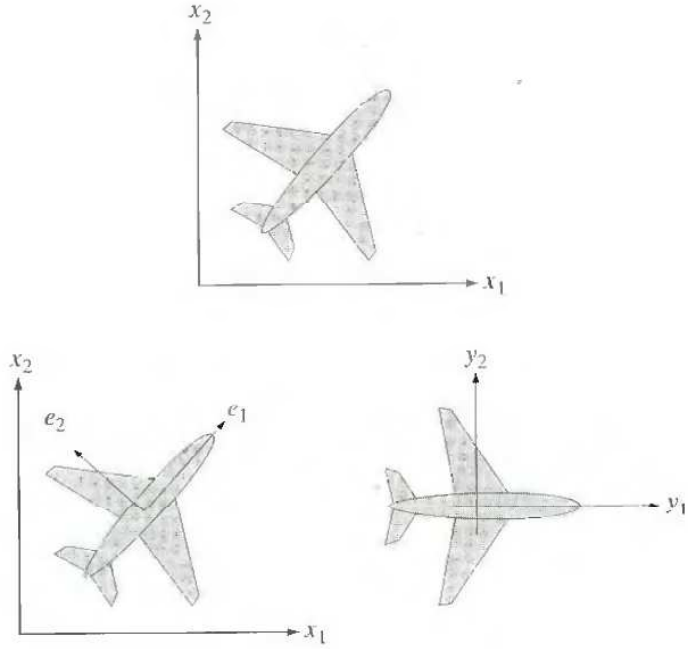


a
b c

FIGURE 11.29 (a) An object. (b) Eigenvectors. (c) Object rotated by using Eq. (11.4-6). The net effect is to align the object along its eigen axes.

as shown in Fig. 11.29(c). In fact, this alignment is precisely the mechanism that decorrelates the data. Furthermore, as the eigenvalues appear along the main diagonal of $\mathbf{C_y}$, λ_i is the variance of component y_i along eigenvector $\mathbf{e}_i$. The two eigenvectors are perpendicular. The y-axes sometimes are called the *eigen axes*, for obvious reasons. ■

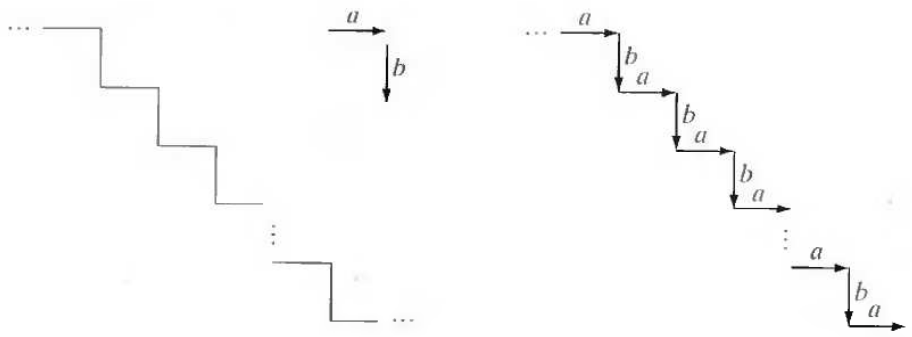
The concept of aligning a 2-D object with its principal eigenvectors plays an important role in description. As noted earlier, description should be as independent as possible to variations in size, translation, and rotation. The ability to *align* the object with its principal axes provides a reliable means for removing the effects of rotation. The eigenvalues are the variances along the eigen axes, and can be used for size normalization. The effects of translation are accounted for by centering the object about its mean, as shown in Eq. (11.4-6). Keep in mind the fact that the method of description derived in this section is equally applicable to both regions and boundaries.

11.5 Relational Descriptors

We introduced in Section 11.3.3 the concept of rewriting rules for describing texture. In this section we expand that concept in the context of relational descriptors. These apply equally well to boundaries or regions, and their main purpose is to capture in the form of rewriting rules basic repetitive patterns in a boundary or region.

a b

FIGURE 11.30 (a) A simple staircase structure. (b) Coded structure.

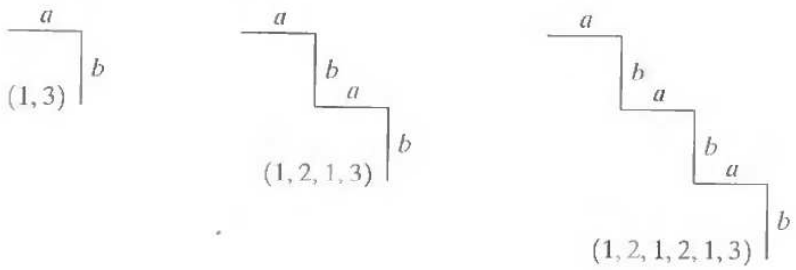


Consider the simple staircase structure shown in Fig. 11.30(a). Assume that this structure has been segmented out of an image and that we want to describe it in some formal way. By defining the two *primitive elements* a and b shown, we may code Fig. 11.30(a) in the form shown in Fig. 11.30(b). The most obvious property of the coded structure is the repetitiveness of the elements a and b. Therefore, a simple description approach is to formulate a recursive relationship involving these primitive elements. One possibility is to use the *rewriting rules*:

(a) $S \rightarrow aA$,
(b) $A \rightarrow bS$, and
(c) $A \rightarrow b$,

where S and A are variables and the elements a and b are constants corresponding to the primitives just defined. Rule 1 indicates that S, called the *starting symbol*, can be replaced by primitive a and variable A. This variable, in turn, can be replaced by b and S or by b alone. Replacing A with bS, leads back to the first rule and the procedure can be repeated. Replacing A with b terminates the procedure, because no variables remain in the expression. Figure 11.31 illustrates some sample derivations of these rules, where the numbers below the structures represent the order in which rules 1, 2, and 3 were applied. The relationship between a and b is preserved, because these rules force an a always to be followed by a b. Notably, these three simple rewriting rules can be used to generate (or describe) infinitely many "similar" structures. As we show in Chapter 12, this approach also has the advantage of a solid theoretical foundation.

FIGURE 11.31 Sample derivations for the rules $S \rightarrow aA$, $A \rightarrow bS$, and $A \rightarrow b$.



Because strings are 1-D structures, their application to image description requires establishing an appropriate method for reducing 2-D positional relations to 1-D form. Most applications of strings to image description are based on the idea of extracting connected line segments from the objects of interest. One approach is to follow the contour of an object and code the result with segments of specified direction and/or length. Figure 11.32 illustrates this procedure.

Another, somewhat more general, approach is to describe sections of an image (such as small homogeneous regions) by directed line segments, which can be joined in other ways besides head-to-tail connections. Figure 11.33(a) illustrates this approach, and Fig. 11.33(b) shows some typical operations that can be defined on abstracted primitives. Figure 11.33(c) shows a set of specific primitives consisting of line segments defined in four directions, and Fig. 11.33(d) shows a step-by-step generation of a specific shape, where $(\sim d)$ indicates the primitive d with its direction reversed. Note that each composite structure has a single head and a single tail. The result of interest is the last string, which describes the complete structure.

String descriptions are best suited for applications in which connectivity of primitives can be expressed in a head-to-tail or other continuous manner. Sometimes regions that are similar in terms of texture or other descriptor may not be contiguous, and techniques are required for describing such situations. One of the most useful approaches for doing so is to use tree descriptors.

A *tree* T is a finite set of one or more nodes for which

(a) there is a unique node \$ designated the *root*, and
(b) the remaining nodes are partitioned into m disjointed sets $T_1, \ldots, T_m$, each of which in turn is a tree called a *subtree* of T.

The *tree frontier* is the set of nodes at the bottom of the tree (the *leaves*), taken in order from left to right. For example, the tree shown in Fig. 11.34 has root \$ and frontier xy.

Generally, two types of information in a tree are important: (1) information about a node stored as a set of words describing the node, and (2) information relating a node to its neighbors, stored as a set of pointers to those neighbors.

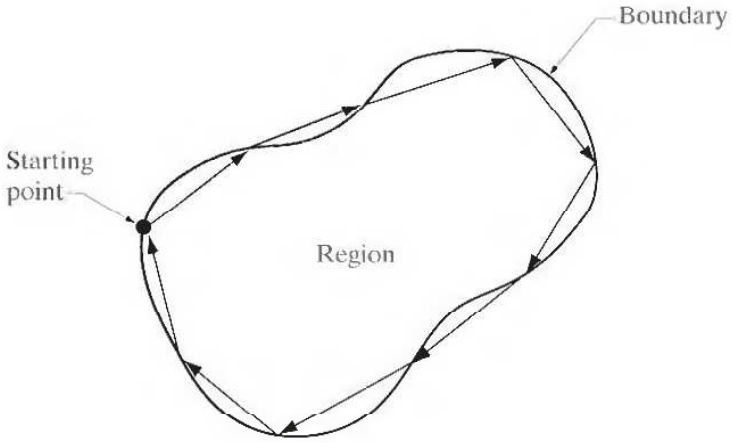
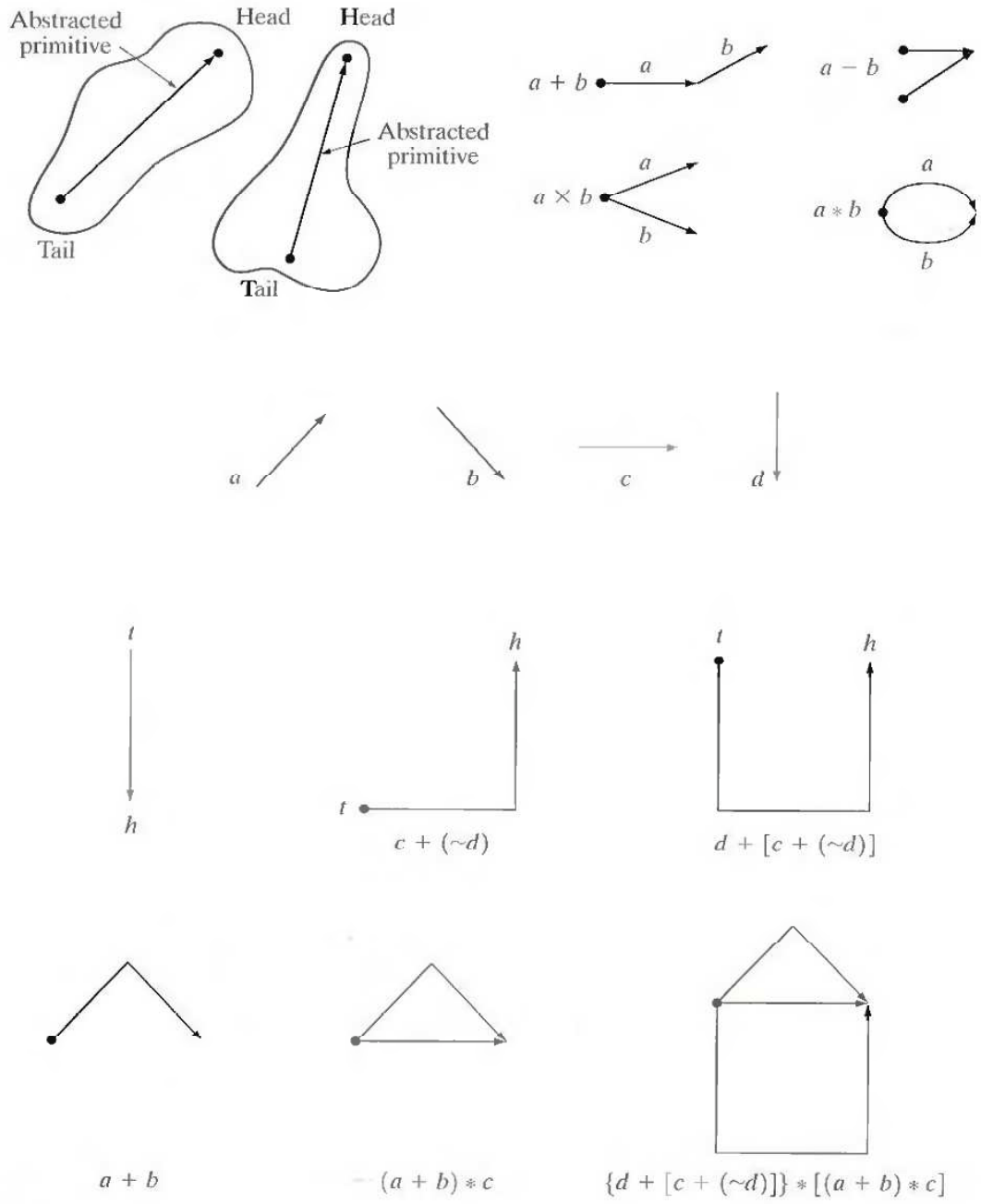


FIGURE 11.32 Coding a region boundary with directed line segments.



a b
c
d

FIGURE 11.33 (a) Abstracted primitives. (b) Operations among primitives. (c) A set of specific primitives. (d) Steps in building a structure.

As used in image description, the first type of information identifies an image substructure (e.g., region or boundary segment), whereas the second type defines the physical relationship of that substructure to other substructures. For example, Fig. 11.35(a) can be represented by a tree by using the relationship "inside of." Thus, if the root of the tree is denoted $, Fig. 11.35(a) shows that the first

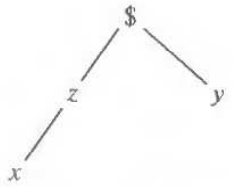
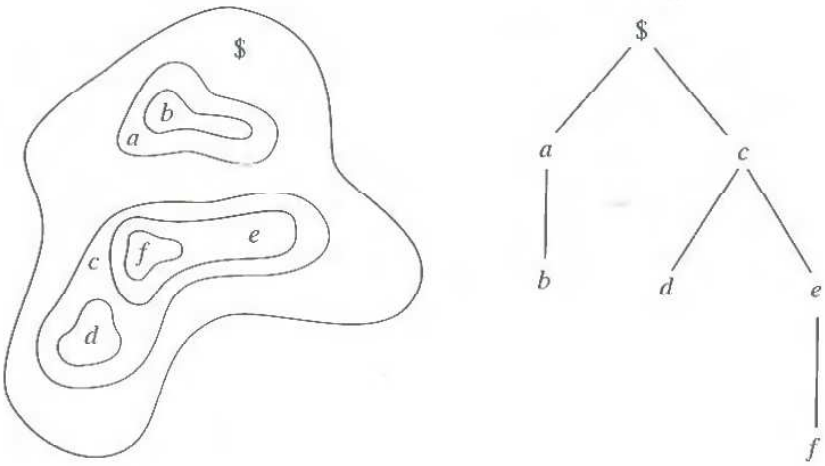


FIGURE 11.34 A simple tree with root $ and frontier xy.



a b

FIGURE 11.35 (a) A simple composite region. (b) Tree representation obtained by using the relationship "inside of."

level of complexity involves a and c inside $, which produces two branches emanating from the root, as shown in Fig. 11.35(b). The next level involves b inside a, and d and e inside c. Finally, f inside e completes the tree.

Summary

The representation and description of objects or regions that have been segmented out of an image are early steps in the operation of most automated processes involving images. These descriptions, for example, constitute the input to the object recognition methods developed in the following chapter. As indicated by the range of description techniques covered in this chapter, the choice of one method over another is determined by the problem under consideration. The objective is to choose descriptors that "capture" essential differences between objects, or classes of objects, while maintaining as much independence as possible to changes in factors such as location, size, and orientation.

References and Further Reading

The chain-code representation discussed in Section 11.1.1 was first proposed by Freeman [1961, 1974]. For current work using chain codes see Bribiesca [1999], who also has extended chain codes to 3-D (Bribiesca [2000]). For a detailed discussion and algorithm to compute minimum-perimeter polygons (Section 11.1.2) see Sklansky et al. [1972]. Typical

work on polygonal approximations a decade ago is illustrated in the papers by Bengtsoon and Eklundh [1991] and by Sato [1992]. The paper by Zhu and Chirlian [1995] presents an interesting approach to the detection of point inflections along a curve. See also Hu and Yan [1997]. More recent work in this area focuses on invariant polygonal fitting (Voss and Suesse [1997]), on methods for evaluating the performance of polygonal approximation algorithms (Rosin [1997]), on generic implementations (Huang and Sun [1999]), and on computational speed (Davis [1999]).

References for the discussion of signatures (Section 11.1.3) are Ballard and Brown [1982] and Gupta and Srinath [1988]. See Preparata and Shamos [1985] regarding fundamental formulations for finding the convex hull and convex deficiency (Section 11.1.4). See also the paper by Liu-Yu and Antipolis [1993]. Katzir et al. [1994] discuss the detection of partially occluded curves. Zimmer et al. [1997] discuss an improved algorithm for computing the convex hull, and Latecki and Lakämper [1999] discuss a convexity rule for shape decomposition.

The skeletonizing algorithm discussed in Section 11.1.5 is based on Zhang and Suen [1984]. Some useful additional comments on the properties and implementation of this algorithm are included in a paper by Lu and Wang [1986]. A paper by Jang and Chin [1990] provides an interesting tie between the discussion in Section 11.1.5 and the morphological concept of thinning introduced in Section 9.5.5. For thinning approaches in the presence of noise see Shi and Wong [1994] and Chen and Yu [1996]. Shaked and Bruckstein [1998] discuss a pruning algorithm useful for removing spurs from a skeleton. Fast computation of the medial axis transform is discussed by Sahni and Jenq [1992] and by Ferreira and Ubéda [1999]. The survey paper by Loncaric [1998] is of interest regarding many of the approaches discussed in Section 11.1.

Freeman and Shapira [1975] give an algorithm for finding the basic rectangle of a closed, chain-coded curve (Section 11.2.1). The discussion on shape numbers in Section 11.2.2 is based on the work of Bribiesca and Guzman [1980] and Bribiesca [1981]. For additional reading on Fourier descriptors (Section 11.2.3), see the early papers by Zahn and Roskies [1972] and by Persoon and Fu [1977]. See also Aguado et al. [1998] and Sonka et al. [1999]. Reddy and Chatterji [1996] discuss an interesting approach using the FFT to achieve invariance to translation, rotation, and scale change. The material in Section 11.2.4 is based on elementary probability theory (see, for example, Peebles [1993] and Popoulis [1991]).

For additional reading on Section 11.3.2, see Rosenfeld and Kak [1982] and Ballard and Brown [1982]. For an excellent introduction to texture (Section 11.3.3), see Haralick and Shapiro [1992]. For an early survey on texture, see Wechsler [1980]. The papers by Murino et al. [1998] and Garcia [1999], and the discussion by Shapiro and Stockman [2001], are representative of current work in this field.

The moment invariant approach discussed in Section 11.3.4 is from Hu [1962]. Also see Bell [1965]. To get an idea of the range of applications of moment invariants, see Hall [1979] regarding image matching and Cheung and Teoh [1999] regarding the use of moments for describing symmetry. Moment invariants were generalized to n dimensions by Mamistvalov [1998].

Hotelling [1933] was the first to derive and publish the approach that transforms discrete variables into uncorrelated coefficients. He referred to this technique as *the method of principal components*. His paper gives considerable insight into the method and is worth reading. Hotelling's transformation was rediscovered by Kramer and Mathews [1956] and by Huang and Schultheiss [1963]. Principal components are still a basic tool for image description used in numerous applications, as exemplified by Swets and Weng [1996] and by Duda, Heart, and Stork [2001]. References for the material in Section 11.5 are Gonzalez and Thomason [1978] and Fu [1982]. See also Sonka et al. [1999].

Problems

11.1 ★ **(a)** Show that redefining the starting point of a chain code so that the resulting sequence of numbers forms an integer of minimum magnitude makes the code independent of the initial starting point on the boundary.

(b) Find the normalized starting point of the code 11076765543322.

11.2 **(a)** Show that the first difference of a chain code normalizes it to rotation, as explained in Section 11.1.1.

(b) Compute the first difference of the code 0101030303323232212111.

11.3 ★ **(a)** Show that the rubber-band polygonal approximation approach discussed in Section 11.1.2 yields a polygon with minimum perimeter.

(b) Show that if each cell corresponds to a pixel on the boundary, the maximum possible error in that cell is $\sqrt{2}d$, where d is the minimum possible horizontal or vertical distance between adjacent pixels (i.e., the distance between lines in the sampling grid used to produce the digital image).

11.4 ★ **(a)** Discuss the effect on the resulting polygon if the error threshold is set to zero in the merging method discussed in Section 11.1.2.

(b) What would be the effect on the splitting method?

11.5 ★ **(a)** Plot the signature of a square boundary using the tangent angle method discussed in Section 11.1.3.

(b) Repeat for the slope density function.

Assume that the square is aligned with the x- and y-axes, and let the x-axis be the reference line. Start at the corner closest to the origin.

11.6 Find an expression for the signature of each of the following boundaries, and plot the signatures.

★ **(a)** An equilateral triangle

(b) A rectangle

(c) An ellipse

11.7 Draw the medial axis of

★ **(a)** A circle

★ **(b)** A square

(c) A rectangle

(d) An equilateral triangle

11.8 For each of the figures shown,

★ **(a)** Discuss the action taken at point p by step 1 of the skeletonizing algorithm presented in Section 11.1.5.

(b) Repeat for step 2 of the algorithm. Assume that $p = 1$ in all cases.

1	1	0
1	p	0
1	1	0

0	0	0
1	p	0
0	0	0

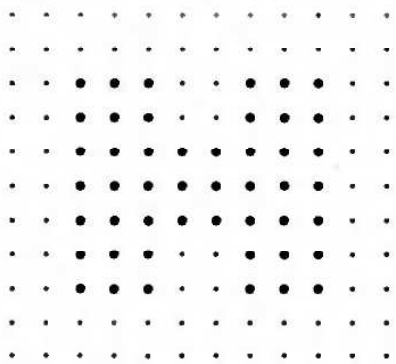
0	1	0
1	p	1
0	1	0

1	1	0
0	p	1
0	0	0

11.9 With reference to the skeletonizing algorithm in Section 11.1.5, what would the figure shown look like after

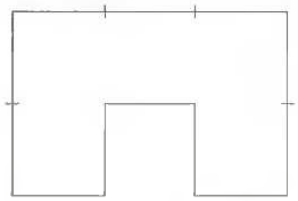
★ **(a)** One pass of step 1 of the algorithm?

(b) One pass of step 2 (on the result of step 1, not the original image)?



11.10 ★ **(a)** What is the order of the shape number for the figure shown?

(b) Obtain the shape number.



11.11 The procedure discussed in Section 11.2.3 for using Fourier descriptors consists of expressing the coordinates of a contour as complex numbers, taking the DFT of these numbers, and keeping only a few components of the DFT as descriptors of the boundary shape. The inverse DFT is then an approximation to the original contour. What class of contour shapes would have a DFT consisting of real numbers and how would the axis system in Fig. 11.13 have to be set up to obtain these real numbers?

★ **11.12** Give the smallest number of statistical moment descriptors needed to differentiate between the signatures of the figures shown in Fig. 11.5.

11.13 Give two boundary shapes that have the same mean and third statistical moment descriptors, but different second moments.

★ **11.14** Propose a set of descriptors capable of differentiating between the shapes of the characters 0, 1, 8, 9, and X. (*Hint:* Use topological descriptors in conjunction with the convex hull.)

11.15 Consider a checkerboard image composed of alternating black and white squares, each of size $m \times m$. Give a position operator that would yield a diagonal co-occurrence matrix.

11.16 Obtain the gray-level co-occurrence matrix of a 5×5 image composed of a checkerboard of alternating 1's and 0's if

★ **(a)** the position operator P is defined as "one pixel to the right," and

(b) "two pixels to the right."

Assume that the top left pixel has value 0.

11.17 Prove the validity of Eqs. (11.4-7), (11.4-8), and (11.4-9).

★ **11.18** It was mentioned in Example 11.10 that a credible job could be done of reconstructing approximations to the six original images by using only the two

principal-component images associated with the largest eigenvalues. What would be the mean square error incurred in doing so? Express your answer as a percentage of the maximum possible error.

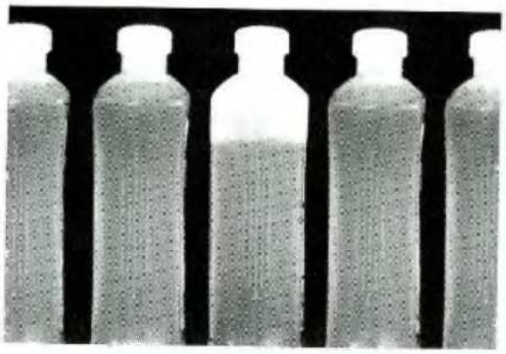
11.19 For a set of images of size 64×64, assume that the covariance matrix given in Eq. (11.4-9) turns out to be the identity matrix. What would be the mean square error between the original images and images reconstructed using Eq. (11.4-11) with only half of the original eigenvectors?

★11.20 Under what conditions would you expect the major axes of a boundary, defined in Section 11.2.1, to be equal to the eigen axes of that boundary?

11.21 Give a spatial relationship and corresponding tree representation for a checkerboard pattern of black and white squares. Assume that the top left element is black and that the root of the tree corresponds to that element. Your tree can have no more than two branches emanating from each node.

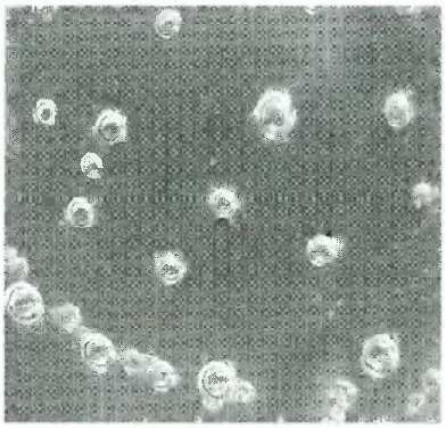
★11.22 You are contracted to design an image processing system for detecting imperfections on the inside of certain solid plastic wafers. The wafers are examined using an X-ray imaging system, which yields 8-bit images of 512×512 resolution. In the absence of imperfections, the images appear "bland," having a mean gray level of 100 and variance of 400. The imperfections appear as bloblike regions in which about 70% of the pixels have excursions in intensity of 50 gray levels or less about a mean of 100. A wafer is considered defective if such a region occupies an area exceeding 20×20 pixels in size. Propose a system based on texture analysis.

11.23 A company that bottles a variety of industrial chemicals has heard of your success solving imaging problems and hires you to design an approach for detecting when bottles are not full. The bottles appear as shown in the following figure as they move along a conveyor line past an automatic filling and capping station. A bottle is considered imperfectly filled when the level of the liquid is below the midway point between the bottom of the neck and the shoulder of the bottle. The shoulder is defined as the region of the bottle where the sides and slanted portion of the bottle intersect. The bottles are moving, but the company has an imaging system equipped with a illumination flash front end that effectively stops motion, so you will be given images that look very close to the sample shown here. Based on the material you have learned up to this point, propose a solution for detecting bottles that are not filled properly. State clearly all assumptions that you make and that are likely to impact the solution you propose.



11.24 Having heard about your success with the bottling problem, you are contacted by a fluids company that wishes to automate bubble-counting in certain processes for quality control. The company has solved the imaging problem and can obtain 8-bit images of resolution 700×700 pixels, such as the ones shown. Each

image represents an area of 7 cm^2. The company wishes to do two things with each image: (1) Determine the ratio of the area occupied by bubbles to the total area of the image, and (2) count the number of distinct bubbles. Based on the material you have learned up to this point, propose a solution to this problem. In your solution, make sure to state the physical dimensions of the smallest bubble your solution can detect. State clearly all assumptions that you make and that are likely to impact the solution you propose.



12 Object Recognition

One of the most interesting aspects of the world is that it can be considered to be made up of patterns.
A pattern is essentially an arrangement. It is characterized by the order of the elements of which it is made, rather than by the intrinsic nature of these elements.

Norbert Wiener

Preview

We conclude our coverage of digital image processing with an introduction to techniques for object recognition. As noted in Section 1.1, we have defined the scope covered by our treatment of digital image processing to include recognition of *individual* image regions, which in this chapter we call *objects* or *patterns*.

The approaches to pattern recognition developed in this chapter are divided into two principal areas: decision-theoretic and structural. The first category deals with patterns described using quantitative descriptors, such as length, area, and texture. The second category deals with patterns best described by qualitative descriptors, such as the relational descriptors discussed in Section 11.5.

Central to the theme of recognition is the concept of "learning" from sample patterns. Learning techniques for both decision-theoretic and structural approaches are developed and illustrated in the material that follows.

12.1 Patterns and Pattern Classes

A *pattern* is an *arrangement of descriptors*, such as those discussed in Chapter 11. The name *feature* is used often in the pattern recognition literature to denote a descriptor. A *pattern class* is a family of patterns that share some common properties. Pattern classes are denoted $\omega_1, \omega_2, \ldots, \omega_W$, where W is the number of classes. Pattern recognition by machine involves techniques for assigning

patterns to their respective classes—automatically and with as little human intervention as possible.

Three common pattern arrangements used in practice are vectors (for quantitative descriptions) and strings and trees (for structural descriptions). Pattern vectors are represented by bold lowercase letters, such as **x**, **y**, and **z**, and take the form

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \tag{12.1-1}$$

where each component, x_i, represents the ith descriptor and n is the total number of such descriptors associated with the pattern. Pattern vectors are represented as columns (that is, $n \times 1$ matrices). Hence a pattern vector can be expressed in the form shown in Eq. (12.1-1) or in the equivalent form $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$, where T indicates transposition. The reader will recognize this notation from Section 11.4.

See inside front cover

Consult the book web site for a brief review of vectors and matrices.

The nature of the components of a pattern vector **x** depends on the approach used to describe the physical pattern itself. Let us illustrate with an example that is both simple and gives a sense of history in the area of classification of measurements. In a classic paper, Fisher [1936] reported the use of what then was a new technique called *discriminant analysis* (discussed in Section 12.2) to recognize three types of iris flowers (*Iris setosa*, *virginica*, and *versicolor*) by measuring the widths and lengths of their petals (Fig. 12.1). In our present

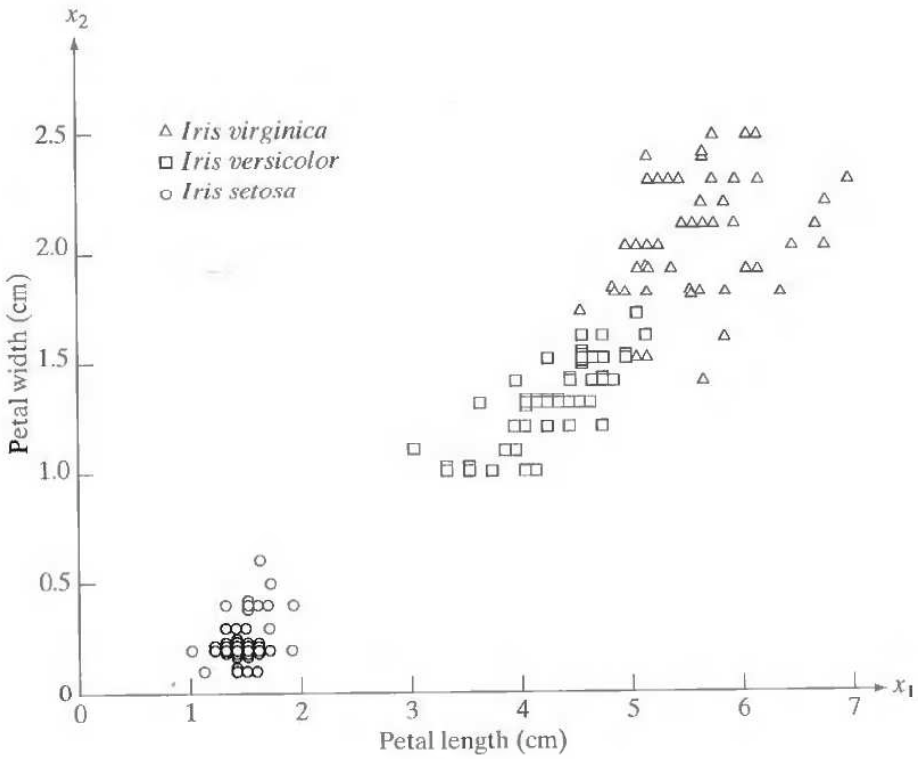


FIGURE 12.1 Three types of iris flowers described by two measurements.

terminology, each flower is *described* by two measurements, which leads to a 2-D pattern vector of the form

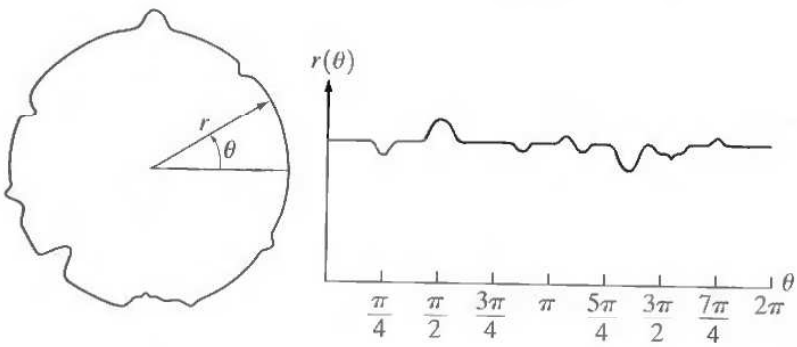
$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{12.1-2}$$

where x_1 and x_2 correspond to petal length and width, respectively. The three pattern classes in this case, denoted ω_1, ω_2, and ω_3, correspond to the varieties *setosa*, *virginica*, and *versicolor*, respectively.

Because the petals of flowers vary in width and length, the pattern vectors describing these flowers also will vary, not only between different classes, but also within a class. Figure 12.1 shows length and width measurements for several samples of each type of iris. After a set of measurements has been selected (two in this case), the components of a pattern vector become the entire description of each physical sample. Thus each flower in this case becomes a point in 2-D Euclidean space. We also note that measurements of petal width and length in this case adequately separated the class of *Iris setosa* from the other two but did not separate as successfully the *virginica* and *versicolor* types from each other. This result illustrates the classic *feature selection* problem, in which the degree of class separability depends strongly on the choice of descriptors selected for an application. We say considerably more about this issue in Sections 12.2 and 12.3.

Figure 12.2 shows another example of pattern vector generation. In this case, we are interested in different types of noisy shapes, a sample of which is shown in Fig. 12.2(a). If we elect to represent each object by its signature (see Section 11.1.3), we would obtain 1-D signals of the form shown in Fig. 12.2(b). Suppose that we elect to describe each signature simply by its sampled amplitude values; that is, we sample the signatures at some specified interval values of θ, denoted $\theta_1, \theta_2, \ldots, \theta_n$. Then we can form pattern vectors by letting $x_1 = r(\theta_1)$, $x_2 = r(\theta_2), \ldots, x_n = r(\theta_n)$. These vectors become points in n-dimensional Euclidean space, and pattern classes can be imagined to be "clouds" in n dimensions.

Instead of using signature amplitudes directly, we could compute, say, the first n statistical moments of a given signature (Section 11.2.4) and use these descriptors as components of each pattern vector. In fact, as may be evident by now, pattern vectors can be generated in numerous other ways. We present some



a b

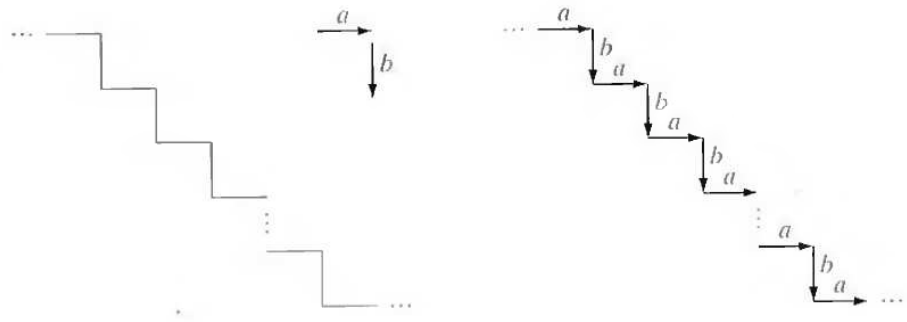
FIGURE 12.2 A noisy object and its corresponding signature.

of them throughout this chapter. For the moment, the key concept to keep in mind is that selecting the descriptors on which to base each component of a pattern vector has a profound influence on the eventual performance of object recognition based on the pattern vector approach.

The techniques just described for generating pattern vectors yield pattern classes characterized by quantitative information. In some applications, pattern characteristics are best described by structural relationships. For example, fingerprint recognition is based on the interrelationships of print features called *minutiae*. Together with their relative sizes and locations, these features are primitive components that describe fingerprint ridge properties, such as abrupt endings, branching, merging, and disconnected segments. Recognition problems of this type, in which not only quantitative measures about each feature but also the spatial relationships between the features determine class membership, generally are best solved by structural approaches. This subject was introduced in Section 11.5. We revisit it briefly here in the context of pattern descriptors.

Figure 12.3(a) shows a simple staircase pattern. This pattern could be sampled and expressed in terms of a pattern vector, similar to the approach used in Fig. 12.2. However, the basic structure, consisting of repetitions of two simple primitive elements, would be lost in this method of description. A more meaningful description would be to define the elements a and b and let the pattern be the string of symbols $w = \ldots ababababab \ldots$, as shown in Fig. 12.3(b). The structure of this particular class of patterns is captured in this description by requiring that connectivity be defined in a head-to-tail manner, and by allowing only alternating symbols. This structural construct is applicable to staircases of any length but excludes other types of structures that could be generated by other combinations of the primitives a and b.

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. A more powerful approach for many applications is the use of tree descriptions, as defined in Section 11.5. Basically, most hierarchical ordering schemes lead to tree structures. For example, Fig. 12.4 is a satellite image of a heavily built downtown area and surrounding



a b

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

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

residential areas. Let us define the entire image area by the symbol $. The (upside down) tree representation shown in Fig. 12.5 was obtained by using the structural relationship "composed of." Thus the root of the tree represents the entire image. The next level indicates that the image is composed of a downtown and residential area. The residential area, in turn is composed of housing, highways, and shopping malls. The next level down further describes the housing and highways. We can continue this type of subdivision until we reach the limit of our ability to resolve different regions in the image.

We develop in the following sections recognition approaches for objects described by all the techniques discussed in the preceding paragraphs.

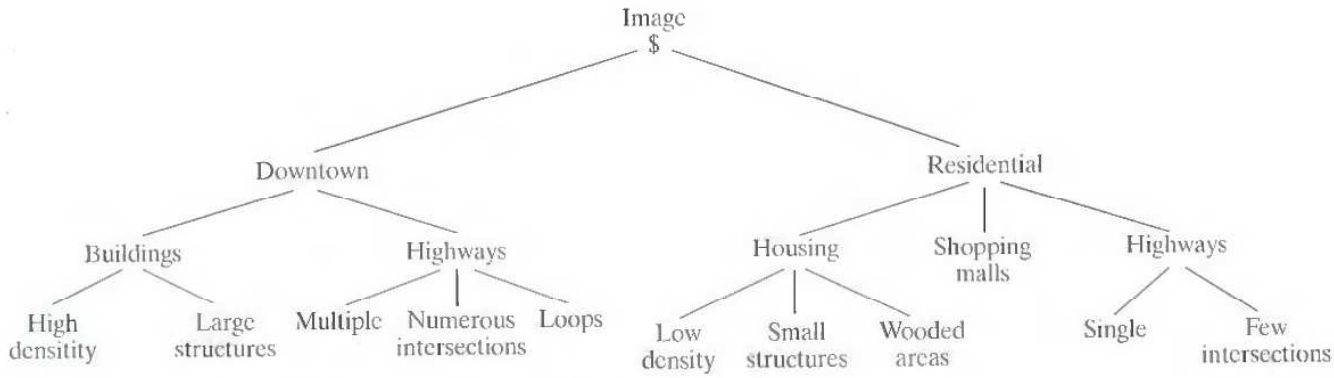


FIGURE 12.5 A tree description of the image in Fig. 12.4.

12.2 Recognition Based on Decision-Theoretic Methods

Decision-theoretic approaches to recognition are based on the use of *decision* (or *discriminant*) *functions*. Let $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$ represent an n-dimensional pattern vector, as discussed in Section 12.1. For W pattern classes $\omega_1, \omega_2, \ldots, \omega_W$, the basic problem in decision-theoretic pattern recognition is to find W decision functions $d_1(\mathbf{x}), d_2(\mathbf{x}), \ldots, d_W(\mathbf{x})$ with the property that, if a pattern $\mathbf{x}$ belongs to class ω_i, then

$$d_i(\mathbf{x}) > d_j(\mathbf{x}) \qquad j = 1, 2, \ldots, W; j \neq i. \tag{12.2-1}$$

In other words, an unknown pattern $\mathbf{x}$ is said to belong to the ith pattern class if, upon substitution of $\mathbf{x}$ into all decision functions, $d_i(\mathbf{x})$ yields the largest numerical value. Ties are resolved arbitrarily.

The *decision boundary* separating class ω_i from ω_j is given by values of $\mathbf{x}$ for which $d_i(\mathbf{x}) = d_j(\mathbf{x})$ or, equivalently, by values of $\mathbf{x}$ for which

$$d_i(\mathbf{x}) - d_j(\mathbf{x}) = 0. \tag{12.2-2}$$

Common practice is to identify the decision boundary between two classes by the single function $d_{ij}(\mathbf{x}) = d_i(\mathbf{x}) - d_j(\mathbf{x}) = 0$. Thus $d_{ij}(\mathbf{x}) > 0$ for patterns of class ω_i and $d_{ij}(\mathbf{x}) < 0$ for patterns of class ω_j. The principal objective of the discussion in this section is to develop various approaches for finding decision functions that satisfy Eq. (12.2-1).

12.2.1 Matching

Recognition techniques based on matching represent each class by a prototype pattern vector. An unknown pattern is assigned to the class to which it is closest in terms of a predefined metric. The simplest approach is the minimum-distance classifier, which, as its name implies, computes the (Euclidean) distance between the unknown and each of the prototype vectors. It chooses the smallest distance to make a decision. We also discuss an approach based on correlation, which can be formulated directly in terms of images and is quite intuitive.

Minimum distance classifier

Suppose that we define the prototype of each pattern class to be the mean vector of the patterns of that class:

$$\mathbf{m}_j = \frac{1}{N_j} \sum_{x \in \omega_j} \mathbf{x}_j \qquad j = 1, 2, \ldots, W \tag{12.2-3}$$

where N_j is the number of pattern vectors from class ω_j and the summation is taken over these vectors. As before, W is the number of pattern classes. One way to determine the class membership of an unknown pattern vector $\mathbf{x}$ is to assign it to the class of its closest prototype, as noted previously. Using the Euclidean distance to determine closeness reduces the problem to computing the distance measures:

$$D_j(\mathbf{x}) = \|\mathbf{x} - \mathbf{m}_j\| \qquad j = 1, 2, \ldots, W \tag{12.2-4}$$

where $\|\mathbf{a}\| = (\mathbf{a}^T\mathbf{a})^{1/2}$ is the Euclidean norm. We then assign $\mathbf{x}$ to class ω_i if $D_i(\mathbf{x})$ is the smallest distance. That is, the smallest distance implies the best match in this formulation. It is not difficult to show (Problem 12.2) that selecting the smallest distance is equivalent to evaluating the functions

$$d_j(\mathbf{x}) = \mathbf{x}^T\mathbf{m}_j - \frac{1}{2}\mathbf{m}_j^T\mathbf{m}_j \qquad j = 1, 2, \ldots, W \tag{12.2-5}$$

and assigning $\mathbf{x}$ to class ω_i if $d_i(\mathbf{x})$ yields the largest numerical value. This formulation agrees with the concept of a decision function, as defined in Eq. (12.2-1).

From Eqs. (12.2-2) and (12.2-5), the decision boundary between classes ω_i and ω_j for a minimum distance classifier is

$$\begin{aligned} d_{ij}(\mathbf{x}) &= d_i(\mathbf{x}) - d_j(\mathbf{x}) \\ &= \mathbf{x}^T(\mathbf{m}_i - \mathbf{m}_j) - \frac{1}{2}(\mathbf{m}_i - \mathbf{m}_j)^T(\mathbf{m}_i - \mathbf{m}_j) = 0. \end{aligned} \tag{12.2-6}$$

The surface given by Eq. (12.2-6) is the perpendicular bisector of the line segment joining $\mathbf{m}_i$ and $\mathbf{m}_j$ (see Problem 12.3). For $n = 2$, the perpendicular bisector is a line, for $n = 3$ it is a plane, and for $n > 3$ it is called a *hyperplane*.

EXAMPLE 12.1: Illustration of the minimum-distance classifier.

■ Figure 12.6 shows two pattern classes extracted from the iris samples in Fig. 12.1. The two classes, *Iris versicolor* and *Iris setosa*, denoted ω_1 and ω_2, respectively, have sample mean vectors $\mathbf{m}_1 = (4.3, 1.3)^T$ and $\mathbf{m}_2 = (1.5, 0.3)^T$. From Eq. (12.2-5), the decision functions are

$$\begin{aligned} d_1(\mathbf{x}) &= \mathbf{x}^T\mathbf{m}_1 - \frac{1}{2}\mathbf{m}_1^T\mathbf{m}_1 \\ &= 4.3x_1 + 1.3x_2 - 10.1 \end{aligned}$$

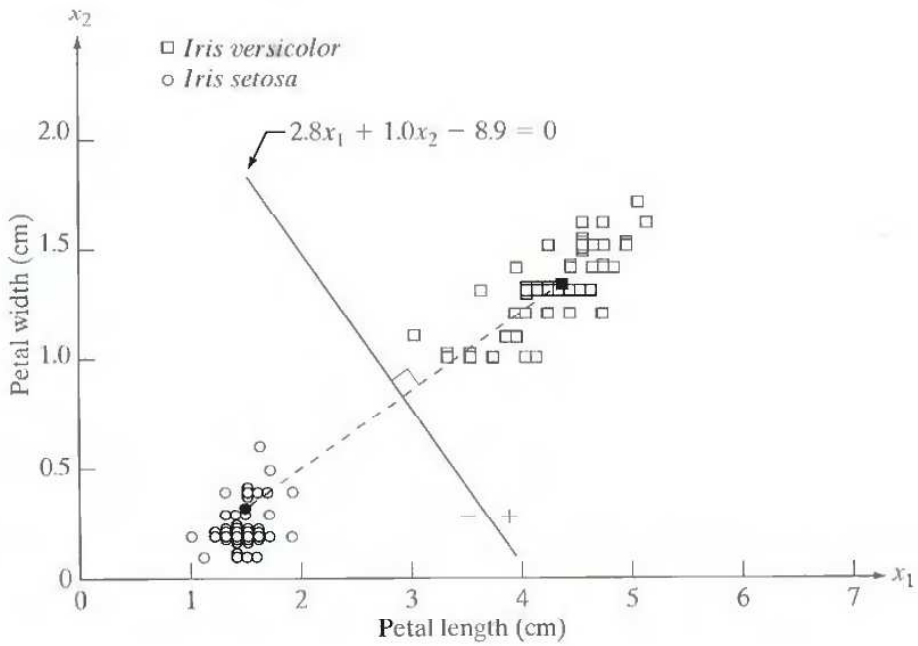


FIGURE 12.6 Decision boundary of minimum distance classifier for the classes of *Iris versicolor* and *Iris setosa*. The dark dot and square are the means.

and

$$d_2(\mathbf{x}) = \mathbf{x}^T\mathbf{m}_2 - \frac{1}{2}\mathbf{m}_2^T\mathbf{m}_2$$
$$= 1.5x_1 + 0.3x_2 - 1.17.$$

From Eq. (12.2-6), the equation of the boundary is

$$d_{12}(\mathbf{x}) = d_1(\mathbf{x}) - d_2(\mathbf{x})$$
$$= 2.8x_1 + 1.0x_2 - 8.9 = 0.$$

Figure 12.6 shows a plot of this boundary (note that the axes are not to the same scale). Substitution of any pattern vector from class ω_1 would yield $d_{12}(\mathbf{x}) > 0$. Conversely, any pattern from class ω_2 would yield $d_{12}(\mathbf{x}) < 0$. In other words, given an unknown pattern belonging to one of these two classes, the sign of $d_{12}(\mathbf{x})$ would be sufficient to determine the pattern's class membership. ■

In practice, the minimum distance classifier works well when the distance between means is large compared to the spread or randomness of each class with respect to its mean. In Section 12.2.2 we show that the minimum distance classifier yields optimum performance (in terms of minimizing the average loss of misclassification) when the distribution of each class about its mean is in the form of a spherical "hypercloud" in n-dimensional pattern space.

The simultaneous occurrence of large mean separations and relatively small class spread occur seldomly in practice unless the system designer controls the nature of the input. An excellent example is provided by systems designed to read stylized character fonts, such as the familiar American Banker's Association E-13B font character set. As Fig. 12.7 shows, this particular font set consists of 14 characters that were purposely designed on a 9×7 grid in order to facilitate their reading. The characters usually are printed in ink that contains finely ground magnetic material. Prior to being read, the ink is subjected to a magnetic field, which accentuates each character to simplify detection. In other words, the segmentation problem is solved by artificially highlighting the key characteristics of each character.

The characters typically are scanned in a horizontal direction with a single-slit reading head that is narrower but taller than the characters. As the head moves across a character, it produces a 1-D electrical signal (a signature) that is conditioned to be proportional to the rate of increase or decrease of the character area under the head. For example, consider the waveform associated with the number 0 in Fig. 12.7. As the reading head moves from left to right, the area seen by the head begins to increase, producing a positive derivative (a positive rate of change). As the head begins to leave the left leg of the 0, the area under the head begins to decrease, producing a negative derivative. When the head is in the middle zone of the character, the area remains nearly constant, producing a zero derivative. This pattern repeats itself as the head enters the right leg of the character. The design of the font ensures that the waveform of each character is distinct from that of all others. It also ensures that the peaks and zeros of each waveform occur approximately on the vertical lines of the background

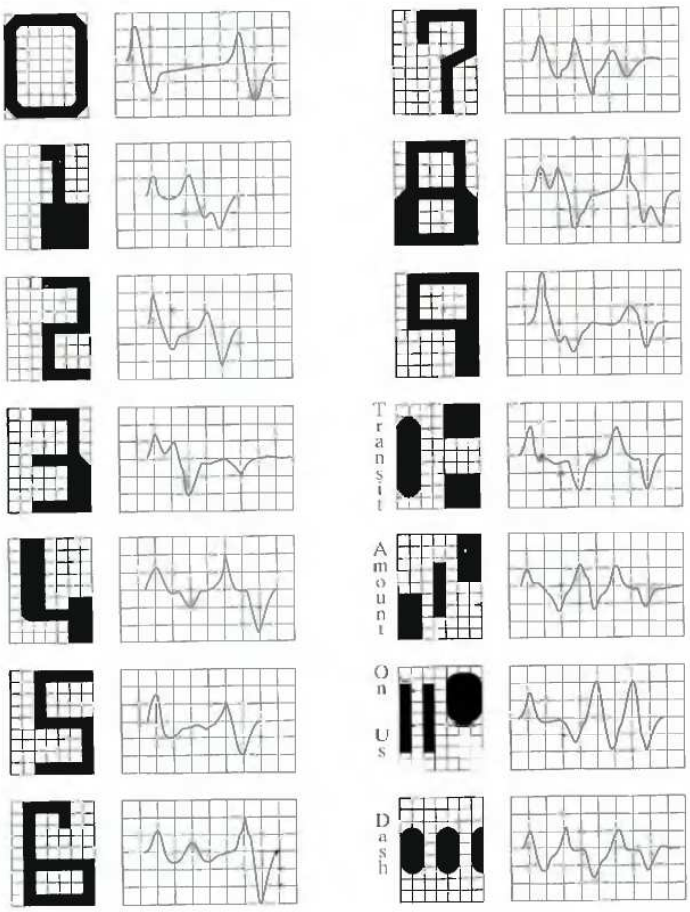


FIGURE 12.7 American Bankers Association E-13B font character set and corresponding waveforms.

grid on which these waveforms are displayed, as shown in Fig. 12.7. The E-13B font has the property that sampling the waveforms only at these points yields enough information for their proper classification. The use of magnetized ink aids in providing clean waveforms, thus minimizing scatter.

Designing a minimum distance classifier for this application is straightforward. We simply store the sample values of each waveform and let each set of samples be represented as a prototype vector $\mathbf{m}_j$, $j = 1, 2, \ldots, 14$. When an unknown character is to be classified, the approach is to scan it in the manner just described, express the grid samples of the waveform as a vector, $\mathbf{x}$, and identify its class by selecting the class of the prototype vector that yields the highest value in Eq. (12.2-5). High classification speeds can be achieved with analog circuits composed of resistor banks (see Problem 12.4).

Matching by correlation

We introduced the basic concept of image correlation in Section 4.6.4. Here, we consider it as the basis for finding matches of a subimage $w(x, y)$ of size $J \times K$ within an image $f(x, y)$ of size $M \times N$, where we assume that $J \leq M$ and $K \leq N$. Although the correlation approach can be expressed in vector form (see Problem 12.5), working directly with an image or subimage format is more intuitive (and traditional).

In its simplest form, the correlation between $f(x, y)$ and $w(x, y)$ is

$$c(x, y) = \sum_s \sum_t f(s, t)w(x + s, y + t) \tag{12.2-7}$$

for $x = 0, 1, 2, \ldots, M - 1$, $y = 0, 1, 2, \ldots, N - 1$, and the summation is taken over the image region where w and f overlap. Note by comparing this equation with Eq. (4.6-30) that it is implicitly assumed that the functions are real quantities and that we left out the MN constant. The reason is that we are going to use a normalized function in which these constants cancel out, and the definition given in Eq. (12.2-7) is used commonly in practice. We also used the symbols s and t in Eq. (12.2-7) to avoid confusion with m and n, which are used for other purposes in this chapter.

Figure 12.8 illustrates the procedure, where we assume that the origin of f is at its top left and the origin of w is at its center. For *one* value of (x, y), say, (x_0, y_0) inside f, application of Eq. (12.2-7) yields *one* value of c. As x and y are varied, w moves around the image area, giving the function $c(x, y)$. The maximum value(s) of c indicates the position(s) where w best matches f. Note that accuracy is lost for values of x and y near the edges of f, with the amount of error being in the correlation proportional to the size of w. This is the familiar border problem that we encountered numerous times in Chapter 3.

The correlation function given in Eq. (12.2-7) has the disadvantage of being sensitive to changes in the amplitude of f and w. For example, doubling all values of f doubles the value of $c(x, y)$. An approach frequently used to over-

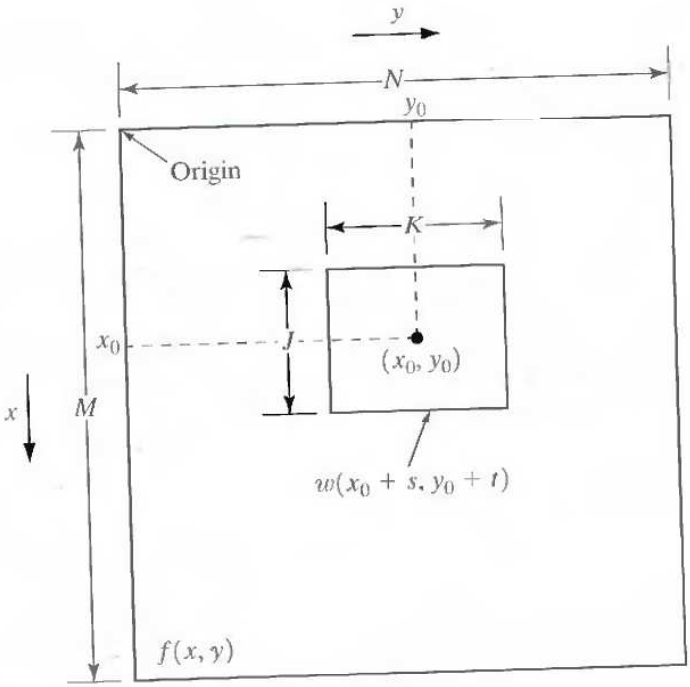


FIGURE 12.8 Arrangement for obtaining the correlation of f and w at point (x_0, y_0).

come this difficulty is to perform matching via the *correlation coefficient*, which is defined as

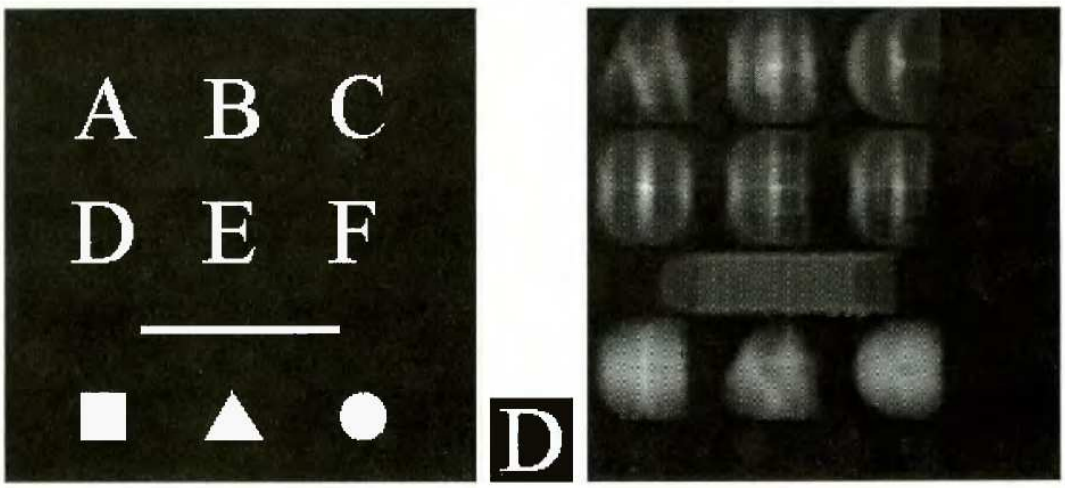
$$\gamma(x, y) = \frac{\sum_s \sum_t [f(s,t) - \bar{f}(s,t)][w(x+s, y+t) - \bar{w}]}{\left\{ \sum_s \sum_t [f(s,t) - \bar{f}(s,t)]^2 \sum_s \sum_t [w(x+s, y+t) - \bar{w}]^2 \right\}^{\frac{1}{2}}} \tag{12.2-8}$$

where $x = 0, 1, 2, \ldots, M - 1$, $y = 0, 1, 2, \ldots, N - 1$, $\bar{w}$ is the average value of the pixels in w (computed only once), $\bar{f}$ is the average value of f in the region coincident with the current location of w, and the summations are taken over the coordinates common to both f and w. The correlation coefficient $\gamma(x, y)$ is scaled in the range -1 to 1, independent of scale changes in the amplitude of f and w (see Problem 12.5).

EXAMPLE 12.2: Object matching via the correlation coefficient.

■ Figure 12.9 illustrates the concepts just discussed. Figure 12.9(a) is $f(x, y)$ and Fig. 12.9(b) is $w(x, y)$. The correlation coefficient $\gamma(x, y)$ is shown as an image in Fig. 12.9(c). The higher (brighter) value of $\gamma(x, y)$ is in the position where the best match between f and w was found. ■

Although the correlation function can be normalized for amplitude changes via the correlation coefficient, obtaining normalization for changes in size and rotation can be difficult. Normalizing for size involves spatial scaling, a process that in itself adds a significant amount of computation. Normalizing for rotation is even more difficult. If a clue regarding rotation can be extracted from $f(x, y)$, then we simply rotate $w(x, y)$ so that it aligns itself with the degree of rotation in $f(x, y)$. However, if the nature of rotation is unknown, looking for the best match requires exhaustive rotations of $w(x, y)$. This procedure is impractical and, as a consequence, correlation seldom is used in cases when arbitrary or unconstrained rotation is present.



a b c

FIGURE 12.9 (a) Image. (b) Subimage. (c) Correlation coefficient of (a) and (b). Note that the highest (brighter) point in (c) occurs when subimage (b) is coincident with the letter "D" in (a).

In Section 4.6.4 we mentioned that correlation also can be carried out in the frequency domain via the FFT. If f and w are the same size, this approach can be more efficient than direct implementation of correlation in the spatial domain. Equation (12.2-7) is used when w is much smaller than f. A trade-off estimate performed by Campbell [1969] indicates that, if the number of nonzero terms in w is less than 132 (a subimage of approximately 13×13 pixels), direct implementation of Eq. (12.2-7) is more efficient than the FFT approach. This number, of course, depends on the machine and algorithms used, but it does indicate approximate subimage size at which the frequency domain should be considered as an alternative. The correlation coefficient is more difficult to implement in the frequency domain. It generally is computed directly in the spatial domain.

12.2.2 Optimum Statistical Classifiers

In this section we develop a probabilistic approach to recognition. As is true in most fields that deal with measuring and interpreting physical events, probability considerations become important in pattern recognition because of the randomness under which pattern classes normally are generated. As shown in the following discussion, it is possible to derive a classification approach that is optimal in the sense that, on average, its use yields the lowest probability of committing classification errors (see Problem 12.10).

Foundation

The probability that a particular pattern $\mathbf{x}$ comes from class ω_i is denoted $p(\omega_i/\mathbf{x})$. If the pattern classifier decides that $\mathbf{x}$ came from ω_j when it actually came from ω_i, it incurs a loss, denoted L_{ij}. As pattern $\mathbf{x}$ may belong to any one of W classes under consideration, the average loss incurred in assigning $\mathbf{x}$ to class ω_j is

$$r_j(\mathbf{x}) = \sum_{k=1}^{W} L_{kj} p(\omega_k/\mathbf{x}). \tag{12.2-9}$$

This equation often is called the *conditional average risk* or *loss* in decision-theory terminology.

See inside front cover
Consult the book web site for a brief review of probability theory.

From basic probability theory, we know that $p(A/B) = [p(A)p(B/A)]/p(B)$. Using this expression, we write Eq. (12.2-9) in the form

$$r_j(\mathbf{x}) = \frac{1}{p(\mathbf{x})} \sum_{k=1}^{W} L_{kj} p(\mathbf{x}/\omega_k) P(\omega_k) \tag{12.2-10}$$

where $p(\mathbf{x}/\omega_k)$ is the probability density function of the patterns from class ω_k and $P(\omega_k)$ is the probability of occurrence of class ω_k. Because $1/p(\mathbf{x})$ is positive and common to all the $r_j(\mathbf{x})$, $j = 1, 2, \ldots, W$, it can be dropped from Eq. (12.2-10) without affecting the relative order of these functions from the smallest to the largest value. The expression for the average loss then reduces to

$$r_j(\mathbf{x}) = \sum_{k=1}^{W} L_{kj} p(\mathbf{x}/\omega_k) P(\omega_k). \tag{12.2-11}$$

The classifier has W possible classes to choose from for any given unknown pattern. If it computes $r_1(\mathbf{x}), r_2(\mathbf{x}), \ldots, r_W(\mathbf{x})$ for each pattern $\mathbf{x}$ and assigns the pattern to the class with the smallest loss, the total average loss with respect to all decisions will be minimum. The classifier that minimizes the total average loss is called the *Bayes classifier*. Thus the Bayes classifier assigns an unknown pattern $\mathbf{x}$ to class ω_i if $r_i(\mathbf{x}) < r_j(\mathbf{x})$ for $j = 1, 2, \ldots, W; j \neq i$. In other words, $\mathbf{x}$ is assigned to class ω_i if

$$\sum_{k=1}^{W} L_{ki}\, p(\mathbf{x}/\omega_k)P(\omega_k) < \sum_{q=1}^{W} L_{qj}\, p(\mathbf{x}/\omega_q)P(\omega_q) \tag{12.2-12}$$

for all $j; j \neq i$. The "loss" for a correct decision generally is assigned a value of zero, and the loss for any incorrect decision usually is assigned the same nonzero value (say, 1). Under these conditions, the loss function becomes

$$L_{ij} = 1 - \delta_{ij} \tag{12.2-13}$$

where $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ if $i \neq j$. Equation (12.2-13) indicates a loss of unity for incorrect decisions and a loss of zero for correct decisions. Substituting Eq. (12.2-13) into Eq. (12.2-11) yields

$$\begin{aligned} r_j(\mathbf{x}) &= \sum_{k=1}^{W} (1 - \delta_{kj})p(\mathbf{x}/\omega_k)P(\omega_k) \\ &= p(\mathbf{x}) - p(\mathbf{x}/\omega_j)P(\omega_j). \end{aligned} \tag{12.2-14}$$

The Bayes classifier then assigns a pattern $\mathbf{x}$ to class ω_i if, for all $j \neq i$,

$$p(\mathbf{x}) - p(\mathbf{x}/\omega_i)P(\omega_i) < p(\mathbf{x}) - p(\mathbf{x}/\omega_j)P(\omega_j) \tag{12.2-15}$$

or, equivalently, if

$$p(\mathbf{x}/\omega_i)P(\omega_i) > p(\mathbf{x}/\omega_j)P(\omega_j) \qquad j = 1, 2, \ldots, W; j \neq i. \tag{12.2-16}$$

With reference to the discussion leading to Eq. (12.2-1), we see that the Bayes classifier for a 0-1 loss function is nothing more than computation of decision functions of the form

$$d_j(\mathbf{x}) = p(\mathbf{x}/\omega_j)P(\omega_j) \qquad j = 1, 2, \ldots, W \tag{12.2-17}$$

where a pattern vector $\mathbf{x}$ is assigned to the class whose decision function yields the largest numerical value.

The decision functions given in Eq. (12.2-7) are optimal in the sense that they minimize the average loss in misclassification. For this optimality to hold, however, the probability density functions of the patterns in each class, as well as the probability of occurrence of each class, must be known. The latter requirement usually is not a problem. For instance, if all classes are equally likely to occur, then $P(\omega_j) = 1/M$. Even if this condition is not true, these probabilities generally can be inferred from knowledge of the problem. Estimation of the probability density functions $p(\mathbf{x}/\omega_j)$ is another matter. If the pattern vectors, $\mathbf{x}$, are n dimensional, then $p(\mathbf{x}/\omega_j)$ is a function of n variables, which, if its form is not known, requires methods from multivariate probability theory for its estimation. These methods are difficult to apply in practice,

especially if the number of representative patterns from each class is not large or if the underlying form of the probability density functions is not well behaved. For these reasons, use of the Bayes classifier generally is based on the assumption of an analytic expression for the various density functions and then an estimation of the necessary parameters from sample patterns from each class. By far the most prevalent form assumed for $p(\mathbf{x}/\omega_j)$ is the Gaussian probability density function. The closer this assumption is to reality, the closer the Bayes classifier approaches the minimum average loss in classification.

Bayes classifier for Gaussian pattern classes

To begin, let us consider a 1-D problem ($n = 1$) involving two pattern classes ($W = 2$) governed by Gaussian densities, with means m_1 and m_2 and standard deviations σ_1 and σ_2, respectively. From Eq. (12.2-17) the Bayes decision functions have the form

$$\begin{aligned} d_j(x) &= p(x/\omega_j)P(\omega_j) \\ &= \frac{1}{\sqrt{2\pi}\sigma_j} e^{-\frac{(x-m_j)^2}{2\sigma_j^2}} P(\omega_j) \qquad j = 1, 2 \end{aligned} \tag{12.2-18}$$

where the patterns are now scalars, denoted by x. Figure 12.10 shows a plot of the probability density functions for the two classes. The boundary between the two classes is a single point, denoted x_0 such that $d_1(x_0) = d_2(x_0)$. If the two classes are equally likely to occur, then $P(\omega_1) = P(\omega_2) = 1/2$, and the decision boundary is the value of x_0 for which $p(x_0/\omega_1) = p(x_0/\omega_2)$. This point is the intersection of the two probability density functions, as shown in Fig. 12.10. Any pattern (point) to the right of x_0 is classified as belonging to class ω_1. Similarly, any pattern to the left of x_0 is classified as belonging to class ω_2. When the classes are not equally likely to occur, x_0 moves to the left if class ω_1 is more likely to occur or, conversely, to the right if class ω_2 is more likely to occur. This result is to be expected, because the classifier is trying to minimize the loss of misclassification. For instance, in the extreme case, if class ω_2 never occurs, the classifier would never make a mistake by always assigning all patterns to class ω_1 (that is, x_0 would move to negative infinity).

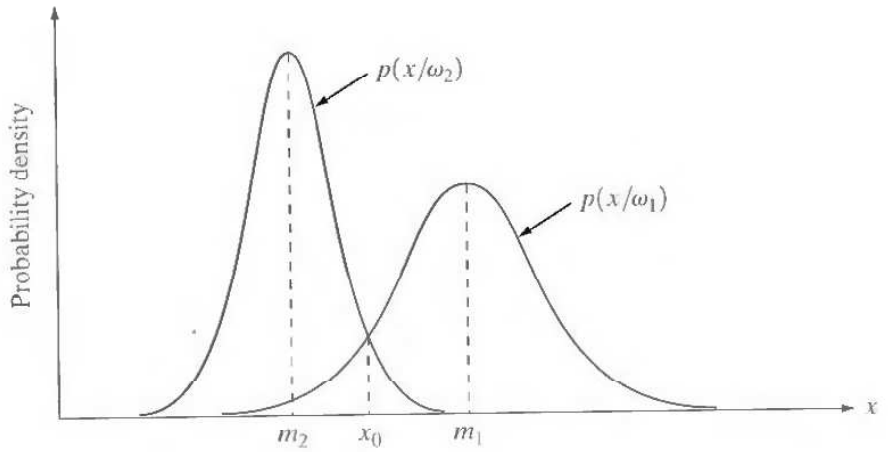


FIGURE 12.10 Probability density functions for two 1-D pattern classes. The point x_0 shown is the decision boundary if the two classes are equally likely to occur.

In the n-dimensional case, the Gaussian density of the vectors in the jth pattern class has the form

$$p(\mathbf{x}/\omega_j) = \frac{1}{(2\pi)^{n/2}|\mathbf{C}_j|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\mathbf{m}_j)^T\mathbf{C}_j^{-1}(\mathbf{x}-\mathbf{m}_j)} \tag{12.2-19}$$

where each density is specified completely by its mean vector $\mathbf{m}_j$ and covariance matrix $\mathbf{C}_j$, which are defined as

$$\mathbf{m}_j = E_j\{\mathbf{x}\} \tag{12.2-20}$$

and

$$\mathbf{C}_j = E_j\{(\mathbf{x} - \mathbf{m}_j)(\mathbf{x} - \mathbf{m}_j)^T\} \tag{12.2-21}$$

where $E_j\{\cdot\}$ denotes the expected value of the argument over the patterns of class ω_j. In Eq. (12.2-19), n is the dimensionality of the pattern vectors, and $|\mathbf{C}_j|$ is the determinant of the matrix $\mathbf{C}_j$. Approximating the expected value E_j by the average value of the quantities in question yields an estimate of the mean vector and covariance matrix:

$$\mathbf{m}_j = \frac{1}{N_j}\sum_{\mathbf{x}\in\omega_j}\mathbf{x} \tag{12.2-22}$$

and

$$\mathbf{C}_j = \frac{1}{N_j}\sum_{\mathbf{x}\in\omega_j}\mathbf{x}\mathbf{x}^T - \mathbf{m}_j\mathbf{m}_j^T \tag{12.2-23}$$

where N_j is the number of pattern vectors from class ω_j, and the summation is taken over these vectors. Later in this section we give an example of how to use these two expressions.

The covariance matrix is symmetric and positive semidefinite. As explained in Section 11.4, the diagonal element c_{kk} is the variance of the kth element of the pattern vectors. The off-diagonal element c_{jk} is the covariance of x_j and x_k. The multivariate Gaussian density function reduces to the product of the univariate Gaussian density of each element of $\mathbf{x}$ when the off-diagonal elements of the covariance matrix are zero. This happens when the vector elements x_j and x_k are uncorrelated.

See inside front cover

Consult the book web site for a brief review of vectors and matrices.

According to Eq. (12.2-17), the Bayes decision function for class ω_j is $d_j(\mathbf{x}) = p(\mathbf{x}/\omega_j)P(\omega_j)$. However, because of the exponential form of the Gaussian density, working with the natural logarithm of this decision function is more convenient. In other words, we can use the form

$$\begin{aligned} d_j(\mathbf{x}) &= \ln[p(\mathbf{x}/\omega_j)P(\omega_j)] \\ &= \ln p(\mathbf{x}/\omega_j) + \ln P(\omega_j). \end{aligned} \tag{12.2-24}$$

This expression is equivalent to Eq. (12.2-17) in terms of classification performance because the logarithm is a monotonically increasing function. In other words, the numerical *order* of the decision functions in Eqs. (12.2-17) and (12.2-24) is the same. Substituting Eq. (12.2-19) into Eq. (12.2-24) yields

$$d_j(\mathbf{x}) = \ln P(\omega_j) - \frac{n}{2}\ln 2\pi - \frac{1}{2}\ln|\mathbf{C}_j| - \frac{1}{2}[(\mathbf{x} - \mathbf{m}_j)^T\mathbf{C}_j^{-1}(\mathbf{x} - \mathbf{m}_j)]. \tag{12.2-25}$$

Figure 12.11 shows a section of this surface, where we note that the classes were separated effectively. ■

One of the most successful applications of the Bayes classifier approach is in the classification of remotely sensed imagery generated by multispectral scanners aboard aircraft, satellites, or space stations. The voluminous image data generated by these platforms make automatic image classification and analysis a task of considerable interest in remote sensing. The applications of remote sensing are varied and include land use, crop inventory, crop disease detection, forestry, air and water quality monitoring, geological studies, weather prediction, and a score of other applications having environmental significance. The following example shows a typical application.

EXAMPLE 12.4: Classification of multispectral data using the Bayes classifier.

■ As discussed in Sections 1.3.4 and 11.4, a multispectral scanner responds to electromagnetic energy in selected wavelength bands; for example, 0.40-0.44, 0.58-0.62, 0.66-0.72, and 0.80-1.00 microns. These ranges are in the violet, green, red, and infrared bands, respectively. A region on the ground scanned in this manner produces four digital images, one image for each band. If the images are registered, a condition which is generally true in practice, they can be visualized as being stacked one behind the other, as Fig. 12.12 shows. Thus, just as we did in Section 11.4, every point on the ground can be represented by a 4-element pattern vector of the form $\mathbf{x} = (x_1, x_2, x_3, x_4)^T$, where x_1 is a shade of violet, x_2 is a shade of green, and so on. If the images are of size 512×512 pixels, each stack of four multispectral images can be represented by 262,144 4-dimensional pattern vectors.

As noted previously, the Bayes classifier for Gaussian patterns requires estimation of the mean vector and covariance matrix for each class. In remote sensing applications these estimates are obtained by collecting multispectral data for each region of interest and then using these samples, as described in the preceding example. Figure 12.13(a) shows a typical image sensed remotely from an aircraft (this is a monochrome version of a multispectral original). In this

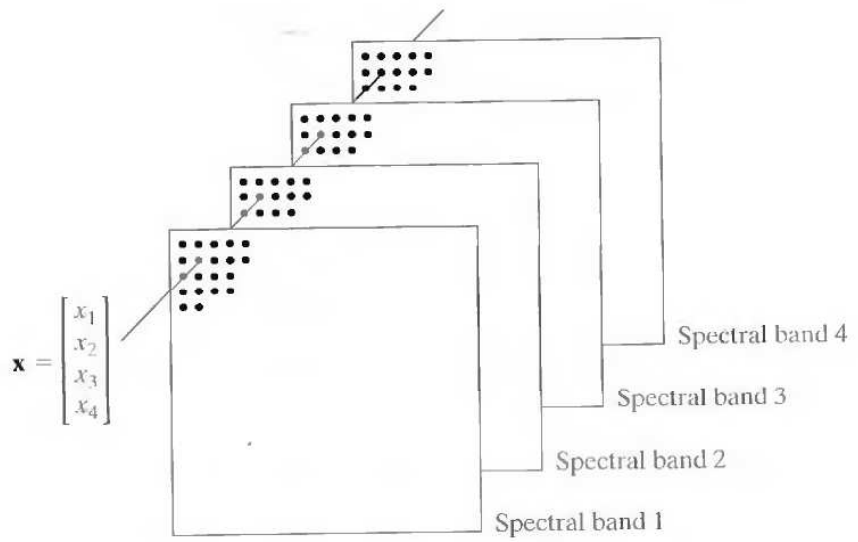
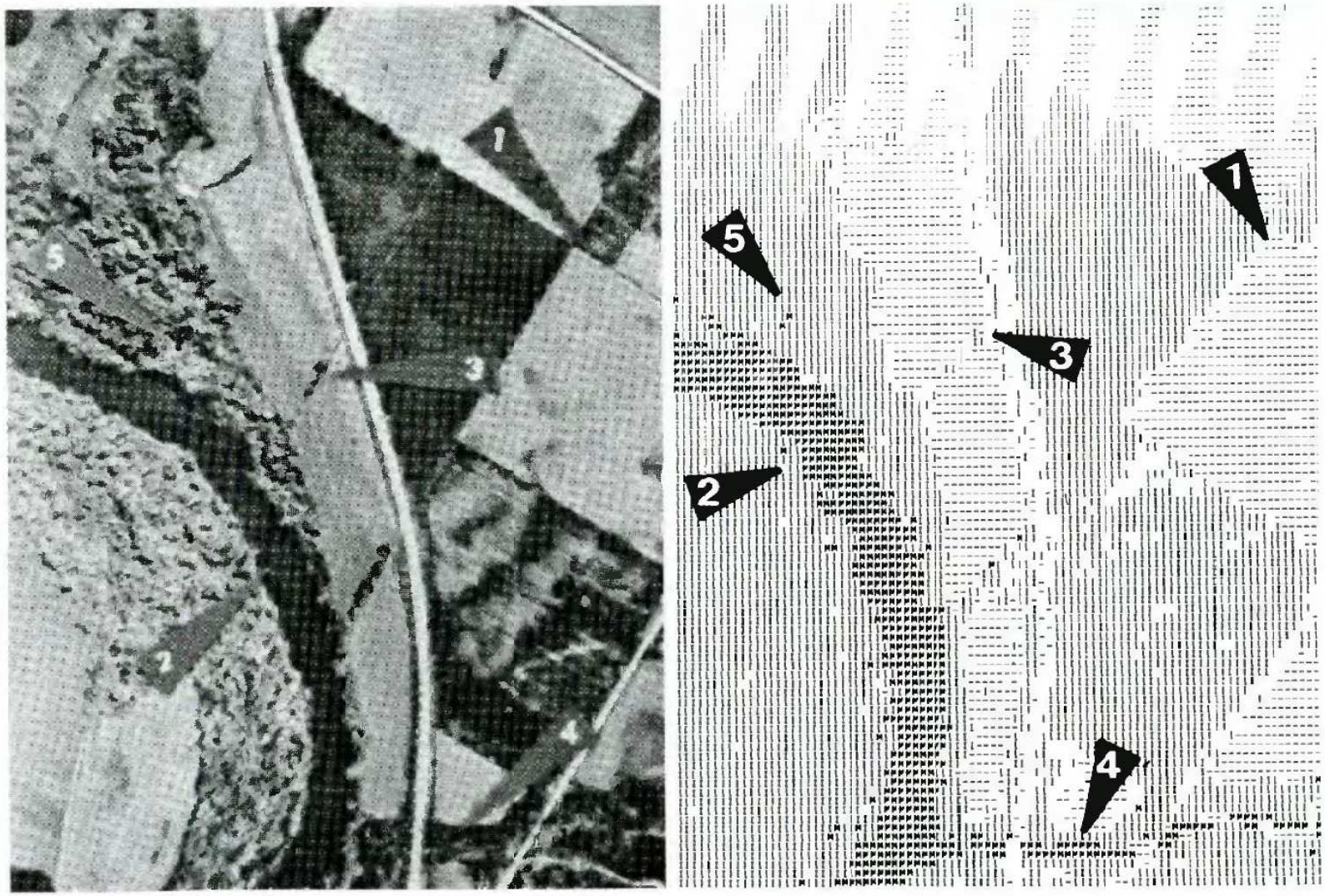
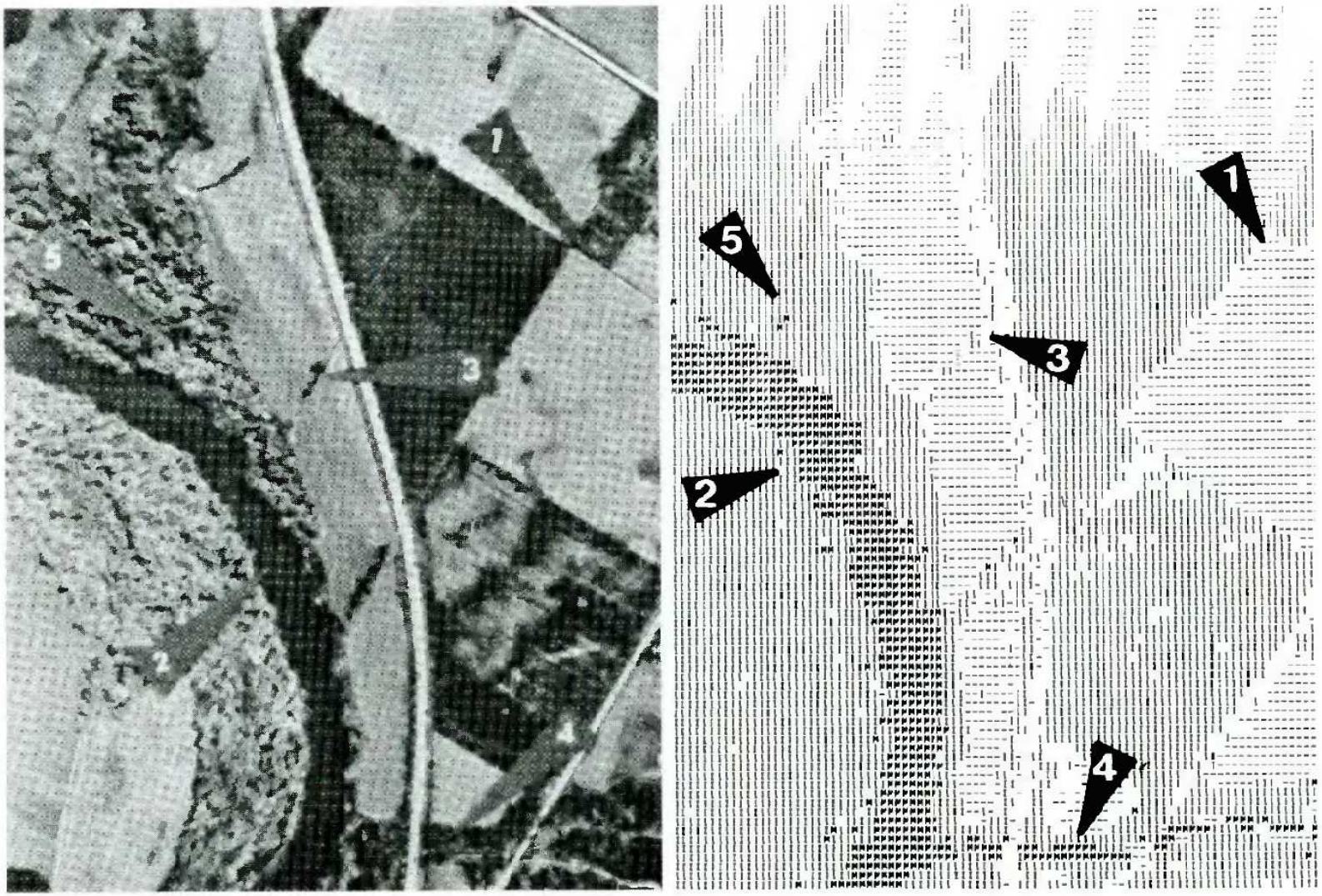


FIGURE 12.12 Formation of a pattern vector from registered pixels of four digital images generated by a multispectral scanner.



a b

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.)



a b

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.)

particular case, the problem was to classify areas such as vegetation, water, and bare soil. Figure 12.13(b) shows the results of machine classification, using a Gaussian Bayes classifier. The arrows indicate some features of interest. Arrow 1 points to a corner of a field of green vegetation, and arrow 2 points to a river. Arrow 3 identifies a small hedgerow between two areas of bare soil. Arrow 4 indicates a tributary correctly identified by the system. Arrow 5 points to a small pond that is almost indistinguishable in Fig. 12.13(a). Comparing the original image with the computer output reveals recognition results that are very close to those that a human would generate by visual analysis. ■

Before leaving this section, it is of interest to note that pixel-by-pixel classification of an image as described in the previous example actually segments the image into various classes. This approach is like segmentation by thresholding with several variables, as discussed briefly in Section 10.3.7.

12.2.3 Neural Networks

The approaches discussed in the preceding two sections are based on the use of sample patterns to estimate statistical parameters of each pattern class. The minimum distance classifier is specified completely by the mean vector of each class. Similarly, the Bayes classifier for Gaussian populations is specified completely by the mean vector and covariance matrix of each class. The patterns (of *known* class membership) used to estimate these parameters usually are called *training patterns*, and a set of such patterns from each class is called a *training set*. The process by which a training set is used to obtain decision functions is called *learning* or *training*.

In the two approaches just discussed, training is a simple matter. The training patterns of each class are used to compute the parameters of the decision function corresponding to that class. After the parameters in question have been estimated, the structure of the classifier is fixed, and its eventual performance will depend on how well the actual pattern populations satisfy the underlying statistical assumptions made in the derivation of the classification method being used.

The statistical properties of the pattern classes in a problem often are unknown or cannot be estimated (recall our brief discussion in the preceding section regarding the difficulty of working with multivariate statistics). In practice, such decision-theoretic problems are best handled by methods that yield the required decision functions directly via training. Then, making assumptions regarding the underlying probability density functions or other probabilistic information about the pattern classes under consideration is unnecessary. In this section we discuss various approaches that meet this criterion.

Background

The essence of the material that follows is the use of a multitude of elemental nonlinear computing elements (called *neurons*) organized as networks reminiscent of the way in which neurons are believed to be interconnected in the brain. The resulting models are referred to by various names, including *neural*

networks, neurocomputers, parallel distributed processing (PDP) *models, neuromorphic systems, layered self-adaptive networks*, and *connectionist models.* Here, we use the name *neural networks*, or *neural nets* for short. We use these networks as vehicles for adaptively developing the coefficients of decision functions via successive presentations of training sets of patterns.

Interest in neural networks dates back to the early 1940s, as exemplified by the work of McCulloch and Pitts [1943]. They proposed neuron models in the form of binary threshold devices and stochastic algorithms involving sudden 0-1 and 1-0 changes of states in neurons as the bases for modeling neural systems. Subsequent work by Hebb [1949] was based on mathematical models that attempted to capture the concept of learning by reinforcement or association.

During the mid-1950s and early 1960s, a class of so-called *learning machines* originated by Rosenblatt [1959, 1962] caused significant excitement among researchers and practitioners of pattern recognition theory. The reason for the great interest in these machines, called *perceptrons*, was the development of mathematical proofs showing that perceptrons, when trained with linearly separable training sets (i.e., training sets separable by a hyperplane), would converge to a solution in a finite number of iterative steps. The solution took the form of coefficients of hyperplanes capable of correctly separating the classes represented by patterns of the training set.

Unfortunately, the expectations following discovery of what appeared to be a well-founded theoretic model of learning soon met with disappointment. The basic perceptron and some of its generalizations at the time were simply inadequate for most pattern recognition tasks of practical significance. Subsequent attempts to extend the power of perceptron-like machines by considering multiple layers of these devices, although conceptually appealing, lacked effective training algorithms such as those that had created interest in the perceptron itself. The state of the field of learning machines in the mid-1960s was summarized by Nilsson [1965]. A few years later, Minsky and Papert [1969] presented a discouraging analysis of the limitation of perceptron-like machines. This view was held as late as the mid-1980s, as evidenced by comments by Simon [1986]. In this work, originally published in French in 1984, Simon dismisses the perceptron under the heading "Birth and Death of a Myth."

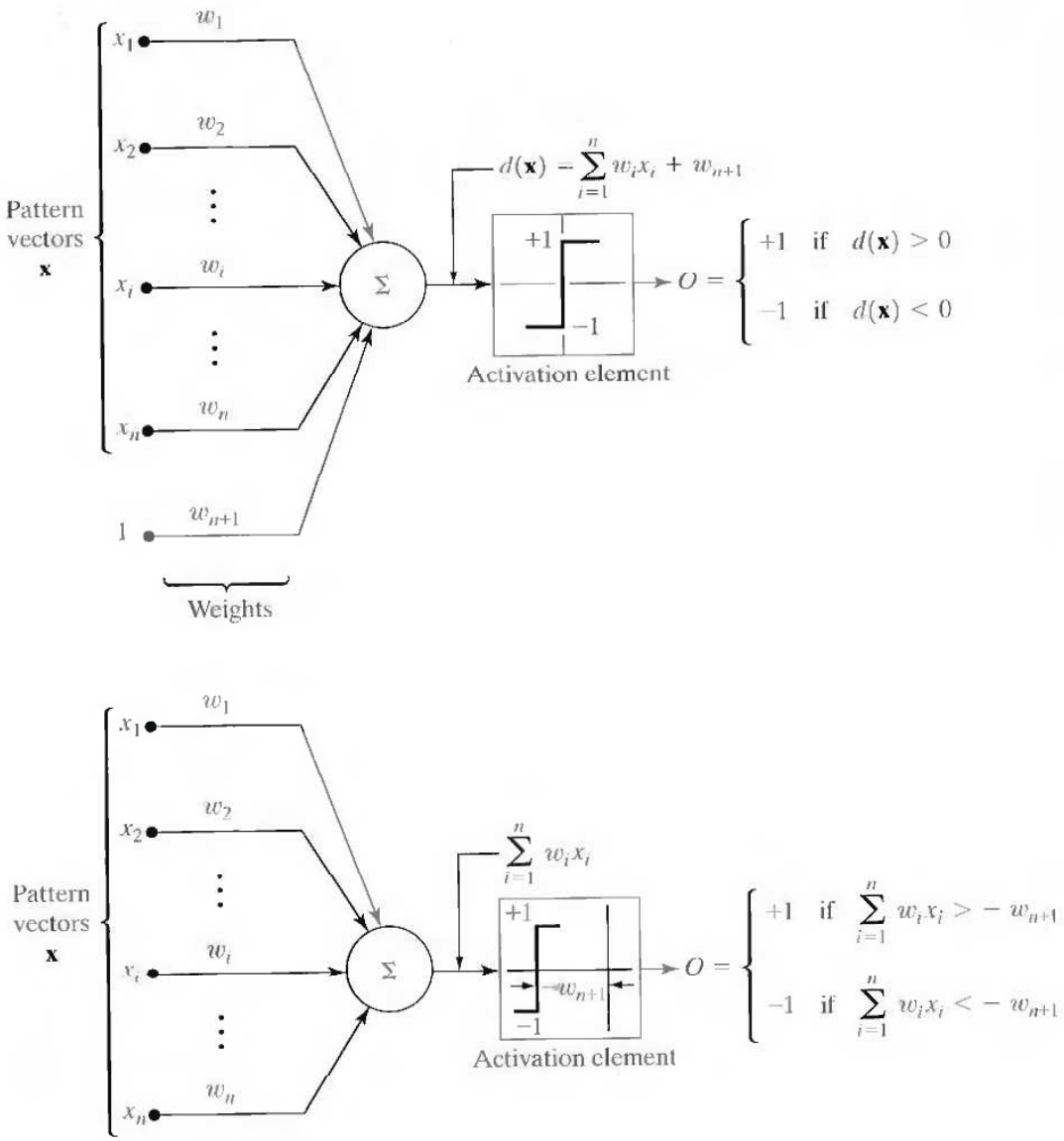
More recent results by Rumelhart, Hinton, and Williams [1986] dealing with the development of new training algorithms for multilayer perceptrons have changed matters considerably. Their basic method, often called the *generalized delta rule for learning by backpropagation*, provides an effective training method for multilayer machines. Although this training algorithm cannot be shown to converge to a solution in the sense of the analogous proof for the single-layer perceptron, the generalized delta rule has been used successfully in numerous problems of practical interest. This success has established multilayer perceptron-like machines as one of the principal models of neural networks currently in use.

Perceptron for two pattern classes

In its most basic form, the perceptron learns a linear decision function that dichotomizes two linearly separable training sets. Figure 12.14(a) shows schematically the perceptron model for two pattern classes. The response of this basic

a
b

FIGURE 12.14 Two equivalent representations of the perceptron model for two pattern classes.



device is based on a weighted sum of its inputs; that is,

$$d(\mathbf{x}) = \sum_{i=1}^{n} w_i x_i + w_{n+1}, \tag{12.2-29}$$

which is a linear decision function with respect to the components of the pattern vectors. The coefficients $w_i, i = 1, 2, \ldots, n, n + 1$, called *weights*, modify the inputs before they are summed and fed into the threshold element. In this sense, weights are analogous to synapses in the human neural system. The function that maps the output of the summing junction into the final output of the device sometimes is called the *activation function*.

When $d(\mathbf{x}) > 0$ the threshold element causes the output of the perceptron to be +1, indicating that the pattern $\mathbf{x}$ was recognized as belonging to class ω_1.

The reverse is true when $d(\mathbf{x}) < 0$. This mode of operation agrees with the comments made earlier in connection with Eq. (12.2-2) regarding the use of a single decision function for two pattern classes. When $d(\mathbf{x}) = 0$, $\mathbf{x}$ lies on the decision surface separating the two pattern classes, giving an indeterminate condition. The decision boundary implemented by the perceptron is obtained by setting Eq. (12.2-29) equal to zero:

$$d(\mathbf{x}) = \sum_{i=1}^{n} w_i x_i + w_{n+1} = 0 \tag{12.2-30}$$

or

$$w_1 x_1 + w_2 x_2 + \cdots + w_n x_n + w_{n+1} = 0, \tag{12.2-31}$$

which is the equation of a hyperplane in n-dimensional pattern space. Geometrically, the first n coefficients establish the orientation of the hyperplane, whereas the last coefficient, w_{n+1}, is proportional to the perpendicular distance from the origin to the hyperplane. Thus if $w_{n+1} = 0$, the hyperplane goes through the origin of the pattern space. Similarly, if $w_j = 0$, the hyperplane is parallel to the x_j-axis.

The output of the threshold element in Fig. 12.14(a) depends on the sign of $d(\mathbf{x})$. Instead of testing the entire function to determine whether it is positive or negative, we could test the summation part of Eq. (12.2-29) against the term w_{n+1}, in which case the output of the system would be

$$O = \begin{cases} +1 & \text{if } \sum_{i=1}^{n} w_i x_i > -w_{n+1} \\ -1 & \text{if } \sum_{i=1}^{n} w_i x_i < -w_{n+1}. \end{cases} \tag{12.2-32}$$

This implementation is equivalent to Fig. 12.14(a) and is shown in Fig. 12.14(b), the only differences being that the threshold function is displaced by an amount $-w_{n+1}$ and that the constant unit input is no longer present. We return to the equivalence of these two formulations later in this section when we discuss implementation of multilayer neural networks.

Another formulation used frequently is to augment the pattern vectors by appending an additional $(n + 1)$st element, which is always equal to 1, regardless of class membership. That is, an augmented pattern vector $\mathbf{y}$ is created from a pattern vector $\mathbf{x}$ by letting $y_i = x_i, i = 1, 2, \ldots, n$, and appending the additional element $y_{n+1} = 1$. Equation (12.2-29) then becomes

$$\begin{aligned} d(\mathbf{y}) &= \sum_{i=1}^{n+1} w_i y_i \\ &= \mathbf{w}^T \mathbf{y} \end{aligned} \tag{12.2-33}$$

where $\mathbf{y} = (y_1, y_2, \ldots, y_n, 1)^T$ is now an *augmented pattern vector*, and $\mathbf{w} = (w_1, w_2, \ldots, w_n, w_{n+1})^T$ is called the *weight vector*. This expression is usually more convenient in terms of notation. Regardless of the formulation used, however, the key problem is to find $\mathbf{w}$ by using a given training set of pattern vectors from each of two classes.

Training algorithms

The algorithms developed in the following discussion are representative of the numerous approaches proposed over the years for training perceptrons.

Linearly separable classes A simple, iterative algorithm for obtaining a solution weight vector for two linearly separable training sets follows. For two training sets of augmented pattern vectors belonging to pattern classes ω_1 and ω_2, respectively, let $\mathbf{w}(1)$ represent the initial weight vector, which may be chosen arbitrarily. Then, at the kth iterative step, if $\mathbf{y}(k) \in \omega_1$ and $\mathbf{w}^T(k)\mathbf{y}(k) \leq 0$, replace $\mathbf{w}(k)$ by

$$\mathbf{w}(k+1) = \mathbf{w}(k) + c\mathbf{y}(k) \tag{12.2-34}$$

where c is a positive correction increment. Conversely, if $\mathbf{y}(k) \leq \omega_2$ and $\mathbf{w}^T(k)\mathbf{y}(k) \geq 0$, replace $\mathbf{w}(k)$ with

$$\mathbf{w}(k+1) = \mathbf{w}(k) - c\mathbf{y}(k). \tag{12.2-35}$$

Otherwise, leave $\mathbf{w}(k)$ unchanged:

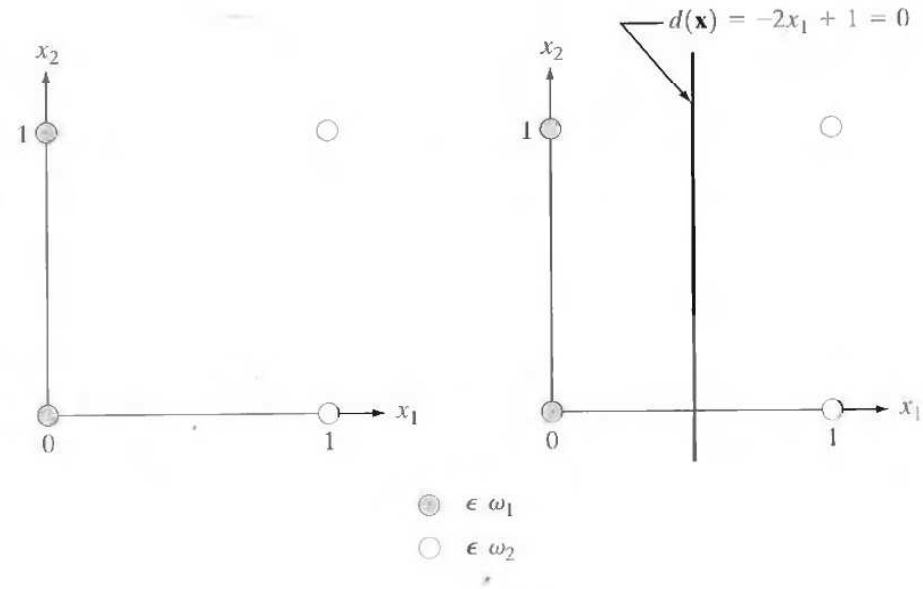
$$\mathbf{w}(k+1) = \mathbf{w}(k). \tag{12.2-36}$$

This algorithm makes a change in $\mathbf{w}$ only if the pattern being considered at the kth step in the training sequence is misclassified. The correction increment c is assumed to be positive and, for now, to be constant. This algorithm sometimes is referred to as the *fixed increment correction rule*.

Convergence of the algorithm occurs when the entire training set for both classes is cycled through the machine without any errors. The fixed increment correction rule converges in a finite number of steps if the two training sets of patterns are linearly separable. A proof of this result, sometimes called the *perceptron training theorem*, can be found in the books by Duda, Hart, and Stork [2001]; Tou and Gonzalez [1974]; and Nilsson [1965].

EXAMPLE 12.5: Illustration of the perceptron algorithm.

■ Consider the two training sets shown in Fig. 12.15(a), each consisting of two patterns. The training algorithm will be successful because the two training sets are linearly separable. Before the algorithm is applied the patterns are augmented,



a b

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

yielding the training set $\{(0,0,1)^T, (0,1,1)^T\}$ for class ω_1 and $\{(1,0,1)^T, (1,1,1)^T\}$ for class ω_2. Letting $c = 1, \mathbf{w}(1) = \mathbf{0}$, and presenting the patterns in order results in the following sequence of steps:

$$\mathbf{w}^T(1)\mathbf{y}(1) = [0,0,0]\begin{bmatrix}0\\0\\1\end{bmatrix} = 0 \qquad \mathbf{w}(2) = \mathbf{w}(1) + \mathbf{y}(1) = \begin{bmatrix}0\\0\\1\end{bmatrix}$$

$$\mathbf{w}^T(2)\mathbf{y}(2) = [0,0,1]\begin{bmatrix}0\\1\\1\end{bmatrix} = 1 \qquad \mathbf{w}(3) = \mathbf{w}(2) = \begin{bmatrix}0\\0\\1\end{bmatrix}$$

$$\mathbf{w}^T(3)\mathbf{y}(3) = [0,0,1]\begin{bmatrix}1\\0\\1\end{bmatrix} = 1 \qquad \mathbf{w}(4) = \mathbf{w}(3) - \mathbf{y}(3) = \begin{bmatrix}-1\\0\\0\end{bmatrix}$$

$$\mathbf{w}^T(4)\mathbf{y}(4) = [-1,0,0]\begin{bmatrix}1\\1\\1\end{bmatrix} = -1 \qquad \mathbf{w}(5) = \mathbf{w}(4) = \begin{bmatrix}-1\\0\\0\end{bmatrix}$$

where corrections in the weight vector were made in the first and third steps because of misclassifications, as indicated in Eqs. (12.2-34) and (12.2-35). Because a solution has been obtained only when the algorithm yields a complete error-free iteration through all training patterns, the training set must be presented again. The machine learning process is continued by letting $\mathbf{y}(5) = \mathbf{y}(1)$, $\mathbf{y}(6) = \mathbf{y}(2)$, $\mathbf{y}(7) = \mathbf{y}(3)$, and $\mathbf{y}(8) = \mathbf{y}(4)$, and proceeding in the same manner. Convergence is achieved at $k = 14$, yielding the solution weight vector $\mathbf{w}(14) = (-2, 0, 1)^T$. The corresponding decision function is $d(\mathbf{y}) = -2y_1 + 1$. Going back to the original pattern space by letting $x_i = y_i$ yields $d(\mathbf{x}) = -2x_1 + 1$, which, when set equal to zero, becomes the equation of the decision boundary shown in Fig. 12.15(b). ■

Nonseparable classes In practice, linearly separable pattern classes are the (rare) exception, rather than the rule. Consequently, a significant amount of research effort during the 1960s and 1970s went into development of techniques designed to handle nonseparable pattern classes. With recent advances in the training of neural networks, many of the methods dealing with nonseparable behavior have become merely items of historical interest. One of the early methods, however, is directly relevant to this discussion: the original delta rule. Known as the *Widrow-Hoff*, or *least-mean-square* (LMS) *delta rule* for training perceptrons, the method minimizes the error between the actual and desired response at any training step.

Consider the criterion function

$$J(\mathbf{w}) = \frac{1}{2}(r - \mathbf{w}^T\mathbf{y})^2 \tag{12.2-37}$$

where r is the desired response (that is, $r = +1$ if the augmented training pattern vector $\mathbf{y}$ belongs to class ω_1, and $r = -1$ if $\mathbf{y}$ belongs to class ω_2). The task

is to adjust $\mathbf{w}$ incrementally in the direction of the negative gradient of $J(\mathbf{w})$ in order to seek the minimum of this function, which occurs when $r = \mathbf{w}^T\mathbf{y}$; that is, the minimum corresponds to correct classification. If $\mathbf{w}(k)$ represents the weight vector at the kth iterative step, a general gradient descent algorithm may be written as

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \alpha\left[\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}}\right]_{\mathbf{w}=\mathbf{w}(k)} \tag{12.2-38}$$

where $\mathbf{w}(k+1)$ is the new value of $\mathbf{w}$, and $\alpha > 0$ gives the magnitude of the correction. From Eq. (12.2-37),

$$\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} = -(r - \mathbf{w}^T\mathbf{y})\mathbf{y}. \tag{12.2-39}$$

Substituting this result into Eq. (12.2-38) yields

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \alpha[r(k) - \mathbf{w}^T(k)\mathbf{y}(k)]\mathbf{y}(k) \tag{12.2-40}$$

with the starting weight vector, $\mathbf{w}(1)$, being arbitrary.

By defining the change (delta) in weight vector as

$$\Delta\mathbf{w} = \mathbf{w}(k+1) - \mathbf{w}(k) \tag{12.2-41}$$

we can write Eq. (12.2-40) in the form of a *delta correction algorithm:*

$$\Delta\mathbf{w} = \alpha e(k)\mathbf{y}(k) \tag{12.2-42}$$

where

$$e(k) = r(k) - \mathbf{w}^T(k)\mathbf{y}(k) \tag{12.2-43}$$

is the error committed with weight vector $\mathbf{w}(k)$ when pattern $\mathbf{y}(k)$ is presented.

Equation (12.2-43) gives the error with weight vector $\mathbf{w}(k)$. If we change it to $\mathbf{w}(k+1)$, but leave the pattern the same, the error becomes

$$e(k) = r(k) - \mathbf{w}^T(k+1)\mathbf{y}(k). \tag{12.2-44}$$

The change in error then is

$$\begin{aligned}\Delta e(k) &= [r(k) - \mathbf{w}^T(k+1)\mathbf{y}(k)] - [r(k) - \mathbf{w}^T(k)\mathbf{y}(k)] \\ &= -[\mathbf{w}^T(k+1) - \mathbf{w}^T(k)]\mathbf{y}(k) \\ &= -\Delta\mathbf{w}^T\mathbf{y}(k).\end{aligned} \tag{12.2-45}$$

But $\Delta\mathbf{w} = \alpha e(k)\mathbf{y}(k)$, so

$$\begin{aligned}\Delta e &= -\alpha e(k)\mathbf{y}^T(k)\mathbf{y}(k) \\ &= -\alpha e(k)\|\mathbf{y}(k)\|^2.\end{aligned} \tag{12.2-46}$$

Hence changing the weights reduces the error by a factor $\alpha\|\mathbf{y}(k)\|^2$. The next input pattern starts the new adaptation cycle, reducing the next error by a factor $\alpha\|\mathbf{y}(k+1)\|^2$, and so on.

The choice of α controls stability and speed of convergence (Widrow and Stearns [1985]). Stability requires that $0 < \alpha < 2$. A practical range for α is $0.1 < \alpha < 1.0$. Although the proof is not shown here, the algorithm of

Eq. (12.2-40) or Eqs. (12.2-42) and (12.2-43) converges to a solution that minimizes the mean square error over the patterns of the training set. When the pattern classes are separable, the solution given by the algorithm just discussed may or may not produce a separating hyperplane. That is, a mean-square-error solution does not imply a solution in the sense of the perceptron training theorem. This uncertainty is the price of using an algorithm that converges under both the separable and nonseparable cases in this particular formulation.

The two perceptron training algorithms discussed thus far can be extended to more than two classes and to nonlinear decision functions. Based on the historical comments made earlier, exploring multiclass training algorithms here has little merit. Instead, we address multiclass training in the context of neural networks.

Multilayer feedforward neural networks

In this section we focus on decision functions of multiclass pattern recognition problems, independent of whether or not the classes are separable, and involving architectures that consist of layers of perceptron computing elements.

Basic architecture Figure 12.16 shows the architecture of the neural network model under consideration. It consists of layers of structurally identical computing nodes (neurons) arranged so that the output of every neuron in one layer feeds into the input of every neuron in the next layer. The number of neurons in the first layer, called layer A, is N_A. Often, $N_A = n$, the dimensionality of the input pattern vectors. The number of neurons in the output layer, called layer Q, is denoted N_Q. The number N_Q equals W, the number of pattern classes that the neural network has been trained to recognize. The network recognizes a pattern vector $\mathbf{x}$ as belonging to class ω_i if the ith output of the network is "high" while all other outputs are "low," as explained in the following discussion.

As the blowup in Fig. 12.16 shows, each neuron has the same form as the perceptron model discussed earlier (see Fig. 12.14), with the exception that the hard-limiting activation function has been replaced by a soft-limiting "sigmoid" function. Differentiability along all paths of the neural network is required in the development of the training rule. The following sigmoid activation function has the necessary differentiability:

$$h_j(I_j) = \frac{1}{1 + e^{-(I_j+\theta_j)/\theta_0}} \tag{12.2-47}$$

where I_j, $j = 1, 2, \ldots, N_J$, is the input to the activation element of each node in layer J of the network, θ_j is an offset, and θ_0 controls the shape of the sigmoid function.

Equation (12.2-47) is plotted in Fig. 12.17, along with the limits for the "high" and "low" responses out of each node. Thus when this particular function is used, the system outputs a high reading for any value of I_j greater than θ_j. Similarly, the system outputs a low reading for any value of I_j less than θ_j. As Fig. 12.17 shows, the sigmoid activation function always is positive, and it can reach its limiting values of 0 and 1 only if the input to the activation element is infinitely negative or positive, respectively. For this reason, values near 0 and 1

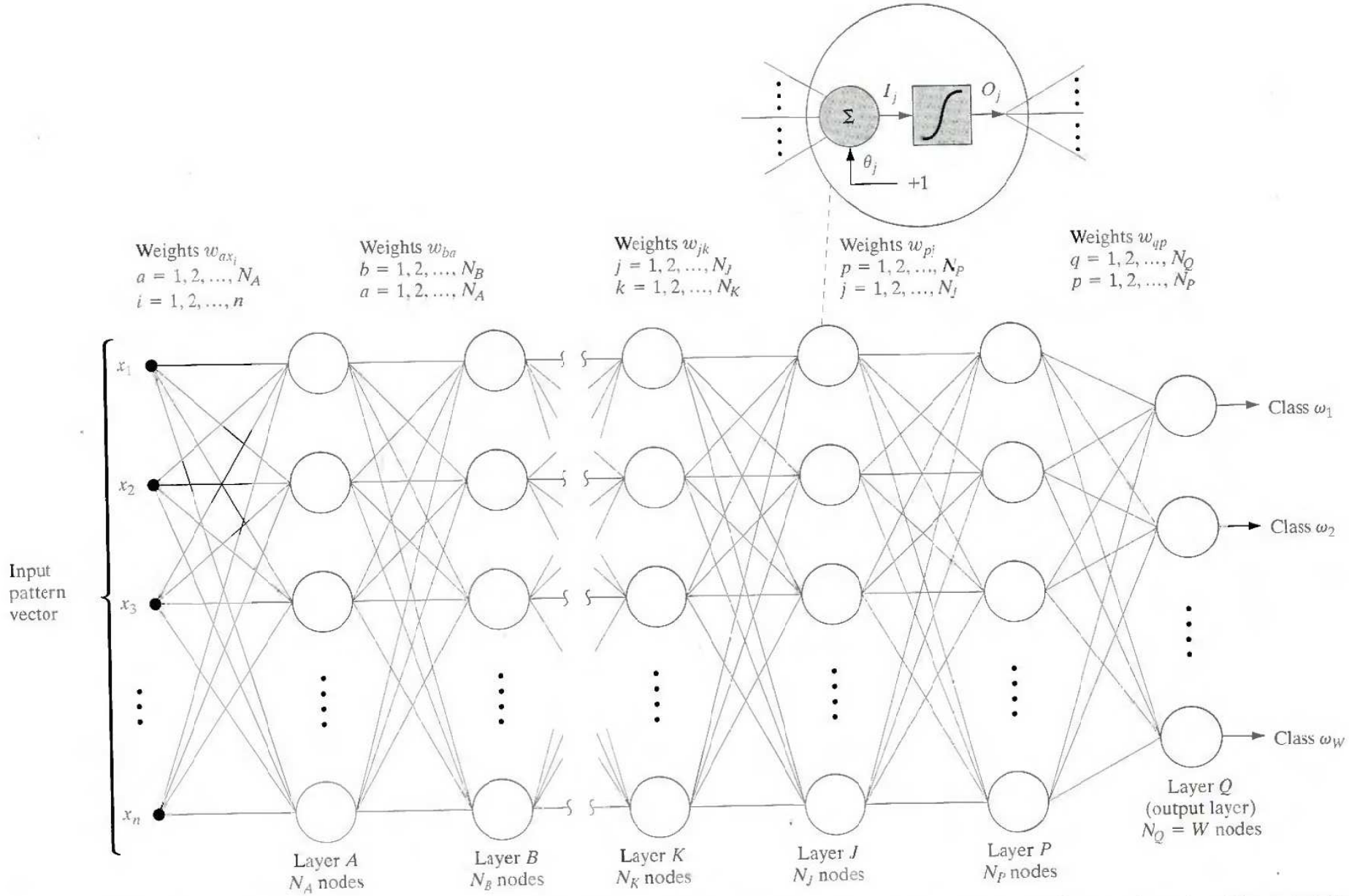


FIGURE 12.16 Multilayer feedforward neural network model. The blowup shows the basic structure of each neuron element throughout the network. The offset, θ_j, is treated as just another weight.

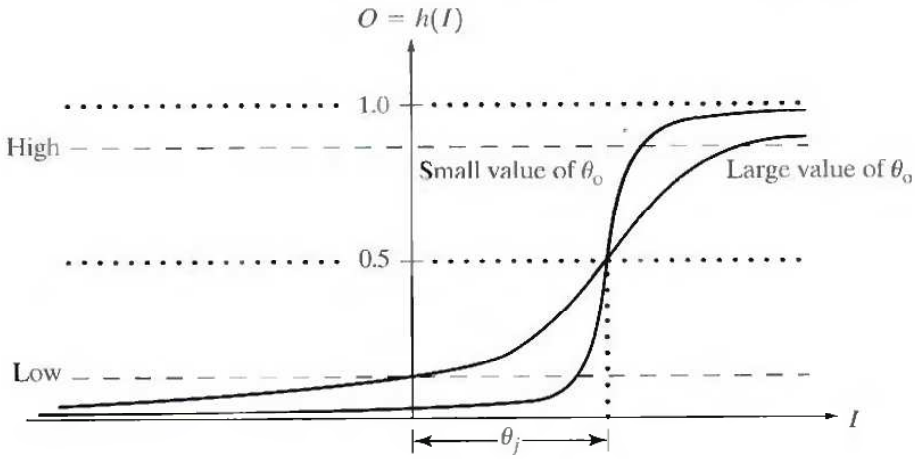


FIGURE 12.17 The sigmoidal activation function of Eq. (12.2-47).

(say, 0.05 and 0.95) define low and high values at the output of the neurons in Fig. 12.16. In principle, different types of activation functions could be used for different layers or even for different nodes in the same layer of a neural network. In practice, the usual approach is to use the same form of activation function throughout the network.

With reference to Fig. 12.14(a), the offset θ_j shown in Fig. 12.17 is analogous to the weight coefficient w_{n+1} in the earlier discussion of the perceptron. Implementation of this displaced threshold function can be done in the form of Fig. 12.14(a) by absorbing the offset θ_j as an additional coefficient that modifies a constant unity input to all nodes in the network. In order to follow the notation predominantly found in the literature, we do not show a separate constant input of +1 into all nodes of Fig. 12.16. Instead, this input and its modifying weight θ_j are integral parts of the network nodes. As noted in the blowup in Fig. 12.16, there is one such coefficient for each of the N_J nodes in layer J.

In Fig. 12.16, the input to a node in any layer is the weighted sum of the outputs from the previous layer. Letting layer K denote the layer preceding layer J (no alphabetical order is implied in Fig. 12.16) gives the input to the activation element of each node in layer J, denoted I_j:

$$I_j = \sum_{k=1}^{N_K} w_{jk} O_k \tag{12.2-48}$$

for $j = 1, 2, \ldots, N_J$, where N_J is the number of nodes in layer J, N_K is the number of nodes in layer K, and w_{jk} are the weights modifying the outputs O_k of the nodes in layer K before they are fed into the nodes in layer J. The outputs of layer K are

$$O_k = h_k(I_k) \tag{12.2-49}$$

for $k = 1, 2, \ldots, N_K$.

A clear understanding of the subscript notation used in Eq. (12.2-48) is important, because we use it throughout the remainder of this section. First, note that $I_j, j = 1, 2, \ldots, N_J$, represents the input to the *activation element* of the jth node in layer J. Thus I_1 represents the input to the activation element of the

Substituting Eqs. (12.2-53) and (12.2-54) into Eq. (12.2-52) yields

$$\begin{aligned}\Delta w_{qp} &= -\alpha \frac{\partial E_Q}{\partial I_q} O_p \\ &= \alpha \delta_q O_p \end{aligned} \tag{12.2-55}$$

where

$$\delta_q = -\frac{\partial E_Q}{\partial I_q}. \tag{12.2-56}$$

In order to compute $\partial E_Q / \partial I_q$, we use the chain rule to express the partial derivative in terms of the rate of change of E_Q with respect to O_q and the rate of change of O_q with respect to I_q. That is,

$$\delta_q = -\frac{\partial E_Q}{\partial I_q} = -\frac{\partial E_Q}{\partial O_q}\frac{\partial O_q}{\partial I_q}. \tag{12.2-57}$$

From Eq. (12.2-51),

$$\frac{\partial E_Q}{\partial O_q} = -(r_q - O_q) \tag{12.2-58}$$

and, from Eq. (12.2-49),

$$\frac{\partial O_Q}{\partial I_q} = \frac{\partial}{\partial I_q} h_q(I_q) = h'_q(I_q). \tag{12.2-59}$$

Substituting Eqs. (12.2-58) and (12.2-59) into Eq. (12.2-57) gives

$$\delta_q = (r_q - O_q) h'_q(I_q), \tag{12.2-60}$$

which is proportional to the error quantity $(r_q - O_q)$. Substitution of Eqs. (12.2-56) through (12.2-58) into Eq. (12.2-55) finally yields

$$\begin{aligned}\Delta w_{qp} &= \alpha (r_q - O_q) h'_q(I_q) O_p \\ &= \alpha \delta_q O_p. \end{aligned} \tag{12.2-61}$$

After the function $h_q(I_q)$ has been specified, all the terms in Eq. (12.2-61) are known or can be observed in the network. In other words, upon presentation of any training pattern to the input of the network, we know what the desired response, r_q, of each output node should be. The value O_q of each output node can be observed as can I_q, the input to the activation elements of layer Q, and O_p, the output of the nodes in layer P. Thus we know how to adjust the weights that modify the links between the last and next-to-last layers in the network.

Continuing to work our way back from the output layer, let us now analyze what happens at layer P. Proceeding in the same manner as above yields

$$\begin{aligned}\Delta w_{pj} &= \alpha (r_p - O_p) h'_p(I_p) O_j \\ &= \alpha \delta_p O_j \end{aligned} \tag{12.2-62}$$

where the error term is

$$\delta_p = (r_p - O_p) h'_p(I_p). \tag{12.2-63}$$

With the exception of r_p, all the terms in Eqs. (12.2-62) and (12.2-63) either are known or can be observed in the network. The term r_p makes no sense in an internal layer because we do not know what the response of an internal node in terms of pattern membership should be. We may specify what we want the response r to be only at the outputs of the network where final pattern classification takes place. If we knew that information at internal nodes, there would be no need for further layers. Thus we have to find a way to restate δ_p in terms of quantities that are known or can be observed in the network.

Going back to Eq. (12.2-57), we write the error term for layer P as

$$\delta_p = -\frac{\partial E_p}{\partial I_p} = -\frac{\partial E_p}{\partial O_p}\frac{\partial O_p}{\partial I_p}. \tag{12.2-64}$$

The term $\partial O_p/\partial I_p$ presents no difficulties. As before, it is

$$\frac{\partial O_p}{\partial I_p} = \frac{\partial h_p(I_p)}{\partial I_p} = h'_p(I_p), \tag{12.2-65}$$

which is known once h_p is specified because I_p can be observed. The term that produced r_p was the derivative $\partial E_p/\partial O_p$, so this term must be expressed in a way that does not contain r_p. Using the chain rule, we write the derivative as

$$\begin{aligned} -\frac{\partial E_P}{\partial O_p} &= -\sum_{q=1}^{N_Q}\frac{\partial E_P}{\partial I_q}\frac{\partial I_q}{\partial O_p} = \sum_{q=1}^{N_Q}\left(-\frac{\partial E_P}{\partial I_q}\right)\frac{\partial}{\partial O_p}\sum_{p=1}^{N_P} w_{qp}O_p \\ &= \sum_{q=1}^{N_Q}\left(-\frac{\partial E_P}{\partial I_q}\right)w_{qp} \\ &= \sum_{q=1}^{N_Q}\delta_q w_{qp} \end{aligned} \tag{12.2-66}$$

where the last step follows from Eq. (12.2-56). Substituting Eqs. (12.2-65) and (12.2-66) into Eq. (12.2-64) yields the desired expression for δ_p:

$$\delta_p = h'_p(I_p)\sum_{q=1}^{N_Q}\delta_q w_{qp}. \tag{12.2-67}$$

The parameter δ_p can be computed now because all its terms are known. Thus Eqs. (12.2-62) and (12.2-67) establish completely the training rule for layer P. The importance of Eq. (12.2-67) is that it computes δ_p from the quantities δ_q and w_{qp}, which are terms that were computed in the layer immediately following layer P. After the error term and weights have been computed for layer P, these quantities may be used similarly to compute the error and weights for the layer immediately preceding layer P. In other words, we have found a way to propagate the error back into the network, starting with the error at the output layer.

We may summarize and generalize the training procedure as follows. For any layers K and J, where layer K immediately precedes layer J, compute the weights w_{jk}, which modify the connections between these two layers, by using

$$\Delta w_{jk} = \alpha\delta_j O_k. \tag{12.2-68}$$

If layer J is the output layer, δ_j is

$$\delta_j = (r_j - O_j)h_j'(I_j). \tag{12.2-69}$$

If layer J is an internal layer and layer P is the next layer (to the right), then δ_j is given by

$$\delta_j = h_j'(I_j)\sum_{p=1}^{N_P}\delta_p w_{jp} \tag{12.2-70}$$

for $j = 1, 2, \ldots, N_j$. Using the activation function in Eq. (12.2-50) with $\theta_0 = 1$ yields

$$h_j'(I_j) = O_j(1 - O_j) \tag{12.2-71}$$

in which case Eqs. (12.2-69) and (12.2-70) assume the following, particularly attractive forms:

$$\delta_j = (r_j - O_j)O_j(1 - O_j) \tag{12.2-72}$$

for the output layer, and

$$\delta_j = O_j(1 - O_j)\sum_{p=1}^{N_P}\delta_p w_{jp} \tag{12.2-73}$$

for internal layers. In both Eqs. (12.2-72) and (12.2-73), $j = 1, 2, \ldots, N_J$.

Equations (12.2-68) through (12.2-70) constitute the generalized delta rule for training the multilayer feedforward neural network of Fig. 12.16. The process starts with an arbitrary (but not all equal) set of weights throughout the network. Then application of the generalized delta rule at any iterative step involves two basic phases. In the first phase, a training vector is presented to the network and is allowed to propagate through the layers to compute the output O_j for each node. The outputs O_q of the nodes in the output layer are then compared against their desired responses, r_p, to generate the error terms δ_q. The second phase involves a backward pass through the network during which the appropriate error signal is passed to each node and the corresponding weight changes are made. This procedure also applies to the bias weights θ_j. As discussed earlier in some detail, these are treated simply as additional weights that modify a unit input into the summing junction of every node in the network.

Common practice is to track the network error, as well as errors associated with individual patterns. In a successful training session, the network error decreases with the number of iterations and the procedure converges to a stable set of weights that exhibit only small fluctuations with additional training. The approach followed to establish whether a pattern has been classified correctly during training is to determine whether the response of the node in the output layer associated with the pattern class from which the pattern was obtained is high, while all the other nodes have outputs that are low, as defined earlier.

After the system has been trained, it classifies patterns using the parameters established during the training phase. In normal operation, all feedback paths are disconnected. Then any input pattern is allowed to propagate through the various layers, and the pattern is classified as belonging to the class of the output node that was high, while all the others were low. If more than one output is labeled high, or if none of the outputs is so labeled, the choice is one of declaring a misclassification or simply assigning the pattern to the class of the output node with the highest numerical value.

a
b

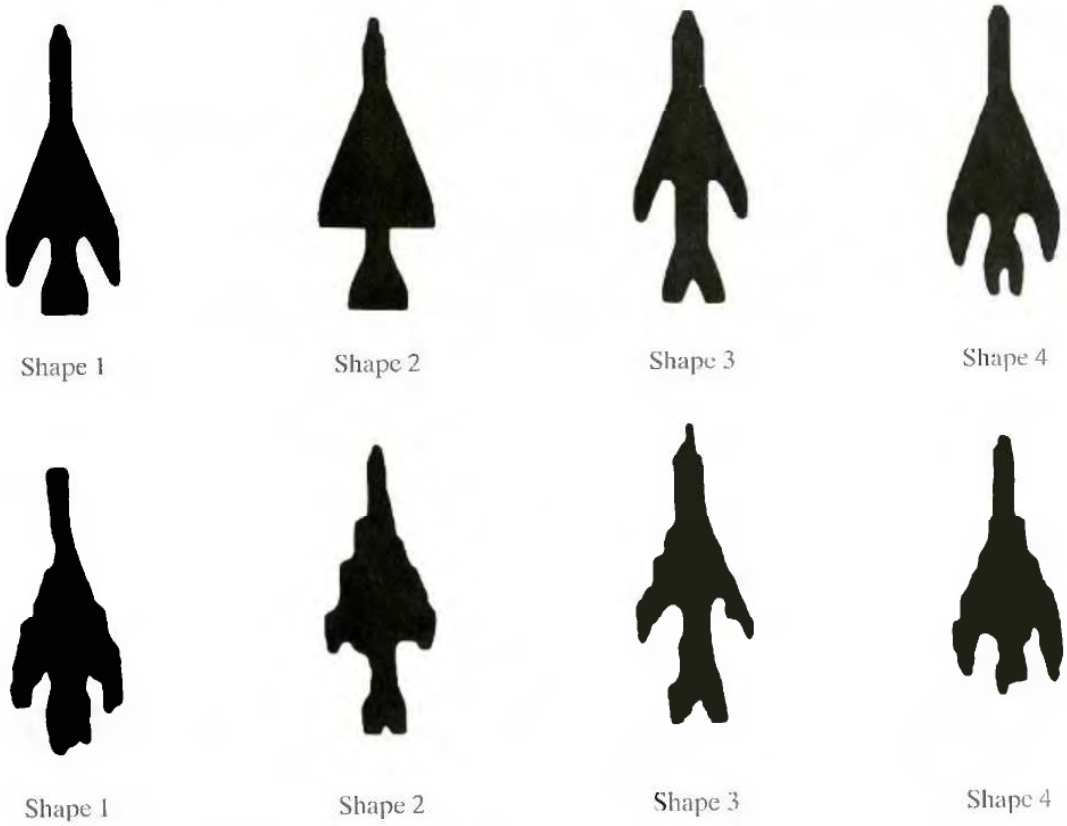


FIGURE 12.18 (a) Reference shapes and (b) typical noisy shapes used in training the neural network of Fig. 12.19. (Courtesy of Dr. Lalit Gupta, ECE Department, Southern Illinois University.)

EXAMPLE 12.6: Shape classification using a neural network.

We illustrate now how a neural network of the form shown in Fig. 12.16 was trained to recognize the four shapes shown in Fig. 12.18(a), as well as noisy versions of these shapes, samples of which are shown in Fig. 12.18(b).

Pattern vectors were generated by computing the normalized signatures of the shapes (see Section 11.1.3) and then obtaining 48 uniformly spaced samples of each signature. The resulting 48-dimensional vectors were the inputs to the three-layer feedforward neural network shown in Fig. 12.19. The number of neuron nodes in the first layer was chosen to be 48, corresponding to the dimensionality of the input pattern vectors. The four neurons in the third (output) layer correspond to the number of pattern classes, and the number of neurons in the middle layer was heuristically specified as 26 (the average of the number of neurons in the input and output layers). There are no known rules for specifying the number of nodes in the internal layers of a neural network, so this number generally is based either on prior experience or simply chosen arbitrarily and then refined by testing. In the output layer, the four nodes from top to bottom in this case represent the classes ω_j, $j = 1, 2, 3, 4$, respectively. After the network structure has been set, activation functions have to be selected for each unit and layer. All activation functions were selected to satisfy Eq. (12.2-50) with $\theta_0 = 1$ so that, according to our earlier discussion, Eqs. (12.2-72) and (12.2-73) apply.

The training process was divided in two parts. In the first part, the weights were initialized to small random values with zero mean, and the network was then trained with pattern vectors corresponding to noise-free samples like the

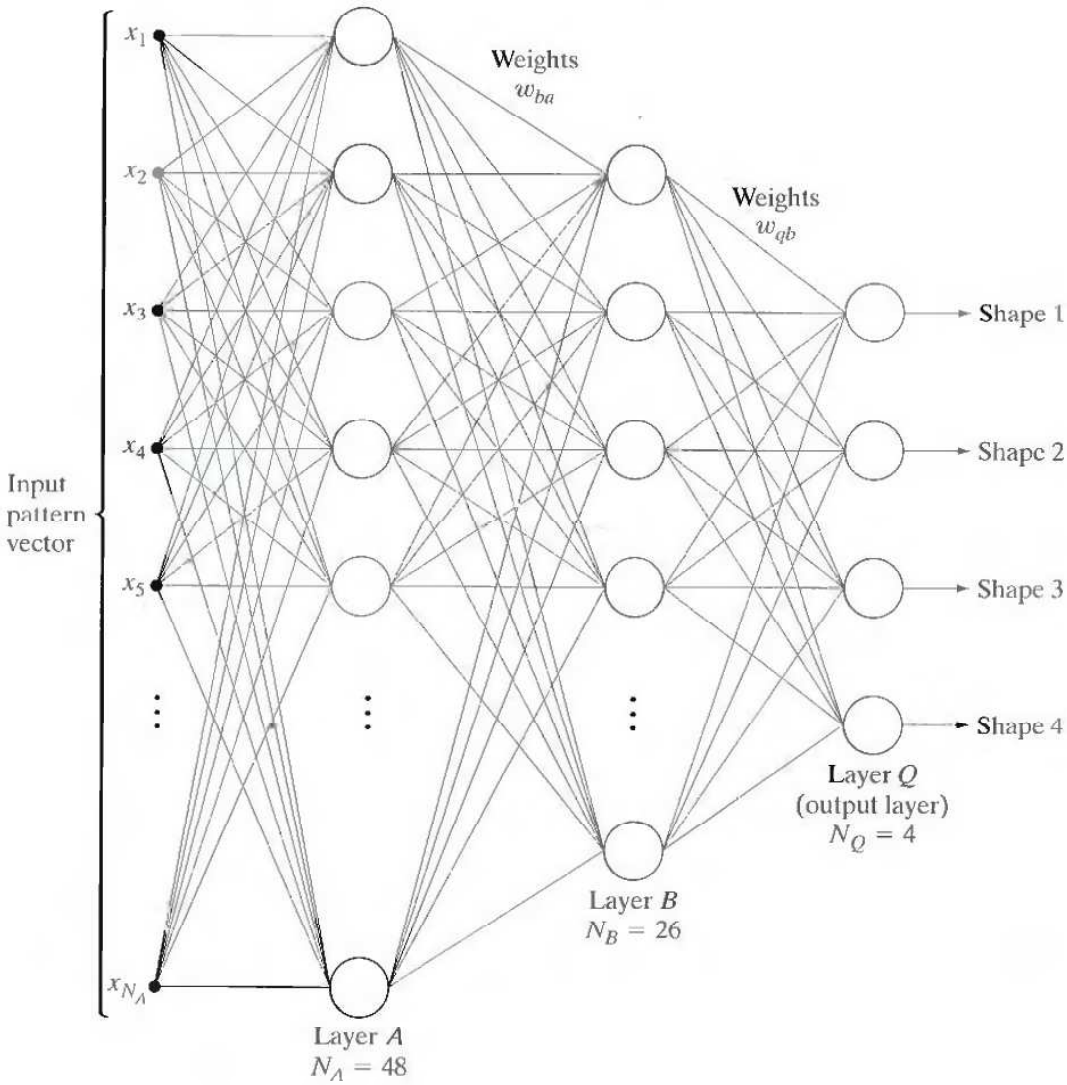


FIGURE 12.19 Three-layer neural network used to recognize the shapes in Fig. 12.18. (Courtesy of Dr. Lalit Gupta, ECE Department, Southern Illinois University.)

shapes shown in Fig. 12.18(a). The output nodes were monitored during training. The network was said to have learned the shapes from all four classes when, for any training pattern from class ω_i, the elements of the output layer yielded $O_i \geq 0.95$ and $O_q \leq 0.05$, for $q = 1, 2, \ldots, N_Q; q \neq i$. In other words, for any pattern of class ω_i, the output unit corresponding to that class had to be high (≥ 0.95) while, simultaneously, the output of all other nodes had to be low (≤ 0.05).

The second part of training was carried out with noisy samples, generated as follows. Each contour pixel in a noise-free shape was assigned a probability V of retaining its original coordinate in the image plane and a probability $R = 1 - V$ of being randomly assigned to the coordinates of one of its eight neighboring pixels. The degree of noise was increased by decreasing V (that is, increasing R). Two sets of noisy data were generated. The first consisted of 100 noisy patterns of each class generated by varying R between 0.1 and 0.6, giving a total of 400 patterns. This set, called the *test set*, was used to establish system a performance after training.

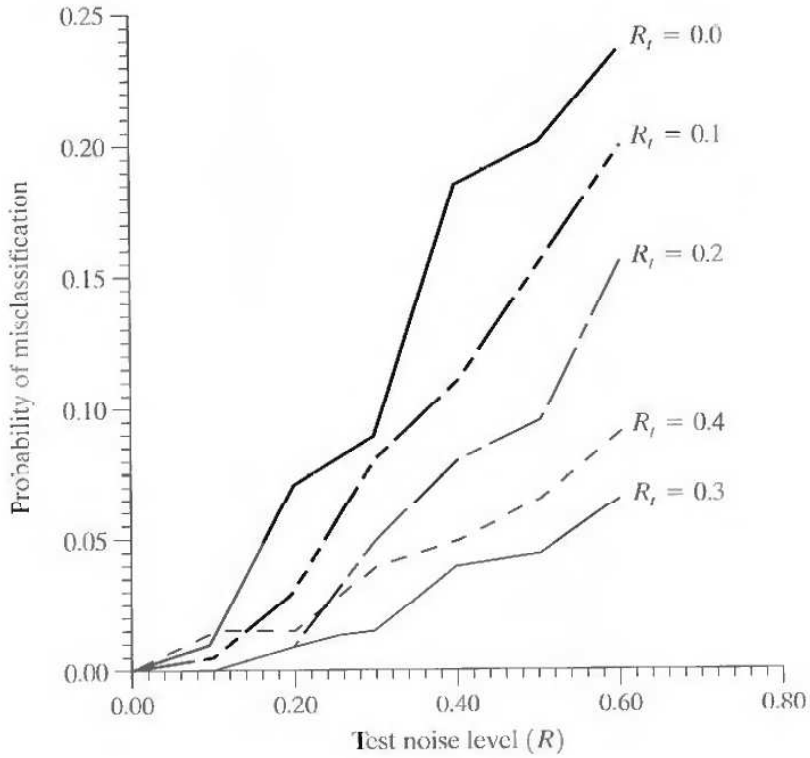


FIGURE 12.20 Performance of the neural network as a function of noise level. (Courtesy of Dr. Lalit Gupta, ECE Department, Southern Illinois University.)

Several noisy sets were generated for training the system with noisy data. The first set consisted of 10 samples for each class, generated by using $R_t = 0$, where R_t denotes a value of R used to generate training data. Starting with the weight vectors obtained in the first (noise-free) part of training, the system was allowed to go through a learning sequence with the new data set. Because $R_t = 0$ implies no noise, this retraining was an extension of the earlier, noise-free training. Using the resulting weights learned in this manner, the network was subjected to the test data set yielding the results shown by the curve labeled $R_t = 0$ in Fig. 12.20. The number of misclassified patterns divided by the total number of patterns tested gives the probability of misclassification, which is a measure commonly used to establish neural network performance.

Next, starting with the weight vectors learned by using the data generated with $R_t = 0$, the system was retrained with a noisy data set generated with $R_t = 0.1$. The recognition performance was then established by running the test samples through the system again with the new weight vectors. Note the significant improvement in performance. Figure 12.20 shows the results obtained by continuing this retraining and retesting procedure for $R_t = 0.2, 0.3$, and 0.4. As expected if the system is learning properly, the probability of misclassifying patterns from the test set decreased as the value of R_t increased because the system was being trained with noisier data for higher values of R_t. The one exception in Fig. 12.20 is the result for $R_t = 0.4$. The reason is the small number of samples used to train the system. That is, the network was not able to adapt itself sufficiently to the larger variations in shape at higher noise levels with the number of samples used. This hypothesis is verified by the results in Fig. 12.21, which show a lower probability of misclassification as the number of training

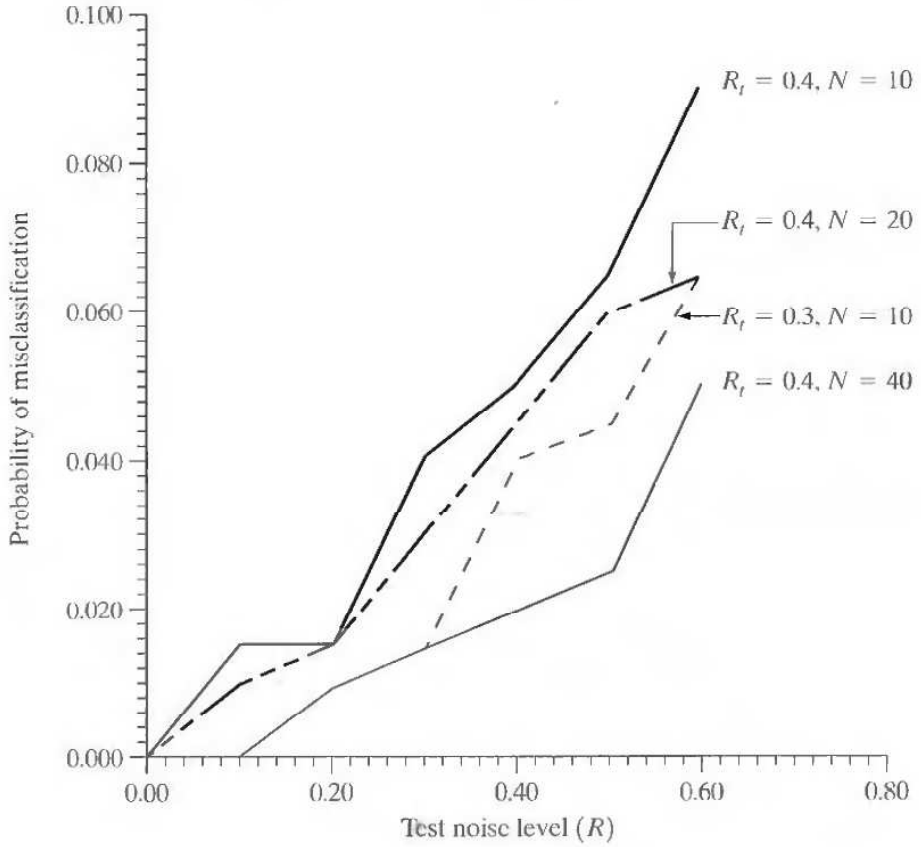


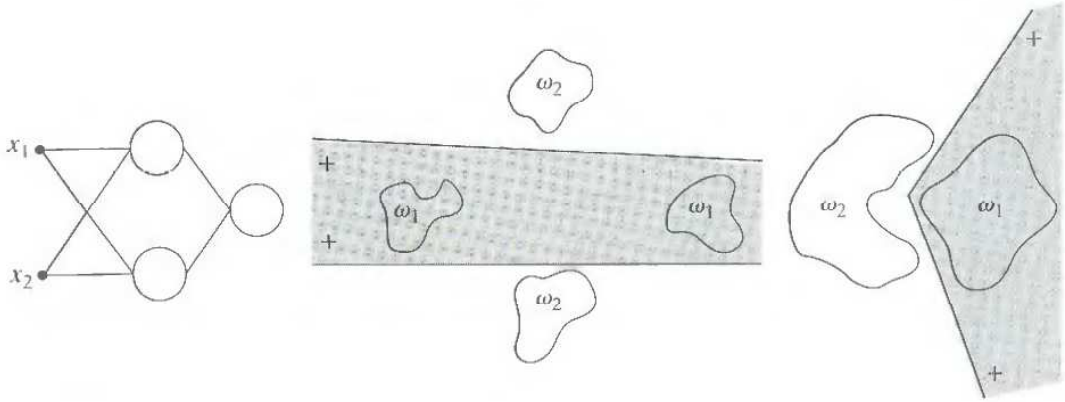
FIGURE 12.21 Improvement in performance for $R_t = 0.4$ by increasing the number of training patterns (the curve for $R_t = 0.3$ is shown for reference). (Courtesy of Dr. Lalit Gupta, ECE Department, Southern Illinois University.)

samples was increased. Figure 12.21 also shows as a reference the curve for $R_t = 0.3$ from Fig. 12.20.

The preceding results show that a three-layer neural network was capable of learning to recognize shapes corrupted by noise after a modest level of training. Even when trained with noise-free data ($R_t = 0$ in Fig. 12.20), the system was able to achieve a correct recognition level of close to 77% when tested with data highly corrupted by noise ($R = 0.6$ in Fig. 12.20). The recognition rate on the same data increased to about 99% when the system was trained with noisier data ($R_t = 0.3$ and 0.4). It is important to note that the system was trained by increasing its classification power via systematic, small incremental additions of noise. When the nature of the noise is known, this method is ideal for improving the convergence and stability properties of a neural network during learning. ■

Complexity of decision surfaces We have already established that a single-layer perceptron implements a hyperplane decision surface. A natural question at this point is, What is the nature of the decision surfaces implemented by a multilayer network, such as the model in Fig. 12.16? It is demonstrated in the following discussion that a three-layer network is capable of implementing arbitrarily complex decision surfaces composed of intersecting hyperplanes.

As a starting point, consider the two-input, two-layer network shown in Fig. 12.22(a). With two inputs, the patterns are two dimensional, and therefore, each node in the first layer of the network implements a line in 2-D space. We



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.

denote by 1 and 0, respectively, the high and low outputs of these two nodes. We assume that a 1 output indicates that the corresponding input vector to a node in the first layer lies on the positive side of the line. Then the possible combinations of outputs feeding the single node in the second layer are (1, 1), (1, 0), (0, 1), and (0, 0). If we define two regions, one for class ω_1 lying on the positive side of both lines and the other for class ω_2 lying anywhere else, the output node can classify any input pattern as belonging to one of these two regions simply by performing a logical AND operation. In other words, the output node responds with a 1, indicating class ω_1, only when both outputs from the first layer are 1. The AND operation can be performed by a neural node of the form discussed earlier if θ_j is set to a value in the half open interval (1, 2]. Thus if we assume 0 and 1 responses out of the first layer, the response of the output node will be high, indicating class ω_1, only when the sum performed by the neural node on the two outputs from the first layer is greater than 1. Figures 12.22(b) and (c) show how the network of Fig. 12.22(a) can successfully dichotomize two pattern classes that could not be separated by a single linear surface.

If the number of nodes in the first layer were increased to three, the network of Fig. 12.22(a) would implement a decision boundary consisting of the intersection of three lines. The requirement that class ω_1 lie on the positive side of all three lines would yield a convex region bounded by the three lines. In fact, an arbitrary open or closed convex region can be constructed simply by increasing the number of nodes in the first layer of a two-layer neural network.

The next logical step is to increase the number of layers to three. In this case the nodes of the first layer implement lines, as before. The nodes of the second layer then perform AND operations in order to form regions from the various lines. The nodes in the third layer assign class membership to the various regions. For instance, suppose that class ω_1 consists of two distinct regions, each of which is bounded by a different set of lines. Then two of the nodes in the second layer are for regions corresponding to the same pattern class. One of the output nodes

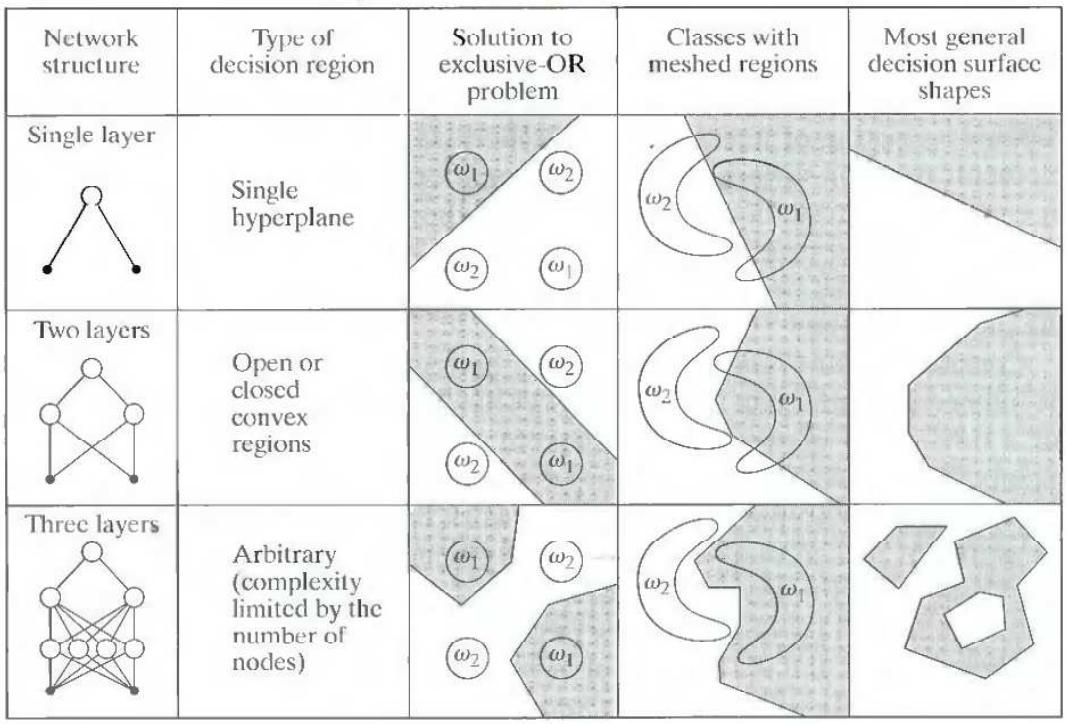


FIGURE 12.23 Types of decision regions that can be formed by single- and multilayer feed-forward networks with one and two layers of hidden units and two inputs. (Lippman)

needs to be able to signal the presence of that class when either of the two nodes in the second layer goes high. Assuming that high and low conditions in the second layer are denoted 1 and 0, respectively, this capability is obtained by making the output nodes of the network perform the logical OR operation. In terms of neural nodes of the form discussed earlier, we do so by setting θ_j to a value in the half-open interval $[0, 1)$. Then, whenever at least one of the nodes in the second layer associated with that output node goes high (outputs a 1), the corresponding node in the output layer will go high, indicating that the pattern being processed belongs to the class associated with that node.

Figure 12.23 summarizes the preceding comments. Note in the third row that the complexity of decision regions implemented by a three-layer network is, in principle, arbitrary. In practice, a serious difficulty usually arises in structuring the second layer to respond correctly to the various combinations associated with particular classes. The reason is that lines do not just stop at their intersection with other lines, and, as a result, patterns of the same class may occur on both sides of lines in the pattern space. In practical terms, the second layer may have difficulty figuring out which lines should be included in the AND operation for a given pattern class—or it may even be impossible. The reference to the exclusive-OR problem in the third column of Fig. 12.23 deals with the fact that, if the input patterns were binary, only four different patterns could be constructed in two dimensions. If the patterns are so arranged that class ω_1 consists of patterns $\{(0, 1), (1, 0)\}$ and class ω_2 consists of the patterns $\{(0, 0), (1, 1)\}$, class membership of the patterns in these two classes is given by the exclusive-OR (XOR) logical function, which is 1 only when one or the other of the two variables is 1, and it is 0 otherwise. Thus an XOR value of 1 indicates patterns of class ω_1, and an XOR value of 0 indicates patterns of class ω_2.

The preceding discussion is generalized to n dimensions in a straightforward way: Instead of lines, we deal with hyperplanes. A single-layer network implements a single hyperplane. A two-layer network implements arbitrarily convex regions consisting of intersections of hyperplanes. A three-layer network implements decision surfaces of arbitrary complexity. The number of nodes used in each layer determines the complexity of the last two cases. The number of classes in the first case is limited to two. In the other two cases, the number of classes is arbitrary, because the number of output nodes can be selected to fit the problem at hand.

Considering the preceding comments, it is logical to ask, Why would anyone be interested in studying neural networks having more than three layers? After all, a three-layer network can implement decision surfaces of arbitrary complexity. The answer lies in the method used to train a network to utilize only three layers. The training rule for the network in Fig. 12.16 minimizes an error measure but says nothing about how to associate groups of hyperplanes with specific nodes in the second layer of a three-layer network of the type discussed earlier. In fact, the problem of how to perform trade-off analyses between the number of layers and the number of nodes in each layer remains unresolved. In practice, the trade-off is generally resolved by trial and error or by previous experience with a given problem domain.

12.3 Structural Methods

The techniques discussed in Section 12.2 deal with patterns quantitatively and largely ignore any structural relationships inherent in a pattern's shape. The structural methods discussed in this section, however, seek to achieve pattern recognition by capitalizing precisely on these types of relationships.

12.3.1 Matching Shape Numbers

A procedure analogous to the minimum distance concept introduced in Section 12.2.1 for pattern vectors can be formulated for the comparison of region boundaries that are described in terms of shape numbers. With reference to the discussion in Section 11.2.2, the *degree of similarity*, k, between two region boundaries (shapes) is defined as the largest order for which their shape numbers still coincide. For example, let a and b denote shape numbers of closed boundaries represented by 4-directional chain codes. These two shapes have a degree of similarity k if

$$\begin{aligned} s_j(a) &= s_j(b) \qquad \text{for } j = 4, 6, 8, \ldots, k \\ s_j(a) &\neq s_j(b) \qquad \text{for } j = k + 2, k + 4, \ldots \end{aligned} \tag{12.3-1}$$

where s indicates shape number and the subscript indicates order. The *distance* between two shapes a and b is defined as the inverse of their degree of similarity:

$$D(a, b) = \frac{1}{k}. \tag{12.3-2}$$

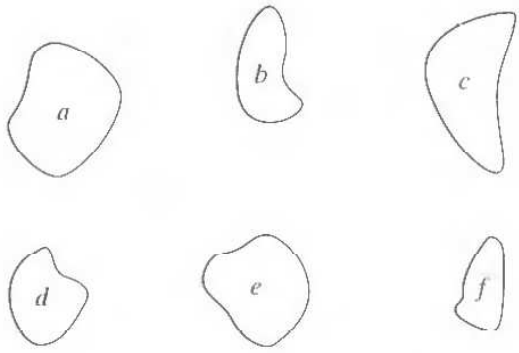
This distance satisfies the following properties:

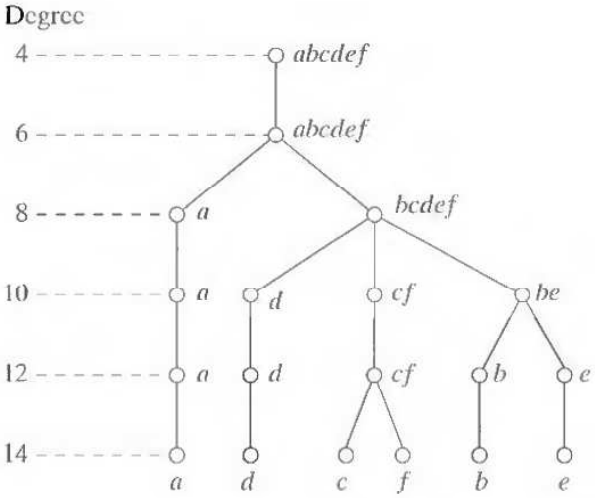
$$
\begin{aligned}
&D(a, b) \geq 0 \\
&D(a, b) = 0 \quad \text{iff } a = b \\
&D(a, c) \leq \max\left[D(a, b), D(b, c)\right].
\end{aligned}
\tag{12.3-3}
$$

Either k or D may be used to compare two shapes. If the degree of similarity is used, the larger k is, the more similar the shapes are (note that k is infinite for identical shapes). The reverse is true when the distance measure is used.

EXAMPLE 12.7: Using shape numbers to compare shapes.

■ Suppose that we have a shape f and want to find its closest match in a set of five other shapes (a, b, c, d, and e), as shown in Fig. 12.24(a). This problem is analogous to having five prototype shapes and trying to find the best match to a given unknown shape. The search may be visualized with the aid of the similarity tree shown in Fig. 12.24(b). The root of the tree corresponds to the lowest possible degree of similarity, which, for this example, is 4. Suppose that the shapes are identical up to degree 8, with the exception of shape a, whose degree of similarity with respect to all other shapes is 6. Proceeding down the tree, we find that shape d has degree of similarity 8 with respect to all others, and so on. Shapes





	a	b	c	d	e	f
a	∞	6	6	6	6	6
b		∞	8	8	10	8
c			∞	8	8	12
d				∞	8	8
e					∞	8
f						∞

a
b c

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

f and c match uniquely, having a higher degree of similarity than any other two shapes. At the other extreme, if a had been an unknown shape, all we could have said using this method is that a was similar to the other five shapes with degree of similarity 6. The same information can be summarized in the form of a *similarity matrix*, as shown in Fig. 12.24(c). ■

12.3.2 String Matching

Suppose that two region boundaries, a and b, are coded into strings (see Section 11.5) denoted $a_1a_2, \ldots, a_n$ and $b_1b_2, \ldots, b_m$, respectively. Let α represent the number of matches between the two strings, where a match occurs in the kth position if $a_k = b_k$. The number of symbols that do not match is

$$\beta = \max(|a|, |b|) - \alpha \tag{12.3-4}$$

where $|\text{arg}|$ is the length (number of symbols) in the string representation of the argument. It can be shown that $\beta = 0$ if and only if a and b are identical (see Problem 12.21).

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

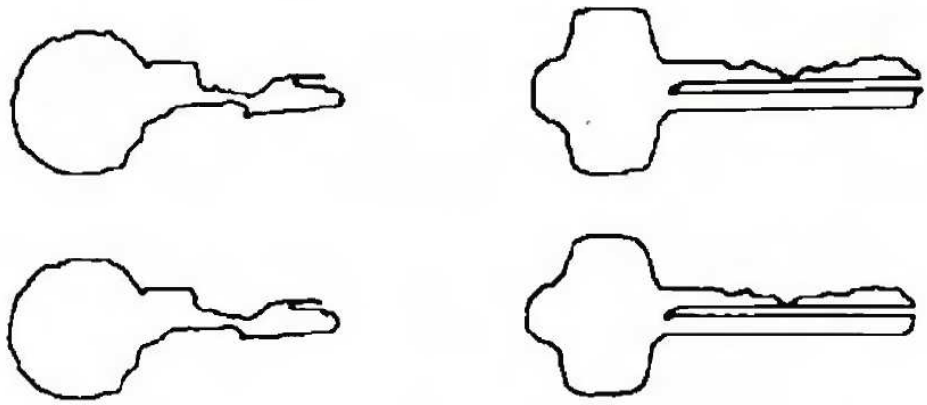
$$R = \frac{\alpha}{\beta} = \frac{\alpha}{\max(|a|, |b|) - \alpha}. \tag{12.3-5}$$

Hence R is infinite for a perfect match and 0 when none of the symbols in a and b match ($\alpha = 0$ in this case). Because matching is done symbol by symbol, the starting point on each boundary is important in terms of reducing the amount of computation. Any method that normalizes to, or near, the same starting point is helpful, so long as it provides a computational advantage over brute-force matching, which consists of starting at arbitrary points on each string and then shifting one of the strings (with wraparound) and computing Eq. (12.3-5) for each shift. The largest value of R gives the best match.

EXAMPLE 12.8: Illustration of string matching.

■ Figures 12.25(a) and (b) show sample boundaries from each of two object classes, which were approximated by a polygonal fit (see Section 11.1.2). Figures 12.25(c) and (d) show the polygonal approximations corresponding to the boundaries shown in Figs. 12.25(a) and (b), respectively. 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° increments; that is, $\alpha_1: 0° < \theta \leq 45°$; $\alpha_2: 45° < \theta \leq 90°; \ldots; \alpha_8: 315° < \theta \leq 360°$.

Figure 12.25(e) shows the results of computing the measure R for five samples of object 1 against themselves. The entries correspond to R values and, for example, the notation 1.c refers to the third string from object class 1. Figure 12.25(f) shows the results of comparing the strings of the second object class against themselves. Finally, Fig. 12.25(g) shows a tabulation of R values obtained by comparing strings of one class against the other. Note that, here, all R values are considerably smaller than any entry in the two preceding tabulations, indicating that the R measure achieved a high degree of discrimination between the two classes of objects. For example, if the class membership of string 1.a had



a b
c d
e f
g

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.)

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	∞

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

been unknown, the *smallest* value of R resulting from comparing this string against sample (prototype) strings of class 1 would have been 4.7 [Fig. 12.25(e)]. By contrast, the *largest* value in comparing it against strings of class 2 would have been 1.24 [Fig. 12.25(g)]. This result would have led to the conclusion that string 1.a is a member of object class 1. This approach to classification is analogous to the minimum distance classifier introduced in Section 12.2.1. ■

12.3.3 Syntactic Recognition of Strings

Syntactic methods provide a unified methodology for handling structural recognition problems. Basically, the idea behind syntactic pattern recognition is the specification of a set of pattern *primitives* (see Section 11.5), a set of rules (in the form of a *grammar*) that governs their interconnection, and a *recognizer* (called an *automaton*), whose structure is determined by the set of rules in the grammar. First we consider string grammars and automata and then extend these ideas in the next section to tree grammars and their corresponding automata.

String grammars

Suppose that we have two classes, ω_1 and ω_2, whose patterns are strings of primitives, generated by one of the methods discussed in Section 11.5. We can interpret each primitive as being a symbol permissible in the *alphabet* of some *grammar*, where a grammar is a set of rules of syntax (hence the name syntactic recognition) that govern the generation of *sentences* formed from symbols of the alphabet. The set of sentences generated by a grammar, G, is called its *language* and is denoted $L(G)$. Here, sentences are strings of symbols (which in turn represent patterns), and languages correspond to pattern classes.

Consider two grammars, G_1 and G_2, whose rules of syntax are such that G_1 only allows generation of sentences that correspond to patterns from class ω_1, and G_2 only allows generation of sentences corresponding to patterns from class ω_2. After two grammars with these properties have been established, the syntactic pattern recognition process, in principle, is straightforward. For a sentence representing an unknown pattern, the task is to decide in which language the pattern represents a valid sentence. If the sentence belongs to $L(G_1)$, we say that the pattern is from class ω_1. Similarly, the pattern is said to be from class ω_2 if the sentence is valid in $L(G_2)$. A unique decision cannot be made if the sentence belongs to both languages. A sentence that is invalid in both languages is rejected.

When there are more than two pattern classes, the syntactic classification approach is the same as described in the preceding paragraph, with the exception that more grammars (at least one per class) are involved in the process. For multiclass classification, a pattern belongs to class ω_i if it is a valid sentence only of $L(G_i)$. As before, a unique decision cannot be made if a sentence belongs to more than one language. A sentence that is invalid over all languages is rejected.

When dealing with strings, we define a grammar as the 4-tuple

$$G = (N, \Sigma, P, S) \tag{12.3-6}$$

where

N is a finite set of variables called *nonterminals,*

Σ is a finite set of constants called *terminals,*

P is a set of rewriting rules called *productions,* and

S in N is called the *starting symbol.*

It is required that N and Σ be disjoint sets. In the following discussion, capital letters, $A, B, \ldots, S, \ldots$, denote nonterminals. Lowercase letters, $a, b, c, \ldots$ at the beginning of the alphabet denote terminals. Lowercase letters, v, w, x, y, z toward the end of the alphabet denote strings of terminals. Lowercase Greek letters $\alpha, \beta, \theta, \ldots$ denote strings of mixed terminals and nonterminals. The *empty sentence* (the sentence with no symbols) is denoted λ. Finally, for a set V of symbols, the notation V^* denotes the set of all sentences composed of elements from V.

String grammars are characterized by the form of their productions. Of particular interest in syntactic pattern recognition are *regular grammars* and *context-free grammars*. Regular grammars have productions only of the form $A \rightarrow aB$ or

is to adjust $\mathbf{w}$ incrementally in the direction of the negative gradient of $J(\mathbf{w})$ in order to seek the minimum of this function, which occurs when $r = \mathbf{w}^T\mathbf{y}$; that is, the minimum corresponds to correct classification. If $\mathbf{w}(k)$ represents the weight vector at the kth iterative step, a general gradient descent algorithm may be written as

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \alpha\left[\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}}\right]_{\mathbf{w}=\mathbf{w}(k)} \tag{12.2-38}$$

where $\mathbf{w}(k+1)$ is the new value of $\mathbf{w}$, and $\alpha > 0$ gives the magnitude of the correction. From Eq. (12.2-37),

$$\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} = -(r - \mathbf{w}^T\mathbf{y})\mathbf{y}. \tag{12.2-39}$$

Substituting this result into Eq. (12.2-38) yields

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \alpha[r(k) - \mathbf{w}^T(k)\mathbf{y}(k)]\mathbf{y}(k) \tag{12.2-40}$$

with the starting weight vector, $\mathbf{w}(1)$, being arbitrary.

By defining the change (delta) in weight vector as

$$\Delta\mathbf{w} = \mathbf{w}(k+1) - \mathbf{w}(k) \tag{12.2-41}$$

we can write Eq. (12.2-40) in the form of a *delta correction algorithm:*

$$\Delta\mathbf{w} = \alpha e(k)\mathbf{y}(k) \tag{12.2-42}$$

where

$$e(k) = r(k) - \mathbf{w}^T(k)\mathbf{y}(k) \tag{12.2-43}$$

is the error committed with weight vector $\mathbf{w}(k)$ when pattern $\mathbf{y}(k)$ is presented.

Equation (12.2-43) gives the error with weight vector $\mathbf{w}(k)$. If we change it to $\mathbf{w}(k+1)$, but leave the pattern the same, the error becomes

$$e(k) = r(k) - \mathbf{w}^T(k+1)\mathbf{y}(k). \tag{12.2-44}$$

The change in error then is

$$\begin{aligned}\Delta e(k) &= [r(k) - \mathbf{w}^T(k+1)\mathbf{y}(k)] - [r(k) - \mathbf{w}^T(k)\mathbf{y}(k)] \\ &= -[\mathbf{w}^T(k+1) - \mathbf{w}^T(k)]\mathbf{y}(k) \\ &= -\Delta\mathbf{w}^T\mathbf{y}(k).\end{aligned} \tag{12.2-45}$$

But $\Delta\mathbf{w} = \alpha e(k)\mathbf{y}(k)$, so

$$\begin{aligned}\Delta e &= -\alpha e(k)\mathbf{y}^T(k)\mathbf{y}(k) \\ &= -\alpha e(k)\|\mathbf{y}(k)\|^2.\end{aligned} \tag{12.2-46}$$

Hence changing the weights reduces the error by a factor $\alpha\|\mathbf{y}(k)\|^2$. The next input pattern starts the new adaptation cycle, reducing the next error by a factor $\alpha\|\mathbf{y}(k+1)\|^2$, and so on.

The choice of α controls stability and speed of convergence (Widrow and Stearns [1985]). Stability requires that $0 < \alpha < 2$. A practical range for α is $0.1 < \alpha < 1.0$. Although the proof is not shown here, the algorithm of

Eq. (12.2-40) or Eqs. (12.2-42) and (12.2-43) converges to a solution that minimizes the mean square error over the patterns of the training set. When the pattern classes are separable, the solution given by the algorithm just discussed may or may not produce a separating hyperplane. That is, a mean-square-error solution does not imply a solution in the sense of the perceptron training theorem. This uncertainty is the price of using an algorithm that converges under both the separable and nonseparable cases in this particular formulation.

The two perceptron training algorithms discussed thus far can be extended to more than two classes and to nonlinear decision functions. Based on the historical comments made earlier, exploring multiclass training algorithms here has little merit. Instead, we address multiclass training in the context of neural networks.

Multilayer feedforward neural networks

In this section we focus on decision functions of multiclass pattern recognition problems, independent of whether or not the classes are separable, and involving architectures that consist of layers of perceptron computing elements.

Basic architecture Figure 12.16 shows the architecture of the neural network model under consideration. It consists of layers of structurally identical computing nodes (neurons) arranged so that the output of every neuron in one layer feeds into the input of every neuron in the next layer. The number of neurons in the first layer, called layer A, is N_A. Often, $N_A = n$, the dimensionality of the input pattern vectors. The number of neurons in the output layer, called layer Q, is denoted N_Q. The number N_Q equals W, the number of pattern classes that the neural network has been trained to recognize. The network recognizes a pattern vector $\mathbf{x}$ as belonging to class ω_i if the ith output of the network is "high" while all other outputs are "low," as explained in the following discussion.

As the blowup in Fig. 12.16 shows, each neuron has the same form as the perceptron model discussed earlier (see Fig. 12.14), with the exception that the hard-limiting activation function has been replaced by a soft-limiting "sigmoid" function. Differentiability along all paths of the neural network is required in the development of the training rule. The following sigmoid activation function has the necessary differentiability:

$$h_j(I_j) = \frac{1}{1 + e^{-(I_j+\theta_j)/\theta_0}} \tag{12.2-47}$$

where I_j, $j = 1, 2, \ldots, N_J$, is the input to the activation element of each node in layer J of the network, θ_j is an offset, and θ_0 controls the shape of the sigmoid function.

Equation (12.2-47) is plotted in Fig. 12.17, along with the limits for the "high" and "low" responses out of each node. Thus when this particular function is used, the system outputs a high reading for any value of I_j greater than θ_j. Similarly, the system outputs a low reading for any value of I_j less than θ_j. As Fig. 12.17 shows, the sigmoid activation function always is positive, and it can reach its limiting values of 0 and 1 only if the input to the activation element is infinitely negative or positive, respectively. For this reason, values near 0 and 1

first (topmost) node in layer J, I_2 represents the input to the activation element of the second node in layer J, and so on. There are N_K inputs to every node in layer J, but *each* individual input can be weighted differently. Thus the N_K inputs to the first node in layer J are weighted by coefficients $w_{1k}, k = 1, 2, \ldots, N_K$; the inputs to the second node are weighted by coefficients $w_{2k}, k = 1, 2, \ldots, N_K$; and so on. Hence a total of $N_J \times N_K$ coefficients are necessary to specify the weighting of the outputs of layer K as they are fed into layer J. An additional N_J offset coefficients, θ_j, are needed to specify completely the nodes in layer J.

Substitution of Eq. (12.2-48) into (12.2-47) yields

$$h_j(I_j) = \frac{1}{1 + e^{-\left(\sum_{k=1}^{N_K} w_{jk} O_k + \theta_j\right)/\theta_0}}, \tag{12.2-50}$$

which is the form of activation function used in the remainder of this section.

During training, adapting the neurons in the output layer is a simple matter because the desired output of each node is known. The main problem in training a multilayer network lies in adjusting the weights in the so-called *hidden layers*. That is, in those other than the output layer.

Training by back propagation We begin by concentrating on the output layer. The total squared error between the desired responses, r_q, and the corresponding actual responses, O_q, of nodes in (output) layer Q, is

$$E_Q = \frac{1}{2} \sum_{q=1}^{N_Q} (r_q - O_q)^2 \tag{12.2-51}$$

where N_Q is the number of nodes in output layer Q and the $\frac{1}{2}$ is used for convenience in notation for taking the derivative later.

The objective is to develop a training rule, similar to the delta rule, that allows adjustment of the weights in each of the layers in a way that seeks a minimum to an error function of the form shown in Eq. (12.2-51). As before, adjusting the weights in proportion to the partial derivative of the error with respect to the weights achieves this result. In other words,

$$\Delta w_{qp} = -\alpha \frac{\partial E_Q}{\partial w_{qp}} \tag{12.2-52}$$

where layer P precedes layer Q, Δw_{qp} is as defined in Eq. (12.2-42), and α is a positive correction increment.

The error E_Q is a function of the outputs, O_q, which in turn are functions of the inputs I_q. Using the chain rule, we evaluate the partial derivative of E_Q as follows:

$$\frac{\partial E_Q}{\partial w_{qp}} = \frac{\partial E_Q}{\partial I_q} \frac{\partial I_q}{\partial w_{qp}}. \tag{12.2-53}$$

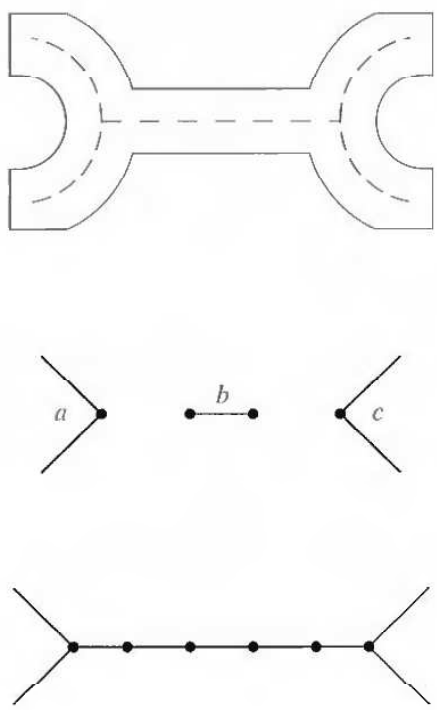
From Eq. (12.2-48),

$$\frac{\partial I_q}{\partial w_{qp}} = \frac{\partial}{\partial w_{qp}} \sum_{p=1}^{N_P} w_{qp} O_p = O_p. \tag{12.2-54}$$

$A \rightarrow a$, with A and B in N and a in Σ. Context-free grammars have productions only of the form $A \rightarrow \alpha$, with A in N and α in the set $(N \cup \Sigma)^*$; that is, α can be any string composed of terminals and nonterminals, except the empty string.

EXAMPLE 12.9: Object class generation using a regular string grammar.

■ Before proceeding, it will be useful to consider the mechanics of how grammars generate object classes. Suppose that the object shown in Fig. 12.26(a) is represented by its (pruned) skeleton and that we define the primitives shown in Fig. 12.26(b) to describe the structure of this (and similar) skeletons. Consider the grammar $G = (N, \Sigma, P, S)$, with $N = \{A, B, S\}$, $\Sigma = \{a, b, c\}$ and $P = \{S \rightarrow aA, A \rightarrow bA, A \rightarrow bB, B \rightarrow c\}$, where the terminals a, b, and c correspond to the primitives shown in Fig. 12.26(b). As indicated earlier, S is the starting symbol from which the strings of $L(G)$ are generated. For instance, applying the first production followed by two applications of the second production yields $S \Rightarrow aA \rightarrow abA \rightarrow abbA$, where $(\Rightarrow)$ indicates a string derivation starting from S and using productions from the set P. The first production allowed rewriting S as aA, and the second production allowed rewriting A as bA. With a nonterminal in the string $abbA$, we can continue the derivation. For example, applying the second production two more times, followed by one application of the third production and one application of the fourth production, yields the string $abbbbbc$, which corresponds to the structure shown in Fig. 12.26(c). No nonterminals remain after application of the fourth production, so the derivation terminates when this production is used. The language generated by the rules of this grammar is $L(G) = \{ab^n c | n \geq 1\}$, where b^n indicates n repetitions of the symbol b. In other words, G is capable of generating *only* skeletons of the form shown in Fig. 12.26(c) but having arbitrary length. ■



a
b
c

FIGURE 12.26 (a) Object represented by its (pruned) skeleton. (b) Primitives. (c) Structure generated by using a regular string grammar.

TABLE 12.1
Example of semantic information attached to production rules.

Production	Semantic Information
$S \rightarrow aA$	Connections to a are made only at the dot. The direction of a, denoted θ, is given by the direction of the perpendicular bisector of the line joining the end points of the two undotted segments. The line segments are 3 cm each.
$A \rightarrow bA$	Connections to b are made only at the dots. No multiple connections are allowed. The direction of b must be the same as the direction of a. The length of b is 0.25 cm. This production cannot be applied more than 10 times.
$A \rightarrow bB$	The direction of a and b must be the same. Connections must be simple and made only at the dots.
$B \rightarrow c$	The direction of c and a must be the same. Connections must be simple and made only at the dots.

Use of semantics

In the preceding example we assumed that the interconnection between primitives takes place only at the dots shown in Fig. 12.26(b). In more complicated situations the rules of connectivity, as well as information regarding other factors (such as primitive length and direction) and the number of times a production can be applied, must be made explicit. This can be accomplished by using *semantic rules* stored in the *knowledge base* of Fig. 1.23. Basically, the syntax inherent in the production rules establishes the structure of an object, whereas semantics deal with its correctness. For example, in a programming language like C, the statement $A = D/E$ is syntactically correct, but it is semantically correct only if $E \neq 0$.

Suppose that we attach semantic information to the grammar discussed in the preceding example. The information can be attached to the production rules in the form shown in Table 12.1. By using semantic information, we are able to use a few rules of syntax to describe a broad (but limited as desired) class of patterns. For instance, by specifying the direction of θ in Table 12.1, we avoid having to specify primitives for each possible orientation. Similarly, by requiring that all primitives be oriented in the same direction, we eliminate from consideration nonsensical structures that deviate from the basic shapes typified by Fig. 12.26(a).

Automata as string recognizers

So far we have demonstrated that grammars are *generators* of patterns. In the following discussion we consider the problem of recognizing whether a pattern belongs to the language $L(G)$ generated by a grammar G. The basic concepts underlying syntactic recognition may be illustrated by the development of mathematical models of computing machines, called *automata*. Given an input pattern string, an automaton is capable of recognizing whether the pattern belongs to the language with which the automaton is associated. Here, we focus only on *finite automata*, which are the recognizers of languages generated by regular grammars.

A *finite automaton* is defined as the 5-tuple

$$A_f = (Q, \Sigma, \delta, q_0, F) \tag{12.3-7}$$

where Q is a finite, nonempty set of *states*, Σ is a finite input *alphabet*, δ is a *mapping* from $Q \times \Sigma$ (the set of ordered pairs formed from elements of Q and Σ) into the collection of all subsets of Q, q_0 is the *starting state*, and F (a subset of Q) is a set of *final*, or *accepting, states*.

EXAMPLE 12.10: A simple automaton.

■ Consider an automaton given by Eq. (12.3-7), with $Q = \{q_0, q_1, q_2\}$, $\Sigma = \{a, b\}$, $F = \{q_0\}$, and mappings $\delta(q_0, a) = \{q_2\}$, $\delta(q_0, b) = \{q_1\}$, $\delta(q_1, a) = \{q_2\}$, $\delta(q_1, b) = \{q_0\}$, $\delta(q_2, a) = \{q_0\}$, and $\delta(q_2, b) = \{q_1\}$. If, for example, the automaton is in state q_0 and an a is input, its state changes to q_2. Similarly, if a b is input next, the automaton changes to state q_1, and so on. The initial and final states are the same in this case. ■

Figure 12.27 shows a *state diagram* for the automaton just discussed. The state diagram consists of a node for each state and directed arcs showing the possible transitions between states. The final state is shown as a double circle, and each arc is labeled with the symbol that causes the transition between the states joined by that arc. In this case the initial and final states are the same. A string w of terminal symbols is said to be *accepted* or *recognized* by an automaton if, starting in state q_0, the sequence of symbols (encountered as w is scanned from left to right) causes the automaton to be in a final state after the last symbol from w has been scanned. For example, the automaton in Fig. 12.27 recognizes the string $w = abbabb$ but rejects the string $w = aabab$.

There is a one-to-one correspondence between regular grammars and finite automata. That is, a language is recognized by a finite automaton if and only if it is generated by a regular grammar. The design of a syntactic string recognizer based on the concepts discussed so far is a straightforward procedure, consisting of obtaining a finite automaton from a given regular grammar. Let the grammar be denoted $G = (N, \Sigma, P, X_0)$, where $X_0 = S$, and suppose that N is composed of X_0 plus n additional nonterminals $X_1, X_2, \ldots, X_n$. The set Q for the automaton is formed by introducing $n + 2$ states $\{q_0, q_1, \ldots, q_n, q_{n+1}\}$ such that q_i corresponds

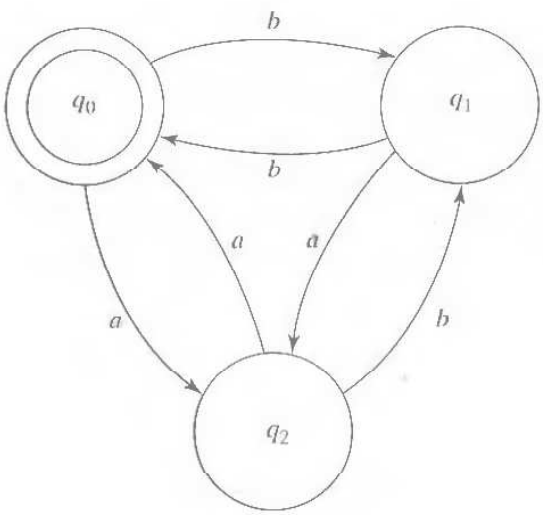


FIGURE 12.27 A finite automaton.

to X_i for $0 \le i \le n$, and q_{n+1} is the final state. The set of input symbols is identical to the set of terminals in G. The mappings in δ are obtained by using two rules based on the productions of G; namely, for each i and j, with $0 \le i \le n, 0 \le j \le n$,

1. If $X_i \rightarrow aX_j$ is in P, then $\delta(q_i, a)$ contains q_j.
2. If $X_i \rightarrow a$ is in P, then $\delta(q_i, a)$ contains q_{n+1}.

Conversely, given a finite automaton, $A_f = (Q, \Sigma, \delta, q_0, F)$, we obtain the corresponding regular grammar, $G = (N, \Sigma, P, X_0)$ by letting N consist of the elements of Q, with the starting symbol X_0 corresponding to q_0, and the productions of G obtained as follows:

1. If q_j is in $\delta(q_i, a)$, there is a production $X_i \rightarrow aX_j$ in P.
2. If a state in F is in $\delta(q_i, a)$ there is a production $X_i \rightarrow a$ in P.

The terminal set, Σ, is the same in both cases.

EXAMPLE 12.11: Finite automaton for recognizing the patterns in Fig. 12.26.

■ The finite automaton for the grammar given in connection with Fig. 12.26 is obtained by writing the productions as $X_0 \rightarrow aX_1, X_1 \rightarrow bX_1, X_1 \rightarrow bX_2$, and $X_2 \rightarrow c$. Then $A_f = (Q, \Sigma, \delta, q_0, F)$, with $Q = \{q_0, q_1, q_2, q_3\}$, $\Sigma = \{a, b, c\}$, $F = \{q_3\}$ and mappings $\delta(q_0, a) = \{q_1\}, \delta(q_1, b) = \{q_1, q_2\}, \delta(q_2, c) = \{q_3\}$. For completeness, we write $\delta(q_0, b) = \delta(q_0, c) = \delta(q_1, a) = \delta(q_1, c) = \delta(q_2, a) = \delta(q_2, b) = \varnothing$, where $\varnothing$ is the null set, indicating that these transitions are not defined for this automaton. ■

12.3.4 Syntactic Recognition of Trees

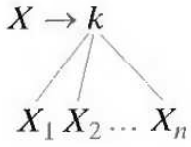
Following a format similar to the preceding discussion for strings, we now expand the discussion to include tree descriptions of patterns. We assume that the image regions or objects of interest have been expressed in the form of trees by using the appropriate primitive elements, as discussed in Section 11.5.

Tree grammars

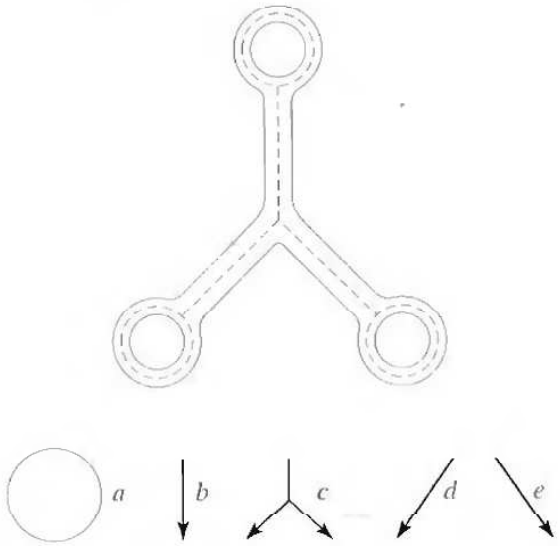
A *tree grammar* is defined as the 5-tuple

$$G = (N, \Sigma, P, r, S) \tag{12.3-8}$$

where, as before, N and Σ are sets of nonterminals and terminals, respectively; S, contained in N, is the start symbol, which in general can be a tree; P is a set of productions of the form $T_i \rightarrow T_j$, where T_i and T_j are trees; and r is a *ranking function* that denotes the number of direct descendants (offspring) of a node whose label is a terminal in the grammar. Of particular relevance to our discussion are *expansive* tree grammars having productions of the form



where $X_1, X_2, \ldots, X_n$ are nonterminals and k is a terminal.



a
b

FIGURE 12.28 (a) An object and (b) primitives used for representing the skeleton by means of a tree grammar.

EXAMPLE 12.12: A simple tree grammar.

■ The skeleton of the structure shown in Fig. 12.28(a) can be generated by using a tree grammar with $N = \{X_1, X_2, X_3, S\}$ and $\Sigma = \{a, b, c, d, e\}$, where the terminals represent the primitives shown in Fig. 12.28(b). Assuming head-to-tail connectivity of the line primitives, and arbitrary connections to the circle along its circumference, the grammar under consideration has productions of the form

$$(1)\ S \rightarrow \begin{array}{c} a \\ | \\ X_1 \end{array} \qquad (2)\ X_1 \rightarrow \begin{array}{c} b \\ | \\ X_1 \end{array} \qquad (3)\ X_1 \rightarrow \begin{array}{c} c \\ X_2 \quad X_3 \end{array}$$

$$(4)\ X_2 \rightarrow \begin{array}{c} d \\ | \\ X_2 \end{array} \qquad (5)\ X_2 \rightarrow e \qquad (6)\ X_3 \rightarrow \begin{array}{c} e \\ | \\ X_3 \end{array} \qquad (7)\ X_3 \rightarrow a$$

The ranking functions in this case are $r(a) = \{0, 1\}$, $r(b) = r(d) = \{1\}$, $r(e) = \{0, 1\}$, and $r(c) = \{2\}$. Restricting application of productions 2, 4, and 6 to the same number of times would generate a structure in which all three legs have the same length. Similarly, requiring application of productions 4 and 6 the same number of times would produce a structure that is symmetrical about its vertical axis. This type of semantic information is similar to the earlier discussion in connection with Table 12.1 and the knowledge base of Fig. 1.23. ■

Tree automata

Whereas a conventional finite automaton scans an input string symbol by symbol from left to right, a tree automaton must begin simultaneously at each node on the frontier (the leaves taken in order from left to right) of an input tree

and proceed along parallel paths toward the root. Specifically, a *frontier-to-root automaton* is defined as

$$A_t = (Q, F, \{f_k \mid k \in \Sigma\}) \tag{12.3-9}$$

where

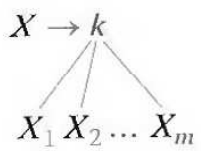
Q is a finite set of states,

F, a subset of Q, is a set of final states, and

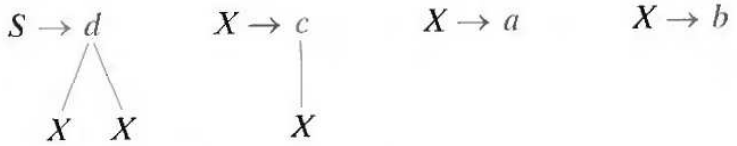
f_k is a relation in $Q^m \times Q$ such that m is a rank of k.

The notation Q^m indicates the Cartesian product of Q with itself m times: $Q^m = Q \times Q \times Q \times \ldots \times Q$. From the definition of the Cartesian product, we know that this expression means the set of all ordered m-tuples with elements from Q. For example, if $m = 3$, than $Q^3 = Q \times Q \times Q = \{x, y, z \mid x \in Q, y \in Q, z \in Q\}$. Recall that a *relation* R from a set A to a set B is a subset of the Cartesian product of A and B; that is, $R \subseteq A \times B$. Thus a relation in $Q^m \times Q$ is simply a subset of the set $Q^m \times Q$.

For an expansive tree grammar, $G = (N, \Sigma, P, r, S)$, we construct the corresponding tree automaton by letting $Q = N$, with $F = \{S\}$ and, for each symbol a in Σ, defining a relation f_k such that $(X_1, X_2, \ldots, X_m, X)$ is in f_k if and only if there is in G a production



For example, consider the tree grammar $G = (N, \Sigma, P, r, S)$, with $N = \{S, X\}$, $\Sigma = \{a, b, c, d\}$, productions



and rankings $r(a) = \{0\}$, $r(b) = \{0\}$, $r(c) = \{1\}$, and $r(d) = \{2\}$. The corresponding tree automaton, $A_t = (Q, F, \{f_k \mid k \in \Sigma\})$, is specified by letting $Q = \{S, X\}$, $F = \{S\}$, and $\{f_k \mid k \in \Sigma\} = \{f_a, f_b, f_c, f_d\}$, where the relations are defined as

$f_a = \{(\varnothing, X)\}$, arising from production $X \to a$

$f_b = \{(\varnothing, X)\}$, arising from production $X \to b$

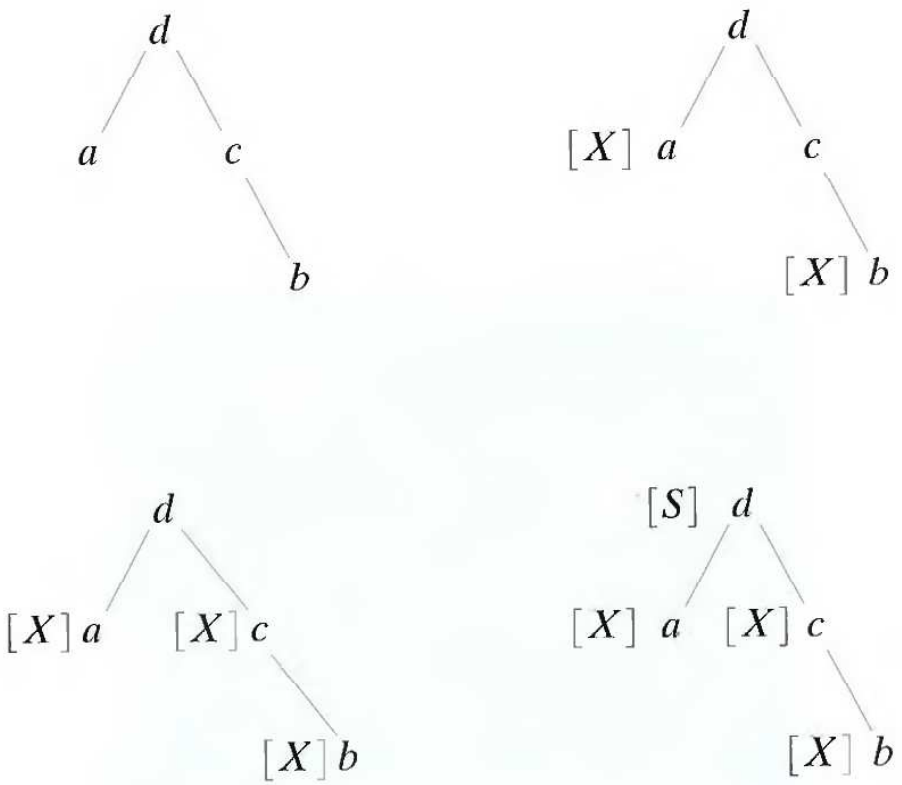
$f_c = \{(X, X)\}$, arising from production $X \to c$

and

$$f_d = \{(X, X, S)\}, \text{ arising from production } S \rightarrow d \begin{array}{c} d \\ \diagup \quad \diagdown \\ X \quad X \end{array}$$

The interpretation of relation f_a is that a node labeled a with no offspring (hence the null symbol $\varnothing$) is assigned state X. The interpretation of f_c is that a node labeled c, with one offspring having state X, is assigned state X. The interpretation of relation f_d is that a node labeled d with two offspring, each having state X, is assigned state S.

In order to see how this tree automaton goes about recognizing a tree generated by the grammar discussed earlier, consider the tree shown in Fig. 12.29(a). Automaton A_t first assigns states to the frontier nodes a and b via relations f_a and f_b, respectively. In this case, according to these two relations, state X is assigned to both leaves, as Fig. 12.29(b) shows. The automaton now moves up one level from the frontier and makes a state assignment to node c on the basis of f_c and the state of this node's offspring. The state assignment based on f_c again is X, as indicated in Fig. 12.29(c). Moving up one more level, the automaton encounters node d and, as its two offspring have been assigned states, relation f_d, which calls for assigning state S to node d, is used. Because this is the last node and the state S is in F, the automaton accepts (recognizes) the tree as being a valid member of the language of the tree grammar given earlier. Figure 12.29(d) shows the final representation of the state sequences followed along the frontier-to-root paths.



a b
c d

FIGURE 12.29 Processing stages of a frontier-to-root tree automaton: (a) Input tree. (b) State assignment to frontier nodes. (c) State assignment to intermediate nodes. (d) State assignment to root node.

EXAMPLE 12.13: Use of tree grammars for recognizing events in bubble chamber images.

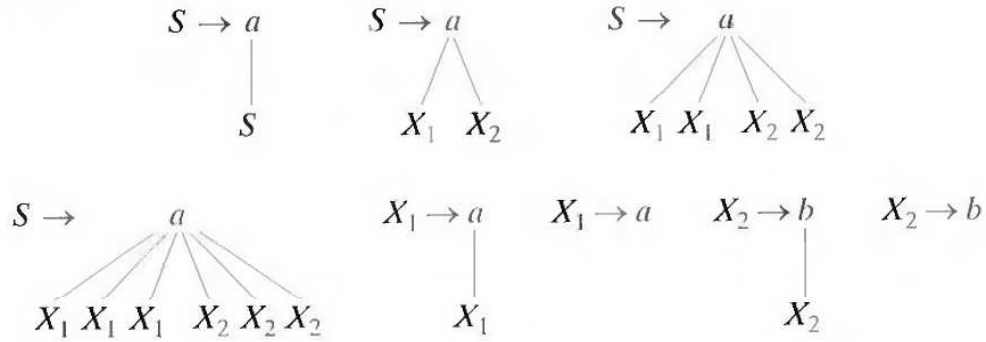
■ Images of bubble chamber events are taken routinely during experiments in high-energy physics in which a beam of particles of known properties is directed onto a target of known nuclei. A typical event consists of tracks of secondary particles emanating from the point of collision, such as the example shown in Fig. 12.30. The incoming tracks are the horizontal parallel lines. Note the natural tree structure of the event near the middle of the photograph.

A typical experiment produces hundreds of thousands of photographs, many of which do not contain events of interest. Examining and categorizing these photographs is tedious and time-consuming for a human interpreter, thus creating a need for automatic event recognition techniques.

A tree grammar $G = (N, \Sigma, P, r, S)$ can be specified that generates trees representing events typical of those found in a hydrogen bubble chamber as a result of incoming positively charged particle streams. In this case, $N = \{S, X_1, X_2\}$, $\Sigma = \{a, b\}$, and the primitives a and b are interpreted as follows:

a: ◠ convex arc
b: ◡ concave arc.

The productions in P are



The rankings are $r(a) = \{0, 1, 2, 4, 6\}$ and $r(b) = \{0, 1\}$. The branching productions represent the number of tracks emanating from a collision, which occur

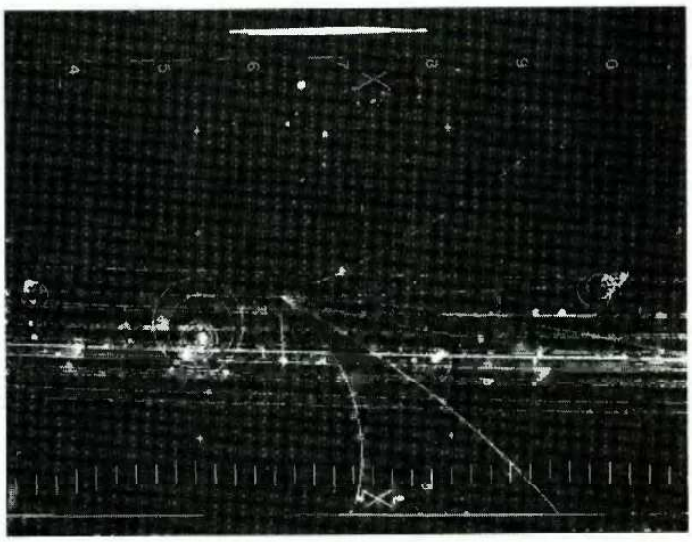
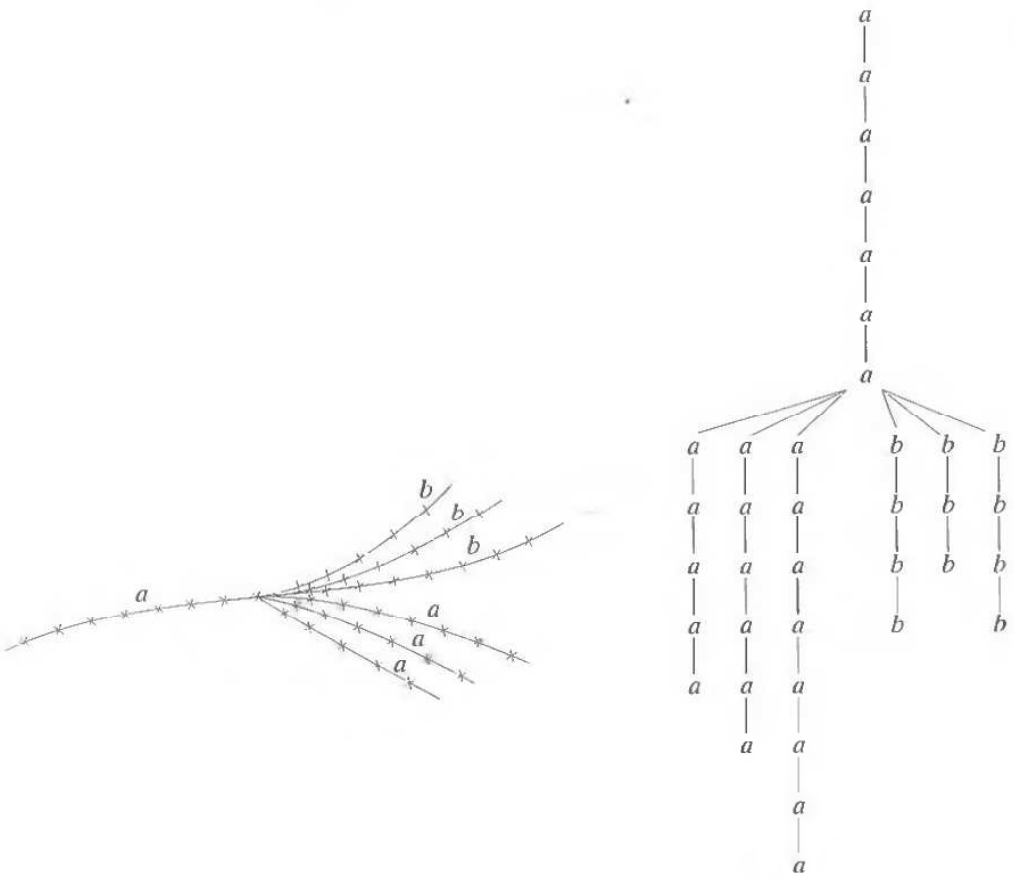


FIGURE 12.30 A bubble chamber photograph. (Fu and Bhargava.)


a
b
b
b
a
a
a

focus is on learning finite automata directly from sample pattern strings. The references at the end of this chapter provide a guide to methods for learning tree grammars and automata, as well as other syntactic recognition approaches.

Suppose that all patterns of a class are generated by an *unknown* grammar G and that a finite set of samples R^+ with the property

$$R^+ \subseteq \{v \mid v \text{ in } L(G)\} \tag{12.3-10}$$

is available. The set R^+, called a *positive sample set*, is simply a set of training patterns from the class associated with grammar G. This sample set is said to be *structurally complete* if each production in G is used to generate at least one element of R^+. We want to learn (synthesize) a finite automaton A_f that will accept the strings of R^+ and possibly some strings that resemble those of R^+.

Based on the definition of a finite automaton and the correspondence between G and A_f, it follows that $R^+ \subseteq \Sigma^*$, where Σ^* is the set of all strings composed of elements from Σ. Let z in Σ^* be a string such that zw is in R^+ for some w in Σ^*. For a positive integer k, we define the k *tail* of z with respect to R^+ as the set $h(z, R^+, k)$, where

$$h(z, R^+, k) = \{w \mid zw \text{ in } R^+, |w| \leq k\}. \tag{12.3-11}$$

In other words, the k tail of z is the set of strings w with the properties (1) zw is in R^+, and (2) the length of w is less than or equal to k.

A procedure for learning an automaton $A_f(R^+, k) = (Q, \Sigma, \delta, q_0, F)$ from a sample set R^+ and a particular value of k consists of letting

$$Q = \{q \mid q = h(z, R^+, k) \text{ for } z \text{ in } \Sigma^*\} \tag{12.3-12}$$

and, for each a in Σ,

$$\delta(q, a) = \{q' \text{ in } Q \mid q' = h(za, R^+, k), \text{with } q = h(z, R^+, k)\}. \tag{12.3-13}$$

In addition, we let

$$q_0 = h(\lambda, R^+, k) \tag{12.3-14}$$

and

$$F = \{q \mid q \text{ in } Q, \lambda \text{ in } q\} \tag{12.3-15}$$

where λ is the empty string (the string with no symbols). We note that the automaton $A_f(R^+, k)$ has as states subsets of the set of all k tails that can be constructed from R^+.

EXAMPLE 12.14: Inferring a finite automaton from sample patterns.

■ Suppose that $R^+ = \{a, ab, abb\}$ and $k = 1$. Then from the preceding discussion,

$$\begin{aligned} z = \lambda, \qquad h(\lambda, R^+, 1) &= \{w \mid \lambda w \text{ in } R^+, |w| \leq 1\} \\ &= \{a\} \\ &= q_0 \\ z = a, \qquad h(a, R^+, 1) &= \{w \mid aw \text{ in } R^+, |w| \leq 1\} \\ &= \{\lambda, b\} \\ &= q_1 \end{aligned}$$

$$z = ab, \quad h(ab, R^+, 1) = \{\lambda, b\} = q_1$$
$$z = abb, \quad h(abb, R^+, 1) = \{\lambda\} = q_2.$$

In this case, other strings z in Σ^* yield strings zw that do not belong to R^+, giving rise to a fourth state, denoted $q_\varnothing$, which corresponds to the condition that h is the null set. The states, therefore, are $q_0 = \{a\}$, $q_1 = \{\lambda, a\}$, $q_2 = \{\lambda\}$, and $q_\varnothing$, which give the set $Q = \{q_0, q_1, q_2, q_\varnothing\}$. Although the states are obtained as sets of symbols (k tails), only the state labels $q_0, q_1, \ldots$ are used in forming the set Q. The next step is to obtain the transition functions. Since $q_0 = h(\lambda, R^+, 1)$, it follows that

$$\delta(q_0, a) = h(\lambda a, R^+, 1) = h(a, R^+, 1) = q_1$$

and

$$\delta(q_0, b) = h(\lambda b, R^+, 1) = h(b, R^+, 1) = q_\varnothing.$$

Similarly, $q_1 = h(a, R^+, 1) = h(ab, R^+, 1)$ and it follows that

$$\delta(q_1, a) = h(aa, R^+, 1) = h(aba, R^+, 1) = q_\varnothing.$$

Also, $\delta(q_1, b) \supseteq h(ab, R^+, 1) = q_1$ and $\delta(q_1, b) \supseteq h(abb, R^+, 1) = q_2$; that is, $\delta(q_1, b) = \{q_1, q_2\}$. Following the procedure just described gives $\delta(q_2, a) = \delta(q_2, b) = \delta(q_\varnothing, a) = \delta(q_\varnothing, b) = q_\varnothing$. The set of final states contains those states that have the empty string λ in their k-tail representation. In this case, $q_1 = \{\lambda, b\}$ and $q_2 = \{\lambda\}$, so $F = \{q_1, q_2\}$.

Based on these results, the inferred automaton is given by

$$A_f(R^+, 1) = (Q, \Sigma, \delta, q_0, F)$$

where $Q = \{q_0, q_1, q_2, q_\varnothing\}$, $\Sigma = \{a, b\}$, $F = \{q_1, q_2\}$, and the transition functions are as given above. Figure 12.32 shows the state diagram. The automaton accepts strings of the form $a, ab, abb, \ldots, ab^n$, which are consistent with the given sample set. ■

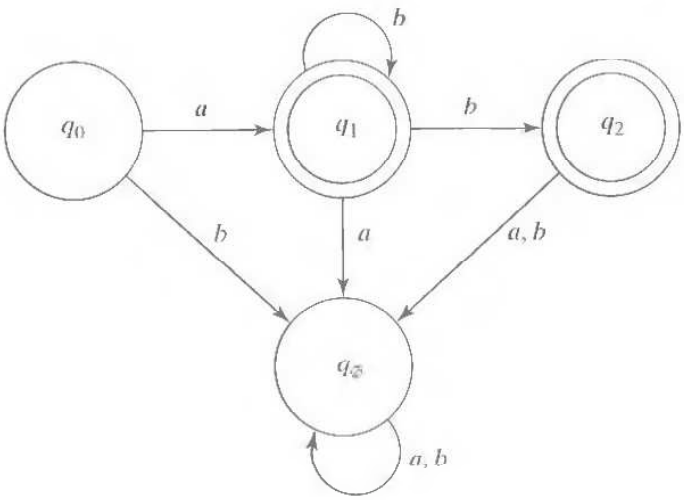


FIGURE 12.32 State diagram for the finite automaton inferred from the sample set $R^+ = \{a, ab, abb\}$.

The preceding example shows that the value of k controls the nature of the resulting automaton. The following properties exemplify the dependence of $A_f(R^+, k)$ on this parameter.

Property 1. $R^+ \subseteq L[A_f(R^+, k)]$ for all $k \geq 0$, where $L[A_f(R^+, k)]$ is the language accepted by $A_f(R^+, k)$.

Property 2. $L[A_f(R^+, k)] = R^+$ if k is equal to, or greater than, the length of the longest string in R^+; $L[A_f(R^+, k)] = \Sigma^*$ if $k = 0$.

Property 3. $L[A_f(R^+, k+1) \subseteq L[A_f(R^+, k)]$.

Property 1 guarantees that $A_f(R^+, k)$ will, as a minimum, accept the strings in the sample set R^+. If k is equal to, or greater than, the length of the longest string in R^+, then by Property 2 the automation will accept *only* the strings in R^+. If $k = 0$, $A_f(R^+, 0)$ will consist of one state $q_0 = \{\lambda\}$, which will act as both the initial and final states. The transition functions will then be of the form $\delta(q_0, a) = q_0$ for a in Σ. Therefore, $L[A_f(R^+, 0)] = \Sigma^*$, and the automaton will accept the empty string λ and all strings composed of symbols from Σ. Finally, Property 3 indicates that the scope of the language accepted by $A_f(R^+, k)$ decreases as k increases.

These three properties allow control of the nature of $A_f(R^+, k)$ simply by varying the parameter k. If $L[A_f(R^+, k)]$ is a guess of the language L_0 from which the sample R^+ was chosen and if k is very small, this guess of L_0 will constitute a liberal inference that may include most or all of the strings in Σ^*. However, if k is equal to the length of the longest string in R^+, the inference will be conservative in the sense that $A_f(R^+, k)$ will accept only the strings contained in R^+. Figure 12.33 shows these concepts graphically.

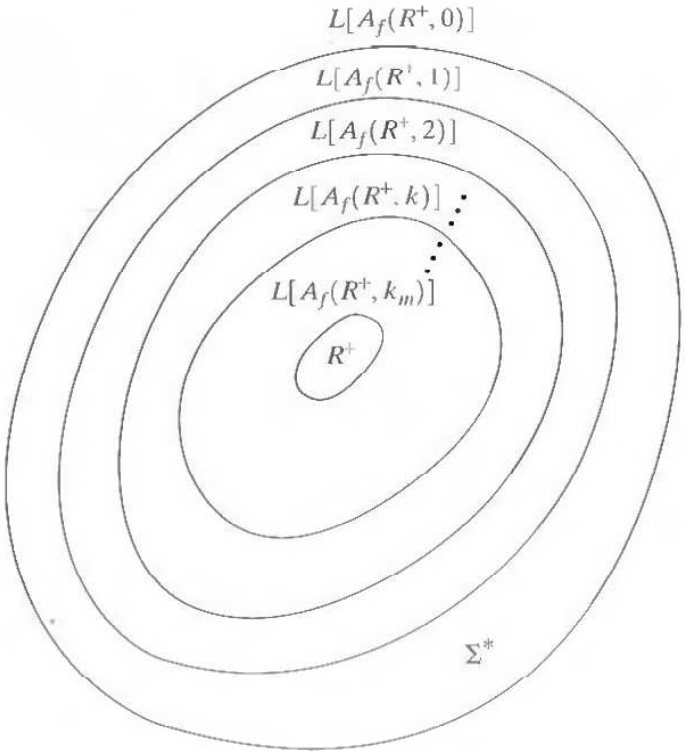


FIGURE 12.33 Relationship between $L[A_f(R^+, k)]$ and k. The value of k_m is such that $k_m \geq$ (length of the longest string in R^+).

EXAMPLE 12.15: Another example of inferring an automaton from a given set of patterns.

■ Consider the set $R^+ = \{caaab, bbaab, caab, bbab, cab, bbb, cb\}$. For $k = 1$, following the same procedure used in the preceding example gives

1. $z = \lambda$, $h(\lambda, R^+, 1) = \{\varnothing\} = q_\varnothing$;
2. $z = c$, $h(z, R^+, 1) = \{b\} = q_1$;
3. $z = ca$, $h(z, R^+, 1) = \{b\} = q_1$;
4. $z = cb$, $h(z, R^+, 1) = \{\lambda\} = q_0$;
5. $z = caa$, $h(z, R^+, 1) = \{b\} = q_1$;
6. $z = cab$, $h(z, R^+, 1) = \{\lambda\} = q_0$;
7. $z = caaa$, $h(z, R^+, 1) = \{b\} = q_1$;
8. $z = caab$, $h(z, R^+, 1) = \{\lambda\} = q_0$;
9. $z = caaab$, $h(z, R^+, 1) = \{\lambda\} = q_0$;
10. $z = b$, $h(z, R^+, 1) = \{\varnothing\} = q_\varnothing$;
11. $z = bb$, $h(z, R^+, 1) = \{b\} = q_1$;
12. $z = bba$, $h(z, R^+, 1) = \{b\} = q_1$;
13. $z = bbb$, $h(z, R^+, 1) = \{\lambda\} = q_0$;
14. $z = bbaa$, $h(z, R^+, 1) = \{b\} = q_1$;
15. $z = bbab$, $h(z, R^+, 1) = \{\lambda\} = q_0$;
16. $z = bbaab$, $h(z, R^+, 1) = \{\lambda\} = q_0$.

The automaton is

$$A_f(R^+, 1) = (Q, \Sigma, \delta, q_0, F)$$

with $Q = \{q_0, q_1, q_\varnothing\}$, $\Sigma = \{a, b, c\}$, $F = \{q_0\}$, and the transitions shown in the state diagram in Fig. 12.34. To be accepted by the automaton, a string must begin with a, b, or c and end with a symbol b. Also, strings with repetitions of a, b, or c are accepted by $A_f(R^+, 1)$. ■

The principal advantage of the preceding method is simplicity of implementation. The synthesis procedure can be simulated in a digital computer with a modest amount of effort. The main disadvantage is deciding on a proper value for k, although this problem is simplified to some degree by the three properties discussed earlier.

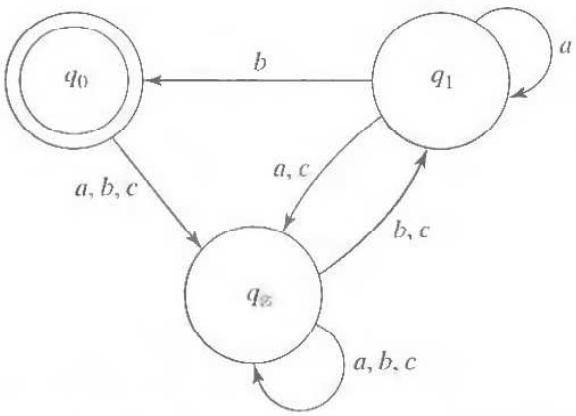


FIGURE 12.34 State diagram for the automaton $A_f(R^+, 1)$ inferred from the sample set $R^+ = \{caaab, bbaab, caab, bbab, cab, bbb, cb\}$.

Summary

Starting with Chapter 9, our treatment of digital image processing began a transition from processes whose outputs are images to processes whose outputs are attributes about images, in the sense defined in Section 1.1. Although the material in the present chapter is introductory in nature, the topics covered are fundamental to understanding the state of the art in object recognition. As mentioned at the beginning of this chapter, recognition of individual objects is a logical place to conclude this book. To go past this point, we need concepts that are beyond the scope we set for our journey back in Section 1.4. Specifically, the next logical step would be the development of image analysis methods whose proper development requires concepts from machine intelligence.

As mentioned in Sections 1.1 and 1.4, machine intelligence and some areas that depend on it, such as scene analysis and computer vision, still are in their relatively early stages of practical development. Solutions of image analysis problems today are characterized by heuristic approaches. While these approaches are indeed varied, most of them share a significant base of techniques that are precisely the methods covered in this book.

Having concluded study of the material in the preceding twelve chapters, the reader is now in the position of being able to understand the principal areas spanning the field of digital image processing, both from a theoretical and practical point of view. Care was taken throughout all discussions to lay a solid foundation upon which further study of this and related fields could be based. Given the task-specific nature of many imaging problems, a clear understanding of basic principles enhances significantly the chances for their successful solution.

References and Further Reading

Background material for Sections 12.1 through 12.2.2 are the books by Duda, Hart, and Stork [2001], and by Tou and Gonzalez [1974]. The survey article by Jain et al. [2000] also is of interest. The book by Principe et al. [1999] presents a good overview of neural networks. A special issue of *IEEE Trans. Image Processing* [1998] is worth comparing with a similar special issue ten years earlier (*IEEE Computer* [1988]). The material presented in Section 12.2.3 is introductory. In fact, the neural network model used in that discussion is one of numerous models proposed over the years. However, the model we discussed is representative and also is used quite extensively in image processing. The example dealing with the recognition of distorted shapes is adapted from Gupta et al. [1990, 1994]. The paper by Gori and Scarselli [1998] discusses the classification power of multilayer neural networks. An approach reported by Ueda [2000] based on using linear combinations of neural networks to achieve minimum classification error is good additional reading in this context.

For additional reading on the material in Section 12.3.1, see Bribiesca and Guzman [1980]. On string matching see Sze and Yang [1981], Oommen and Loke [1997], and Gdalyahu and Weinshall [1999]. References for Sections 12.3.3 and 12.3.4 are Gonzalez and Thomason [1978], Fu [1982], and Bunke and Sanfeliu [1990]. See also Tanaka [1995], Vailaya et al. [1998], Aizaka and Nakamura [1999], and Jonk et al. [1999].

See inside front cover

Detailed solutions to the problems marked with a star can be found in the book web site. The site also contains suggested projects based on the material in this chapter.

Problems

12.1 **(a)** Compute the decision functions of a minimum distance classifier for the patterns shown in Fig. 12.1. You may obtain the required mean vectors by (careful) inspection.

(b) Sketch the decision surfaces implemented by the decision functions in (a).

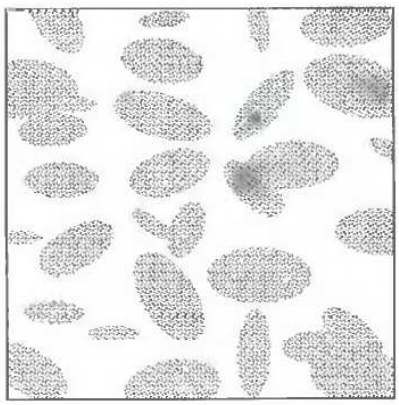
★**12.2** Show that Eqs. (12.2-4) and (12.2-5) perform the same function in terms of pattern classification.

12.3 Show that the surface given by Eq. (12.2-6) is the perpendicular bisector of the line joining the n-dimensional points $\mathbf{m}_i$ and $\mathbf{m}_j$.

★**12.4** Show how the minimum distance classifier discussed in connection with Fig. 12.7 could be implemented by using W resistor banks (W is the number of classes), a summing junction at each bank (for summing currents), and a maximum selector capable of selecting the maximum of W inputs, where the inputs are currents.

12.5 Show that the correlation coefficient of Eq. (12.2-8) has values in the range $[-1, 1]$. *Hint:* Express $\gamma(x, y)$ in vector form.

★**12.6** An experiment produces binary images of blobs that are nearly elliptical in shape (see the following figure). The blobs are of three sizes, with the average values of the principal axes of the ellipses being (1.3, 0.7), (1.0, 0.5), and (0.75, 0.25). The dimensions of these axes vary +10% about their average values. Develop an image processing system capable of rejecting incomplete or overlapping ellipses and then classifying the remaining single ellipses into one of the three size classes given. Show your solution in block diagram form, giving specific details regarding the operation of each block. Solve the classification problem using a minimum distance classifier, indicating clearly how you would go about obtaining training samples and how you would use these samples to train the classifier.



12.7 The following pattern classes have Gaussian probability density functions: ω_1: $\{(0,0)^T, (2,0)^T, (2,2)^T, (0,2)^T\}$ and ω_2: $\{(4,4)^T, (6,4)^T, (6,6)^T, (4,6)^T\}$.

(a) Assume that $P(\omega_1) = P(\omega_2) = \frac{1}{2}$ and obtain the equation of the Bayes decision boundary between these two classes.

(b) Sketch the boundary.

★**12.8** Repeat Problem 12.7, but use the following pattern classes: ω_1: $\{(-1,0)^T, (0,-1)^T, (1,0)^T, (0,1)^T\}$ and ω_2: $\{(-2,0)^T, (0,-2)^T, (2,0)^T, (0,2)^T\}$. Observe that these classes are not linearly separable.

12.9 Repeat Problem 12.6, but use a Bayes classifier (assume Gaussian densities). Indicate clearly how you would go about obtaining training samples and how you would use these samples to train the classifier.

★**12.10** The Bayes decision functions $d_j(\mathbf{x}) = p(\mathbf{x}/\omega_j)P(\omega_j)$, $j = 1, 2, \ldots, W$, were derived using a 0-1 loss function. Prove that these decision functions minimize the probability of error. (*Hint:* The probability of error $p(e)$ is $1 - p(c)$, where $p(c)$ is the probability of being correct. For a pattern vector $\mathbf{x}$ belonging to class ω_i,

$p(c/\mathbf{x}) = p(\omega_i/\mathbf{x})$. Find $p(c)$ and show that $p(c)$ is maximum [$p(e)$ is minimum] when $p(\mathbf{x}/\omega_i)P(\omega_i)$ is maximum.)

12.11 **(a)** Apply the perceptron algorithm to the following pattern classes: ω_1: $\{(0, 0, 0)^T, (1, 0, 0)^T, (1, 0, 1)^T, (1, 1, 0)^T\}$ and ω_2: $\{(0, 0, 1)^T, (0, 1, 1)^T, (0, 1, 0)^T, (1, 1, 1)^T\}$. Let $c = 1$, and $\mathbf{w}(1) = (-1, -2, -2, 0)^T$.

(b) Sketch the decision surface obtained in (a). Show the pattern classes and indicate the positive side of the surface.

★12.12 The perceptron algorithm given in Eqs. (12.2-34) through (12.2-36) can be expressed in a more concise form by multiplying the patterns of class ω_2 by -1, in which case the correction steps in the algorithm become $\mathbf{w}(k+1) = \mathbf{w}(k)$, if $\mathbf{w}^T(k)\mathbf{y}(k) > 0$, and $\mathbf{w}(k+1) = \mathbf{w}(k) + c\mathbf{y}(k)$ otherwise. This is one of several perceptron algorithm formulations that can be derived by starting from the general gradient descent equation

$$\mathbf{w}(k+1) = \mathbf{w}(k) - c\left[\frac{\partial J(\mathbf{w}, \mathbf{y})}{\partial \mathbf{w}}\right]_{\mathbf{w}=\mathbf{w}(k)}$$

where $c > 0$, $J(\mathbf{w}, \mathbf{y})$ is a criterion function, and the partial derivative is evaluated at $\mathbf{w} = \mathbf{w}(k)$. Show that the perceptron algorithm formulation is obtainable from this general gradient descent procedure by using the criterion function $J(\mathbf{w}, \mathbf{y}) = \frac{1}{2}(|\mathbf{w}^T\mathbf{y}| - \mathbf{w}^T\mathbf{y})$, where $|\arg|$ is the absolute value of the argument.

(*Note:* The partial derivative of $\mathbf{w}^T\mathbf{y}$ with respect to $\mathbf{w}$ equals $\mathbf{y}$.)

12.13 Prove that the perceptron training algorithm given in Eqs. (12.2-34) through (12.2-36) converges in a finite number of steps if the training pattern sets are linearly separable. [*Hint:* Multiply the patterns of class ω_2 by -1 and consider a nonnegative threshold, T, so that the perceptron training algorithm (with $c = 1$) is expressed as $\mathbf{w}(k+1) = \mathbf{w}(k)$, if $\mathbf{w}^T(k)\mathbf{y}(k) > T$, and $\mathbf{w}(k+1) = \mathbf{w}(k) + \mathbf{y}(k)$ otherwise. You may need to use the Cauchy-Schwartz inequality: $\|\mathbf{a}\|^2\|\mathbf{b}\|^2 \geq (\mathbf{a}^T\mathbf{b})^2$.]

★12.14 Specify the structure and weights of a neural network capable of performing *exactly* the same function as a minimum distance classifier for two pattern classes in n-dimensional space.

12.15 Specify the structure and weights of a neural network capable of performing *exactly* the same function as a Bayes classifier for two pattern classes in n-dimensional space. The classes are Gaussian with different means but equal covariance matrices.

★12.16 **(a)** Under what conditions are the neural networks in Problems 12.14 and 12.15 identical?

(b) Would the generalized delta rule for multilayer feedforward neural networks developed in Section 12.2.3 yield the particular neural network in (a) if trained with a sufficiently large number of samples?

12.17 Two pattern classes in two dimensions are distributed in such a way that the patterns of class ω_1 lie randomly along a circle of radius r_1. Similarly, the patterns of class ω_2 lie randomly along a circle of radius r_2, where $r_2 = 2r_1$. Specify the structure of a neural network with the minimum number of layers and nodes needed to classify properly the patterns of these two classes.

★**12.18** Repeat Problem 12.6, but use a neural network. Indicate clearly how you would go about obtaining training samples and how you would use these samples to train the classifier. Select the simplest possible neural network that, in your opinion, is capable of solving the problem.

12.19 Show that the expression $h_j'(I_j) = O_j(1 - O_j)$ given in Eq. (12.2-71), where $h_j'(I_j) = \partial h_j(I_j)/\partial I_j$, follows from Eq. (12.2-50) with $\theta_0 = 1$.

★**12.20** Show that the distance measure $D(A, B)$ of Eq. (12.3-2) satisfies the properties given in Eq. (12.3-3).

12.21 Show that $\beta = \max(|a|, |b|) - \alpha$ in Eq. (12.3-4) is 0 if and only if a and b are identical strings.

12.22★(a) Specify a finite automaton capable of recognizing pattern strings of the form $ab^n a$.

(b) Obtain the corresponding regular grammar from your solution in (a). (Do not solve by inspection.)

12.23 Give an expansive tree grammar for generating images consisting of alternating 1's and 0's in both spatial directions (in a checkerboard pattern). Assume that the top left element is a 1 and that all images terminate with a 1 as the bottom left element.

★**12.24** Use the learning procedure specified in Eqs. (12.3-12) through (12.3-15) to learn a finite automaton capable of recognizing strings of the form $ab^n a$, with $n > 0$. Start with the sample set $\{aba, abba, abbba\}$. If this set is insufficient for the algorithm to discover the iterative regularity of symbol b, add more sample strings until it can.

12.25 Show that the tree automaton given in connection with Fig. 12.30 accepts the tree given in Fig. 12.31(b).

12.26 A certain factory mass produces small American flags for sporting events. The quality assurance team has observed that, during periods of peak production, some printing machines have a tendency to drop (randomly) between one and three stars and one or two entire stripes. Aside from these errors, the flags are perfect in every other way. Although the flags containing errors represent a small percentage of total production, the plant manager decides to solve the problem. After much investigation, he concludes that automatic inspection using image processing techniques is the most economical way to handle the problem. The basic specifications are as follows: The flags are approximately 7.5 cm by 12.5 cm in size. They move lengthwise down the production line (individually, but with a $\pm 15°$ variation in orientation) at approximately 50 cm/s, with a separation between flags of approximately 5 cm. In all cases, "approximately" means $\pm 5\%$. The plant manager hires you to design an image processing system for each production line. You are told that cost and simplicity are important parameters in determining the viability of your approach. Design a complete system based on the model of Fig. 1.23. Document your solution (including assumptions and specifications) in a brief (but clear) written report addressed to the plant manager.

Bibliography

Abidi, M. A., and Gonzalez, R. C. (eds.) [1992]. *Data Fusion in Robotics and Machine Intelligence*, Academic Press, New York.

Abidi, M. A., Eason, R. O., and Gonzalez, R. C. [1991]. "Autonomous Robotics Inspection and Manipulation Using Multisensor Feedback." *IEEE Computer*, vol. 24, no. 4, pp. 17–31.

Abramson, N. [1963]. *Information Theory and Coding*, McGraw-Hill, New York.

Adiv, G. [1985]. "Determining Three-Dimensional Motion and Structure from Optical Flow Generated by Several Moving Objects." *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-7, no. 4, pp. 384–401.

Aggarwal, J. K., and Badler, N. I. (eds.) [1980]. "Motion and Time-Varying Imagery." *IEEE Trans. Pattern Anal. Mach. Intell.*, Special Issue, vol. PAMI-2, no. 6, pp. 493–588.

Aguado, A. S., Nixon, M. S., and Montiel, M. M. [1998], "Parameterizing Arbitrary Shapes via Fourier Descriptors for Evidence-Gathering Extraction," *Computer Vision and Image Understanding*, vol. 69, no. 2, pp. 202–221.

Ahmed, N., Natarajan, T., and Rao, K. R. [1974]. "Discrete Cosine Transforms." *IEEE Trans. Comp.*, vol. C-23, pp. 90–93.

Ahmed, N., and Rao, K. R. [1975]. *Orthogonal Transforms for Digital Signal Processing*, Springer-Verlag, New York.

Aizaka, K., and Nakamura, A. [1999]. "Parsing of Two-Dimensional Images Represented by Quadtree Adjoining Grammars," *Pattern Recog.*, vol. 32, no. 2, pp. 277–294.

Alliney, S. [1993]. "Digital Analysis of Rotated Images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, no. 5, pp. 499–504.

Ando, S. [2000]. "Consistent Gradient Operators," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, no. 3, pp. 252–265.

Andrews, H. C. [1970]. *Computer Techniques in Image Processing*, Academic Press, New York.

Andrews, H. C., and Hunt, B. R. [1977]. *Digital Image Restoration*, Prentice Hall, Englewood Cliffs, N.J.

Anelli, G., Broggi, A., and Destri, G. [1998]. "Decomposition of Arbitrarily-Shaped Morphological Structuring Elements Using Genetic Algorithms," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, no. 2, pp. 217–224.

Ang, P. H., Ruetz, P. A., and Auld, D. [1991]. "Video Compression Makes Big Gains." *IEEE Spectrum*, vol. 28, no. 10, pp. 16–19.

Antonini, M., Barlaud, M., Mathieu, P., and Daubechies, I. [1992]. "Image Coding Using Wavelet Transform." *IEEE Trans. Image Processing*, vol. 1, no. 2, pp. 205–220.

Atchison, D. A., and Smith, G. [2000]. *Optics of the Human Eye*, Butterworth-Heinemann, Boston, Mass.

Baccar, M., Gee, L. A., Abidi, M. A., and Gonzalez, R. C. [1996]. "Segmentation of Range Images Via Data Fusion and Morphological Watersheds," *Pattern Recog.*, vol. 29, no. 10, pp. 1671–1685.

Bajcsy, R., and Lieberman, L. [1976]. "Texture Gradient as a Depth Cue." *Comput. Graph. Image Proc.*, vol. 5, no. 1, pp. 52–67.

Bakir, T., and Reeves, J. S. [2000]. "A Filter Design Method for Minimizing Ringing in a Region of Interest in MR Spectroscopic Images," *IEEE Trans. Medical Imaging*, vol. 19, no. 6, pp. 585–600.

Ballard, D. H. [1981]. "Generalizing the Hough Transform to Detect Arbitrary Shapes." *Pattern Recognition*, vol. 13, no. 2, pp. 111–122.

Ballard, D. H., and Brown, C. M. [1982]. *Computer Vision*, Prentice Hall, Englewood Cliffs, N.J.

Banham, M. R., Galatsanos, H. L., Gonzalez, H. L., and Katsaggelos, A. K. [1994]. "Multichannel Restoration of Single Channel Images Using a Wavelet-Based Subband Decomposition." *IEEE Trans. Image Processing*, vol. 3, no. 6, pp. 821–833.

Banham, M. R., and Katsaggelos, A. K. [1996]. "Spatially Adaptive Wavelet-Based Multiscale Image Restoration." *IEEE Trans. Image Processing*, vol. 5, no. 5, pp. 619–634.

Basart, J. P., and Gonzalez, R. C. [1992]. "Binary Morphology," in *Advances in Image Analysis*, Y. Mahdavieh and R. C. Gonzalez (eds.), SPIE Press, Bellingham, Wash., pp. 277–305.

Basart, J. P., Chacklackal, M. S., and Gonzalez, R. C. [1992]. "Introduction to Gray-Scale Morphology," in *Advances in Image Analysis*, Y. Mahdavieh and R. C. Gonzalez (eds.), SPIE Press, Bellingham, Wash., pp. 306–354.

Bates, R. H. T., and McDonnell, M. J. [1986]. *Image Restoration and Reconstruction*. Oxford University Press, New York.

Battle, G. [1987]. "A Block Spin Construction of Ondelettes. Part I: Lemarié Functions." *Commun. Math. Phys.*, vol. 110, pp. 601–615.

Battle, G. [1988]. "A Block Spin Construction of Ondelettes. Part II: the QFT Connection." *Commun. Math. Phys.*, vol. 114, pp. 93–102.

Baumert, L. D., Golomb, S. W., and Hall, M., Jr. [1962]. "Discovery of a Hadamard Matrix of Order 92." *Bull. Am. Math. Soc.*, vol. 68, pp. 237–238.

Baxes, G. A. [1994]. *Digital Image Processing: Principles and Applications*, John Wiley & Sons, New York.

Baylon, D. M., and Lim, J. S. [1990]. "Transform/Subband Analysis and Synthesis of Signals." *Tech. Report*, MIT Research Laboratory of Electronics. Cambridge. Mass.

Bell, E. T. [1965]. *Men of Mathematics*, Simon & Schuster, New York.

Bengtsson, A., and Eklundh, J. O. [1991]. "Shape Representation by Multiscale Contour Approximation." *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, no. 1, pp. 85–93.

Benson, K. B. [1985]. *Television Engineering Handbook*, McGraw-Hill, New York.

Berger, T. [1971]. *Rate Distortion Theory*, Prentice Hall, Englewood Cliffs, N.J.

Beucher, S. [1990]. Doctoral Thesis, Centre de Morphologie Mathématique, École des Mines de Paris, France. (The core of this material is contained in the following paper.)

Beucher, S., and Meyer, F. [1992]. "The Morphological Approach of Segmentation: The Watershed Transformation," in *Mathematical Morphology in Image Processing*, E. Dougherty (ed.), Marcel Dekker, New York.

Bhaskaran, V., and Konstantinos, K. [1997]. *Image and Video Compression Standards: Algorithms and Architectures*. Kluwer, Boston, Mass.

Bhatt, B., Birks, D., Hermreck, D. [1997]. "Digital Television: Making It Work." *IEEE Spectrum*, vol. 34, no. 10, pp. 19–28.

Biberman, L. M. [1973]. "Image Quality." In *Perception of Displayed Information*. Biberman, L. M. (ed.), Plenum Press, New York.

Bichsel, M. [1998]. "Analyzing a Scene's Picture Set under Varying Lightning," *Computer Vision and Image Understanding*, vol. 71, no. 3, pp. 271–280.

Bieniek, A., and Moga, A. [2000]. "An Efficient Watershed Algorithm Based on Connected Components," *Pattern Recog.*, vol. 33, no. 6, pp. 907–916.

Bisignani, W. T., Richards, G. P., and Whelan, J. W. [1966]. "The Improved Grey Scale and Coarse-Fine PCM Systems: Two New Digital TV Bandwidth Reduction Techniques." *Proc. IEEE*, vol. 54, no. 3, pp. 376–390.

Blahut, R. E. [1987]. *Principles and Practice of Information Theory*, Addison-Wesley, Reading, Mass.

Bleau, A. and Leon, L. J. [2000]. "Watershed-Based Segmentation and Region Merging," *Computer Vision and Image Understanding*, vol. 77, no. 3, pp. 317–370.

Blouke, M. M., Sampat, N., and Canosa, J. [2001]. *Sensors and Camera Systems for Scientific, Industrial, and Digital Photography Applications-II*, ISBN 0–8194–3583–X, SPIE Press, Bellingham, Wash.

Blum, H. [1967]. "A Transformation for Extracting New Descriptors of Shape." In *Models for the Perception of Speech and Visual Form*, Wathen-Dunn, W. (ed.), MIT Press, Cambridge, Mass.

Blume, H., and Fand, A. [1989]. "Reversible and Irreversible Image Data Compression Using the S-Transform and Lempel-Ziv Coding." *Proc. SPIE Medical Imaging III: Image Capture and Display*, vol. 1091, pp. 2–18.

Boie, R. A., and Cox, I. J. [1992]. "An Analysis of Camera Noise," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, no. 6, pp. 671–674.

Born, M., and Wolf, E. [1999]. *Principles of Optics: Electromagnetic Theory of Propagation, Interference and Diffraction of Light*, 7th ed., Cambridge University Press, Cambride, UK.

Boulgouris, N. V., Tzovaras, D., and Strintzis, M. G. [2001]. "Lossless Image Compression Based on Optimal Prediction, Adaptive Lifting, and Conditional Arithmetic Coding," *IEEE Trans. Image Processing*, vol. 10, no. 1, pp. 1–14.

Bouman, C., and Liu, B. [1991]. "Multiple Resolution Segmentation of Textured Images." *IEEE Trans. Pattern. Anal. Machine Intell.*, vol. 13, no. 2, pp. 99–113.

Boyd, J. E. and Meloche, J. [1998]. "Binary Restoration of Thin Objects in Multidimensional Imagery," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, no. 6, pp. 647–651.

Bracewell, R. N. [1995]. *Two-Dimensional Imaging*, Prentice Hall, Upper Saddle River, N.J.

Bracewell, R. N. [2000]. *The Fourier Transform and its Applications*, 3rd ed. McGraw-Hill, New York.

Bribiesca, E. [1981]. "Arithmetic Operations Among Shapes Using Shape Numbers." *Pattern Recog.*, vol. 13, no. 2, pp. 123–138.

Bribiesca, E. [1999]. "A New Chain Code," *Pattern Recog.*, vol. 32, no. 2, pp. 235–251.

Bribiesca, E. [2000]. "A Chain Code for Representing 3–D Curves," *Pattern Recog.*, vol. 33, no. 5, pp. 755–765.

Bribiesca, E., and Guzman, A. [1980]. "How to Describe Pure Form and How to Measure Differences in Shape Using Shape Numbers." *Pattern Recog.*, vol. 12, no. 2, pp. 101–112.

Brigham, E. O. [1988]. *The Fast Fourier Transform and its Applications*, Prentice Hall, Upper Saddle River, N.J.

Brinkman, B. H., Manduca, A., and Robb, R. A. [1998]. "Optimized Homomorphic Unsharp Masking for MR Grayscale Inhomogeneity Correction," *IEEE Trans. Medical Imaging*, vol. 17, no. 2, pp. 161–171.

Brummer, M. E. [1991]. "Hough Transform Detection of the Longitudinal Fissure in Tomographic Head Images." *IEEE Trans. Biomed. Images*, vol. 10, no. 1, pp. 74–83.

Brzakovic, D., Patton, R., and Wang, R. [1991]. "Rule-Based Multi-template Edge Detection." *Comput. Vision, Graphics, Image Proc: Graphical Models and Image Proc.*, vol. 53, no. 3, pp. 258–268.

Bunke, H. and Sanfeliu, A. (eds.) [1990]. *Syntactic and Structural Pattern Recognition: Theory and Applications*, World Scientific, Teaneck, N.J.

Burrus, C. S., Gopinath, R. A., and Guo, H. [1998]. *Introduction to Wavelets and Wavelet Transforms*, Prentice Hall, Upper Saddle River, N J., pp. 250–251.

Burt, P. J., and Adelson, E. H. [1983]. "The Laplacian Pyramid as a Compact Image Code." *IEEE Trans. Commun.*, vol. COM-31, no. 4, pp. 532–540.

Campbell, J. D. [1969]. "Edge Structure and the Representation of Pictures." Ph.D. dissertation, Dept. of Elec. Eng., University of Missouri, Columbia.

Candy, J. C., Franke, M. A., Haskell, B. G., and Mounts, F. W. [1971]. "Transmitting Television as Clusters of Frame-to-Frame Differences." *Bell Sys. Tech. J.*, vol. 50, pp. 1889–1919.

Cannon, T. M. [1974]. "Digital Image Deblurring by Non-Linear Homomorphic Filtering." Ph.D. thesis, University of Utah.

Canny, J. [1986]. "A Computational Approach for Edge Detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 8, no. 6, pp. 679–698.

Carey, W. K., Chuang, D. B., and Hamami, S. S. [1999]. "Regularity-Preserving Image Interpolation," *IEEE Trans. Image Processing*, vol. 8, no. 9, pp. 1293–1299.

Caselles, V., Lisani, J.-L., Morel, J.-M., and Sapiro, G. [1999]. "Shape Preserving Local Histogram Modification," *IEEE Trans. Image Processing*, vol. 8, no. 2, pp. 220–230.

Castleman, K. R. [1996]. *Digital Image Processing*, 2nd ed. Prentice Hall, Upper Saddle River, N.J.

Centeno, J. A. S., and Haertel, V. [1997]. "An Adaptive Image Enhancement Algorithm," *Pattern Recog.*, vol. 30, no. 7 pp. 1183–1189.

Chan, R. C., Karl, W. C., and Lees, R. S. [2000]. "A New Model-Based Technique for Enhanced Small-Vessell Measurements in X-Ray Cine-Angiograms," *IEEE Trans. Medical Imaging*, vol. 19, no. 3, pp. 243–255.

Chang, S. G., Yu, B., and Vetterli, M. [2000]. "Spatially Adaptive Wavelet Thresholding with Context Modeling for Image Denoising." *IEEE Trans. Image Processing*, vol. 9, no. 9, pp. 1522–1531.

Chang, S. K. [1989]. *Principles of Pictorial Information Systems Design*, Prentice Hall, Englewood Cliffs, N.J.

Chang, T., and Kuo, C.-C. J. [1993]. "Texture Analysis and Classification with Tree-Structures Wavelet Transforms." *IEEE Trans. Image Processing*, vol. 2, no. 4, pp. 429–441.

Chaudhuri, B. B. [1983]. "A Note on Fast Algorithms for Spatial Domain Techniques in Image Processing." *IEEE Trans. Syst. Man Cyb.*, vol. SMC-13, no. 6, pp. 1166–1169.

Chen, M. C., and Wilson, A. N. [2000]. "Motion-Vector Optimization of Control Grid Interpolation and Overlapped Block Motion Compensation Using Iterated Dynamic Programming," *IEEE Trans. Image Processing.*, vol. 9, no. 7, pp. 1145–1157.

Chen, Y.-S., and Yu, Y.-T. [1996]. "Thinning Approach for Noisy Digital Patterns," *Pattern Recog.*, vol. 29, no. 11, pp. 1847–1862.

Cheng, H. D., and Huijuan Xu, H. [2000]. "A Novel Fuzzy Logic Approach to Contrast Enhancement," *Pattern Recog.*, vol. 33, no. 5, pp. 809–819.

Cheriet, M., Said, J. N., and Suen, C. Y. [1998]. "A Recursive Thresholding Technique for Image Segmentation," *IEEE Trans. Image Processing*, vol. 7, no. 6, pp. 918–921.

Cheung, J., Ferris, D., and Kurz, L. [1997]. "On Classification of Multispectral Infrared Image Data," *IEEE Trans. Image Processing*, vol. 6, no. 10, pp. 1456–1464.

Cheung, K. K. T., and Teoh, E. K. [1999]. "Symmetry Detection by Generalized Complex (GC) Moments: A Closed-Form Solution," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 21, no. 5, pp. 466–476.

Chow, C. K., and Kaneko, T. [1972]. "Automatic Boundary Detection of the Left Ventricle from Cineangiograms." *Comp., and Biomed. Res.*, vol. 5, pp. 388–410.

Chu, C.-C., and Aggarwal, J. K [1993]. "The Integration of Image Segmentation Maps Using Regions and Edge Information," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, no. 12, pp. 1241–1252.

CIE [1978]. *Uniform Color Spaces—Color Difference Equations—Psychometric Color Terms*, Commission Internationale de L'Eclairage, Publication No. 15, Supplement No. 2, Paris.

Clark, J. J. [1989]. "Authenticating Edges Produced by Zero-Crossing Algorithms." *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, no. 8, pp. 830–831.

Clarke, R. J. [1985]. *Transform Coding of Images*, Academic Press, New York.

Cochran, W. T., Cooley, J. W., et al. [1967]. "What Is the Fast Fourier Transform?" *IEEE Trans. Audio and Electroacoustics*, vol. AU-15, no. 2, pp. 45–55.

Cohen, A., and Daubechies, I. [1992]. *A Stability Criterion for Biorthogonal Wavelet Bases and Their Related Subband Coding Schemes*, Technical report, AT&T Bell Laboratories.

Cohen, A., Daubechies, I., and Feauveau, J.-C. [1992]. "Biorthogonal Bases of Compactly Supported Wavelets." *Commun. Pure and Appl. Math.*, vol. 45, pp. 485–560.

Coifman, R. R., and Wickerhauser, M. V. [1992]. "Entropy-Based Algorithms for Best Basis Selection." *IEEE Tran. Information Theory*, vol. 38, no. 2, pp. 713–718.

Cooley, J. W., Lewis, P. A. W., and Welch, P. D. [1967a]. "Historical Notes on the Fast Fourier Transform." *IEEE Trans. Audio and Electroacoustics*, vol. AU-15, no. 2, pp. 76–79.

Cooley, J. W., Lewis, P. A. W., and Welch, P. D. [1967b]. "Application of the Fast Fourier Transform to Computation of Fourier Integrals." *IEEE Trans. Audio and Electroacoustics*, vol. AU-15, no. 2, pp. 79–84.

Cooley, J. W., Lewis, P. A. W., and Welch, P. D. [1969]. "The Fast Fourier Transform and Its Applications." *IEEE Trans. Educ.*, vol. E-12, no. 1, pp. 27–34.

Cooley, J. W., and Tukey, J. W. [1965]. "An Algorithm for the Machine Calculation of Complex Fourier Series." *Math. of Comput.*, vol. 19, pp. 297–301.

Cornsweet, T. N. [1970]. *Visual Perception*, Academic Press, New York.

Cortelazzo, G. M., Lucchese, L., and Monti, C. M. [1999]. "Frequency Domain Analysis of General Planar Rigid Motion with Finite Duration," *J. Opt. Soc. Amer.-A. Optics, Image Science, and Vision*, vol. 16, no. 6, pp. 1238–1253.

Cowart, A. E., Snyder, W. E., and Ruedger, W. H. [1983]. "The Detection of Unresolved Targets Using the Hough Transform." *Comput. Vision Graph Image Proc.*, vol. 21, pp. 222–238.

Croisier, A., Esteban, D., and Galand, C. [1976]. "Perfect Channel Splitting by Use of Interpolation/Decimation/Tree Decomposition Techniques." *Int. Conf. On Inform. Sciences and Systems*, Patras, Greece, pp. 443–446.

Cumani, A., Guiducci, A., and Grattoni, P. [1991]. "Image Description of Dynamic Scenes," *Pattern Recog.*, vol. 24, no. 7, pp. 661–674.

Cutrona, L. J., and Hall, W. D. [1968]. "Some Considerations in Post-Facto Blur Removal." In *Evaluation of Motion-Degraded Images*, NASA Publ. SP-193, pp. 139–148.

Danielson, G. C., and Lanczos, C. [1942]. "Some Improvements in Practical Fourier Analysis and Their Application to X-Ray Scattering from Liquids." *J. Franklin Institute*, vol. 233, pp. 365–380, 435–452.

Daubechies, I. [1988]. "Orthonormal Bases of Compactly Supported Wavelets." *Commun. On Pure and Appl. Math.*, vol. 41, pp. 909–996.

Daubechies, I. [1990]. "The Wavelet Transform, Time-Frequency Localization and Signal Analysis." *IEEE Transactions on Information Theory*, vol. 36, no. 5, pp. 961–1005.

Daubechies, I. [1992]. *Ten Lectures on Wavelets*, Society for Industrial and Applied Mathematics, Philadelphia, Pa.

Daubechies, I. [1993]. "Orthonormal Bases of Compactly Supported Wavelets II, Variations on a Theme." *SIAM J. Mathematical Analysis*, vol. 24, no. 2, pp. 499–519.

Daubechies, I. [1996]. "Where Do We Go from Here?—A Personal Point of View." *Proc. IEEE*, vol. 84, no. 4, pp. 510–513.

Daul, C., Graebling, P., and Hirsch, E. [1998]. "From the Hough Transform to a New Approach for the Detection of and Approximation of Elliptical Arcs," *Computer Vision and Image Understanding*, vol. 72, no. 3, pp. 215–236.

Davis, L. S. [1982]. "Hierarchical Generalized Hough Transforms and Line-Segment Based Generalized Hough Transforms." *Pattern Recog.*, vol. 15, no. 4, pp. 277–285.

Davis, T. J. [1999]. "Fast Decomposition of Digital Curves into Polygons Using the Haar Transform," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 21, no. 8, pp. 786–790.

Davisson, L. D. [1972]. "Rate-Distortion Theory and Application." *Proc. IEEE*, vol. 60, pp. 800–808.

Delaney, A. H., and Bresler, Y. [1995]. "Multiresolution Tomographic Reconstruction Using Wavelets." *IEEE Trans. Image Processing*, vol. 4, no. 6, pp. 799–813.

Delp, E. J., and Mitchell, O. R. [1979]. "Image Truncation using Block Truncation Coding." *IEEE Trans. Comm.*, vol. COM-27, pp. 1335–1342.

Di Zenzo, S. [1986]. "A Note on the Gradient of a Multi-Image," *Computer Vision, Graphics, and Image Processing*, vol. 33, pp. 116–125.

Dijkstra, E. [1959]. "Note on Two Problems in Connection with Graphs." *Numerische Mathematik*, vol. 1, pp. 269–271.

Djeziri, S., Nouboud, F., and Plamondon, R. [1998]. "Extraction of Signatures from Check Background Based on a Filiformity Criterion," *IEEE Trans. Image Processing*, vol. 7, no. 102, pp. 1425–1438.

Dougherty, E. R. [1992]. *An Introduction to Morphological Image Processing*, SPIE Press, Bellingham, Wash.

Dougherty, E. R. (ed.) [2000]. *Random Processes for Image and Signal Processing*, IEEE Press, New York.

Drew, M. S., Wei, J., and Li, Z.-N. [1999]. "Illumination Invariant Image Retrieval and Video Segmentation," *Pattern Recog.*, vol. 32, no. 8, pp. 1369–1388.

Duda, R. O., and Hart, P. E. [1972]. "Use of the Hough Transformation to Detect Lines and Curves in Pictures." *Comm. ACM*, vol. 15, no. 1, pp. 11–15.

Duda, R. O, Hart, P. E., and Stork, D. G. [2001]. *Pattern Classification*, John Wiley & Sons, New York.

Edelman, S. [1999]. *Representation and Recognition in Vision*, The MIT Press, Cambridge, Mass.

Elias, P. [1952]. "Fourier Treatment of Optical Processes." *J. Opt. Soc. Am.*, vol. 42, no. 2, pp. 127–134.

Elliott, D. F., and Rao, K. R. [1983]. *Fast Transforms: Algorithms and Applications*. Academic Press, New York.

Eng, H.-L., and Ma, K.-K. [2001]. "Noise Adaptive Soft-Switching Median Filter," *IEEE Trans. Image Processing*, vol. 10, no. 2, pp. 242–251.

Equitz, W. H. [1989]. "A New Vector Quantization Clustering Algorithm." *IEEE Trans. Acous. Speech Signal Processing*, vol. ASSP-37, no. 10, pp. 1568–1575.

Etienne, E. K., and Nachtegael, M. (eds.) [2000]. *Fuzzy Techniques in Image Processing*, Springer-Verlag, New York.

Falconer, D. G. [1970]. "Image Enhancement and Film Grain Noise." *Opt. Acta*, vol. 17, pp. 693–705.

Fairchild, M. D. [1998]. *Color Appearance Models*, Addison-Wesley, Reading, Mass.

Federal Bureau of Investigation [1993]. *WSQ Gray-Scale Fingerprint Image Compression Specification*, IAFIS-IC-0110v2, Washington, D. C.

Felsen, L. B., and Marcuvitz, N. [1994]. *Radiation and Scattering of Waves*, IEEE Press, New York.

Ferreira, A., and Ubéda, S. [1999]. "Computing the Medial Axis Transform in Parallel with Eight Scan Operators," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 21, no. 3, pp. 277–282.

Fischler, M. A. [1980]. "Fast Algorithms for Two Maximal Distance Problems with Applications to Image Analysis." *Pattern Recog.*, vol. 12, pp. 35–40.

Fisher, R. A. [1936]. "The Use of Multiple Measurements in Taxonomic Problems," *Ann. Eugenics*, vol. 7, Part 2, pp. 179–188. (Also in *Contributions to Mathematical Statistics*, John Wiley & Sons, New York, 1950.)

Fortner, B., and Meyer, T. E. [1997]. *Number by Colors*, Springer-Verlag, New York.

Fox, E. A. [1991]. "Advances in Interactive Digital Multimedia Systems." *Computer*, vol. 24, no. 10, pp. 9–21.

Fram, J. R., and Deutsch, E. S. [1975]. "On the Quantitative Evaluation of Edge Detection Schemes and Their Comparison with Human Performance." *IEEE Trans. Computers*, vol. C-24, no. 6, pp. 616–628.

Freeman, A. (translator) [1878]. J. Fourier, *The Analytical Theory of Heat*, Cambridge: University Press, London.

Freeman, C. [1987]. *Imaging Sensors and Displays*, ISBN 0–89252–800–1, SPIE Press, Bellingham, Wash.

Freeman, H. [1961]. "On the Encoding of Arbitrary Geometric Configurations." *IEEE Trans. Elec. Computers*, vol. EC-10, pp. 260–268.

Freeman, H. [1974]. "Computer Processing of Line Drawings." *Comput. Surveys*, vol. 6, pp. 57–97.

Freeman, H., and Shapira, R. [1975]. "Determining the Minimum-Area Encasing Rectangle for an Arbitrary Closed Curve." *Comm. ACM*, vol. 18, no. 7, pp. 409–413.

Freeman, J. A., and Skapura, D. M. [1991]. *Neural Networks: Algorithms, Applications, and Programming Techniques*, Addison-Wesley, Reading, Mass.

Frei, W., and Chen, C. C. [1977]. "Fast Boundary Detection: A Generalization and a New Algorithm." *IEEE Trans. Computers*, vol. C-26, no. 10, pp. 988–998.

Frendendall, G. L., and Behrend, W. L. [1960]. "Picture Quality—Procedures for Evaluating Subjective Effects of Interference." *Proc. IRE*, vol. 48, pp. 1030–1034.

Fu, K. S. [1982]. *Syntactic Pattern Recognition and Applications*, Prentice Hall, Englewood Cliffs, N.J.

Fu, K. S., and Bhargava, B. K. [1973]. "Tree Systems for Syntactic Pattern Recognition." *IEEE Trans. Comput.*, vol. C-22, no. 12, pp. 1087–1099.

Fu, K. S., Gonzalez, R. C., and Lee, C. S. G. [1987]. *Robotics: Control, Sensing, Vision, and Intelligence*, McGraw-Hill, New York.

Fu, K. S., and Mui, J. K. [1981]. "A Survey of Image Segmentation." *Pattern Recog.*, vol. 13, no. 1, pp. 3–16.

Garcia, P. [1999]. "The Use of Boolean Model for Texture Analysis of Grey Images," *Computer Vision and Image Understanding*, vol. 74, no. 3, pp. 227–235.

Gdalyahu, Y., and Weinshall, D. [1999]. "Flexible Syntactic Matching of Curves and Its Application to Automated Hierarchical Classification of Silhouettes," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 21, no. 12, pp. 1312–1328.

Gegenfurtner, K. R., and Sharpe, L. T. (eds.) [1999]. *Color Vision: From Genes to Perception*, Cambridge University Press, New York.

Geladi, P., and Grahn, H. [1996]. *Multivariate Image Analysis*, John Wiley & Sons, New York.

Geman, D., and Reynolds, G. [1992]. "Constrained Restoration and the Recovery of Discontinuities." *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, no. 3, pp. 367–383.

Gentleman, W. M. [1968]. "Matrix Multiplication and Fast Fourier Transformations." *Bell System Tech. J.*, vol. 47, pp. 1099–1103.

Gentleman, W. M., and Sande, G. [1966]. "Fast Fourier Transform for Fun and Profit." *Fall Joint Computer Conf.*, vol. 29, pp. 563–578, Spartan, Washington, D. C.

Gharavi, H., and Tabatabai, A. [1988]. "Sub-Band Coding of Monochrome and Color Images." *IEEE Trans. Circuits Sys.*, vol. 35, no. 2, pp. 207–214.

Giannakis, G. B., and Heath, R. W., Jr. [2000]. "Blind Identification of Multichannel FIR Blurs and Perfect Image Restoration," *IEEE Trans. Image Processing*, vol. 9, no. 11, pp. 1877–1896.

Giardina, C. R., and Dougherty, E. R. [1988]. *Morphological Methods in Image and Signal Processing*, Prentice Hall, Upper Saddle River, N.J.

Gonzalez, R. C. [1985]. "Computer Vision." *Yearbook of Science and Technology*, McGraw-Hill, New York, pp. 128–132.

Gonzalez, R. C. [1985]. "Industrial Computer Vision." In *Advances in Information Systems Science*, Tou, J. T. (ed.), Plenum, New York, pp. 345–385.

Gonzalez, R. C. [1986]. "Image Enhancement and Restoration." In *Handbook of Pattern Recognition and Image Processing*, Young, T. Y., and Fu, K. S. (eds.), Academic Press, New York, pp. 191–213.

Gonzalez, R. C., Edwards, J. J., and Thomason, M. G. [1976]. "An Algorithm for the Inference of Tree Grammars." *Int. J. Comput. Info. Sci.*, vol. 5, no. 2, pp. 145–163.

Gonzalez, R. C., and Fittes, B. A. [1977]. "Gray-Level Transformations for Interactive Image Enhancement." *Mechanism and Machine Theory*, vol. 12, pp. 111–122.

Gonzalez, R. C., and Safabakhsh, R. [1982]. "Computer Vision Techniques for Industrial Applications." *Computer*, vol. 15, no. 12, pp. 17–32.

Gonzalez, R. C., and Thomason, M. G. [1978]. *Syntactic Pattern Recognition: An Introduction*. Addison-Wesley, Reading, Mass.

Gonzalez, R. C., and Woods, R. E. [1992]. *Digital Image Processing*. Addison-Wesley, Reading, Mass.

Good, I. J. [1958]. "The Interaction Algorithm and Practical Fourier Analysis." *J. R. Stat. Soc. (Lond.)*, vol. B20, pp. 361–367; *Addendum*, vol. 22, 1960, pp. 372–375.

Goodson, K. J., and Lewis, P. H. [1990]. "A Knowledge-Based Line Recognition System." *Pattern Recog. Letters*, vol. 11, no. 4, pp. 295–304.

Gordon, I. E. [1997]. *Theories of Visual Perception*, 2nd ed., John Wiley & Sons, New York.

Gori, M., and Scarselli, F. [1998]. "Are Multilayer Perceptrons Adequate for Pattern Recognition and Verification?" *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, no. 11, pp. 1121–1132.

Goutsias, J., Vincent, L., and Bloomberg, D. S. (eds) [2000]. *Mathematical Morphology and Its Applications to Image and Signal Processing*, Kluwer Academic Publishers, Boston, Mass.

Graham, R. E. [1958]. "Predictive Quantizing of Television Signals." *IRE Wescon Conv. Rec.*, vol. 2, pt. 2, pp. 147–157.

Graham, R. L., and Yao, F. F. [1983]. "Finding the Convex Hull of a Simple Polygon." *J. Algorithms*, vol. 4, pp. 324–331.

Gray, R. M. [1984]. "Vector Quantization." *IEEE Trans. Acous. Speech Signal Processing*, vol. ASSP-1, no. 2, pp. 4–29.

Gröchenig, K., and Madych, W. R. [1992]. "Multiresolution Analysis, Haar Bases and Self-Similar Tilings of R^n." *IEEE Trans. Information Theory*, vol. 38, no. 2, pp. 556–568.

Grossman, A., and Morlet, J. [1984]. "Decomposition of Hardy Functions into Square Integrable Wavelets of Constant Shape." *SIAM J. Appl. Math.* vol. 15, pp. 723–736.

Guil, N., Villalba, J., and Zapata, E. L. [1995]. "A Fast Hough Transform for Segment Detection," *IEEE Trans. Image Processing*, vol. 4, no. 11, pp. 1541–1548.

Guil, N., and Zapata, E. L. [1997]. "Lower Order Circle and Ellipse Hough Transform," *Pattern Recog.*, vol. 30, no. 10, pp. 1729–1744.

Gunn, S. R. [1998]. "Edge Detection Error in the Discrete Laplacian of a Gaussian," *Proc. 1998 Int'l Conference on Image Processing*, vol. II, pp. 515–519.

Gunn, S. R. [1999]. "On the Discrete Representation of the Laplacian of a Gaussian," *Pattern Recog.*, vol. 32, no. 8, pp. 1463–1472.

Gupta, L., Mohammad, R. S., and Tammana, R. [1990]. "A Neural Network Approach to Robust Shape Classification." *Pattern Recog.*, vol. 23, no. 6, pp. 563–568.

Gupta, L., and Srinath, M. D. [1988]. "Invariant Planar Shape Recognition Using Dynamic Alignment." *Pattern Recog.*, vol. 21, pp. 235–239.

Gupta, L., Wang., J., Charles, A., and Kisatsky, P. [1994]. "Three-Layer Perceptron Based Classifiers for the Partial Shape Classification Problem," *Pattern Recog.*, vol. 27, no. 1, pp. 91–97.

Haar, A. [1910]. "Zur Theorie der Orthogonalen Funktionensysteme." *Math. Annal.*, vol. 69, pp. 331–371.

Habibi, A. [1971]. "Comparison of *N*th Order DPCM Encoder with Linear Transformations and Block Quantization Techniques." *IEEE Trans. Comm. Tech.*, vol. COM-19, no. 6, pp. 948–956.

Habibi, A. [1974]. "Hybrid Coding of Pictorial Data." *IEEE Trans. Comm.*, vol. COM-22, no. 5, pp. 614–624.

Haddon, J. F., and Boyce, J. F. [1990]. "Image Segmentation by Unifying Region and Boundary Information." *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, no. 10, pp. 929–948.

Hall, E. L. [1979]. *Computer Image Processing and Recognition*, Academic Press, New York.

Hamming, R. W. [1950]. "Error Detecting and Error Correcting Codes." *Bell Sys. Tech. J.*, vol. 29, pp. 147–160.

Hannah, I., Patel, D., and Davies, R. [1995]. "The Use of Variance and Entropy Thresholding Methods for Image Segmentation," *Pattern Recog.*, vol. 28, no. 8, pp. 1135–1143.

Haralick, R. M., and Lee, J. S. J. [1990]. "Context Dependent Edge Detection and Evaluation." *Pattern Recog.*, vol. 23, no. 1–2, pp. 1–20.

Haralick, R. M., and Shapiro, L. G. [1985]. "Survey: Image Segmentation." *Comput. Vision, Graphics, Image Proc.*, vol. 29, pp. 100–132.

Haralick, R. M., and Shapiro, L. G. [1992]. *Computer and Robot Vision*, vols. 1 & 2, Addison-Wesley, Reading, Mass.

Haralick, R. M., Sternberg, S. R., and Zhuang, X. [1987]. "Image Analysis Using Mathematical Morphology." *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-9, no. 4, pp. 532–550.

Haralick, R. M., Shanmugan, R., and Dinstein, I. [1973]. "Textural Features for Image Classification." *IEEE Trans Syst. Man Cyb.*, vol. SMC-3, no. 6, pp. 610–621.

Harikumar, G., and Bresler, Y. [1999]. "Perfect Blind Restoration of Images Blurred by Multiple Filters: Theory and Efficient Algorithms," *IEEE Trans. Image Processing*, vol. 8, no. 2, pp. 202–219.

Harmuth, H. F. [1970]. *Transmission of Information by Orthogonal Signals*, Springer-Verlag, New York.

Haris, K., Efstratiadis, S. N, Maglaveras, N., and Katsaggelos, A. K. [1998]. "Hybrid Image Segmentation Using Watersheds and Fast Region Merging," *IEEE Trans. Image Processing*, vol. 7, no. 12, pp. 1684–1699.

Hart, P. E., Nilsson, N. J., and Raphael, B. [1968]. "A Formal Basis for the Heuristic Determination of Minimum-Cost Paths." *IEEE Trans. Syst. Man Cyb*, vol. SMC-4, pp. 100–107.

Hartenstein, H., Ruhl, M., and Saupe, D. [2000]. "Region-Based Fractal Image Compression." *IEEE Trans. Image Processing*, vol. 9, no. 7, pp. 1171–1184.

Haskell, B. G., and Netravali, A. N. [1997]. *Digital Pictures: Representation, Compression, and Standards*, Perseus Publishing, New York.

Haykin, S. [1996]. *Adaptive Filter Theory*, Prentice Hall, Upper Saddle River, N.J.

Healy, D. J., and Mitchell, O. R. [1981]. "Digital Video Bandwidth Compression Using Block Truncation Coding." *IEEE Trans. Comm.*, vol. COM 29, no. 12, pp. 1809–1817.

Heath, M. D., Sarkar, S., Sanocki, T., and Bowyer, K. W. [1997]. "A Robust Visual Method for Assessing the Relative Performance of Edge-Detection Algorithms," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, no. 12, pp. 1338–1359.

Heath, M., Sarkar, S., Sanoki, T., and Bowyer, K. [1998]. "Comparison of Edge Detectors: A Methodology and Initial Study," *Computer Vision and Image Understanding*, vol. 69, no. 1, pp. 38–54.

Hebb, D. O. [1949]. *The Organization of Behavior: A Neuropsychological Theory*, John Wiley & Sons, New York.

Heijmans, H. J. A. M., and Goutsias, J. [2000]. "Nonlinear Multiresolution Signal Decomposition Schemes—Part II: Morphological Wavelets." *IEEE Trans. Image Processing*, vol. 9, no. 11, pp. 1897–1913.

Highnam, R., and Brady, M. [1997]. "Model-Based Image Enhancement of Far Infrared Images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, no. 4, pp. 410–415.

Hojjatoleslami, S. A., and Kittler, J. [1998]. "Region Growing: A New Approach," *IEEE Trans. Image Processing*, vol. 7, no. 7, pp. 1079–1084.

Hoover, R. B., and Doty, F. [1996]. *Hard X-Ray/Gamma-Ray and Neutron Optics, Sensors, and Applications*, ISBN 0-8194-2247-9, SPIE Press, Bellingham, Wash.

Horn, B. K. P. [1986]. *Robot Vision*, McGraw-Hill, New York.

Hotelling, H. [1933]. "Analysis of a Complex of Statistical Variables into Principal Components." *J. Educ. Psychol.*, vol. 24, pp. 417–441, 498–520.

Hough, P. V. C. [1962]. "Methods and Means for Recognizing Complex Patterns." U. S. Patent 3,069,654.

Hsu, C. C., and Huang, J. S. [1990]. "Partitioned Hough Transform for Ellipsoid Detection." *Pattern Recog.*, vol. 23, no. 3–4, pp. 275–282.

Hu, J., and Yan, H. [1997]. "Polygonal Approximation of Digital Curves Based on the Principles of Perceptual Organization," *Pattern Recog.*, vol. 30, no. 5, pp. 701–718.

Hu, M. K. [1962]. "Visual Pattern Recognition by Moment Invariants." *IRE Trans. Info. Theory*, vol. IT-8, pp. 179–187.

Huang, S.-C., and Sun, Y.-N. [1999]. "Polygonal Approximation Using Generic Algorithms," *Pattern Recog.*, vol. 32, no. 8, pp. 1409–1420.

Huang, T. S. [1965]. "PCM Picture Transmission." *IEEE Spectrum*, vol. 2, no. 12, pp. 57–63.

Huang, T. S. [1966]. "Digital Picture Coding." *Proc. Natl. Electron. Conf.*, pp. 793–797.

Huang, T. S., ed. [1975]. *Picture Processing and Digital Filtering*, Springer-Verlag, New York.

Huang, T. S. [1981]. *Image Sequence Analysis*, Springer-Verlag, New York.

Huang, T. S., and Hussian, A. B. S. [1972]. "Facsimile Coding by Skipping White." *IEEE Trans. Comm.*, vol. COM-23, no. 12, pp. 1452–1466.

Huang, T. S., and Tretiak, O. J. (eds.). [1972]. *Picture Bandwidth Compression*, Gordon and Breech, New York.

Huang, T. S., Yang, G. T., and Tang, G. Y. [1979]. "A Fast Two-Dimensional Median Filtering Algorithm." *IEEE Trans. Acoust., Speech, Sig. Proc.*, vol. ASSP-27, pp. 13–18.

Huang, Y., and Schultheiss, P. M. [1963]. "Block Quantization of Correlated Gaussian Random Variables." *IEEE Trans. Commun. Syst.*, vol. CS-11, pp. 289–296.

Hubbard, B. B. [1998]. *The World According to Wavelets-The Story of a Mathematical Technique in the Making*, 2nd ed, A. K. Peters, Ltd., Wellesley, Mass.

Hubel, D. H. [1988]. *Eye, Brain, and Vision*, Scientific Amer. Library, W. H. Freeman, New York.

Huertas, A., and Medione, G. [1986]. "Detection of Intensity Changes with Subpixel Accuracy using Laplacian-Gaussian Masks," *IEEE Trans. Pattern. Anal. Machine Intell.*, vol. PAMI-8, no. 5, pp. 651–664.

Huffman, D. A. [1952]. "A Method for the Construction of Minimum Redundancy Codes." *Proc. IRE*, vol. 40, no. 10, pp. 1098–1101.

Hufnagel, R. E., and Stanley, N. R. [1964]. "Modulation Transfer Function Associated with Image Transmission Through Turbulent Media," *J. Opt. Soc. Amer.*, vol. 54, pp. 52–61.

Hummel, R. A. [1974]. "Histogram Modification Techniques." Technical Report TR-329, F-44620–72C-0062, Computer Science Center, University of Maryland, College Park, Md.

Hunt, B. R. [1971]. "A Matrix Theory Proof of the Discrete Convolution Theorem." *IEEE Trans. Audio and Electroacoust.*, vol. AU-19, no. 4, pp. 285–288.

Hunt, B. R. [1973]. "The Application of Constrained Least Squares Estimation to Image Restoration by Digital Computer." *IEEE Trans. Comput.*, vol. C-22, no. 9, pp. 805–812.

Hunter, R., and Robinson, A. H. [1980]. "International Digital Facsimile Coding Standards." *Proc. IEEE*, vol. 68, no. 7, pp. 854–867.

Hurn, M., and Jennison, C. [1996]. "An Extension of Geman and Reynolds' Approach to Constrained Restoration and the Recovery of Discontinuities," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, no. 6, pp. 657–662.

Hwang, H., and Haddad, R. A. [1995]. "Adaptive Median Filters: New Algorithms and Results," *IEEE Trans. Image Processing*, vol. 4, no. 4, pp. 499–502.

IEEE Computer [1974]. Special issue on digital image processing. vol. 7, no. 5.

IEEE Computer [1988]. Special issue on artificial neural systems. vol. 21, no. 3.

IEEE Trans. Circuits and Syst. [1975]. Special issue on digital filtering and image processing, vol. CAS-2, pp. 161–304.

IEEE Trans. Computers [1972]. Special issue on two-dimensional signal processing, vol. C-21, no. 7.

IEEE Trans. Comm. [1981]. Special issue on picture communication systems, vol. COM-29, no. 12.

IEEE Trans. on Image Processing [1994]. Special issue on image sequence compression, vol. 3, no. 5.

IEEE Trans. on Image Processing [1996]. Special issue on vector quantization, vol. 5, no. 2.

IEEE Trans. Image Processing [1996]. Special issue on nonlinear image processing, vol. 5, no. 6.

IEEE Trans. Image Processing [1997]. Special issue on automatic target detection, vol. 6, no. 1.

IEEE Trans. Image Processing [1997]. Special issue on color imaging, vol. 6, no. 7.

IEEE Trans. Image Processing [1998]. Special issue on applications of neural networks to image processing, vol. 7, no. 8.

IEEE Trans. Information Theory [1992]. Special issue on wavelet transforms and mulitresolution signal analysis, vol. 11, no. 2, Part II.

IEEE Trans. Pattern Analysis and Machine Intelligence [1989]. Special issue on multiresolution processing, vol. 11, no. 7.

IEEE Trans. Signal Processing [1993]. Special issue on wavelets and signal processing, vol. 41, no. 12.

IES Lighting Handbook, 9th ed. [2000]. Illuminating Engineering Society Press, New York.

ISO/IEC [1999]. *ISO/IEC 14495-1:1999: Information technology—Lossless and near-lossless compression of continuous-tone still images: Baseline.*

ISO/IEC JTC 1/SC 29/WG 1 [2000]. *ISO/IEC FCD 15444-1: Information technology—JPEG 2000 image coding system: Core coding system.*

Jahne, B. [1997]. *Digital Image Processing: Concepts, Algorithms, and Scientific Applications*, Springer-Verlag, New York.

Jain, A. K. [1981]. "Image Data Compression: A Review." *Proc. IEEE*, vol. 69, pp. 349–389.

Jain, A. K. [1989]. *Fundamentals of Digital Image Processing*, Prentice Hall, Englewood Cliffs, N.J.

Jain, A. K., Duin, R. P. W., and Mao, J. [2000]. "Statistical Pattern Recognition: A Review," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, no. 1, pp. 4–37.

Jain, J. R., and Jain, A. K. [1981]. "Displacement Measurement and Its Application in Interframe Image Coding." *IEEE Trans. Comm.*, vol. COM-29, pp. 1799–1808.

Jain, R. [1981]. "Dynamic Scene Analysis Using Pixel-Based Processes." *Computer*, vol. 14, no. 8, pp. 12–18.

Jain, R., Kasturi, R., and Schunk, B. [1995]. *Computer Vision*, McGraw-Hill, New York.

Jang, B. K., and Chin, R. T. [1990]. "Analysis of Thinning Algorithms Using Mathematical Morphology." *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, no. 6, pp. 541–551.

Jayant, N. S. (ed.) [1976]. *Waveform Quantization and Coding*, IEEE Press, New York.

Jones, R., and Svalbe, I. [1994]. "Algorithms for the Decomposition of Gray-Scale Morphological Operations," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, no. 6, pp. 581–588.

Jonk, A., van den Boomgaard, S., and Smeulders, A. [1999]. "Grammatical Inference of Dashed Lines," *Computer Vision and Image Understanding*, vol. 74, no. 3, pp. 212–226.

Kahaner, D. K. [1970]. "Matrix Description of the Fast Fourier Transform." *IEEE Trans. Audio Electroacoustics*, vol. AU-18, no. 4, pp. 442–450.

Kak, A. C., and Slaney, M. [2001]. *Principles of Computerized Tomographic Imaging*, Published by the Society for Industrial and Applied Mathematics, Philadelphia, Pa.

Karhunen, K. [1947]. "Über Lineare Methoden in der Wahrscheinlichkeitsrechnung." *Ann. Acad. Sci. Fennicae*, Ser. A137. (Translated by I. Selin in "On Linear Methods in Probability Theory." T-131, 1960, The RAND Corp., Santa Monica, Calif.)

Kasson, J., and Plouffe, W. [1992]. "An Analysis of Selected Computer Interchange Color Spaces." *ACM Trans. on Graphics*, vol. 11, no. 4, pp. 373–405.

Katzir, N., Lindenbaum, M., and Porat, M. [1994]. "Curve Segmentation Under Partial Occlusion," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, no. 5, pp. 513–519.

Khanna, T. [1990]. *Foundations of Neural Networks*, Addison-Wesley, Reading, Mass.

Kim, J. K., Park, J. M., Song, K. S., and Park, H. W. [1997]. "Adaptive Mammographic Image Enhancement Using First Derivative and Local Statistics," *IEEE Trans. Medical Imaging*, vol. 16, no. 5, pp. 495–502.

Kimme, C., Ballard, D. H., and Sklansky, J. [1975]. "Finding Circles by an Array of Accumulators." *Comm. ACM*, vol. 18, no. 2, pp. 120–122.

Kirsch, R. [1971]. "Computer Determination of the Constituent Structure of Biological Images." *Comput. Biomed. Res.*, vol. 4, pp. 315–328.

Kiver, M. S. [1965]. *Color Television Fundamentals*, McGraw-Hill, New York.

Klinger, A. [1976]. "Experiments in Picture Representation Using Regular Decomposition." *Comput. Graphics Image Proc.*, vol. 5, pp. 68–105.

Knowlton, K. [1980]. "Progressive Transmission of Gray-Scale and Binary Pictures by Simple, Efficient, and Lossless Encoding Schemes." *Proc. IEEE*, vol. 68, no. 7, pp. 885–896.

Kohler, R. J., and Howell, H. K. [1963]. "Photographic Image Enhancement by Superposition of Multiple Images." *Photogr. Sci. Eng.*, vol. 7, no. 4, pp. 241–245.

Kokaram, A. [1998]. *Motion Picture Restoration*, Springer-Verlag, New York.

Kramer, H. P., and Mathews, M. V. [1956]. "A Linear Coding for Transmitting a Set of Correlated Signals." *IRE Trans. Info. Theory*, vol. IT-2, pp. 41–46.

Langdon, G. C., and Rissanen, J. J. [1981]. "Compression of Black—White Images with Arithmetic Coding." *IEEE Trans. Comm.*, vol. COM-29, no. 6, pp. 858–867.

Lantućjoul, C. [1980]. "Skeletonization in Quantitative Metallography." In *Issues of Digital Image Processing*, Haralick, R. M., and Simon, J. C. (eds.), Sijthoff and Noordhoff, Groningen, The Netherlands.

Latecki, L. J., and Lakämper, R. [1999]. "Convexity Rule for Shape Decomposition Based on Discrete Contour Evolution," *Computer Vision and Image Understanding*, vol. 73, no. 3, pp. 441–454.

Le Gall, D., and Tabatabai, A. [1988]. "Sub-band Coding of Digital Images Using Symmetric Short Kernel Filters and Arithmetic Coding Techniques." *IEEE International Conference on Acoustics, Speech, and Signal Processing*, New York, NY, pp. 761–765.

Ledley, R. S. [1964]. "High-Speed Automatic Analysis of Biomedical Pictures." *Science*, vol. 146, no. 3461, pp. 216–223.

Lee, J.-S., Sun, Y.-N., and Chen, C.-H. [1995]. "Multiscale Corner Detection by Using Wavelet Transforms." *IEEE Trans. Image Processing*, vol. 4, no. 1, pp. 100–104.

Lee, S. U., Chung, S. Y., and Park, R. H. [1990]. "A Comparative Performance Study of Several Global Thresholding Techniques for Segmentation." *Comput. Vision, Graphics, Image Proc.*, vol. 52, no. 2, pp. 171–190.

Lehmann, T. M., Gönner, C., and Spitzer, K. [1999]. "Survey: Interpolation Methods in Medical Image Processing," *IEEE Trans. Medical Imaging*, vol. 18, no. 11, pp. 1049–1076.

Lema, M. D., and Mitchell, O. R. [1984]. "Absolute Moment Block Truncation Coding and Its Application to Color Images." *IEEE Trans. Comm.*, vol. COM-32, no. 10, pp. 1148–1157.

Levine, M. D. [1985]. *Vision in Man and Machine*, McGraw-Hill, New York.

Liang, K.-C., and Kuo, C.-C. J. [1991]. "Waveguide: A Joint Wavelet-Based Image Representation and Description System." *IEEE Trans. Image Processing*, vol. 8, no. 11, pp. 1619–1629.

Liang, Q., Wendelhag, J. W., and Gustavsson, T. [2000]. "A Multiscale Dynamic Programming Procedure for Boundary Detection in Ultrasonic Artery Images," *IEEE Trans. Medical Imaging*, vol. 19, no. 2, pp. 127–142.

Lillesand, T. M., and Kiefer, R. W. [1999]. *Remote Sensing and Image Interpretation*, John Wiley & Sons, New York.

Lim, J. S. [1990]. *Two-Dimensional Signal and Image Processing*, Prentice Hall, Upper Saddle River, N.J.

Limb, J. O., and Rubinstein, C. B. [1978]. "On the Design of Quantizers for DPCM Coders: A Functional Relationship Between Visibility, Probability, and Masking." IEEE Trans. Comm., vol. COM-26, pp. 573–578.

Lindblad, T., and Kinser, J. M. [1998]. *Image Processing Using Pulse-Coupled Neural Networks*, Springer-Verlag, New York.

Linde, Y., Buzo, A., and Gray, R. M. [1980]. "An Algorithm for Vector Quantizer Design." *IEEE Trans. Comm.*, vol. COM-28, no. 1, pp. 84–95.

Lippmann, R. P. [1987]. "An Introduction to Computing with Neural Nets." *IEEE ASSP Magazine*, vol. 4, pp. 4–22.

Liu, J., and Yang, Y.-H. [1994]. "Multiresolution Color Image Segmentation," *IEEE Trans Pattern Anal. Machine Intell.*, vol. 16, no. 7, pp. 689–700.

Liu-Yu, S., and Antipolis, M. [1993]. "Description of Object Shapes by Apparent Boundary and Convex Hull," *Pattern Recog.*, vol. 26, no. 1, pp. 95–107.

Lo, R.-C., and Tsai, W.-H. [1995]. "Gray-Scale Hough Transform for Thick Line Detection in Gray-Scale Images," *Pattern Recog.*, vol. 28, no. 5, pp. 647–661.

Loncaric, S. [1998]. "A Survey of Shape Analysis Techniques," *Pattern Recog.*, vol. 31, no. 8, pp. 983–1010.

Lu, H. E., and Wang, P. S. P. [1986]. "A Comment on 'A Fast Parallel Algorithm for Thinning Digital Patterns.' " *Comm. ACM*, vol. 29, no. 3, pp. 239–242.

Lu, N. [1997]. *Fractal Imaging*, Academic Press, New York.

Lu, W.-S and Antoniou, A. [1992]. "Two-Dimensional Digital Filters," Marcel Dekker, New York.

MacAdam, D. L. [1942]. "Visual Sensitivities to Color Differences in Daylight," *J. Opt. Soc. Am.*, vol. 32, pp. 247–274.

MacAdam, D. P. [1970]. "Digital Image Restoration by Constrained Deconvolution." *J. Opt. Soc. Am.*, vol. 60, pp. 1617–1627.

Maki, A., Nordlund, P., and Eklundh, J.-O. [2000]. "Attentional Scene Segmentation: Integrating Depth and Motion," *Computer Vision and Image Understanding*, vol. 78, no. 3, pp. 351–373.

Malacara, D. [2001]. *Color Vision and Colorimetry: Theory and Applications*, SPIE Optical Engineering Press, Bellingham, Wash.

Mallat, S. [1987]. "A Compact Multiresolution Representation: The Wavelet Model." *Proc. IEEE Computer Society Workshop on Computer Vision*, IEEE Computer Society Press, Washington, D. C., pp. 2–7.

Mallat, S. [1989a]. "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation." *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-11, pp. 674–693.

Mallat, S. [1989b]. "Multiresolution Approximation and Wavelet Orthonormal Bases of L^2." *Trans. American Mathematical Society*, vol. 315, pp. 69–87.

Mallat, S. [1989c]. "Multifrequency Channel Decomposition of Images and Wavelet models." *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 37, pp. 2091–2110.

Mallat, S. [1998]. *A Wavelet Tour of Signal Processing*, Academic Press, Boston, Mass.

Mallat, S. [1999]. *A Wavelet Tour of Signal Processing*. 2nd ed., Academic Press, San Diego, Calif.

Mallot, A. H. [2000]. *Computational Vision*, The MIT Press, Cambridge, Mass.

Mamistvalov, A. [1998]. "n-Dimensional Moment Invariants and Conceptual Mathematical Theory of Recognition [of] n-Dimensional Solids," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, no. 8, pp. 819–831.

Maragos, P. [1987]. "Tutorial on Advances in Morphological Image Processing and Analysis." Optical Engineering, vol. 26, no. 7, pp. 623–632.

Marchand-Maillet, S., and Sharaiha, Y. M. [2000]. *Binary Digital Image Processing: A Discrete Approach*, Academic Press, New York.

Maren, A. J., Harston, C. T., and Pap, R. M. [1990]. *Handbook of Neural Computing Applications*, Academic Press, New York.

Marr, D. [1982]. *Vision*, Freeman, San Francisco.

Marr, D., and Hildreth, E. [1980]. "Theory of Edge Detection." *Proc. R. Soc. Lond.*, vol. B207, pp. 187–217.

Martelli, A. [1972]. "Edge Detection Using Heuristic Search Methods." *Comput. Graphics Image Proc.*, vol. 1, pp. 169–182.

Martelli, A. [1976]. "An Application of Heuristic Search Methods to Edge and Contour Detection." *Comm. ACM*, vol. 19, no. 2, pp. 73–83.

Martin, M. B., and Bell, A. E. [2001]. "New image compression techniques using multiwavelets and multiwavelet packets," *IEEE Trans. on Image Proc.*, vol. 10, no. 4, pp. 500–510.

Mather, P. M. [1999]. *Computer Processing of Remotely Sensed Images: An Introduction*, John Wiley & Sons, New York.

Max, J. [1960]. "Quantizing for Minimum Distortion." *IRE Trans. Info. Theory*, vol. IT-6, pp. 7–12.

McClelland, J. L., and Rumelhart, D. E. (eds.) [1986]. *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*, vol. 2: *Psychological and Biological Models*, The MIT Press, Cambridge, Mass.

McCulloch, W. S., and Pitts, W. H. [1943]. "A Logical Calculus of the Ideas Imminent in Nervous Activity." *Bulletin of Mathematical Biophysics*, vol. 5, pp. 115–133.

McFarlane, M. D. [1972]. "Digital Pictures Fifty Years Ago." *Proc. IEEE*, vol. 60, no. 7, pp. 768–770.

McGlamery, B. L. [1967]. "Restoration of Turbulence-Degraded Images." *J. Opt. Soc. Am.*, vol. 57, no. 3, pp. 293–297.

Meijering, E. H. W., Niessen, W. J., and Viergever, M. A. [1999]. "Retrospective Motion Correction in Digital Subtraction Angiography: A Review," *IEEE Trans. Medical Imaging*, vol. 18, no. 1, pp. 2–21. [chap 3]

Memon, N., Neuhoff, D. L., and Shende, S. [2000]. "An Analysis of Some Common Scanning Techniques for Lossless Image Coding," *IEEE Trans. Image Processing*, vol. 9, no. 11, pp. 1837–1848.

Mesarović, V. Z. [2000]. "Iterative Linear Minimum Mean-Square-Error Image Restoration from Partially Known Blur," *J. Opt. Soc. Amer.-A. Optics, Image Science, and Vision*, vol. 17, no. 4, pp. 711–723.

Meyer, Y. [1987]. "L'analyses par Ondelettes." *Pour la Science*.

Meyer, Y. [1990]. *Ondelettes et ope'rateurs*, Hermann, Paris.

Meyer, Y. (ed.) [1992a]. *Wavelets and Applications: Proceedings of the International Conference, Marseille, France*, Mason, Paris, and Springer-Verlag, Berlin.

Meyer, Y. (translated by D. H. Salinger) [1992b]. *Wavelets and Operators*, Cambridge University Press, Cambridge, UK.

Meyer, Y. (translated by R. D. Ryan) [1993]. *Wavelets: Algorithms and Applications*, Society for Industrial and Applied Mathematics, Philadelphia.

Meyer, F. G., Averbuch, A. Z., and Strömberg, J.-O. [2000]. "Fast Adaptive Wavelet Packet Image Compression," *IEEE Trans. Image Processing*, vol. 9, no. 7, pp. 792–800.

Meyer, F., and Beucher, S. [1990]. "Morphological Segmentation." *J. Visual Comm., and Image Representation*, vol. 1, no. 1, pp. 21–46.

Meyer, H., Rosdolsky, H. G., and Huang, T. S. [1973]. "Optimum Run Length Codes." *IEEE Trans. Comm.*, vol. COM-22, no. 6, pp. 826–835.

Minsky, M., and Papert, S. [1969]. *Perceptrons: An Introduction to Computational Geometry*, the MIT Press, Cambridge, Mass.

Mirmehdi, M., and Petrou, M. [2000]. "Segmentation of Color Textures," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, no. 2, pp. 142–159.

Misiti, M., Misiti, Y., Oppenheim, G., and Poggi, J.-M. [1996]. *Wavelet Toolbox User's Guide*, The MathWorks, Inc., Natick, Mass.

Mitiche, A. [1994]. *Computational Analysis of Visual Motion*, Perseus Publishing, New York.

Mitra, S. K., and Sicuranza, G. L. (eds.) [2000]. *Nonlinear Image Processing*, Academic Press, New York.

Murase, H., and Nayar, S. K. [1994]. "Illumination Planning for Object Recognition Using Parametric Eigen Spaces," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, no. 12, pp. 1219–1227.

Murino, V., Ottonello, C., and Pagnan, S. [1998]. "Noisy Texture Classfication: A Higher-Order Statistical Approach," *Pattern Recog.*, vol. 31, no. 4, pp. 383–393.

Nagao, M., and Matsuyama, T. [1980]. *A Structural Analysis of Complex Aerial Photographs*, Plenum Press, New York.

Najman, L., and Schmitt, M. [1996]. "Geodesic Saliency of Watershed Contours and Hierarchical Segmentation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, no. 12, pp. 1163–1173.

Narendra, P. M., and Fitch, R. C. [1981]. "Real-Time Adaptive Contrast Enhancement." *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-3, no. 6, pp. 655–661.

Netravali, A. N. [1977]. "On Quantizers for DPCM Coding of Picture Signals." *IEEE Trans. Info. Theory*, vol. IT-23, no. 3, pp. 360–370.

Netravali, A. N., and Limb, J. O. [1980]. "Picture Coding: A Review." *Proc. IEEE*, vol. 68, no. 3, pp. 366–406.

Nevatia, R. [1982]. *Machine Perception*, Prentice Hall, Upper Saddle River, N.J.

Ngan, K. N., Meier, T., and Chai, D. [1999]. *Advanced Video Coding: Principles and Techniques*. Elsevier, Boston.

Nilsson, N. J. [1965]. *Learning Machines: Foundations of Trainable Pattern-Classifying Systems*, McGraw-Hill, New York.

Nilsson, N. J. [1971]. *Problem Solving Methods in Artificial Intelligence*, McGraw-Hill, New York.

Nilsson, N. J. [1980]. *Principles of Artificial Intelligence*, Tioga, Palo Alto, Calif.

Noble, B., and Daniel, J. W. [1988]. *Applied Linear Algebra*, 3rd ed., Prentice Hall, Upper Saddle River, N.J.

Odegard, J. E., Gopinath, R. A., and Burrus, C. S. [1992]. "Optimal Wavelets for Signal Decomposition and the Existence of Scale-Limited Signals." *Proceedings of IEEE Int. Conf. On Signal Proc.*, ICASSP-92, San Francisco, CA, vol. IV, 597–600.

Olson, C. F. [1999]. "Constrained Hough Transforms for Curve Detection," *Computer Vision and Image Understanding*, vol. 73, no. 3, pp. 329–345.

O'Neil, J. B. [1971]. "Entropy Coding in Speech and Television Differential PCM Systems." *IEEE Trans. Info. Theory*, vol. IT-17, pp. 758–761.

Oommen, R. J., and Loke, R. K. S. [1997]. "Pattern Recognition of Strings with Substitutions, Insertions, Deletions, and Generalized Transpositions," *Pattern Recog.*, vol. 30, no. 5, pp. 789–800.

Oppenheim, A. V., and Schafer, R. W. [1975]. *Digital Signal Processing*, Prentice Hall, Englewood Cliffs, N.J.

Oppenheim, A. V., Schafer, R. W., and Stockham, T. G., Jr. [1968]. "Nonlinear Filtering of Multiplied and Convolved Signals." *Proc. IEEE*, vol. 56, no. 8, pp. 1264–1291.

Oyster, C. W. [1999]. *The Human Eye: Structure and Function*, Sinauer Associates, Sunderland, Mass.

Paez, M. D., and Glisson, T. H. [1972]. "Minimum Mean-Square-Error Quantization in Speech PCM and DPCM Systems." *IEEE Trans. Comm.*, vol. COM-20, pp. 225–230.

Pao, Y. H. [1989]. *Adaptive Pattern Recognition and Neural Networks*, Addison-Wesley, Reading, Mass.

Papamarkos, N., and Atsalakis, A. [2000]. "Gray-Level Reduction Using Local Spatial Features," *Computer Vision and Image Understanding*, vol. 78, no. 3, pp. 336–350.

Papoulis, A. [1991]. *Probability, Random Variables, and Stochastic Processes*, 3rd ed., McGraw-Hill, New York.

Park, H., and Chin, R. T. [1995]. "Decomposition of Arbitrarily-Shaped Morphological Structuring," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, no. 1, pp. 2–15.

Parker, J. R. [1991]. "Gray Level Thresholding in Badly Illuminated Images." *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, no. 8, pp. 813–819.

Pattern Recognition [2000]. Special issue on Mathematical Morphology and Nonlinear Image Processing, vol. 33, no. 6, pp. 875–1117.

Pavlidis, T. [1977]. *Structural Pattern Recognition*, Springer-Verlag, New York.

Pavlidis, T. [1982]. *Algorithms for Graphics and Image Processing*, Computer Science Press, Rockville, Md.

Pavlidis, T., and Liow, Y. T. [1990]. "Integrating Region Growing and Edge Detection." *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 3, pp. 225–233.

Peebles, P. Z. [1993]. *Probability, Random Variables, and Random Signal Principles*, 3rd ed., McGraw-Hill, New York.

Pennebaker, W. B., and Mitchell, J. L. [1992]. *JPEG: Still Image Data Compression Standard*. Van Nostrand Reinhold, New York.

Pennebaker, W. B., Mitchell, J. L., Langdon, G. G., Jr., and Arps, R. B. [1988]. "An Overview of the Basic Principles of the Q-coder Adaptive Binary Arithmetic Coder." *IBM J. Res. Dev.*, vol. 32, no. 6, pp. 717–726.

Perez, A., and Gonzalez, R. C. [1987]. "An Iterative Thresholding Algorithm for Image Segmentation." *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-9, no. 6, pp. 742–751.

Perona, P., and Malik, J. [1990]. "Scale-Space and Edge Detection Using Anisotropic Diffusion." *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, no. 7, pp. 629–639.

Persoon, E., and Fu, K. S. [1977]. "Shape Discrimination Using Fourier Descriptors." *IEEE Trans. Systems Man Cyb.*, vol. SMC-7, no. 2, pp. 170–179.

Petrou, M., and Bosdogianni, P. [1999]. *Image Processing: The Fundamentals*, John Wiley & Sons, UK.

Petrou, M., and Kittler, J. [1991]. "Optimal Edge Detector for Ramp Edges." *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, no. 5, pp. 483–491.

Piech, M. A. [1990]. "Decomposing the Laplacian." *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, no. 8, pp. 830–831.

Pitas, I., and Vanetsanopoulos, A. N. [1990]. *Nonlinear Digital Filters: Principles and Applications*, Kluwer Academic Publishers, Boston, Mass.

Plataniotis, K. N, and Venetsanopoulos, A. N. [2000]. *Color Image Processing and Applications*, Springer-Verlag, New York.

Pokorny, C. K., and Gerald, C. F. [1989]. *Computer Graphics: The Principles Behind the Art and Science*, Franklin, Beedle & Associates, Irvine, Calif.

Poynton, C. A. [1996]. *A Technical Introduction to Digital Video*, John Wiley & Sons, New York.

Prasad, L., and Iyengar, S. S. [1997]. *Wavelet Analysis with Applications to Image Processing*, CRC Press, Boca Raton, Fla.

Pratt, W. K. [1991]. *Digital Image Processing*, 2nd ed., John Wiley & Sons, New York.

Pratt, W. K. [2001]. *Digital Image Processing*, 3rd ed., John Wiley & Sons, New York.

Preparata, F. P., and Shamos, M. I. [1985]. *Computational Geometry: An Introduction*, Springer-Verlag, New York.

Preston, K. [1983]. "Cellular Logic Computers for Pattern Recognition." *Computer*, vol. 16, no. 1, pp. 36–47.

Prewitt, J. M. S. [1970]. "Object Enhancement and Extraction." in *Picture Processing and Psychopictorics*, Lipkin, B. S., and Rosenfeld, A. (eds.), Academic Press, New York.

Principe, J. C., Euliano, N. R., and Lefebre, W. C. [1999]. *Neural and Adaptive Systems: Fundamentals through Simulations*, John Wiley & Sons, New York.

Pritchard, D. H. [1977]. "U. S. Color Television Fundamentals—A Review." *IEEE Trans. Consumer Electronics*, vol. CE-23, no. 4, pp. 467–478.

Proc. IEEE [1967]. Special issue on redundancy reduction, vol. 55, no. 3.

Proc. IEEE [1972]. Special issue on digital picture processing, vol. 60, no. 7.

Proc. IEEE [1980]. Special issue on the encoding of graphics, vol. 68, no. 7.

Proc. IEEE [1985]. Special issue on visual communication systems, vol. 73, no. 2.

Qian, R. J., and Huang, T. S. [1996]. "Optimal Edge Detection in Two-Dimensional Images," *IEEE Trans. Image Processing*, vol. 5, no. 7, pp. 1215–1220.

Rabbani, M., and Jones, P. W. [1991]. *Digital Image Compression Techniques*, SPIE Press, Bellingham, Wash.

Rajala, S. A., Riddle, A. N., and Snyder, W. E. [1983]. "Application of One-Dimensional Fourier Transform for Tracking Moving Objects in Noisy environments." *Comp., Vision, Image Proc.*, vol. 21, pp. 280–293.

Reddy, B. S., and Chatterji, B. N. [1996]. "An FFT-Based Technique for Translation, Rotation, and Scale Invariant Image Registration," *IEEE Trans. Image Processing*, vol. 5, no. 8, pp. 1266–1271.

Regan, D. D. [2000]. *Human Perception of Objects: Early Visual Processing of Spatial Form Defined by Luminance, Color, Texture, Motion, and Binocular Disparity*, Sinauer Associates, Sunderland, Mass.

Ritter, G. X., and Wilson, J. N. [2001]. *Handbook of Computer Vision Algorithms in Image Algebra*, CRC Press, Boca Raton, Fla.

Roberts, L. G. [1965]. "Machine Perception of Three-Dimensional Solids." In *Optical and Electro-Optical Information Processing*, Tippet, J. T. (ed.), MIT Press, Cambridge, Mass.

Robertson, A. R. [1977]. "The CIE 1976 Color Difference Formulae," *Color Res. Appl.*, vol. 2, pp. 7–11.

Robinson, G. S. [1976]. "Detection and Coding of Edges Using Directional Masks." University of Southern California, Image Processing Institute, Report no. 660.

Robinson, J. A. [1965]. "A Machine-Oriented Logic Based on the Resolution Principle." *J. ACM*, vol. 12, no. 1, pp. 23–41.

Rock, I. [1984]. *Perception*, W. H. Freeman, New York, New York.

Roese, J. A., Pratt, W. K., and Robinson, G. S. [1977]. "Interframe Cosine Transform Image Coding." *IEEE Trans. Comm.*, vol. COM-25, pp. 1329–1339.

Rosenblatt, F. [1959]. "Two Theorems of Statistical Separability in the Perceptron." In *Mechanisation of Thought Processes: Proc. of Symposium No. 10*, held at the National Physical Laboratory, November 1958, H. M. Stationery Office, London, vol. 1, pp. 421–456.

Rosenblatt, F. [1962]. *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*, Spartan, Washington, D. C.

Rosenfeld, A. (ed.) [1984]. *Multiresolution Image Processing and Analysis*, Springer-Verlag, New York.

Rosenfeld, A. [1999]. "Image Analysis and Computer Vision: 1998," *Computer Vision and Image Understanding*, vol. 74, no. 1, pp. 36–95.

Rosenfeld, A. [2000]. "Image Analysis and Computer Vision: 1999," *Computer Vision and Image Understanding*, vol. 78, no. 2, pp. 222–302.

Rosenfeld, A., and Kak, A. C. [1982]. *Digital Picture Processing*, vols. 1 and 2, 2nd ed., Academic Press, New York.

Rosin, P. L. [1997]. "Techniques for Assessing Polygonal Approximations of Curves," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, no. 6, pp. 659–666.

Rudnick, P. [1966]. "Note on the Calculation of Fourier Series." *Math. Comput.*, vol. 20, pp. 429–430.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. [1986]. "Learning Internal Representations by Error Propagation." In *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*, vol. 1: *Foundations*, Rumelhart, D. E., et al. (eds.), MIT Press, Cambridge, MA., pp. 318–362.

Rumelhart, D. E., and McClelland, J. L. (eds.) [1986]. *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*, vol. 1: *Foundations*, MIT Press, Cambridge, Mass.

Runge, C. [1903]. *Zeit. für Math., and Physik*, vol. 48, p. 433.

Runge, C. [1905]. *Zeit. für Math., and Physik*, vol. 53, p. 117.

Runge, C., and König, H. [1924]. "Die Grundlehren der Mathematischen Wissenschaften." *Vorlesungen über Numerisches Rechnen*, vol. 11, Julius Springer, Berlin.

Russ, J. C. [1999]. *The Image Processing Handbook*, 3rd ed., CRC Press, Boca Raton, Fla.

Sahni, S., and Jenq, J.-F. [1992]. "Serial and Parallel Algorithms for the Medial Axis Transform," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, no. 12, pp. 1218–1224.

Sahoo, P. K., Soltani, S., Wong, A. K. C., and Chan, Y. C. [1988]. "A Survey of Thresholding Techniques." *Comput. Vision, Graphics, Image Proc.*, vol. 4, pp. 233–260.

Saito, N., and Cunningham, M. A. [1990]. "Generalized E-Filter and its Application to Edge Detection." *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, no. 8, pp. 814–817.

Sakrison, D. J., and Algazi, V. R. [1971]. "Comparison of Line-by-Line and Two-Dimensional Encoding of Random Images." *IEEE Trans. Info. Theory*, vol. IT-17, no. 4, pp. 386–398.

Salari, E., and Siy, P. [1984]. "The Ridge-Seeking Method for Obtaining the Skeleton of Digital Images." *IEEE Trans. Syst. Man Cyb.*, vol. SMC-14, no. 3, pp. 524–528.

Salinas, R. A., Abidi, M. A., and Gonzalez, R. C. [1996]. "Data Fusion: Color Edge Detection and Surface Reconstruction Through Regularization," *IEEE Trans. Industrial Electronics*, vol. 43, no. 3, pp. 355–363, 1996.

Sato, Y. [1992]. "Piecewise Linear Approximation of Plane Curves by Perimeter Optimization," *Pattern Recog.*, vol. 25, no. 12, pp. 1535–1543.

Sauvola, J., and Pietikainen, M. [2000]. "Adaptive Document Image Binarization," *Pattern Recog.*, vol. 33, no. 2, pp. 225–236.

Schalkoff, R. J. [1989]. *Digital Image Processing and Computer Vision*, John Wiley & Sons, New York.

Schonfeld, D., and Goutsias, J. [1991]. "Optimal Morphological Pattern Restoration from Noisy Binary Images." *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, no. 1, pp. 14–29.

Schowengerdt, R. A. [1983]. *Techniques for Image Processing and Classification in Remote Sensing*, Academic Press, New York.

Schreiber, W. F. [1956]. "The Measurement of Third Order Probability Distributions of Television Signals." *IRE Trans. Info. Theory*, vol. IT-2, pp. 94–105.

Schreiber, W. F. [1967]. "Picture Coding." *Proc. IEEE* (Special issue on Redundancy Reduction), vol. 55, pp. 320–330.

Schreiber, W. F., and Knapp, C. F. [1958]. "TV Bandwidth Reduction by Digital Coding." *Proc. IRE National Convention*, pt. 4, pp. 88–99.

Schwartz, J. W., and Barker, R. C. [1966]. "Bit-Plane Encoding: A Technique for Source Encoding." *IEEE Trans. Aerosp. Elec. Systems*, vol. AES-2, no. 4, pp. 385–392.

Serra, J. [1982]. *Image Analysis and Mathematical Morphology*, Academic Press, New York.

Serra, J. (ed.) [1988]. *Image Analysis and Mathematical Morphology*, vol. 2, Academic Press, New York.

Sezan, M. I., Rabbani, M., and Jones, P. W. [1989]. "Progressive Transmission of Images Using a Prediction/Residual Encoding Approach." *Opt. Eng.*, vol. 28, no. 5, pp. 556–564.

Shack, R. V. [1964]. "The Influence of Image Motion and Shutter Operation on the Photographic Transfer Function." *Appl. Opt.*, vol. 3, pp. 1171–1181.

Shafarenko, L., Petrou, M., and Kittler, J. [1998]. "Histogram-Based Segmentation in a Perceptually Uniform Color Space." *IEEE Trans. Image Processing*, vol. 7, no. 9, pp. 1354–1358.

Shaked, D., and Bruckstein, A. M. [1998]. "Pruning Medial Axes," *Computer Vision and Image Understanding*, vol. 69, no. 2, pp. 156–169.

Shannon, C. E. [1948]. "A Mathematical Theory of Communication." *The Bell Sys. Tech. J.*, vol. XXVII, no. 3, pp. 379–423.

Shapiro, L. G., and Stockman, G. C. [2001]. *Computer Vision*, Prentice Hall, Upper Saddle River, N.J.

Shapiro, V. A. [1996]. "On the Hough Transform of Multi-Level Pictures," *Pattern Recog.*, vol. 29, no. 4, pp. 589–602.

Shariat, H., and Price, K. E. [1990]. "Motion Estimation with More Than Two Frames." *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, no. 5, pp. 417–434.

Sheppard, J. J., Jr., Stratton, R. H., and Gazley, C., Jr. [1969]. "Pseudocolor as a Means of Image Enhancement." *Am. J. Optom. Arch. Am. Acad. Optom.*, vol. 46, pp. 735–754.

Shih, F. Y., and Wong, W.-T. [1994]. "Fully Parallel Thinning with Tolerance to Boundary Noise," *Pattern Recog.*, vol. 27, no. 12, pp. 1677–1695.

Shih, F. Y. C., and Mitchell, O. R. [1989]. "Threshold Decomposition of Gray-Scale Morphology into Binary Morphology." *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, no. 1, pp. 31–42.

Sid-Ahmed, M. A. [1995]. *Image Processing: Theory, Algorithms, and Architectures*, McGraw-Hill, New York.

Sikora, T. [1997]. "MPEG Digital Video-Coding Standards." *IEEE Signal Processing*, vol. 14, no. 5, pp. 82–99.

Simon, J. C. [1986]. *Patterns and Operators: The Foundations of Data Representations*, McGraw-Hill, New York.

Sklansky, J., Chazin, R. L., and Hansen, B. J. [1972]. "Minimum-Perimeter Polygons of Digitized Silhouettes." *IEEE Trans. Comput.*, vol. C-21, no. 3, pp. 260–268.

Smirnov, A. [1999]. *Processing of Multidimensional Signals*, Springer-Verlag, New York.

Smith, A. R. [1978]. "Color Gamut Transform Pairs." *Proc. SIGGRAPH '78*, published as *Computer Graphics*, vol. 12, no. 3, pp. 12–19.

Smith, M. J. T., and Barnwell, T. P. III [1984]. "A Procedure for Building Exact Reconstruction Filter Banks for Subband Coders." *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, San Diego, Calif.

Smith, M. J. T., and Barnwell, T. P. III [1986]. "Exact Reconstruction Techniques for Tree-Structured Subband Coders." *IEEE Trans. On Acoust., Speech, and Signal Proc.*, vol. 34, no. 3, pp. 434–441.

Sobel, I. E. [1970]. "Camera Models and Machine Perception," Ph.D. dissertation, Stanford University, Palo Alto, Calif.

Sonka, M., Hlavac, V., and Boyle, R. [1999]. *Image Processing, Analysis, and Machine Vision*, 2nd ed., PWS Publishing, New York.

Stark, H. (ed.) [1987]. *Image Recovery: Theory and Application*, Academic Press, New York.

Stark, J. A. [2000]. "Adaptive Image Contrast Enhancement Using Generalizations of Histogram Equalization," *IEEE Trans. Image Processing*, vol. 9, no. 5, pp. 889–896.

Stockham, T. G., Jr. [1972]. "Image Processing in the Context of a Visual Model." *Proc. IEEE*, vol. 60, no. 7, pp. 828–842.

Storer, J. A., and Reif, J. H., eds. [1991]. *Proceedings of DDC '91*, IEEE Computer Society Press, Los Alamitos, Calif.

Strang, G., and Nguyen, T. [1996]. *Wavelets and Filter Banks*, Wellesley-Cambridge Press Wellesley, Mass.

Stumpff, K. [1939]. *Tafeln und Aufgaben zur Harmonischen Analyse und Periodogrammrechnung*, Julius Springer, Berlin.

Sussner, P., and Ritter, G. X. [1997]. "Decomposition of Gray-Scale Morphological Templates Using the Rank Method," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, no. 6, pp. 649–658.

Swets, D. L., and Weng, J. [1996]. "Using Discriminant Eigenfeatures for Image Retrieval," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, no. 8, pp. 1831–1836.

Symes, P. D. [2001]. *Video Compression Demystified*. McGraw-Hill, New York.

Sze, T. W., and Yang, Y. H. [1981]. "A Simple Contour Matching Algorithm." *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-3, no. 6, pp. 676–678.

Tanaka, E. [1995]. "Theoretical Aspects of Syntactic Pattern Recognition," *Pattern Recog.*, vol. 28, no. 7 pp. 1053–1061.

Tanimoto, S. L. [1979]. "Image Transmission with Gross Information First." *Comput. Graphics Image Proc.*, vol. 9, pp. 72–76.

Tasto, M., and Wintz, P. A. [1971]. "Image Coding by Adaptive Block Quantization." *IEEE Trans. Comm. Tech.*, vol. COM-19, pp. 957–972.

Tasto, M., and Wintz, P. A. [1972]. "A Bound on the Rate-Distortion Function and Application to Images." *IEEE Trans. Info. Theory*, vol. IT-18, pp. 150–159.

Teh, C. H., and Chin, R. T. [1989]. "On the Detection of Dominant Points on Digital Curves." *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, no. 8, pp. 859–872.

Thévenaz, P., and Unser, M [2000]. "Optimization of Mutual Information for Multiresolution Image Registration." *IEEE Trans. Image Processing*, vol. 9, no. 12, pp. 2083–2099.

Thomas, L. H. [1963]. "Using a Computer to Solve Problems in Physics." *Application of Digital Computers*, Ginn, Boston.

Thomason, M. G., and Gonzalez, R. C. [1975]. "Syntactic Recognition of Imperfectly Specified Patterns." *IEEE Trans. Comput.*, vol. C-24, no. 1, pp. 93–96.

Thompson, W. B. (ed.) [1989]. "Special Issue on Visual Motion." *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, no. 5, pp. 449–541.

Thompson, W. B., and Barnard, S. T. [1981]. "Lower-Level Estimation and Interpretation of Visual Motion." *Computer*, vol. 14, no. 8, pp. 20–28.

Thorell, L. G., and Smith, W. J. [1990]. *Using Computer Color Effectively*, Prentice Hall, Upper Saddle River, N.J.

Tian, J., and Wells, R. O., Jr. [1995]. *Vanishing Moments and Wavelet Approximation*, Technical Report CML TR-9501, Computational Mathematics Lab., Rice University, Houston, Texas.

Tomita, F., Shirai, Y., and Tsuji, S. [1982]. "Description of Texture by a Structural Analysis." *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-4, no. 2, pp. 183–191.

Topiwala, P. N. (ed.) [1998]. *Wavelet Image and Video Compression*, Kluwer Academic Publishers, Boston, Mass.

Tou, J. T., and Gonzalez, R. C. [1974]. *Pattern Recognition Principles*, Addison-Wesley, Reading, Mass.

Toussaint, G. T. [1982]. "Computational Geometric Problems in Pattern Recognition." In *Pattern Recognition Theory and Applications*, Kittler, J., Fu, K. S., and Pau, L. F. (eds.), Reidel, New York, pp. 73–91.

Tsai, J.-C., Hsieh, C.-H, and Hsu, T.-C. [2000]. "A New Dynamic Finite-State Vector Quantization Algorithm for Image Compression," *IEEE Trans. Image Processing*, vol. 9, no. 11, pp. 1825–1836.

Tsujii, O., Freedman, M. T., and Mun, K. S. [1998]. "Anatomic Region-Based Dynamic Range Compression for Chest Radiographs Using Warping Transformation of Correlated Distribution," *IEEE Trans. Medical Imaging*, vol. 17, no. 3, pp. 407–418.

Udpikar, V. R., and Raina, J. P. [1987]. "BTC Image Coding Using Vector Quantization." *IEEE Trans. Comm.*, vol. COM-35, no. 3, pp. 352–356.

Ueda, N. [2000]. "Optimal Linear Combination of Neural Networks for Improving Classification Performance," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, no. 2, pp. 207–215.

Ullman, S. [1981]. "Analysis of Visual Motion by Biological and Computer Systems," *IEEE Computer*, vol. 14, no. 8, pp. 57–69.

Umbaugh, S. E. [1998]. *Computer Vision and Image Processing: A Practical Approach Using CVIPtools*, Prentice Hall, Upper Saddle River, N.J.

Umeyama, S. [1988]. "An Eigendecomposition Approach to Weighted Graph Matching Problems." *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 10, no. 5, pp. 695–703.

Unser, M. [1995]. "Texture Classification and Segmentation Using Wavelet Frames." *IEEE Trans. on Image Processing*, vol. 4, no. 11, pp. 1549–1560.

Unser, M., Aldroubi, A., and Eden, M. [1993]. "A Family of Polynomial Spline Wavelet Transforms." *Signal Proc.*, vol. 30, no. 2, pp. 141–162.

Unser, M., Aldroubi, A., and Eden, M. [1993]. "B-Spline Signal Processing, Parts I and II." *IEEE Trans. Signal Proc.*, vol. 41, no. 2, pp. 821–848.

Unser, M., Aldroubi, A., and Eden, M. [1995]. "Enlargement or Reduction of Digital Images with Minimum Loss of Information," *IEEE Trans. Image Processing*, vol. 4, no. 5, pp. 247–257.

Vaidyanathan, P. P., and Hoang, P.-Q. [1988]. "Lattice Structures for Optimal Design and Robust Implementaion of Two-Channel Perfect Reconstruction Filter Banks." *IEEE Trans. Acoust., Speech, and Signal Proc.*, vol. 36, no. 1, pp. 81–94.

Vailaya, A., Jain, A., and Zhang, H. J. [1998]. "On Image Classification: City Images vs. Landscapes," *Pattern Recog.*, vol. 31, no. 12, pp. 1921–1935.

Vetterli, M. [1986]. "Filter Banks Allowing Perfect Reconstruction Signal Proc., vol. 10, no. 3, pp. 219–244".

Vetterli, M., and Kovacevic, J. [1995]. *Wavelets and Suband Coding*, Prentice Hall, Englewood Cliffs, N.J.

Voss, K., and Suesse, H. [1997]. "Invariant Fitting of Planar Objects by Primitives." *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, no. 1, pp. 80–84.

Vuylsteke, P., and Kittler, J. [1990]. "Edge-Labeling Using Dictionary-Based Relaxation." *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, no. 2, pp. 165–181.

Walsh, J. W. T. [1958]. *Photometry*, Dover, New York.

Wang, G., Zhang, J., and Pan, G.-W. [1995]. "Solution of Inverse Problems in Image Processing by Wavelet Expansion." *IEEE Trans. Image Processing*, vol. 4, no. 5, pp. 579–593.

Wang, Y.-P., Lee, S. L., and Toraichi, K. [1999]. "Multiscale Curvature-Based Shape Representation Using β-Spline Wavelets." *IEEE Trans. Image Processing*, vol. 8, no. 11, pp. 1586–1592.

Wang, Z., Rao, K. R., and Ben-Arie, J. [1996]. "Optimal Ramp Edge Detection Using Expansion Matching," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, no. 11, pp. 1092–1097.

Watt, A. [1993]. *3D Computer Graphics*, 2nd ed., Addison-Wesley, Reading, Mass.

Wechsler [1980]. "Texture Analysis—A Survey." *Signal Process*, vol. 2, pp. 271–280.

Wei, D., Tian, J., Wells, R. O., Jr., and Burrus, C. S. [1998]. "A New Class of Biorthogonal Wavelet Systems for Image Transform Coding." *IEEE Trans. Image Processing*, vol. 7, no. 7, pp. 1000–1013.

Weinberger, M. J., Seroussi, G., and Sapiro, G. [2000]. "The LOCO-I Lossless Image Compression Algorithm: Principles and Standardization into JPEG-LS." *IEEE Trans. Image Processing*, vol. 9, no. 8, pp. 1309–1324.

Westenberg, M. A., and Roerdink, J. B. T. M. [2000]. "Frequency Domain Volume Rendering by the Wavelet X-ray Transform." *IEEE Trans. Image Processing*, vol. 9, no. 7, pp. 1249–1261.

Weszka, J. S. [1978]. "A Survey of Threshold Selection Techniques." *Comput. Graphics Image Proc.*, vol. 7, pp. 259–265.

White, J. M., and Rohrer, G. D. [1983]. "Image Thresholding for Optical Character Recognition and Other Applications Requiring Character Image Extraction." *IBM J. Res. Devel.*, vol. 27, no. 4, pp. 400–411.

Widrow, B. [1962]. "Generalization and Information Storage in Networks of 'Adaline' Neurons." In *Self-Organizing Systems 1962*, Yovitz, M. C., et al. (eds.), Spartan, Washington, D. C., pp. 435–461.

Widrow, B., and Hoff, M. E. [1960]. "Adaptive Switching Circuits." *1960 IRE WESCON Convention Record*, Part 4, pp. 96–104.

Widrow, B., and Stearns, S. D. [1985]. *Adaptive Signal Processing*, Prentice Hall, Englewood Cliffs, N.J.

Wiener, N. [1942]. *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*, the MIT Press, Cambridge, Mass.

Wilburn, J. B. [1998]. "Developments in Generalized Ranked-Order Filters." *J. Opt. Soc. Amer.-A. Optics, Image Science, and Vision*, vol. 15, no. 5, pp. 1084–1099.

Windyga, P. S. [2001]. "Fast Impulsive Noise Removal," *IEEE Trans. Image Processing*, vol. 10, no. 1, pp. 173–179.

Wintz, P. A. [1972]. "Transform Picture Coding." *Proc. IEEE*, vol. 60, no. 7, pp. 809–820.

Witten, I. H., Neal, R. M., and Cleary, J. G. [1987]. "Arithmetic Coding for Data Compression." *Comm. ACM*, vol. 30, no. 6, pp. 520–540.

Wolff, R. S., and Yaeger, L. [1993]. *Visualization of Natural Phenomena*, Springer-Verlag, New York.

Woods, J. W., and O'Neil, S. D. [1986]. "Subband Coding of Images." *IEEE Trans. Acous. Speech Signal Proc.*, vol. ASSP-35, no. 5, pp. 1278–1288.

Woods, R. E., and Gonzalez, R. C. [1981]. "Real-Time Digital Image Enhancement." *Proc. IEEE*, vol. 69, no. 5, pp. 643–654.

Xu, Y., Weaver, J. B., Healy, D. M., Jr., and Lu, J. [1994]. "Wavelet Transform Domain Filters: A Spatially Selective Noise Filtration Technique." *IEEE Trans. Image Processing*, vol. 3, no. 6, pp. 747–758.

Yachida, M. [1983]. "Determining Velocity Maps by Spatio-Temporal Neighborhoods from Image Sequences." *Comput. Vis. Graph. Image Proc.*, vol. 21, no. 2, pp. 262–279.

Yamazaki, Y., Wakahara, Y., and Teramura, H. [1976]. "Digital Facsimile Equipment 'Quick-FAX' Using a New Redundancy Reduction Technique." *NTC '76*, pp. 6.2-1–6.2-5.

Yang, X., and Ramchandran, K. [2000]. "Scalable Wavelet Video Coding Using Aliasing Reduced Hierarchical Motion Compensation," *IEEE Trans. Image Processing*, vol. 9, no. 5, pp. 778–791.

Yates, F. [1937]. "The Design and Analysis of Factorial Experiments." Commonwealth Agricultural Bureaux, Farnam Royal, Burks, England.

Yitzhaky, Y., Lantzman, A., and Kopeika, N. S. [1998]. "Direct Method for Restoration of Motion Blurred Images," *J. Opt. Soc. Amer.-A. Optics, Image Science, and Vision*, vol. 15, no. 6, pp. 1512–1519.

You, J., and Bhattacharya, P. [2000]. "A Wavelet-Based Coarse-to-Fine Image Matching Scheme in a Parallel Virtual Machine Environment." *IEEE Trans. Image Processing*, vol. 9, no. 9, pp. 1547–1559.

Yu, D., and Yan, H. [2001]. "Reconstruction of Broken Handwritten Digits Based on Structural Morphology," *Pattern Recog.*, vol. 34, no. 2, pp. 235–254.

Yu, S. S., and Tsai, W. H. [1990]. "A New Thinning Algorithm for Gray-Scale Images." *Pattern Recog.*, vol. 23, no. 10, pp. 1067–1076.

Yuan, M., and Li, J. [1987]. "A Production System for LSI Chip Anatomizing." *Pattern Recog. Letters*, vol. 5, no. 3, pp. 227–232.

Zahn, C. T., and Roskies, R. Z. [1972]. "Fourier Descriptors for Plane Closed Curves." *IEEE Trans. Comput.*, vol. C-21, no. 3, pp. 269–281.

Zhang, T. Y., and Suen, C. Y. [1984]. "A Fast Parallel Algorithm for Thinning Digital Patterns." *Comm. ACM*, vol. 27, no. 3, pp. 236–239.

Zhu, H., Chan F. H. Y., and Lam, F. K. [1999]. "Image Contrast Enhancement by Constrained Local Histogram Equalization," *Computer Vision and Image Understanding*, vol. 73, no. 2, pp. 281–290.

Zhu, P., and Chirlian, P. M. [1995]. "On Critical Point Detection of Digital Shapes," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, no. 8, pp. 737–748.

Zimmer, Y., Tepper, R., and Akselrod, S. [1997]. "An Improved Method to Compute the Convex Hull of a Shape in a Binary Image," *Pattern Recog.*, vol. 30, no. 3, pp. 397–402.

Ziou, D. [2001]. "The Influence of Edge Direction on the Estimation of Edge Contrast and Orientation," *Pattern Recog.*, vol. 34, no. 4, pp. 855–863.

Ziv, J., and Lempel, A. [1977]. "A Universal Algorithm for Sequential Data Compression." *IEEE Trans. Info. Theory*, vol. IT-23, no. 3, pp. 337–343.

Ziv, J., and Lempel, A. [1978]. "Compression of Individual Sequences Via Variable-Rate Coding." *IEEE Trans. Info. Theory*, vol. IT-24, no. 5, pp. 530–536.

Zucker, S. W. [1976]. "Region Growing: Childhood and Adolescence." *Comput. Graphics Image Proc.*, vol. 5, pp. 382–399.

Zugaj, D., and Lattuati, V. [1998]. "A New Approach of Color Images Segmentation Based on Fusing Region and Edge Segmentation Outputs," *Pattern Recog.*, vol. 31, no. 2, pp. 105–113.

Index

A

Absolute ADI, 627–628
Accumulative difference image (ADI), 627–628
Accumulator cells, 588
Achromatic light, 44, 283
Acquisition, 45–51
Activation function, 714
Adaptive filters, 237–243
Adaptive mean filter, 241–243
Adaptive median filtering algorithm, 241–242
Adaptive thresholding, 596, 600–607
Adaptive transform coding, 470
Additive cost function, 400
Additive uniform noise, 238
Additivity, 254
ADI, 627–628
Adjacency, 66–68
Admissibility criterion, 377
Aliasing, 62–63
All-systems-safe colors, 292
Alpha-trimmed mean filter, 235–236
ALU, 28
Analysis trees, 396
Angiograms, 9–11, 583
Angiography, 9–11, 583
Apollo mission, 5
Approximation pyramids, 351–352
Area, 661
Arithmetic coding, 444–446
Arithmetic/logic operations, 108–116, 522
Arithmetic logic unit (ALU), 28
Arithmetic mean filter, 231
Array sensor, 46
Artificial intelligence, 2
Astronomy, 13
Augmented pattern vector, 715
Autocorrelation, 206
  coefficients, 414
  theorem, 206
Automata
  finite, 738
  string recognizers, 738
  tree, 741
Automated visual inspection, 16–17
Automaton. *See also* Finite automaton
  frontier-to-root, 742–743
  structural methods example, 739
  syntactic recognition, 735
Average value, 155, 210
Averaging filters, 119, 231, 581
Axis convention (images), 55

B

Background point, 595
Back propagation training, 722–726
Band-limited functions, 62
Bandpass filters, 245–246
Bandreject filters, 244–245
Bandwidth compression, 410
Bartlane cable picture transmission system, 3–4
Basis functions and images, 471–473, 475
Bayes classifier, 705–710
Bayes decision function, 707
Bell Laboratories, 5
Bidirectional frame (B-frame), 511
Bilinear interpolation, 64, 273
Binary compression technique comparison, 455–456
Binary entropy function, 429
Binary image
  compression
    error-free, 455
    standards, 493
  logic operations, 522
  morphology, 520–549
Binary information source, 428–429
Binary symmetric channel (BSC), 428–429
  capacity, 429
Binary symmetric source, 436
Binary tree, 394
Biorthogonal, 358
  coiflet family, 358
  spline family, 358
  wavelets in lossy compression, 487
Bipolar impulse noise, 225–226
Bit allocation, 480
Bit plane
  coding, 448–456
  decomposition, 445
  examples, 450, 451
  slicing, 86–87
Blind deconvolution, 256
Blind spot, 36
Blobs, 169
Block code, 442
Blocking artifact, 477
Block random variable, 430–431
Blurring, 259, 264, 267, 280
Bone scan, 8
Boost filtering, 132, 187
Boundary, 67, 528
  basic rectangle of, 654
  curvature, 654
  decomposing, 649
  descriptors, 653–660
  detection, 585–595
  diameter, 653
  eccentricity, 654
  eigen axes, 683
  extraction, 535
  length, 653
  major axis, 654
  minor axis, 654
  pixels, 66–68

Boundary characteristics
histogram improvement, 608–610
local thresholding, 608–610
Box filter, 120
Bright-light vision, 36
Brightness, 44, 284, 286
Brightness adaptation, 38–42
BSC, 428–429
Bubble chamber image recognition, 744–745
Buffer, 511
Butterworth bandreject filters, 244
Butterworth highpass filters, 183
Butterworth lowpass filter, 173–175
Butterworth notch filter, 246–248

C

CAC, 452
Canny edge detection, 634
CAT, 5
Cartesian product, 55
Cataracts, 36
Catchment basin, 617
CCD imaging, 37
CCITT, 493–495
CDF, 92
Central moments, 673
Chain codes
direction numbers, 644–645
first difference, 645
representation, 644–646
Channel alphabet, 426
Channel decoder, 421, 423–424
Channel encoder, 421, 423–424
Channel matrix, 426
Charge-coupled device (CCD) imaging, 37
Chessboard distance, 69
Choroid, 35, 36
Chromaticity, 286
Chromaticity diagram, 287
CIE (Commission Internationale de l'Eclairage)
chromaticity diagram, 287
standards, 284
CIELAB, 322
Ciliary body, 36
Circular sensor strip, 48
City-block distance, 68
Cleanup, 509
Closing, 528–532
CMS, 322
CMY (cyan, magenta, yellow), 290
color models, 294–295
CMYK (cyan, magenta, yellow, black)
component example, 317
models, 290, 294
scan, 316
Code(s) and Coding
arithmetic, 444
B, 443
bit-plane, 448
block, 442
block truncation, 514
constant area (CAC), 452
contour, 454
differential, 457
distance of, 423
double delta (DDC), 455
efficiency of, 432
Gray, 449
Hamming, 423
hierarchcical, 514
Huffman, 441
truncated, 443
hybrid, 513
instantaneous, 442
interframe, 513
Limpel-Ziv-Welch (LZW), 446
minimum distance of, 423
N-largest, 483
predictive
lossless, 456
lossy, 459
previous pixel, 457
rate of, 433
redundancy, 411–414
relative address (RAC), 453
run-length (RLC), 417, 452, 453
shift, 444
subband, 354
theorems, 430–437
threshold, 480, 482
transform, 467
uniquesly decodeable, 442
variable-length, 413, 440
near optimal, 442
wavelet, 486
zonal, 480
Coefficients
autocorrelation, 414
bit modeling, 509
correlation, 703–704
expansion, 364
Haar example, 371–372
filter subimage, 116
JPEG, 500
scaling function, 369
transform, 471, 474
wavelet function, 370
Cohen-Daubechies-Feauveau wavelets, 490
Coiflet family, 358
Color balancing
corrections, 325
example, 323
Color circle, 319
Color coded image, 307
Color coding
monochrome X-ray image, 306
multispectral images, 311–313
Color complements, 318–320
transformations, 319
Color conversions, 299
Color corrections, 322
Color edge detection, 335–339
Color fundamentals, 283–289
Color gamut, 288
Color image compression, 342–343
example, 342
Color image noise, 339–342
Color image processing, 26, 282–344
exercise problems related to, 344–348
full color, 313
pseudocolor, 302
Color image segmentation, 331–339
Color image sharpening, 330
Color image smoothing, 328–330
Color management systems (CMS), 322
Color mapping functions, 315

Color models, 289–301
CMY(K), 294
converting between RGB and HSI, 299
HSI, 295
L*a*b*, 322
RGB, 290
R′G′B′, 506
$Y'C_bC_b$, 506
Color pixel, 291
Color profiles, 322
Color segmentation, 331–339
Color slicing, 320–321
Color space, 289–301. *See also* Color models
Color spectrum, 283
Color system, 289–301. *See also* Color models
Color transformations, 315–327
Colorimetric, 322
Commission Internationale de l'Eclairage (CIE)
chromaticity diagram, 287
standards, 284
Communication system model, 430
Compactness in regional descriptors, 661
Compression, 27, 409–513. *See also* Codes(s)
bandwidth, 410
binary technique comparison, 455–456
by quantization, 418
by removing redundancies, 411
color image, 342–343
description by principal components, 681
error-free, 440–459
error-free binary, 455
fundamentals, 411–420
lossy, 410, 440, 459–492
ratio, 411
standards, 492–512
binary image, 493–498
continuous tone image, 498–510
JPEG, 498
JPEG 2000, 505
video, 510–512
system model, 421
types, 410
Computer
history of, 4
image processing system, 29
vision, 2
Computer generated illumination function, 597
Computer generated reflectance function, 597
Computerized axial tomography (CAT), 5, 49
Concave boundary descriptor, 654
Conditional average loss, 704
Conditional average risk, 704
Conditional entropy function, 426
Cones, 36
distribution, 37
Cone vision, 36
Conjugate quadrature filters (CQFs), 358
Conjugate symmetry, 155, 196, 210
Conjugate symmetry
Connected components, 67
extraction, 536
topological descriptor, 661
Connected set, 67
Connectionist models, 713
Connectivity, 66
Constant area coding (CAC), 452
Constant quadrature filters, 394
Constrained least squares filtering, 266–269
Consultative Committee of the International Telephone and Telegraph (CCITT), 493–494
makeup codes, 495
terminating codes, 494
Context free grammars, 736–737
Continuous tone image compression standards, 498
Continuous wavelet transform (CWT), 372, 376–377
Contour point, 651–652
Contour tracing and coding, 454
Contraharmonic mean filter, 232
Contrast stretching, 77, 85–86
Controlled redundancy, 423
Convex
boundary descriptor, 654
deficiency, 649
hull, 649
morphological algorithms, 539–541
Convolution, 118
expression, 211
integral, 255
kernel, 118
masks, 118
padding, need for, 199
theorem, 161–162, 199, 205–208
Cornea, 35
Corner boundary descriptor, 654
Correlation
coefficient
matching recognition techniques, 703
object matching example, 703–704
cross, 206
expression, 211
finding matches, 701–702
padding, 207
matching recognition technique, 701–704
theorem, 205–208
Cosine, 211
Cosmetic processing, 178
Cost function, 400
Covariance, 677
Covariance matrix, 677, 707
computation, 677–678
moment invariants, 677
CQFs, 358
Cross correlation, 206
Cross modulated synthesis filters, 357
Cross sectional image, 21–22
CT, 5
Cubic convolution interpolation, 273
Cumulative distribution function (CDF), 92
Curtain of noise, 265
Curvature boundary descriptor, 654

Cutoff frequency, 168
CWT, 372, 376–377
Cyan, magenta, yellow, 290, 294–295
Cyan, magenta, yellow, black, 290, 294–295, 316
Cygnus Loop, 8

D

D-admissible, 434
Dam construction, 620–622
Data compression. *See also* Image compression
  description by principal components, 681
  fundamentals, 411–420
Data redundancy, 411
Daubechies orthonormal filters, 360
Daubechies wavelets, 487
dc component, 155
DCT, 472–477
DDC, 455
Dead zone interval selection, 492
Deblurring filter, 264
Decision boundary, 698
Decision functions, 698
Decision regions, 731
Decision surface complexity, 729–732
Decision-theoretic methods, 698–732
Decoder, 421
Deconvolution filters, 256
Degradation function, 221, 254
  estimating, 256–260
Degree of similarity, 732–733
Delta modulation (DM), 460–461
Density slicing, 303. *See also* Slicing
Derivative operator
  gradient, 134, 577, 608
  Laplacian, 129, 581, 608
Description, 27, 653–687
Descriptors, 653–687
  boundary, 653
  Fourier, 655–659
  moments, 659, 672
  principal components, 675
  regional, 660–675
  relational, 683–687
  topological, 661–665
Device-independent color model, 322
DFT. *See* Discrete Fourier transform (DFT)
Diagonal edge detection, 581
Diagonal masks, 580
Diameter boundary descriptor, 653
Differential coding, 457
Differential pulse code modulation (DPCM), 462
Differentiation, 210
Digital boundary. *See* Boundary
Digital darkroom, 322
Digital edge. *See* Edge
Digital function derivatives, 125
Digital image processing. *See also* Image
  definition of scope, 1–3
  examples, 7–25
  origins of, 3
  steps in, 25–28
  system components, 28–30
Digital images, 1, 25–28, 34–70. *See also* Image
  examples of, 7–25
  redundancies, 411
  representing, 54–56
  zooming and shrinking, 64–66
Digitizer, 28
Dilation
  binary, 523
  equation, 369
  gray-scale, 550
  in watersheds, 620
Dim-light vision, 36
Direct contour tracing, 454
Discontinuity detection, 568–585
Discrete cosine basis functions, 473
Discrete cosine transform (DCT), 472–477
Discrete Fourier transform (DFT), 150. *See also* Fourier transform
  in lossy compression, 470, 472–478
Discrete quantities, 54
Discrete wavelet transform (DWT), 372, 374–376, 379–386. *See also* Wavelet transform
  using Haar function, 362–363
Discriminant analysis, 694
Discriminant functions, 698
Discrimination, 38
Distance measures, 68–70
Distortion measure, 434
Distributivity, 195, 210
Divide lines, 617
DM, 460–461
Double delta coding (DDC), 455
DPCM, 462
DWT. *See* Discrete wavelet transform (DWT)
Dynamic image processing, 627
Dynamic range, 55, 137
Dynamic thresholding, 596

E

Edge
  angle, 577
  concept, 68
  color images, 335
  definition, 572
  detection, 572–585
  direction, 577
  enhancement, 125, 180
  first derivative, 125, 134, 573
  global processing, 587–590
  gradient, 136, 577
  linking, 585–595
  local processing, 585–587
  masks, 578, 579, 582,
  models, 572
  gradient operators, 336, 577
  modifying DWT, 392
  noise, 574
  ramp, 126, 573
  second derivative, 126, 128, 574, 581
  step, 126, 573
  in vector space, 338–339

using wavelets, 391–392
zero-crossings, 573, 583–585
Edge point, 576
Edge segment, 576
Eigen axes, 683
Eigenvalues, 678, 680–681
Eigenvectors, 678, 680–681
Eight-bit fractal image, 88
Eight-directional chain code, 644–645
Electromagnetic spectrum, 2, 43
Element difference moment of order k, 669
Empty sentence, 736
Encoder, 421
End-of-line (EOL) code word, 494
Enhanced SEM image, 107
Enhancement. *See* Image enhancement
Entropy
first-order estimate, 438
measures of texture, 669
second-order estimate, 438
source, 426
EOL code word, 494
Equivocation, 427
Erlang density, 225
Erlang noise, 223–225
Erosion
binary, 525
gray-scale, 552
Error-free binary image compression, 455
Error-free bit-plane coding, 455
Error-free compression, 440–459
bit planes coding, 445–452
lossless predictive coding, 456–459
Euclidean distance, 68
Euclidean vector norm, 267
Euler's formula, 151
Euler's formula (description), 663
Euler number, 661–664
Expansive tree grammars, 740
Explicit quantization, 508
Exponential noise, 225
Extended coding system, 498
Extended function, 201
Extension coding, 432
External markers in watershed segmentation, 624–626
External representation, 643
Extraction of connected components, 536–539
Eye
diagram, 35
light intensity discrimination, 39
image formation, 37–38

F

False color image processing, 302–313
False contouring, 60
Fast Fourier transform (FFT), 208–213. *See also* Fourier transform
Fast wavelet transform (FWT), 372, 379–387
analysis bank, 380
Feature, 27, 693
FFT, 208–213. *See also* Fourier transform
Fidelity criteria, 419–420
Filter, 77, 116. *See also* individual types
adaptive, 237
averaging, 119, 231
bandpass, 245
bandreject, 244,
bank, 397
concept, 116
constrained least squares, 267
design, 213–214, 277–278
frequency domain, 156–160
highpass, 160, 180
homomorphic, 191
geometric mean, 270
inverse, 261–262
lowpass, 119, 167
mean, 119, 231
notch, 246, 248,
order-statistics, 123, 233
padding, 205
parametric Wiener, 270
properties, 159
sharpening 180. *See also* Laplacian, Gradient
spatial, 116
spatial and frequency domains, 161–167
spectrum equalization, 270
transform function, 158
Wiener, 262
Finite automata, 738
Finite automaton, 739. *See also* Automaton
inferring patterns, 746, 749
pattern recognition, 740
state diagram, 747
Finite ensemble, 425
First derivative, 125–126, 573
First difference chain code, 645
Fixed increment correction rule, 716
Flooding in watershed segmentation, 622–624
Fluorescence microscopy, 11–12
Focal length, 37–38
Forward channel transition matrix, 426
Forward Fourier transform, 162. *See also* Fourier transform
Four-directional chain code, 644–645
Fourier, Jean Baptiste Joseph, 148
Fourier descriptors
boundary, 655–659
illustration, 657
properties, 659
reconstruction examples, 658
Fourier series, 148
Fourier spectrum, 80, 152, 157
Fourier transform, 149–213
dc component, 155
expression, 210
fast, 208
filtering, 156–191, 243, 261–270
implementation, 194–214
interference noise pattern, 250
inverse, 150, 154, 198
one and two dimensions, 149–167
phase angle, 152, 154

Fourier transform *(cont.)*
properties, table of, 210
spectrum, 152, 154, 210
use in segmentation, 630–634
Fovea, 36, 37, 38
Fractal image, 24–25, 86, 110
Frame buffers, 30
Fredholm integral of the first kind, 255
Frequency component, 151
Frequency domain, 149–167
definition, 151
filter function, 204
filtering, 156–160
Fourier transform, 151
image enhancement, 147–214
Laplacian, 185–187
moving object detection, 630
properties, 156
Frequency rectangle, 154
Frequency variables, 154
Front-end subsystem, 28
Frontier-to-root automaton, 742–743
Full-color image, 291
processing, 313–315
Full wavelet packet 64-leaf decomposition, 400
Function spaces
relationship between scaling and wavelet, 369
FWT, 372, 379–387

G

Gamma
correction, 81
noise, 223–225
radiation, 8
Gamma-ray imaging, 8–9
Gating function, 63
Gaussian
expression, 164
bandreject filters, 244
curve, 164
difference of, 165
filter, 164, 176, 215
Fourier transform of, 211
function, 164
highpass filters, 165, 184–185
Laplacian of (LoG), 582
lowpass filters, 165, 175, 215, 583
noise, 222–223
notch filter, 246–248
pattern classes, 706–712
pyramid, 353
smoothing function, 584
thresholding function, 605
Generalized delta rule, 713
Geometric fitting property, 528
Geometric mean filter, 232, 270
Geometric redundancy, 416
Geometric transformations, 270–276, 281
GEOS satellites, 15
GIF, 446
Global thresholding, 596, 598. *See also*, Threshold
Gradient, 134–137, 577–581
angle, 577
color images, 336
direction, 577
discontinuity detection, 577–578, 608
edge, 136
magnitude, 134, 577
masks, 578
morphological, 556
operators, 135–136, 577–579
Grammars
expansive tree, 740
inference, 745
learning, 745
string, 736–738
tree, 740–745
Grammatical inference, 745–746
Granular noise, 460
Granulometry, 559
Graphic interchange format (GIF), 446
Graph-theoretic edge linking, 591–595
Gray code, 446
Gray level, 1, 44, 283
color transformation, 308–313
co-occurrence matrix, 669
image, 1
interpolation, 271, 272–276
PDF, 92, 604
resolution, 57–62
slicing, 86
transformations, 78–87
Gray scale, 51

H

Haar function, 362–363
Haar scaling function, 366–369
Haar transform, 360–363
Haar wavelet function coefficients, 371–372
Haar wavelet series expansion, 373–374
Hamming code, 423
Hardcopy devices, 30
Hard thresholding, 392
Harmonic mean filter, 232
Heisenberg box, 385
Heisenberg cell, 385
Heisenberg uncertainty principle, 385
Heuristic approach to global thresholding, 598–600
Hidden layers in neural networks, 722
High-boost filtering, 132, 187–191
High-contrast image examples, 90
Higher-level processes, 2
High-frequency emphasis filtering, 187–191
High-key image, 323
Highpass filters, 160, 165, 180
ideal, 182
Butterworth, 183
Gaussian, 184
spatial representation, 182
Histogram
color images, 326
definition, 88
equalization, 91–94
in HSI color space, 326
implementation, 97–102
linearization, 93
matching, 94–102
specification, 94–102
local, 103
segmentation, 596
statistics, 103–108, 350
Hit-or-miss transformation, 532–534

Homogeneity, 254
Homomorphic filter function, 193
Homomorphic filtering, 191–194
Homomorphic systems, 193
Horizontal mode, 495–497
Horizontal neighbors, 66
Horizontal shifts, 30
Hotelling transform, 678–679
Hough transform, 587–591
Hounsfield, Godfrey N., 5
HSI (hue, saturation, intensity)
  color model, 295–301
    conceptual relationships, 296
  component images, 303
    manipulating, 301
  components, 295–302
    of noisy RGB image, 340
    of RGB image, 302, 317,329, 332
  converting RGB to, 299
  model, 290
  to RGB conversion, 299
  space segmentation, 332
Hue, 286. *See also* HSI (hue, saturation, intensity)
Hue, saturation, intensity. *See* HSI (hue, saturation, intensity)
Huffman coding, 441–442
Huffman source reductions, 441
Human eye
  diagram, 35
  structure, 35–37
Hyperplane, 699
Hypersphere, 708

I

Ideal highpass filters (IHPF), 182–183
Ideal lowpass filters (ILPF), 167–173
I-frame, 511
IGS quantization, 418, 419
IHPF, 182–183
Illumination, 45, 50, 596–598
Illumination-reflectance model, 50, 191, 596
ILPF, 167–173
Image
  acquisition, 25
  analysis, 2
  autocorrelation, 206
  classification, 712
  color processing, 282–348
  compression, 409–518
  convolution, 162, 171, 205, 255
  correlation, 206, 701
  cross correlation, 206
  description, 643–692
  enhancement, 25, 75–219
  model, 50
  morphological processing, 519–566
  multiresolution processing, 349–408
  noise, 112, 222–230, 339–342, 574
  recognition, 693–753
  representation, 644–653
  restoration, 220–281
  segmentation, 567–642
  sensors, 45–49
  understanding, 2
  wavelet processing, 349–408
Image area, 661
Image average value, 155
Image averaging, 112–116
Image blocks in MPEG, 511
Image compression, 409–513
  exercise problems related to, 514–518
  models, 421–424
  standards, 492
Image coordinates convention, 54
Image correlation, 206, 211, 701
Image deconvolution, 256
Image degradation/restoration process, 221–222
Image displays, 30
Image elements, 2, 55
Image enhancement, 25
  arithmetic/logic operations, 108–116
  combining techniques, 137
  frequency domain filtering, 147–214
  gray-level transformations, 78–88
  histogram processing, 88–108
  homomorphic filtering, 191–194
  sharpening, 125–137, 180–191
  smoothing, 119–125, 167–180
  spatial domain filtering, 75–141
Image entropy, 437
Image interpolation, 64, 271
Image negatives, 78–79
Image operation on a pixel basis, 69
Image padding, 199–205, 207, 211
Image processing. *See also* Digital image processing
  vs. image analysis, 2
  system components, 28
Image pyramids, 351–354
Image quality comparisons, 61, 420
Image quantization, 52, 417
Image representation, 54, 644
Image resolution, 57
Image restoration, 25, 220–281
  adaptive filters, 237–243
  bandpass filtering, 245
  bandreject filtering, 244
  constrained least squares filtering, 266
  degradation functions, 256–260
  denoising (noise reduction), 230–254
  frequency-domain filtering, 243–270
  geometric mean filtering, 270
  geometric transformations, 270–276
  inverse filtering, 261
  mean filters, 231–233
  noise models, 222–230
  notch filtering, 246–254
  order-statistics filters, 233–237
  spatial filtering, 230–243
  Wiener filtering, 262–266
Image sampling, 52–66
Image segmentation, 27, 331–339, 567–634
  of color images, 331–339
  connected components, 664–665
  definition, 567
  discontinuity detection, 568

Image segmentation *(cont.)*
edge detection, 572–581
edge linking, 585–595
frequency domain techniques, 630–634
granulometry, 559
Laplacian of Gaussian (LoG), 580–585
line detection, 570
point detection, 569
region-based, 612–617
textural, 558
thresholding, 595–612
morphological watershed, 622–624
motion, 626–633
Image sensing, 7–25, 45–51
Image showing periodic texture, 672
Image subsampling, 58
Image subtraction, 110–112, 274, 626–630
Image understanding, 2
Imaging modalities, 7–25, 45–49
Implicit quantization, 508
Improved gray-scale (IGS), 418–419
Impulse
in degradation estimation, 257
expression, 211
function, 163, 254
noise, 123, 225–226
response, 255
Independent frame (I-frame), 511
Information channel, 425–427
Information preserving, 410
Information systems, 425
Information theory
elements, 424–440
measuring, 424
using, 437
Information transmission theorem, 437
Information user, 426
Infrared images, 12–18, 591, 662, 664
Instantaneous image zoom, 30
Integral inner products, 364
Integrated circuit, 5
Intensity, 1, 44, 283. *See also* HSI (hue, saturation, intensity)
discrimination, 39–40
image, 1
plot, 633
slicing, 303–305
transformation function, 77
Interference noise pattern, 250
Interframe redundancy, 416
Internal markers in watershed segmentation, 624–626
Internal representation, 643
International Commission on Illumination, 284
International Standardization Organization (ISO), 493
Interpixel redundancy, 411, 414–417
Interpolation, 64, 271
Interpolation filter, 352
Intraframe, 511
Invariant moments, 674–675
Inverse continuous wavelet transform, 377
Inverse element difference moment of order k, 669
Inverse filtering, 261–262
Inverse Fourier transform. *See* Fourier transform
Inverse mapper, 422
Inverse transformation kernels, 471
Iris diaphragm, 36
ISO, 493
Isopreference curves, 61, 62
Isotropic filters, 128

J

Jet Propulsion Laboratory (JPL), 5
Joint Bilevel Imaging Group (JBIG), 493
JPEG, 498–505
baseline coding and decoding, 499
coefficient coding categories, 500
default AC code, 501–502
default DC code, 500
JPEG 2000, 505–510

K

Karhunen-Loeve transform (KLT), 476–477
Kernels, 77, 116. *See also* Filter
Key (of an image), 323
Knowledge base, 27
Known set points, 272

L

LANDSAT, 14
Laplacian, 128
color images, 330
expression, 210
filtered images, 131–132
frequency domain, 185–187
Lloyd-Max quantizers, 466
masks, 129, 132, 185, 267, 584
of Gaussian (LoG), 582
operators, 129, 132, 185, 267, 584
pyramids, 353
spatial domain, 128–134
Layered self-adaptive networks, 713
Learning, 712, 745. *See also* Training
Least-mean-square (LMS) delta rule, 717
Least square error filter, 263–264
Leaves, 394
Lempel-Ziv-Welch (LZW) coding, 446–448
decoding, 448
example, 447
Length boundary descriptor, 653
Lifting-based wavelet transform, 506
Light
achromatic, 44, 283
brightness, 44, 284
characterization, 283
chromatic, 284
color, 283
definition, 44
discrimination, 39
gray level, 44, 283

intensity, 44, 283
luminance, 44, 284
monochromatic, 44
primary and secondary colors, 286
radiance, 44, 284
spectrum, 7, 42–45, 284
visible, 44, 283
Light microscopy, 13
Line
detection, 570–572
mask, 570
pair, 57
representation, 589
sensor, 46
Linear degradation, 254–256
Linear operation, 70
Linear spatial filtering, 116
Linear wedge gray-scale image, 82
Linearly separable classes, 716–717
Lloyd-Max decision and reconstruction levels, 466–467
Lloyd-Max quantizers, 466
LMS delta rule, 717
Local noise reduction filter, 239
Local thresholding, 596, 608–610
LoG (Laplacian of a Gaussian), 582, 583
Logic operations
binary images, 522
pixel-by-pixel basis, 108–109
Log transformation, 79–80
Lossless independent coding system, 498
Lossless predictive coding, 456–459
Lossy baseline coding system, 498
Lossy compression, 410, 440, 459–492
Lossy DPCM root-mean-square error summary, 467
Lossy predictive coding, 459–467
Low-contrast images, 90
Low-level processes, 2
Lowpass filters. *See* Filter
Luminance, 44, 284
LZW. *See* Lempel-Ziv-Welch (LZW)

M

Mach bands, 40
Macroblocks, 510, 511
Magnetic resonance image, 21
Magnitude refinement, 509
Major axis boundary descriptor, 654
Mallat's herringbone algorithm, 379–386
Mapper, 422
Mappings, 416
entropy reducing, 439
transformation function, 77
Mariner mission, 5
Markers in watershed segmentation, 624–626
Mask. *See also* Filter
definition, 77
implementation, 116–119
Mask mode radiography, 111
Masking, 109, 137
Mass storage, 29
MAT, 650
Matching
correlation, 206, 701
recognition techniques, 698–704
shape numbers, 732–733
Max filter, 124, 234–235
Maximum probability in texture, 669
Mean
computation, 104, 145, 229
filters, 119, 231–233
vector, 676–678, 707,
Mean-square signal-to-noise ratio, 420
Mean vector computation, 677–678
Medial axis transformation (MAT), 650
Median filter, 123, 234
Mexican hat function, 582
Mexican hat wavelet, 377
Microdensitometers, 47
Microns, 43
Microprocessor, 5
Microscopy, 13
Microwave band imaging, 18–19
Middle-key image, 323
Mid-level processes, 2
Midpoint filter, 235
Min filter, 124, 234–235
Minimum distance classifier, 698–701
Minimum mean square error filtering, 262–266
Minimum perimeter polygons, 646
Minor axis boundary descriptor, 654
Minutiae recognition, 696
Mixed adjacent pixels, 67
Modeling in degradation estimation, 258–260
Modulation function, 251
Moiré patterns, 62–63
Moment invariants, 674
Moments
one-dimensional functions, 659
two-dimensional functions, 672
Monochromatic light, 44
Morphological mage processing, 27, 519–560
algorithms, 534–549
filtering, 531
gradient, 556
region filling, 536
smoothing, 556
textural segmentation, 558
watersheds segmentation, 617–626
Motion segmentation,
spatial techniques, 626–630
frequency domain techniques, 630–634
Motion blur, 259, 264, 267, 280
Motion Picture Experts Group, 510
MPEG (Motion Picture Experts Group), 510
MRA, 363, 366–369
MRI (magnetic resonance image), 21

Multilayer feedforward neural networks, 719–732
Multilevel thresholding, 596
Multiresolution analysis (MRA), 363
  dilation, MRA, and refinement equation, 369
  fundamental requirements, 366–369
Multiresolution expansions, 363–372
Multiresolution processing, 349–403
  exercise problems related to, 404–409
Multiresolution theory, 349
Multispectral images, 14, 311–313, 611, 664, 710
Multispectral thresholding, 611–612
Mutual information, 427

N

Nanometers, 43
Nearest neighbor interpolation, 64
Near-infrared region, 45
Negative ADI, 627–628
Negative image, 79
Neighborhood averaging, 120, 231, 328–330
Neighbors of a pixel, 66
Networking, 30
Neural networks, 719–732
Neurocomputers, 713
Neuromorphic systems, 713
Neurons, 712
Newton, Isaac, 283
Newton-Raphson algorithm, 269
Nodes, 394
Noise
  additive, 265
  additive uniform, 238
  color image, 339–342
  curtain, 265
  granular, 460
  interference pattern, 250
  models, 222
  parameter estimation, 227–230
  PDF, 222–227
  bipolar impulse, 225–226
  Erlang, 223–225
  exponential, 225
  gamma, 223–225
  Gaussian, 222–223
  uniform, 225
  periodic, 227
  properties, 222
  reduction
    filter, 239
    image averaging, 113
    periodic, 243–253
  removal using wavelets, 391–394
  salt-and-pepper, 123, 243
  shot, 226
  spatially periodic, 222
  spike, 226
  white, 222
Noiseless coding theorem, 430–432
Noisy binary channel, 432
Noisy coding theorem, 432–433
Noisy edge, 574–576
Noisy object signature, 695
Noisy shapes, 726
Nonadaptive transform coding, 470
Nonlinear computing elements, 712
Nonlinear operation, 70
Nonlinear spatial filters, 118
Nonseparable classes, 717–719
Nonterminals in grammars, 736, 740
Normalized central moments, 674
Notch filter, 159, 246–248
Nyquist rate, 381

O

Object class generation, 737
Objective fidelity criteria, 419
Object point, 595
Object recognition, 27, 693–750
  Bayes classifier, 705
  correlation, 701
  correlation coefficient, 703
  decision-theoretic methods, 698–732
  feature selection, 695
  matching, 698
  minimum-distance classifier, 699, 708
  neural networks, 712
  optimum classifier, 704
  pattern, 693
  pattern class, 693
  pattern vector, 694
  perceptron, 713
  semantics, 738
  string, 696, 736
  string matching, 734
  string recognition, 735
  structural methods, 733–749
  tree, 697, 740
  tree recognition, 740
One-dimensional compression, 493–494
One-dimensional continuous wavelet transform
  example, 377
One-dimensional discrete wavelet transform, 375–376
One-dimensional fast wavelet transform, 382–383
One-dimensional filters, 359
One-dimensional inverse fast wavelet transform, 384
One-dimensional pattern classes, 706
One-dimensional run-length coding, 452–453
Opening in morphology, 528, 548, 554
Optical illusions, 42
Optimal global thresholding, 602–607
Optimal predictors, 461
Optimal quantization, 465–466
Optimal variable length codes, 442–444
Optimal wavelet packet analysis tree, 401
Optimum notch filtering, 248–253
Optimum statistical classifiers, 704–710
Order
  boundary, 654

statistics filters, 123–124, 233–237
Orthonormal, 358
Over segmentation, 624–626

P

Packets (in JPEG 2000), 509
Padding, 199–205
Pan, 30
Parallel distributed processing (PDP), 713
Pass mode, 495–497
Pattern, 693–697
Pattern classes, 693–697
Pattern primitives, 683–687, 735
Pattern recognition. *See* Object recognition
PDF (portable document format), 446
PDF (probability density function). *See* Probability density function (PDF)
PDP, 713
PDQ, 454
Pel, 2, 55
Perceptually uniform, 322
Periodic noise, 227
Periodic noise reduction, 243–253
Periodic texture, 672
Periodicity in DFT, 196, 210
PET image, 8
P-frame, 511
Phase angle, 152, 154, 210
Photon, 44
Photopic, 36
Picture elements, 2, 55
Piecewise linear transformation functions, 85–87
Pigments, 286
Pixel
  adjacency, 66
  border, 67
  connectivity, 67
  definition, 2, 55
  depth, 291
  distance between, 68
  neighbors of, 66
  path, 67
  regions, 67
  relationships, 66
  replication, 64
Pixel basis image operation, 69
Pixel-by-pixel basis, 69, 108–109
Pixel-by-pixel classification, 712
Planck's constant, 43
Point. *See also* Pixel
  definition, 576
  detection, 569–570
  isolated, 570
  processing, 77
Point spread function (PSF), 255
Polar representation, 210
Polygonal approximations
  digital boundary, 646–648
  merging techniques, 647
  representations, 646–648
  splitting techniques, 647–648
Portable document format (PDF), 446
Position-invariant degradation, 254–256
Positive ADI, 627–628
Positive sample set, 746
Positron emission tomography (PET), 8
Power-law transformation, 80–84
Power spectrum, 152, 154, 157, 210, 263, 270, 281
Perceptron, 713–719
Prediction error, 456
Prediction error example, 458
Prediction residual, 352
Prediction residual pyramid, 352–353
Prediction technique comparison, 463–465
Predictive coding, 457
Predictive differential quantizing (PDQ), 454
Predictive frame (P-frame), 511
Previous pixel coding, 457
Previous pixel predictor, 457
Prewitt operators, 578–579
Primary colors, 284
Primitive elements, 684, 741
Principal component description, 678–679
Probability density function (PDF), 92, 222–229
Production rules, 736–738, 740
Pruning, 545–547
Pseudocolor image processing, 302–313
PSF, 255
Psychovisual redundancy, 411, 417–419
Pupil, 36
Pyramidal image structure, 351

Q

Quadrature mirror filters (QMFs), 358
Quantization, 52–66, 417
  basic concepts, 52–54
  function, 465
  illustration, 466–467
  optimal lossy, 465–466
Quantizer design, 492

R

RAC, 453
Radiance, 44–45, 284
Radio band imaging, 20
Radiography, 9
Ranger 7, 5
Ranking function, 740–742
Rate controller, 511
Rate distortion function, 434
  binary symmetric source, 436
  computing, 435
Rate distortion theory, 434
Rayleigh density function, 482
Rayleigh noise, 223
Real-valued expansion coefficients, 364
Real-valued expansion functions, 364
Receptor density, 36
Recognition. *See* Object recognition
Reconstruction error *vs.* subimage size, 478
Rectangle
  basic, 654
  Fourier transform of, 211
Red, green, blue. *See* RGB (red, green, blue)
Redundancy, 412–419
  controlled, 423

Reference element, 494
Reference image, 629–630
Reference line, 494
Reference shapes, 726
Refinement equation, 369
Reflectance components, 50, 191, 596
Reflectivity, 51
Region, 66. *See also* Connected component, Boundary
  description, 660
  growing, 613–615
  filling, 535–536
  growing, 613–615
  merging, 615
  moments, 672–675
  pixels, 66–68, 67
  segmentation, 612
  splitting, 615
  texture, 665–671
Region of interest (ROI) processing, 109
Regular grammars, 736
Relational descriptors, 683–687
Relative address coding (RAC), 453
Relative data redundancy, 411
Remote sensing, 13
Representation, 27, 643–653
  boundary segments, 649–650
  polygonal approximations, 646–648
  signatures, 648–649
  skeletons, 650–653
Reseau marks, 5, 272
Resolution
  gray-level, 57
  line pairs, 57
  spatial, 57
Restoration. *See* Image restoration
Retina, 35
Reversible mappings, 416
RGB (red, green, blue)
  color cube, 290
  color model, 290–294
    conceptual relationships, 296
  components
    examples, 329, 317
    thresholding, 611–612
  conversion
    to CMY, 294
    to and from HSI, 299
  image description, 675
  model, 290
  safe-color cube, 294
  safe colors, 292
  vector space segmentation, 333–335
Ringing, 169, 173
Roberts cross-gradient operators, 134, 577–578
Rods, 36
Roentgen, Wilhelm Conrad, 5
ROI processing, 109
Rolling ball, 555
Root-mean-square error, 419
Root nodes, 394
Rotation
  expression, 210
  Fourier transform, 195
  invariance, 128
Rotational invariance
  filters, 128
  descriptors, 643
Rubber sheet distortions, 661
Rubber sheet transformation, 270
Run-length coding, 452–453
  illustration, 416

S

Safe browser colors, 292
Safe RGB colors, 292
Safe Web colors, 292
Salt-and-pepper noise, 123, 225, 243
Sampling, 52–66
Sampling rate, 62
Saturation, 286. *See also* HSI (hue, saturation, intensity)
Scaling
  expression, 210
  Fourier transform, 195
  function, 363
    MRA, 369
    multiresolution expansions, 365–369
  vector, 369
Scanning electron microscope (SEM), 23–25, 105, 157
Scene, 45. *See also* Image
Sclera, 35
Scotopic, 36
Scroll, 30
Secondary colors, 285
Second-order derivative, 126, 573
Seed regions, 614
Segmentation. *See* Image segmentation
SEM. *See* Scanning electron microscope
Semantic rules, 738
Sensing, 28, 45. *See also* Image sensing
Sensor. *See also* Image sensing
  arrangements, 46–47
  arrays, 49, 54
  strips, 48
Separability of DFT, 197, 210
Separable scaling function, 386
Sequential baseline system, 498
Series expansion, 364–365
Set theory concepts, 520–521
Shannon's first theorem, 430–431
Shannon's second theorem, 433
Shape classification using neural networks, 726–729
Shape number
  boundary, 654
  for comparing shapes, 733–734
  computing, 655
  generating procedure, 656
Sharpening
  color image, 327–331
  frequency domain filters, 180–191
  Laplacian, 330
  morphological, 556
  spatial filters, 125–137
Shift codes, 444
Short-term storage, 30
Shot noise, 226
Shrinking a digital image, 64–66
Sigmoidal activation function, 721
Signatures
  distance *vs.* angle, 648
  noisy object, 695
  representations, 648–649
Significance propagation, 509

Simple image formation model, 50–51
Simultaneous contrast, 40–41
Sine (two-dimensional), 211, 270
Single imaging sensor, 46
Single-scale filter bank, 387–388
Size invariance of descriptors, 643
Skeletonizing, 650
Skeletons
  definitions, 543, 650
  morphological algorithms, 543–545
  representations, 650–653
Slicing
  bit planes, 86–87
  color, 320–321
  density, 303
  gray-level, 86
  intensity, 303
Slope density function, 649
Slope overload, 460
Smith and Barnwell filter, 358
Smoothing
  color image, 327–331
  effects, 121–122
  example, 581
  frequency-domain filters, 167–180
  linear filters, 119–123
  morphological, 556
  spatial filters, 119–124
Sobel operators, 136, 138, 578–579, 584
Soft thresholding, 392
Software for image processing, 29
Sound imaging, 20
Source alphabet, 425
Source coding theorem, 433–437
Source decoder, 421–422
Source encoder, 421
Spaceborne radar image, 20
Space invariant, 254
Spatial coordinates, 1,
Spatial domain
  definition, 75
  image enhancement, 75–141
  Laplacian, 186–187
  padded lowpass filter, 204
Spatial enhancement methods
  combining, 137–141
Spatial filtering implementation, 116–119
Spatial Gaussian filter, 176
Spatial Gaussian smoothing function, 584
Spatial masks. *See also* Filter
  gray-scale and RGB color images, 314
Spatial periodic noise, 222
Spatial redundancy, 416
Spatial resolution, 57–62
Spatial transformations, 271–272
Specialized image processing hardware, 28
Spectral density. *See* Power spectrum
Spectrum, 7, 42
  equalization filter, 270
  expression, 210
  Fourier transform, 152, 154, 210
  splitting, 397
Speed of light, 43
Spike noise, 226
Spline family, 358
Splitting techniques in polygonal approximations, 647–648
Starting point invariance, 645
Starting symbol, 736
State diagram, 739
Statistical moments, 659–660
String, 735
String descriptions
  pattern generation, 696
  relational descriptors, 685
String grammars, 736–738
String matching, 734–35
String recognizers, 738
Structurally complete, 746
Structural methods in object recognition, 733–749
Structuring element, 524
Subband, 354
  coding, 354–360
Subimage size selection, 477–479
Subjective brightness, 38
Subjective fidelity criteria, 419
Successive doubling method, 209
Superposition integral of the first kind, 255
Symbol coder, 422
Symlets, 391
  reconstruction filters, 389
Symmetrical wavelets. *See* Symlets
Syntactic recognition, 738–745
Synthesis filters, 355
  cross-modulated, 357

T

Tagged image file format (TIFF), 446
TEM, 23
Template. *See also* Filter
  correlation, 206
  definition, 77
  implementation, 116–119
Terminals, 736
Texture
  spectral approaches, 670–671
  structural approaches, 670
  image showing periodic, 672
  measures, 668
  regional descriptors, 665–671
  segmentation, 558
Thematic bands, 14
Thickening, 541–543
Thinning, 541
Three-dimensional objects, 48
Three-scale FWT, 389
  filter bank, 395
Three-scale wavelet packets, 396–398
Threshold, 595–612
  adaptive, 600
  estimating, 602–607
  foundation, 595–596
  function, 86
  gradient image, 591
  hard, 392
  LoG, 584
  mask, 483
  segmentation, 712
  soft, 392
  types, 596
  variables, 611–612
Threshold coding, 480, 482–484
Thresholding. *See* Threshold, Image segmentation
Tiepoints, 271

TIFF, 446
Tile, 24, 385
  JPEG 2000 components, 506
Time domain, 151
Time-frequency plane, 385
Time-frequency tiles, 385–386
Tonal corrections, 324
Tonal range, 323
Tonal transformations, 323–324
Tone corrections, 322
Top-hat transformation, 557–558
Topological descriptors, 661–665
Topology, 661
Training, 712, 745. *See also* Learning
  algorithms, 716–717
  back propagation, 722–726
  patterns, 712
  set, 712
Transform
  coding, 467–485
  coefficient masking function, 474
  coefficients, 471
  Discrete Fourier (DFT). *See* Fourier transform
  discrete cosine (DCT), 472
  Fourier. *See* Fourier transform
  Haar, 360
  Hotelling, 678–679
  Hough, 587–591
  Karhunen-Loeve, 476–477
  Principal components, 679
  selection, 470–477
  Walsh-Hadamard (WHT), 471
  wavelet. *See* Wavelet transform
Transformation function examples, 310
Translation (of DFT), 194–195, 210
Translation invariance of descriptors, 643
Transmission electron microscope (TEM), 23
Transmissivity, 51
Tree automata, 741
Tree frontier, 685
Tree grammars, 740–745
Tristimulus values, 286
Truncated Huffman coding, 443
Two-band filter bank, 354
Two-band subband coding, 354–355
Two-dimensional compression standards, 494
Two-dimensional DFT. *See* Fourier transform
Two-dimensional Fourier transform. *See* Fourier transform
Two-dimensional FWT, 388–389. *See also* Transform
Two-dimensional moment invariants, 675
Two-dimensional run-length coding, 453–454
Two-dimensional wavelet packet decomposition, 399
Two-scale FWT
  analysis bank, 381
  computing, 382
  synthesis bank, 384
Two-scale inverse FWT, 385

U

ULSI, 5
Ultra large scale integration (ULSI), 5
Ultrasound imaging, 22
Ultraviolet light imaging, 11
Uncertainty, 426
Undersampled, 62
Uniformity measures, 669
Uniform noise, 225
Uniform probability density function, 93, 225
Unipolar noise impulses, 226
Uniquely decodable, 442
Unsharp masking, 132–133, 187–191

V

Vaidyanathan and Hoang filter, 359
Variable-length coding, 440–443
  example, 413
  graphic representation, 413
  near optimal, 442–444
Vector space edge detection, 338–339
Vertical mode, 495–497
Vertical neighbors, 66
Vertical shifts, 30
Very high resolution radiometer (VHRR) image, 179
Very large scale integration (VLSI), 5
VHRR image, 179
Video compression standards, 510
Vision, 2
Visual
  inspection, 16
  perception, 34–42
VLSI, 5
Von Neumann, John, 4–5

W

Walsh-Hadamard basis functions, 472
Walsh-Hadamard transform, 471–477
Watersheds, 617–624
Wavelength, 44
Wavelet, 26, 349–403
  exercise problems related to, 404–409
  image processing, 391–394
Wavelet-based edge detection, 391
Wavelet-based noise removal, 391
Wavelet coding, 486–492
  decomposition level selection, 491
  quantizer, 492
  system, 486
  wavelet selection, 487
Wavelet functions, 369–372
  coefficients, 370
  Cohen-Daubecies-Feauveau biorthogonal, 402–403
  Haar, 371
  Mexican hat, 377
  symlets, 389–391
Wavelet packets, 394–402
Wavelet selection, 487–491
Wavelet series expansion, 372–374
Wavelet transform, 349, 372–402
  approximation and detail coefficients, 375

continuous (CWT), 376
discrete (DWT), 374
fast (FWT), 379–386
analysis bank, 380, 381, 388, 395, 397
filters
Cohen-Daubecies-Feauveau biorthogonal, 403
Daubechies 8–tap orthonormal, 360
fourth order symlet, 390
Haar, 382
JPEG 2000 9–7 biorthogonal, 507
synthesis bank, 383, 384, 388
inverse, 383
spectrum splitting characteristics, 394
tree representation, 394
one dimension, 372–379
two dimensions, 386
Wavelet vector, 370
WBS, 452
Web colors, safe, 292
Weber ratio, 39–40
Weighted average, 120
Weighting function, 251
Weight vector, 715
White block skipping (WBS), 452
White noise, 222
Widrow-Hoff delta rule, 717
Wiener filter, 262–266
Windows, 77, 116
Wraparound error, 201

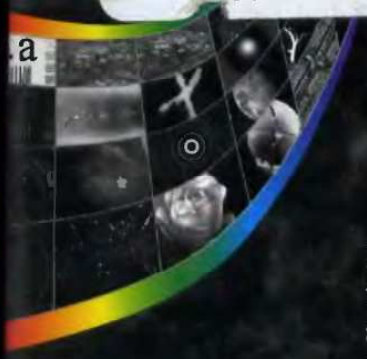
X

X-ray imaging, 9–11

Z

Zero-crossings, 573–574, 583–585
Zero-memory channel, 433
Zero-memory source, 430–431
Zero-order interpolation, 273
Zero-phase-shift filters, 158
Zonal coding, 480
Zonal mask, 480
Zooming and shrinking, 64–66
Z-transform, 355, 358

BIBLIOTECA U.D.B.
031326



Digital Image Processing *Second Edition*

Rafael C. Gonzalez • Richard E. Woods

Digital Image Processing has been the leading textbook in its field for more than 20 years. As was the case with the 1977 and 1987 editions by Gonzalez and Wintz, and the 1992 edition by Gonzalez and Woods, the present edition was prepared with students and instructors in mind. The material is timely, highly readable, and illustrated with numerous examples of practical significance. All mainstream areas of image processing are covered, including a totally revised introduction and discussion of image fundamentals, image enhancement in the spatial and frequency domains, restoration, color image processing, wavelets, image compression, morphology, segmentation, and image description. Coverage concludes with a discussion of the fundamentals of object recognition.

Although the book is completely self-contained, a Companion Website (see inside front cover) provides additional support in the form of review material, answers to selected problems, laboratory project suggestions, and a score of other features. A supplementary instructor's manual is available to instructors who have adopted the book for classroom use.

New Features

- New chapters on wavelets, image morphology, and color image processing.
- More than 500 new images and over 200 new line drawings and tables.
- A revision and update of all chapters, including topics such as segmentation by watersheds.
- Numerous new examples with processed images of higher resolution.
- A reorganization that allows the reader to get to the material on actual image processing much sooner than before.
- Updated image compression standards and a new section on compression using wavelets.
- A more intuitive development of traditional topics such as image transforms and image restoration.
- Updated bibliography.

ISBN 0-20-118075-8
90000
9 780201 180756

Pearson Education

Prentice Hall
Upper Saddle River, NJ 07458
www.prenhall.com

l Image
ssing *Second Edition*

ez • Richard E. Woods

n the leading textbook in its field for more
h the 1977 and 1987 editions by Gonzalez
by Gonzalez and Woods, the present edi-
and instructors in mind. The material is
rated with numerous examples of practi-
areas of image processing are covered,
duction and discussion of image, funda-
e spatial and frequency domains, restora-
velets, image compression, morphology,
tion. Coverage concludes with a discus-
t recognition.
ely self-contained, a Companion Website
additional support in the form of review
blems, laboratory project suggestions, and
ementary instructor's manual is available
the book for classroom use.

morphology, and color image processing.
over 200 new line drawings and tables.
apters, including topics such as segmenta-

processed images of higher resolution.
e reader to get to the material on actual
than before.
andards and a new section on compres-

of traditional topics such as image trans-

ISBN 0-20-118075-8
90000
9 780201 180756

Gonzalez
Woods

Digital Image Processi

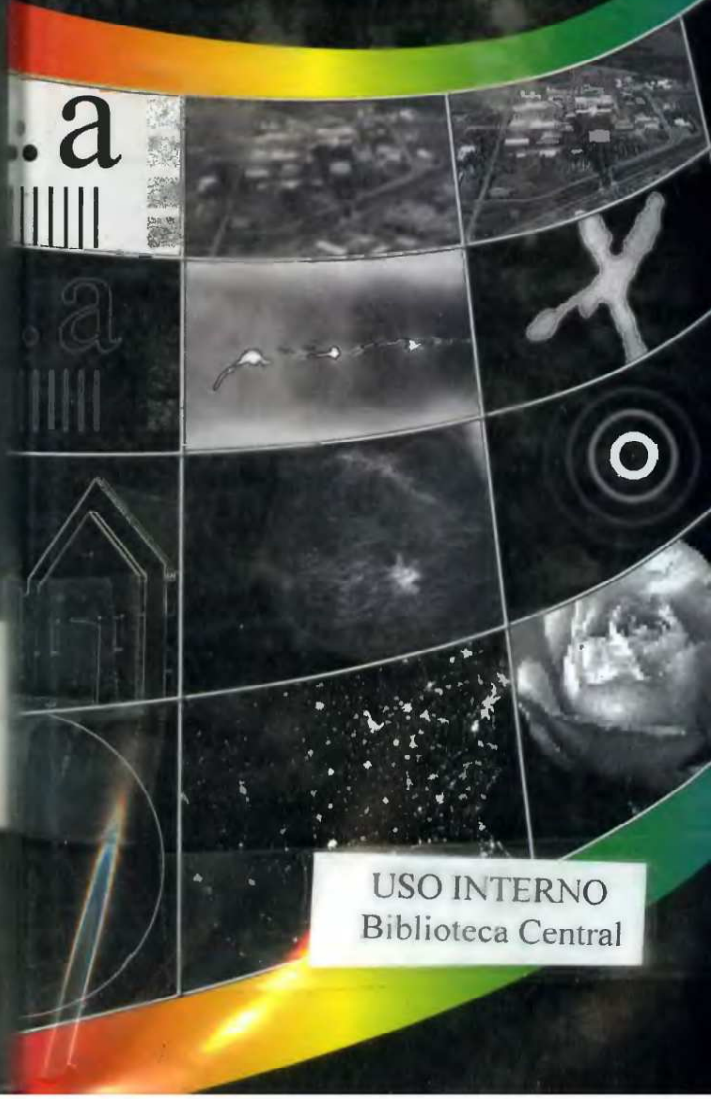
621.3
G643
2002

Second Edition

Prentice Hall

Digital Image Processing

Second Edition



USO INTERNO
Biblioteca Central