CHAPTER

Feature Generation II

7

7.1 INTRODUCTION

In the previous chapter we dealt with the task of feature generation via linear or nonlinear transformation techniques. This is just one of the possibilities available to the designer. There are a number of alternatives, however, that are very much application dependent. Although similarities among various applications do exist, there are also major differences. We will start by focusing on one major application area; that of *image analysis*. Clearly, we cannot review all techniques that have been suggested and used. Their number is really large. Instead, we will focus on basic directions, with a wide range of applications in mind, such as medical imaging, remote sensing, robot vision, and optical character recognition.

The major goal may be summarized as follows: *given an image, or a region within an image, generate the features that will subsequently be fed to a classifier in order to classify the image in one of the possible classes*. A digital (monochrome) image is usually the result of a discretization process (sampling) of a continuous image function $I(x, y)$ and is stored in the computer as a two-dimensional array $I(m, n)$ with $m = 0, 1, \ldots, N_x - 1$ and $n = 0, 1, \ldots, N_y - 1$. That is, it is stored as an $N_x \times N_y$ array. Every (m, n) element of the array corresponds to a *pixel* (picture element or image element) of the image, whose brightness or intensity is equal to $I(m, n)$. Furthermore, when the intensity $I(m, n)$ is quantized in N_g discrete (gray) levels N_g is known as the depth of the image. Then, the gray-level sequence $I(m, n)$ can take one of the integer values $0, 1, \ldots, N_g - 1$. The depth N_g is usually a power of 2 and can take large values (e.g., 64, 256) when the image is stored in the computer. However, for the human eye it is difficult to discern detailed intensity differences, and in practice $N_g = 32$ or 16 is a sufficient choice for image representation.

The need for feature generation stems from our inability to use the raw data. Even for a small 64×64 image the number of pixels is 4096. For most classification tasks this number is too large, raising computational as well as generalization problems, as discussed in earlier chapters. Feature generation is a procedure that computes new variables that in one way or another originate from the stored values

of the image array $I(m, n)$. *The goal is to generate features that exhibit high information-packing properties*, from the class separability point of view. Because we cannot use the raw data $I(m, n)$ directly, the features should encode efficiently the relevant information residing in the original data.

The other application area (discussed at the end of this chapter) is that of audio classification. Although some years ago image and audio analysis were considered to a large extent to be two scientific disciplines with different and distinct application areas, this is no longer the case. In a *multimedia* document its semantics are embedded in multiple forms that are usually complementary to each other. Thus, effective indexing for efficient handling (browsing, searching, manipulation, and information retrieval) requires a *multimodal* approach, in which either the most appropriate modality is selected or the different modalities are used in an *integrated* fashion. The visual modality contains everything that can be sensed by the eye (i.e., images that are naturally or artificially generated). The auditory modality contains the speech, music, and environmental sounds that can be heard in a video document. The focus of the last part of this chapter will be on typical features used to characterize and classify audio information. Some of the feature generation techniques can be considered common and can be applicable in both visual and audio modalities. On the other hand, a large number of features are the result of different approaches to exploit the specific nature of the signals and encode the required classification information in a more efficient way.

7.2 REGIONAL FEATURES

7.2.1 Features for Texture Characterization

The texture of an image region is determined by the way the gray levels are distributed over the pixels in this region. Although there is no clear definition of "texture," we are all in a position to describe an image by the look of it as *fine or coarse, smooth or irregular, homogeneous or inhomogeneous*, and so forth. Our goal in this subsection is to generate appropriate features that somehow quantify these properties of an image region. These features will emerge by exploiting space relations underlying the gray-level distribution.

First-Order Statistics Features

Let I be the random variable representing the gray levels in the region of interest. The first-order histogram $P(I)$ is defined as

$$P(I) = \frac{\textit{number of pixels with gray-level } I}{\textit{total number of pixels in the region}} \tag{7.1}$$

That is, $P(I)$ is the fraction of pixels with gray-level I. Let N_g be the number of possible gray levels. Based on (7.1), the following quantities are defined.

Moments:

$$m_i = E[I^i] = \sum_{I=0}^{N_g-1} I^i P(I), \quad i = 1, 2, \ldots \tag{7.2}$$

Obviously $m_0 = 1$ and $m_1 = E[I]$, the mean value of I.

Central moments:

$$\mu_i = E[(I - E[I])^i] = \sum_{I=0}^{N_g-1} (I - m_1)^i P(I) \tag{7.3}$$

The most frequently used central moments are μ_2, μ_3, and μ_4. $\mu_2 = \sigma^2$ is the variance, and μ_3 is known as the *skewness* (sometimes and is normalized by σ^3) and μ_4 as the *kurtosis* (sometimes is normalized by σ^4) of the histogram. The variance is a measure of the histogram width, that is, a measure of how much the gray levels differ from the mean. Skewness is a measure of the degree of histogram asymmetry around the mean, and μ_4 is a measure of the histogram sharpness. Depending on the value of μ_4, the resulting histogram is called platykurtic, for large values, leptokurtic, for small values, and mesokurtic otherwise. The normal distribution is a mesokurtic one. Figure 7.1 shows six variations of the same image (with 16 gray levels) with their corresponding histograms. We can observe the difference between the platykurtic and the leptokurtic one. In the latter, only the middle gray levels are present (with no $I = 0$ or $I = 15$), in contrast to the platykurtic one, where all gray levels are present. For the two asymmetric cases, one corresponds to a majority of low gray levels and the other to a majority of high levels. The resulting values of μ_3

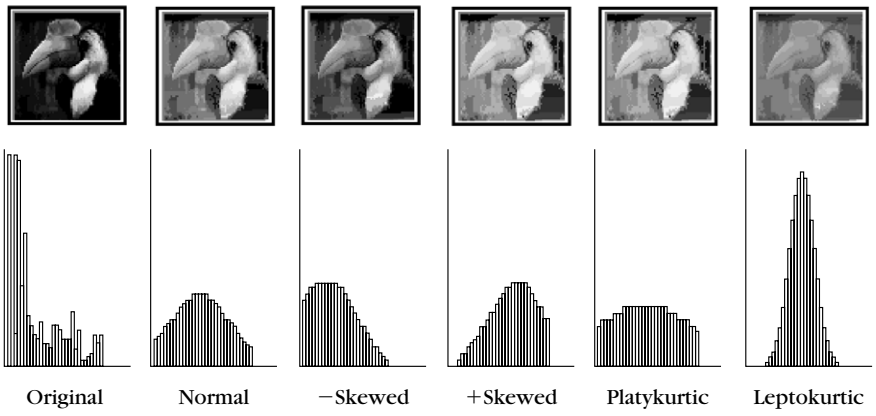


FIGURE 7.1

Examples of images and corresponding histograms.

and μ_4 from left to right are

$$\begin{array}{lcccccc} \mu_3: & 587 & 0 & -169 & 169 & 0 & 0 \\ \mu_4: & 16609 & 7365 & 7450 & 7450 & 9774 & 1007 \end{array}$$

Other quantities that result from the first-order histogram are:

Absolute moments:

$$\hat{\mu}_i = E[\mid I - E[I] \mid^i] = \sum_{I=0}^{N_g-1} \mid I - E[I] \mid^i P(I) \tag{7.4}$$

Entropy:

$$H = -E[\log_2 P(I)] = -\sum_{I=0}^{N_g-1} P(I) \log_2 P(I) \tag{7.5}$$

Entropy is a measure of histogram uniformity. The closer to the uniform distribution ($P(I)$ = constant), the higher the H. For the six images of Figure 7.1 the corresponding values are

$$H: \quad 4.61 \quad 4.89 \quad 4.81 \quad 4.81 \quad 4.96 \quad 4.12$$

Second-Order Statistics Features—Co-occurrence Matrices

The features resulting from the first-order statistics provide information related to the gray-level distribution of the image, but they do not give any information about the relative positions of the various gray levels within the image. Are all low-value gray levels positioned together, or are they interchanged with the high-value ones? This type of information can be extracted from the second-order histograms, where the pixels are considered in pairs. Two more parameters now enter into the scene. These are the relative distance among the pixels and their relative orientation. Let d be the relative distance measured in pixel numbers ($d = 1$ for neighboring pixels, etc.). The orientation ϕ is quantized in four directions: horizontal, diagonal, vertical, and antidiagonal (0°, 45°, 90°, 135°), as shown in Figure 7.2. For each combination of d and ϕ a two-dimensional histogram is defined

$$0^\circ: \quad P\big(I(m,n) = I_1,\ I(m \pm d, n) = I_2\big) \tag{7.6}$$

$$= \frac{\textit{number of pairs of pixels at distance d with values } (I_1, I_2)}{\textit{total number of possible pairs}}$$

In a similar way

$$45^\circ: P\big(I(m,n) = I_1, I(m \pm d, n \mp d) = I_2\big)$$

$$90^\circ: P\big(I(m,n) = I_1, I(m, n \mp d) = I_2\big)$$

$$135^\circ: P\big(I(m,n) = I_1, I(m \pm d, n \pm d) = I_2\big)$$

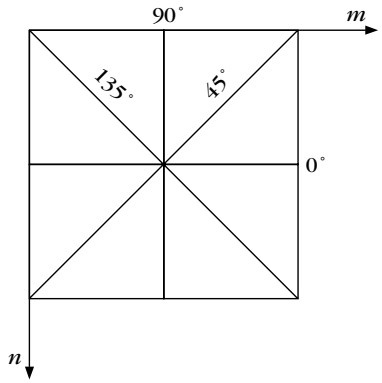


FIGURE 7.2

The four orientations used to construct co-occurrence matrices.

For each of these histograms an array is defined, known as the *co-occurrence or spatial dependence matrix*. Let, for example, an image array $I(m, n)$ be

$$I = \begin{bmatrix} 0 & 0 & 2 & 2 \\ 1 & 1 & 0 & 0 \\ 3 & 2 & 3 & 3 \\ 3 & 2 & 2 & 2 \end{bmatrix} \tag{7.7}$$

which corresponds to a 4×4 image. We have also assumed that $N_g = 4$ ($I(m, n) \in \{0, 1, 2, 3\}$). The co-occurrence matrix for a pair (d, ϕ) is defined as the $N_g \times N_g$ matrix

$$A = \frac{1}{R} \begin{bmatrix} \eta(0,0) & \eta(0,1) & \eta(0,2) & \eta(0,3) \\ \eta(1,0) & \eta(1,1) & \eta(1,2) & \eta(1,3) \\ \eta(2,0) & \eta(2,1) & \eta(2,2) & \eta(2,3) \\ \eta(3,0) & \eta(3,1) & \eta(3,2) & \eta(3,3) \end{bmatrix}$$

where $\eta(I_1, I_2)$ is the number of pixel pairs, at relative position (d, ϕ), which have gray-level values I_1, I_2, respectively. R is the total number of possible pixel pairs. Hence $\frac{1}{R}\eta(I_1, I_2) = P(I_1, I_2)$. For the image of (7.7) and relative pixel position $(1, 0°)$ we have

$$A^0(d = 1) = \frac{1}{24} \begin{bmatrix} 4 & 1 & 1 & 0 \\ 1 & 2 & 0 & 0 \\ 1 & 0 & 6 & 3 \\ 0 & 0 & 3 & 2 \end{bmatrix}$$

In words, for each of the intensity pairs, such as $(0, 0)$, we count the number of pixel pairs at relative distance $d = 1$ and orientation $\phi = 0°$ that take these values.

For our example this is 4. Two of them result from searching in the positive direction and two in the negative. According to the definition (7.6), these pixel pairs have coordinates (m, n) and $(m \pm 1, n)$ and gray levels $I_1 = 0, I_2 = 0$. The total number of pixel pairs for this case is 24. Indeed, for each row there are $N_x - 1$ pairs and there are N_y rows. Thus, the total number *for both positive and negative directions* is $2(N_x - 1)N_y = 2(3 \times 4) = 24$. For the diagonal direction 45° and $d = 1$ for each row we have $2(N_x - 1)$ pairs, except the first (or last) one, for which no pairs exist. Thus, the total number is $2(N_x - 1)(N_y - 1) = 2(3 \times 3) = 18$. For $d = 1$ and 90° we have $2(N_y - 1)N_x$ pairs, and finally for $d = 1$ and 135° $2(N_x - 1)(N_y - 1)$. For our example image and $(d = 1,\ \phi = 45°)$, we obtain

$$A^{45}(d = 1) = \frac{1}{18}\begin{bmatrix} 0 & 1 & 2 & 1 \\ 1 & 0 & 1 & 1 \\ 2 & 1 & 0 & 3 \\ 1 & 1 & 3 & 0 \end{bmatrix}$$

From the definition of the co-occurrence matrix, it is apparent that it is a symmetric one, something that can be used to reduce subsequent computations.

Having defined the probabilities of occurrence of gray levels with respect to relative spatial pixel position, we will go ahead to define the corresponding features. Some of them have a direct physical interpretation with respect to texture, for example, to quantify coarseness, smoothness, and so on. On the other hand, others do not possess such a property, but they still encode texture-related information with high discriminatory power.

- *Angular second moment*

$$ASM = \sum_{i=0}^{N_g-1}\sum_{j=0}^{N_g-1}(P(i,j))^2 \tag{7.8}$$

This feature is a measure of the smoothness of the image. Indeed, if all pixels are of the same gray-level $I = k$, then $P(k, k) = 1$ and $P(i, j) = 0, i \neq k$ or $j \neq k$, and $ASM = 1$. At the other extreme, if we could have all possible pairs of gray levels with equal probability $\frac{1}{R}$, then $ASM = \frac{R}{R^2} = \frac{1}{R}$. The less smooth the region is, the more uniformly distributed $P(i, j)$ and the lower the ASM (Problem 7.5).

- *Contrast*

$$CON = \sum_{n=0}^{N_g-1} n^2 \left\{ \sum_{\substack{i=0 \\ |i-j|=n}}^{N_g-1} \sum_{j=0}^{N_g-1} P(i,j) \right\} \tag{7.9}$$

This is a measure of the image contrast—that is, a measure of local gray-level variations. Indeed, $\sum_i \sum_j P(i,j)$ is the percentage of pixel pairs whose intensity differs by n. The n^2 dependence weighs the big differences more; thus, *CON* takes high values for images of high contrast.

- *Inverse difference moment*

$$IDF = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} \frac{P(i,j)}{1+(i-j)^2} \tag{7.10}$$

This feature takes high values for low-contrast images due to the inverse $(i-j)^2$ dependence.

- *Entropy*

$$H_{xy} = -\sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} P(i,j) \log_2 P(i,j) \tag{7.11}$$

Entropy is a measure of randomness and takes low values for smooth images.

These features are only a few from a larger set that can be derived. In the classical [Hara 73] paper, fourteen of those are summarized. They are repeated in Table 7.1. $P_x(P_y)$ (and related quantities) refer to the statistics with respect to the $x(y)$-axis. All features in the table are functions of the distance d and the orientation ϕ. Thus, if an image is rotated, the values of the features will be different. In practice, for each d the resulting values for the four directions are averaged out. In this way, we make these textural features *rotation tolerant*.

Besides the previous list of features, a number of other statistics-related features have been proposed. For example, in [Tamu 78] textural features are generated with an emphasis on the human visual perception. A set of features is suggested corresponding to texture coarseness, contrast, regularity, and so on. In [Davi 79] features based on a generalized definition of co-occurrence matrices are suggested, which are more appropriate for textures with long scale variations (macrotextures). An extensive treatment of texture is given in [Petr 06].

Example 7.1

Figure 7.3 shows two texture images, one coarse, known as grass [Brod 66], and the other smooth. Table 7.2 summarizes the values of some of the features for both of them.

Features Using Gray-Level Run Lengths

A gray-level *run* is a set of *consecutive* pixels having the *same gray-level value*. The *length of the run* is the number of pixels in the run [Gall 75, Tang 98]. Run length features encode textural information related to the number of times each gray-level,

Table 7.1 Features for Texture Characterization

Angular Second Moment:

$$f_1 = \sum_i \sum_j (P(i,j))^2$$

Contrast:

$$f_2 = \sum_{n=0}^{N_g-1} n^2 \left\{ \sum_i \sum_{\substack{j \\ |i-j|=n}} P(i,j) \right\}$$

Correlation:

$$f_3 = \frac{\{\sum_i \sum_j (ij) P(i,j)\} - \mu_x \mu_y}{\sigma_x \sigma_y}$$

Variance:

$$f_4 = \sum_i \sum_j (i-\mu)^2 P(i,j)$$

Inverse Difference Moment:

$$f_5 = \sum_i \sum_j \frac{P(i,j)}{1+(i-j)^2}$$

Sum (Difference) Average:

$$f_6(\hat{f}_6) = \sum_{i=0}^{2N_x-2\ (N_g-1)} i P_{x+(-)y}(i)$$

Sum Variance:

$$f_7 = \sum_{i=0}^{2N_g-2} (i-f_6)^2 P_{x+y}(i)$$

Sum Entropy:

$$f_8 = -\sum_{i=0}^{2N_g-2} P_{x+y}(i) \log P_{x+y}(i)$$

Entropy:

$$f_9 = -\sum_i \sum_j P(i,j) \log P(i,j) \equiv H_{xy}$$

Difference Variance:

$$f_{10} = \sum_{i=0}^{N_g-1} (i-\hat{f}_6)^2 P_{x-y}(i)$$

Difference Entropy:

$$f_{11} = -\sum_{i=0}^{N_g-1} P_{x-y}(i) \log P_{x-y}(i)$$

Information Measure I:

$$f_{12} = \frac{H_{xy} - H_{xy}^1}{max\{H_x, H_y\}}$$

Information Measure II:

$$f_{13} = \sqrt{1 - \exp(-2(H_{xy}^2 - H_{xy}))}$$

Maximal Correlation Coefficient:

$$f_{14} = (\text{2nd largest eigenvalue of } Q)^{\frac{1}{2}}$$

Definitions: $Q(i,j) = \sum_k \frac{P(i,k)P(j,k)}{P_x(i)P_y(k)}$

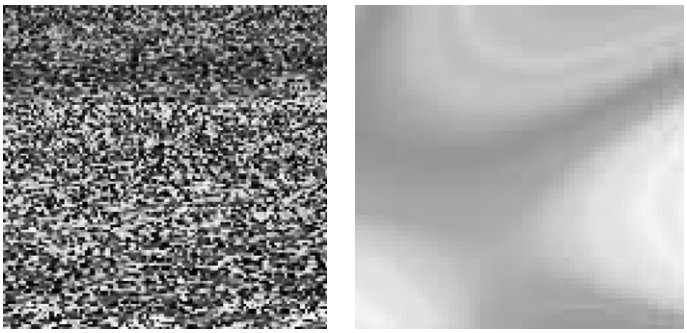
$$H_{xy}^1 = -\sum_i \sum_j P(i,j) \log(P_x(i)P_y(j))$$

$$H_{xy}^2 = -\sum_j \sum_i P_x(i)P_y(j) \log(P_x(i)P_y(j))$$

$$P_x(i) = \sum_j P(i,j) \qquad P_y(j) = \sum_i P(i,j)$$

$$P_{x\pm y}(k) = \sum_i \sum_{j,|i\pm j|=k} P(i,j)$$

$\mu, \mu_x, \mu_y, \sigma_x, \sigma_y; H_x, H_y$ means, st. deviations and entropies.



(a) (b)

FIGURE 7.3
Examples of (a) coarse and (b) smooth images.

Table 7.2 Second-Order Histogram Features for the Two Images of Figure 7.3

	Coarse	Smooth
ASM	0.0066	0.0272
CON	989.5	0.613
IDF	0.117	0.783
H_{xy}	8.352	5.884

for example, "1," appears in the image by itself, the number of times it appears in pairs, and so on. Take, for example, the image

$$I = \begin{bmatrix} 0 & 0 & 2 & 2 \\ 1 & 1 & 0 & 0 \\ 3 & 2 & 3 & 3 \\ 3 & 2 & 2 & 2 \end{bmatrix}$$

with four possible levels of gray ($N_g = 4$). For each of the four directions (0°, 45°, 90°, 135°) we define the corresponding run length matrix Q_{RL}. Its (i,j) element gives the number of times a gray-level $i - 1, i = 1, \ldots, N_g$, appears in the image with run length $j, j = 1, 2, \ldots, N_r$. This is an $N_g \times N_r$ array, where N_r is the largest possible run length in the image. For 0° we obtain

$$Q_{RL}(0°) = \begin{bmatrix} 0 & 2 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 2 & 1 & 0 & 0 \end{bmatrix} \tag{7.12}$$

The first element of the first row of the matrix is the number of times gray-level "0" appears by itself (0 for our example), the second element is the number of times it appears in pairs (2 in the example), and so on. The second row provides the same information for gray-level "1" and so on. For the 45° direction we have

$$Q_{RL}(45^\circ) = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 6 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 \end{bmatrix} \tag{7.13}$$

Based on the preceding definition of the run length matrix, the following features are defined.

- *Short-run emphasis*

$$SRE = \frac{\sum_{i=1}^{N_g} \sum_{j=1}^{N_r} (Q_{RL}(i,j)/j^2)}{\sum_{i=1}^{N_g} \sum_{j=1}^{N_r} Q_{RL}(i,j)} \tag{7.14}$$

  The denominator is the total number of run lengths in the matrix, 9 for (7.12) and 16 for (7.13). This feature emphasizes small run lengths, due to the division by j^2.

- *Long-run emphasis*

$$LRE = \frac{\sum_{i=1}^{N_g} \sum_{j=1}^{N_r} (Q_{RL}(i,j)j^2)}{\sum_{i=1}^{N_g} \sum_{j=1}^{N_r} Q_{RL}(i,j)} \tag{7.15}$$

  This gives emphasis to long-run lengths. Thus, we expect *SRE* to be large for coarser and *LRE* to be large for smoother images.

- *Gray-level nonuniformity*

$$GLNU = \frac{\sum_{i=1}^{N_g} \left[\sum_{j=1}^{N_r} Q_{RL}(i,j)\right]^2}{\sum_{i=1}^{N_g} \sum_{j=1}^{N_r} Q_{RL}(i,j)} \tag{7.16}$$

  The term in the brackets is the total number of run lengths for each gray-level. Large run length values contribute a great deal because of the square. When runs are uniformly distributed among the gray levels, *GNLU* takes small values.

- *Run length nonuniformity*

$$RLN = \frac{\sum_{j=1}^{N_r} \left[\sum_{i=1}^{N_g} Q_{RL}(i,j)\right]^2}{\sum_{i=1}^{N_g} \sum_{j=1}^{N_r} Q_{RL}(i,j)} \tag{7.17}$$

  In a similar way, *RLN* is a measure of run length nonuniformity.

Table 7.3 Run Length Features for the Images of Figure 7.3

	Coarse	Smooth
SRE	0.932	0.563
LRE	1.349	16.929
GLNU	255.6	71.6
RLN	3108	507
RP	0.906	0.4

- *Run percentage*

$$RP = \frac{\sum_{i=1}^{N_g} \sum_{j=1}^{N_r} Q_{RL}(i,j)}{L} \tag{7.18}$$

where L is the total possible number of runs in the image, if *all* runs had length equal to one, that is, the total number of pixels. *RP* takes low values for smooth images.

Example 7.2

For the two images of Figure 7.3 the values of Table 7.3 have resulted.

7.2.2 Local Linear Transforms for Texture Feature Extraction

Second-order statistics features were introduced in order to exploit the spatial dependencies that characterize the texture of an image region. We will now focus on an alternative possibility, which has been used extensively in practice. Let us consider a neighborhood of size $N \times N$ centered at pixel location (m, n). Let $\boldsymbol{x}_{mn}$ be the vector with elements the N^2 points within the area, arranged in a row-by-row mode. A *local linear transform or local feature extractor* is defined as

$$\boldsymbol{y}_{mn} = A^T \boldsymbol{x}_{mn} \equiv \begin{bmatrix} \boldsymbol{a}_1^T \\ \boldsymbol{a}_2^T \\ \vdots \\ \boldsymbol{a}_{N^2}^T \end{bmatrix} \boldsymbol{x}_{mn} \tag{7.19}$$

The respective correlation matrices are related via the $N^2 \times N^2$ nonsingular transformation matrix A as

$$R_y \equiv E[\boldsymbol{y}_{mn}\, \boldsymbol{y}_{mn}^T] = A^T R_x A \tag{7.20}$$

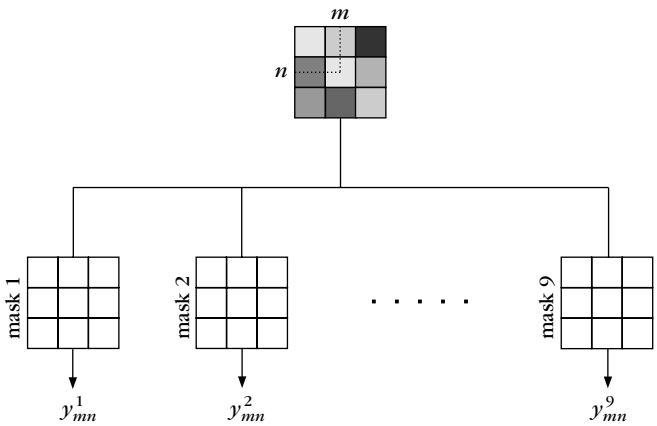


FIGURE 7.4

Filtering the image with each of the masks results in new transformed images/channels.

From these definitions it is readily seen that each element of $\boldsymbol{y}$ contains information about *all* the elements of $\boldsymbol{x}$. This becomes clearer if we look more closely at the way the two correlation matrices are related. Indeed, the diagonal elements of R_y are the respective variances of the elements of $\boldsymbol{y}$. These are first-order statistics quantities, yet their values contain information about the spatial dependencies (second-order statistics) of the original image. *Here lies the essence of the technique. Texture-related spatial dependencies of an image can be accommodated in the first-order statistics of the transformed image*. Using appropriately defined local transform matrices, various aspects of texture properties can be extracted. Of course, the philosophy does not change if instead of transforming vectors we use two-dimensional (separable) transforms of the corresponding subimage region.

One way to look at (7.19) is to interpret it as a series of N^2 filtering operations (convolutions, Appendix D), with a common input vector, $\boldsymbol{x}_{mn}$, that is, the $N \times N$ subimage centered at (m, n). The elements of $\boldsymbol{y}_{mn}$ are the respective filter output samples. This is illustrated in Figure 7.4, where the $N \times N$ subimage ($N = 3$) is filtered through 9 equivalent two-dimensional filters, each characterized by a different coefficient matrix, known as *mask*. In [Laws 80] it is suggested that the corresponding masks be constructed from three basic vectors, namely, $[1, 2, 1]^T$, $[-1, 0, 1]^T$, $[-1, 2, -1]^T$, for $N = 3$. The first corresponds to a local averaging operator, the second to an edge detection operator, and the third to a spot detector. These form a complete (nonorthogonal) set of vectors in the $\mathcal{R}^3$ space. The respective nine masks are formed by their cross-products, that is,

$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & 2 & -1 \\ -2 & 4 & -2 \\ -1 & 2 & -1 \end{bmatrix}$$

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & -2 & 1 \\ 0 & 0 & 0 \\ -1 & 2 & -1 \end{bmatrix}$$

$$\begin{bmatrix} -1 & -2 & -1 \\ 2 & 4 & 2 \\ -1 & -2 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 \\ -2 & 0 & 2 \\ 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

Each element of the vector $\boldsymbol{y}_{mn}$ is the result of filtering the local image neighborhood centered at (m, n) with each of the masks. By moving the masks around at the various (m, n) positions, nine different images, *channels*, will be obtained, *each encoding different aspects of the texture of the original image.* First-order-statistics quantities, such as variance and kurtosis, computed from each of these images, can then be used as features for texture classification. Masks larger than 3×3 have also been used. In some cases, an attempt to optimize the masks has been made, so that the resulting variances of the channels for the different classes are as different as possible [Unse 86]. This turns out to be an eigenvalue–eigenvector task, similar to the ones we have already met in Chapter 5. A comparative study of a number of optimal or suboptimal local linear transforms, including orthogonal ones, such as DCT, DST, and Karhunen-Loève, is given in [Unse 86, Unse 89, Rand 99]. Finally, it must be pointed out that all these techniques are closely related to the Gabor filtering approach of the previous chapter.

7.2.3 Moments

Geometric Moments

Let $I(x, y)$ be a continuous image function. Its *geometric moment* of order $p + q$ is defined as

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q I(x, y)\, dx\, dy \tag{7.21}$$

Geometric moments provide rich information about the image and are popular features for pattern recognition. Their information content stems from the fact that moments provide an equivalent representation of an image, in the sense that an image can be reconstructed from its moments (of all orders) [Papo 91, p. 115]. Thus, each moment coefficient conveys a certain amount of the information residing in an image.

It is by now commonplace to state that a desirable property in pattern recognition is invariance in geometric transformations. Moments, as defined in (7.21), depend on the coordinates of the object of interest within an image; thus, they lack the invariance property. This problem can be circumvented by defining appropriate combinations of normalized versions of the moments. Specifically, our goal will be to define moments that are invariant to:

Translations:

$$x' = x + a, \quad y' = y + b$$

Scaling:

$$x' = \alpha x, \quad y' = \alpha y$$

Rotations:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

To this end, let us define

Central moments:

$$\mu_{pq} = \int\int I(x,y)(x-\bar{x})^p(y-\bar{y})^q \, dx \, dy \tag{7.22}$$

where

$$\bar{x} = \frac{m_{10}}{m_{00}}, \quad \bar{y} = \frac{m_{01}}{m_{00}}$$

Central moments are invariant to translations.

Normalized central moments:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}}, \quad \gamma = \frac{p+q+2}{2} \tag{7.23}$$

These are easily shown to be invariant to both translation and scaling (Problem 7.6).

The Seven Moments of Hu

Hu [Hu 62] has defined a set of seven moments that are invariant under the actions of translation, scaling, and rotation. These are

$$\begin{aligned}
p+q &= 2 \\
\phi_1 &= \eta_{20} + \eta_{02} \\
\phi_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\
p+q &= 3 \\
\phi_3 &= (\eta_{30} - 3\eta_{12})^2 + (\eta_{03} - 3\eta_{21})^2 \\
\phi_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{03} + \eta_{21})^2 \\
\phi_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\
&\quad + (\eta_{03} - 3\eta_{21})(\eta_{03} + \eta_{21})[(\eta_{03} + \eta_{21})^2 - 3(\eta_{12} + \eta_{30})^2] \\
\phi_6 &= (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{03} + \eta_{21})
\end{aligned}$$

$$\phi_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2]$$
$$+ (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[(\eta_{03} + \eta_{21})^2 - 3(\eta_{30} + \eta_{12})^2]$$

The first six of these moments are also invariant under the action of reflection, while ϕ_7 changes sign. The values of these quantities can be quite different. In practice, in order to avoid precision problems, the logarithms of their absolute values are usually used as features. A number of other moment-based features that are invariant to more general transformations have also been proposed [Reis 91, Flus 93, Flus 94]. The case of moment invariants in the general l-dimensional space is treated in [Mami 98].

For a digital image $I(i,j)$, with $i = 0, 1, \ldots, N_x - 1, j = 0, 1, \ldots, N_y - 1$, the preceding moments can be *approximated* by replacing integrals by summations,

$$m_{pq} = \sum_i \sum_j I(i,j) i^p j^q \tag{7.24}$$

In order to keep the dynamic range of the moment values consistent for different-sized images, normalization of the $x - y$ axis can be performed, prior to computation of the moments. The moments are then approximated by

$$m_{pq} = \sum_i I(x_i, y_i) x_i^p y_i^q \tag{7.25}$$

where the sum is over all image pixels. Then x_i, y_i are the coordinates of the center point of the ith pixel and are no longer integers but real numbers in the interval $x_i \in [-1, +1]$, $y_i \in [-1, +1]$. *For digital images, the invariance properties of the moments we have defined are only approximately true.* An analysis in [Liao 96] reveals that the approximation error increases with the coarseness of the sampling grid as well as with the order of the moments.

Example 7.3

Figure 7.5 shows the Byzantine music symbol known as "petasti," resulting from a scanner, in scaled and various rotated versions. From left to right in the clockwise sense we have the original version, the scaled, the mirrored, and the rotated by 15°, 90°, and 180° versions, respectively.

Table 7.4 shows the resulting Hu moments for each of the version. The (approximate) invariance of the moments is apparent. Note the minus sign in ϕ_7 for the reflected (mirror) version.

Zernike Moments

The geometric moments defined in (7.21) can also be viewed as projections (Chapter 6) of $I(x,y)$ on the basis functions formed by the monomials $x^p y^q$. These monomials are not orthogonal; thus, the resulting geometric moment features are not optimal from an information redundancy point of view. In this subsection we will derive moments based on alternative complex polynomial functions, known as

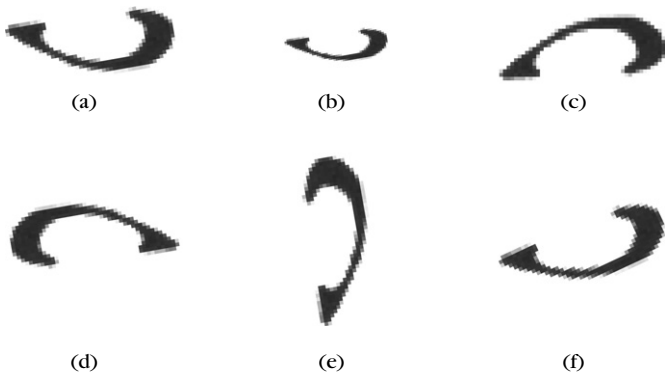


FIGURE 7.5

The Byzantine symbol "petasti" in various scaled and rotated versions, from (a) to (f).

Table 7.4 The Invariant Moments of Hu for the Versions of the "Petasti" Symbol

Moments	0°	Scaled	180°	15°	Mirror	90°
ϕ_1	93.13	91.76	93.13	94.28	93.13	93.13
ϕ_2	58.13	56.60	58.13	58.59	58.13	58.13
ϕ_3	26.70	25.06	26.70	27.00	26.70	26.70
ϕ_4	15.92	14.78	15.92	15.83	15.92	15.92
ϕ_5	3.24	2.80	3.24	3.22	3.24	3.24
ϕ_6	10.70	9.71	10.70	10.57	10.70	10.70
ϕ_7	0.53	0.46	0.53	0.56	−0.53	0.53

Zernike polynomials. These form a *complete orthogonal set over the interior of the unit circle* $x^2 + y^2 \leq 1$ (Problem 7.7) and are defined as

$$V_{pq}(x, y) = V_{pq}(\rho, \theta) = R_{pq}(\rho) \exp(jq\theta)$$

where:

p is a nonnegative integer

q is an integer subject to the constraint $p - |q|$ even, $|q| \leq p$

$\rho = \sqrt{x^2 + y^2}$

$\theta = \tan^{-1} \frac{y}{x}$

$$R_{pq}(\rho) = \sum_{s=0}^{(p-|q|)/2} \frac{(-1)^s[(p-s)!]\rho^{p-2s}}{s!\left(\frac{p+|q|}{2}-s\right)!\left(\frac{p-|q|}{2}-s\right)!}$$

The Zernike moments of a function $I(x, y)$ are given by

$$A_{pq} = \frac{p+1}{\pi}\iint_{x^2+y^2\le 1} I(x,y)V^*(\rho,\theta)\,dx\,dy$$

where the $*$ denotes complex conjugation. For a digital image, the respective Zernike moments are computed as

$$A_{pq} = \frac{p+1}{\pi}\sum_i I(x_i,y_i)V^*(\rho_i,\theta_i), x_i^2+y_i^2 \le 1$$

where i runs over all the image pixels. The computation of the corresponding moments of an image considers the center of the image as the origin and pixels are mapped into the unit circle, that is, $x_i^2 + y_i^2 \le 1$. The pixels falling outside the unit circle are not taken into consideration. The magnitude of the Zernike moments is invariant to rotations [Teag 80] (Problem 7.8). Translation and scaling invariance is treated in [Khot 90a, Chon 03]. A drawback of the Zernike moments is the computational complexity associated with the computation of the radial polynomials. A common approach used in reducing complexity includes the application of recurrence relations between successive radial polynomials and coefficients. Computational aspects of the Zernike moments are examined in [Muku 95, Wee 06, Huan 06]. Numerical error issues associated with the computations of the Zernike moments are treated in [Sing 06]. Comparative studies of the performance of the Zernike moments against the moments of Hu, in the context of character recognition, have demonstrated that the former behave better, especially in noisy environments [Khot 90b]. In [Wang 98], Zernike moments are used to cope with both geometry and illumination invariance, in the context of multispectral texture classification. Variants of the Zernike moments, called pseudo-Zernike moments, have also been proposed and used. Comparative studies can be found in [Teh 88, Heyw 95]. Besides Zernike moments, other types of moments have also been suggested and used, such as the Fourier–Mellin moments and moments based on Legendre polynomials, as in [Kan 02, Chon 04, Muku 98].

7.2.4 Parametric Models

So far, in various parts of the book, we have treated the gray levels as random variables and looked at aspects of their first- and second-order statistics. In this subsection, their randomness will be approached from a different perspective. We will assume that $I(m, n)$ is a real *nondiscrete* random variable, and we will try *to model* its underlying generation mechanism by adopting an appropriate *parametric model*. The parameters of the resulting models encode useful information and

lend themselves as powerful feature candidates for a number of pattern recognition tasks.

We will move in two directions. One is to treat an image as a successive sequence of rows or columns. That is, our random variables will be considered as successive realization samples from a one-dimensional random process $I(n)$. The alternative looks at the image as a two-dimensional random process $I(m, n)$, also known as *random field*.

One-Dimensional Parametric Models

Let $I(n)$ denote the random sequence. We will assume that it is stationary in the wide sense. This means that its autocorrelation sequence $r(k)$ exists and is of the form

$$r(k) = E[I(n)I(n-k)]$$

and the Fourier transform of $r(k)$ also exists and is a *positive* function (power spectral density)

$$I(\omega) = \sum_{k=-\infty}^{+\infty} r(k)\exp(-j\omega k)$$

Under certain assumptions, which are met in practice most of the time [Papo 91, Theo 93], it can be shown that such a random sequence can be generated at the output of a linear, causal, stable, time-invariant system with impulse response $h(n)$, whose input is excited by a white noise sequence, as shown in Figure 7.6. In simple terms, this means that we can write

$$I(n) = \sum_{k=0}^{\infty} h(k)\eta(n-k)$$

where $h(n)$ satisfies the stability condition $\sum_n |h(n)| < \infty$. The sequence $\eta(n)$ is a white noise sequence, that is, $E[\eta(n)] = 0$ and $E[\eta(n)\eta(n-l)] = \sigma^2\delta(l) : \delta(l) = 1$ for $l = 0$ and zero otherwise. Such processes of a special type are the so-called *autoregressive processes (AR)*, which are generated by systems of the form

$$I(n) = \sum_{k=1}^{p} a(k)I(n-k) + \eta(n) \tag{7.26}$$

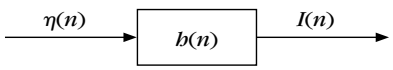


FIGURE 7.6

Generation model of a stationary random process at the output of a stable, linear, time-invariant system excited by a white noise sequence.

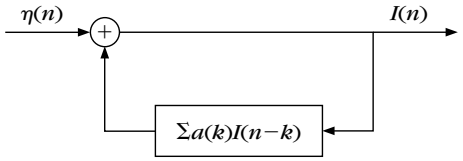


FIGURE 7.7

Generation model of an AR stationary random process.

In words, the random sequence $I(n)$ is given as a linear combination of previous samples $I(n-k)$ and the current input sample $\eta(n)$ (Figure 7.7). Here p is the order of the AR model, and we write AR(p). The coefficients $a(k)$, $k = 1, 2, \ldots, p$ are the AR model parameters. AR models are a special case of a more general class of models, known as *autoregressive–moving average (ARMA(p,m))*, for which

$$I(n) = \sum_{k=1}^{p} a(k)I(n-k) + \sum_{l=0}^{m} b(l)\eta(n-l) \tag{7.27}$$

That is, the model is regressive with respect to both input and output sequences. The major advantage of the AR models, compared with their ARMA relatives, is that the former lead to *linear* systems of equations for the estimation of the model parameters.

Estimation of the AR Parameters

Another way to look at (7.26) is to interpret the coefficients $a(k), k = 1, \ldots, p$, as the predictor parameters of the sequence $I(n)$. That is, the parameters weigh previous samples, $I(n-1), \ldots, I(n-p)$, in order to predict the value of the current sample $I(n)$, and $\eta(n)$ is the prediction error,

$$\hat{I}_n = \sum_{k=1}^{p} a(k)I(n-k) \equiv \boldsymbol{a}_p^T \boldsymbol{I}_p(n-1) \tag{7.28}$$

where $\boldsymbol{I}_p^T(n-1) \equiv [I(n-1), \ldots, I(n-p)]$. The unknown parameter vector $\boldsymbol{a}_p^T = [a(1), a(2), \ldots, a(p)]$ is optimally estimated, for example, by minimizing the mean square prediction error,

$$E[\eta^2(n)] = E[(I(n) - \hat{I}(n))^2] = E[(I(n) - \boldsymbol{a}_p^T \boldsymbol{I}_p(n-1))^2] \tag{7.29}$$

The problem is exactly the same as that of the mean square linear classifier estimation of Chapter 3, and the unknown parameters result from the solution of

$$E[\boldsymbol{I}_p(n-1)\boldsymbol{I}_p^T(n-1)]\boldsymbol{a}_p = E[I(n)\boldsymbol{I}_p(n-1)] \tag{7.30}$$

or

$$\begin{bmatrix} r(0) & r(-1) & \dots & r(-p+1) \\ r(1) & r(0) & \dots & r(-p+2) \\ \vdots & \vdots & \vdots & \vdots \\ r(p-2) & r(p-3) & \dots & r(-1) \\ r(p-1) & r(p-2) & \dots & r(0) \end{bmatrix} \begin{bmatrix} a(1) \\ a(2) \\ \vdots \\ a(p-1) \\ a(p) \end{bmatrix} = \begin{bmatrix} r(1) \\ r(2) \\ \vdots \\ r(p-1) \\ r(p) \end{bmatrix}$$

or

$$\boldsymbol{R}\boldsymbol{a}_p = \boldsymbol{r}_p \tag{7.31}$$

with $\boldsymbol{r}_p \equiv [r(1), \dots, r(p)]^T$. The relation of the optimal parameters $a(k)$ with the mean square error (variance of generating noise) is obtained from (7.29) and (7.31) and is given by

$$\sigma_\eta^2 = E[\eta^2(n)] = r(0) - \sum_{k=1}^{p} a(k)r(k) \tag{7.32}$$

The autocorrelation matrix has a computationally rich structure. It is symmetric ($r(k) = r(-k)$) and Toeplitz—that is, all the elements across any of its diagonals are the same. Exploitation of these properties leads to the development of a computationally efficient scheme for the solution of (7.31). This is *Levinson's algorithm*, which solves the linear system of equations in $O(p^2)$ multiplications and additions, as opposed to $O(p^3)$ required by more classical algorithmic schemes [Theo 93, Hayk 96]. In Chapter 3, we saw that when the autocorrelation sequence is not known, it is often preferable to adopt the least sum of squares instead of the mean square criterion. Then the AR parameters are still provided by a linear system of equations, but the associated matrix is no longer Toeplitz. However, it is still computationally rich, and Levinson-type $O(p^2)$ algorithms for the efficient solution of such systems have also been derived [Theo 93].

Besides images, AR (ARMA) models have been used extensively to model other type of random sequences, such as those resulting from digitizing speech signals and electroencephalographic signals. *For all these cases the resulting AR parameters can be used as features to classify one type of signal from another.*

Example 7.4

Let the AR random sequence of order $p = 2$ be

$$I(n) = \sum_{k=1}^{2} a(k)I(n-k) + \eta(k)$$

with $r(0) = 1$, $r(1) = 0.5$, $r(2) = 0.85$. Computing the mean square estimates of $a(k)$, $k = 1, 2$, we obtain

$$\begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix} \begin{bmatrix} a(1) \\ a(2) \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.85 \end{bmatrix}$$

and its solution gives $a(1) = 0.1$, $a(2) = 0.8$.

Two-Dimensional AR Models

A two-dimensional AR random sequence $I(m, n)$ is defined as

$$\hat{I}(m, n) = \sum_k \sum_l a(k, l) I(m - k, n - l), (k, l) \in W \tag{7.33}$$

$$I(m, n) = \hat{I}(m, n) + \eta(m, n) \tag{7.34}$$

Figure 7.8 shows the region W of the pixels that contribute to the prediction of $\hat{I}(m, n)$, for a number of possible choices. The case in Figure 7.8a corresponds to what is known as a *strongly causal predictor model*. This is because all pixels in the contributing area have coordinates smaller than the coordinates m, n of the predicted pixel, which is represented by the unshaded node in the figure. The corresponding window is $W_1 = \{0 \leq k \leq p, 0 \leq l \leq q, (k, l) \neq (0, 0)\}$. However, the notions of past and present have no real meaning for an image, and alternative windows can also be used. A *noncausal predictor* is defined as

$$I(m, n) = \sum_{k=-p}^{p} \sum_{l=-q}^{q} a(k, l) I(m - k, n - l) + \eta(m, n)$$

In Figure 7.8d the corresponding window is shown for the case of $p = q = 2$. Figure 7.8c shows a third possibility, which is known as a *semicausal predictor*, and Figure 7.8b shows the case of a *causal* predictor. Next we summarize the last three cases, which are the most common in practice:

$$\begin{aligned} \textit{Causal} &: W_2 = \{(-p \leq k \leq p, 1 \leq l \leq q) \cup (1 \leq k \leq p, l = 0)\} \\ \textit{Semicausal} &: W_3 = \{-p \leq k \leq p, 0 \leq l \leq q \ , \ (k, l) \neq (0, 0)\} \\ \textit{Noncausal} &: W_4 = \{-p \leq k \leq p, -q \leq l \leq q \ , \ (k, l) \neq (0, 0)\} \end{aligned}$$

AR Parameter Estimation

We have

$$\hat{I}(m, n) = \sum_k \sum_l a(k, l) I(m - k, n - l)$$

Recalling the orthogonality condition from Chapter 3, in its two-dimensional generalization, we obtain that the minimum mean square error solution satisfies

$$E\left[I(m - i, n - j) \left(I(m, n) - \sum_k \sum_l a(k, l) I(m - k, n - l) \right) \right] = 0, \quad (i, j) \in W \tag{7.35}$$

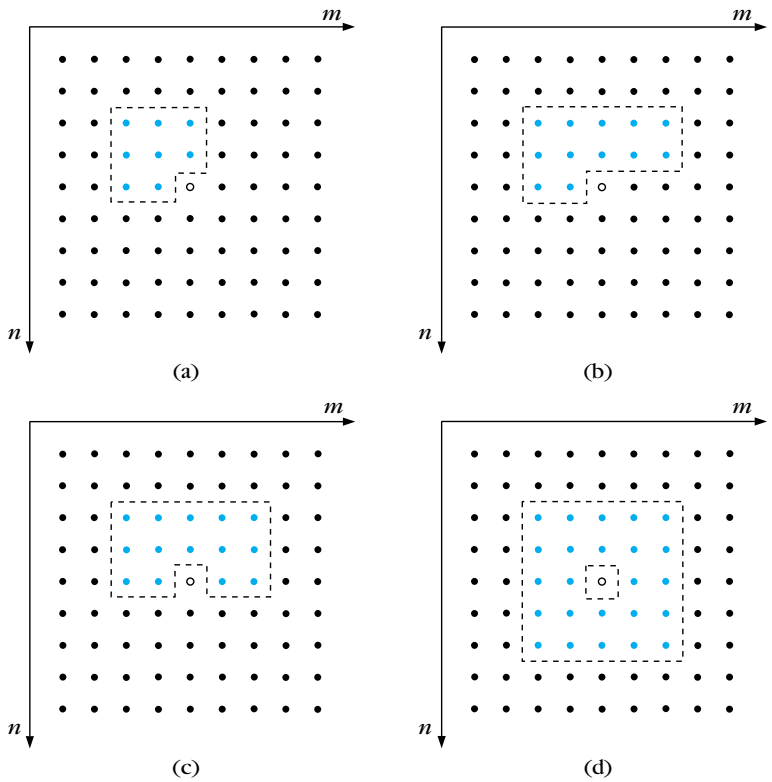


FIGURE 7.8

Different types of two-dimensional prediction models. The predicted pixel is represented by the unshaded node. The red pixels are those that take part in the prediction and the corresponding window W is the area enclosed by the dotted line. (a) Strictly causal, (b) causal, (c) semicausal, and (d) noncausal.

or

$$r(i,j) = \sum_k \sum_l a(k,l) r(i-k, j-l), \quad (i,j) \in W \tag{7.36}$$

where $r(i,j) \equiv E[I(m,n)I(m-i,n-j)]$ is the two-dimensional autocorrelation sequence of the random field $I(m,n)$. The set of equations in (7.36) constitutes a linear system of equations leading to the estimates of $a(k,l)$. The associated matrix also has a computationally rich structure, which can be exploited to develop efficient schemes to compute the solution. Let us take, for example, the noncausal window for $p = q$. This is a symmetric window, in the sense that for each index pair (i,j), the $(-i,-j)$ is also present. Combining (7.36) with the equation of the

variance of the minimum error, which is given by (Problem 7.11)

$$\sigma_\eta^2 = r(0,0) - \sum_k \sum_l a(k,l) r(k,l) \tag{7.37}$$

the following system results:

$$R\boldsymbol{a} = -\begin{bmatrix} \mathbf{0} \\ \sigma_n^2 \\ \mathbf{0} \end{bmatrix} \tag{7.38}$$

where $\mathbf{0}$ is the zero vector of appropriate dimension and

$$\boldsymbol{a}^T = [a(p,p), \ldots, a(p,-p), \ldots, a(0,0), \ldots, a(-p,p), \ldots, a(-p,-p)]$$

where $a(0,0) \equiv 1$ and R is the corresponding autocorrelation matrix. The dimension of $\boldsymbol{a}$ is $(2p+1)^2$. The correlation of a homogeneous (i.e., $E[I(m,n)I(m-k,n-l)] = r(k,l)$) and isotropic (no direction dependence) image depends only on the relative distance between pixels,

$$r(k,l) = r\left(\sqrt{k^2+l^2}\right)$$

and the resulting autocorrelation matrix is easily shown to be *symmetric and block Toeplitz with each block being itself a Toeplitz matrix*,

$$R = \begin{bmatrix} R_0 & R_1 & \ldots & R_{2p} \\ R_1 & R_0 & \ldots & R_{2p-1} \\ \vdots & \vdots & \vdots & \vdots \\ R_{2p} & R_{2p-1} & \ldots & R_0 \end{bmatrix} \tag{7.39}$$

where

$$R_i = \begin{bmatrix} r(i,0) & \ldots & r(i,2p) \\ \vdots & \vdots & \vdots \\ r(i,2p) & \ldots & r(i,0) \end{bmatrix} \tag{7.40}$$

For homogeneous images and symmetric windows it is easy to show that the AR parameters are symmetric $a(k,l) = a(-k,-l)$ and the system can be solved efficiently by a Levinson-type algorithm [Kalo 89]. If the image is homogeneous but anisotropic, the resulting system's associated matrix is block Toeplitz, but the elements are no longer Toeplitz. Furthermore, more general windows than the ones introduced in this section have also been suggested and used. Efficient Levinson-type algorithms for such cases have also been developed (e.g., [Glen 94]). Finally, besides the squared error criteria, maximum likelihood techniques can be employed for the estimation of the unknown parameters, which can lead to more accurate estimates. Of course, in such cases assumptions about the underlying statistics have to be adopted (e.g., [Kash 82]).

Remarks

- The AR modeling of images has been used in the classification context in a number of cases [Chel 85, Cros 83, Kash 82, Sark 97]. In [Kash 86, Mao 92] extensions have been proposed for rotation-invariant models.
- The AR random field models are related to a class of models known as *Markov random fields*. The essence of these fields is that for each pixel (m, n) the image is divided into three areas: Ω^+ ("future"), Ω ("present"), and Ω^- ("past"). It is then assumed that the random variable $I(m, n)$, $(m, n) \in \Omega^+$, is independent of its values in Ω^- and depends only on the values in Ω; thus, the conditional density function satisfies

$$p\left(I(m,n),(m,n) \in \Omega^+ | I(m,n),(m,n) \in \Omega^- \cup \Omega\right) = p\left(I(m,n),(m,n) \in \Omega^+ | I(m,n),(m,n) \in \Omega\right)$$

  In words, the "future" depends only on the "present" and not on the "past"; that is, the value of the random variable at a pixel depends on the values that the random variable takes in a specific (neighboring) area only, and it does not depend on the values in the remaining regions of the image.
- It can be shown that every Gaussian AR model is a Markov random field. [Wood 72, Chel 85].

Example 7.5

For an image whose autocorrelation sequence obeys

$$r(k, l) = 0.8^{\sqrt{k^2+l^2}}$$

estimate the AR parameters for a noncausal $p = q = 1$ window.

From the definition we have

$$\begin{aligned}\hat{I}(m,n) = {} & a(1,1)I(m-1,n-1) + a(1,0)I(m-1,n) \\ & + a(1,-1)I(m-1,n+1) + a(0,1)I(m,n-1) \\ & + a(0,-1)I(m,n+1) + a(-1,1)I(m+1,n-1) \\ & + a(-1,0)I(m+1,n) + a(-1,-1)I(m+1,n+1)\end{aligned}$$

The resulting matrix $\boldsymbol{R}$ is a block $(2p+1) \times (2p+1) = 3 \times 3$ matrix with elements the 3×3 matrices

$$R = \begin{bmatrix} R_0 & R_1 & R_2 \\ R_1 & R_0 & R_1 \\ R_2 & R_1 & R_0 \end{bmatrix}$$

where

$$R_0 = \begin{bmatrix} r(0,0) & r(0,1) & r(0,2) \\ r(0,1) & r(0,0) & r(0,1) \\ r(0,2) & r(0,1) & r(0,0) \end{bmatrix}$$

$$R_1 = \begin{bmatrix} r(1,0) & r(1,1) & r(1,2) \\ r(1,1) & r(1,0) & r(1,1) \\ r(1,2) & r(1,1) & r(1,0) \end{bmatrix}$$

$$R_2 = \begin{bmatrix} r(2,0) & r(2,1) & r(2,2) \\ r(2,1) & r(2,0) & r(2,1) \\ r(2,2) & r(2,1) & r(2,0) \end{bmatrix}$$

For this specific model the linear system in (7.38) has nine unknowns and the solution gives

$$a(1,1) = a(-1,-1) = -0.011, \quad a(1,0) = a(-1,0) = -0.25$$
$$a(1,-1) = a(-1,1) = -0.011, \quad a(0,1) = a(0,-1) = -0.25$$
$$\sigma_\eta^2 = 0.17$$

7.3 FEATURES FOR SHAPE AND SIZE CHARACTERIZATION

In a number of image analysis applications, an important piece of information is the shape and size of an object of interest within the image. For example, in medical applications the shape and size of nodules are crucial in classifying them as malignant, or benign. Nodules with an irregular boundary have a high probability of being malignant, and those with a more regular boundary are usually benign. Also, it has been observed that in certain cases nodules with a perimeter of more than 3 cm are usually malignant [Cavo 92].

Another example in which the shape of the object is of major importance is the automatic character recognition in an *optical character recognition (OCR)* system [Mori 92, Plam 00, Vinc 02]. Although OCR systems employing our already familiar regional features, there is a large class of techniques that use the shape information residing in the *boundary curve* of the characters.

Figure 7.9a shows the character "5" as seen from the scanner of an OCR system. An appropriate image segmentation algorithm (e.g., [Pita 94]) has first been applied to separate the character from the rest of the image. The character in Figure 7.9b is in binary form. This is the result of the binarization phase, in which all gray

(a) (b) (c)

FIGURE 7.9

The character "5" after (a) the segmentation of the scanned image and then (b) the application of a binarization algorithm and (c) its boundary after the application of a boundary extraction algorithm in the binarized version.

levels of the character region below a certain threshold become 0 and all above it become 1 [Trie 95]. Figure 7.9c shows the resulting boundary, after the application of a boundary extraction algorithm (e.g., [Pita 94]) on the binary version. Thus, in the last version there is no texture of interest inside the character. What is of paramount importance in such systems is feature invariance in geometric transformations. The recognition of the character must be insensitive to its position, size, and orientation. A review of various methodologies for invariant pattern recognition techniques can be found in [Wood 96].

The shape characterization of a region or an object can be achieved in various ways. Two are the major directions along which we will proceed. One is to develop techniques that provide a full description of the boundary of the object in a regenerative manner. In words, the boundary can be reobtained from the description coefficients, such as by using a Fourier expansion of the boundary, which in turn can be reconstructed from its Fourier coefficients. The other direction is to use features that are descriptive of the characteristics of the shape of the region but are not regenerative. Examples of such features are the number of corners in the boundary and the perimeter. They provide useful information about the boundary, but they are not sufficient to reproduce it. In the following we will focus on some basic techniques, which have in turn given birth to a large number of variants shaped to fit specific application requirements (for example, see [Trie 96] for a review).

7.3.1 Fourier Features

Let (x_k, y_k), $k = 0, 1, \ldots, N-1$, be the coordinates of N samples on the boundary of an image region, Figure 7.10a. For each pair (x_k, y_k) we define the complex variable

$$u_k = x_k + jy_k$$

For the N u_k points we obtain the DFT f_l

$$f_l = \sum_{k=0}^{N-1} u_k \exp\left(-j\frac{2\pi}{N}lk\right), \quad l = 0, 1, \ldots, N-1$$

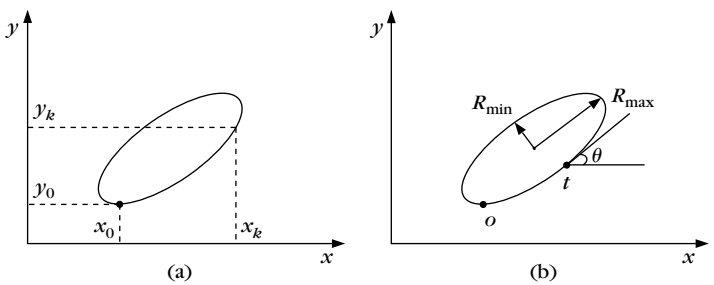


FIGURE 7.10

Boundary of an image region (a) and associated parameters (b).

The coefficients f_l are also known as the *Fourier descriptors* of the boundary. Once the f_l are available, the u_k can be recovered and the boundary can be reconstructed. However, our goal in pattern recognition is not to reconstruct the boundary. Thus, a smaller number of coefficients (or descriptors) is usually used, enough to include sufficient discriminatory information. In the sequel, we will investigate how the Fourier descriptors are affected by the actions of translation, rotation, and scaling. For translation we have

$$x'_k = x_k + \Delta x$$
$$y'_k = y_k + \Delta y$$

Then

$$u'_k = u_k + (\Delta x + j\Delta y) \equiv u_k + \Delta u'$$

For rotation, it is not difficult to verify that in rotating all points of the region by θ, with respect to the origin, the rotated coordinates correspond to (Problem 7.13)

$$u'_k = u_k \exp(j\theta)$$

If f_l, f'_l are the DFTs of u_k, u'_k, respectively, then from the DFT definition we get

$$\textit{Translation}: u'_k = u_k + \Delta u' \Longrightarrow f'_l = f_l + \Delta u' \delta(l)$$

$$\textit{Rotation}: u'_k = u_k \exp(j\theta) \Longrightarrow f'_l = f_l \exp(j\theta)$$

$$\textit{Scaling}: u'_k = a u_k \Longrightarrow f'_l = a f_l$$

$$\textit{Translation of the sampling origin}: u'_k = u_{k-k_0} \Longrightarrow$$

$$f'_l = f_l \exp\left(-j2\pi k_0 \frac{l}{N}\right)$$

In words, translation affects only the f_0' coefficient. Rotation affects *the phase* of all the coefficients by the *same factor*, and it has *no effect on their magnitude*. Scaling affects all coefficients in the same way, and thus it has no effect on the ratios $\frac{f_i}{f_j}$. The sampling point origin, within the boundary, affects the phase but leaves invariant the magnitude $|f_l|$.

This *deterministic* manner, in which the three geometric transformations affect the Fourier coefficients, allows the development of appropriate normalized versions that are invariant to these actions [Crim 82, Arbt 90, Gran 72]. Let us demonstrate the rationale of such approaches via an example, by considering the boundary of an object. The first decision to be taken, prior to the computation of the Fourier coefficients, is to define the first sampling point (x_0, y_0) on the boundary. In practice, the choice of this point for each character has a degree of randomness. The choice of a different sampling origin corresponds to a relative translation of, say, $k_0 < N$ samples (since the boundary is a closed curve, the relative

translation will always be $(k - k_0)$ modulo $N < N$). As we have seen earlier, this affects the Fourier descriptors

$$u'_k = u_{k-k_0} \Longrightarrow f_l' = f_l \exp\left(-j2\pi k_0 \frac{l}{N}\right) \tag{7.41}$$

hence

$$f_1' = f_1 \exp\left(-j2\pi \frac{k_0}{N}\right) \Longrightarrow f_1' = |f_1| \exp(-j\phi_1) \exp\left(-j2\pi \frac{k_0}{N}\right)$$

where $|f_1|, \phi_1$ are the magnitude and phase of f_1, respectively. Hence, the phase of f_1' is $\phi_1' = \phi_1 + 2\pi \frac{k_0}{N}$. In the sequel we define the following normalized Fourier coefficients:

$$\hat{f}_l = f_l \exp(jl\phi_1) \tag{7.42}$$

The corresponding normalized coefficient with shifted origin will be

$$\hat{f}'_l = f_l' \exp(jl\phi'_1) = f_l' \exp\left(jl\phi_1 + j2\pi k_0 \frac{l}{N}\right) \tag{7.43}$$

Taking into account (7.41), we obtain

$$\hat{f}'_l = \hat{f}_l$$

Thus, the preceding normalization generates features that are *invariant to the choice of the sampling origin* (x_0, y_0).

This method of exploiting the power of the Fourier transform as a tool for boundary description is not the only possibility. An alternative is to express the coordinates of the boundary contour points as functions of the boundary length t, measured from an origin within the boundary, that is, $(x(t), y(t))$. Since the boundary is a closed curve, these are periodic functions and they can be expanded in their Fourier series. Invariant versions of the Fourier coefficients can then be computed and used as features for pattern recognition [Kuhl 82, Lin 87]. Comparative performance studies of a number of invariant Fourier-based features, in the context of handwritten character recognition, can be found in [Pers 77, Taxt 90].

Another way is to generate Fourier descriptors from the curvature $k(t)$ function of the boundary, defined as

$$k(t) = \frac{d\theta(t)}{dt}$$

where $\theta(t)$ is the tangent angle (Figure 7.10b) at a point a distance t from the origin, which is marked "o" in the figure. Such a description is justified by Gauss's theorem, stating that every curvature function corresponds to one and only one curve in space (with the exception of its position in space). The advantage of such a description stems from its obvious scale invariance property. If we measure the length of the boundary at a point by the number of pixels n

between this point and the origin of the curve, the curvature of the boundary is approximated by

$$\begin{aligned} \theta_n &= \tan^{-1}\frac{y_{n+1}-y_n}{x_{n+1}-x_n}, \quad && n = 0, 1, \ldots, N-1 \\ k_n &= \theta_{n+1} - \theta_n, && n = 0, 1, \ldots, N-1 \end{aligned} \tag{7.44}$$

In the previous chapter we have seen that an alternative to Fourier descriptors is to use wavelet coefficients. However, as we pointed out there, the definition of invariant wavelet descriptors is not a straightforward task, and invariance is attempted via indirect methods.

7.3.2 Chain Codes

Chain coding is among the most widely used techniques for boundary shape description. In [Free 61], the boundary curve is approximated via a sequence of connected straight line segments of preselected direction and length. Every line segment is coded with a specific coding number depending on its direction.

In Figure 7.11 two possible choices, usually encountered in practice, are shown. In this way a *chain code* $[d_i]$ is created, where d_i is the coding number of the direction of the line segment that connects boundary pixel (x_i, y_i) with the next one (x_{i+1}, y_{i+1}), sweeping the boundary in, say, the clockwise sense. A disadvantage of this description is that the resulting chain codes are usually long and at the same time are very sensitive in the presence of noise. This leads to chain codes with variations due to noise and not necessarily to the boundary curve. A way out is to resample the boundary curve by selecting a grid of larger dimensions. For each of the boxes of the grid all points inside a box are assigned the value of the respective box center. In Figure 7.12a the original samples are shown alongside the larger sampling grid. Figure 7.12b is the resulting resampled version. The chain

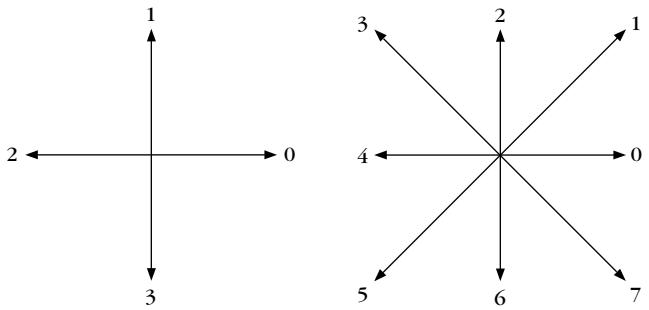


FIGURE 7.11

Directions for a (a) four-directional chain code and (b) an eight-directional chain code.

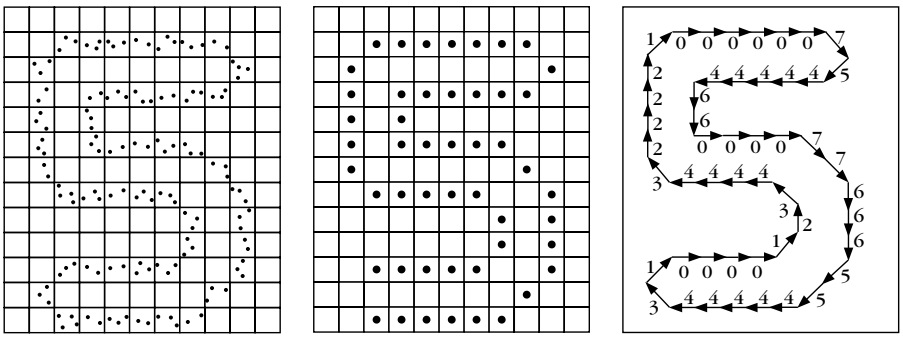


FIGURE 7.12

The character "5" and (a) its original sampled image, (b) its resampled version on a coarser grid, and (c) the resulting chain code.

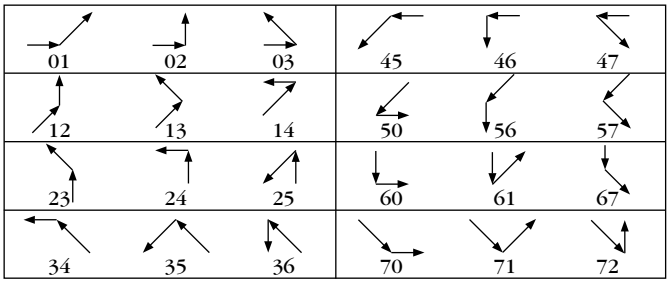
code is formed from the sequential connection of these pixels, Figure 7.12c. If we consider the length of the grid side as the basic measurement unit, then for even-coded directions, $0, 2, 4, 6$, the length of the corresponding straight line segment is 1, and for the odd-coded directions, $1, 3, 5, 7$, it is $\sqrt{2}$ (from the Pythagoras theorem). For the case of Figure 7.12b and for a coding with eight possible directions, the resulting chain code is shown in the Figure 7.12c. This sequence of numbers constitutes the spine on which a number of shape-related features are built. Two possibilities, for example, are the following [Lai 81, Mahm 94].

Direction and Direction Length Features

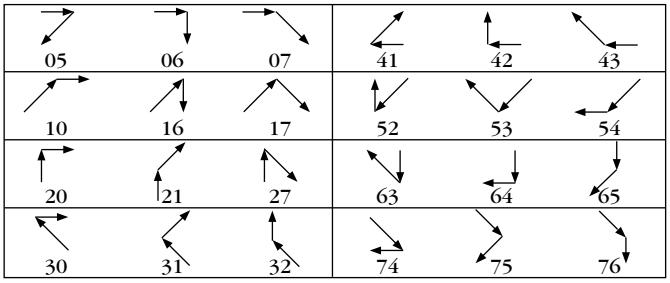
For each direction we count the number of times a specific chain code number appears in the chain. Then this number is divided by the total number of chain codes that appear in the chain description of the boundary. For an eight-code scheme this procedure gives rise to eight features (one for each direction). Another way that also provides eight features is to divide the total length of the line segments in each direction by the total length of the boundary line.

Curvature Features

These features quantify concave (smaller than 180°) and convex (larger than 180°) external angles between adjacent edges at the corners of the polygon that is formed by the line segments when the boundary curve is scanned in the *clockwise sense*. Figure 7.13 shows the possible cases. For example, successive directions $01, 02, 23, 71$ correspond to concave external angles, and the pairs $06, 41, 64, 75$ to convex angles. The occurrence percentage of each of these cases in the chain code defines a respective feature. Sometimes chain code pairs are grouped according to whether the first chain code is even or odd. Thus, a total of 16 features are generated, 8 for the convex and 8 for the concave case.



(a) Concave features



(b) Convex features

FIGURE 7.13

Curvature features characterizing the boundary polygon that results from an eight-directional chain code description of the boundary.

7.3.3 Moment-Based Features

In (7.21) and (7.25) the geometric moments and central moments were defined. If in the place of $I(i,j)$ we consider the sequence

$$I(i,j) = \begin{cases} 1 & (i,j) \in C \\ 0 & (i,j) \text{ otherwise} \end{cases}$$

where C is the set of points (i,j) *inside* the object of interest, then we obtain a way to describe the shape of the object through the moments. Indeed, in such a case only the limits in the summations (hence the object's shape) are taken into account, whereas the details inside the object (i.e., texture) do not participate. Hence

$$m_{pq} = \sum_i \sum_j i^p j^q, \quad (i,j) \in C$$

with $m_{00} = N$, the total number of pixels inside the region. The features

$$\bar{x} = \frac{m_{10}}{m_{00}} \quad \text{and} \quad \bar{y} = \frac{m_{01}}{m_{00}}$$

define the *center of mass* $(\bar{x}, \bar{y})$. The respective central moments become

$$\mu_{pq} = \sum_i \sum_j (i - \bar{x})^p (j - \bar{y})^q, \quad (i,j) \in C$$

The invariant moments can in turn be computed and used, whenever appropriate. Two useful quantities that are related to these moments and provide useful discriminatory information are:

1. *Orientation*

$$\theta = \frac{1}{2} \tan^{-1} \left[\frac{2\mu_{11}}{\mu_{20} - \mu_{02}} \right]$$

which is the angle between the axis with the least moment of inertia and the x-coordinate axis (Problem 7.18).

2. *Eccentricity*

$$\epsilon = \frac{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}}{area}$$

Another representation of the eccentricity is via the ratio $\frac{R_{\max}}{R_{\min}}$ of the maximum to the minimum distance of the center of mass $(\bar{x}, \bar{y})$ from the object's boundary (Figure 7.10b).

7.3.4 Geometric Features

The features of this subsection are derived directly from the geometry of the object's shape. The *perimeter* P of the object and its *area* A are two widely used features. If $\boldsymbol{x}_i, i = 1, 2, \ldots, N$, are the samples of the boundary, then the perimeter is given by

$$P = \sum_{i=1}^{N-1} \|\boldsymbol{x}_{i+1} - \boldsymbol{x}_i\| + \|\boldsymbol{x}_N - \boldsymbol{x}_1\|$$

If we consider the area of a pixel as the measuring unit, a straightforward way to compute the area enclosed by a boundary is by counting the number of pixels inside the region of the object. The *roundness ratio* is a third quantity, defined as

$$\gamma = \frac{P^2}{4\pi A}$$

A useful feature that is related to the curvature of the boundary, as defined in (7.44), is the so-called *bending energy* at a point n, given by

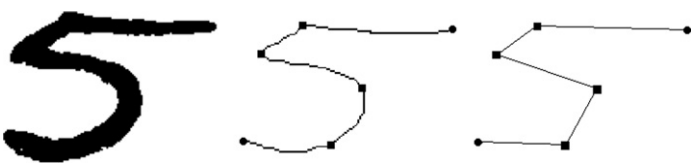
$$E(n) = \frac{1}{P} \sum_{i=0}^{n-1} |k_i|^2$$

Another popular feature is the *number of corners* in the boundary contour. These correspond to points where the curvature k_i takes large values (infinity in theory). In [Ghos 97] corners as well as other topological features are detected via the use of Zernike moments and appropriate parametric modeling of the respective topological image intensity profile.

The *number of holes* inside the region of an object is another useful quantity. For example, a large error percentage in handwriting character recognition tasks is related to the difficulty of the classifiers in distinguishing "8" from "0," because their boundaries look alike. The detection of the presence of holes inside the object, using appropriate algorithms, is extra information that can be beneficially used for recognition [Lai 81, Mahm 94].

In our study so far, we have demonstrated how to derive geometric features from the boundary curve. However, this is not the only possibility. For example, in [Wang 98] geometric features are extracted directly from the gray-level variation within the image region. In this way, the binarization phase is avoided, which in some cases can become tricky. Another direction that has been used extensively in OCR is to work on the thinned version of the binarized character.

Figure 7.14 illustrates the procedure via an example. Figure 7.14b is the result of the application of a thinning algorithm (e.g., [Pita 94]) on the binary version of the character "5" of Figure 7.14a. Also in Figure 7.14b the so-called *key points* are denoted. These can be node points where one or more lines (strokes) of the character are crossed or corner points or end points. These can be computed by processing neighboring pixels. For example, in order to identify an end point, we look at its eight neighboring pixels. An end point has only one neighbor at gray-level 1, and the rest are 0. In the sequel, the thinned version of the character is simplified as a set of line segments (edges) connecting the key points, Figure 7.14c. Each edge can then be characterized by its direction, for example, using the chain code; its length, for example, long or short; and its relation to its neighboring edges. In the sequel each character is described by an array providing this information in a coded form. Classification is then based on these coded matrices by defining appropriate costs. The interested reader may consult for example [Lu 91, Alem 90] for more details.



(a) (b) (c)

FIGURE 7.14

(a) Binarized version of 5 (b) the thinned version with the key points and (c) version with edges connecting key points.

7.4 A GLIMPSE AT FRACTALS

We have already seen that the 1980s was the decade in which two major tools were introduced into the realm of pattern recognition applications (among others): neural networks and wavelets. The same decade was also the time when another tool was adopted in many application areas to offer its potential power. *Fractals* and *fractal dimension* have become the focus of considerable research effort. This section aims at giving the basic definitions and outlining the basic concepts behind the use of fractals in pattern recognition. A deeper study of the area is beyond the goals of a short section, and the interested reader may refer to a number of books and articles available [Mand 77, Tson 92, Falc 90].

7.4.1 Self-Similarity and Fractal Dimension

Let us consider the straight-line segment of length L in Figure 7.15a. Divide L into N (two for the example of the figure) equal parts of length l. Each of the resulting parts is still a straight-line segment, and its length has been scaled down by a factor $m = \frac{l}{L} = \frac{1}{N}$. Magnification of any of these parts by the same factor will reproduce the original line segment. We refer to such types of structures as *self-similar*. If instead of a straight-line segment we had a square of side L (Figure 7.15b), then scaling

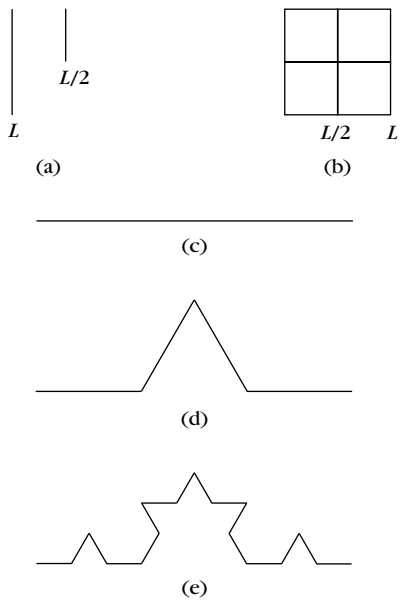


FIGURE 7.15

Self similar structures (a) line segment, (b) square, (c)–(e) three stages in the generation of Koch's curve.

down the side by $m = \frac{1}{N^{1/2}}$ would result into N square parts. Each part looks like the original square, which can be reobtained from the parts after magnification. The same is true for all dimensions, where N similar parts result after scaling the sides of the D-dimensional (hyper)cube by $m = \frac{1}{N^{1/D}}, D = 1, 2, \ldots$. That is, the Euclidean dimension D is directly related to the scaling and the number N of the resulting self-similar parts. We can write

$$N = \left(\frac{L}{l}\right)^D \equiv m^{-D} \tag{7.45}$$

If we now want to measure the length (area, volume) of the original segment (hypercube) using as a measurement unit a scaled element of length l (l^2, l^3, etc.), then the result is independent of the size l of the measuring unit. Indeed, the resulting metric property (length, area, etc.) is given by

$$M = N(l)l^D \tag{7.46}$$

where $N(l)$ is the number of parts that cover the curve (area, etc.) to be measured and l is the size of the measuring unit. Combination of (7.45) and (7.46) leads to a metric M that is always constant (L^D) for the same structure, and it is independent of the size l of the chosen unit, as expected.

Let us now turn our attention to some more interesting structures, such as the one in Figure 7.15d. The curve in Figure 7.15d results from the straight-line segment of Figure 7.15c, known as the *initiator*, by the following strategy: (a) divide the segment into three equal parts and (b) replace the central one by the two sides of an equilateral triangle, with sides of size equal to the size of the scaled parts. The procedure is then repeated for each of the line segments in Figure 7.15d, and this results in the structure of Figure 7.15e. This process can go on indefinitely, and the limit curve is the so-called Koch curve [Mand 77]. Such a curve is everywhere continuous but nowhere differentiable. It is readily observed that at each step of the iteration, the resulting structure is part of the structure that will result in the next iteration, after a scaling by 3. The curve therefore has a self-similar structure. In the sequel we will try to measure the length of the curve. Using as a (measuring) unit a segment of length $l = \frac{L}{3}$ (i.e., Figure 7.15d), the resulting length is 4. For a unit segment $l = \frac{L}{3^2}$ (i.e., Figure 7.15e), the measured length is 4^2. It is not difficult to see that the length keeps increasing with decreasing unit size and tends to infinity as the size of the measuring unit tends to zero! That is, the length of the curve depends not only on the curve itself but also on the adopted measurement unit! This strange result is the outcome of an "unfair" measurement process. Indeed, in the case of the Koch curve, scaling by 3 results in four similar parts. In contrast, in the case of a straight-line segment, scaling by $m = \frac{1}{N}$ results in the same number N of similar parts. In higher dimensional Euclidean space, scaling by $m = N^{-1/D}$ results in N parts. The measurement process involves this number N, the scaled side length l, and the *Euclidean dimension* D, as (7.46) suggests. From this discussion, the Euclidean dimension can also be seen as the ratio $\frac{\ln N}{-\ln m} = D$. Starting from this observation, let us now define the *similarity dimension* of a

general self-similar structure as

$$D = \frac{\ln N}{-\ln m} \tag{7.47}$$

where N is the number of the resulting similar parts, when scaling by a factor m. For hypercube structures the similarity dimension is the respective Euclidean dimension, which is an *integer*. In contrast, the corresponding similarity dimension of the Koch curve $D = \frac{\ln 4}{-\ln(\frac{1}{3})}$ is a *fraction* and not an integer. Such structures are called *fractals*, and the corresponding similarity dimension is called a *fractal dimension*. Measuring a fractal structure, we can adopt (7.46) with the corresponding fractal dimension in the place of D. The result of the measurement process now becomes independent of the measuring tool l. Indeed, it is easy to see that using the definition in (7.47), (7.46) results in a constant $M = L^D$ for $m = \frac{l}{L}$. *The use of similarity dimension, therefore, results in a consistent description of the metric properties of such self-similar structures*. For a deeper treatment and other definitions of the dimension the interested reader may consult more specialized texts (e.g., [Falc 90]).

7.4.2 Fractional Brownian Motion

A major part of our effort in this chapter was dedicated to the description of statistical properties of signals and images and to the ways these can be exploited to extract information-rich features for classification (e.g., co-occurrence matrices, AR models). In this section we will focus our attention on whether the notion of self-similarity is extendable to stochastic processes and, if it is, how useful it can be for our interests. In the previous section "similarity" referred to the shape of a curve. For statistics such a view would be of no interest. From a statistical point of view it would be more reasonable and justifiable to interpret similarity from the perspective of "similar statistical properties," that is, mean, standard deviation, and so forth. Indeed, it can be shown that stochastic processes that are self-similar under scaling do exist. Furthermore, such processes can model adequately a number of processes met in practice.

Let $\eta(n)$ be a white (Gaussian) noise sequence with variance $\sigma_\eta^2 = 1$. The process defined as

$$x(n) = \sum_{i=1}^{n} \eta(i)$$

is known as a *random walk* sequence, and it belongs to a more general class of stochastic processes known as *Brownian motion* processes [Papo 91, p. 350]. It is straightforward to see that

$$E[x(n)] = 0$$

and that its variance is given by

$$E[x^2(n)] = n\sigma_\eta^2$$

Thus, the process is a nonstationary one because its variance is time dependent. A direct generalization of the previous result is

$$E[\Delta^2 x(n)] \equiv E[(x(n+n_0) - x(n_0))^2] = n\sigma_\eta^2 \tag{7.48}$$

where by definition $\Delta x(n)$ is the sequence of increments. Scaling the time axis by m results in

$$E[\Delta^2 x(mn)] \equiv E[(x(mn+n_0) - x(n_0))^2] = mn\sigma_\eta^2 \tag{7.49}$$

Hence, if the sequence of increments is to retain the same variance after scaling, it should be scaled by $\sqrt{m}$. Furthermore, it is easy to see that the sequence of increments, as well as the scaled versions, follow a Gaussian distribution (e.g., [Falc 90]). Recalling that Gaussian processes are completely specified by their mean and variance, we conclude that the increments $\Delta x(n)$ of $x(n)$ are *statistically self-similar in the sense that*

$$\Delta x(n) \quad \text{and} \quad \frac{1}{\sqrt{m}}\Delta x(mn) \tag{7.50}$$

are described by the same probability density functions, for any n_0 and m. Figure 7.16 shows three curves of the scaled random walk increments for $m = 1, 3, 6$. It is readily observed that they indeed "look" alike. Such curves, for which different scaling has been used for the coordinates $(\Delta x, n)$, are also known as *statistically self-affine*.

The random walk Brownian motion is a special case of a more general class of processes known as *fractional Brownian motion sequences* (fBm), introduced in [Mand 68]. The increments of this type of processes have variance of the general form

$$E[\Delta^2 x] \propto (\Delta n)^{2H} \tag{7.51}$$

with $0 < H < 1, \Delta n \equiv n - n_0$ and $\propto$ denoting proportionality. The parameter H is also known as the *Hurst parameter*. As in the case of Brownian motion, the increments of such processes are *statistically self-affine* in the sense that the processes

$$\Delta x(n) \quad \text{and} \quad \frac{1}{m^H}\Delta x(mn)$$

are described by the same probability density functions. The parameter H relates to the visual smoothness or coarseness of the respective graph of the increments versus time. This is an implication of (7.51). Let us start from a maximum interval Δn, corresponding to an incremental variance σ^2. In the sequel we halve the interval to $\Delta n/2$. The respective variance will be reduced by the factor $(1/2)^{2H}$.

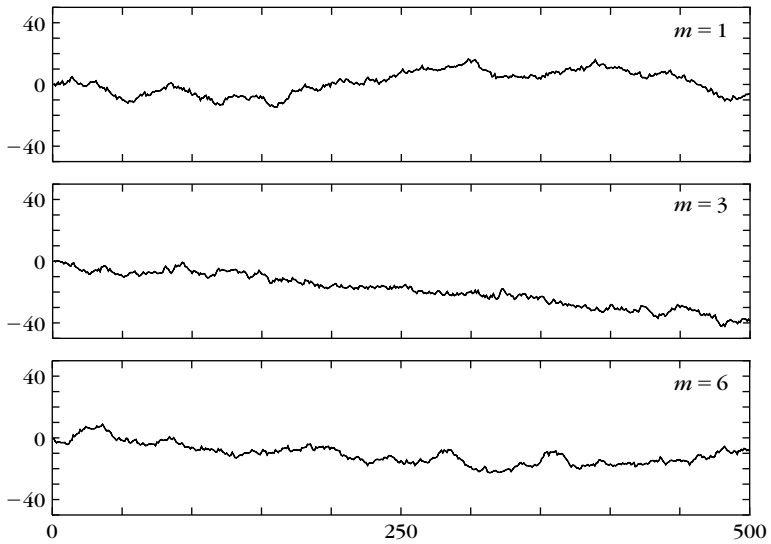


FIGURE 7.16

Time evolution of the random walk ($m = 1$) sequence and two of its self-affine versions. They all do look alike.

This process can go on. Each time we reduce the interval Δn by half, we look at increments between points located closer in time. The higher the value of H, the greater the reduction of the variance of the increments between these points, indicating smoother curves. For $H = 0$, the variance of the increments remains constant and independent of Δn. This process is not an fBm, and it corresponds to a white noise stationary process, with no dependence between adjacent time instants. Hence, it exhibits the most erratic behavior, and its graph has the most coarse appearance. *This observation indicates that the parameter H could be used as a measure of the "smoothness" of such curves*. By varying H one can get curves of varying degree of smoothness [Saup 91]. Figure 7.17 indeed verifies that the curve for $H = 0.8$ is smoother than the one for $H = 0.2$, and both are smoother than the top one, which corresponds to a white noise sequence. As was the case with the fractal curves of the previous subsection, a dimension can also be defined for curves resulting from fBm processes. It can be shown [Falc 90, p. 246] *that an fBm process with parameter H corresponds to a curve with fractal dimension* $2 - H$. In general, if l is the number of free parameters of the graph, the corresponding fractal dimension is $l + 1 - H$. For a graph as in Figure 7.17, $l = 1$ and for an image $l = 2$.

The question now is how all these no doubt mind-stimulating points can be of use to us in the context of pattern recognition. The terms *smoothness* and *coarseness* have been used in association with the parameter H and equivalently with the dimension D of an fBm process. On the other hand, the terms *smoothness*

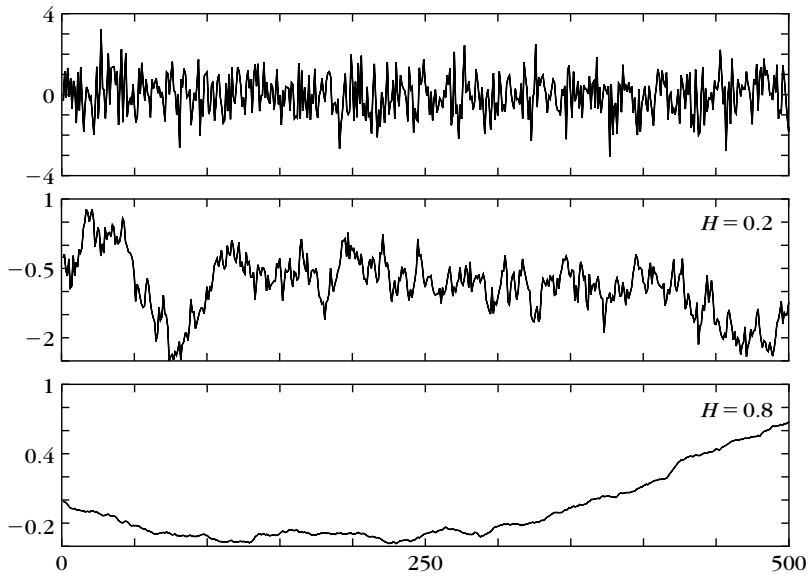


FIGURE 7.17

Time evolution of a white noise sequence (top) and two fBm processes of different Hurst parameters H. The lower the value of H, the coarser the appearance of the graph.

and *roughness*, were in a central position when dealing with feature generation for texture classification. We have now reached our crucial point. Provided that we can describe the sequence of gray levels of an image as an fBm process, the corresponding fractal dimension can be used as a potential feature for texture classification. In [Pent 84] it is reported that this is indeed true for a number of cases. Using a number of textured images from [Brod 66], as well as images from natural scenes, it was found that a large percentage of them exhibited fBm behavior. This was easily verified by constructing the histogram of differences (increments) of the gray-level intensities for various relative pixel distances Δn. It turned out that for each value of Δn the corresponding histogram was close to a Gaussian pdf centered at zero. Furthermore, the widths of the Gaussian-like histograms were different for the different relative pixel distances Δn. The larger the Δn, the wider the resulting histogram. However, we know that the width of a Gaussian is directly related to its variance. The plot of the variance as a function of relative pixel distance revealed an underlying fBm nature of the intensity processes, at least over a 10:1 range of relative distances measured. The parameter H, or equivalently the fractal dimension D, was then used successfully to distinguish a number of different textured regions in an image. The estimation of the H can take place via its definition in (7.51). Taking the logarithm, we get

$$\ln E[\Delta^2 x] = \ln c + 2H \ln \Delta n$$

where c is the proportionality constant and Δx is now the difference in the gray-level intensities between pixels at relative distance Δn, that is, $\Delta n = 1, 2$, and so on. Obviously, $c = E[\Delta^2 x]$ for $\Delta n = 1$. For each pixel distance Δn the corresponding average $\Delta^2 x$ is computed over the image window of interest. The resulting points $(E[\Delta^2 x], \Delta n)$ are plotted in a two-dimensional logarithmic plot. A straight line is then fitted through the points using a least squares linear regression technique. The parameter H is provided by the slope of the line. This is also a test of the fractal nature of the underlying process. If the resulting points do not lie approximately on a straight line, the fractal model assumption will not be valid. Figure 7.18 demonstrates the procedure for two images. The one on the right is an artificially produced fractal image with $H = 0.76$. The least squares fit in the logarithmic plot of the standard deviation against Δn results in a straight line of slope 0.76. The image on the left is from [Brod 66]. We observe that the resulting least squares fit is reasonable, suggesting that the image is approximately fractal in nature. The slope is now $H = 0.27$. The lower value of H reflects the fact that the latter image is coarser than the former.

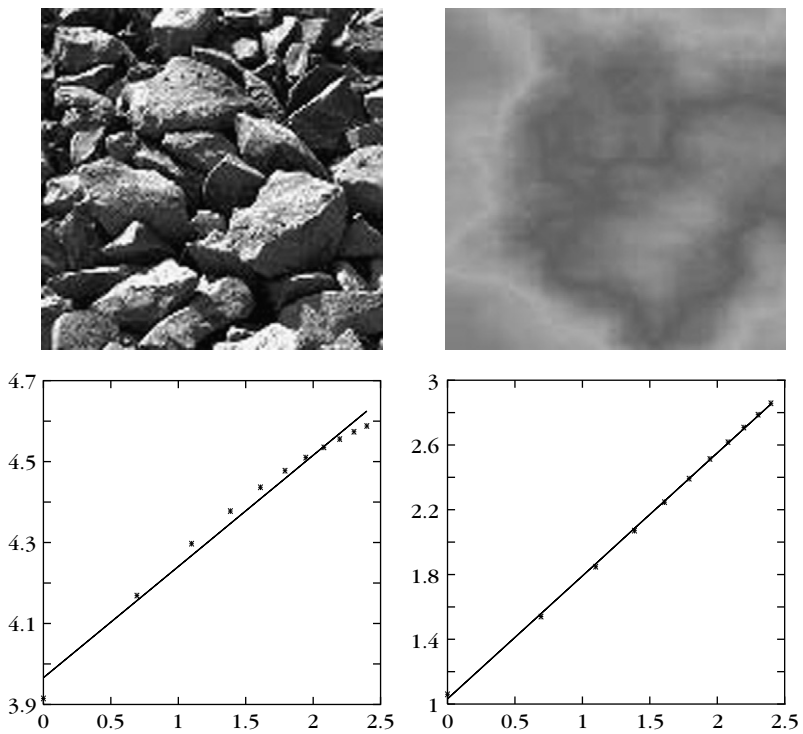


FIGURE 7.18

Examples of images with corresponding logarithmic plots of the standard deviation of increments (vertical axis) versus relative distance (horizontal axis).

The method presented previously for the computation of the Hurst parameter is not the only one, and a number of other techniques have been suggested. A popular alternative is based on the wavelet analysis of the underlying fBm process. The basis sequences (functions) used in the wavelet analysis are scaled and translated versions of a mother sequence (function). This underlying notion of scale invariance is shown to relate to fBm processes, whose statistical properties are scale invariant [Flan 92]. It turns out that the wavelet coefficients of an fBm process, at a given resolution level, i, form a stationary sequence, with a variance proportional to $2^{-i(2H+1)}$, see [Worn 96]. This leads to a simple method for estimating the associated Hurst parameter. Other methods for estimating the fractal dimension include the box-counting and variation method [Huan 94, Kell 87]; maximum likelihood estimates, as in [Lund 86, Deri 93, Fieg 96]; morphological covers [Mara 93]; methods in the spectral domain, as in [Gewe 83]; and fractal interpolation function models [Penn 97].

Fractional modeling and the use of fractal dimension D as a feature for classification have been demonstrated in a number of applications [Chen 89, Lund 86, Rich 95]. However, the method is not without drawbacks. Indeed, it may happen that different textures result in the same fractal dimension, thus limiting the classification potential of the method. Another shortcoming is that in practice physical processes retain their fractal characteristics over a range of distances but not over all ranges. Thus, the fractal dimension may change as we pass from one range of scales to another [Pent 84, Pele 84], and the use of a single Hurst parameter may not lead to sufficient modeling. To overcome these drawbacks, a number of possible solutions have been suggested. The *multifractional Brownian motion* (mBm) is an extension of an fBm process with a parameter H, which is allowed to vary, as in [Ayac 00]. *Extended self similar* (ESS) processes allow in (7.51) for a more general dependence on Δn, via a so-called structure function [Kapl 94]. For more on these issues the interested reader is referred, for example, to [Bass 92, Ardu 92, Kapl 95, Kapl 99, Pesq 02]. A comparative study of various textural features, including fractal modeling techniques, can be found in [Ohan 92, Ojal 96].

7.5 TYPICAL FEATURES FOR SPEECH AND AUDIO CLASSIFICATION

As we have already commented in the Preface, speech recognition is a major application area of pattern recognition, and a number of speech-recognizing systems are already available in the market. Audio classification and recognition have also received a lot of attention in recent years. A great number of commercial applications are envisaged for the future in the field of multimedia databases. Techniques for automatic indexing tools, intelligent browsers, and search engines with content-based retrieval capabilities are currently the focus of a major research effort. In this context, *audiovisual* data segmentation and indexing, based not only on visual information but on the accompanying audio signal, greatly enhance the performance. For instance, classifying video scenes of gun fights using the audio

information related to shooting and explosions will, no doubt, enhance the performance of the classifier compared to a system that is based on the corresponding visual information only.

Content-based retrieval from music databases is another application that attracts the interest of current research. It is very likely that not far in the future a large corpus of the recorded music in human history will be available on the Web. Automatic music analysis is envisaged to be one of the main services to facilitate content distribution. Automatic music genre classification, querying music databases by *humming* the tune of a song or querying by *example* (i.e., providing a music extract of short duration in order to locate and retrieve the complete recording) are examples of services that vendors would very much like to offer in such systems. More on these issues can be found in [Wold 96, Wang 00, Zhan 01, Pikr 03, Pikr 06, Pikr 08, Frag 01, Clau 04].

This section focuses on the generation of some typical and commonly used features for speech recognition and audio classification/recognition. However, as has already been stated elsewhere in the book one must keep in mind that feature generation is very much a problem-dependent task. Thus, the combination of the designer's imagination with his or her good knowledge of the peculiarities of the specific task can only benefit the generation of informative features.

7.5.1 Short Time Processing of Signals

Speech and audio signals are statistically nonstationary; that is, their statistical properties vary with time. A way to circumvent this problem and be able to use analysis tools that have been developed and make sense for stationary signals only, such as the Fourier transform, is to divide the time signal into a series of successive *frames*. Each frame consists of a *finite* number, N, of samples. During the time interval of a frame, the signal is assumed to be "reasonably stationary." Such signals are also known as *quasistationary*. Figure 7.19 shows a signal segment and three successive frames, each consisting of $N = 20$ samples, with an overlap among neighboring frames of 5 samples. Choosing the length, N, of the frames is a "bit of an art" and is a problem-dependent task. First, each frame must be long enough for an analysis method to have enough "data resources" to build up the required information. For example, if we are interested in estimating the period of a periodic signal the number of samples in each frame must be large enough to allow the signal periodicity to be revealed. Of course, this will depend on the value of the period. For short periods, a few samples can be sufficient. On the other hand, for long periods more samples will be necessary. Second, N must be short enough to guarantee the approximate stationarity of the signal within the time scale of each frame in order for the results to be meaningful. For speech signals sampled at a frequency of $f_s = 10$ KHz, reasonable frame sizes range from 100 to 200 samples, corresponding to 10–20 msecs time duration. For music signals sampled at 44.1 KHz, reasonable frame sizes range from 2048 to 4096 samples, corresponding to 45–95 msecs, approximately.

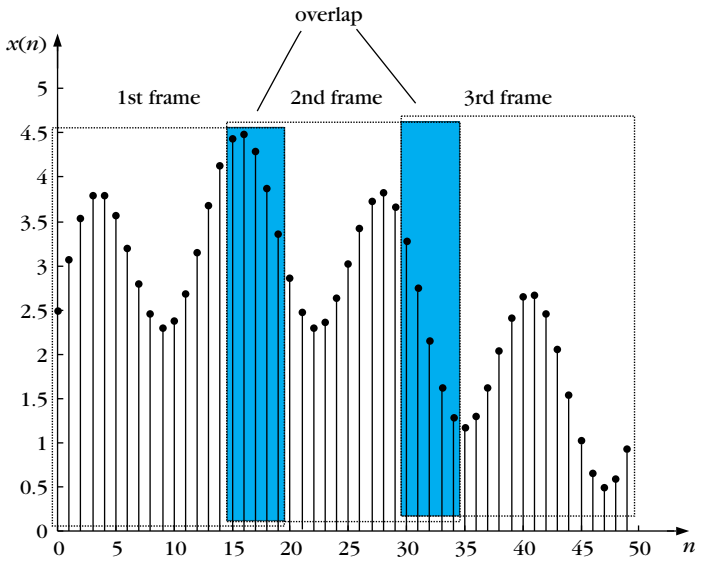


FIGURE 7.19

Three successive frames, each of length $N = 20$ samples. The overlap between successive frames is 5 samples.

From a mathematical formulation point of view, dividing the time signal in a sequence of successive frames is equivalent to multiplying the signal segment by a *window* sequence, $w(n)$, of finite duration N. The simplest window sequence is the *rectangular window*, defined as

$$w(n) = \begin{cases} 1 & 0 \leq n \leq N-1 \\ 0 & \text{elsewhere} \end{cases}$$

For different frames, the window is shifted to different points, m_i, in the time axis. Hence, if $x(n)$ denotes the signal sequence, the samples of the ith frame can be written as

$$x_i(n) = x(n + m_i)w(n)$$

where m_i is the corresponding window shift associated with the ith frame. This implies that all samples in the ith frame are identically zero, except for the time instants $n = 0, \ldots, N-1$, which correspond to the original signal samples, $x(n)$, with $n \in [m_i,\ m_i + N - 1]$. The procedure is illustrated in Figure 7.20. As it is known from the Fourier transform theory basics, multiplying a sequence by a window in the time domain smooths out its Fourier transform by convolving it with the Fourier transform of the window sequence. Some of the effects of this smoothing action can be minimized by using a different window sequence, with a smoother decay to

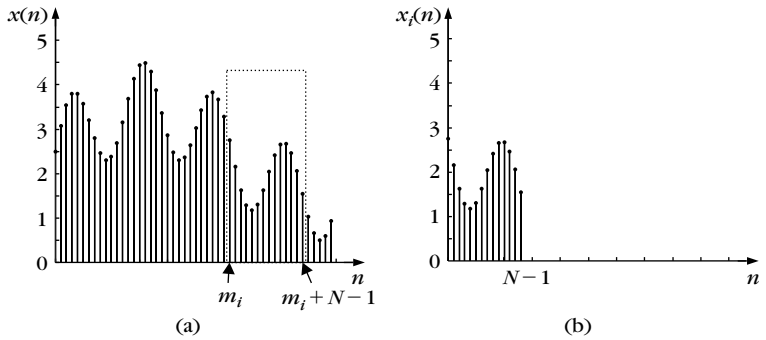


FIGURE 7.20

A signal segment (a) and the resulting frame (b) after the application of a rectangular window sequence of duration equal to 14 samples and shifted at m_i.

zero. A popular choice is the *Hamming window*, defined as

$$w(n) = \begin{cases} 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right) & 0 \le n \le N-1 \\ 0 & \text{elsewhere} \end{cases}$$

More on these issues can be found in [Rabi 78, Dell 00].

As an example, let us assume that we have divided a speech segment into a sequence of I frames, each of length N. Then, for each frame, $i = 1, 2, \ldots, I$, we can compute the discrete Fourier transform (DFT) as

$$X_i(m) = \sum_{n=0}^{N-1} x_i(n) \exp\left(-j\frac{2\pi}{N}mn\right), \quad m = 0, 1, \ldots, N-1$$

It is common to refer to this DFT as the *short-time DFT*. It must be pointed out that this definition implies much more theory and interesting implementation issues (which, however, are not of interest to us and we will not delve into this topic any deeper). Selecting $l \le N$ DFT coefficients from each frame, we can construct a sequence of feature vectors

$$\boldsymbol{x}_i = \begin{bmatrix} X_i(0) \\ X_i(1) \\ \vdots \\ X_i(l) \end{bmatrix}, \quad i = 1, 2, \ldots, I \tag{7.52}$$

Thus, the pattern of interest (i.e., the speech segment) is not represented by a single feature vector but by a *sequence* of feature vectors. We will see how to attack such problems in Chapters 8 and 9.

The autocorrelation sequence is another very important statistical quantity, which is also defined for stationary processes. Recall from Section 7.2.4 that, if $x(n)$ is a stationary process, the autocorrelation sequence is defined as

$$r(k) = E[x(n)x(n-k)] = E[x(n)x(n+k)] = r(-k) \tag{7.53}$$

In other words, it is the expectation of the product of $x(n)$ with its shifted version $x(n \pm k)$. In practice, the expectation can be obtained as

$$r(k) = \frac{1}{2N+1} \sum_{n=-N}^{N} x(n)x(n+k)$$

which under mild assumptions (i.e., ergodicity) tends to the true value of $r(k)$ as N tends to infinity. In the case of a quasistationary process, the *short-time autocorrelation* sequence, $r_i(k)$, is defined for each of the frames as

$$r_i(k) = \frac{1}{N} \sum_{n=0}^{N-1-|k|} x_i(n)x_i(n+|k|) \tag{7.54}$$

where $|\cdot|$ denotes the absolute value operator. The limits in the sum indicate that outside the interval $[0,\ N-1-|k|]$ the product $x_i(n)x_i(n+|k|)$ is identically zero, due to the finite duration, N, of the frame. This definition complies with the definition in (7.53) in the sense that for stationary processes $r_i(k)$ is an asymptotically unbiased estimate of the autocorrelation as the length of the frame $N \Longrightarrow \infty$. Indeed, viewing $r_i(k)$ as an estimate of $r(k)$, its mean value for different realizations is easily shown to be

$$E[r_i(k)] = \frac{N-|k|}{N} r(k) \tag{7.55}$$

Thus, for finite frame length, N, Eq. (7.54) results in a biased estimate of $r(k)$. However, for small values of the lag k, with respect to N, the bias is small. On the other hand, for values of k close to N we expect $r_i(k)$ to get values close to zero, something that is verified in practice. To remedy this drawback, other definitions of the short-time autocorrelation have been proposed. See, for example, [Rabi 78]. Another important property of the definition in (7.54), from the computational point of view, is that the corresponding (short-time) autocorrelation matrix retains the computationally elegant properties of being symmetric and having a Toeplitz structure (Section 7.2.4).

7.5.2 Cepstrum

Let $x(0), x(1), \ldots, x(N-1)$ be the samples from the current data frame (the index i has been dropped for notational simplicity). The Fourier transform (FT) of this finite-length sequence of data is defined as the periodic complex function

$$X(\omega) = \sum_{n=0}^{N-1} x(n) \exp\left(-j\omega Tn\right) \tag{7.56}$$

with period (in the frequency domain) $2\pi/T$, where T is the sampling period. It is well known from the basic theory of signal processing (e.g., [Proa 92] and Chapter 6), that the coefficients of the DFT transform

$$X(m) = \sum_{n=0}^{N-1} x(n) \exp\left(-j\frac{2\pi}{N}mn\right), \quad m = 0, 1, \ldots, N-1 \tag{7.57}$$

are the samples of the FT taken at the frequency points $0, \frac{2\pi}{NT}, \ldots, \frac{2\pi}{NT}(N-1)$. Assuming, without loss of generality, $T = 1$, the inverse FT is defined as

$$x(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(\omega)\exp\left(j\omega n\right)d\omega, \quad n = 0, 1, \ldots, N-1 \tag{7.58}$$

That is, the resulting samples are equal to the samples of the original sequence and identical to what is obtained by the inverse DFT. That is,

$$x(n) = \frac{1}{N}\sum_{m=0}^{N-1} X(m)\exp\left(j\frac{2\pi}{N}mn\right), \quad n = 0, 1, \ldots, N-1 \tag{7.59}$$

The *cepstrum*, $c(n)$, of a sequence, $x(n)$, is the sequence resulting from the inverse FT of the logarithm of the magnitude of its FT. That is,

$$c(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \log_{10}|X(\omega)|\exp\left(j\omega n\right)d\omega \tag{7.60}$$

Although the base 10 logarithm has been used, the logarithm of any base can be adopted. Another way of looking at the cepstral coefficients, $c(n)$, is the following. Since the FT function $X(\omega)$ is a periodic function of ω, with period 2π, the function $\log_{10}|X(\omega)|$ is also periodic with the same period. Therefore, it can be written in terms of its Fourier series expansion

$$\log_{10}|X(\omega)| = \sum_{n=-\infty}^{\infty} c(n)\exp\left(-j\omega\frac{2\pi}{2\pi}n\right) = \sum_{n=-\infty}^{\infty} c(n)\exp\left(-j\omega n\right) \tag{7.61}$$

Hence, Eq. (7.60) is the formula that provides the coefficients of the Fourier series expansion in (7.61). However, the function $\log_{10}|X(\omega)|$ is defined in the frequency and not in the time domain. Its Fourier transform domain is known as the *quefrency* domain, and the respective Fourier series coefficients, $c(n)$, are known as *cepstral coefficients*. This is only to remind us that the original transformed function is in the frequency domain. Otherwise, all Fourier transform/series properties are still valid. Thus, since $\log_{10}|X(\omega)|$ is a real and even function, ($|X(\omega)|$ is even for a real sequence $x(n)$), the cepstral coefficients are real and even. That is,

$$c^*(n) = c(n) = c(-n)$$

Cepstral coefficients have very good information-packing properties, from the class discrimination point of view, and are very popular candidates as features, both for speech recognition and audio classification tasks [Rabi 93, Tzan 02].

Computation of the cepstral coefficients is achieved via the DFT (using the FFT) of $X(\omega)$. However, this computation is not as innocent as it is for the case of $X(\omega)$. As we have already seen, the inverse transforms of $X(\omega)$ and $X(m)$ coincide [Eqs. (7.58) and (7.59)]. This is because the input sequence is of finite length, N, and the sampling period ω_s, in the frequency domain, is chosen to obey the Nyquist criterion; that is, $\omega_s = \frac{2\pi}{N}$. This is not the case with the cepstral coefficients. Using $\log_{10}|X(m)|$, $m = 0, 1, \ldots, N-1$, and taking the inverse DFT results in

$$\hat{c}(n) = \frac{1}{N}\sum_{m=0}^{N-1} \log_{10}|X(m)|\exp\left(j\frac{2\pi}{N}mn\right), \quad n = 0, 1, \ldots, N-1 \tag{7.62}$$

where $\hat{c}(n)$ and $c(n)$ are related as

$$\hat{c}(n) = \sum_{r=-\infty}^{\infty} c(n + rN) \tag{7.63}$$

For those readers familiar with basic digital signal processing this is most natural. The sequence $c(n)$ is not of finite duration. Thus, sampling its FT ($C(\omega) \equiv \log_{10} |X(\omega)|$) with a sampling period of $\frac{2\pi}{N}$ does not satisfy Nyquist's criterion. Hence, taking the inverse DFT [Eq. (7.62)] will result in aliasing with a periodic repetition of the aliased sequence ($c(n)$) every N samples, which is expressed via (7.63). See, for example, [Proa 92]. In practice, the effects of this aliasing are minimized if one extends the length of the frame from N to M by appending $M - N$ zeros at the end of it. That is,

$$x(n) : \; x(0), \ldots, x(N-1), x(N) = 0, \ldots, x(M-1) = 0 \tag{7.64}$$

These zeros (as it can easily be checked out) *have no effect on the* FT $X(\omega)$. However, the DFT is now of length M (corresponding to sampling the FT every $\frac{2\pi}{M}$ frequency points), which makes the implicit repetition period in (7.63) equal to M. Assuming that the cepstral coefficients decay fast enough, with respect to the repetition period M, we can assume that $\hat{c}(n) \approx c(n), n = 0, 1, \ldots, N-1$, since successive repetitions in (7.63) now have little overlap. In practice, a number of 512 or 1024 zeros may be necessary. For further information related to cepstrum, the interested reader may refer to [Rabi 78, Dell 00].

In summary, the computational steps to obtain the cepstral coefficients of a frame, $x_i(n), n = 0, 1, \ldots, N-1$, are the following:

- Extend the length of the frame by appending $M - N$ zeros at the end of the frame.
- Obtain the DFT of length M of the extended frame.
- Compute the logarithm of the magnitude of the DFT coefficients.
- Compute the inverse DFT of length M.

The obtained coefficients are the (approximate) cepstral coefficients of the sequence $x_i(n)$.

7.5.3 The Mel-Cepstrum

Human perception of audio has often been studied from a psychophysical point of view. Experiments have suggested that perception of the frequency content of pure tones does not follow a linear scale. This led to the idea of mapping acoustic frequency content to a linear "perceptual" frequency scale. A popular approximation to this type of mapping is known as the *mel scale* [Pico 93, Rabi 93]:

$$f_{mel} = 2595 \log_{10} 10(1 + f/700.0) \tag{7.65}$$

Equation (7.65) suggests that an actual frequency of 1 KHz is mapped to 1000 mel units. The plot of (7.65) is shown in Figure 7.21, and as can be seen the mapping

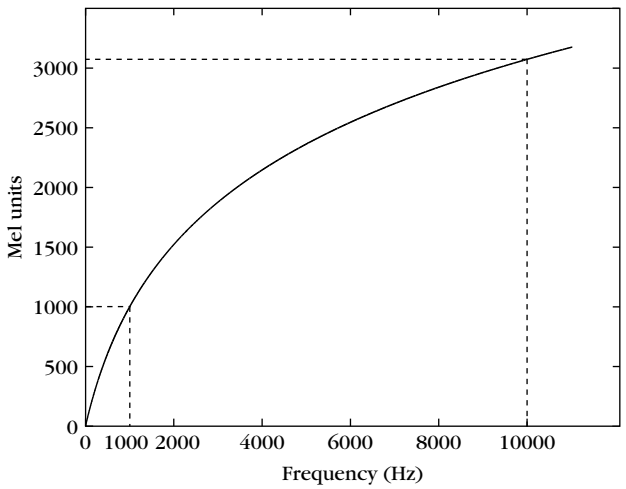


FIGURE 7.21

Subjectively perceived pitch, in mel units, as a function of the measured frequency.

is approximately linear from 0 Hz up to 1000 Hz, and its logarithmic nature prevails for frequencies above 1000 Hz. As an example, Eq. (7.65) suggests that the perceived frequency of a pure tone at the frequency of 10 KHz will be approximately 3000 mel units (mels). That is, a tenfold increase in the frequency will be subjectively perceived as a threefold increase.

Another psychoacoustics phenomenon is related to the way our auditory system perceives the differences in frequency among different tones that contribute to the formation of a more complex sound. It turns out that the tones cannot be individually distinguished if they fall within a certain bandwidth around the center frequency of the sound. We refer to this bandwidth as the *critical bandwidth* [Pico 93, Rabi 93]. Furthermore, if the bandwidth of a complex sound is less than the critical bandwidth around its center frequency the ear would perceive it as a single tone at the center of the critical band and with its loudness being equal to a weighted average of the loudness of each one of the contributing tones. The critical bandwidth around a frequency f can approximately be given by

$$BW_{critical} = 25 + 75[1 + 1.4(f/1000)^2]^{0.69} \tag{7.66}$$

A plot of this equation reveals that the critical bandwidth is approximately linear below 1 KHz and increases logarithmically for frequencies above 1 KHz.

In an effort to generate features that are rich in information, we will try to "manipulate" the frequency content of a sound segment by imitating nature, in the way our auditory system perceives and recognizes sounds. To this end the following steps are adopted:

- A sound segment, of length N, is analyzed via the DFT transform. As we have already observed, it is useful to append a number of zeros at the end and let M denote the number of samples after the extension. If $f_s = 1/T$ is the sampling frequency, each DFT coefficient $X(m)$ corresponds to a real frequency of $\frac{mf_s}{M}$.
- In the sequel, a number, L, of critical bands are "spread" over the frequency range up to $f_s/2$. Figure 7.22 is an example, where only the first 17 of $L = 35$ such bands are shown (for illustration simplicity) to occupy the frequency range from 0 Hz up to approximately 3700 Hz. The shape of each frequency band is a graphical representation of the weighting imposed on the corresponding frequency sample (frequency bin) within the bandwidth of the band. The sampling frequency is $f_s = 44.1$ KHz. These bands are uniformly distributed in a mel scale, and their bandwidth has been chosen to be approximately equal to 110 mels. In the frequency scale, these bands are almost uniformly spaced below 1 KHz and logarithmically above it. The shape of the bands has been chosen to be triangular. In general, the shape, bandwidth, and number of bands are critical design issues, and several approaches have been suggested throughout the years, depending on the application domain (see, for example, [Pico 93, Rabi 93, Davi 80]). In our example, we chose nonoverlapping bands, although in some cases an overlap between successive bands is allowed to exist.
- Since, in general, the center frequencies of these frequency bands do not coincide with the frequency quantization performed by the DFT, each band is moved so that its center frequency coincides with the nearest DFT frequency

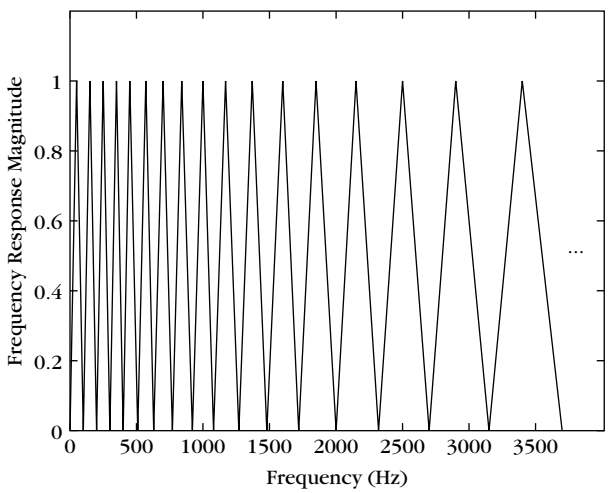


FIGURE 7.22

A critical-band filter bank consisting of nonoverlapping triangular bandpass filters.

bin ($\frac{mf_s}{M}$). Denote by m_i this (center) frequency bin of the ith band in the bank, $i = 1, 2, \ldots, L$.

- For each of the bands in the bank, compute the weighted average of the log-magnitude of the DFT coefficients that fall within the frequency band. That is,

$$Y(m_i) \equiv \sum_m \log_{10} |X(m)| H_i\left(\frac{m}{M} f_s\right), \quad i = 1, 2, \ldots, M \tag{7.67}$$

  where $H_i(\cdot)$ is the corresponding weighting value. Note that since the width of each band is finite, this summation, over m, is restricted to the DFT coefficients located within the bandwidth of the respective critical band.

- Define the sequence

$$Y(m) = \begin{cases} Y(m_i) & \text{if } m = m_i, \quad i = 1, 2, \ldots, L \\ 0 & \text{otherwise} \end{cases} \tag{7.68}$$

  In other words, this sequence is zero everywhere except at the frequency bins corresponding to the centers of the bands in the bank, where the value is equal to the weighted average of the log-magnitude of the DFT coefficients at the frequency bins within the bandwidth of the respective band. We can think of this new sequence as the psychologically perceived log-magnitude spectrum equivalent to the physically measured one.

- Take the inverse DFT

$$c_{\text{mel}}(n) = \frac{1}{M} \sum_{m=0}^{M-1} Y(m) \exp\left(j \frac{2\pi}{M} mn\right), \quad n = 0, 1, 2, \ldots, N-1 \tag{7.69}$$

  These are known as the *mel-cepstral* coefficients and are among the most powerful features in speech and audio recognition/classification. Note that since the log-magnitude DFT coefficients are real and symmetrical, the previous inverse DFT can also be efficiently computed via a cosine transform, as in [Proa 92].

The reader should note that the previous method of defining mel-cepstral coefficients is just one of many variants that have been proposed over the years. For a more extensive treatment the reader is referred to more specialized texts and articles, such as [Pico 93, Rabi 93, Dell 00].

7.5.4 Spectral Features

Let $x_i(n)$, $n = 0, 1, \ldots, N-1$ be the samples of the ith frame and $X_i(m)$, $m = 0, 1, \ldots, N-1$, the corresponding DFT coefficients. The following features are quite common in speech recognition and audio classification/recognition, each providing information for different acoustic qualities.

Spectral Centroid

$$C(i) = \frac{\sum_{m=0}^{N-1} m|X_i(m)|}{\sum_{m=0}^{N-1} |X_i(m)|}$$

The centroid is a measure of the spectral shape. High values of the centroid correspond to "brighter" acoustic structures with more energy in the high frequencies.

Spectral Roll-off

The spectral roll-off is the frequency sample, $m_c^R(i)$, below which the c% (e.g., c = 85 or 90) of the magnitude distribution of the DFT coefficients is concentrated. That is, for this frequency sample the following is true:

$$\sum_{m=0}^{m_c^R(i)} |X_i(m)| = \frac{c}{100} \sum_{m=0}^{N-1} |X_i(m)|$$

This is another measure indicating where most of the spectral energy is concentrated. It is a measure of skewness of the spectral shape, with right-skewed shapes (brighter sounds) resulting in higher values.

Spectral Flux

$$F(i) = \sum_{m=0}^{N-1} (N_i(m) - N_{i-1}(m))^2$$

Here, $N_i(m)$ is the normalized (by its maximum value) magnitude of the respective DFT coefficient of the ith frame and is a measure of the *local* spectral change between successive frames.

Fundamental Frequency

Speech and audio signals can be either noise-like in nature, such as unvoiced speech segments or audio segments corresponding to an applause or footstep recordings, or can exhibit a periodic nature. In the latter case, we talk about *harmonic* signals to distinguish them from their noise-like *inharmonic* counterparts. Audio signals produced from musical instruments and voiced speech segments are two examples of harmonic signals. A distinct characteristic of a harmonic sound signal is its *fundamental frequency*.

In the case of voiced speech signals, this is the frequency of successive vocal fold openings and is also known as the *pitch* of the signal. For men this lies in the range of 80 to 200 Hz and for women in the range of 150 Hz to 350 Hz. For musical instruments, the fundamental frequency may vary a lot, and in some cases the fundamental frequency may not be present in the frequency spectrum, although the ear can have the ability to perceive it, by processing the information provided by the higher harmonics. This is a *psychoacoustics* phenomenon. Psychoacousticians as well as musicologists use the term *pitch* to define the frequency *perceived* by the ear, which in some cases may even be different from the fundamental.

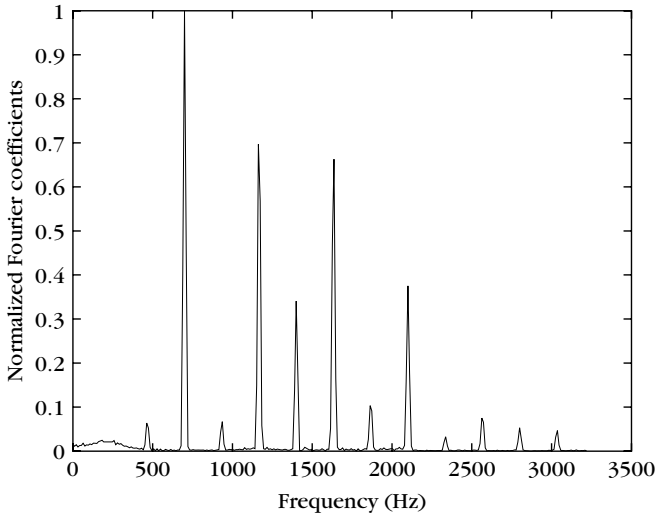


FIGURE 7.23

Normalized DFT coefficients of a clarinet sound, whose fundamental frequency is absent.

The fundamental frequency estimation is not an easy task, and a number of techniques have been proposed in the published literature, including [Schr 68, Brow 91, Wu 03, Tolo 00, Klap 03, Goto 04] and the references therein. Figure 7.23 is an example of the (normalized) amplitude of the DFT of a signal segment corresponding to a harmonic sound produced by a clarinet. The length of the frame is 4096 samples long. The spectrum consists of the regularly spaced harmonics of the fundamental frequency, which is equal to 230 Hz but is missing from the spectrum. Note that, listening to this sound, a trained ear will perceive that the pitch of this signal is indeed 230 Hz. It can be observed that odd and even multiples of the fundamental frequency are present as peaks. (For this frame, it turned out that the amplitudes of the odd multiples are considerably smaller than the amplitudes of the even multiples of the fundamental frequency.) Application of the algorithm given in [Schr 68], for the estimation of the fundamental frequency, returns the true value of 230 Hz. The method builds on the idea of exploiting the greatest common divisor of all peaks present in the spectrum.

7.5.5 Time Domain Features

Zero-Crossing Rate

The zero-crossing rate is defined as

$$Z(i) = \frac{1}{2N} \sum_{n=0}^{N-1} |\text{sgn}[x_i(n)] - \text{sgn}[x_i(n-1)]|$$

where

$$\text{sgn}[x_i(n)] = \begin{cases} 1 & x_i(n) \geq 0 \\ -1 & x_i(n) < 0 \end{cases}$$

This is a measure of the noisiness of the signal. Thus, unvoiced speech signals have higher zero-crossing values compared to the voiced ones. Temporal curves of variation of the zero-crossing rate from frame to frame may also be informative of the type of signal.

Energy

This is a very simple feature, defined as

$$E(i) = \frac{1}{N} \sum_{n=0}^{N-1} |x_i(n)|^2$$

and it can be used to discriminate voiced from unvoiced speech signals, since the latter tend to have much lower energy values than the former. It is also useful to discriminate silent periods in a recording and can be useful during the segmentation process.

All features discussed in this section are also known as *frame-level* features. They provide local information with respect to individual frames, and their goal is to capture short-term characteristics. However, to extract semantic content information, one needs to follow how the previously cited features change from frame to frame over a longer time scale. To this end, one can develop various methods to quantify this variation. In [Tzan 02], the mean and variances of the frame-level features have been used as features for music genre classification.

Besides the above features, other features—such as wavelet coefficients, fractal dimension, AR modeling, and independent components (ICA), presented in the previous and the present chapter—are also popular candidates. For music signals, based on early studies on the human perception of pitch, it has been proposed to use a 12-element representation of the spectral energy of a music signal, known as the chroma vector [Bart 05]. Each element of the vector corresponds to one of the 12 traditional pitch classes (i.e., 12 notes) of the equal-tempered scale of Western music. The chroma vector can encode and represent harmonic relationships within a particular music signal.

7.5.6 An Example

To demonstrate the classification power of two of the previously discussed features, let us take a simple example. Figure 7.24 shows the variation of the zero-crossing rate from frame to frame as time evolves, for a clapping sound. The sampling frequency was 44.1 KHz, and the length of each frame was equal to 1024 samples, with a successive frame overlap of 512 samples. For each frame the Hamming window was used. One can observe the noisy look of the resulting graph, with a large change of the feature values from frame to frame. This noisy nature of the clapping sound can also be revealed if the fundamental frequency is adopted as a

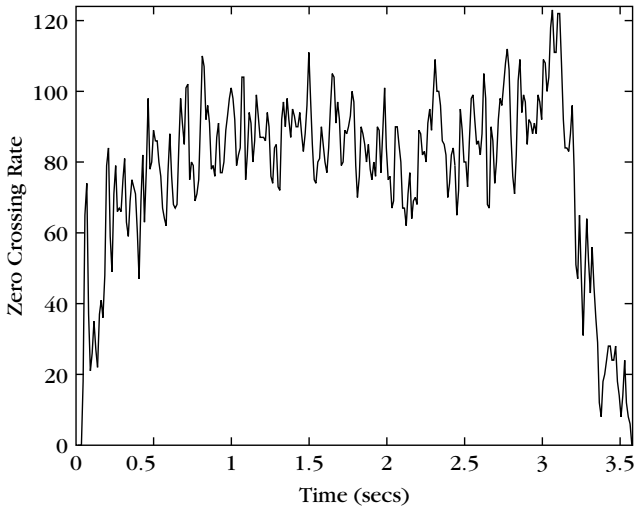


FIGURE 7.24

Zero-crossing rate results for a clapping sound recording, using a Hamming moving window technique.

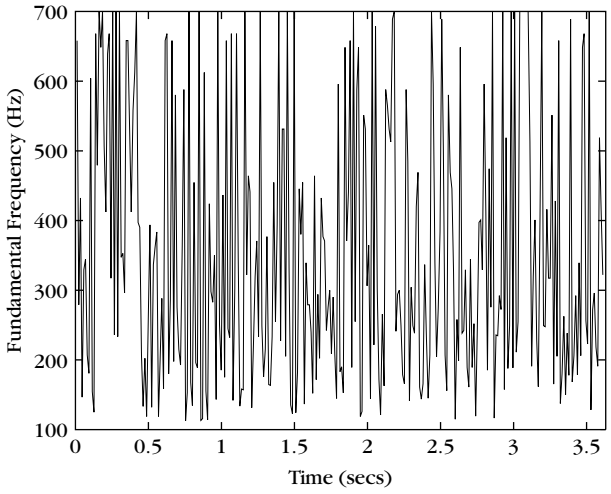


FIGURE 7.25

Fundamental frequency tracking results for the same clapping sound recording as in Figure 7.24.

feature. Figure 7.25 shows the change of the fundamental frequency from frame to frame. The algorithm used for the fundamental frequency tracking was that proposed in [Brow 91]. In contrast to the noisy nature of the previous audio recording, Figures 7.26 and 7.27 show the graphs of the change, from frame to frame, of the

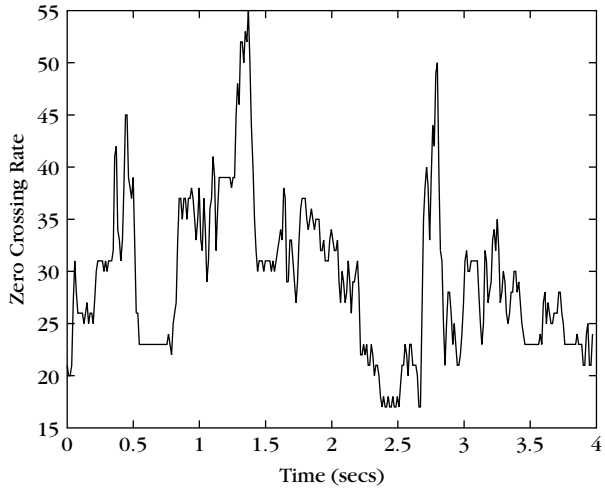


FIGURE 7.26

Zero-crossing rate results for the piano music recording, using a Hamming moving window technique.

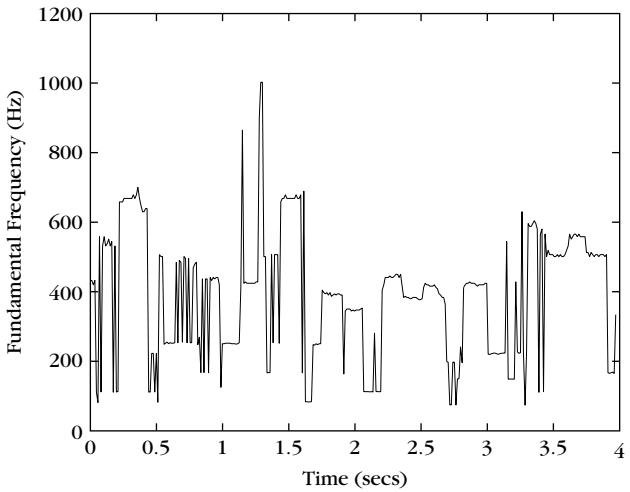


FIGURE 7.27

Fundamental frequency tracking results for the same piano music recording as in Figure 7.26.

zero-crossing rate and fundamental frequency, respectively, for a piano recording (from Bach's English suite in A major). The frame parameters used for the analysis were the same as before. One can observe that both graphs are now much less noisy. The variation from frame to frame is much smaller, and at some points of the graph the values of both features remain almost constant for relatively long periods of time, revealing the "structured" harmonic nature of the analyzed sound.

7.6 PROBLEMS

7.1 Consider the image array

$$I = \begin{bmatrix} 0 & 1 & 2 & 2 & 3 \\ 1 & 2 & 0 & 2 & 0 \\ 3 & 0 & 3 & 2 & 1 \\ 1 & 2 & 2 & 2 & 3 \\ 0 & 0 & 1 & 1 & 2 \end{bmatrix}$$

Compute the co-occurrence matrix for $d = 1$ and the four directions. Then compute the ASM and CON features.

7.2 For the image array of the previous problem compute the run length matrix for the four directions and then the features SRE, LRE.

7.3 Construct a 4×4 array that will have a high contrast (CON) value in the $0°$ direction and low CON value in the $45°$ direction.

7.4 For two of the test images provided in the Web site of the book, develop a program that computes the co-occurrence and run length matrices for the four directions and $d = 1$. Then compute the $ASM, CON, IDF, H_{xy}, SRE, LRE, GLNU, RLN$ features and average their respective values over the four directions. Justify your findings.

7.5 Show that if

$$\sum_{i=1}^{N} P_i = 1$$

then

$$S = \sum_{i=1}^{N} P_i^2$$

becomes minimum if $P_i = \frac{1}{N}, i = 1, 2, \ldots, N$.

7.6 Show that the central moments defined in (7.22) are invariant to translations and the normalized central moments in (7.23) are invariant to both translation and scaling.

7.7 Show that the Zernike polynomials are orthogonal on the unit circle disk, that is,

$$\iint_{x^2+y^2\le 1} V^*_{nm}(x,y)V_{pq}(x,y)\,dx\,dy = \frac{\pi}{n+1}\delta_{np}\delta_{mq}$$

7.8 Show that if a region in an image is rotated by an angle θ with respect to the origin, the Zernike moments of the rotated region are related to the unrotated ones by

$$A'_{pq} = A_{pq}\exp(-jq\theta)$$

7.9 Write a program to compute the moments of Hu. Then apply it to two of the test images involving objects, available from the web, and compute the respective moments.

7.10 Repeat Problem 7.9 for the Zernike moments of order A_{11}, A_{20}, A_{02}.

7.11 Show Eq. (7.37).

7.12 Write a program that computes the AR parameters for a noncausal prediction model. Apply it to a homogeneous and isotropic image whose autocorrelation sequence is given by

$$r(k,l) = \exp(-\sqrt{k^2+l^2})$$

for a window W of order $p = q = 1$.

7.13 Let $u_k = x_k + jy_k$, where (x_k, y_k) are the coordinates of the points on the boundary of an object in an image. Show that if the object is rotated by an angle θ with respect to the origin of the axis, the new complex sequence is

$$u'_k = u_k\exp(j\theta)$$

7.14 Let t denote the length along a closed boundary curve measured from an origin within the curve, and $x(t), y(t)$ the coordinates as functions of t. If T is the total length of the curve, then the following Fourier series expansion holds.

$$x(t) = a_0 + \sum_{n=1}^{\infty}\left[a_n\cos\frac{2\pi nt}{T} + b_n\sin\frac{2\pi nt}{T}\right]$$

$$y(t) = c_0 + \sum_{n=1}^{\infty}\left[c_n\cos\frac{2\pi nt}{T} + d_n\sin\frac{2\pi nt}{T}\right]$$

Prove that if $x(t), y(t)$ are piecewise linear functions between the sampled points $(x(t), y(t)), t = 0, 1, \ldots, m-1$, the Fourier coefficients a_n, b_n, c_n, d_n are given by

$$a_n = \frac{T}{2\pi^2 n^2} \sum_{i=1}^{m} \frac{\Delta x_i}{\Delta t_i} [\cos \phi_i - \cos \phi_{i-1}]$$

$$b_n = \frac{T}{2\pi^2 n^2} \sum_{i=1}^{m} \frac{\Delta x_i}{\Delta t_i} [\sin \phi_i - \sin \phi_{i-1}]$$

$$c_n = \frac{T}{2\pi^2 n^2} \sum_{i=1}^{m} \frac{\Delta y_i}{\Delta t_i} [\cos \phi_i - \cos \phi_{i-1}]$$

$$d_n = \frac{T}{2\pi^2 n^2} \sum_{i=1}^{m} \frac{\Delta y_i}{\Delta t_i} [\sin \phi_i - \sin \phi_{i-1}]$$

where

$$\Delta x_i = x_i - x_{i-1}, \quad \Delta y_i = y_i - y_{i-1}$$

$$\Delta t_i = \sqrt{\Delta x_i^2 + \Delta y_i^2}, \quad t_i = \sum_{j=1}^{i} \Delta t_j$$

$$T = t_m, \quad \phi_i = \frac{2n\pi t_i}{T}$$

7.15 For the Fourier coefficients in Problem 7.14, prove that the following parameters are rotation invariant:

$$I_n = a_n^2 + b_n^2 + c_n^2 + d_n^2$$

$$J_n = a_n d_n - b_n c_n$$

$$K_{1,n} = (a_1^2 + b_1^2)(a_n^2 + b_n^2) + (c_1^2 + d_1^2)(c_n^2 + d_n^2) + 2(a_1 c_1 + b_1 d_1)(a_n c_n + b_n d_n)$$

7.16 If $(x(t), y(t))$ are defined as in Problem 7.14 and

$$z(t) = x(t) + jy(t)$$

the respective complex exponential Fourier series is given as

$$z(t) = \sum_{n=-\infty}^{\infty} a_n \exp(j2\pi nt/T)$$

$$a_n = \frac{1}{T} \int_0^T z(t) \exp(-j2\pi nt/T)\, dt$$

Prove that the following parameters are scale and rotation invariant [Gran 72]:

$$b_n = \frac{a_{1+n}a_{1-n}}{a_1^2}, \quad d_{mn} = \frac{a_{1+m}^n a_{1-n}^m}{a_1^{(m+n)}}$$

where $n \neq 1$.

7.17 Prove that if $x(t), y(t)$ of the previous problem are piecewise linear functions between the points $(x(t), y(t)), t = 0, 1, \ldots, m-1$, then the Fourier coefficients a_n are given by [Lai 81]

$$a_n = \frac{T}{(2\pi n)^2} \sum_{i=1}^{m} (b_{i-1} - b_i) \exp(-jn2\pi t_i/T)$$

where

$$b_i = \frac{V_{i+1} - V_i}{|V_{i+1} - V_i|}, \quad t_i = \sum_{k=1}^{i} |V_k - V_{k-1}|, \quad i > 0, t_0 = 0$$

and $V_i, i = 1, 2, \ldots, m$, the phasors at the respective points.

7.18 Show that the orientation θ in Section 7.3.3 results from minimizing

$$I(\theta) = \sum_i \sum_j [(i - \bar{x}) \cos\theta - (j - \bar{y}) \sin\theta]^2$$

7.19 Show that the power spectrum of an fdm process with Hurst parameter H is given by

$$S(f) \propto \frac{1}{f^{(2H+1)}}$$

7.20 Show that the definition of M in (7.46) results in a consistent metric for the Koch curve.

7.21 Assuming that $x(0) = 0$, show that

$$E[x(n)(x(n + n_0) - x(n))] = \frac{1}{2}\left\{(n + n_0)^{2H} - n^{2H} - n_0^{2H}\right\}$$

For the case of a Brownian motion ($H = \frac{1}{2}$) this suggests that $x(n)$ is uncorrelated to the increment. This is not true for $H \neq \frac{1}{2}$, where a nonzero correlation exists, *positive* for $H > 1/2$ and *negative* for $H < 1/2$. To prove this, use, the definition in (7.51). This can be generalized. That is, if $n_1 \leq n_2 \leq n_3 \leq n_4$ and the process is Brownian then

$$E[(x(n_2) - x(n_1))(x(n_4) - x(n_3))] = 0$$

MATLAB PROGRAMS AND EXERCISES

Computer Exercises

7.1 *First-order image statistics.* Write a MATLAB function named *first_order_stats* that computes the first-order statistics of a set of images. Specifically, the function takes as inputs (a) a *num_in* $\times$ *q* dimensional array *name_images*, whose *i*th row contains the name of the *i*th image file, (b) the number *N_gray* specifying the range [0, *N_gray* $-$ 1] in which the intensity of the pixels will be scaled. It returns a *num_im* $\times$ 4 dimensional matrix *features*, the *i*th row of which contains the mean, the standard deviation, the skewness and the kurtosis of the intensity of the pixels of the *i*th image.

Solution

In practice, it is more convenient to work with images where the intensity of the pixels lie in a rather small range of values (e.g. [0, 31]), in order to produce smoother histograms and smaller (nonsparse and easier to handle) co-occurence matrices (see next program). This is the reason for the *N_gray* input argument in the functions of this section.

```
function features=first_order_stats(name_images,N_gray)
  [num_im,q]=size(name_images);
  features=zeros(num_im,4);
  for i=1:num_im
    A=imread(name_images(i,:));
    A=double(A);
    %Normalization of the pixels intensity in [0,N_gray-1]
    A=round((N_gray-1)*((A-min(A(:)))/(max(A(:))-min(A(:)))));
    features(i,1)=mean2(A);
    features(i,2)=std2(A);
    features(i,3)=skewness(A(:));
    features(i,4)=kurtosis(A(:));
  end
```

7.2 *Second-order image statistics.* Write a MATLAB function named *second_order_stats* that computes second-order statistics of a set of images. Specifically, the function takes as inputs: (a) a *num_in*$\times$*q* dimensional array *name_images*, whose *i*th row contains the name of the *i*th image file, (b) the number *N_gray* specifying the range [0, *N_gray* $-$ 1] in which the intensity of the pixels will be scaled. For each image, four co-occurence matrices determined by the pixel pairs that are at relative positions $(1, 0)°$, $(1, 45°)$, $(1, 90°)$, $(1, 135°)$ should be computed. The function returns a *num_im* $\times$ 8 dimensional matrix *features*, the *i*th row of which contains (a) *in its first four entries*: the means of the contrast, the correlation, the angular second moment (in MATLAB termed "Energy") and the inverse difference moment (in MATLAB termed

"Homogeneity") computed from the four co-occurence matrices of the *i*th image and (b) *in its last four entries*: the ranges of the values of the previous features of the *i*th image.

Solution

In the following implementation, we make use of the *graycomatrix* and *graycoprops* built-in MATLAB functions. The first computes the co-occurence matrices of an image, while the second is applied on co-occurence matrices and computes the features 'Contrast','Correlation','Energy','Homogeneity'.

```
function features=second_order_stats(name_images,N_gray)
  [num_im,q]=size(name_images);
  features=zeros(num_im,8);
  for i=1:num_im
    A=imread(name_images(i,:));
    A=double(A);
    %Normalization of the pixels intensity in [0, N_gray-1]
    A=round((N_gray-1)*((A-min(A(:)))/(max(A(:))-min(A(:)))));
    [glcm,SI]=graycomatrix(A,'GrayLimits',[0,N_gray-1],...
    'NumLevels',...
    N_gray,'Offset',[0 1;-1 0;-1 1; -1 -1],'Symmetric',true);
    stats=graycoprops(glcm,{'Contrast','Correlation',...
    'Energy','Homogeneity'});
    features(i,1)=mean(stats.Contrast);
    features(i,2)=mean(stats.Correlation);
    features(i,3)=mean(stats.Energy);
    features(i,4)=mean(stats.Homogeneity);
    features(i,5)=range(stats.Contrast);
    features(i,6)=range(stats.Correlation);
    features(i,7)=range(stats.Energy);
    features(i,8)=range(stats.Homogeneity);
  end
```

7.3 *Second-order image statistics (masks).* Write a MATLAB function named *mask_order_stats* that takes as input a set of original images. Each one of them is convolved with nine masks, and for each one of the nine resulting images first order statistics are computed. Specifically, the function takes as inputs: (a) a *num_in* $\times$ *q* dimensional array *name_images*, whose *i*th row contains the name of the *i*th original image file, (b) the number *N_gray* specifying the range [0, *N_gray* $-$ 1] in which the intensity of the pixels will be scaled. The function should convolve each one of the original images with each one of the nine masks, defined in Section 7.2.2, producing nine (convolved) images for each original image. Then, the first-order statistics (mean, standard deviation, skewness, kurtosis) for each one of the convolved images is computed. The function returns a *num_in* $\times$ 4 $\times$ 9 three-dimensional matrix *features*, where

its *i*th *num_im* $\times$ 4 two-dimensional component corresponds to the results produced when the *i*th mask is applied to each one of the original images. Each one of the two-dimensional components is defined as in the *first_order_stats* function.

Solution

```
function features=mask_stats(name_images,N_gray)
  [num_im,q]=size(name_images);
  features=zeros(num_im,4,9);
  %Definition of the masks
  mask(:,:,1)=[1 2 1; 2 4 2; 1 2 1];
  mask(:,:,2)=[-1 0 1; -2 0 2; -1 0 1];
  mask(:,:,3)=[-1 2 -1; -2 4 -2; -1 2 -1];
  mask(:,:,4)=[-1 -2 -1; 0 0 0; 1 2 1];
  mask(:,:,5)=[1 0 -1; 0 0 0; -1 0 1];
  mask(:,:,6)=[1 -2 1; 0 0 0; -1 2 -1];
  mask(:,:,7)=[-1 -2 -1; 2 4 2; -1 -2 -1];
  mask(:,:,8)=[1 0 -1; -2 0 2; 1 0 -1];
  mask(:,:,9)=[1 -2 1; -2 4 -2; 1 -2 1];
  % The following is useful in normalizing the convolution result
  sum_mask=sum(sum(mask))+(sum(sum(mask))==0);
  for i=1:num_im
  A=imread(name_images(i,:));
  A=double(A);
  %Normalization of the pixels intensity in [0, N_gray-1]
  A=round((N_gray-1)*((A-min(A(:)))/(max(A(:))-min(A(:)))));
  for j=1:9
  B=conv2(A,mask(:,:,j),'same')/sum_mask(j);
  features(i,1,j)=mean2(B);
  features(i,2,j)=std2(B);
  features(i,3,j)=skewness(B(:));
  features(i,4,j)=kurtosis(B(:));
  end
  end
```

Computer Experiments

Notes:

- All filenames included in the rows of the *name_images* array should have the same number of characters.
- Test images on which the above programs can be applied can be found in www.elsevierdirect.com/9781597492720 ('*ROI*_01_*seeds.bmp*', '*ROI*_02_*seeds.bmp*', . . ., '*ROI*_10_*seeds.bmp*'). In the sequel, this set of images is called "set of seeds".

7.1 Compute the first-order statistics for the set of seeds, using $N\_gray = 32$ and comment on the results.

7.2 Compute the second-order statistics for the set of seeds, using $N\_gray = 32$ and comment on the results.

7.3 Compute the first-order statistics of the nine images produced by each image of the set of seeds, after its convolution with each one of the nine masks, given in Section 7.2.2 and comment on the results. Use $N\_gray = 32$.

REFERENCES

[Alem 90] Al-Emami S., Usher M. "On-line recognition of handwritten Arabic characters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 12(7), pp. 704–710, 1990.

[Arbt 89] Arbter K. "Affine-invariant Fourier descriptors," in *From Pixel to Features* (Simon J.C., ed.), pp. 153–164, Elsevier, 1989.

[Arbt 90] Arbter K., Snyder W.E., Burkhardt H., Hirzinger G. "Application of affine-invariant Fourier descriptors to recognition of 3-D objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 12, pp. 640–647, 1990.

[Ardu 92] Ardunini F., Fioravanti S., Giusto D.D., Inzirillo F. "Multifractals and texture classification," *IEEE International Conference on Image Processing*, pp. 454–457, 1992.

[Ayac 00] Ayache A., Véhel J.L. "The generalized multifractional Brownian motion," *Statistical Inference for Stochastic Processes*, Vol. 3, pp. 7–18, 2000.

[Bart 05] Bartch M., Wakefield G.H. "Audio thumbnailing of popular music using chroma-based representations," *IEEE Transactions on Multimedia*, Vol. 7(1), pp. 96–104, February 2005.

[Bass 92] Bassevile M., Benveniste A., Chou K., Golden S.A., Nikoukhah R., Willsky A.S. "Modeling and estimation of multiresolution stochastic processes," *IEEE Transactions on Information Theory*, Vol. 38, pp. 766–784, 1992.

[Brod 66] Brodatz P. *Textures—A Photographic Album for Artists and Designers*, Dover, 1966.

[Brow 91] Brown J.C., Zhang B. "Musical frequency tracking using the methods of conventional and narrowed autocorrelation," *Journal of the Acoustical Society of America*, Vol. 89(5), 1991.

[Cavo 92] Cavouras D., Prassopoulos P., Pantelidis N. "Image analysis methods for solitary pulmonary nodule characterization by CT," *European Journal of Radiology*, Vol. 14, pp. 169–172, 1992.

[Chel 85] Chellapa R. "Two dimensional discrete Gaussian Markov random field models for image processing," in *Progress in Pattern Recognition* (Kanal L.N., Rosenfeld A., eds.), Vol. 2, pp. 79–112, North Holland, 1985.

[Chen 89] Chen C.C., Daponee J.S., Fox M.D. "Fractal feature analysis and classification in medical imaging," *IEEE Transactions on Medical Imaging*, Vol. 8, pp. 133–142, 1989.

[Chon 03] Chong C.-W., Raveendran P., Mukundan R. " Translation invariants of Zernike moments," *Pattern Recognition*, Vol. 36, pp. 1765–1773, 2003.

[Chon 04] Chong C.-W., Raveendran P., Mukundan R. "Translation and scale invariants of Legendre moments," *Pattern Recognition*, Vol. 37, pp. 119–129, 2004.

[Clau 04] Clausen M., Kurth F. "A unified approach to content-based and fault-tolerant music recognition," *IEEE Transactions on Multimedia*, Vol. 6(5), pp. 717–731, October 2004.

[Crim 82] Crimmins T.R. "A complete set of Fourier descriptors," *IEEE Transactions on Systems Man and Cybernetics*, Vol. 12, pp. 236–258, 1982.

[Cros 83] Cross G.R., Jain A.K. "Markov random field texture models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 5(1), pp. 25–39, 1983.

[Davi 79] Davis L., Johns S., Aggrawal J.K. "Texture analysis using generalized co-occurrence matrices," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 1(3), pp. 251–259, 1979.

[Davi 80] Davis S.B., Mermelstein P. "Comparison of parametric representations of monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 28(4), pp. 357–366, 1980.

[Dell 00] Deller J.R., Hansen J.H.L., Proakis J.G. *Discrete Processing of Speech Signals*, John Wiley & Sons, New York, 2000.

[Deri 93] Deriche M., Tewfik A.H. "Signal modeling with filtered discrete fractional noise processes," *IEEE Transactions on Signal Processing*, Vol. 41, pp. 2839–2850, 1993.

[Falc 90] Falconer K. *Fractal Geometry: Mathematical Foundations and Applications*, John Wiley & Sons, 1990.

[Fieg 96] Fieguth P.W., Willsky A.S. "Fractal estimation using models on multiscale trees," *IEEE Transactions on Signal Processing*, Vol. 41, pp. 1297–1300, 1996.

[Flan 92] Flandrin P. "Wavelet analysis and synthesis of fractional Brownian motion," *IEEE Transactions on Information Theory*, Vol. 38, pp. 910–917, 1992.

[Flus 93] Flusser J., Suk T. "Pattern recognition by affine moment invariants," *Pattern Recognition*, Vol. 26(1), pp. 167–174, 1993.

[Flus 94] Flusser J., Suk T. "Affine moment invariants: A new tool for character recognition," *Pattern Recognition*, Vol. 15(4), pp. 433–436, 1994.

[Frag 01] Fragoulis D., Rousopoulos G., Panagopoulos T., Alexiou C., Papaodysseus C. "On the automated recognition of seriously distorted musical recordings," *IEEE Transactions on Signal Processing*, Vol. 49(4), pp. 898–908, 2001.

[Free 61] Freeman H. "On the encoding of arbitrary geometric configurations," *IRE Transactions on Electronic Computers*, Vol. 10(2), pp. 260–268, 1961.

[Gall 75] Galloway M. "Texture analysis using gray-level run lengths," *Computer Graphics and Image Processing*, Vol. 4, pp. 172–179, 1975.

[Gewe 83] Geweke J., Porter-Hudak S. "The estimation and application of long memory time series," *Journal of Time Series Analysis*, Vol. 4, pp. 221–237, 1983.

[Ghos 97] Ghosal S., Mehrotra R. "A moment based unified approach to image feature detection," *IEEE Transactions on Image Processing*, Vol. 6(6), pp. 781–794, 1997.

[Glen 94] Glentis G., Slump C., Herrmann O. "An efficient algorithm for two-dimensional FIR filtering and system identification," *SPIE Proceedings*, VCIP, pp. 220–232, Chicago, 1994.

[Goto 04] Goto M. "A real-time music-scene-description system: Predominant - F0 estimation for detecting melody and bass lines in real-world audio signals," *Speech Communication (ISCA) Journal*, Vol. 43(4), pp. 311–329, 2004.

[Gran 72] Granlund G.H. "Fourier preprocessing for hand print character recognition," *IEEE Transactions on Computers*, Vol. 21, pp. 195–201, 1972.

[Hara 73] Haralick R., Shanmugam K., Distein I. "Textural features for image classification," *IEEE Transactions on Systems Man and Cybernetics*, Vol. 3(6), pp. 610–621, 1973.

[Hayk 96] Haykin S. *Adaptive Filter Theory*, 3rd ed., Prentice Hall, 1996.

[Heyw 95] Heywodd M.I., Noakes P.D. "Fractional central moment method for movement-invariant object classification," *IEE Proceedings Vision, Image and Signal Processing*, Vol. 142 (4), pp. 213–219, 1995.

[Hu 62] Hu M.K. "Visual pattern recognition by moment invariants," *IRE Transactions on Information Theory*, Vol. 8(2), pp. 179–187, 1962.

[Huan 94] Huang Q., Lorch J.R., Dubes R.C. "Can the fractal dimension of images be measured?" *Pattern Recognition*, Vol. 27(3), pp. 339–349, 1994.

[Huan 06] Huang S.-K., Kim W.-Y. "A novel approach to the fast computation of Zernike moments," *Pattern Recognition*, Vol. 39(11), pp. 2065–2076, 2006.

[Kalo 89] Kalouptsidis N., Theodoridis S. "Concurrent algorithms for a class of 1-D and 2-D Wiener filters with symmetric impulse response," *IEEE Transactions on Signal Processing*, Vol. ASSP-37, pp. 1780–1782, 1989.

[Kan 02] Kan C., Srinath M.D. "Invariant character recognition with Zernike moments and orthogonal Fourier-Mellin moments," *Pattern Recognition*, Vol. 35, pp. 143–154, 2003.

[Kapl 99] Kaplan L.M. "Extended fractal analysis for the texture classification and segmentation," *IEEE Transactions on Image Processing*, Vol. 8(11), pp. 1572–1585, 1999.

[Kapl 94] Kaplan L.M., Kuo C.-C.J. "Extending self similarity for fractional Brownian motion," *IEEE Transactions on Signal Processing*, Vol. 42(12), pp. 3526–3530, 1994.

[Kapl 95] Kaplan L.M., Kuo C.C.J. "Texture roughness analysis and synthesis via extended self-similar model," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 17(11), pp. 1043–1056, 1995.

[Kara 95] Karayannis Y.A., Stouraitis T. "Texture classification using fractal dimension as computed in a wavelet decomposed image," *IEEE Workshop on Nonlinear Signal and Image Processing*, pp. 186–189, Neos Marmaras, Halkioliki, June 95.

[Kash 82] Kashyap R.L., Chellapa R., Khotanzad A. "Texture classification using features derived from random field models," *Pattern Recognition Letters*, Vol. 1, pp. 43–50, 1982.

[Kash 86] Kashyap R.L., Khotanzad A. "A model based method for rotation invariant texture classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 8(4), pp. 472–481, 1986.

[Kell 87] Keller J.M., Crownover R., Chen R.Y. "Characteristics of natural scenes related to fractal dimension," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 9, pp. 621–627, 1987.

[Khot 90a] Khotanzad A., Hong Y.H. "Invariant image recognition by Zernike moments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 12(5), pp. 489–497, 1990.

[Khot 90b] Khotanzad A., Lu J.H. "Classification of invariant image representations using a neural network," *IEEE Transactions on Acoustics Speech and Signal Processing*, Vol. 38(6), pp. 1028–1038, 1990.

[Klap 03] Klapuri A. "Multiple fundamental frequency estimation by harmonicity and spectral smoothness," *IEEE Transactions on Speech and Audio Processing*, Vol. 11(6), pp. 804–816, 2003.

[Kuhl 82] Kuhl F.P., Giardina C.R. "Elliptic Fourier features of a closed contour," *Comput. Vis. Graphics Image Processing*, Vol. 18, pp. 236–258, 1982.

[Lai 81] Lai M., Suen Y.C. "Automatic recognition of characters by Fourier descriptors and boundary line encoding," *Pattern Recognition*, Vol. 14, pp. 383–393, 1981.

[Laws 80] Laws K.I. "Texture image segmentation" Ph.D. Thesis, University of Southern California, 1980.

[Liao 96] Liao S., Pawlak M. "On image analysis by moments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 18(3), pp. 254–266, March 1996.

[Lin 87] Lin C.S., Hwang C.L. "New forms of shape invariants from elliptic Fourier descriptors," *Pattern Recognition*, Vol. 20(5), pp. 535–545, 1987.

[Lu 91] Lu S.Y., Ren Y., Suen C.Y. "Hierarchical attributed graph representation and recognition of handwritten Chinese characters," *Pattern Recognition*, Vol. 24(7), pp. 617–632, 1991.

[Lund 86] Lundahl T., Ohley W.J., Kay S.M., Siffer R. "Fractional Brownian motion: A maximum likelihood estimator and its application to image texture," *IEEE Transactions on Medical Imaging*, Vol. 5, pp. 152–161, 1986.

[Mahm 94] Mahmoud S. "Arabic character recognition using Fourier descriptors and character contour encoding," *Pattern Recognition*, Vol. 27(6), pp. 815–824, 1994.

[Mami 98] Mamistvalov A.G. "n-Dimensional moment invariants and the conceptual mathematical theory of recognition n-dimensional objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20(8), pp. 819–831, 1998.

[Mand 68] Mandelbrot B.B, Van Ness J.W. "Fractional Brownian motion, fractional noises and applications," *SIAM Review*, Vol. 10, pp. 422–437, 1968.

[Mand 77] Manderbrot B.B. *The Fractal Geometry of Nature*, W.H. Freeman, New York, 1982.

[Mao 92] Mao J., Jain A.K. "Texture classification and segmentation using multiresolution simultaneous autoregressive models," *Pattern Recognition*, Vol. 25(2), pp. 173–188, 1992.

[Mara 93] Maragos P., Sun F.K. "Measuring the fractal dimension of signals: Morphological covers and iterative optimization," *IEEE Transactions on Signal Processing*, Vol. 41, pp. 108–121, 1993.

[Mori 92] Mori S., Suen C. "Historical review of OCR research and development," *Proceedings of IEEE*, Vol. 80(7), pp. 1029–1057, 1992.

[Muku 95] Mukundan R., Ramakrshnan J. "Fast computation of Legendre and Zernike moments," *Pattern Recognition*, Vol. 28(9), pp. 1433–1442, 1995.

[Muku 98] Mukundan R., Ramakrshnan J. *Moment Functions in Image Analysis–Theory and Applications*, World Scientific, Singapore, 1998.

[Ohan 92] Ohanian P., Dubes R. "Performance evaluation for four classes of textural features," *Pattern Recognition*, Vol. 25(8), pp. 819–833, 1992.

[Ojal 96] Ojala T., Pietikainen M., Harwood D. "A comparative study of texture measures with classification based on feature distributions," *Pattern Recognition*, Vol. 29(1), pp. 51–59, 1996.

[Papo 91] Papoulis A. *Probability, Random Variables, and Stochastic Processes*, 3rd ed., McGraw-Hill, 1991.

[Pele 84] Peleg S., Naor J., Hartley R., Anvir D. "Multiple resolution texture analysis and classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 6, pp. 818–523, 1984.

[Penn 97] Penn A.I., Loew M.H. "Estimating fractal dimension with fractal interpolation function models," *IEEE Transcations on Medical Imaging*, Vol. 16, pp. 930–937, 1997.

[Pent 84] Pentland A. "Fractal based decomposition of natural scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 6(6), pp. 661–674, 1984.

[Pers 77] Persoon E., Fu K.S. "Shape discrimination using Fourier descriptors," *IEEE Transactions on Systems Man and Cybernetics*, Vol. 7, pp. 170–179, 1977.

[Petr 06] Petrou M., Sevilla P.G. *Image Processing: Dealing with Texture*, John Wiley & Sons, 2006.

[Pesq 02] Pesquet-Popescu B., Vehel J.L. "Stochastic fractal models for image processing," *IEEE Signal Processing Magazine*, Vol. 19(5), pp. 48–62, 2002.

[Pico 93] Picone J. "Signal modeling techniques in speech recognition," *Proceedings of the IEEE*, Vol. 81(9), pp. 1215–1247, 1993.

[Pikr 03] Pikrakis A., Theodoridis S., Kamarotos D. "Recognition of isolated musical patterns using context dependent dynamic time warping," *IEEE Transactions on Speech and Audio Processing*, Vol. 11(3), pp. 175–183, 2003.

[Pikr 06] Pikrakis A., Theodoridis S., Kamarotos D. "Classification of musical patterns using variable duration hidden Markov models," *IEEE Transactions on Speech and Audio Processing*, to appear in 2006.

[Pikr 08] Pikrakis A., Gannakopoulos T., Theodoridis S. "A speech-music discriminator of radio recordings based on dynamic programming and Bayesian networks," *IEEE Transactions on Multimedia*, Vol. 10(5), pp. 846–856, 2008.

[Pita 94] Pitas I. *Image Processing Algorithms*, Prentice Hall, 1994.

[Plam 00] Plamondon R., Srihari S.N. "On-line and off-line handwriting recognition: A comprehensive survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22(1), pp. 63–84, 2000.

[Proa 92] Proakis J., Manolakis D. *Digital Signal Processing: Principles, Algorithms, and Applications*, 2nd ed., Macmillan, 1992.

[Rabi 93] Rabiner L., Juang B.H. *Fundamentals of Speech Recognition*, Prentice Hall, Englewood Cliffs, NJ, 1993.

[Rabi 78] Rabiner L.R., Schafer R.W. *Digital Processing of Speech Signals*, Prentice Hall, 1978.

[Rand 99] Randen T., Husoy H.H. "Filtering for texture classification: A comparative study," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 21(4), pp. 291–310, 1999.

[Reis 91] Reiss T.H. "The revised fundamental theorem of moment invariants," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13, pp. 830–834, 1991.

[Rich 95] Richardson W. "Applying wavelets to mammograms," *IEEE Engineering in Medicine and Biology*, Vol. 14, pp. 551–560, 1995.

[Sark 97] Sarkar A., Sharma K.M.S., Sonak R.V. "A new approach for subset 2-D AR model identification for describing textures," *IEEE Transactions on Image Processing*, Vol. 6(3), pp. 407–414, 1997.

[Saup 91] Saupe D. "Random fractals in image processing," in *Fractals and Chaos* (Crilly A.J., Earnshaw R.A., Jones H., eds.), pp. 89–118, Springer-Verlag, 1991.

[Schr 68] Schroeder M.R. "Period histogram and product spectrum: New methods for fundamental frequency measurement," *Journal of Acoustical Society of America*, Vol. 43(4), pp. 829–834, 1968.

[Sing 06] Singh C. "Improved quality of reconstructed images using floating point arithmetic for moment calculation," *Pattern Recognition*, Vol. 39(11), pp. 2047–2064, 2006.

[Tamu 78] Tamura H., Mori S., Yamawaki T. "Textural features corresponding to visual Perception," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 8(6), pp. 460–473, 1978.

[Tang 98] Tang X. "Texture information in run-length matrices," *IEEE Transactions on Image Processing*, Vol. 7(11), pp. 1602–1609, 1998.

[Taxt 90] Taxt T., Olafsdottir J.B., Daechlen M. "Recognition of hand written symbols," *Pattern Recognition*, Vol. 23(11), pp. 1155–1166, 1990.

[Teag 80] Teague M. "Image analysis via the general theory of moments," *Journal of Optical Society of America*, Vol. 70(8), pp. 920–930, 1980.

[Teh 88] Teh C.H., Chin R.T. "On image analysis by the method of moments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 10(4), pp. 496–512, 1988.

[Theo 93] Theodoridis S., Kalouptsidis N. "Spectral analysis," in *Adaptive System Identification and Signal Processing Algorithms* (Kalouptsidis N., Theodoridis S., eds.), Prentice Hall, 1993.

[Tolo 00] Tolonen T., Karjalainen M. "A computationally efficient multipitch analysis model," *IEEE Transactions on Speech and Audio Processing*, Vol. 8(6), pp. 708–716, November 2000.

[Trie 95] Trier O.D., Jain A.K. "Goal-directed evaluation of binarization methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 17(12), pp. 1191–1201, 1995.

[Trie 96] Trier O.D., Jain A.K., Taxt T. "Feature extraction methods for character recognition—A survey," *Pattern Recognition*, Vol. 29(4), pp. 641–661, 1996.

[Tson 92] Tsonis A. *Chaos: From Theory to Applications*, Plenum Press, 1992.

[Tzan 02] Tzanetakis G., Cook P. "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, Vol. 10(5), pp. 293–302, 2002.

[Unse 86] Unser M. "Local linear transforms for texture measurements," *Signal Processing*, Vol. 11, pp. 61–79, 1986.

[Unse 89] Unser M., Eden M. "Multiresolution feature extraction and selection for texture segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 11(7), pp. 717–728, 1989.

[Vinc 02] Vinciarelli A. "A survey on off-line cursive word recognition," *Pattern Recognition*, Vol. 35, pp. 1433–1446, 2002.

[Wang 98] Wang L., Healey G. "Using Zernike moments for the illumination and geometry invariant classification of multispectral textures," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 7(2), pp. 196–203, 1998.

[Wang 93] Wang L., Pavlidis T. "Direct gray-scale extraction of features for character recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 15(10), pp. 1053–1067, 1993.

[Wang 00] Wang Y., Huang J.C. "Multimedia content analysis," *IEEE Signal Processing Magazine*, Vol. 17(6), pp. 12–36, 2000.

[Wee 06] Wee C.-Y., Paramesran R. "Efficient computation of radial moment functions using symmetrical property," *Pattern Recognition*, Vol. 39(11), pp. 2036–2046, 2006.

[Wold 96] Wold E., Blum T., Keislar D., Wheaton J. "Content-based classification, search, and retrieval of audio," *IEEE Multimedia Magazine*, Vol. 22, pp. 27–36, 1996.

[Wood 72] Woods J. "Markov image modeling," *IEEE Transactions on Information Theory*, Vol. 18(3), pp. 232–240, 1972.

[Wood 96] Wood J. "Invariant pattern recognition," *Pattern Recognition*, Vol. 29(1), pp. 1–17, 1996.

[Worn 96] Wornell W.G. *Signal Processing with Fractals. A Wavelet Based Approach*, Prentice Hall, 1996.

[Wu 03] Wu M., Wang D., Brown G.J. "A multipitch tracking algorithm for noisy speech," *IEEE Transactions on Speech and Audio Processing*, Vol. 11(3), pp. 229–241, May 2003.

[Zhan 01] Zhang T., Kuo C.C.J. "Audio content analysis for online audiovisual data segmentation and classification," *IEEE Transactions on Speech and Audio Processing*, Vol. 9(4), pp. 441–458, 2001.