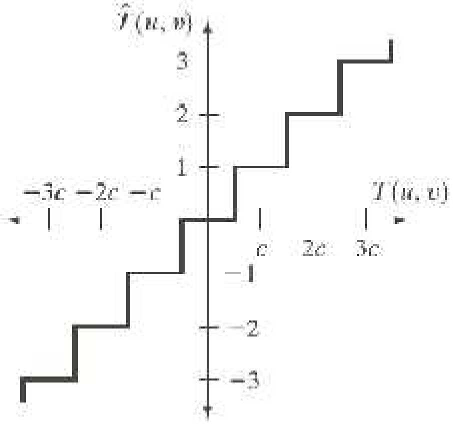
8	7	6	4	3	2	1	0
7	6	5	4	3	2	1	0
6	5	4	3	3	1	1	0
4	4	3	3	2	1	0	0
3	3	3	2	1	1	0	0
2	2	1	1	1	0	0	0
1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0

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

0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

a b
c d

FIGURE 8.29 A typical (a) zonal mask, (b) zonal bit allocation, (c) threshold mask, and (d) thresholded coefficient ordering sequence. Shading highlights the coefficients that are retained.



16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

a b

FIGURE 8.30 (a) A threshold coding quantization curve [see Eq. (8.2-29)]. (b) A typical normalization matrix.

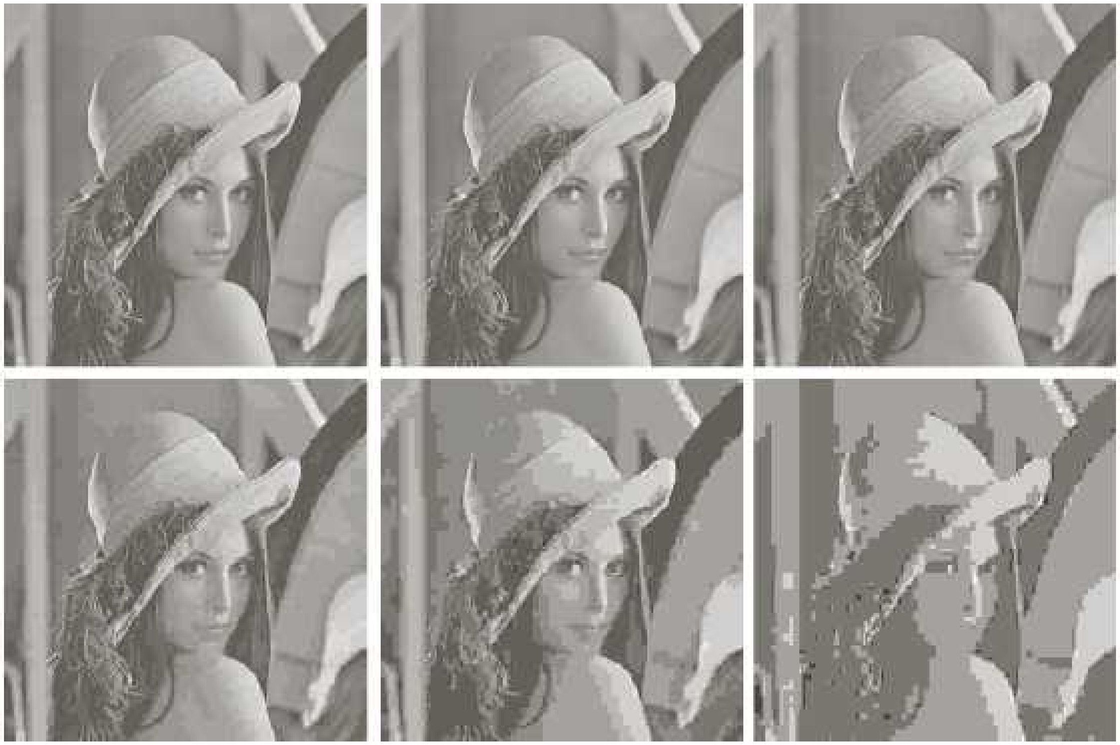


FIGURE 8.31 Approximations of Fig. 8.9(a) using the DCT and normalization array of Fig. 8.30(b): (a) **Z**, (b) 2**Z**, (c) 4**Z**, (d) 8**Z**, (e) 16**Z**, and (f) 32**Z**.

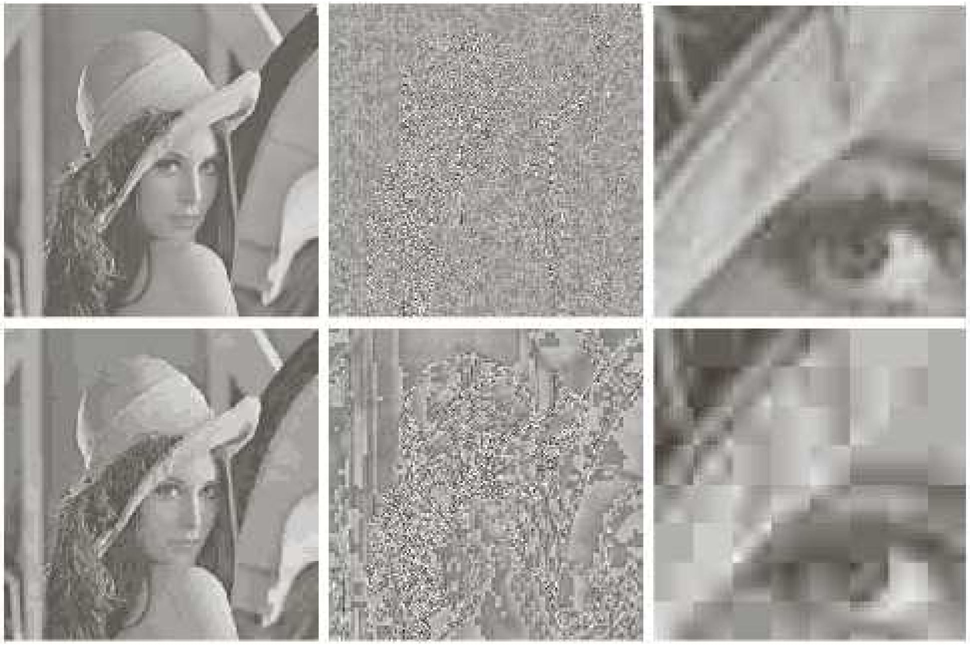
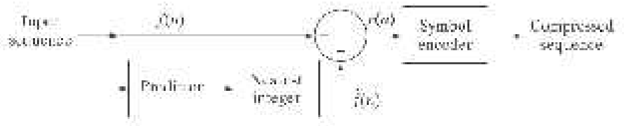
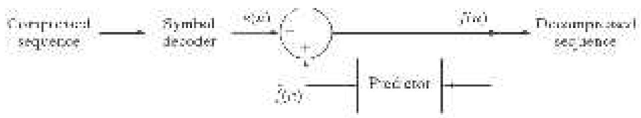


FIGURE 8.32 Two JPEG approximations of Fig. 8.9(a). Each row contains a result after compression and reconstruction, the scaled difference between the result and the original image, and a zoomed portion of the reconstructed image.





Lossless Predictive coding

The encoder expects a discrete sampl of a signal f(n).

es

1. A predictor is applied and its output is rounded to the nearest integer. $\hat{f}(n)$
2. The error is estimated as

   $e(n)$ □ $f(n)$ □ $\hat{f}(n)$

3. The compressed stream consist of first sample and the errors, encoded using variable length coding

The decoder uses the predictor and the error stream to reconstructs the original signal f(n).

1. The predictor is initialized using the first sample.
2. The received error is added to predictor result.

$f(n)$ □ $\hat{f}(n)$ □$e(n)$

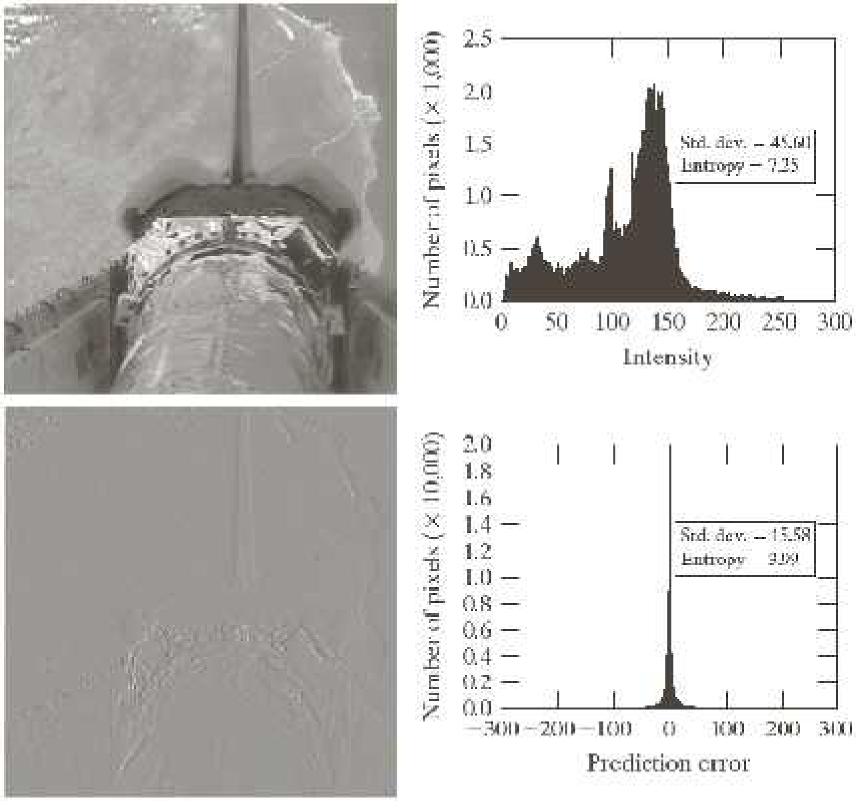
Lossless Predictive coding

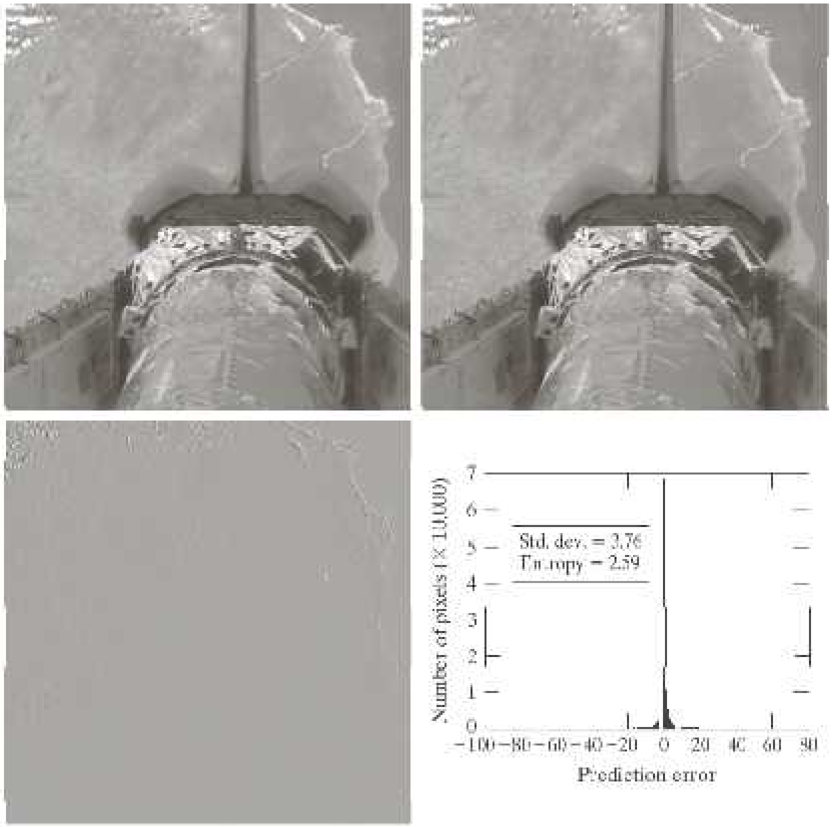
Linear predictors usually have the form:

$$\hat{f}(n) = round\left[\sum_{i=0}^{m} a_i f(n-i)\right]$$

Original Image (view of the earth). The prediction error and its histogram.

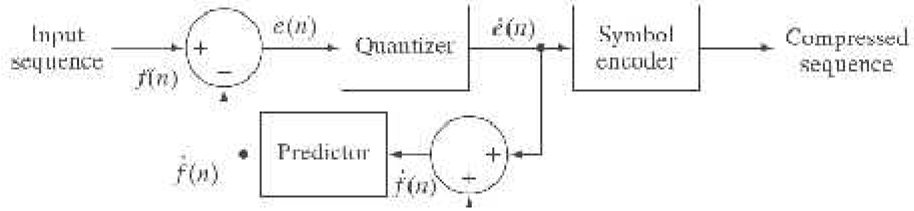
1. The error is small in uniform regions
2. Large close to edges and sharp changes in pixel intensities

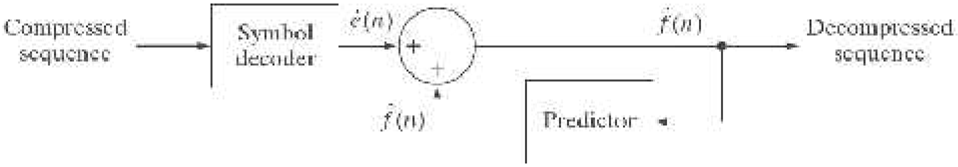



Number of pixels (× 10,000)
7
6
5
4
3
2
1
0
Std. dev. = 3.76
Entropy = 2.59
−100 −80 −60 −40 −20 0 20 40 60 80
Prediction error

Lossy Predictive coding *The encoder expects a discrete samples of a signal f(n).*

1. A predictor is applied and its output is rounded to the nearest integer, $\hat{f}(n)$
2. The error is mapped into limited rage of values (quantized) e□(n)
3. The compressed stream consist of first sample and the mapped errors, encoded using variable length coding





Lossy Predictive coding

The decoder uses error stream to reconstructs an approximatic the

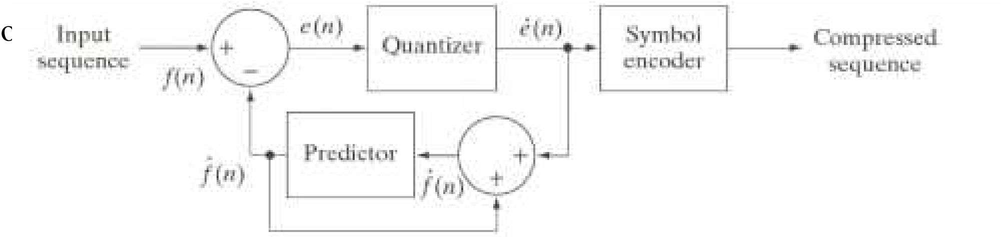
original signal, $\dot{f}(n)$

1. The predictor is initialized using first sample.
2. The received error is added to predictor result.

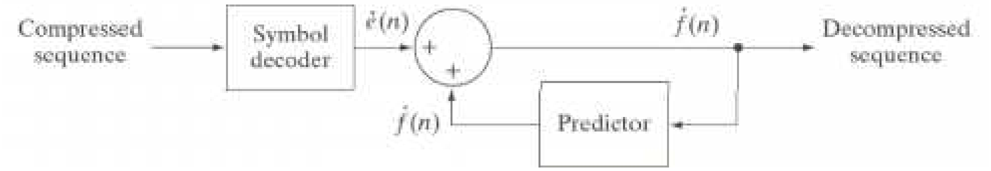
$$\dot{f}(n) = \dot{e}(n) + \hat{f}(n)$$

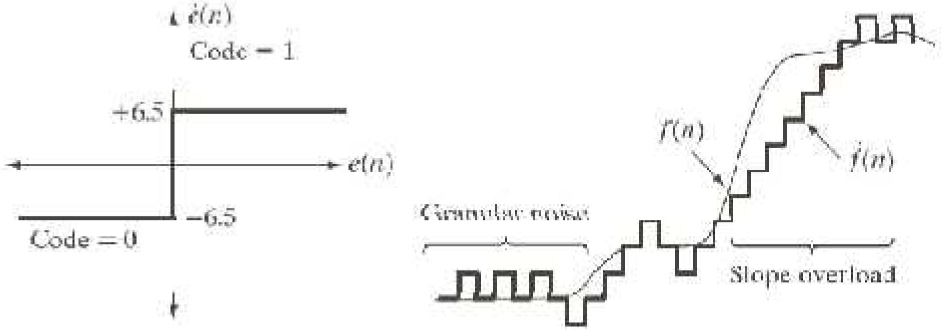
□□

$$\dot{e}(n) = \begin{cases} \square & e(n) \square 0 \\ \square & otherwise \end{cases}$$



the





Input		Encoder				Decoder		Error
n	$f(n)$	$\hat{f}(n)$	$e(n)$	$\dot{e}(n)$	$\dot{f}(n)$	$\hat{f}(n)$	$\dot{f}(n)$	$f(n) - \dot{f}(n)$
0	14	—	—	—	14.0	—	14.0	0.0
1	15	14.0	1.0	6.5	20.5	14.0	20.5	−5.5
2	14	20.5	−6.5	−6.5	14.0	20.5	14.0	0.0
3	15	14.0	1.0	6.5	20.5	14.0	20.5	−5.5
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
14	29	20.5	8.5	6.5	27.0	20.5	27.0	2.0
15	37	27.0	10.0	6.5	33.5	27.0	33.5	3.5
16	47	33.5	13.5	6.5	40.0	33.5	40.0	7.0
17	62	40.0	22.0	6.5	46.5	40.0	46.5	15.5
18	75	46.5	28.5	6.5	53.0	46.5	53.0	22.0
19	77	53.0	24.0	6.5	59.6	53.0	59.6	17.5
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.

Prediction Error

The following images show the prediction error of

the predictor $\hat{f}(x, y) = 0.97 f(x, y-1)$

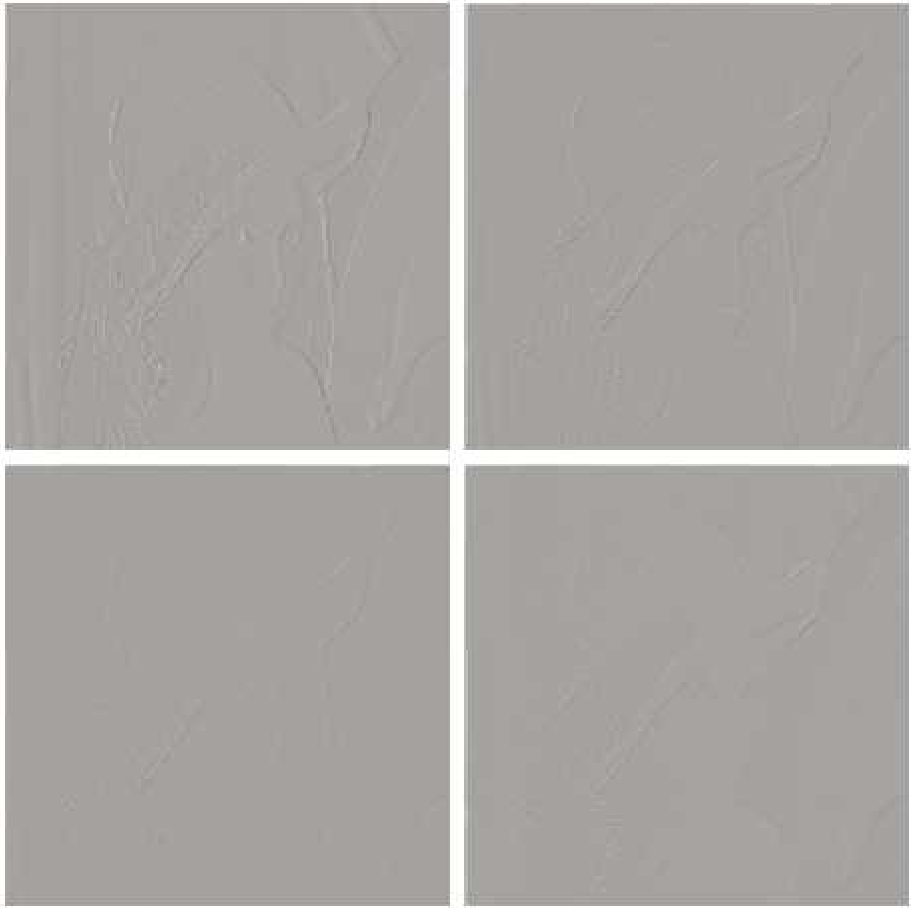
$$\hat{f}(x, y) = 0.5\, f(x, y-1) + 0.5\, f(x-1, y)$$

$$\hat{f}(x, y) = 0.75\, f(x, y-1) + 0.75\, f(x-1, y) - 0.5\, f(x-1, y-1)$$

$$\hat{f}(x, y) = \begin{cases} 0.97 f(x, y-1) & \Delta h \le \Delta v \\ 0.97 f(x-1, y) & otherwise \end{cases}$$

$$\Delta h = | f(x-1, y) - f(x-1, y-1)|$$

$$\Delta v = | f(x,y-1) - f(x-1, y-1)|$$



Optimal Predictors

What are the parameters of a linear predictor that minimize error

$$E\{e^2(n)\} \square\ E\ \square f(n) \square\ \hat{f}(n)\square^{2}$$

While taking into account

$$\overset{\square}{f}(n)\ \square\ e\square(n)\square \qquad \hat{f}(n)\ \square \qquad e(n)\ \square \qquad \hat{f}(n)\ \square \qquad f(n)$$

Using the definition of linear predictor

$$E\{e^2(n)\} = E\left\{\left[\ f(n) - \sum_{i=1}^{m} \alpha_i f(n-1)\right]^2\right\}$$

We assume that f(n) has a mean zero and variance σ^2

$$= R^{-1} r$$

And R^{-1} is the mxn autocorrelation matrix

$$R = \begin{bmatrix} E\{f(n-1)f(n-1)\} & E\{f(n-1)f(n-2)\} & \cdots & E\{f(n-1)f(n-m)\} \\ E\{f(n-2)f(n-1)\} & E\{f(n-2)f(n-2)\} & \cdots & E\{f(n-2)f(n-m)\} \\ \vdots & \vdots & & \vdots \\ E\{f(n-m)f(n-1)\} & E\{f(n-m)f(n-1)\} & \cdots & E\{f(n-m)f(n-1)\} \end{bmatrix}$$

$$r = \begin{bmatrix} E\{f(n)f(n-1)\} \\ \vdots \\ E\{f(n-1)f(n-m)\} \end{bmatrix}$$

$$a = \begin{bmatrix} \alpha_1 \\ \\ \alpha_m \end{bmatrix}$$

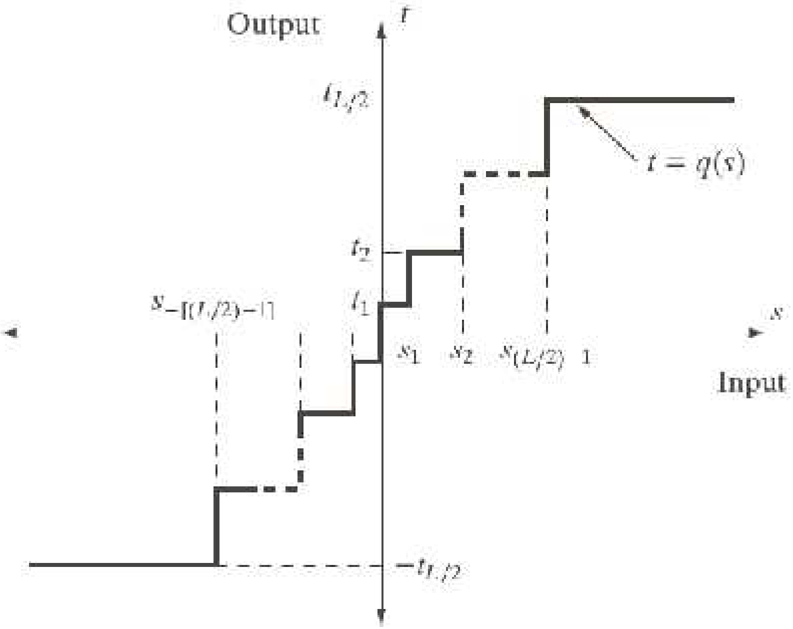
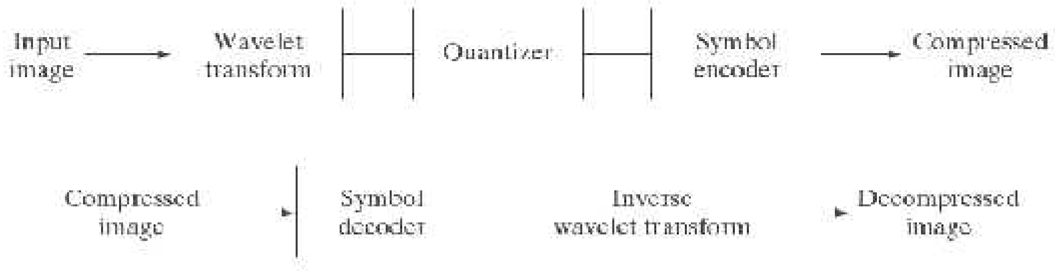


FIGURE 8.44 A typical quantization function.

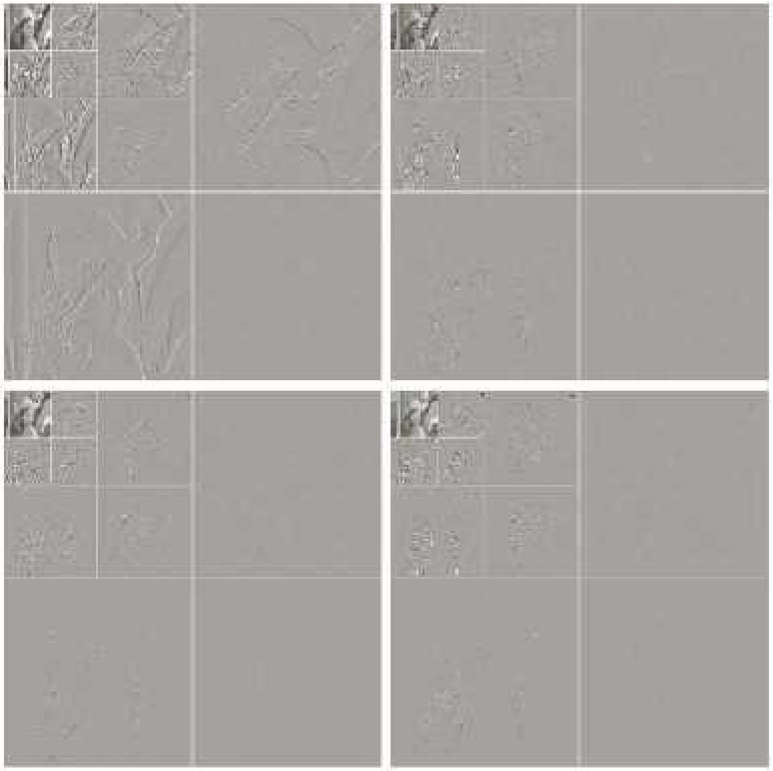
TABLE 8.12 Lloyd-Max quantizers for a Laplacian probability density function of unit variance.

Levels	2		4		8	
i	s_i	t_i	s_i	t_i	s_i	t_i
1	∞	0.707	1.102	0.395	0.504	0.222
2			∞	1.810	1.181	0.785
3					2.285	1.576
4					∞	2.994
θ	1.414		1.087		0.731	



a
b

FIGURE 8.45 A wavelet coding system: (a) encoder; (b) decoder.



a b
c d

FIGURE 8.46 Three-scale wavelet transforms of Fig. 8.9(a) with respect to (a) Haar wavelets, (b) Daubechies wavelets, (c) symlets, and (d) Cohen-Daubechies-Feauveau biorthogonal wavelets.

□

Wavelet	Filter Taps (Scaling + Wavelet)	Zeroed Coefficients
Haar (see Ex. 7.10)	2 + 2	33.8%
Daubechies (see Fig. 7.8)	8 + 8	40.9%
Symlet (see Fig. 7.26)	8 + 8	41.2%
Biorthogonal (see Fig. 7.39)	17 + 11	42.1%

TABLE 8.13 Wavelet transform filter taps and zeroed coefficients when truncating the transforms in Fig. 8.46 below 1.5.

TABLE 8.14 Decomposition level impact on wavelet coding the 512 × 512 image of Fig. 8.9(a).

Decomposition Level (Scales or Filter Bank Iterations)	Approximation Coefficient Image	Truncated Coefficients (%)	Reconstruction Error (rms)
1	256 × 256	74.7%	3.27
2	128 × 128	91.7%	4.23
3	64 × 64	95.1%	4.54
4	32 × 32	95.6%	4.61
5	16 × 16	95.5%	4.63

□

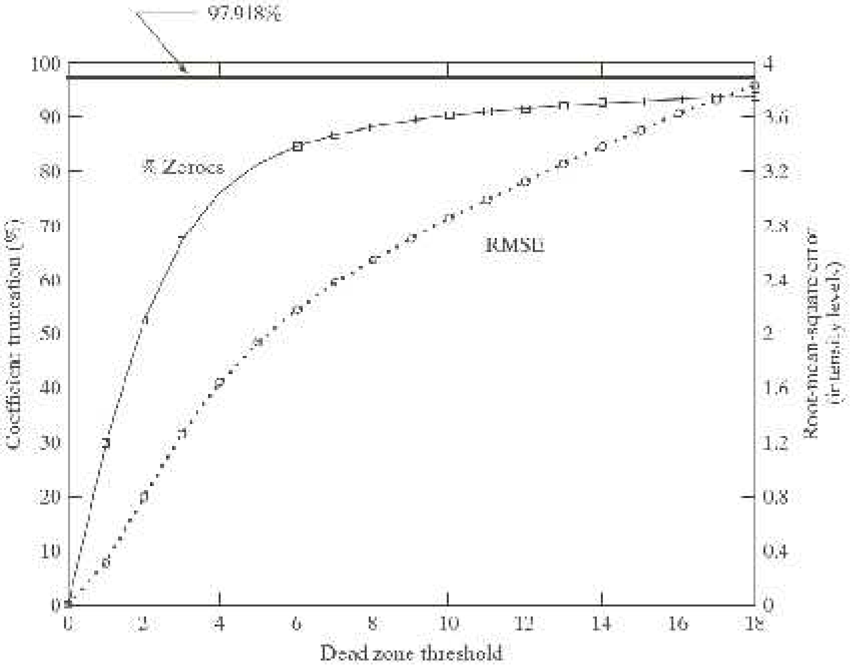


FIGURE 8.47 The impact of dead zone interval selection on wavelet coding.

□

Filter Tap	Highpass Wavelet Coefficient	Lowpass Scaling Coefficient
0	−1.115087052456994	0.6029490182363579
±1	0.5912717631142470	0.2668641184428723
±2	0.05754352622849957	−0.07822326652898785
±3	0.09127176311424948	0.01686411844287495
±4	0	0.02674875741080976

TABLE 8.15 Impulse responses of the low- and highpass analysis filters for an irreversible 9-7 wavelet transform.

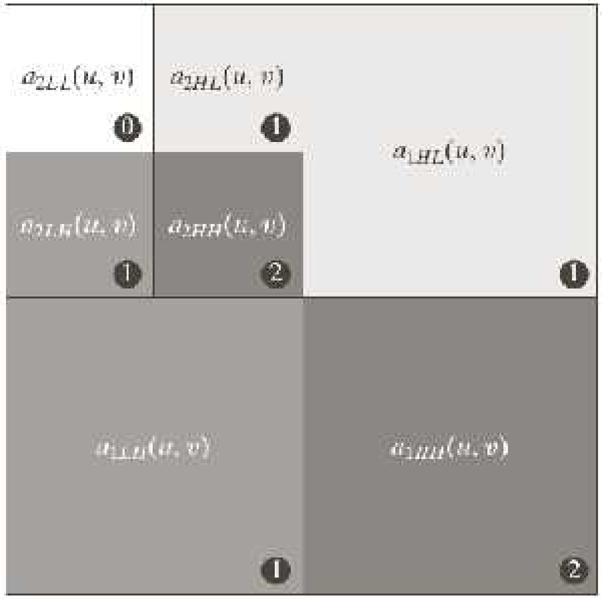
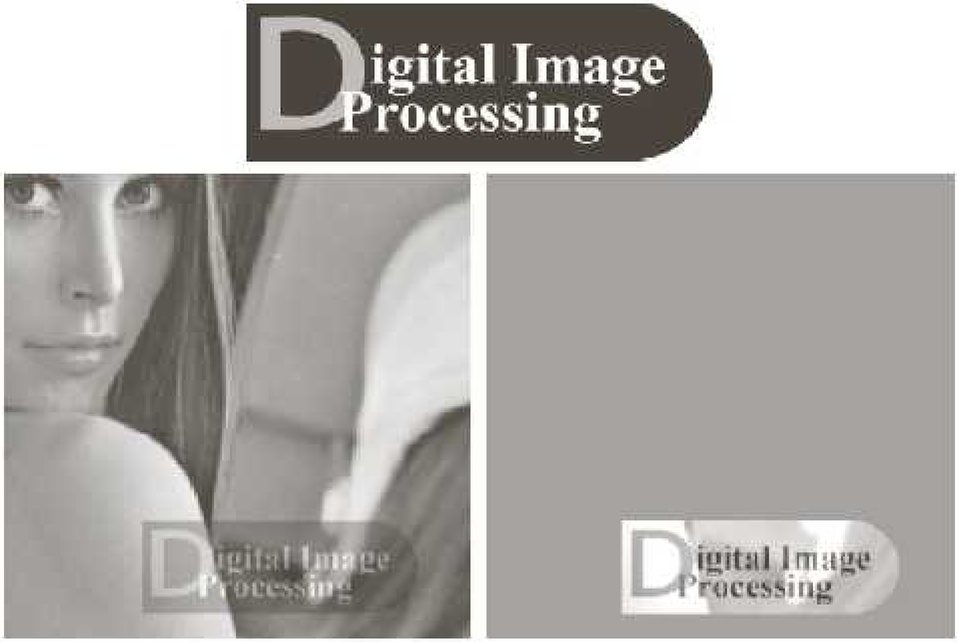


FIGURE 8.48 JPEG 2000 two-scale wavelet transform tile-component coefficient notation and analysis gain.

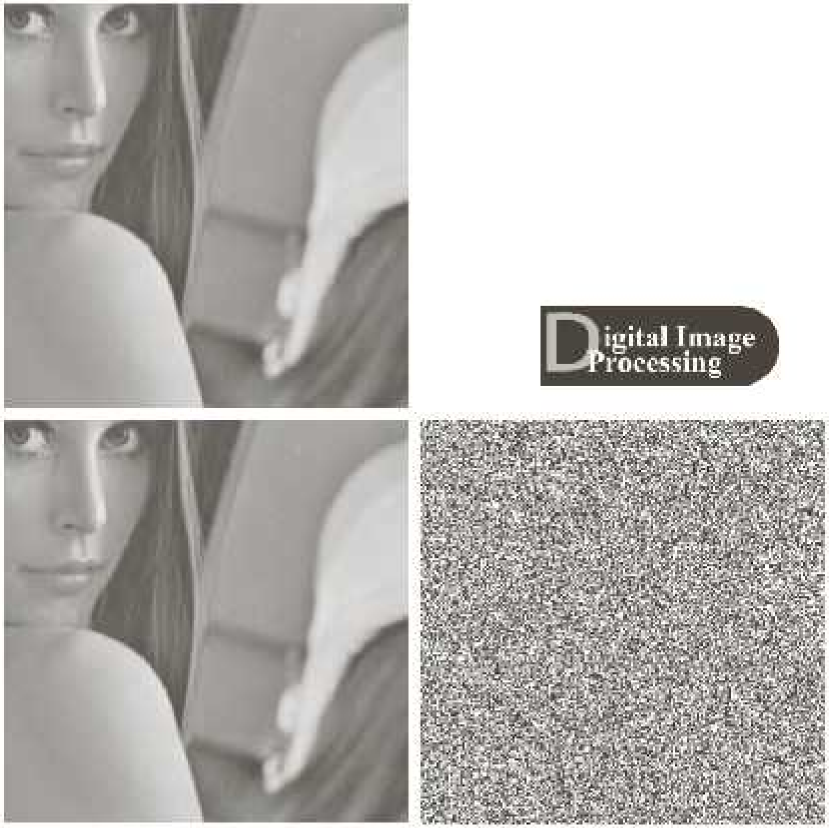
□



a
b c

FIGURE 8.50 A simple visible watermark: (a) watermark; (b) the watermarked image; and (c) the difference between the watermarked image and the original (non-watermarked) image.

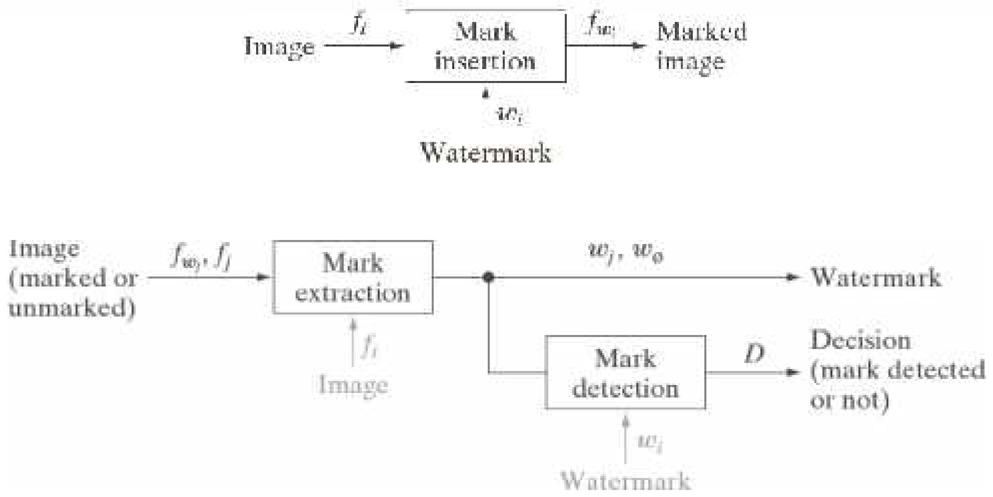
FIGURE 8.49 Four JPEG-2000 approximations of Fig. 8.9(a). Each row contains a result after compression and reconstruction, the scaled difference between the result and the original image, and a zoomed portion of the reconstructed image. (Compare the results in rows 1 and 2 with the JPEG results in Fig. 8.32.)



a b
c d

FIGURE 8.51 A simple invisible watermark: (a) watermarked image; (b) the extracted watermark; (c) the watermarked image after high quality JPEG compression and decompression; and (d) the extracted watermark from (c).

□



a
b

FIGURE 8.52 A typical image watermarking system: (a) encoder; (b) decoder.

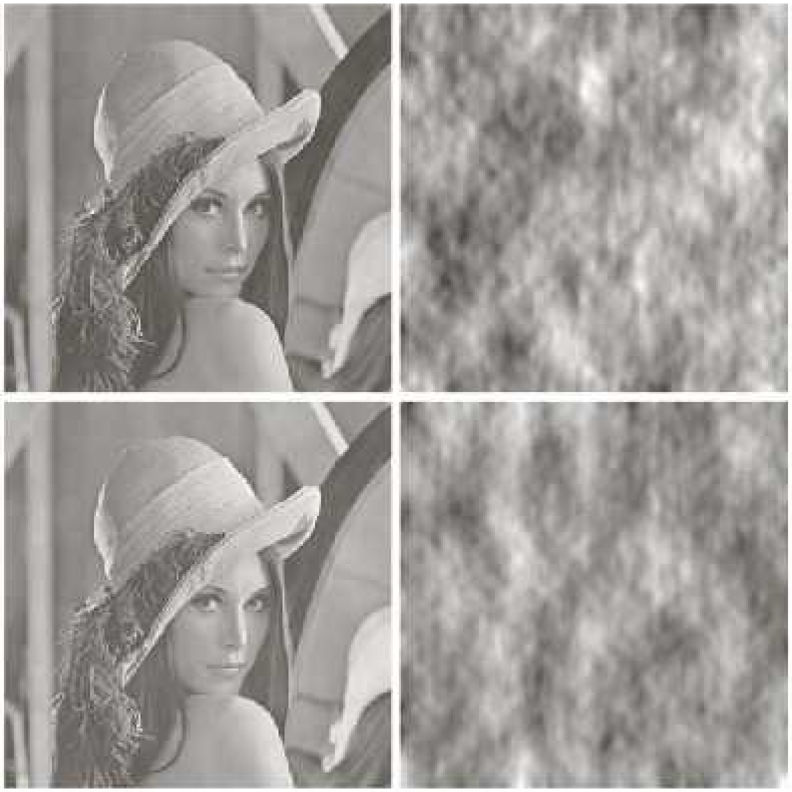
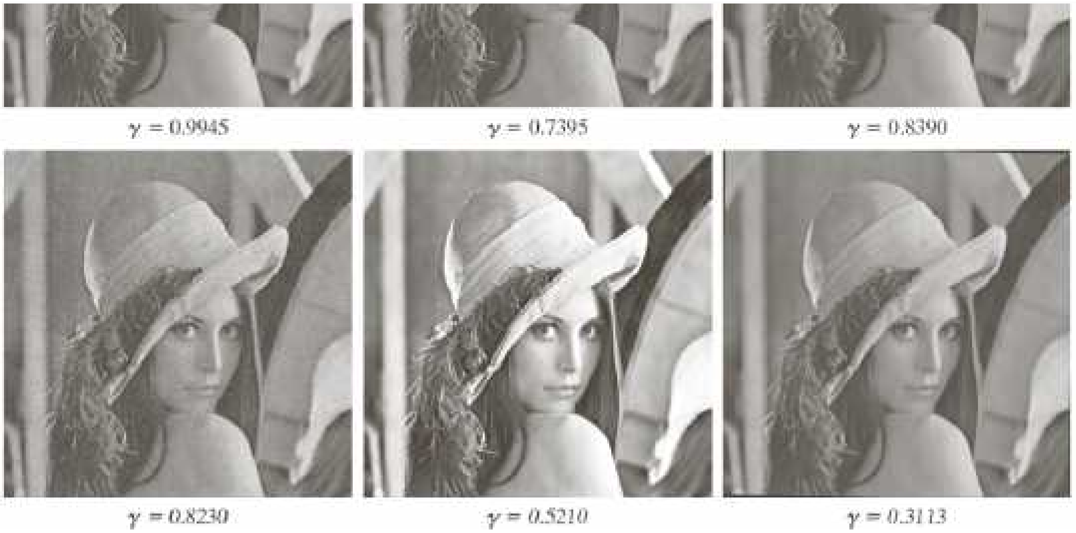


FIGURE 8.53 (a) and (c) Two watermarked versions of Fig. 8.9(a); (b) and (d) the differences (scaled in intensity) between the watermarked versions and the unmarked image. These two images show the intensity contribution (although scaled dramatically) of the pseudo-random watermarks on the original image.

a b c
d e f

FIGURE 8.54 Attacks on the watermarked image in Fig. 8.53(a): (a) lossy JPEG compression and decompression with an rms error of 7 intensity levels; (b) lossy JPEG compression and decompression with an rms error of 10 intensity levels (note the blocking artifact); (c) smoothing by spatial filtering; (d) the addition of Gaussian noise; (e) histogram equalization; and (f) rotation. Each image is a modified version of the watermarked image in Fig. 8.53(a). After modification, they retain their watermarks to varying degrees, as indicated by the correlation coefficients below each image.



$\gamma = 0.9945$ $\gamma = 0.7395$ $\gamma = 0.8390$

$\gamma = 0.8230$ $\gamma = 0.5210$ $\gamma = 0.3113$

UNIT –V
CHAPTER-9

Morphological Image Processing

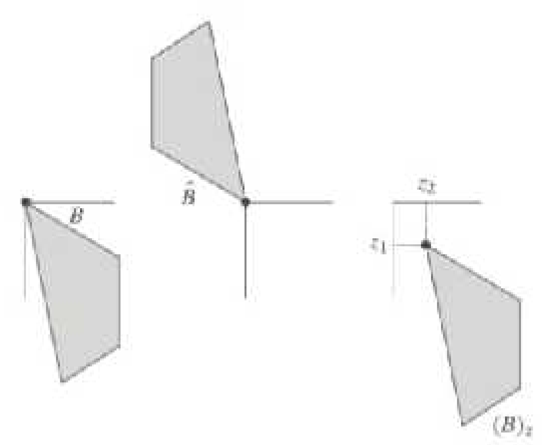
- ✓ Morphology “ – a branch in biology that deals with the form and structure of animals and plants.
- ✓ “Mathematical Morphology” – as a tool for extracting image components, that are useful in the representation and description of region shape.
- ✓ The language of mathematical morphology is – Set theory.
- ✓ Unified and powerful approach to numerous image processing problems.

- In binary images , the set elements are members of the 2-D integer space – Z2. where each element (x,y) is a coordinate of a black (or white) pixel in the image.

Preliminaries:
Relfection:

- In a corresponding output **image reflection** is mainly used as an aid to **image** visualization, but may be used as a preprocessing operator in much the same way as rotation. **Reflection** is a special case of affine **transformation**.
- **Translation** is used to improve visualization of an **image**, but also has a role as a preprocessor in applications where registration of two or more **images** is required.
- **Translation** is a special case of affine **transformation**.

Examples of Reflection and Translation



a b c

FIGURE 9.1 (a) A set, (b) its reflection, and (c) its translation by z.

Structuring Element

- A **structuring element** is a matrix that identifies the pixel in the **image** being **processed** and defines the neighborhood used in the **processing** of each pixel.
- Choose a **structuring element** the same size and shape as the objects you want to **process** in the input **image**.
- a **structuring element** is a shape, used to probe or interact with a given image, with the purpose of drawing conclusions on how this shape fits or misses the shapes in the image.
- It is typically used in morphological operations, such as dilation, erosion, opening, and closing, as well as the hit-or-miss transform.

Used to extract image components that are useful in the representation and description of region shape, such as

- Boundaries extraction
- skeletons
- convex hull
- morphological filtering
- thinning
- pruning

Erosion:

- ✓ **Erosion** (usually represented by ⊖) is one of two fundamental operations (the other being dilation) in morphological image processing from which all other morphological operations are based.
- ✓ . The erosion operation usually uses a structuring element for probing and reducing the shapes contained in the input image.
- ✓ Erosion is used for shrinking of element A by using element B
- ✓ Erosion for Sets A and B in Z^2, is defined by the
- ✓ following equation:

$$A \ominus B = \{z | [(B)z \subseteq A\} \quad (9.2-3)$$

- This equation indicates that the erosion of A by B is the set of all points z such that B, translated by z, is combined in A.

Dialation:

- ✓ **Dilation** (usually represented by ⊕) is one of the basic operations in mathematical morphology. Originally developed for binary images,
- ✓ The dilation operation usually uses a structuring element for probing and expanding the shapes contained in the input image.
- ✓ Dilation is used for expanding an element A by using structuring element B
- ✓ Dilation of A by B and is defined by the following equation:

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

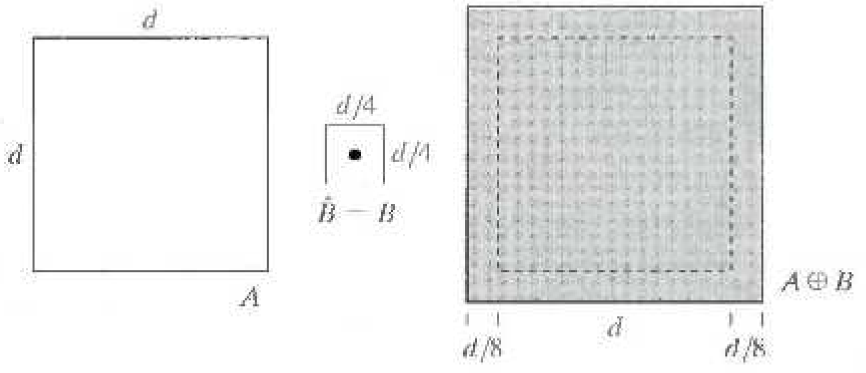
- This equation is based 0n obtaining the reflection 0f B about its origin and shifting this reflection by z.
- The dilation of A by B is the set of all displacements z, such that and A overlap by at least one element. Based

  On this interpretation the equation of (9.2-1) can be rewritten as:

$$A \oplus B = \{z | [(\hat{B})z \cap A] \subset A\} \quad (9.2-2)$$

a b c

FIGURE 9.4 (a) Set A. (b) Square structuring element (dot is the center). (c) Dilation of A by B, shown shaded.



Duality between dilation and erosion:

- Dilation and erosion are duals of each other with respect to set complementation and reflection. That is,

$$(A \ominus B)^c = A^c \oplus \hat{B}$$

- One of the simplest uses of erosion is for eliminating irrelevant details (in terms of size) from a binary image.

Opening And Closing

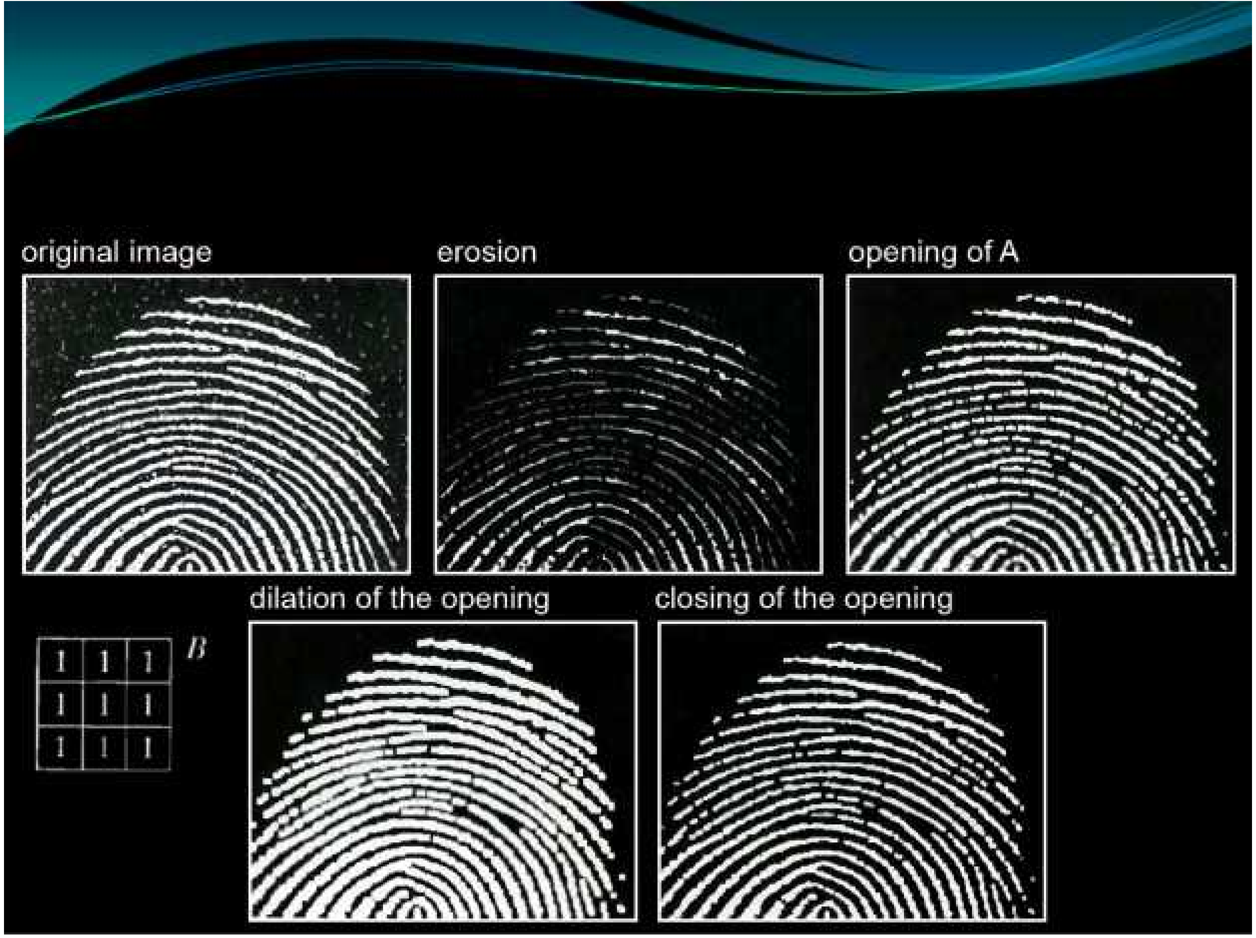
- Opening – smoothes contours , eliminates protrusions
- Closing – smoothes sections of contours, fuses narrow breaks and long thin gulfs, eliminates small holes and fills gaps in contours
- These operations are dual to each other
- These operations are can be applied few times, but has effect only once
- **Opening –:**
    - First – erode A by B, and then dilate the result by B
    - In other words, opening is the unification of all B objects Entirely Contained in A

$$\boldsymbol{A \circ B = (A \ominus B) \oplus B}$$

- **Closing:**
  - First – dilate A by B, and then erode the result by B
  - In other words, closing is the group of points, which the intersection of object B around them with object A – is not empty

$$A \cdot B = (A \oplus B) \ominus B$$

Use of opening and closing for morphological filtering



$$\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \quad B$$

The Hit-or-Miss Transformation:

- A basic morphological tool for <u>**shape detection**</u>.

- Let the origin of each shape be located at its center of gravity.
- If we want to find the location of a shape , say – **X** ,

  at (larger) image, say – **A** :
  - Let X be enclosed by a small window, say – **W**.
  - The ***local background*** of X with respect to W is defined as the *set difference*(**W - X**).
  - Apply *erosion* operator of A by X, will get us the set of locations of the origin of X, such that X is completely contained in A.
  - It may be also view geometrically as the set of all locations of the origin of X at which X found a match (**<u>hit</u>**) in A.
  - Apply *erosion* operator on the *complement of A* by the *local background* set (W – X).
  - Notice, that the set of locations for which X **<u>exactly</u>** fits inside A is the ***intersection*** of these two last operators above.

  This intersection is precisely the location sought.
  - <u>Formally:</u>
  - If B denotes the set composed of X and it's background –
  - B = (B1,B2) ; B1 = X , B2 = (W-X).
  - The match (or set of matches) of B in A, denoted $A \circledast B$ is $A \circledast B = (A \ominus B_1) \cap (A^c \ominus B_2)$

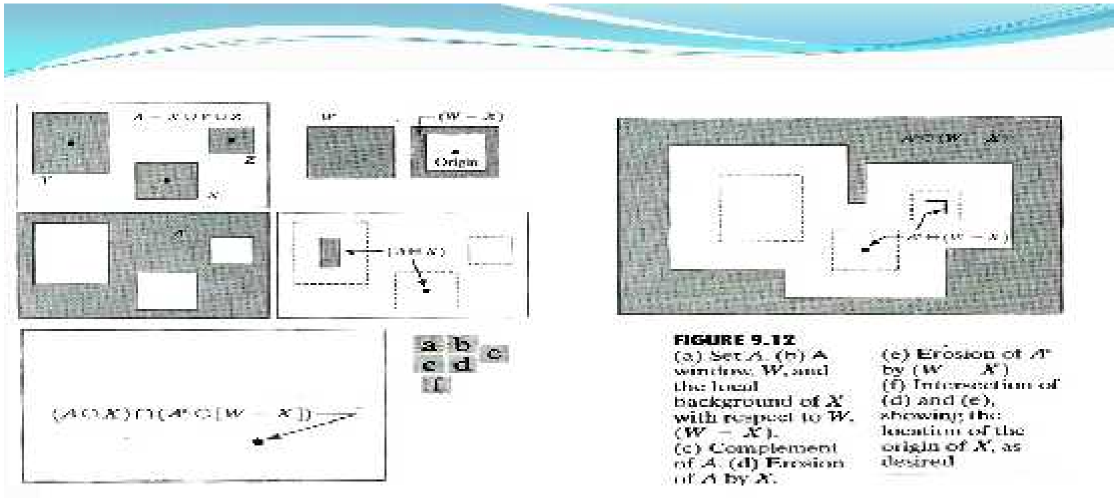


FIGURE 9.12
(a) Set *A*. (b) A window, *W*, and the local background of *X* with respect to *W*, $(W - X)$. (c) Complement of *A*. (d) Erosion of *A* by *X*. (e) Erosion of A^c by $(W - X)$. (f) Intersection of (d) and (e), showing the location of the origin of *X*, as desired.

  - ○ The reason for using these kind of structuring element – B = (B1,B2) is based on an assumed definition that,

two or more objects are distinct only if they are disjoint (disconnected) sets.

- In some applications , we may interested in detecting **certain patterns (combinations)** of 1's and 0 and not for detecting individual objects and not for detecting individual objects.

- In this case a background is not required, and the *hit-or-miss transform* reduces to simple *erosion*.
- This simplified pattern detection scheme is used in some of the algorithms for – **identifying characters within a text**

Other application

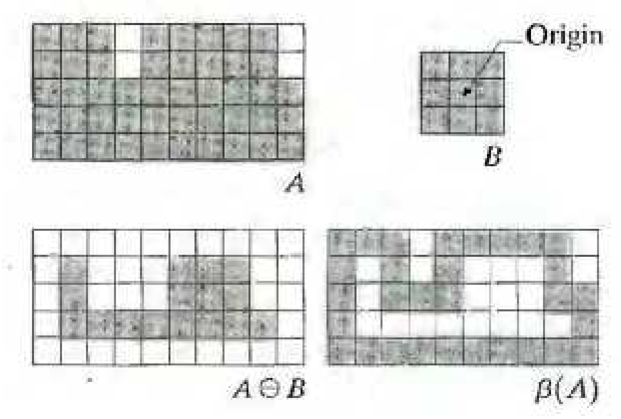
- **Pattern detection**. By definition, the hit-or-miss transform indicates the positions where a certain pattern (characterized by the composite structuring element *B*) occurs in the input image.
-
- **Pruning**. The hit-or-miss transform can be used to identify the end-points of a line to allow this line to be shrunk from each end to remove unwanted branches

Some basic Morphological Algorithm:

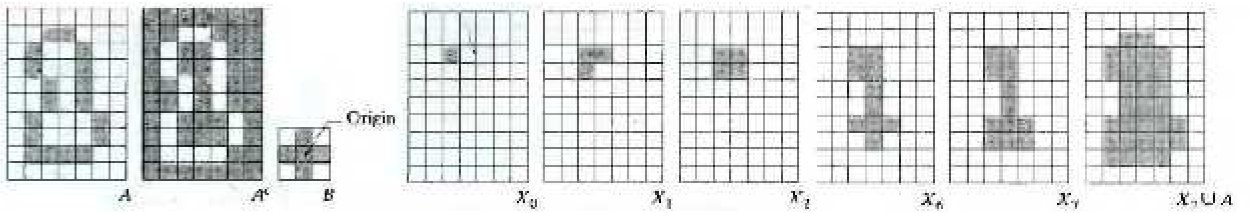
Boundary Extraction:

- First, erode A by B, then make set difference between A and the erosion
- The thickness of the contour depends on the size of constructing object – B

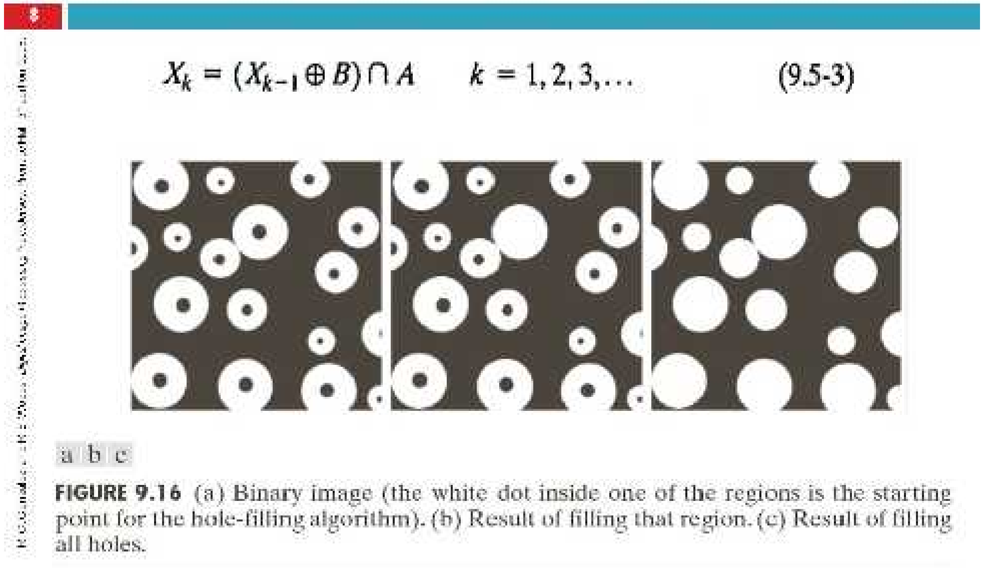
$$\beta(A) = A - (A \ominus B)$$



Hole Filling:

- This algorithm is based on a set of dilations, complementation and intersections
- p is the point inside the boundary, with the value of 1
- X(k) = (X(k-1) xor B) conjunction with complemented A
- The process stops when X(k) = X(k-1)
- The result that given by union of A and X(k), is a set contains the filled set and the boundary
- $$X_k = (X_{k-1} \oplus B) \cap A^c$$
- 

Hole Filling



$$X_k = (X_{k-1} \oplus B) \cap A \qquad k = 1, 2, 3, \ldots \qquad (9.5\text{-}3)$$

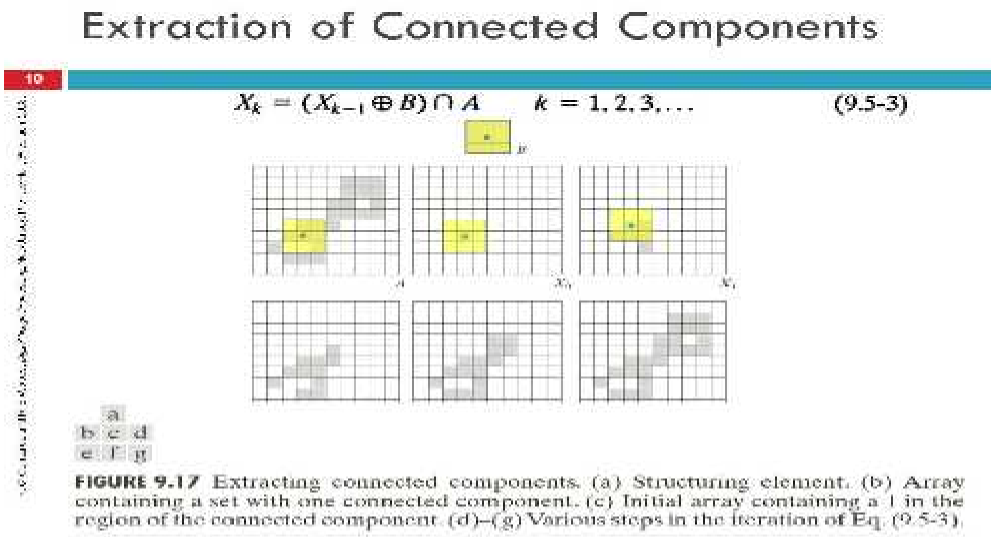
FIGURE 9.16 (a) Binary image (the white dot inside one of the regions is the starting point for the hole-filling algorithm). (b) Result of filling that region. (c) Result of filling all holes.

Extraction of conneceted Components:

This algorithm extracts a component by selecting a point on a binary object A

Works similar to region filling, but this time we use in the conjunction the object A, instead of it's complement.

$$X_k = (X_{k-1} \oplus B) \cap A$$



Extraction of Connected Components

$$X_k = (X_{k-1} \oplus B) \cap A \qquad k = 1, 2, 3, \ldots \qquad (9.5\text{-}3)$$

FIGURE 9.17 Extracting connected components. (a) Structuring element. (b) Array containing a set with one connected component. (c) Initial array containing a 1 in the region of the connected component. (d)–(g) Various steps in the iteration of Eq. (9.5-3).

Convex Hull:\

A is said to be convex if a straight line segment joining any two points in A lies entirely within A

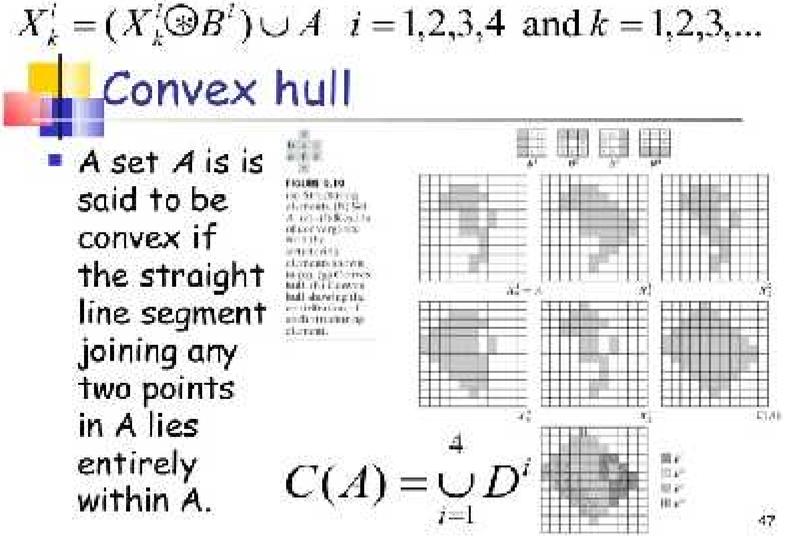
The convex hull H of set S is the smallest convex set containing S

Convex deficiency is the set difference H-S

Useful for object description

This algorithm iteratively applying the hit-or-miss transform to A with the first of B element, unions it with A, and repeated with second element of B

$$X_k^i = (X_{k-1} \circledast B^i) \cup A$$



$$X_k^i = (X_k^i \circledast B^i) \cup A \quad i = 1,2,3,4 \text{ and } k = 1,2,3,\ldots$$

Convex hull

- A set *A* is is said to be convex if the straight line segment joining any two points in A lies entirely within A.

$$C(A) = \bigcup_{i=1}^{4} D^i$$

47

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:

  $\{B\}=\{B^1, B^2, B^3, \ldots, B^n\}$
- Where B^i is a rotated version of B^{i-1}. Using this concept we define thinning by a sequence of structuring elements:

$$A \otimes \{B\} = ((\ldots((A \otimes B^1) \otimes B^2)\ldots) \otimes B^n)$$

- 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:
- $A \otimes B = A - (A \circledast B) = A \cap (A \circledast B)^c$

Thickining:

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:

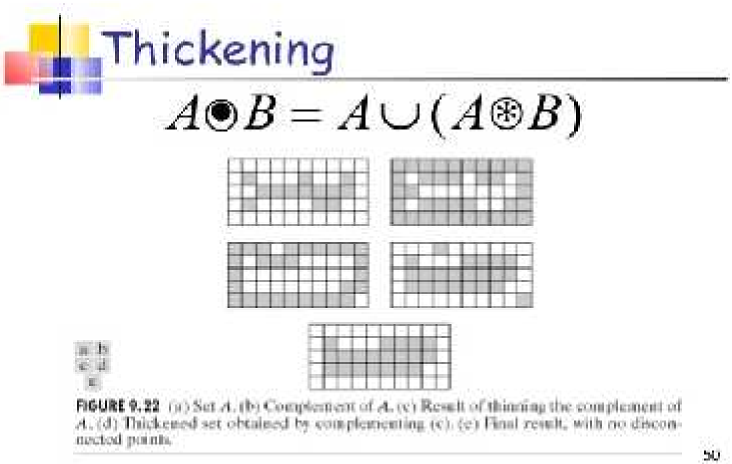
$$A \odot \{B\} = ((\ldots((A \odot B^1) \odot B^2)\ldots) \odot B^n)$$

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 than 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.



Thickening

$$A \odot B = A \cup (A \circledast B)$$

FIGURE 9.22 (a) Set A. (b) Complement of A. (c) Result of thinning the complement of A. (d) Thickened set obtained by complementing (c). (e) Final result, with no disconnected points.

50

Skeletons:

skeletonization is a **transformation** of a component of a digital **image** into a subset of the original component.

The notion of a skeleton S(A) of a set A is intuitively defined, we deduce from this figure that:

If z is a point of S(A) and (D)z is the largest disk centered in z and contained in A (one cannot find a larger disk that fulfils this terms) – this disk is called “maximum disk”.

The disk (D)z touches the boundary of A at two or more different places.

The skeleton of A is defined by terms of erosions and openings:

$$S(A) = \bigcup_{k=0}^{K} S_k(A)$$

With $S_k(A) = (A \ominus kB) - (A \ominus kB) \circ B$

Where B is the structuring element and $(A \ominus kB)$ indicates k successive erosions of A:

$$(A \ominus kB) = (\ldots((A \ominus B) \ominus B) \ominus \ldots) \ominus B$$

k times, and K is the last iterative step before A erodes to an empty set, in other words

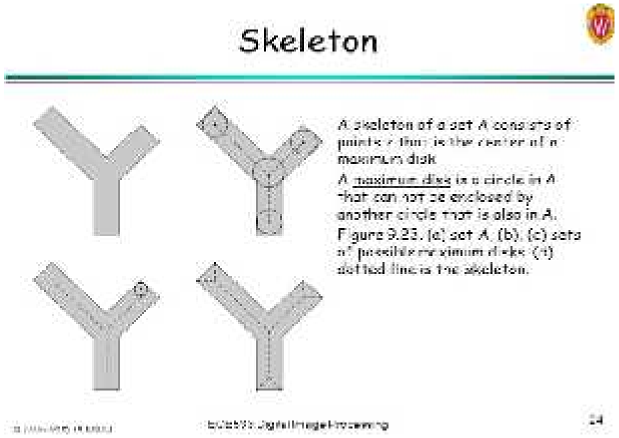
$$K = \max\{k | (A \ominus kB) \neq \emptyset\}$$

in conclusion S(A) can be obtained as the union of skeleton subsets Sk(A).
A can be also reconstructed from subsets Sk(A) by using the equation

$$A = \bigcup_{k=0}^{K} (S_k(A) \oplus kB)$$

Where $(S_k(A) \oplus kB)$ denotes k successive dilations of Sk(A) that is:

$$(S_k(A) \oplus kB) = ((\ldots((S_k(A) \oplus B) \oplus B) \oplus \ldots) \oplus B$$



Skeleton

A skeleton of a set A consists of points z that is the center of a maximum disk

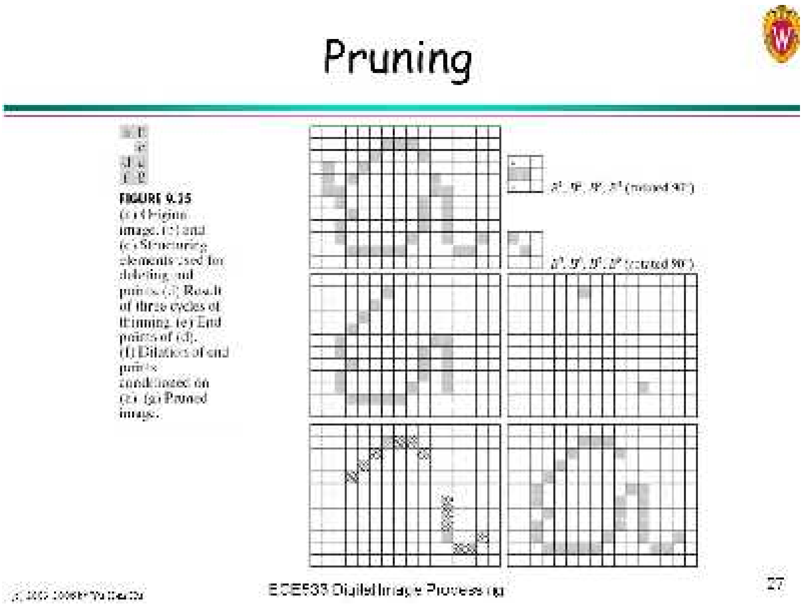
A maximum disk is a circle in A that can not be enclosed by another circle that is also in A.

Figure 9.23. (a) set A, (b), (c) sets of possible maximum disks (d) dotted line is the skeleton.

[illegible] ECE533 Digital Image Processing 24

Pruning:

- ✓ The **pruning** algorithm is a technique used in digital **image processing** based on mathematical **morphology**.
- ✓ It is used as a complement to the skeleton and thinning algorithms to remove unwanted parasitic components (spurs



Pruning

a b
c
d e
f g

FIGURE 9.35 (a) Original image. (b) and (c) Structuring elements used for deleting end points. (d) Result of three cycles of thinning. (e) End points of (d). (f) Dilation of end points conditioned on (a). (g) Pruned image.

B^1, B^2, B^3, B^4 (rotated 90°)
B^5, B^6, B^7, B^8 (rotated 90°)

© 2002-2008 by Yu Hen Hu

ECE533 Digital Image Processing

27

Morphological Reconstruction Skeleton:

Morphological reconstruction can be thought of conceptually as repeated dilations of an image, called the marker image, until the contour of the marker image fits under a second image, called the mask image.

In morphological reconstruction, the peaks in the marker image “spread out,” or dilate.Reconsctruction can be done by geodesic dialation and by erosion.

Sample Applications: Opening by reconstruction
Filling Holes
Border Clearning

Gray Scale Morphology:

In gray scale images on the contrary to binary images we deal with digital image functions of the form f(x,y) as an input image and b(x,y) as a structuring element.

(x,y) are integers from Z*Z that represent a coordinates in the image.

f(x,y) and b(x,y) are functions that assign gray level value to each distinct pair of coordinates.

For example the domain of gray values can be 0-255, whereas 0 – is black, 255- is white

Erosion and Dialation in Gray scale :

The grayscale dilation of an image involves assigning to each pixel, the maximum value found over the neighborhood of the structuring element.

Equation for gray-scale dilation is:

$$(f \oplus b)(s,t) =$$

$$\max\{f(s-x,t-y)+b(x,y)|(s-x),(t-y)\in D_f,(x,y)\in D_b\}$$

Df and Db are domains of f and b.

The condition that (s-x),(t-y) need to be in the domain of f and x,y in the domain of b, is analogous to the condition in the binary definition of dilation, where the two sets need to overlap by at least one element.

We will illustrate the previous equation in terms of

1-D. and we will receive an equation for 1 variable:

$$(f \oplus b)(s) = \max\{f(s-x)+b(x)|(s-x)\in D_f \; and \; x \in D_b\}$$

The requirements the (s-x) is in the domain of f and x is in the domain of b imply that f and b overlap by at least one element.

Unlike the binary case, f, rather than the structuring element b is shifted.

Conceptually f sliding by b is really not different than b sliding by f.

Gray-scale erosion is defined as:

$$(f \ominus b)(s,t) = \min\{f(s+x,t+y)-b(x,y)|(s+x),(t+y)\in D_f,(x,y)\in D_b\}$$

The condition that (s+x),(t+y) have to be in the domain of f, and x,y have to be in the domain of b, is completely analogous to the condition in the binary definition of erosion, where the structuring element has to be completely combined by the set being eroded.

The same as in erosion we illustrate with 1-D function

$$(f \ominus b)(s) = \min\{f(s+x)-b(x)|(s+x)\in D_f \; and \; x \in D_b\}$$

General effect of performing an erosion in grayscale images:

If all elements of the structuring element are positive, the output image tends to be darker than the input image.

The effect of bright details in the input image that are smaller in area than the structuring element is reduced, with the degree of reduction being determined by

the grayscale values surrounding by the bright detail and by shape and amplitude values of the structuring element itself.

Similar to binary image grayscale erosion and dilation are duals with respect to function complementation and reflection

Opening and Closing in Grayscale:

Similar to the binary algorithm

Opening –

$$f \circ b = (f \ominus b) \oplus b.$$

Closing –

$$f \bullet b = (f \oplus b) \ominus b.$$

In the opening of a gray-scale image, we remove small light details, while relatively undisturbed overall gray levels and larger bright features

In the closing of a gray-scale image, we remove small dark details, while relatively undisturbed overall gray levels and larger dark features

Some Basic gray scale Morphological algorithms:

Morphological smoothing

Perform *opening* followed by a *closing*

The net result of these two operations is to remove or attenuate both bright and dark artifacts or noise.

Morphological gradient

Dilation and *erosion* are use to compute the *morphological gradient* of an image, denoted g:

$$g = (f \oplus b) - (f \ominus b)$$

It uses to highlights sharp gray-level transitions in the input image.

Obtained using symmetrical structuring elements tend to depend less on edge directionality.

Top-hat transformation

Denoted h, is defined as: $h = f - (f \circ b)$

Cylindrical or parallelepiped *structuring element function* with a flat top.

Useful for **enhancing detail in the presence of shading**.

Granulometry

Granulometry is a field that deals principally with

determining the size distribution of particles in an image.

Because the particles are lighter than the background, we can use a morphological approach to determine size distribution. To construct at the end a

histogram of it.

Based on the idea that *opening* operations of particular size have the most effect on regions of the input image that contain particles of similar size.

This type of processing is **useful for describing regions with a predominant particle-like character**

Key features of the application include:

use of mathematical morphology functions such as closing or opening

computation of granulometric curves, obtained when the size of the structuring element varies

application to grey scale images, avoiding image segmentation

batch processing: all the images in a directory are processes the same way, making it possible to apply groupwise analyses

the different processing steps were embedded within a graphical user interface.

Textural Segmentation:

Textural segmentation

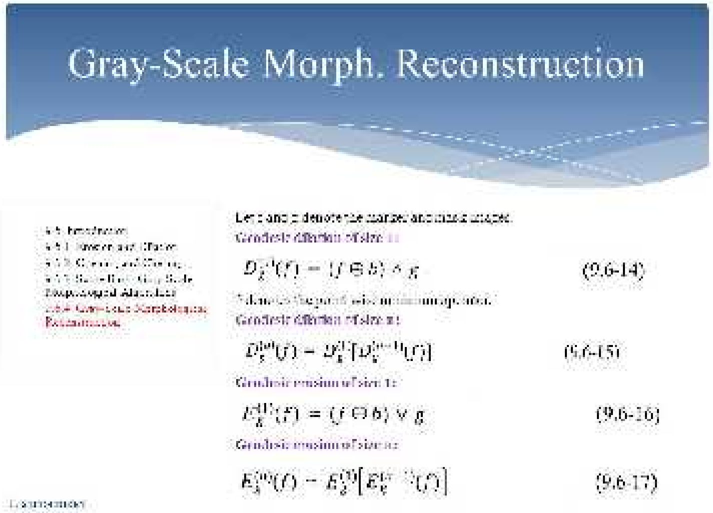
The objective is to find the boundary between different image regions based on their textural content.

Close the input image by using successively larger*structuring elements*.

Then, single *opening* is preformed ,and finally a simple *threshold* that yields the boundary between the textural regions.

Texture segmentation is the process of partitioning an image into regions with different textures containing similar group of pixels.

Gray scale Morphological Reconstruction:



Gray-Scale Morph. Reconstruction

9.6 Introduction
9.6.1 Erosion and Dilation
9.6.2 Opening and Closing
9.6.3 Some Basic Gray-Scale Morphological Algorithms
9.6.4 Gray-Scale Morphological Reconstruction

Let f and g denote the marker and mask images.

Geodesic dilation of size 1:

$$D_g^{(1)}(f) = (f \oplus b) \wedge g \qquad (9.6\text{-}14)$$

∧ denotes the point-wise minimum operator.

Geodesic dilation of size n:

$$D_g^{(n)}(f) = D_g^{(1)}\left[D_g^{(n-1)}(f)\right] \qquad (9.6\text{-}15)$$

Geodesic erosion of size 1:

$$E_g^{(1)}(f) = (f \ominus b) \vee g \qquad (9.6\text{-}16)$$

Geodesic erosion of size n:

$$E_g^{(n)}(f) = E_g^{(1)}\left[E_g^{(n-1)}(f)\right] \qquad (9.6\text{-}17)$$

[illegible]

Image segmentation:

in digital image processing and computer vision, **image segmentation** is the process of partitioning a digital image into multiple segments (sets of pixels, also known as image objects). The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze.[1][2] Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain characteristics.

The result of image segmentation is a set of segments that collectively cover the entire image, or a set of contours extracted from the image (see edge detection). Each of the pixels in a region are similar with respect to some characteristic or computed property, such as color, intensity, or texture. Adjacent regions are significantly different with respect to the same characteristic(s).[1] When applied to a stack of images, typical in medical imaging, the resulting contours after image segmentation can be used to create 3D reconstructions with the help of interpolation algorithms like marching cubes.[3]

Image Processing :

Image processing is a method to convert an image into digital form and perform some operations on it, in order to get an enhanced image or to extract some useful information from it.

It is a type of signal dispensation in which input is an image, like video frame or photograph and output may be image or characteristics associated with that image.

Usually Image Processing system includes treating images as two dimensional signals while applying already set signal processing methods to them.

Purpose of Image processing

The purpose of image processing is divided into 5 groups. They are :

Visualization - Observe the objects that are not visible.

Image sharpening and restoration - To create a better image.

Image retrieval - Seek for the image of interest.

Measurement of pattern – Measures various objects in an image.

Image Recognition – Distinguish the objects in an image.

Fundamental steps in Digital 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 preprocessing, 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.

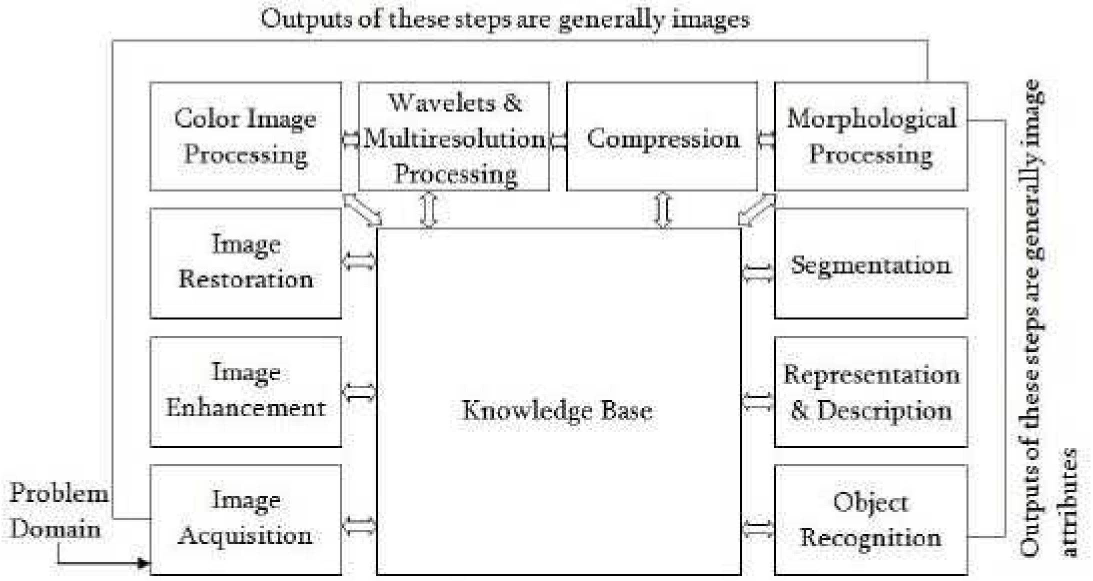


Figure 1

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 Multiresolution 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.

7. Morphological Processing

Morphological processing deals with tools for extracting image components that are useful in the representation and description of shape.

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.

Point line edge detection:

n image processing, **line detection** is an algorithm that takes a collection of n edge points and finds all the lines on which these edge points lie.[1] The most popular line detectors are the Hough transform and convolution-based techniques.[2]

Hough transform

The Hough transform[3] can be used to detect lines and the output is a parametric description of the lines in an image, for example ρ = r cos(θ) + c sin(θ).[1] If there is a line in a row and column based image space, it can be defined ρ, the distance from the origin to the line along a perpendicular to the line, and θ, the angle of the perpendicular projection from the origin to the line measured in degrees clockwise from the positive row axis. Therefore, a line in the image corresponds to a point in the Hough space.[4] The Hough space for lines has therefore these two dimensions θ and ρ, and a line is represented by a single point corresponding to a unique set of these parameters. The Hough transform can then be implemented by choosing a set of values of ρ and θ to use. For each pixel (r, c) in the image, compute r cos(θ) + c sin(θ) for each values of θ, and place the result in the appropriate position in the (ρ, θ) array. At the end, the values of (ρ, θ) with the highest values in the array will correspond to strongest lines in the image

Convolution-based technique

In a convolution-based technique, the line detector operator consists of a convolution masks tuned to detect the presence of lines of a particular width n and a θ orientation. Here are the four convolution masks to detect horizontal, vertical, oblique (+45 degrees), and oblique (−45 degrees) lines in an image.

a) Horizontal mask(R1)

−1	−1	−1
2	2	2
−1	−1	−1

(b) Vertical (R3)

−1	2	−1
−1	2	−1
−1	2	−1

(C) Oblique (+45 degrees)(R2)

−1	−1	2
−1	2	−1
2	−1	−1

(d) Oblique (−45 degrees)(R4)

2	−1	−1
−1	2	−1
−1	−1	2

[5]

In practice, masks are run over the image and the responses are combined given by the following equation:

R(x, y) = max(|R1 (x, y)|, |R2 (x, y)|, |R3 (x, y)|, |R4 (x, y)|)

If R(x, y) > T, then discontinuity

As can be seen below, if mask is overlay on the image (horizontal line), multiply the coincident values, and sum all these results, the output will be the (convolved image). For example, (−1)(0)+(−1)(0)+(−1)(0) + (2)(1) +(2)(1)+(2)(1) + (−1)(0)+(−1)(0)+(−1)(0) = 6 pixels on the second row, second column in the (convolved image) starting from the upper left corner of the horizontal lines.[1]

Detection of isolated points:

Segmentation algorithms generally are based on one of 2 basis properties of intensity values: θ discontinuity : to partition an image based on sharp changes in intensity θ similarity : to partition an image into regions that are similar according to a set of predefined criteria.
A more formal definition – Let R represent the entire image. Segmentation is a process that divides R into n subregions $R1$, $R2$, … , Rn such that: 1. $Ri\ n\ i{=}1 = R$. 2. Ri is a connected set for each $i = 1, 2, \ldots, n$. 3. $Ri \cap Rj = \emptyset$ for all i and $j, j \neq i$. 4. $Q\ Ri = TRUE$ for each $i = 1, 2, \ldots, n$. 5. $Q\ Ri \cup Rj = FALSE$ for any adjacent regions Ri and . Here $Q\ Rk$ is a predicate that indicates some property over the region.
First derivative:
First derivative generally produce thicker edges in an image ν Second derivative has a very strong response to fine details and noise ν Second derivative sign can be used to determine transition direction.

ν Based on the fact that a second order derivative is very sensitive to sudden changes we will use it to detect an isolated point. ν We will use a Laplacian which is the second order derivative over a two dimensional function.
$\nabla 2\ f\ x, y = \partial 2\ f(x, y)\ \partial x\ 2 + \partial 2\ f(x, y)\ \partial y\ 2\ \partial 2\ f(x, y)\ \partial x\ 2 = f\ x + 1, y - 2f\ x, y + f(x - 1, y)\ \partial 2\ f(x, y)\ \partial y\ 2 = f\ x, y + 1 - 2f\ x, y + f(x, y - 1)\ \nabla 2\ f\ x, y = f\ x + 1, y + f\ x - 1, y + f\ x, y + 1 + f\ x, y - 1 - 4f\ x, y$
Line detection:
The Laplacian is isotropic, i.e. independent of direction. ν If we would like to detect lines on a certain direction only we might want to use masks that would emphasize a certain direction and be less sensitive to other directions. 3 differentt edge types are observed: ν Step edge – Transition of intensity level over 1 pixel only in ideal, or few pixels on a more practical use ν Ramp edge – A slow and graduate transition ν Roof edge – A transition to a different intensity and back. Some kind of spread line.

Chapter 12
Object Recognition

Pattern: An arrangement of descriptors (or features).
Pattern class: a family of **patterns** sharing some common properties. –
They are denoted by ω1, ω2,…,ωW, W being the number of **classes**. **patterns** to their **classes** with as little human interaction as possible.

- Patterns and features

- Pattern classes: a pattern class is a family of patterns that share some common properties

- Pattern recognition: to assign patterns to their respective classes

- Three common pattern arrangements used in practices are

  - Vectors
  - Strings
  - Trees

- **Patterns and Pattern Classes Vector Example**

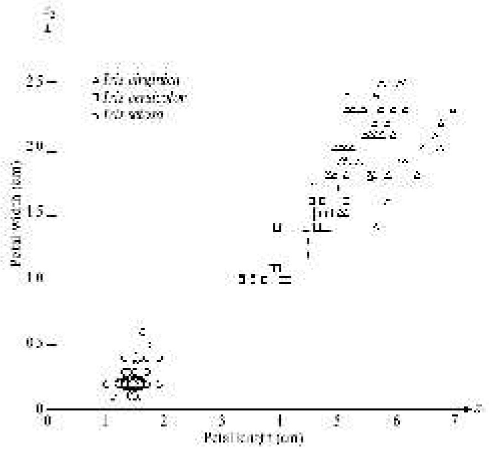
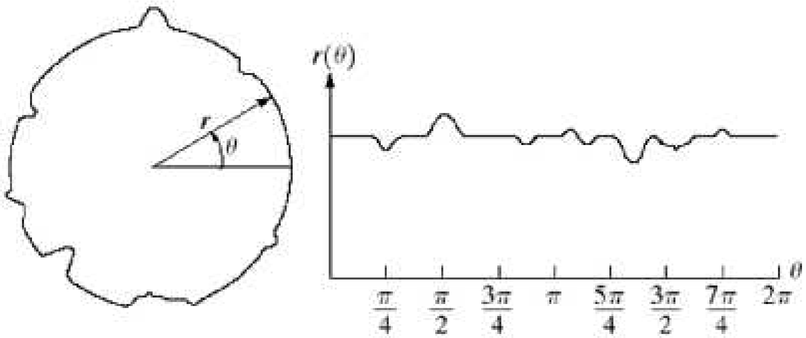


FIGURE 12.1 Three types of iris flowers described by two measurements.

–

Patterns and Pattern Classes Another Vector Example

- **Here is another example of pattern vector generation.**



a b

FIGURE 12.2 A noisy object and its corresponding signature.

- **In this case, we are interested in different types of noisy shapes.**

Patterns and Pattern Classes String Example

- String descriptions adequately generate patterns of objects and other entities whose structure is based on relatively simple connectivity of primitives, usually associated with boundary shape.

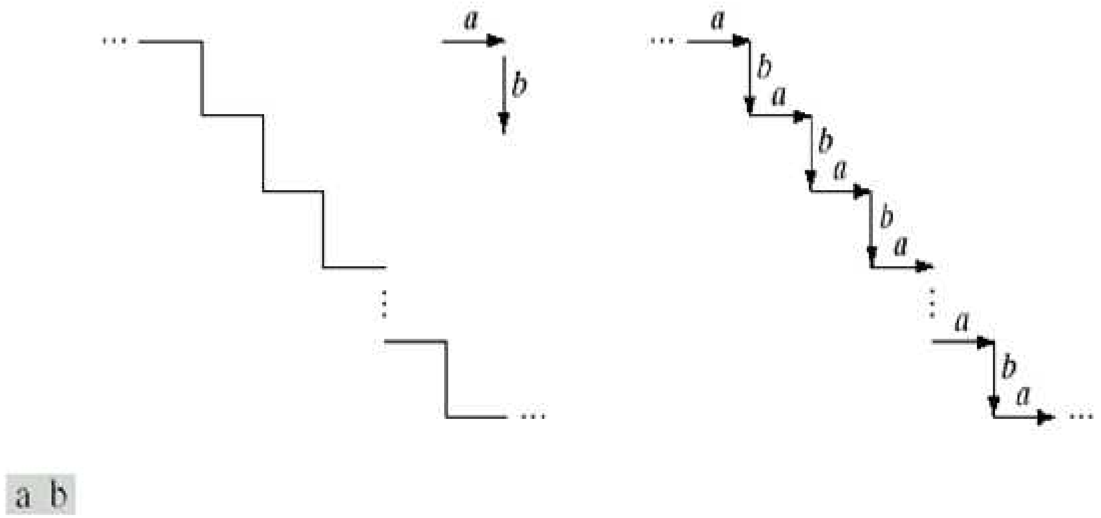


FIGURE 12.3 (a) Staircase structure. (b) Structure coded in terms of the primitives ***a*** and ***b*** to yield the string description . . . ***ababab*** . . . .

Patterns and Pattern Classes Tree Example

- Tree descriptions is more powerful than string ones.

- Most hierarchical ordering schemes lead to tree structure

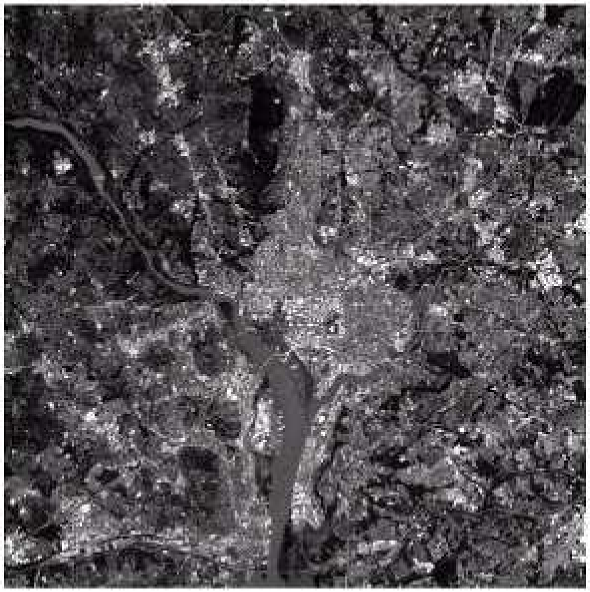


FIGURE 12.4 Satellite image of a heavily built downtown area (Washington, D.C.) and surrounding residential areas. (Courtesy of NASA.)

Recognition Based on Decision-Theoretic Methods

- Decision-theoretic approaches to recognition are based on the use decision functions.
  - Let $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$ represent an *n*-dimensional pattern vector. For *W* pattern classes $\omega_1, \omega_2, \ldots, \omega_W$, we want to find *W* decision functions with the property that, if a pattern **x** belongs to class ω_i, then

$$d_i(\mathbf{x}) > d_j(\mathbf{x}) \qquad j = 1,2,\ldots,W; j \neq i$$

- The decision boundary separating class and is given by $d_i(\mathbf{x}) = d_j(\mathbf{x})$ or $d_i(\mathbf{x}) - d_j(\mathbf{x}) = 0$

Recognition Based on Decision-Theoretic MethodsMatching

- **Minimum distance classifier**

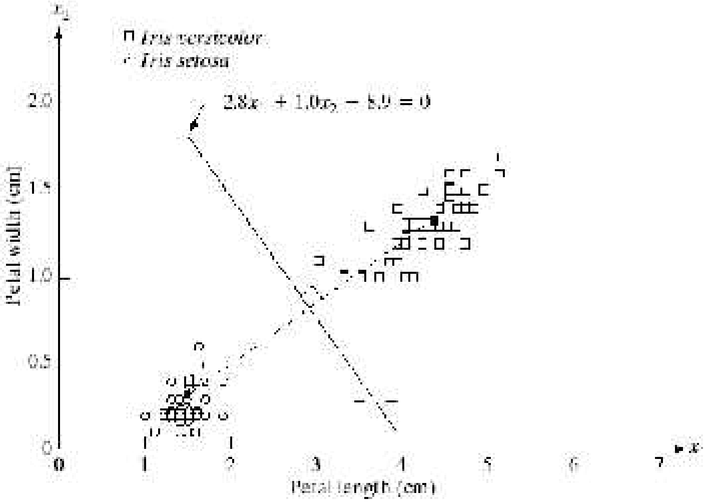


FIGURE 12.6 Decision boundary of minimum distance classifier for the classes of *Iris versicolor* and *Iris setosa*. The dark dot and square are the means.

Recognition Based on Decision-Theoretic MethodsMatching by Correlation

- **The correlation between *f*(*x*,*y*) and *w*(*x*,*y*) is**

$$c(x, y) = \sum_{s} \sum_{t} f(s, t) w(x + s, y + t)$$

for $x = 0,1,2,\ldots, M-1,$

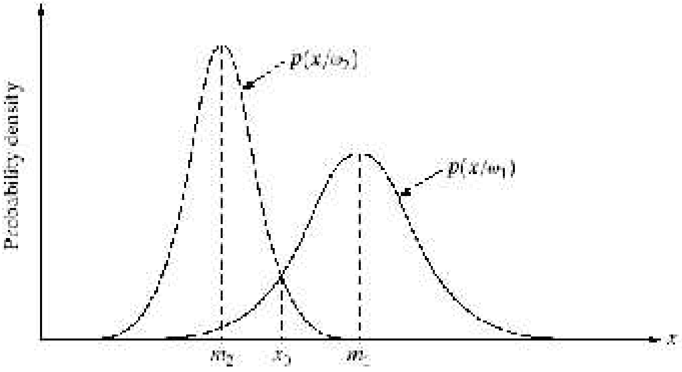
$y = 0,1,2,\ldots, N-1$

Recognition Based on Decision-Theoretic Methods Optimum Statistical Classifiers:

- Bayes classifier for Gaussian pattern classes

$$d_j(x) = p(x \mid \omega_j) p(\omega_j) = \frac{1}{\sqrt{2\pi}\sigma_j} e^{-\frac{(x-m_j)^2}{2\sigma_j^2}} p(\omega_j)$$

FIGURE 12.10 Probability density functions for two 1-D pattern classes. The point x_0 shown is the decision boundary if the two classes are equally likely to occur.



Recognition Based on Decision-Theoretic Methods Optimum Statistical Classifiers

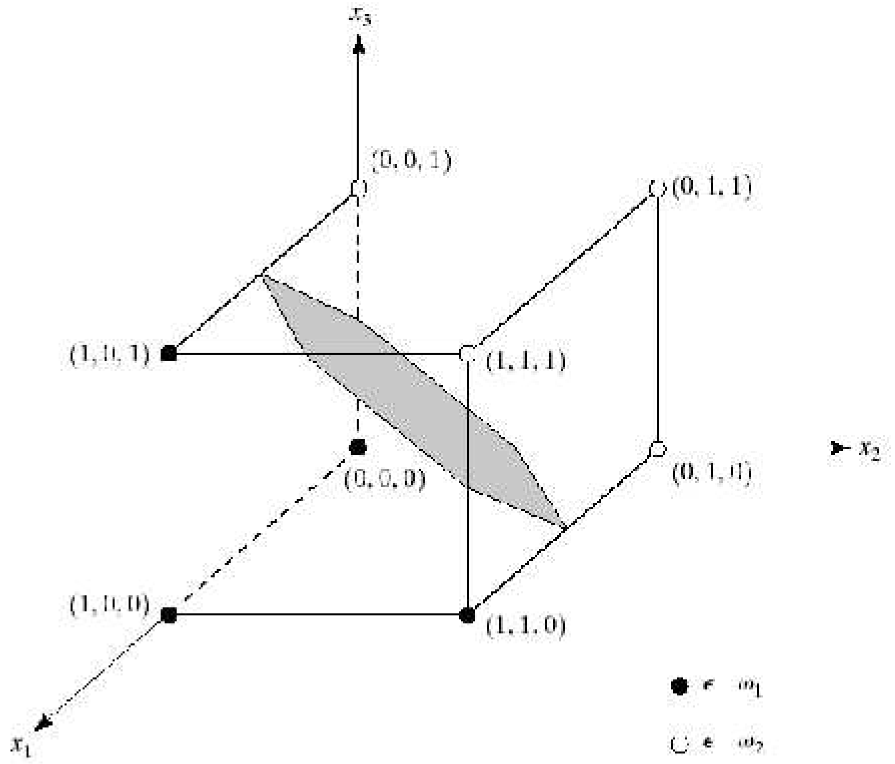
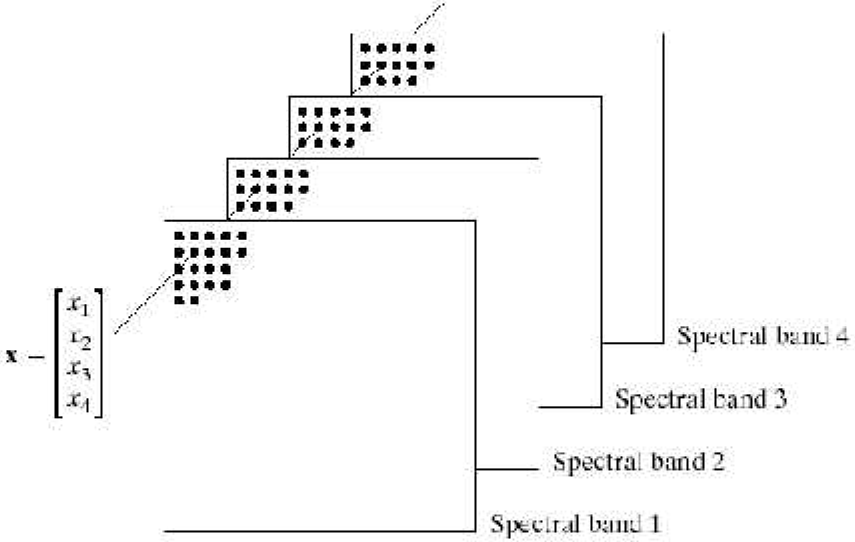


FIGURE 12.11 Two simple pattern classes and their Bayes decision boundary (shown shaded).

Recognition Based on Decision-Theoretic Methods Optimum Statistical Classifiers

- Classification of multi-spectral data using the Bayes classifier

FIGURE 12.12 Formation of a pattern vector from registered pixels of four digital images generated by a multispectral scanner.


$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$
Spectral band 4
Spectral band 3
Spectral band 2
Spectral band 1

Recognition Based on Decision-Theoretic Methods Optimum Statistical Classifiers

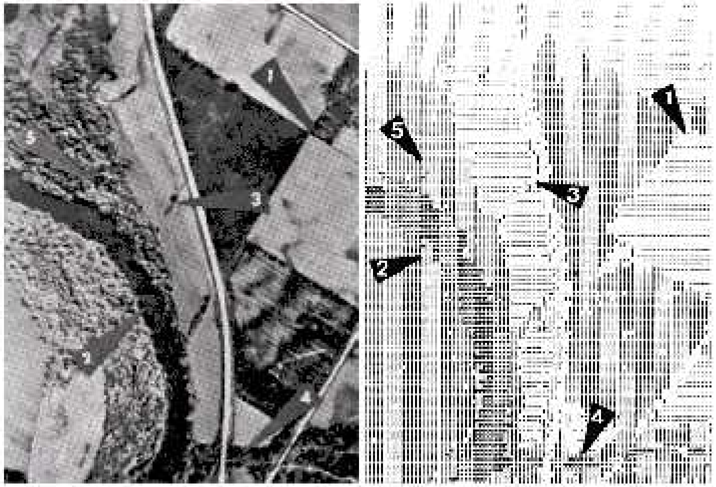
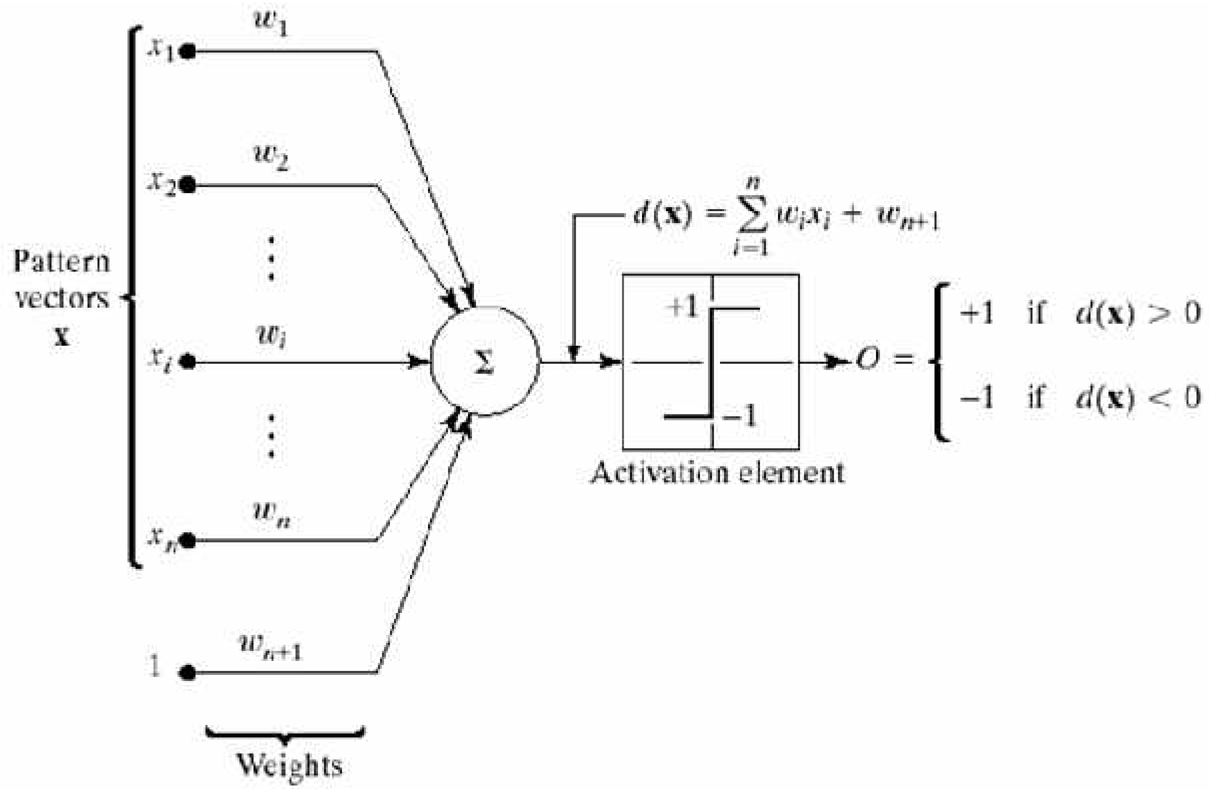


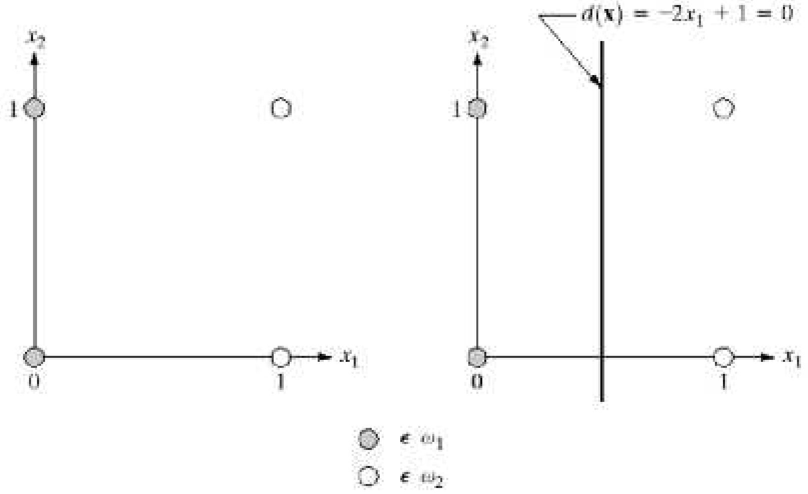
FIGURE 12.13 (a) Multispectral image. (b) Printout of machine classification results using a Bayes classifier. (Courtesy of the Laboratory for Applications of Remote Sensing, Purdue University.)

Recognition Based on Decision-Theoretic Methods Neural Networks


Pattern vectors
$\mathbf{x}$
x_1
x_2
x_i
x_n
1
w_1
w_2
w_i
w_n
w_{n+1}
Weights
Σ
$d(\mathbf{x}) = \sum_{i=1}^{n} w_i x_i + w_{n+1}$
+1
−1
Activation element
$O = \begin{cases} +1 & \text{if} \quad d(\mathbf{x}) > 0 \\ -1 & \text{if} \quad d(\mathbf{x}) < 0 \end{cases}$

Training Alogrithms: Linearly separable classes

- Illustration of Perception algorithm



a b

FIGURE 12.15 (a) Patterns belonging to two classes. (b) Decision boundary determined by training.

Recognition Based on Decision-Theoretic Methods Multilayer Feedforward Neural Networks

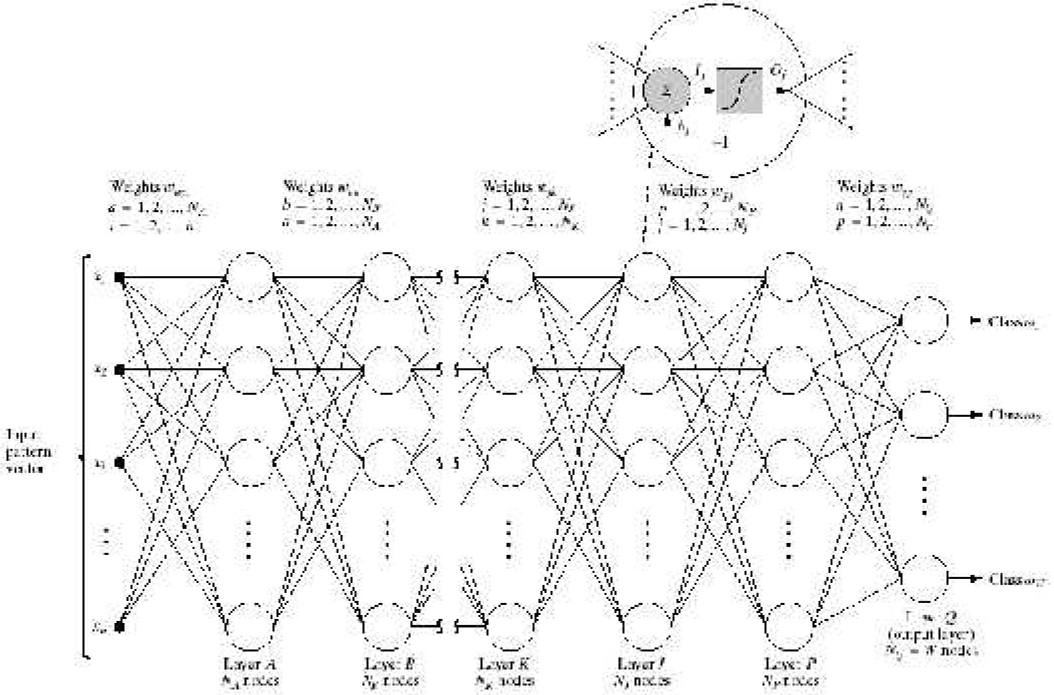


FIGURE 12.16 Multilayer feedforward neural network model. The blowup shows the basic structure of each neuron element throughout the network. The offset θ_j is treated as just another weight.

Recognition Based on Decision-Theoretic Methods Multilayer Feedforward Neural Networks

- **The activation function: a sigmoid function**

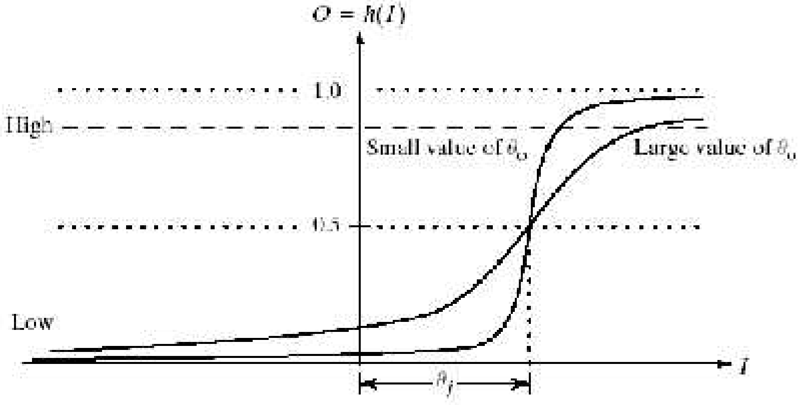
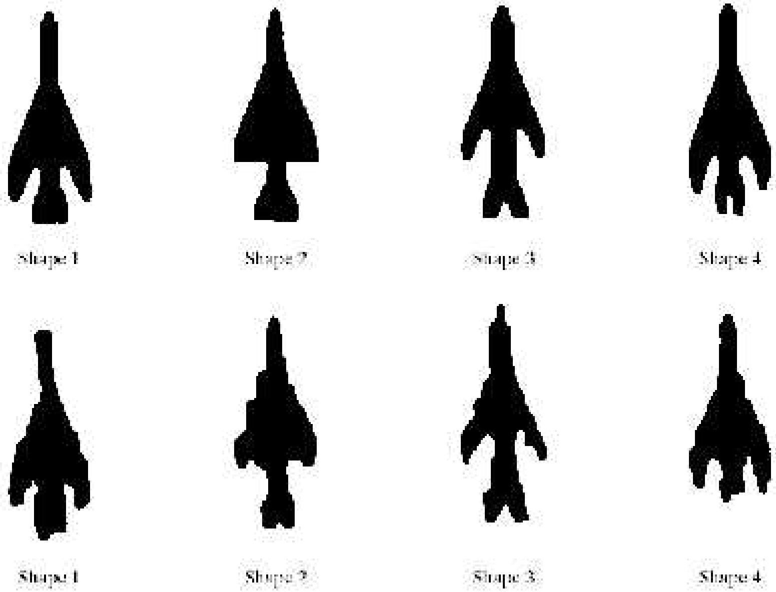


FIGURE 12.17 The sigmoidal activation function of Eq. (12.2-47).

Recognition Based on Decision-Theoretic Methods Multilayer Feedforward Neural Networks

a
b

FIGURE 12.18 (a) Reference shapes and (b) typical noisy shapes used in training the neural network of Fig. 12.19. (Courtesy of Dr. Lalit Gupta, ECE Department, Southern Illinois University.)



- Pattern vectors were generated by computing the normalized signatures of the shapes

Recognition Based on Decision-Theoretic Methods Multilayer Feedforward Neural Networks

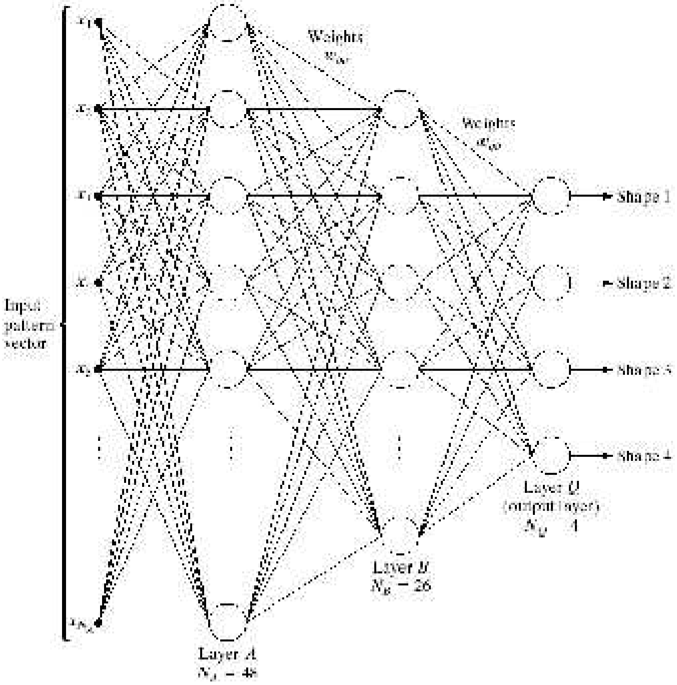


FIGURE 12.19 Three-layer neural network used to recognize the shapes in Fig. 12.18. (Courtesy of Dr. Lalit Gupta, ECE Department, Southern Illinois University.)

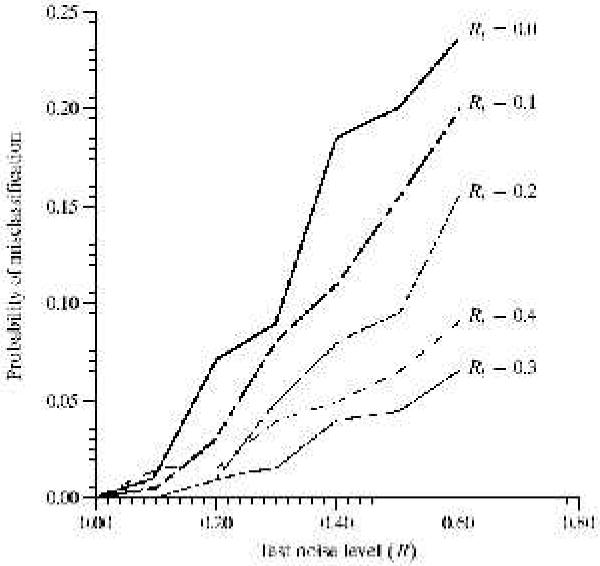
R_t denote a value of R

used to generate

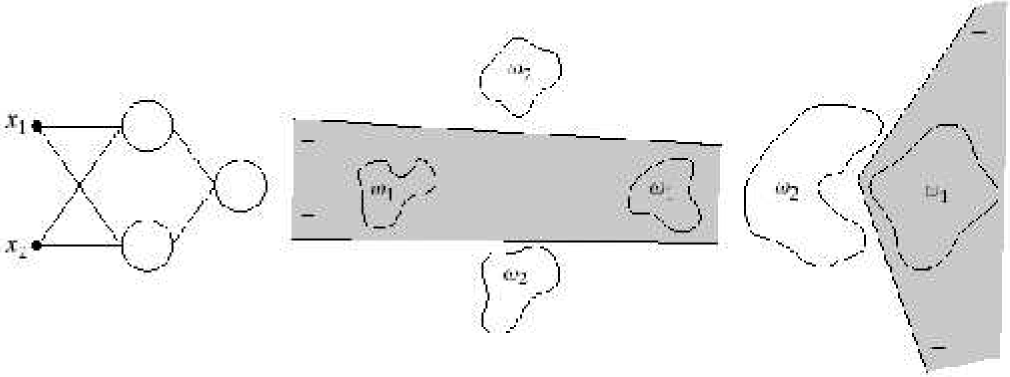
training data.

Rt =0 implies noise-free training

FIGURE 12.20 Performance of the neural network as a function of noise level. (Courtesy of Dr. Lalit Gupta, ECE Department, Southern Illinois University.)

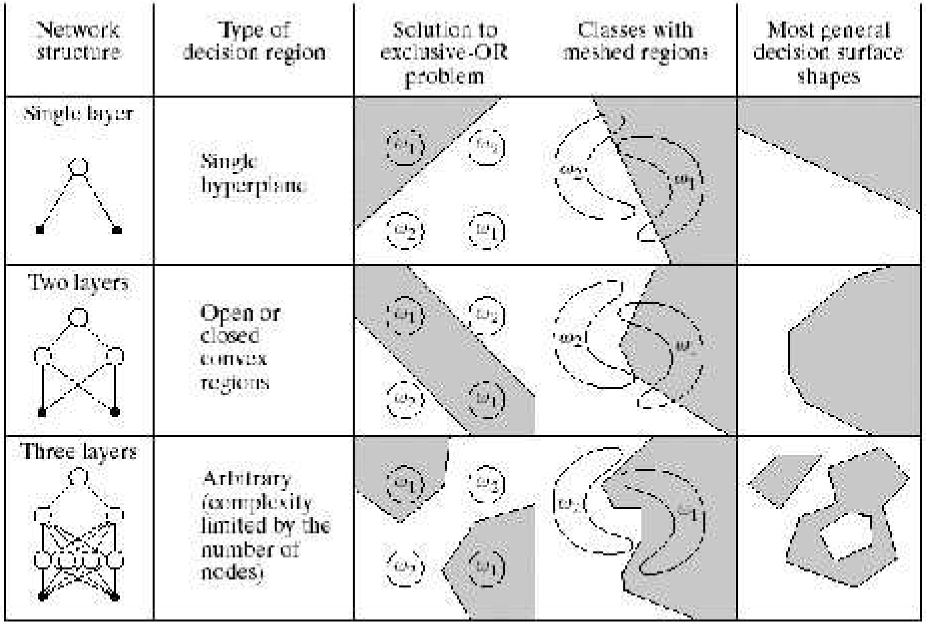


- Complexity of decision surface
    - Two input, tow-layer, feedforward neural networks



a b c

FIGURE 12.22 (a) A two-input, two-layer, feedforward neural network. (b) and (c) Examples of decision boundaries that can be implemented with this network.



Network structure	Type of decision region	Solution to exclusive-OR problem	Classes with meshed regions	Most general decision surface shapes
Single layer	Single hyperplane			
Two layers	Open or closed convex regions			
Three layers	Arbitrary (complexity limited by the number of nodes)			

FIGURE 12.23 Types of decision regions that can be formed by single- and multilayer feed-forward networks with one and two layers of hidden units and two inputs. (Lippman)

Structural Methods Matching Shape Number

- **Let *a* and*b* denote shape numbers of closed boundaries represented by 4-directional chain codes. There two shapes have a degree of similarity *k* if**

$$s_j(a) = s_j(b), \quad \text{for} \quad j = 4,6,8,\ldots,k$$

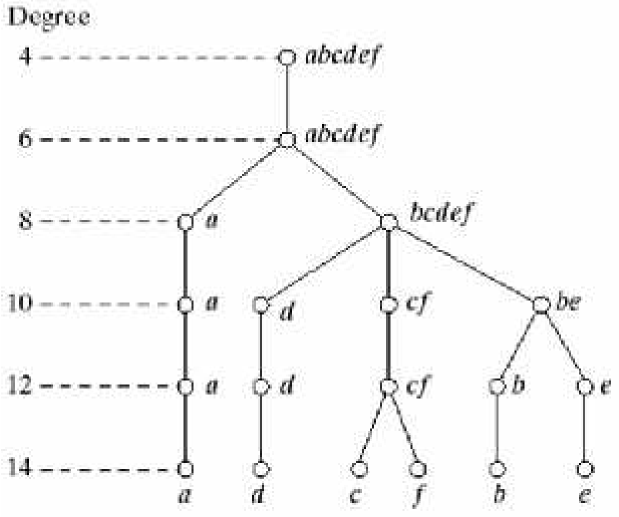
$$s_j(a) \neq s_j(b), \quad \text{for} \quad j = k+2, k+4, \ldots$$

where*s* indicates shape number and the subscript indicates order

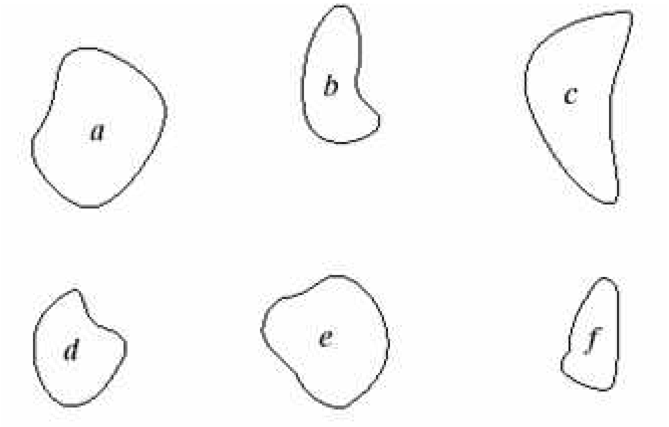
- **The distance between two shapes *a* and *b* defined as**

$$D(a,b) = \frac{1}{k}$$

FIGURE 12.24
(a) Shapes.
(b) Hypothetical similarity tree.
(c) Similarity matrix. (Bribiesca and Guzman.)



Similarity tree



Shape no. order

Structural Methods String Matching

- **Suppose that two region boundaries, *a* and *b*, are coded into strings denoted*a*1, *a*2, …, a_n and *b*1, *b*2, …,b_m, respectively.**
- **Let α represent the number of matches between the two strings, where a match occurs in the *k*th position if $a_k = b_k$.**

$$\beta = \max(|a|,|b|) - \alpha$$

- **A simple measure of similarity between *a* and *b* is the ratio**

$$R = \frac{\alpha}{\beta} = \frac{\alpha}{\max(|a|,|b|) - \alpha}$$

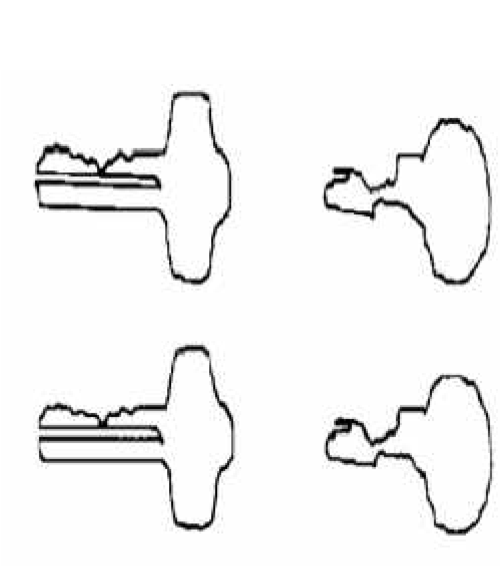
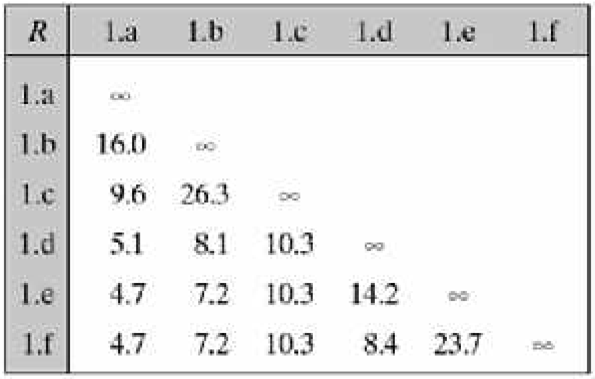


FIGURE 12.25 (a) and (b) Sample boundaries of two different object classes; (c) and (d) their corresponding polygonal approximations; (e)–(g) tabulations of *R*. (Sze and Yang.)

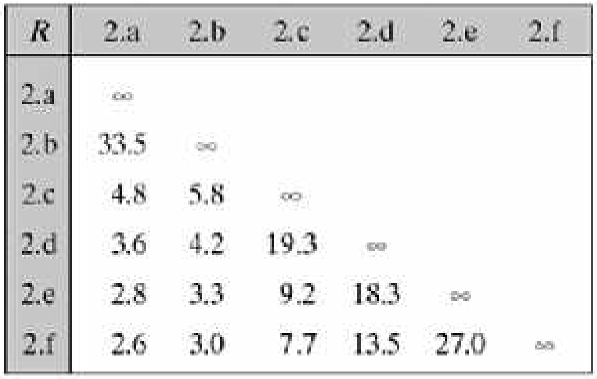
- Strings were formed from the polygons by computing the interior angle, θ , between segments as each polygon was traversed clockwise.
- Angles were coded into one of eight possible symbols, corresponding to 45^o increments.
- Figure 12.25(e) shows the results of computing the measure R for six samples of object 1 against themselves.

$\alpha_1 : 0 < \theta \le 45 ; \alpha_2 : 45 < \theta \le 90 ; ..$

- The notation 1.c refers to the third string from object class 1.

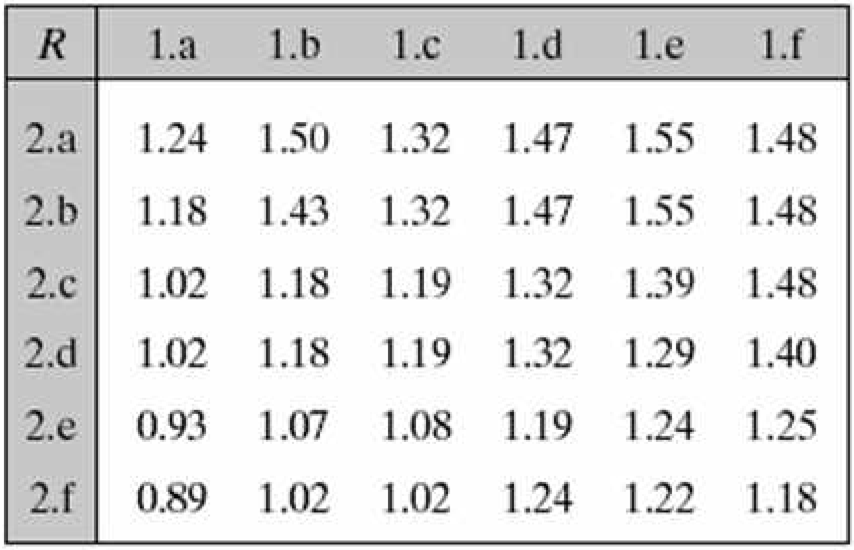


R	1.a	1.b	1.c	1.d	1.e	1.f
1.a	∞					
1.b	16.0	∞				
1.c	9.6	26.3	∞			
1.d	5.1	8.1	10.3	∞		
1.e	4.7	7.2	10.3	14.2	∞	
1.f	4.7	7.2	10.3	8.4	23.7	∞



R	2.a	2.b	2.c	2.d	2.e	2.f
2.a	∞					
2.b	33.5	∞				
2.c	4.8	5.8	∞			
2.d	3.6	4.2	19.3	∞		
2.e	2.8	3.3	9.2	18.3	∞	
2.f	2.6	3.0	7.7	13.5	27.0	∞

- Figure 12.25(g) shows a tabulation of *R* values obtained by comparing strings of one class against the other.
- Note that all *R* values are considerable smaller than any entry in the two preceding tabulations.



R	1.a	1.b	1.c	1.d	1.e	1.f
2.a	1.24	1.50	1.32	1.47	1.55	1.48
2.b	1.18	1.43	1.32	1.47	1.55	1.48
2.c	1.02	1.18	1.19	1.32	1.39	1.48
2.d	1.02	1.18	1.19	1.32	1.29	1.40
2.e	0.93	1.07	1.08	1.19	1.24	1.25
2.f	0.89	1.02	1.02	1.24	1.22	1.18

Figure 12.25 (g)