$$= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 10 \\ -2+j2 \\ -2 \\ -2-j2 \end{bmatrix}$$

■

Haar Transform

The Haar transform is one (of several) wavelet transforms that can be calculated with a formula. The formula (actually, two formulas) is given by

$$c_{00} = \int_0^1 v(t)\,\varphi_{00}(t)\,dt \qquad (1.13a)$$

$$d_{kj} = \int_0^1 v(t)\,\psi_{kj}(t)\,dt \qquad (1.13b)$$

where the basis functions φ_{00} and ψ_{kj} are as shown in Fig. 1.16. The function $\varphi_{00}(t)$ is called the *scaling function*, and the $\psi_{kj}(t)$ are called *wavelets*.

This diagram shows the level 0, 1, and 2 Haar basis functions, but these can continue to higher and higher levels. Level 0 contains the functions $\varphi_{00}(t)$ (called the *scaling function*) and $\psi_{00}(t)$ (called the *mother wavelet*). Level 1 contains the two functions $\psi_{10}(t)$ and $\psi_{11}(t)$. Level 2 contains the four functions $\psi_{20}(t)$, $\psi_{21}(t)$, $\psi_{22}(t)$, and $\psi_{23}(t)$. Each higher level contains twice as many wavelets $\{\psi_{jk}\}$. Thus, level 3 contains eight wavelets, each just half as long as the level 2 wavelets and scaled appropriately to have unit energy. To calculate the Haar transform, expand the time function in terms of these basis functions. Equation 1.13 gives the formulas for calculating the transform.

Example 1.10. Calculate the Haar transform of the signal in Fig. 1.17.

Solution: Applying Eq. 1.13 to the function v(t) gives the following numbers.

$$c_{00} = \int_0^1 v(t)\,\varphi_{00}(t)\,dt = 0$$

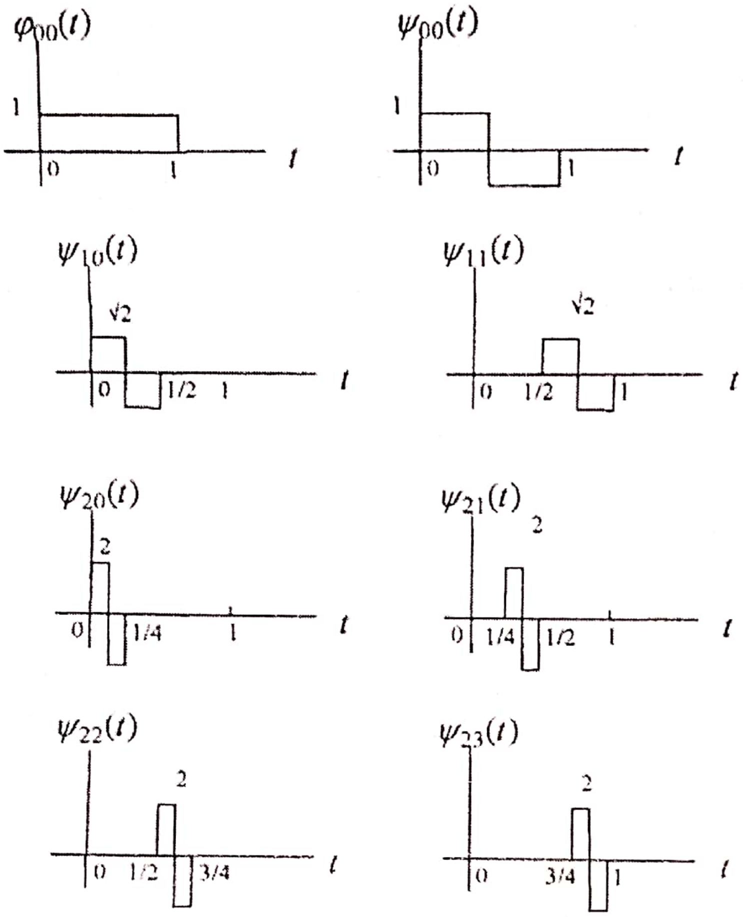


Figure 1.16. Haar basis functions to level 2.

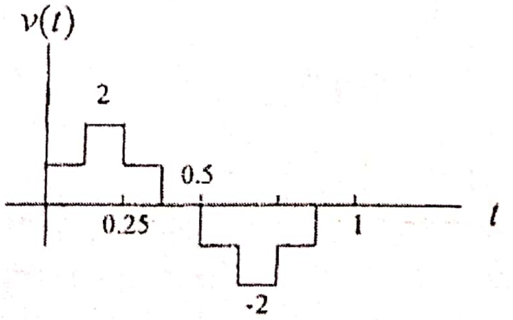


Figure 1.17. The function $v(t)$.

$$d_{00} = \int_0^1 v(t)\psi_{00}(t)\,dt = 1$$

$$d_{10} = \int_0^{0.5} v(t)\psi_{10}(t)\,dt = \frac{\sqrt{2}}{4}$$

$$d_{11} = \int_{0.5}^{1} v(t)\psi_{11}(t)\,dt = -\frac{\sqrt{2}}{4}$$

$$d_{20} = \int_0^{0.25} v(t)\psi_{20}(t)\,dt = -\frac{1}{4}$$

$$d_{21} = \int_{0.25}^{0.5} v(t)\psi_{21}(t)\,dt = \frac{1}{4}$$

$$d_{22} = \int_{0.5}^{0.75} v(t)\psi_{22}(t)\,dt = \frac{1}{4}$$

$$d_{23} = \int_{0.75}^{1} v(t)\psi_{23}(t)\,dt = -\frac{1}{4}$$

Thus, the first eight terms of the Haar transform of $v(t)$ consist of the eight values $\left\{0,\ 1,\ \frac{\sqrt{2}}{4},\ -\frac{\sqrt{2}}{4},\ -\frac{1}{4},\ \frac{1}{4},\ \frac{1}{4},\ -\frac{1}{4}\right\}$.

■

Equation 8.12 gives the equation for the inverse wavelet transform. When applied to the function in Fig. 1.17, it takes the form

$$v(t) = c_{00}\varphi_{00}(t) + d_{00}\psi_{00}(t) + d_{10}\psi_{10}(t) + \cdots + d_{23}\psi_{23}(t)$$

Example 1.11. Show that the inverse transform gives the function $v(t)$ in Fig. 1.17.

Solution: Multiplying the functions in Fig. 1.16 by the coefficients, $c_{00} = 0$, $d_{00} = 1$, and so on, results in the functions in Fig. 1.18. If these functions are summed, the result is $v(t)$ in Fig. 1.17. (Try it if you don't believe it.)

■

From these examples one can see many similarities between Fourier and wavelet transforms. Time functions are expanded in terms of basis functions in both transforms. These basis functions are exponentials for the Fourier transform, and they are square pulses for the Haar transform. The coefficients are calculated by an inner product operation [i.e., by integrating the product of $v(t)$ and basis function].

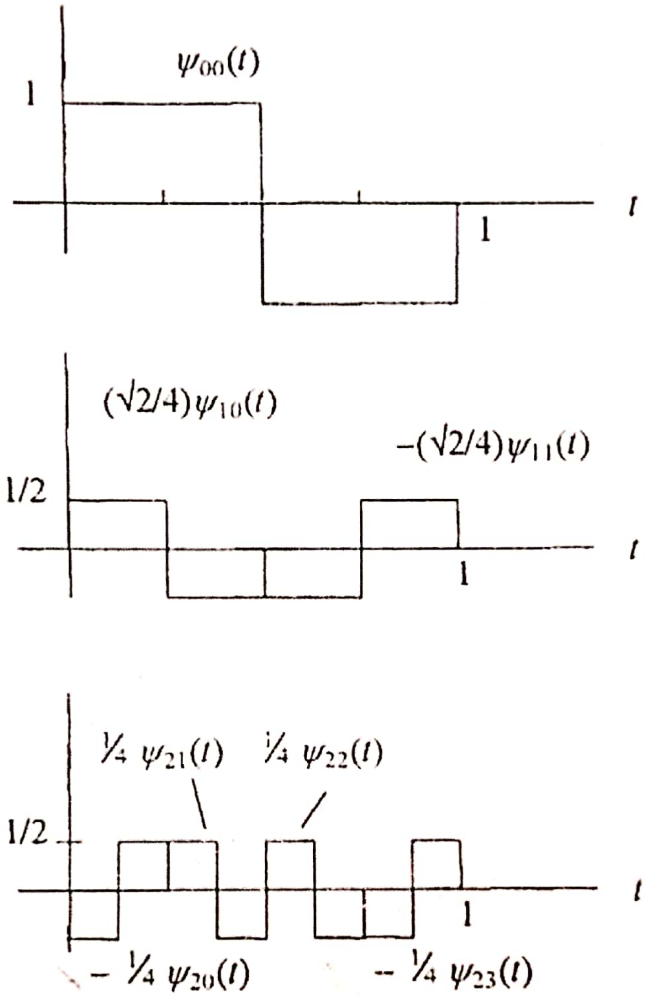


Figure 1.18. Sum these functions to obtain $v(t)$.

However, there are differences that are not apparent in these simple examples. There are many wavelet basis functions other than the Haar functions, and there is one wavelet transform for each set of basis functions. This is similar to the Fourier transforms, where there are four forms of the Fourier transform, but not similar in that there is a vast array of wavelet basis functions. Another important difference is in the way the coefficients for the wavelet transform are calculated. Note that in the above example using an inner product formulation, the formula for the basis function must be known. Most wavelet coefficients are calculated in a different way, using multirate sampling theory. There it is necessary to know only the filter coefficients. This method does not use the basis functions. We will devote much of this book to this procedure (see Section 6.2.3); using the outer boundary, compactness assumes values in the interval $[16, \infty)$. Independence from linear transformations is gained only if an outer boundary representation is used. Examples are shown in Figure 8.26.

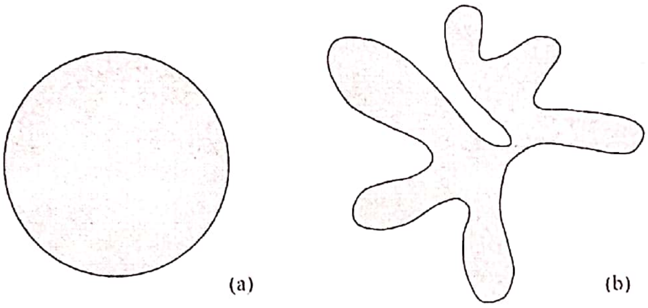


Figure 8.26: Compactness: (a) compact; (b) non-compact. © *Cengage Learning 2015.*

8.3.2 Moments

Region moment representations interpret a normalized gray-level image function as a probability density of a 2D random variable. Properties of this random variable can be described using statistical characteristics—**moments** [Papoulis, 1991]. Assuming that non-zero pixel values represent regions, moments can be used for binary or gray-level region description. A moment of order $(p+q)$ is dependent on scaling, translation, rotation, and even on gray-level transformations and is given by

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p \, y^q \, f(x, y) \, dx \, dy \, . \tag{8.41}$$

In digitized images we evaluate sums

$$m_{pq} = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} i^p \, j^q \, f(i, j) \, , \tag{8.42}$$

where x, y, i, j are the region point co-ordinates (pixel co-ordinates in digitized images). Translation invariance can be achieved if we use the central moments

$$\mu_{pq} = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} (i - x_c)^p \, (j - y_c)^q \, f(i, j) \, , \tag{8.43}$$

where x_c, y_c are the co-ordinates of the region's center of gravity (centroid), which can be obtained using:

$$x_c = \frac{m_{10}}{m_{00}} \, , \qquad y_c = \frac{m_{01}}{m_{00}} \, . \tag{8.44}$$

In the binary case, m_{00} represents the region area (see equations 8.41 and 8.42). Scale-invariant features can be found from scaled central moments η_{pq} (scale change $x' = \alpha x, y' = \alpha y$)

$$\eta_{pq} = \frac{\mu_{pq}}{(\mu_{00})^{(p+q)/2+1}} \tag{8.45}$$

and normalized un-scaled central moments ϑ_{pq}

$$\vartheta_{pq} = \frac{\mu_{pq}}{(\mu_{00})^{\gamma}} \, . \tag{8.46}$$

Rotation invariance can be achieved if the co-ordinate system is chosen such that $\mu_{11} = 0$ [Cash and Hatamian, 1987]. Many aspects of moment properties, normalization, descriptive power, sensitivity to noise, and computational cost are discussed in [Savini, 1988]. A less general form of invariance was given in [Hu, 1962] and is discussed in [Jain, 1989; Pratt, 1991], in which seven rotation-, translation-, and scale-invariant moment characteristics are used.

$$\varphi_1 = \vartheta_{20} + \vartheta_{02} \,. \tag{8.47}$$

$$\varphi_2 = (\vartheta_{20} - \vartheta_{02})^2 + 4\,\vartheta_{11}^2 \,. \tag{8.48}$$

$$\varphi_3 = (\vartheta_{30} - 3\,\vartheta_{12})^2 + (3\,\vartheta_{21} - \vartheta_{03})^2 \,, \tag{8.49}$$

$$\varphi_4 = (\vartheta_{30} + \vartheta_{12})^2 + (\vartheta_{21} + \vartheta_{03})^2 \,. \tag{8.50}$$

$$\begin{aligned}\varphi_5 = {} & (\vartheta_{30} - 3\,\vartheta_{12})(\vartheta_{30} + \vartheta_{12})\big((\vartheta_{30} + \vartheta_{12})^2 - 3(\vartheta_{21} + \vartheta_{03})^2\big) \\ & + (3\,\vartheta_{21} - \vartheta_{03})(\vartheta_{21} + \vartheta_{03})\big(3(\vartheta_{30} + \vartheta_{12})^2 - (\vartheta_{21} + \vartheta_{03})^2\big) \,,\end{aligned} \tag{8.51}$$

$$\varphi_6 = (\vartheta_{20} - \vartheta_{02})\big((\vartheta_{30} + \vartheta_{12})^2 - (\vartheta_{21} + \vartheta_{03})^2\big) + 4\,\vartheta_{11}(\vartheta_{30} + \vartheta_{12})(\vartheta_{21} + \vartheta_{03}) \,, \tag{8.52}$$

$$\begin{aligned}\varphi_7 = {} & (3\,\vartheta_{21} - \vartheta_{03})(\vartheta_{30} + \vartheta_{12})\big((\vartheta_{30} + \vartheta_{12})^2 - 3(\vartheta_{21} + \vartheta_{03})^2\big) \\ & - (\vartheta_{30} - 3\vartheta_{12})(\vartheta_{21} + \vartheta_{03})\big(3(\vartheta_{30} + \vartheta_{12})^2 - (\vartheta_{21} + \vartheta_{03})^2\big) \,,\end{aligned} \tag{8.53}$$

where the ϑ_{pq} values can be computed from equation (8.46).

While these seven characteristics were shown to be useful, they are invariant only to translation, rotation, and scaling. Improved algorithms for fast computation of translation-, rotation-, and scale-invariant moments were given in [Li and Shen, 1991; Jiang and Bunke, 1991]. However, these do not yield descriptors that are invariant under general affine transforms. Details of a process for the derivation of invariants and examples of invariant moment object descriptions can be found in [Flusser and Suk, 1993], where a complete set of four affine moment invariants derived from second- and third-order moments is presented.

$$I_1 = \frac{\mu_{20}\,\mu_{02} - \mu_{11}^2}{\mu_{00}^4} \,, \tag{8.54}$$

$$I_2 = \frac{\mu_{30}^2\,\mu_{03}^2 - 6\,\mu_{30}\,\mu_{21}\,\mu_{12}\,\mu_{03} + 4\,\mu_{30}\,\mu_{12}^3 + 4\,\mu_{21}^3\,\mu_{03} - 3\,\mu_{21}^2\,\mu_{12}^2}{\mu_{00}^{10}} \,, \tag{8.55}$$

$$I_3 = \frac{\mu_{20}(\mu_{21}\,\mu_{03} - \mu_{12}^2) - \mu_{11}(\mu_{30}\,\mu_{03} - \mu_{21}\,\mu_{12}) + \mu_{02}(\mu_{30}\,\mu_{12} - \mu_{21}^2)}{\mu_{00}^7} \,, \tag{8.56}$$

$$\begin{aligned}I_4 = \Big(& \mu_{20}^3\,\mu_{03}^2 - 6\,\mu_{20}^2\,\mu_{11}\,\mu_{12}\,\mu_{03} - 6\,\mu_{20}^2\,\mu_{02}\,\mu_{21}\,\mu_{03} + 9\,\mu_{20}^2\,\mu_{02}\,\mu_{12}^2 \\ & + 12\,\mu_{20}\,\mu_{11}^2\,\mu_{21}\,\mu_{03} + 6\,\mu_{20}\,\mu_{11}\,\mu_{02}\,\mu_{30}\,\mu_{03} - 18\,\mu_{20}\,\mu_{11}\,\mu_{02}\,\mu_{21}\,\mu_{12} \\ & - 8\,\mu_{11}^3\,\mu_{30}\,\mu_{03} - 6\,\mu_{20}\,\mu_{02}^2\,\mu_{30}\,\mu_{12} + 9\,\mu_{20}\,\mu_{02}^2\,\mu_{21}^2 \\ & + 12\,\mu_{11}^2\,\mu_{02}\,\mu_{30}\,\mu_{12} - 6\,\mu_{11}\,\mu_{02}^2\,\mu_{30}\,\mu_{21} + \mu_{02}^3\,\mu_{30}^2\Big) \Big/ \mu_{00}^{11} \,.\end{aligned} \tag{8.57}$$

All moment characteristics are dependent on the linear gray-level transformations of regions; to describe region shape properties, we work with binary image data ($f(i,j) = 1$ in region pixels) and dependence on the linear gray-level transform disappears.

Moment characteristics can be used in shape description even if the region is represented by its boundary. A closed boundary is characterized by an ordered sequence $z(i)$

that represents the Euclidean distance between the centroid and all N boundary pixels of the digitized shape. No extra processing is required for shapes having spiral or concave contours. Translation-, rotation-, and scale-invariant one-dimensional normalized contour sequence moments $\overline{m}_r, \overline{\mu}_r$ are defined in [Gupta and Srinath, 1987]. The r^{th} contour sequence moment m_r and the r^{th} central moment μ_r can be estimated as

$$m_r = \frac{1}{N} \sum_{i=1}^{N} \big(z(i)\big)^r . \tag{8.58}$$

$$\mu_r = \frac{1}{N} \sum_{i=1}^{N} \big(z(i) - m_1\big)^r . \tag{8.59}$$

The r^{th} normalized contour sequence moment $\overline{m}_r$ and normalized central contour sequence moment $\overline{\mu}_r$ are defined as

$$\overline{m}_r = \frac{m_r}{\mu_2^{r/2}} = \frac{\frac{1}{N}\sum_{i=1}^{N} \big(z(i)\big)^r}{\left(\frac{1}{N}\sum_{i=1}^{N} \big(z(i) - m_1\big)^2\right)^{r/2}} . \tag{8.60}$$

$$\overline{\mu}_r = \frac{\mu_r}{(\mu_2)^{r/2}} = \frac{\frac{1}{N}\sum_{i=1}^{N} \big(z(i) - m_1\big)^r}{\left(\frac{1}{N}\sum_{i=1}^{N} \big(z(i) - m_1\big)^2\right)^{r/2}} . \tag{8.61}$$

While the set of invariant moments $\overline{m}_r, \overline{\mu}_r$ can be used directly for shape representation, less noise-sensitive results can be obtained from [Gupta and Srinath, 1987]

$$F_1 = \frac{(\mu_2)^{1/2}}{m_1} = \frac{\left(\frac{1}{N}\sum_{i=1}^{N} \big(z(i) - m_1\big)^2\right)^{1/2}}{\frac{1}{N}\sum_{i=1}^{N} z(i)} . \tag{8.62}$$

$$F_2 = \frac{\mu_3}{(\mu_2)^{3/2}} = \frac{\frac{1}{N}\sum_{i=1}^{N} \big(z(i) - m_1\big)^3}{\left(\frac{1}{N}\sum_{i=1}^{N} \big(z(i) - m_1\big)^2\right)^{3/2}} . \tag{8.63}$$

$$F_3 = \frac{\mu_4}{(\mu_2)^2} = \frac{\frac{1}{N}\sum_{i=1}^{N} \big(z(i) - m_1\big)^4}{\left(\frac{1}{N}\sum_{i=1}^{N} \big(z(i) - m_1\big)^2\right)^2} . \tag{8.64}$$

$$F_4 = \overline{\mu}_5 . \tag{8.65}$$

Lower probabilities of error classification were obtained using contour sequence moments than area-based moments (equations 8.47–8.53) in a shape recognition test; also, contour sequence moments are less computationally demanding.

8.3.3 Convex hull

A region R is convex if and only if for any two points $\mathbf{x}_1, \mathbf{x}_2 \in R$, the whole line segment $\mathbf{x}_1\mathbf{x}_2$ defined by its end points $\mathbf{x}_1, \mathbf{x}_2$ is inside the region R. The convex hull of a region is the smallest convex region H which satisfies the condition $R \subseteq H$—see Figure 8.27. The convex hull has some special properties in digital data which do not exist in the continuous case. For instance, concave parts can appear and disappear in digital data

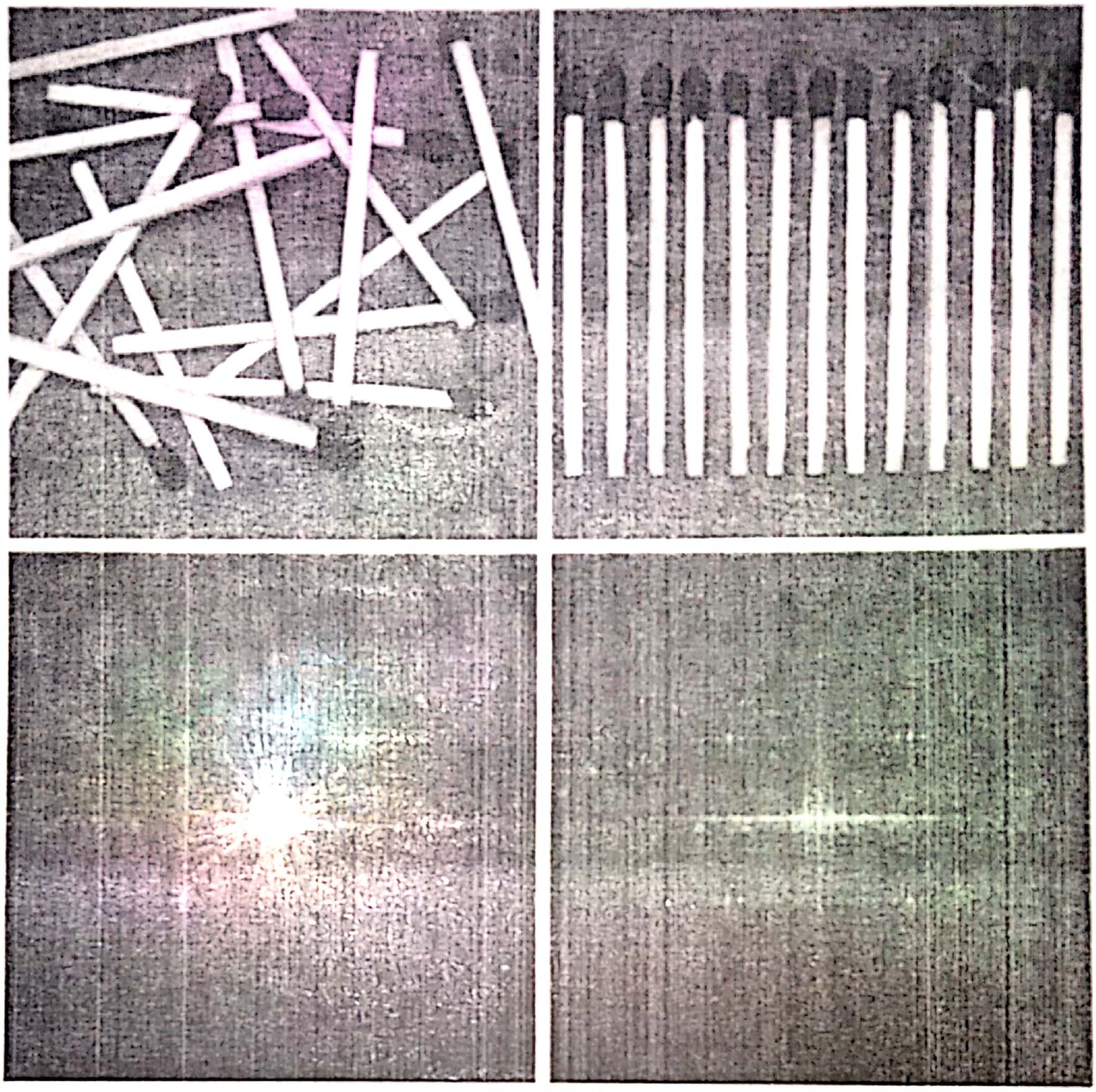


FIGURE 12.26 (a) and (b) Images of unordered and ordered objects. (c) and (d) Corresponding spectra.

12.4.3 Moment Invariants

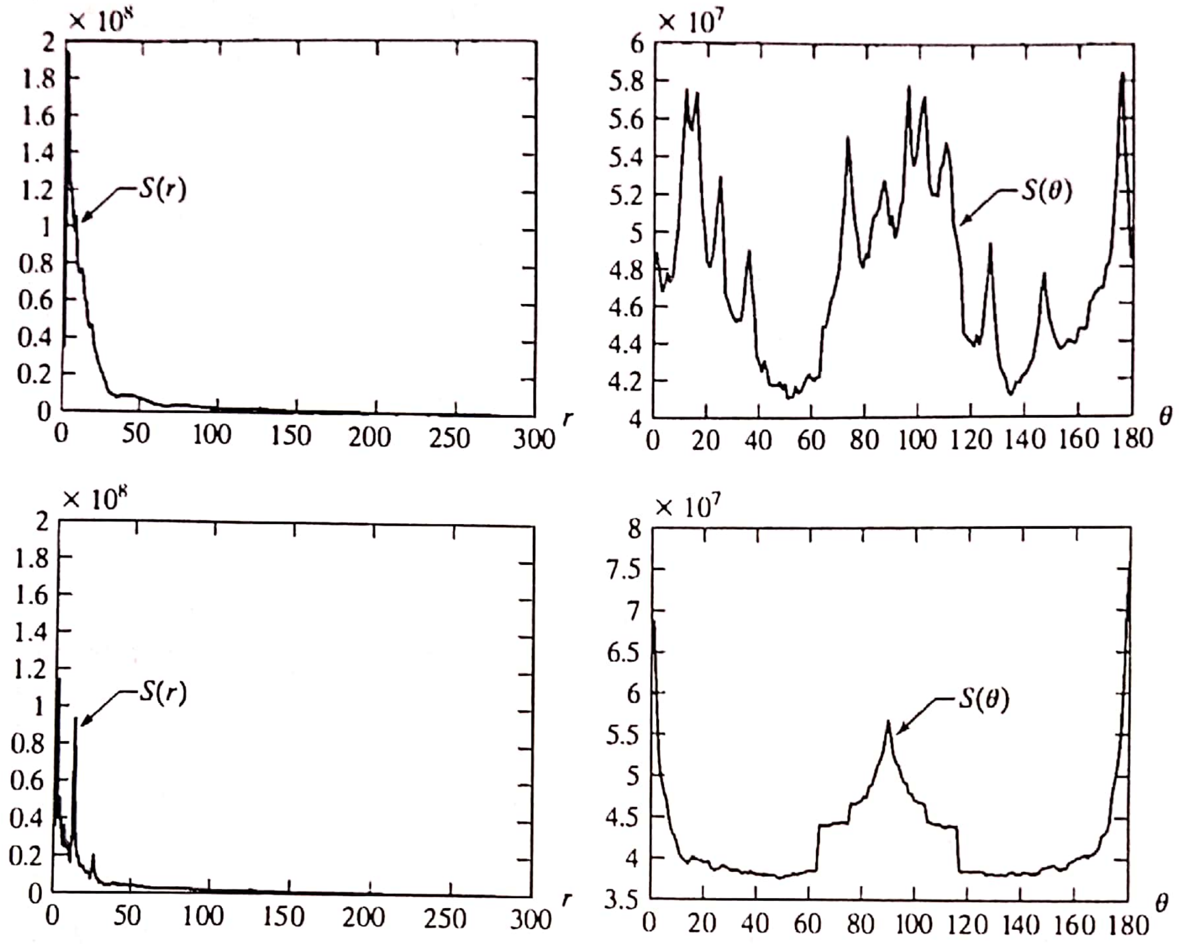
The 2-D *moment* of order $(p + q)$ of a digital image $f(x, y)$ of size $M \times N$ is defined as

$$m_{pq} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} x^p y^q f(x, y)$$

where $p = 0, 1, 2, \ldots$ and $q = 0, 1, 2, \ldots$ are integers. The corresponding *central moment* of order $(p + q)$ is defined as

$$\mu_{pq} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (x - \bar{x})^p (y - \bar{y})^q f(x, y)$$

for $p = 0, 1, 2, \ldots$ and $q = 0, 1, 2, \ldots$, where



a b
c d

FIGURE 12.27 (a) and (b) Plots of $S(r)$ and $S(\theta)$ for the random image in Fig. 12.26(a). (c) and (d) Plots of $S(r)$ and $S(\theta)$ for the ordered image.

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

The *normalized central moment* of order $(p + q)$ is defined as

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}}$$

where

$$\gamma = \frac{p + q}{2} + 1$$

for $p + q = 2, 3, \ldots$.

A set of seven 2-D *moment invariants* that are insensitive to translation, scale change, mirroring (to within a minus sign), and rotation can be derived from these equations.[†] They are listed in Table 12.6.

[†] Derivation of these results involves concepts that are beyond the scope of this discussion. The book by Bell [1965] and a paper by Hu [1962] contain detailed discussions of these concepts. For generating moment invariants of order higher than seven, see Flusser [2000]. Moment invariants can be generalized to *n* dimensions (see Mamistvalov [1998]).

TABLE 12.6 A set of seven moment invariants.

Moment order	Expression
1	$\phi_1 = \eta_{20} + \eta_{02}$
2	$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2$
3	$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2$
4	$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2$
5	$\phi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2$
	$- 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})$
	$[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$
6	$\phi_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$
	$+ 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})$
7	$\phi_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2$
	$- 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{30})(\eta_{21} + \eta_{03})$
	$[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$

Custom M-function invmoments implements these seven equations. The syntax is as follows (see Appendix C for the code):

invmoments

```
phi = invmoments(f)
```

where f is the input image and phi is a seven-element row vector containing the moment invariants just defined.

EXAMPLE 12.13: Moment invariants.

■ The image in Fig. 12.28(a) was obtained from an original of size 400 × 400 pixels using the following commands:

```
>> f = imread('Fig1228(a).tif');
>> fp = padarray(f, [84 84], 'both'); % Padded for display.
```

This image was created using zero padding to make all displayed images consistent in size with the image occupying the largest area (568 × 568) which, as explained below, is the image rotated by 45°. The padding is for display purposes only, and was not used in moment computations. A translated image was created using the following commands: