26

Fourier Shape Descriptors

Fourier descriptors are an interesting method for modeling 2D shapes that are described as closed contours. Unlike polylines or splines, which are explicit and local descriptions of the contour, Fourier descriptors are *global* shape representations, that is, each component stands for a particular characteristic of the entire shape. If one component is changed, the whole shape will change. The advantage is that it is possible to capture coarse shape properties with only a few numeric values, and the level of detail can be increased (or decreased) by adding (or removing) descriptor elements. In the following, we describe what is called "cartesian" (or "elliptical") Fourier descriptors, how they can be used to model the shape of closed 2D contours and how they can be adapted to compare shapes in a translation-, scale-, and rotation-invariant fashion.

26.1 Closed Curves in the Complex Plane

Any continuous curve C in the 2D plane can be expressed as a function $f\colon \mathbb{R} \to \mathbb{R}^2$, with

$$f(t) = \begin{pmatrix} x_t \\ y_t \end{pmatrix} = \begin{pmatrix} f_x(t) \\ f_y(t) \end{pmatrix}, \tag{26.1}$$

with the continuous parameter t being varied over the range $[0, t_{\max}]$. If the curve is closed, then $f(0) = f(t_{\max})$ and $f(t) = f(t + t_{\max})$. Note that $f_x(t)$, $f_y(t)$ are independent, real-valued functions, and t is the *path length* along the curve.

26.1.1 Discrete 2D Curves

Sampling a closed curve C at M regularly spaced positions $t_0, t_1, \ldots, t_{M-1}$, with $t_i - t_{i-1} = \Delta_t = \text{Length}(C)/M$, results in a sequence (vector) of discrete 2D coordinates $V = (\boldsymbol{v}_0, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_{M-1})$, with

$$\boldsymbol{v}_k = (x_k, y_k) = f(t_k). \tag{26.2}$$

W. Burger, M.J. Burge, *Digital Image Processing*, Texts in Computer Science,
DOI 10.1007/978-1-4471-6684-9_26

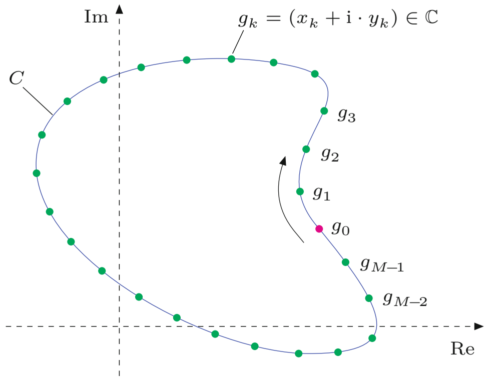


Fig. 26.1 A closed, continuous 2D curve C, represented as a sequence of M uniformly placed samples $\boldsymbol{g} = (g_0, g_1, \ldots, g_{M-1})$ in the complex plane.

Since the curve C is closed, the vector V represents a discrete function that is infinite and periodic, that is,

$$\boldsymbol{v}_k = \boldsymbol{v}_{k+pM}, \tag{26.3}$$

for $0 \leq k < M$ and any $p \in \mathbb{Z}$.

Contour points in the complex plane

Any 2D contour sample $\boldsymbol{v}_k = (x_k, y_k)$ can be interpreted as a point g_k in the complex plane,

$$g_k = x_k + \mathrm{i} \cdot y_k, \tag{26.4}$$

with x_k and y_k taken as the real and imaginary components, respectively.[1] The result is a sequence (vector) of complex values

$$\boldsymbol{g} = (g_0, g_1, \ldots, g_{M-1}), \tag{26.5}$$

representing the discrete 2D contour (see Fig. 26.1).

Regular position sampling

The assumption of input data being obtained by regular sampling is quite fundamental in traditional discrete Fourier analysis. In practice, contours of objects are typically not available as regularly sampled point sequences. For example, if an object has been segmented as a binary region, the coordinates of its boundary pixels could be used as the original contour sequence. However, the number of boundary pixels is usually too large to be used directly and their positions are not strictly uniformly spaced (at least under 8-connectivity). To produce a useful contour sequence from a region boundary, one could choose an arbitrary contour point as the start position $\boldsymbol{x}_0$ and then sample the x/y positions along the contour at regular (equidistant) steps, treating the centers of the boundary pixels as the vertices of a closed polygon. Algorithm 26.1 shows how to calculate a predefined number of contour points on an arbitrary polygon, such that the path

[1] Instead of $g \leftarrow x + \mathrm{i} \cdot y$, we sometimes use the short notation $g \leftarrow (x, y)$ or $g \leftarrow \boldsymbol{v}$ for assigning the components of a 2D vector $\boldsymbol{v} = (x, y) \in \mathbb{R}^2$ to a complex variable $g \in \mathbb{C}$.

```
SamplePolygonUniformly($V, M$)
    Input: $V = (\boldsymbol{v}_0, \ldots, \boldsymbol{v}_{N-1})$, a sequence of $N$ points representing
    the vertices of a 2D polygon; $M$, number of desired sample points.
    Returns a sequence $\boldsymbol{g} = (g_0, \ldots, g_{M-1})$ of complex values rep-
    resenting points sampled uniformly along the path of the input
    polygon $V$.

    $N \leftarrow |V|$
    $\Delta \leftarrow \frac{1}{M} \cdot \mathsf{PathLength}(V)$                ▷ const. segment length $\Delta$
    Create map $\boldsymbol{g}\colon [0, M-1] \to \mathbb{C}$            ▷ complex point sequence $\boldsymbol{g}$
    $\boldsymbol{g}(0) \leftarrow \mathsf{Complex}(V(0))$
    $i \leftarrow 0$                          ▷ index of polygon segment $\langle \boldsymbol{v}_i, \boldsymbol{v}_{i+1} \rangle$
    $k \leftarrow 1$                          ▷ index of next point to be added to $\boldsymbol{g}$
    $\alpha \leftarrow 0$                          ▷ path position of polygon vertex $\boldsymbol{v}_i$
    $\beta \leftarrow \Delta$                    ▷ path position of next point to be added to $\boldsymbol{g}$

    while $(i < N) \wedge (k < M)$ do
        $\boldsymbol{v}_{\mathrm{A}} \leftarrow V(i)$
        $\boldsymbol{v}_{\mathrm{B}} \leftarrow V((i+1) \bmod N)$
        $\delta \leftarrow \|\boldsymbol{v}_{\mathrm{B}} - \boldsymbol{v}_{\mathrm{A}}\|$                ▷ length of segment $\langle \boldsymbol{v}_{\mathrm{A}}, \boldsymbol{v}_{\mathrm{B}} \rangle$
        while $(\beta \leq \alpha + \delta) \wedge (k < M)$ do
            $\boldsymbol{x} \leftarrow \boldsymbol{v}_{\mathrm{A}} + \frac{\beta - \alpha}{\delta} \cdot (\boldsymbol{v}_{\mathrm{B}} - \boldsymbol{v}_{\mathrm{A}})$      ▷ linear path interpolation
            $\boldsymbol{g}(k) \leftarrow \mathsf{Complex}(\boldsymbol{x})$
            $k \leftarrow k + 1$
            $\beta \leftarrow \beta + \Delta$
        $\alpha \leftarrow \alpha + \delta$
        $i \leftarrow i + 1$
    return $\boldsymbol{g}$.

PathLength($V$)      ▷ returns the path length of the closed polygon $V$
    $N \leftarrow |V|$
    $L \leftarrow 0$
    for $i \leftarrow 0, \ldots, N-1$ do
        $\boldsymbol{v}_{\mathrm{A}} \leftarrow V(i)$
        $\boldsymbol{v}_{\mathrm{B}} \leftarrow V((i+1) \bmod N)$
        $L \leftarrow L + \|\boldsymbol{v}_{\mathrm{B}} - \boldsymbol{v}_{\mathrm{A}}\|$
    return $L$.
```

Alg. 26.1
Regular sampling of a polygon path. Given a sequence V of 2D points representing the vertices of a closed polygon, SamplePolygonUniformly(V, M) returns a sequence of M complex values $\boldsymbol{g}$ on the polygon V, such that $\boldsymbol{g}(0) \equiv V(0)$ and all remaining points $\boldsymbol{g}(k)$ are uniformly positioned along the polygon path. See Alg. 26.9 for an alternate solution.

length between the sample points is uniform. This algorithm is used in all examples involving contours obtained from binary regions.

Note that if the shape is given as an arbitrary polygon, the corresponding Fourier descriptor can also be calculated directly (and exactly) from the vertices of the polygon, without sub-sampling the polygon contour path at all. This "trigonometric" variant of the Fourier descriptor calculation is described in Sec. 26.3.7.

26.2 Discrete Fourier Transform (DFT)

Fourier descriptors are obtained by applying the 1D Discrete Fourier Transform (DFT)[2] to the complex-valued vector $\boldsymbol{g}$ of 2D contour points (Eqn. (26.5)). The DFT is a transformation of a finite, complex-valued *signal* vector $\boldsymbol{g} = (g_0, g_1, \ldots, g_{M-1})$ to a complex-valued *spec-*

[2] See Chapter 18, Sec. 18.3.

trum $\boldsymbol{G} = (G_0, G_1, \ldots, G_{M-1})$.[3] Both the signal and the spectrum are of the same length (M) and periodic. In the following, we typically use k to denote the index in the time or space domain,[4] and m for a frequency index in the spectral domain.

26.2.1 Forward Fourier Transform

The discrete Fourier spectrum $\boldsymbol{G} = (G_0, G_1, \ldots, G_{M-1})$ is calculated from the discrete, complex-valued signal $\boldsymbol{g} = (g_0, g_1, \ldots, g_{M-1})$ using the forward DFT, defined as[5]

$$G_m = \frac{1}{M} \cdot \sum_{k=0}^{M-1} g_k \cdot e^{-\mathrm{i}\cdot 2\pi m \cdot \frac{k}{M}} = \frac{1}{M} \cdot \sum_{k=0}^{M-1} g_k \cdot e^{-\mathrm{i}\cdot \omega_m \cdot \frac{k}{M}} \tag{26.6}$$

$$= \frac{1}{M} \cdot \sum_{k=0}^{M-1} \underbrace{[x_k + \mathrm{i}\cdot y_k]}_{g_k} \cdot \bigl[\cos\bigl(\underbrace{2\pi m}_{\omega_m} \tfrac{k}{M}\bigr) - \mathrm{i}\cdot\sin\bigl(\underbrace{2\pi m}_{\omega_m} \tfrac{k}{M}\bigr)\bigr] \tag{26.7}$$

$$= \frac{1}{M} \cdot \sum_{k=0}^{M-1} [x_k + \mathrm{i}\cdot y_k] \cdot \bigl[\cos\bigl(\omega_m \tfrac{k}{M}\bigr) - \mathrm{i}\cdot\sin\bigl(\omega_m \tfrac{k}{M}\bigr)\bigr], \tag{26.8}$$

for $0 \leq m < M$.[6] Note that $\omega_m = 2\pi m$ denotes the *angular frequency* for the frequency index m. By applying the usual rules of complex multiplication, we obtain the *real* (Re) and *imaginary* (Im) parts of the spectral coefficients $G_m = (A_m + \mathrm{i} \cdot B_m)$ explicitly as

$$A_m = \mathrm{Re}(G_m) = \frac{1}{M} \sum_{k=0}^{M-1} \bigl[x_k \cdot \cos\bigl(\omega_m \tfrac{k}{M}\bigr) + y_k \cdot \sin\bigl(\omega_m \tfrac{k}{M}\bigr)\bigr], \tag{26.9}$$

$$B_m = \mathrm{Im}(G_m) = \frac{1}{M} \sum_{k=0}^{M-1} \bigl[y_k \cdot \cos\bigl(\omega_m \tfrac{k}{M}\bigr) - x_k \cdot \sin\bigl(\omega_m \tfrac{k}{M}\bigr)\bigr]. \tag{26.10}$$

The DFT is defined for any signal length $M \geq 1$. If the signal length M is a power of two (that is, $M = 2^n$ for some $n \in \mathbb{N}$), the Fast Fourier Transform (FFT)[7] can be used in place of the DFT for improved performance.

26.2.2 Inverse Fourier Transform (Reconstruction)

The inverse DFT reconstructs the original signal $\boldsymbol{g}$ from a given spectrum $\boldsymbol{G}$. The formulation is almost symmetrical (except for the scale

[3] In most traditional applications of the DFT (e.g. in acoustic processing), the signals are real-valued, that is, the imaginary components of the samples are zero. The Fourier spectrum is generally complex-valued, but it is symmetric for real-valued signals.

[4] We use k instead of the usual i as the running index to avoid confusion with the imaginary constant "i" (despite the deliberate use of different glyphs).

[5] This definition deviates slightly from the one used in Chapter 18, Sec. 18.3 but is otherwise equivalent.

[6] Recall that $z = x + \mathrm{i}y = |z| \cdot (\cos\psi + \mathrm{i} \cdot \sin\psi) = |z| \cdot e^{\mathrm{i}\psi}$, with $\psi = \tan^{-1}(y/x)$.

[7] See Chapter 18, Sec. 18.4.2.

```
FourierDescriptorUniform($\boldsymbol{g}$)
    Input: $\boldsymbol{g} = (g_0, \ldots, g_{M-1})$, a sequence of $M$ complex values,
    representing regularly sampled 2D points along a contour path.
    Returns a Fourier descriptor $\boldsymbol{G}$ of length $M$.
    $M \leftarrow |\boldsymbol{g}|$
    Create map $\boldsymbol{G}\colon [0, M-1] \to \mathbb{C}$
    for $m \leftarrow 0, \ldots, M-1$ do
        $A \leftarrow 0, \quad B \leftarrow 0$              ▷ real/imag. part of coefficient $G_m$
        for $k \leftarrow 0, \ldots, M-1$ do
            $g \leftarrow \boldsymbol{g}(k)$
            $x \leftarrow \mathrm{Re}(g), \quad y \leftarrow \mathrm{Im}(g)$
            $\phi \leftarrow 2 \cdot \pi \cdot m \cdot \frac{k}{M}$
            $A \leftarrow A + x \cdot \cos(\phi) + y \cdot \sin(\phi)$              ▷ Eq. 26.10
            $B \leftarrow B - x \cdot \sin(\phi) + y \cdot \cos(\phi)$
        $\boldsymbol{G}(m) \leftarrow \frac{1}{M} \cdot (A + \mathrm{i} \cdot B)$
    return $\boldsymbol{G}$.
```

Alg. 26.2
Calculating the Fourier descriptor for a sequence of uniformly sampled contour points. The complex-valued contour points in C represent 2D positions sampled uniformly along the contour path. Applying the DFT to $\boldsymbol{g}$ yields the raw Fourier descriptor $\boldsymbol{G}$.

factor and the different signs in the exponent) to the forward transformation in Eqns. (26.6)–(26.8); its full expansion is

$$g_k = \sum_{m=0}^{M-1} G_m \cdot e^{\mathrm{i} \cdot 2\pi m \cdot \frac{k}{M}} = \sum_{m=0}^{M-1} G_m \cdot e^{\mathrm{i} \cdot \omega_m \cdot \frac{k}{M}} \tag{26.11}$$

$$= \sum_{m=0}^{M-1} \underbrace{\bigl[\mathrm{Re}(G_m) + \mathrm{i} \cdot \mathrm{Im}(G_m)\bigr]}_{G_m} \cdot \bigl[\cos\bigl(\underbrace{2\pi m}_{\omega_m} \tfrac{k}{M}\bigr) + \mathrm{i} \cdot \sin\bigl(\underbrace{2\pi m}_{\omega_m} \tfrac{k}{M}\bigr)\bigr] \tag{26.12}$$

$$= \sum_{m=0}^{M-1} \bigl[A_m + \mathrm{i} \cdot B_m\bigr] \cdot \bigl[\cos\bigl(\omega_m \tfrac{k}{M}\bigr) + \mathrm{i} \cdot \sin\bigl(\omega_m \tfrac{k}{M}\bigr)\bigr]. \tag{26.13}$$

Again we can expand Eqn. (26.13) to obtain the real and imaginary parts of the reconstructed signal, that is, the x/y-components of the corresponding curve points $g_k = (x_k, y_k)$ as

$$x_k = \mathrm{Re}(g_k) = \sum_{m=0}^{M-1} \bigl[\mathrm{Re}(G_m) \cdot \cos\bigl(2\pi m \tfrac{k}{M}\bigr) - \mathrm{Im}(G_m) \cdot \sin\bigl(2\pi m \tfrac{k}{M}\bigr)\bigr], \tag{26.14}$$

$$y_k = \mathrm{Im}(g_k) = \sum_{m=0}^{M-1} \bigl[\mathrm{Im}(G_m) \cdot \cos\bigl(2\pi m \tfrac{k}{M}\bigr) + \mathrm{Re}(G_m) \cdot \sin\bigl(2\pi m \tfrac{k}{M}\bigr)\bigr], \tag{26.15}$$

for $0 \leq k < M$. If *all* coefficients of the spectrum are used, this reconstruction is *exact*, that is, the resulting discrete points g_k are identical to the original contour points.[8]

With the aforementioned formulation we can not only reconstruct the discrete contour points g_k from the DFT spectrum, but also a smooth, interpolating curve as the sum of continuous sine and cosine components. To calculate *arbitrary* points on this curve, we replace the discrete quantity $\frac{k}{M}$ in Eqn. (26.15) by the continuous parameter t in the range $[0, 1)$. We must be careful about the frequencies, though. To achieve the desired *smooth* interpolation, the set of *lowest* possible

[8] Apart from inaccuracies caused by finite floating-point precision.

frequencies ω_m must be used,[9] that is,

$$x(t) = \sum_{m=0}^{M-1} \bigl[\mathrm{Re}(G_m) \cdot \cos(\omega_m \cdot t) - \mathrm{Im}(G_m) \cdot \sin(\omega_m \cdot t)\bigr], \tag{26.16}$$

$$y(t) = \sum_{m=0}^{M-1} \bigl[\mathrm{Im}(G_m) \cdot \cos(\omega_m \cdot t) + \mathrm{Re}(G_m) \cdot \sin(\omega_m \cdot t)\bigr], \tag{26.17}$$

$$\text{with} \qquad \omega_m = \begin{cases} 2\pi m & \text{for } m \leq (M \div 2), \\ 2\pi(m-M) & \text{for } m > (M \div 2), \end{cases} \tag{26.18}$$

where ÷ denotes the quotient (i.e., integer division). Alternatively, we could write Eqn. (26.17) in the form

$$x(t) = \sum_{\substack{m= \\ -(M-1)\div 2}}^{M\div 2} [\mathrm{Re}(G_{m \bmod M}) \cdot \cos(2\pi m t) - \mathrm{Im}(G_{m \bmod M}) \cdot \sin(2\pi m t)], \tag{26.19}$$

$$y(t) = \sum_{\substack{m= \\ -(M-1)\div 2}}^{M\div 2} [\mathrm{Im}(G_{m \bmod M}) \cdot \cos(2\pi m t) + \mathrm{Re}(G_{m \bmod M}) \cdot \sin(2\pi m t)]. \tag{26.20}$$

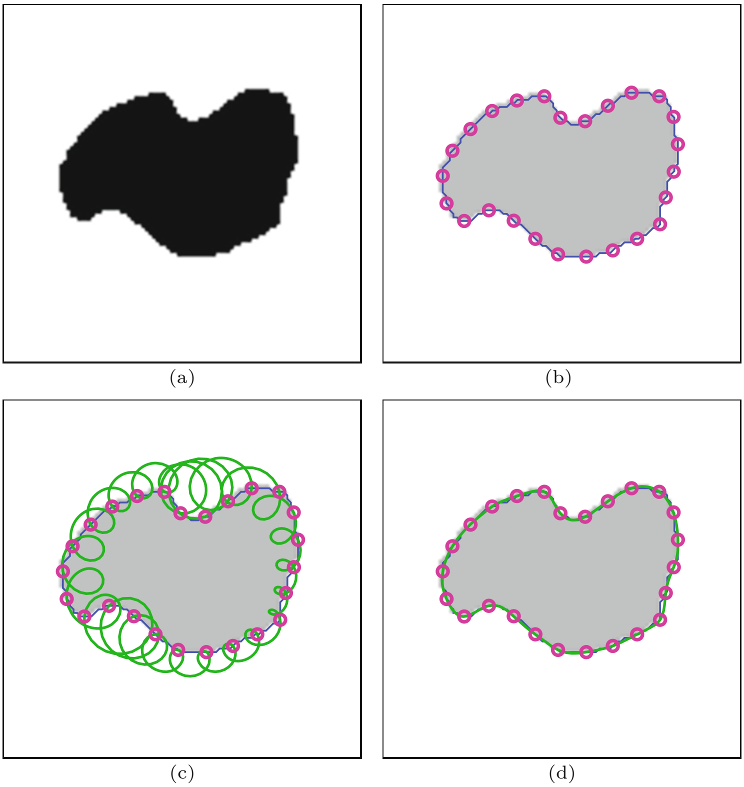
This formulation is used for the purpose of shape reconstruction from Fourier descriptors in Alg. 26.4.

Figure (26.2) shows the reconstruction of the discrete contour points as well as the calculation of a continuous outline from the DFT spectrum obtained from a sequence of discrete contour positions. The original sample points were taken at $M = 25$ uniformly spaced positions along the region's contour. The discrete points in Fig. 26.2(b) are exactly reconstructed from the complete DFT spectrum, as specified in Eqn. (26.15). The interpolated (green) outline in Fig. 26.2(c) was calculated with Eqn. (26.15) for continuous positions, based on the frequencies $m = 0, \ldots, M-1$. The oscillations of the resulting curve are explained by the high-frequency components. Note that the curve still passes exactly through each of the original sample points, in fact, these can be perfectly reconstructed from *any* contiguous range of M coefficients and the corresponding harmonic frequencies. The smooth interpolation in Fig. 26.2(d), based on the symmetric low-frequency coefficients $m = -(M-1) \div 2, \ldots, M \div 2$ (see Eqn. (26.20)) shows no such oscillations, since no high-frequency components are included.

26.2.3 Periodicity of the DFT Spectrum

When we apply the DFT, we implicitly assume that both the signal vector $\boldsymbol{g} = (g_0, g_1, \ldots, g_{M-1})$ and the spectral vector $\boldsymbol{G} = (G_0, G_1, \ldots, G_{M-1})$ represent discrete, periodic functions of infinite extent

[9] Due to the periodicity of the discrete spectrum, any summation over M successive frequencies ω_m can be used to reconstruct the original discrete x/y samples. However, a smooth interpolation between the discrete x/y samples can only be obtained from the set of *lowest* frequencies in the range $[-\frac{M}{2}, +\frac{M}{2}]$ centered around the zero frequency, as in Eqns. (26.17) and (26.20).



(a) (b) (c) (d)

Fig. 26.2
Contour reconstruction by inverse DFT. Original image (a), $M = 25$ uniformly spaced sample points on the region's contour (b). Continuous contour (green line) reconstructed by using frequencies ω_m with $m = 0, \ldots, 24$ (c). Note that despite the oscillations introduced by the high frequencies, the continuous contour passes exactly through the original sample points. Smooth interpolation reconstructed with Eqn. (26.17) from the lowest-frequency coefficients in the symmetric range $m = -12, \ldots, +12$ (d).

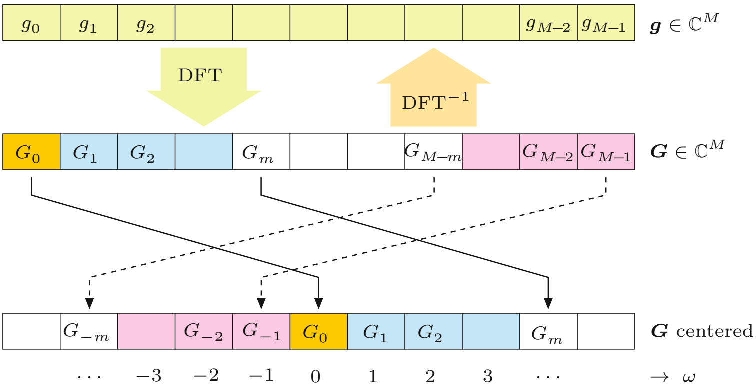


Fig. 26.3
Applying the DFT to a complex-valued vector $\boldsymbol{g}$ of length M yields the complex-valued spectrum $\boldsymbol{G}$ that is also of length M. The DFT spectrum is infinite and periodic with M, thus $G_{-m} = G_{M-m}$, as illustrated by the centered representation of the DFT spectrum (bottom). ω at the bottom denotes the harmonic number (multiple of the fundamental frequency) associated with each coefficient.

(see [39, Ch. 13] for details). Due to this periodicity, $\boldsymbol{G}(0) = \boldsymbol{G}(M)$, $\boldsymbol{G}(1) = \boldsymbol{G}(M+1)$, etc. In general,

$$\boldsymbol{G}(q \cdot M + m) = \boldsymbol{G}(m) \quad \text{and} \quad \boldsymbol{G}(m) = \boldsymbol{G}(m \bmod M), \tag{26.21}$$

for arbitrary integers $q, m \in \mathbb{Z}$. Also, since $(-m \bmod M) = (M-m) \bmod M$, we can state that

$$\boldsymbol{G}(-m) = \boldsymbol{G}(M-m), \tag{26.22}$$

for any $m \in \mathbb{Z}$, such that $\boldsymbol{G}(-1) = \boldsymbol{G}(M-1)$, $\boldsymbol{G}(-2) = \boldsymbol{G}(M-2)$, etc., as illustrated in Fig. 26.3.

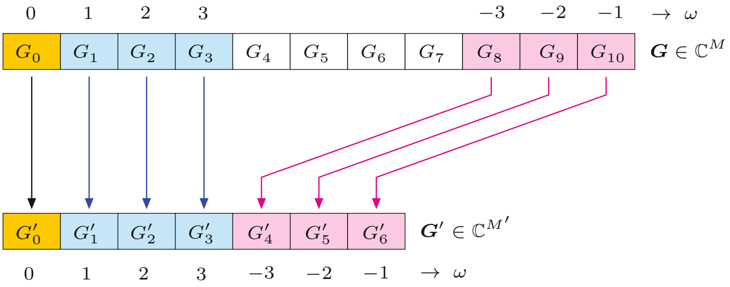


Fig. 26.4
Truncating a DFT spectrum from $M = 11$ to $M' = 7$ coefficients, as specified in Eqns. (26.23) and (26.24). Coefficients $G_4, \ldots, G_7$ are discarded ($M' \div 2 = 3$). Note that the associated harmonic number ω remains the same for each coefficient.

26.2.4 Truncating the DFT Spectrum

In the original formulation in Eqns. (26.6)–(26.8), the DFT is applied to a signal $\boldsymbol{g}$ of length M and yields a discrete Fourier spectrum $\boldsymbol{G}$ with M coefficients. Thus the signal and the spectrum have the same length. For shape representation, it is often useful to work with a truncated spectrum, that is, a reduced number of low-frequency Fourier coefficients.

By truncating a spectrum we mean the removal of coefficients above a certain harmonic number, which are (considering positive and negative frequencies) located around the center of the coefficient vector. Truncating a given spectrum $\boldsymbol{G}$ of length $|\boldsymbol{G}| = M$ to a shorter spectrum $\boldsymbol{G}'$ of length $M' \leq M$ is done as

$$\boldsymbol{G}'(m) \leftarrow \begin{cases} \boldsymbol{G}(m) & \text{for } 0 \leq m \leq M' \div 2, \\ \boldsymbol{G}(M{-}M'{+}m) & \text{for } M' \div 2 < m < M', \end{cases} \tag{26.23}$$

or simply

$$\boldsymbol{G}'(m \bmod M') \leftarrow \boldsymbol{G}(m \bmod M), \tag{26.24}$$

for $(M'{\div}2{-}M'{+}1) \leq m \leq (M'{\div}2)$. This works for M and M' being even or odd. The example in Fig. 26.4 illustrates how an original DFT spectrum $\boldsymbol{G}$ of length $M = 11$ is truncated to $\boldsymbol{G}'$ with only $M' = 7$ coefficients.

Of course it is also possible to calculate the truncated spectrum directly from the contour samples, without going through the full DFT spectrum. With M being the length of the signal vector $\boldsymbol{g}$ and $M' \leq M$ the desired length of the (truncated) spectrum $\boldsymbol{G}'$, Eqn. (26.6) modifies to

$$\boldsymbol{G}'(m \bmod M') = \frac{1}{M} \cdot \sum_{k=0}^{M-1} g_k \cdot e^{-\mathrm{i} 2\pi m \frac{k}{M}}, \tag{26.25}$$

for m in the same range as in Eqn. (26.24). This approach is more efficient than truncating the complete spectrum, since unneeded coefficients are never calculated. Algorithm 26.3, which is a modified version of Alg. 26.2, summarizes the steps we have described.

Since some of the coefficients are missing, it is not possible to reconstruct the original signal vector $\boldsymbol{g}$ from the truncated DFT spectrum $\boldsymbol{G}'$. However, the calculation of a partial reconstruction is possible, for example, using the formulation in Eqn. (26.20). In this

```
FourierDescriptorUniform($\boldsymbol{g}, M'$)
    Input: $\boldsymbol{g} = (g_0, \ldots, g_{M-1})$, a sequence of $M$ complex values,
    representing regularly sampled 2D points along a contour path.
    $M'$, the number of Fourier coefficients ($M' \leq M$).
    Returns a truncated Fourier descriptor $\boldsymbol{G}$ of length $M'$.
    $M \leftarrow |\boldsymbol{g}|$
    Create map $\boldsymbol{G}\colon [0, M'-1] \to \mathbb{C}$
    for $m \leftarrow (M' \div 2 - M' + 1), \ldots, (M' \div 2)$ do
        $A \leftarrow 0, \quad B \leftarrow 0$          ▷ real/imag. part of coefficient $G_m$
        for $k \leftarrow 0, \ldots, M-1$ do
            $g \leftarrow \boldsymbol{g}(k)$
            $x \leftarrow \mathrm{Re}(g), \quad y \leftarrow \mathrm{Im}(g)$
            $\phi \leftarrow 2 \cdot \pi \cdot m \cdot \frac{k}{M}$
            $A \leftarrow A + x \cdot \cos(\phi) + y \cdot \sin(\phi)$          ▷ Eq. 26.10
            $B \leftarrow B - x \cdot \sin(\phi) + y \cdot \cos(\phi)$
        $\boldsymbol{G}(m \bmod M') \leftarrow \frac{1}{M} \cdot (A + \mathrm{i} \cdot B)$
    return $\boldsymbol{G}$.
```

Alg. 26.3
Calculating a truncated Fourier descriptor for a sequence of uniformly sampled contour points (adapted from Alg. 26.2). The M complex-valued contour points in $\boldsymbol{g}$ represent 2D positions sampled uniformly along the contour path. The resulting Fourier descriptor $\boldsymbol{G}$ contains only M' coefficients for the M' lowest harmonic frequencies.

case, the discarded (high-frequency) coefficients are simply assumed to have zero values (see Sec. 26.3.6 for more details).

26.3 Geometric Interpretation of Fourier Coefficients

The contour reconstructed by the inverse transformation (Eqn. (26.15)) is the sum of M terms, one for each Fourier coefficient $G_m = (A_m, B_m)$. Each of these M terms represents a particular 2D shape in the spatial domain and the original contour can be obtained by point-wise addition of the individual shapes. So what are the spatial shapes that correspond to the individual Fourier coefficients?

26.3.1 Coefficient G_0 Corresponds to the Contour's Centroid

We first look only at the specific Fourier coefficient G_0 with frequency index $m = 0$. Substituting $m = 0$ and $\omega_0 = 0$ in Eqn. (26.10), we get

$$A_0 = \frac{1}{M} \sum_{k=0}^{M-1} \Bigl[x_k \cdot \cos(0) + y_k \cdot \sin(0) \Bigr] \tag{26.26}$$

$$= \frac{1}{M} \sum_{k=0}^{M-1} \Bigl[x_k \cdot 1 + y_k \cdot 0 \Bigr] \;=\; \frac{1}{M} \sum_{k=0}^{M-1} x_k \;=\; \bar{x}, \tag{26.27}$$

$$B_0 = \frac{1}{M} \sum_{k=0}^{M-1} \Bigl[y_k \cdot \cos(0) - x_k \cdot \sin(0) \Bigr] \tag{26.28}$$

$$= \frac{1}{M} \sum_{k=0}^{M-1} \Bigl[y_k \cdot 1 - x_k \cdot 0 \Bigr] \;=\; \frac{1}{M} \sum_{k=0}^{M-1} y_k \;=\; \bar{y}. \tag{26.29}$$

Thus $G_0 = (A_0, B_0) = (\bar{x}, \bar{y})$ is simply the average of the x/y-coordinates, that is, the *centroid* of the original contour points g_k (see

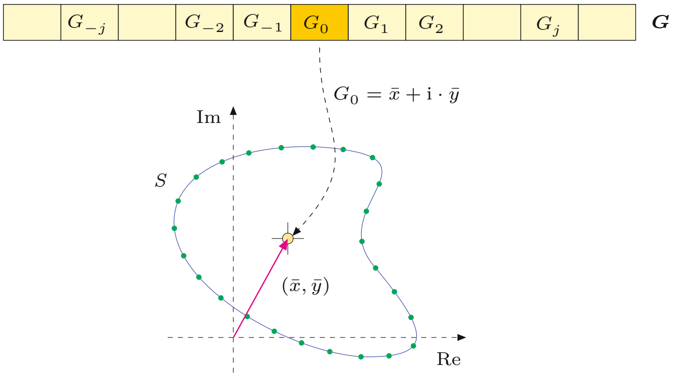


Fig. 26.5 DFT coefficient G_0 corresponds to the centroid of the contour points.

Fig. 26.5).[10] If we apply the *inverse* Fourier transform (Eqn. (26.15)) by ignoring (i.e., zeroing) all coefficients except G_0, we get the *partial reconstruction*[11] of the 2D contour coordinates $g_k^{(0)} = \big(x_k^{(0)}, y_k^{(0)}\big)$ as

$$x_k^{(0)} = \big[A_0 \cdot \cos\big(\omega_0 \tfrac{k}{M}\big) - B_0 \cdot \sin\big(\omega_0 \tfrac{k}{M}\big)\big] \tag{26.30}$$
$$= \bar{x} \cdot \cos(0) - \bar{y} \cdot \sin(0) = \bar{x} \cdot 1 - \bar{y} \cdot 0 \; = \; \bar{x}, \tag{26.31}$$
$$y_k^{(0)} = \big[B_0 \cdot \cos\big(\omega_0 \tfrac{k}{M}\big) + A_0 \cdot \sin\big(\omega_0 \tfrac{k}{M}\big)\big] \tag{26.32}$$
$$= \bar{y} \cdot \cos(0) + \bar{x} \cdot \sin(0) = \bar{y} \cdot 1 + \bar{x} \cdot 0 \; = \; \bar{y}. \tag{26.33}$$

Thus the contribution of the spectral value G_0 is the *centroid* of the reconstructed shape (see Fig. 26.5). If we perform a partial reconstruction of the contour using only the spectral coefficient G_0, then all contour points

$$g_0^{(0)} = g_1^{(0)} = \; \ldots \; = g_k^{(0)} = \; \ldots \; = g_{M-1}^{(0)} = (\bar{x}, \bar{y}) \tag{26.34}$$

would have the same (centroid) coordinate. This is because G_0 is the coefficient for the zero frequency and thus the sine and cosine terms in Eqns. (26.27) and (26.29) are constant. Alternatively, if we reconstruct the signal by *omitting* G_0 (i.e., $\boldsymbol{g}^{(1,\ldots,M-1)}$), the resulting contour is identical to the original shape, except that it is centered at the coordinate origin.

26.3.2 Coefficient G_1 Corresponds to a Circle

Next, we look at the geometric interpretation of $G_1 = (A_1, B_1)$, that is, the coefficient with frequency index $m = 1$, which corresponds to the angular frequency $\omega_1 = 2\pi$. Assuming that all coefficients G_m in the DFT spectrum are set to zero, except the single coefficient G_1,

[10] Note that the centroid of a boundary is generally not the same as the centroid of the enclosed region.

[11] We use the notation $\boldsymbol{g}^{(m)} = (g_0^{(m)}, g_1^{(m)}, \ldots, g_{M-1}^{(m)})$ for the *partial reconstruction* of the contour $\boldsymbol{g}$ from only a single Fourier coefficient G_m. For example, $\boldsymbol{g}^{(0)}$ is the reconstruction from the zero-frequency coefficient G_0 only. Analogously, we use $\boldsymbol{g}^{(a,b,c)}$ to denote a partial reconstruction based on selected Fourier coefficients G_a, G_b, G_c.

we get the partially reconstructed contour points $\boldsymbol{g}^{(1)}$ by Eqn. (26.11) as

$$
\begin{aligned}
g_k^{(1)} &= G_1 \cdot e^{\mathrm{i}\cdot 2\pi\cdot \frac{k}{M}} && (26.35)\\
&= [A_1 + \mathrm{i}\cdot B_1] \cdot [\cos(2\pi\tfrac{k}{M}) + \mathrm{i}\cdot \sin(2\pi\tfrac{k}{M})], && (26.36)
\end{aligned}
$$

for $0 \leq k < M$. Remember that the complex values of $e^{\mathrm{i}\varphi}$ describe a *unit circle* in the complex plane that performs one full (counter-clockwise) revolution, as the angle φ runs from $0, \ldots, 2\pi$. Analogously, $e^{\mathrm{i}2\pi t}$ also describes a complete unit circle as t goes from 0 to 1. Since the term $\frac{k}{M}$ (for $0 \leq k < M$) also varies from 0 to 1 in Eqn. (26.36), the M reconstructed contour points are placed on a circle at equal angular steps. Multiplying $e^{\mathrm{i}\cdot 2\pi t}$ by a complex factor z stretches the *radius* of the circle by $|z|$, and also changes the *phase* (starting angle) of the circle by an angle θ, that is,

$$
z \cdot e^{\mathrm{i}\cdot\varphi} = |z| \cdot e^{\mathrm{i}\cdot(\varphi+\theta)}, \qquad (26.37)
$$

with $\theta = \sphericalangle z = \arg(z) = \tan^{-1}(\mathrm{Im}(z)/\mathrm{Re}(z))$.

We now see that the points $g_k^{(1)} = G_1 \cdot e^{\mathrm{i}\,2\pi k/M}$, generated by Eqn. (26.36), are positioned uniformly on a circle with radius $r_1 = |G_1|$ and starting angle (phase)

$$
\theta_1 = \sphericalangle G_1 = \tan^{-1}\Bigl(\frac{\mathrm{Im}(G_1)}{\mathrm{Re}(G_1)}\Bigr) = \tan^{-1}\Bigl(\frac{B_1}{A_1}\Bigr). \qquad (26.38)
$$

This point sequence is traversed in counter-clockwise direction for $k = 0, \ldots, M-1$ at frequency $m = 1$, that is, the circle performs one full revolution while the contour is traversed once. The circle is centered at the coordinate origin $(0, 0)$, its radius is $|G_1|$, and its starting point (Eqn. (26.36) for $k = 0$) is

$$
g_0^{(1)} = G_1 \cdot e^{\mathrm{i}\cdot 2\pi m\cdot\frac{k}{M}} = G_1 \cdot e^{\mathrm{i}\cdot 2\pi 1\cdot\frac{0}{M}} = G_1 \cdot e^0 = G_1, \qquad (26.39)
$$

as illustrated in Fig. 26.6.

26.3.3 Coefficient G_m Corresponds to a Circle with Frequency m

Based on the aforementioned result for the frequency index $m = 1$, we can easily generalize the geometric interpretation of Fourier coefficients with arbitrary index $m > 0$. Using Eqn. (26.11), the partial reconstruction for the single Fourier coefficient $G_m = (A_m, B_m)$ is the contour $\boldsymbol{g}^{(m)}$, with coordinates

$$
\begin{aligned}
g_k^{(m)} &= G_m \cdot e^{\mathrm{i}\cdot 2\pi m\cdot \frac{k}{M}} && (26.40)\\
&= [A_m + \mathrm{i}\cdot B_m] \cdot [\cos(2\pi m\tfrac{k}{M}) + \mathrm{i}\cdot \sin(2\pi m\tfrac{k}{M})], && (26.41)
\end{aligned}
$$

which again describe a circle with radius $r_m = |G_m|$, phase $\theta_m = \arg(G_m) = \tan^{-1}(B_m/A_m)$, and starting point $g_0^{(m)} = G_m$. In this case, however, the angular velocity is scaled by m, that is, the resulting circle revolves m times faster than the circle for G_1. In other

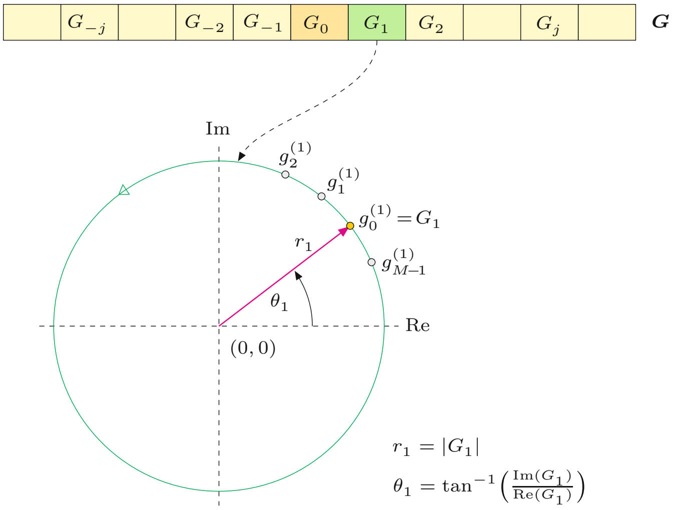


Fig. 26.6 A single DFT coefficient corresponds to a circle. The partial reconstruction from the single DFT coefficient G_m yields a sequence of M points $g_0^{(m)}, \ldots, g_{M-1}^{(m)}$ on a circle centered at the coordinate origin, with radius r_m and starting angle (phase) θ_m.

words, while the contour is traversed once, this circle performs m full revolutions.

Note that G_0 (see Sec. 26.3.1) does not really constitute a special case at all. Formally, it also describes a circle but one that oscillates with zero frequency, that is, all points have the same (constant) position

$$g_k^{(0)} = G_0 \cdot e^{\mathrm{i} \cdot 2\pi m \cdot \frac{k}{M}} = G_0 \cdot e^{\mathrm{i} \cdot 2\pi 0 \cdot \frac{k}{M}} = G_0 \cdot e^0 = G_0, \tag{26.42}$$

for $k = 0, \ldots, M{-}1$, which is equivalent to the curve's centroid $G_0 = (\bar{x}, \bar{y})$, as shown in Eqns. (26.27)–(26.29). Since the corresponding frequency is zero, the point never moves away from G_0.

26.3.4 Negative Frequencies

The DFT spectrum is periodic and defined for all frequencies $m \in \mathbb{Z}$, including negative frequencies. From Eqn. (26.21) we know that for any DFT coefficient with negative index G_{-m} there is an equivalent coefficient G_n whose index n is in the range $0, \ldots, M{-}1$. The partial reconstruction of the spectrum with the single coefficient G_{-m} is

$$g_k^{(-m)} = G_{-m} \cdot e^{-\mathrm{i} \cdot 2\pi m \cdot \frac{k}{M}} = G_n \cdot e^{-\mathrm{i} \cdot 2\pi m \cdot \frac{k}{M}}, \tag{26.43}$$

with $n = -m \bmod M$, which is again a sequence of points on the circle with radius $r_{-m} = r_n = |G_n|$ and phase $\theta_{-m} = \theta_n = \arg(G_n)$. The absolute rotation frequency is m, but this circle spins in the opposite, that is, *clockwise* direction, since angles become increasingly negative with growing k.

26.3.5 Fourier Descriptor Pairs Correspond to Ellipses

It follows therefore that the space-domain circles for the Fourier coefficients G_m and G_{-m} rotate with the same absolute frequency m but with different phase angles θ_m, θ_{-m} and in opposite directions. We denote the tuple

$$\mathrm{FP}_m = (G_{-m}, G_{+m})$$

the "Fourier descriptor pair" (or "FD pair") for the frequency index m. If we perform a partial reconstruction from only the two Fourier coefficients G_{-m}, G_{+m} of this FD pair, we obtain the spatial points

$$\begin{aligned} g_k^{(\pm m)} &= g_k^{(-m)} + g_k^{(+m)} \\ &= G_{-m} \cdot e^{-\mathrm{i}\cdot 2\pi m \cdot \frac{k}{M}} + G_m \cdot e^{\mathrm{i}\cdot 2\pi m \cdot \frac{k}{M}} \\ &= G_{-m} \cdot e^{-\mathrm{i}\cdot \omega_m \cdot \frac{k}{M}} + G_m \cdot e^{\mathrm{i}\cdot \omega_m \cdot \frac{k}{M}}. \end{aligned} \tag{26.44}$$

By Eqn. (26.15) we can expand the result from Eqn. (26.44) to cartesian x/y coordinates as[12]

$$\begin{aligned} x_k^{(\pm m)} &= A_{-m} \cdot \cos\bigl(-\omega_m \cdot \tfrac{k}{M}\bigr) - B_{-m} \cdot \sin\bigl(-\omega_m \cdot \tfrac{k}{M}\bigr) + \\ &\quad\; A_m \cdot \cos(\omega_m \cdot \tfrac{k}{M}) - B_m \cdot \sin(\omega_m \cdot \tfrac{k}{M}) \\ &= (A_{-m} + A_m) \cdot \cos(\omega_m \cdot \tfrac{k}{M}) + (B_{-m} - B_m) \cdot \sin(\omega_m \cdot \tfrac{k}{M}), \end{aligned} \tag{26.45}$$

$$\begin{aligned} y_k^{(\pm m)} &= B_{-m} \cdot \cos\bigl(-\omega_m \cdot \tfrac{k}{M}\bigr) + A_{-m} \cdot \sin\bigl(-\omega_m \cdot \tfrac{k}{M}\bigr) + \\ &\quad\; B_m \cdot \cos\bigl(\omega_m \cdot \tfrac{k}{M}\bigr) + A_m \cdot \sin\bigl(\omega_m \cdot \tfrac{k}{M}\bigr) \\ &= (B_{-m} + B_m) \cdot \cos(\omega_m \cdot \tfrac{k}{M}) - (A_{-m} - A_m) \cdot \sin(\omega_m \cdot \tfrac{k}{M}), \end{aligned} \tag{26.46}$$

for $k = 0, \ldots, M-1$. The 2D point sequence $\boldsymbol{g}^{(\pm m)} = (g_0^{(\pm m)}, \ldots, g_{M-1}^{(\pm m)})$, obtained with Eqns. (26.45) and (26.46), describes an oriented *ellipse* that is centered at the origin (see Fig. 26.7). The parametric equation for this ellipse is

$$\begin{aligned} x_t^{(\pm m)} &= (A_{-m} + A_m) \cdot \cos(\omega_m \cdot t) + (B_{-m} - B_m) \cdot \sin(\omega_m \cdot t), \\ &= (A_{-m} + A_m) \cdot \cos(2\pi m t) + (B_{-m} - B_m) \cdot \sin(2\pi m t), \end{aligned} \tag{26.47}$$

$$\begin{aligned} y_t^{(\pm m)} &= (B_{-m} + B_m) \cdot \cos(\omega_m \cdot t) - (A_{-m} - A_m) \cdot \sin(\omega_m \cdot t) \\ &= (B_{-m} + B_m) \cdot \cos(2\pi m t) - (A_{-m} - A_m) \cdot \sin(2\pi m t), \end{aligned} \tag{26.48}$$

for $t = 0, \ldots, 1$.

Ellipse parameters

In general, the parametric equation of an ellipse with radii a, b, centered at (x_c, y_c) and oriented at an angle α is

$$\begin{aligned} x(\psi) &= x_c + a \cdot \cos(\psi) \cdot \cos(\alpha) - b \cdot \sin(\psi) \cdot \sin(\alpha), \\ y(\psi) &= y_c + a \cdot \cos(\psi) \cdot \sin(\alpha) + b \cdot \sin(\psi) \cdot \cos(\alpha), \end{aligned} \tag{26.49}$$

with $\psi = 0, \ldots, 2\pi$. From Eqns. (26.45) and (26.46) we see that the parameters a_m, b_m, α_m of the ellipse for a single Fourier descriptor pair $\mathrm{FP}_m = (G_{-m}, G_{+m})$ are

$$a_m = r_{-m} + r_{+m} = |G_{-m}| + |G_{+m}|, \tag{26.50}$$

$$b_m = |r_{-m} - r_{+m}| = \bigl|\, |G_{-m}| - |G_{+m}| \,\bigr|, \tag{26.51}$$

$$\begin{aligned} \alpha_m &= \frac{1}{2} \cdot \bigl(\underbrace{\sphericalangle G_{-m}}_{\theta_{-m}} + \underbrace{\sphericalangle G_{+m}}_{\theta_{+m}} \bigr) \\ &= \frac{1}{2} \cdot \Bigl[\tan^{-1}\Bigl(\frac{B_{-m}}{A_{-m}}\Bigr) + \tan^{-1}\Bigl(\frac{B_{+m}}{A_{+m}}\Bigr) \Bigr]. \end{aligned} \tag{26.52}$$

[12] Using the relations $\sin(-a) = -\sin(a)$ and $\cos(-a) = \cos(a)$.

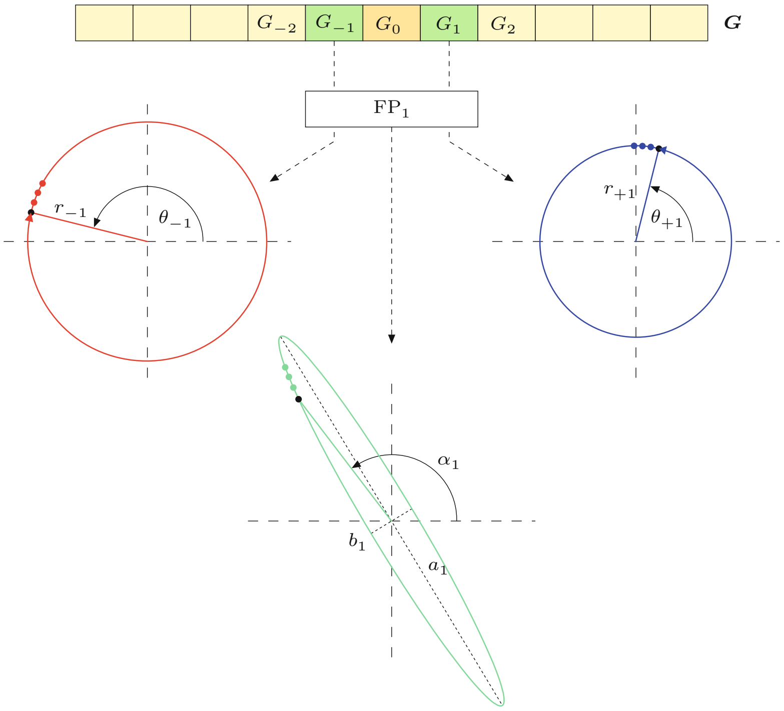


Fig. 26.7
DFT coefficients G_{-m}, G_{+m} form a Fourier descriptor pair FP_m. Each of the two descriptors corresponds to M points on a circle of radius r_{-m}, r_{+m} and phase θ_{-m}, θ_{+m}, respectively, revolving with the same frequency m but in opposite directions. The sum of each point pair is located on an ellipse with radii a_m, b_m and orientation α_m. The orientation α_m of the ellipse's major axis is centered between the starting angles of the circles defined by G_{-m} and G_{+m}; its radii are $a_m = r_{-m} + r_{+m}$ for the major axis and $b_m = |r_{-m} - r_{+m}|$ for the minor axis. The figure shows the situation for $m = 1$.

Like its constituting circles, this ellipse is centered at $(x_c, y_c) = (0, 0)$ and performs m revolutions for one traversal of the contour. G_{-m} specifies the circle

$$z_{-m}(\varphi) = G_{-m} \cdot e^{\mathrm{i}\cdot(-\varphi)} = r_{-m} \cdot e^{\mathrm{i}\cdot(\theta_{-m}-\varphi)}, \tag{26.53}$$

for $\varphi \in [0, 2\pi]$, with starting angle θ_{-m} and radius r_{-m}, rotating in a clockwise direction. Similarly, G_{+m} specifies the circle

$$z_{+m}(\varphi) = G_{+m} \cdot e^{\mathrm{i}\cdot(\varphi)} = r_{+m} \cdot e^{\mathrm{i}\cdot(\theta_{+m}+\varphi)}, \tag{26.54}$$

with starting angle θ_{+m} and radius r_{+m}, rotating in a counter-clockwise direction. Both circles thus rotate at the same angular velocity but in opposite directions, as mentioned before. The corresponding (complex-valued) ellipse points are

$$z_m(\varphi) = z_{-m}(\varphi) + z_{+m}(\varphi). \tag{26.55}$$

The ellipse radius $|z_m(\varphi)|$ is a *maximum* at position $\varphi = \varphi_{\max}$, where the angles on both circles are identical (i.e., the corresponding vectors have the same direction). This occurs when

$$\theta_{-m} - \varphi_{\max} = \theta_{+m} + \varphi_{\max} \qquad \text{or} \qquad \varphi_{\max} = \frac{1}{2} \cdot (\theta_{-m} - \theta_{+m}),$$

that is, at mid-angle between the two starting angles θ_{-m} and θ_{+m}. Therefore, the orientation of the ellipse's major axis is

$$\alpha_m = \theta_{+m} + \frac{\theta_{-m} - \theta_{+m}}{2} = \frac{1}{2} \cdot (\theta_{-m} + \theta_{+m}), \tag{26.56}$$

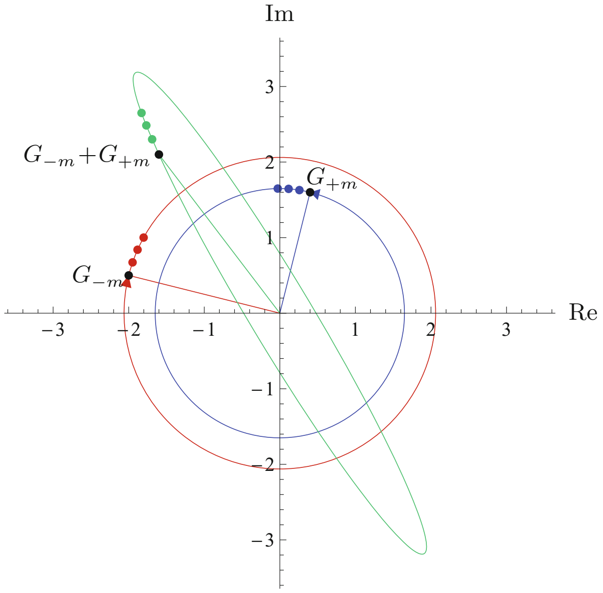


Fig. 26.8 Ellipse created by partial reconstruction from a single Fourier descriptor pair $\mathrm{FP}_m = \big(G_{-m}, G_{+m}\big)$. The two complex-valued Fourier coefficients $G_{-m} = (-2, 0.5)$ and $G_m = (0.4, 1.6)$ represent circles with starting points G_{-m} and G_{+m}, respectively. The circle for G_{-m} (red) rotates in clockwise direction, the circle for G_{+m} (blue) rotates in counter-clockwise direction. The ellipse (green) is the result of point-wise addition of the two circles, as shown for four successive points, starting with point $G_{-m}+G_{+m}$.

as already stated in Eqn. (26.52). At $\varphi = \varphi_{\max}$ the two radial vectors align, and thus the radius of the ellipse's major axis a_m is the sum of the two circle radii, that is,

$$a_m = r_{-m} + r_{+m} \tag{26.57}$$

(cf. Eqn. (26.50)). Analogously, the ellipse radius is *minimized* at position $\varphi = \varphi_{\min}$, where the $z_{-m}(\varphi_{\min})$ and $z_{+m}(\varphi_{\min})$ lie on opposite sides of the circle. This occurs at angle

$$\varphi_{\min} = \varphi_{\max} + \frac{\pi}{2} = \frac{\pi + \theta_{-m} - \theta_{+m}}{2} \tag{26.58}$$

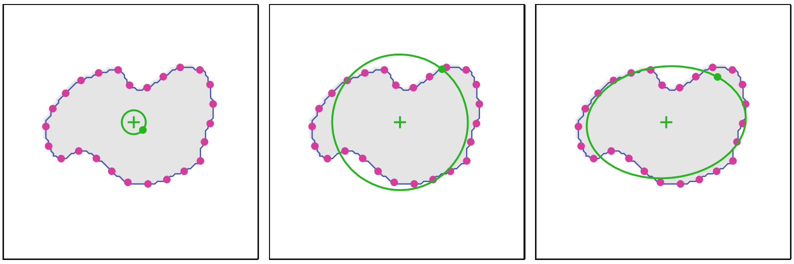
and the corresponding radius for the ellipse's minor axis is (cf. Eqn. (26.51))

$$b_m = r_{+m} - r_{-m}. \tag{26.59}$$

Figure 26.8 illustrates this situation for a specific Fourier descriptor pair $\mathrm{FP}_m = (G_{-m}, G_{+m}) = (-2 + \mathrm{i}\cdot 0.5, 0.4 + \mathrm{i}\cdot 1.6)$. Note that the ellipse parameters a_m, b_m, α_m (see Eqns. (26.50)–(26.52)) are not explicitly required for reconstructing (drawing) the contour, since the ellipse can also be generated by simply adding the x/y-coordinates of the two counter-revolving circles for the participating Fourier descriptors, as given in Eqn. (26.55). Another example is shown in Fig. 26.9.

26.3.6 Shape Reconstruction from Truncated Fourier Descriptors

Due to the periodicity of the DFT spectrum, the complete reconstruction of the contour points g_k from the Fourier coefficients G_m (see Eqn. (26.11)) could also be written with a different summation range, as long as all spectral coefficients are included, that is,



(a) (b) (c)

Fig. 26.9
Partial reconstruction from single coefficients and an FD descriptor pair. The two circles reconstructed from DFT coefficient G_{-1} (a) and coefficient G_{+1} (b) are positioned at the centroid of the contour (G_0). The combined reconstruction for $\left(G_{-1}, G_{+1}\right)$ produces the ellipse in (c). The dots on the green curves show the path position for $t = 0$.

$$g_k = \sum_{m=0}^{M-1} G_m \cdot e^{\mathrm{i}\cdot 2\pi m \cdot \frac{k}{M}} = \sum_{m=m_0}^{m_0+M-1} G_m \cdot e^{\mathrm{i}\cdot 2\pi m \cdot \frac{k}{M}}, \tag{26.60}$$

for any start index $m_0 \in \mathbb{Z}$. As a special (though important) case we can perform the summation symmetrically around the zero index and write

$$g_k = \sum_{m=0}^{M-1} G_m \cdot e^{\mathrm{i}\cdot 2\pi m \cdot \frac{k}{M}} = \sum_{\substack{m= \\ -(M-1)\div 2}}^{M\div 2} G_m \cdot e^{\mathrm{i}\cdot 2\pi m \cdot \frac{k}{M}}. \tag{26.61}$$

To understand the reconstruction in terms of Fourier descriptor pairs, it is helpful to distinguish if M (the number of contour points and Fourier coefficients) is *even* or *odd*.

Odd number of contour points

If M is *odd*, then the spectrum consists of G_0 (representing the contour's centroid) plus exactly $M \div 2$ Fourier descriptor pairs FP_m, with $m = 1, \ldots, M \div 2$.[13] We can thus rewrite Eqn. (26.60) as

$$\begin{aligned} g_k &= \sum_{m=0}^{M-1} G_m \cdot e^{\mathrm{i}\cdot 2\pi m \cdot \frac{k}{M}} = \underbrace{G_0}_{g_k^{(0)}} + \sum_{m=1}^{M\div 2} \underbrace{\left[G_{-m} \cdot e^{-\mathrm{i}\cdot 2\pi m \cdot \frac{k}{M}} + G_m \cdot e^{\mathrm{i}\cdot 2\pi m \cdot \frac{k}{M}}\right]}_{g_k^{(\pm m)} = g_k^{(-m)} + g_k^{(m)}} \\ &= g_k^{(0)} + \sum_{m=1}^{M\div 2} g_k^{(\pm m)} = g_k^{(0)} + g_k^{(\pm 1)} + g_k^{(\pm 2)} + \ldots + g_k^{(\pm M\div 2)}, \end{aligned} \tag{26.62}$$

where $g_k^{(\pm m)}$ denotes the partial reconstruction from the single Fourier descriptor pair FP_m (see Eqn. (26.44)).

As we already know, the partial reconstruction $g_k^{(\pm m)}$ of an individual Fourier descriptor pair FP_m is a set of points on an ellipse that is centered at the origin $(0, 0)$. The partial reconstruction of the *three* DFT coefficients G_0, G_{-m}, G_{+m} (i.e., FP_m plus the single coefficient G_0) is the point sequence

$$g_k^{(-m,0,m)} = g_k^{(0)} + g_k^{(\pm m)}, \tag{26.63}$$

which is the ellipse for $g_k^{(\pm m)}$ shifted to $g_k^{(0)} = (\bar{x}, \bar{y})$, the centroid of the original contour. For example, the partial reconstruction from the coefficients G_{-1}, G_0, G_{+1},

[13] If M is odd, then $M = 2 \cdot (M \div 2) + 1$.

$$g_k^{(-1,0,1)} = g_k^{(-1,\ldots,1)} = g_k^{(0)} + g_k^{(\pm 1)}, \tag{26.64}$$

yields an ellipse with frequency $m = 1$ that revolves around the (fixed) centroid of the original contour. If we add another Fourier descriptor pair FP_2, the resulting reconstruction is

$$g_k^{(-2,\ldots,2)} = \underbrace{g_k^{(0)} + g_k^{(\pm 1)}}_{\text{ellipse 1}} + \underbrace{g_k^{(\pm 2)}}_{\text{ellipse 2}} . \tag{26.65}$$

The resulting ellipse $g_k^{(\pm 2)}$ has the frequency $m = 2$, but note that it is centered at a moving point on the "slower" ellipse (with frequency $m = 1$), that is, ellipse 2 effectively "rides" on ellipse 1. If we add FP_3, its ellipse is again centered at a point on ellipse 2, and so on. For an illustration, see the examples in Figs. 26.11 and 26.12. In general, the ellipse for descriptor pair FP_j revolves around the (moving) center obtained as the superposition of $j-1$ "slower" ellipses,

$$g_k^{(0)} + \sum_{m=1}^{j-1} g_k^{(\pm m)}. \tag{26.66}$$

Consequently, the curve obtained by the partial reconstruction from descriptor pairs $\mathrm{FP}_1, \ldots, \mathrm{FP}_j$ (for $j \leq M \div 2$) is the point sequence

$$g_k^{(-j,\ldots,j)} = g_k^{(0)} + \sum_{m=1}^{j} g_k^{(\pm m)}, \tag{26.67}$$

for $k = 0, \ldots, M-1$. The fully reconstructed shape is the sum of the centroid (defined by G_0) and $M \div 2$ ellipses, one for each Fourier descriptor pair $\mathrm{FP}_1, \ldots, \mathrm{FP}_{M \div 2}$.

Even number of contour points

If M is *even*,[14] then the reconstructed shape is a superposition of the centroid (defined by G_0), $(M-1) \div 2$ ellipses from the Fourier descriptor pairs $\mathrm{FP}_1, \ldots, \mathrm{FP}_{(M-1)\div 2}$, plus one additional *circle* specified by the single (highest frequency) Fourier coefficient $G_{M \div 2}$. The complete reconstruction from an even-length Fourier descriptor can thus be written as

$$g_k = \sum_{m=0}^{M-1} G_m \cdot e^{\mathrm{i} \cdot 2\pi m \cdot \frac{k}{M}} = \underbrace{g_k^{(0)}}_{\text{center}} + \underbrace{\sum_{m=1}^{(M-1)\div 2} g_k^{(\pm m)}}_{(M-1)\div 2 \text{ ellipses}} + \underbrace{g_k^{(M \div 2)}}_{\text{1 circle}} . \tag{26.68}$$

The single high-frequency circle associated with $g_k^{(M \div 2)}$ has its (moving) center at the sum of all lower-frequency ellipses that correspond to the Fourier coefficients $G_{-m}, \ldots, G_{+m}$, with $m < (M \div 2)$.

Reconstruction algorithm

Algorithm 26.4 describes the reconstruction of shapes from a Fourier descriptor using only a specified number (M_{p}) of Fourier descriptor pairs. The number of points on the reconstructed contour (N) can be freely chosen.

[14] In this case, $M = 2 \cdot (M \div 2) = (M-1) \div 2 + 1 + M \div 2$.

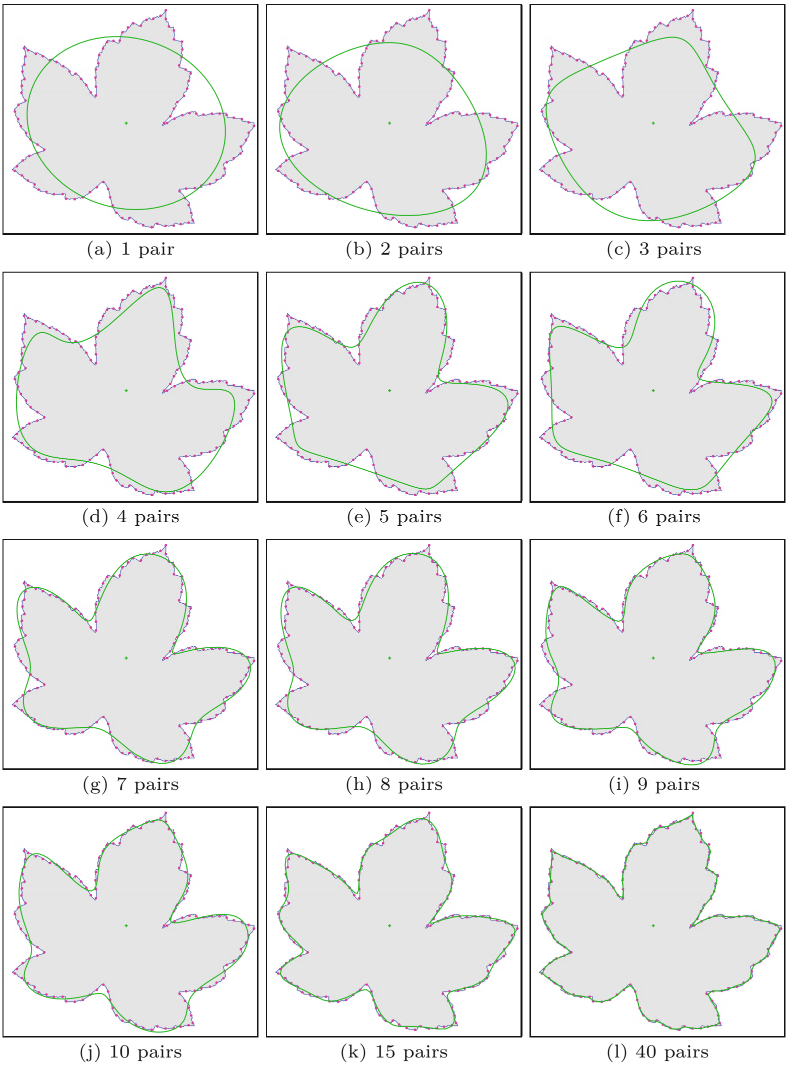

(a) 1 pair (b) 2 pairs (c) 3 pairs (d) 4 pairs (e) 5 pairs (f) 6 pairs (g) 7 pairs (h) 8 pairs (i) 9 pairs (j) 10 pairs (k) 15 pairs (l) 40 pairs

Fig. 26.10
Partial shape reconstruction from a limited set of Fourier descriptor pairs. The full descriptor contains 125 coefficients (G_0 plus 62 FD pairs).

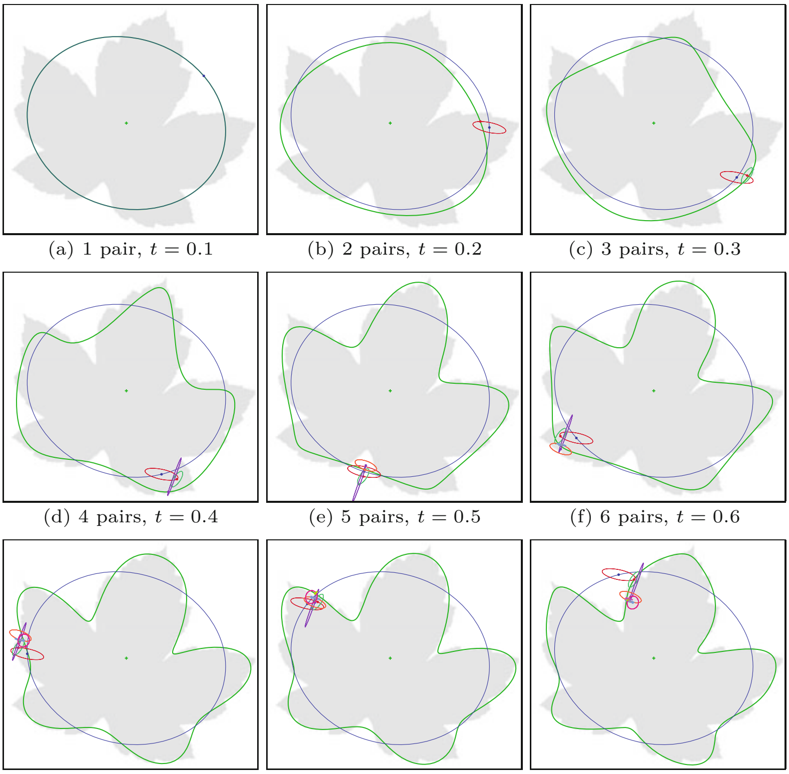
26.3.7 Fourier Descriptors from Unsampled Polygons

The requirement to distribute sample points uniformly along the contour path stems from classical signal processing and Fourier theory, where uniform sampling is a common assumption. However, as shown in [143] (see also [183, 262]), the Fourier descriptors for a polygonal shape can be calculated directly from the original polygon vertices without sub-sampling the contour. This "trigonometric" approach, described in the following, works for arbitrary (convex and non-convex) polygons.

We assume that the shape is specified as a sequence of P points $V = (\boldsymbol{v}_0, \ldots, \boldsymbol{v}_{P-1})$, with $V(i) = \boldsymbol{v}_i = (x_i, y_i)$ representing the 2D vertices of a closed polygon. We define the quantities

$$\boldsymbol{d}(i) = \boldsymbol{v}_{(i+1) \bmod P} - \boldsymbol{v}_i \qquad \text{and} \qquad \lambda(i) = \|\boldsymbol{d}(i)\|\,, \tag{26.69}$$

for $i = 0, \ldots, P-1$, where $\boldsymbol{d}(i)$ is the vector representing the polygon segment between the vertices $\boldsymbol{v}_i, \boldsymbol{v}_{i+1}$, and $\lambda(i)$ is the length of that

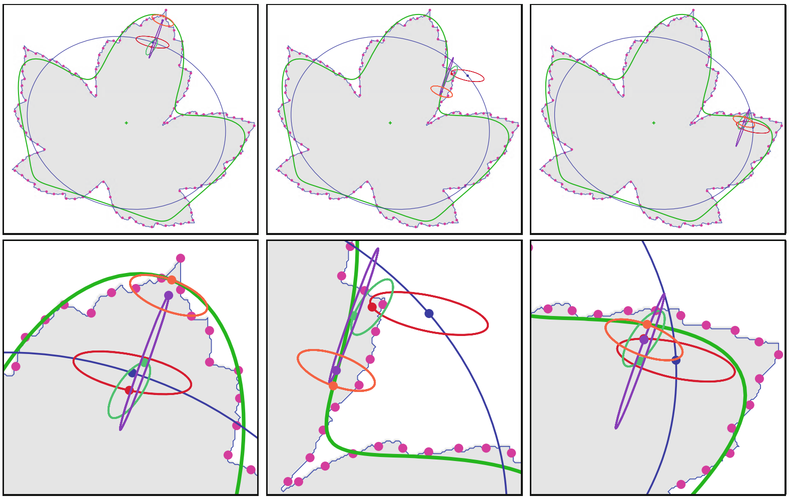


(a) 1 pair, $t = 0.1$ (b) 2 pairs, $t = 0.2$ (c) 3 pairs, $t = 0.3$

(d) 4 pairs, $t = 0.4$ (e) 5 pairs, $t = 0.5$ (f) 6 pairs, $t = 0.6$

(g) 7 pairs, $t = 0.7$ (h) 8 pairs, $t = 0.8$ (i) 9 pairs, $t = 0.9$

Fig. 26.11
Partial reconstruction by ellipse superposition (details). The green curve shows the partial reconstruction from $1, \ldots, 9$ FD pairs. This curve performs one full revolution as the path parameter t runs from 0 to 1. Subfigures (a–i) depict the situation for $1, \ldots, 9$ FD pairs and different path positions $t = 0.1, 0.2, \ldots, 0.9$. Each Fourier descriptor pair corresponds to an ellipse that is centered at the current position t on the previous ellipse. The individual Fourier descriptor pair FP_1 in (a) corresponds to a single ellipse. In (b), the point for $t = 0.2$ on the blue ellipse (for FP_1) is the center of the red ellipse (for FP_2). In (c), the green ellipse (for FP_3) is centered at the point marked on the previous ellipse, and so on. The reconstructed shape is obtained by superposition of all ellipses. See Fig. 26.12 for a detailed view.



(a) $t = 0.0$ (b) $t = 0.1$ (c) $t = 0.2$

Fig. 26.12
Partial reconstruction by ellipse superposition (details). The green curve shows the partial reconstruction from 5 FD pairs $\mathrm{FP}_1, \ldots, \mathrm{FP}_5$. This curve performs one full revolution as the path parameter t runs from 0 to 1. Subfigures (a–c) show the composition of the contour by superposition of the 5 ellipses, each corresponding to one FD pair, at selected positions $t = 0.0, 0.1, 0.2$. The blue ellipse corresponds to FP_1 and revolves once for $t = 0, \ldots, 1$. The blue dot on this ellipse marks the position t, which serves as the center of the next (red) ellipse corresponding to FP_2. This ellipse makes 2 revolutions for $t = 0, \ldots, 1$ and the red dot for position t is again the center of green ellipse (for FP_3), and so on. Position t on the orange ellipse (for FP_1) coincides with the final reconstruction (green curve). The original contour was sampled at 125 equidistant points.

segment. We also define

$$L(i) = \sum_{j=0}^{i-1} \lambda(j), \tag{26.70}$$

for $i = 0, \ldots, P$, which is the cumulative length of the polygon path from the start vertex $\boldsymbol{v}_0$ to vertex $\boldsymbol{v}_i$, such that $L(0)$ is zero and $L(P)$ is the closed path length of the polygon V.

Alg. 26.4
Partial shape reconstruction from a truncated Fourier descriptor $\boldsymbol{G}$. The shape is reconstructed by considering up to M_p Fourier descriptor pairs. The resulting sequence of contour points may be of arbitrary length (N). See Figs. 26.10–26.12 for examples.

```
GetPartialReconstruction($\boldsymbol{G}, M_\mathrm{p}, N$)
    Input: $\boldsymbol{G} = (G_0, \ldots, G_{M-1})$, Fourier descriptor with $M$ coefficients; $M_\mathrm{p}$, number of Fourier descriptor pairs to consider; $N$, number of points on the reconstructed shape. Returns the reconstructed contour as a sequence of $N$ complex values.

    Create map $\boldsymbol{g} \colon [0, N-1] \to \mathbb{C}$
    $M \leftarrow |\boldsymbol{G}|$                                   ▷ total number of Fourier coefficients
    $M_\mathrm{p} \leftarrow \min(M_\mathrm{p}, (M-1) \div 2)$        ▷ available Fourier coefficient pairs
    for $k \leftarrow 0, \ldots, N-1$ do
        $t \leftarrow k/N$                                         ▷ continuous path position $t \in [0, 1]$
        $\boldsymbol{g}(k) \leftarrow$ GetSinglePoint($\boldsymbol{G}, -M_\mathrm{p}, M_\mathrm{p}, t$)      ▷ see below
    return $\boldsymbol{g}$.

GetSinglePoint($\boldsymbol{G}, m_-, m_+, t$)
    Returns a single point (as a complex value) on the reconstructed shape for the continuous path position $t \in [0, 1]$, based on the Fourier coefficients $\boldsymbol{G}(m_-), \ldots, \boldsymbol{G}(m_+)$.
    $M \leftarrow |\boldsymbol{G}|$
    $x \leftarrow 0, \quad y \leftarrow 0$
    for $m \leftarrow m_-, \ldots, m_+$ do
        $\phi \leftarrow 2 \cdot \pi \cdot m \cdot t$
        $G \leftarrow \boldsymbol{G}(m \bmod M)$
        $A \leftarrow \mathrm{Re}(G), \quad B \leftarrow \mathrm{Im}(G)$
        $x \leftarrow x + A \cdot \cos(\phi) - B \cdot \sin(\phi)$
        $y \leftarrow y + A \cdot \sin(\phi) + B \cdot \cos(\phi)$

    return $(x + \mathrm{i}\, y)$.
```

For a (freely chosen) number of Fourier descriptor pairs (M_p), the corresponding Fourier descriptor $\boldsymbol{G} = (G_{-M_\mathrm{p}}, \ldots, G_0, \ldots, G_{+M_\mathrm{p}})$, has $2M_\mathrm{p} + 1$ complex-valued coefficients G_m, where

$$G_0 = a_0 + \mathrm{i} \cdot c_0 \tag{26.71}$$

and the remaining coefficients are calculated as

$$G_{+m} = (a_m + d_m) + \mathrm{i} \cdot (c_m - b_m), \tag{26.72}$$
$$G_{-m} = (a_m - d_m) + \mathrm{i} \cdot (c_m + b_m), \tag{26.73}$$

from the "trigonometric coefficients" a_m, b_m, c_m, d_m. As described in [143], these coefficients are obtained directly from the P polygon vertices $\boldsymbol{v}_i$ as

$$\begin{pmatrix} a_0 \\ c_0 \end{pmatrix} = \boldsymbol{v}_0 + \frac{\sum\limits_{i=0}^{P-1} \bigl[\frac{L^2(i+1)-L^2(i)}{2\,\lambda(i)} \cdot \boldsymbol{d}(i) + \lambda(i) \cdot \sum\limits_{j=0}^{i-1} \boldsymbol{d}(j) - \boldsymbol{d}(i) \cdot \sum\limits_{j=0}^{i-1} \lambda(j) \bigr]}{L(P)} \tag{26.74}$$

(representing the shape's center), with $\boldsymbol{d}, \lambda, L$ as defined in Eqns. (26.69) and (26.70). This can be simplified to

$$\begin{pmatrix} a_0 \\ c_0 \end{pmatrix} = \boldsymbol{v}_0 + \frac{\sum\limits_{i=0}^{P-1} \bigl[\bigl(\frac{L^2(i+1)-L^2(i)}{2\,\lambda(i)} - L(i) \bigr) \cdot \boldsymbol{d}(i) + \lambda(i) \cdot (\boldsymbol{v}_i - \boldsymbol{v}_0) \bigr]}{L(P)}. \tag{26.75}$$

```
FourierDescriptorFromPolygon($V, M_\mathrm{p}$)
    Input: $V = (\boldsymbol{v}_0,\ldots,\boldsymbol{v}_{P-1})$, a sequence of $P$ points representing
    the vertices of a closed 2D polygon; $M_\mathrm{p}$, the desired number of
    FD pairs. Returns a new Fourier descriptor of length $2M_\mathrm{p}+1$.

    $P \leftarrow |V|$                                   ▷ number of polygon vertices in $V$
    $M \leftarrow 2 \cdot M_\mathrm{p}+1$                ▷ number of Fourier coefficients in $\boldsymbol{G}$
    Create maps $\boldsymbol{d}\colon [0,P-1] \to \mathbb{R}^2$,   $\lambda\colon [0,P-1] \to \mathbb{R}$,
                $L\colon [0,P] \to \mathbb{R}$,   $\boldsymbol{G}\colon [0,M-1] \to \mathbb{C}$

    $L(0) \leftarrow 0$
    for $i \leftarrow 0,\ldots,P-1$ do
        $\boldsymbol{d}(i) \leftarrow V((i+1) \bmod P) - V(i)$          ▷ Eq. 26.69
        $\lambda(i) \leftarrow \|\boldsymbol{d}(i)\|$
        $L(i+1) \leftarrow L(i) + \lambda(i)$

    $\binom{a}{c} \leftarrow \binom{0}{0}$                              ▷ $a = a_0, c = c_0$
    for $i \leftarrow 0,\ldots,P-1$ do
        $s \leftarrow \frac{L^2(i+1)-L^2(i)}{2\cdot\lambda(i)} - L(i)$
        $\binom{a}{c} \leftarrow \binom{a}{c} + s\cdot\boldsymbol{d}(i) + \lambda(i)\cdot(V(i)-V(0))$     ▷ Eq. 26.75
    $\boldsymbol{G}(0) \leftarrow \boldsymbol{v}_0 + \frac{1}{L(P)}\cdot\binom{a}{c}$                ▷ Eq. 26.71

    for $m \leftarrow 1,\ldots,M_\mathrm{p}$ do                          ▷ for FD-pairs $G_{\pm 1},\ldots,G_{\pm M_\mathrm{p}}$
        $\binom{a}{c} \leftarrow \binom{0}{0}$,  $\binom{b}{d} \leftarrow \binom{0}{0}$        ▷ $a_m, b_m, c_m, d_m$
        for $i \leftarrow 0,\ldots,P-1$ do
            $\omega_0 \leftarrow 2\pi m \cdot \frac{L(i)}{L(P)}$
            $\omega_1 \leftarrow 2\pi m \cdot \frac{L((i+1) \bmod P)}{L(P)}$
            $\binom{a}{c} \leftarrow \binom{a}{c} + \frac{\cos(\omega_1)-\cos(\omega_0)}{\lambda(i)}\cdot\boldsymbol{d}(i)$     ▷ Eq. 26.76
            $\binom{b}{d} \leftarrow \binom{b}{d} + \frac{\sin(\omega_1)-\sin(\omega_0)}{\lambda(i)}\cdot\boldsymbol{d}(i)$     ▷ Eq. 26.77
        $\boldsymbol{G}(m) \leftarrow \frac{L(P)}{(2\pi m)^2}\cdot\binom{a+d}{c-b}$                ▷ Eq. 26.72
        $\boldsymbol{G}(-m \bmod M) \leftarrow \frac{L(P)}{(2\pi m)^2}\cdot\binom{a-d}{c+b}$        ▷ Eq. 26.73

    return $\boldsymbol{G}$.
```

Alg. 26.5 Fourier descriptor from trigonometric data (arbitrary polygons). Parameter M_p specifies the number of Fourier coefficient pairs.

The remaining coefficients a_m, b_m, c_m, d_m $(m = 1,\ldots,M_\mathrm{p})$ are calculated as

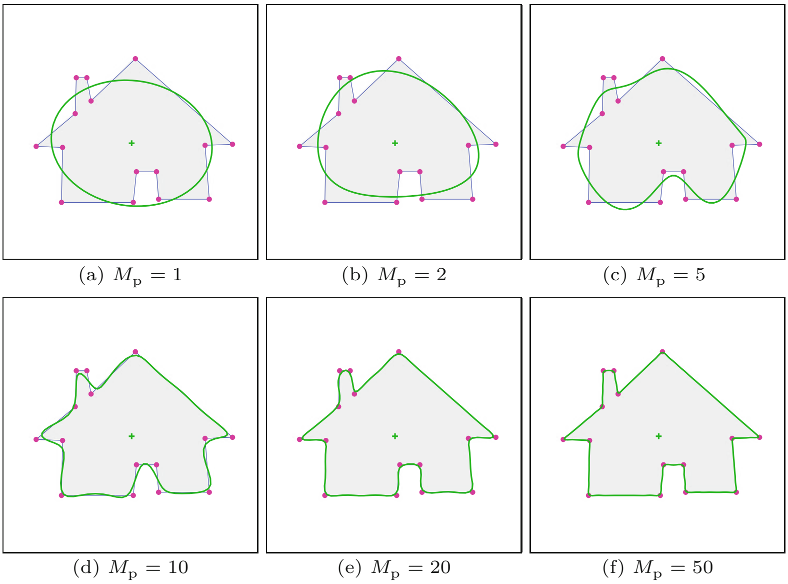
$$\begin{pmatrix} a_m \\ c_m \end{pmatrix} = \frac{L(P)}{(2\pi m)^2}\cdot\sum_{i=0}^{P-1}\Bigl[\frac{\cos\bigl(2\pi m\frac{L(i+1)}{L(P)}\bigr) - \cos\bigl(2\pi m\frac{L(i)}{L(P)}\bigr)}{\lambda(i)}\cdot\boldsymbol{d}(i)\Bigr], \tag{26.76}$$

$$\begin{pmatrix} b_m \\ d_m \end{pmatrix} = \frac{L(P)}{(2\pi m)^2}\cdot\sum_{i=0}^{P-1}\Bigl[\frac{\sin\bigl(2\pi m\frac{L(i+1)}{L(P)}\bigr) - \sin\bigl(2\pi m\frac{L(i)}{L(P)}\bigr)}{\lambda(i)}\cdot\boldsymbol{d}(i)\Bigr], \tag{26.77}$$

respectively. The complete calculation of a Fourier descriptor from trigonometric coordinates (i.e., from arbitrary polygons) is summarized in Alg. 26.5.

An approximate reconstruction of the original shape can be obtained directly from the trigonometric coefficients a_m, b_m, c_m, d_m de-

Fig. 26.13
Fourier descriptors calculated from trigonometric data (arbitrary polygons). Shape reconstructions with different numbers of Fourier descriptor pairs (M_p).


(a) $M_\mathrm{p} = 1$
(b) $M_\mathrm{p} = 2$
(c) $M_\mathrm{p} = 5$
(d) $M_\mathrm{p} = 10$
(e) $M_\mathrm{p} = 20$
(f) $M_\mathrm{p} = 50$

fined in Eqns. (26.75) and (26.76) as[15]

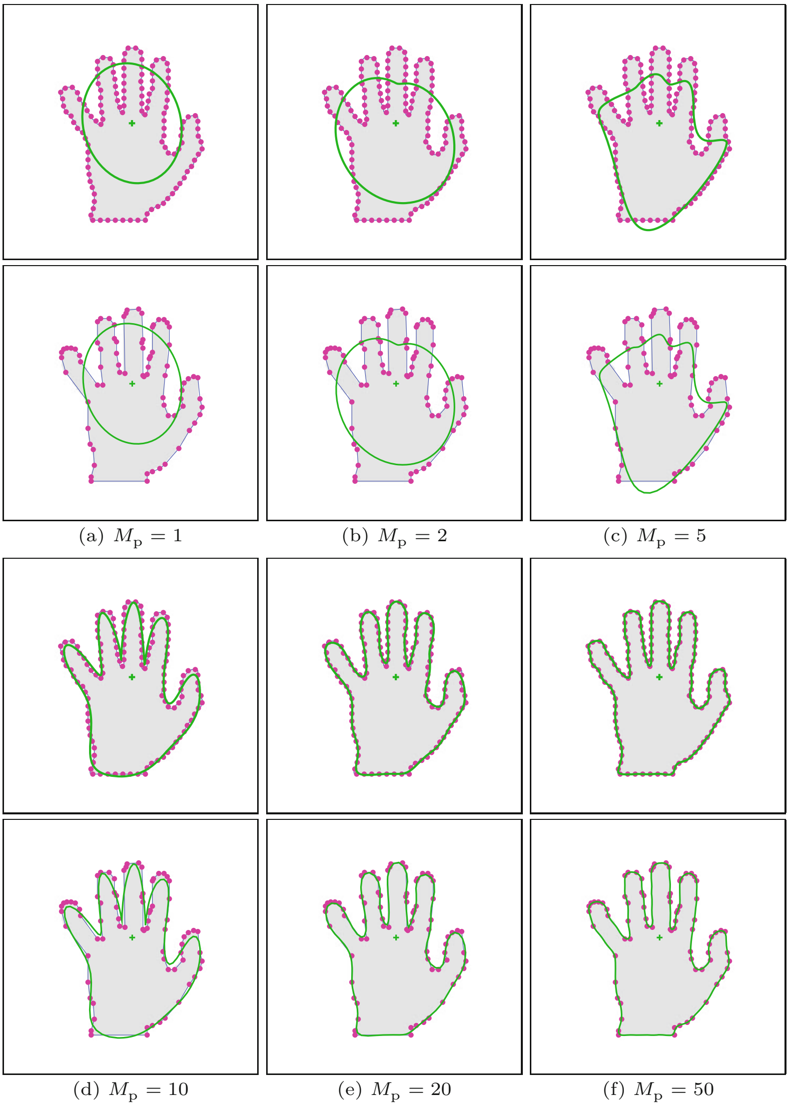
$$\boldsymbol{x}(t) = \begin{pmatrix} a_0 \\ c_0 \end{pmatrix} + \sum_{m=1}^{M_\mathrm{p}} \Big[\begin{pmatrix} a_m \\ c_m \end{pmatrix} \cdot \cos(2\pi m t) + \begin{pmatrix} b_m \\ d_m \end{pmatrix} \cdot \sin(2\pi m t) \Big], \quad (26.78)$$

for $t = 0, \ldots, 1$. Of course, this reconstruction can also be calculated from the actual DFT coefficients $\boldsymbol{G}$, as described in Eqn. (26.20). Again the reconstruction error is reduced by increasing the number of Fourier descriptor pairs (M_p), as demonstrated in Fig. 26.13.[16] The reconstruction is theoretically perfect as M_p goes to infinity.

Working with the trigonometric technique is an advantage, in particular, if the boundary curvature along the outline varies strongly. For example, the silhouette of a human hand typically exhibits high curvature along the fingertips while other contour sections are almost straight. Capturing the high-curvature parts requires a significantly higher density of samples than in the smooth sections, as illustrated in Fig. 26.14. This figure compares the partial shape reconstructions obtained from Fourier descriptors calculated with uniform and non-uniform contour sampling, using identical numbers of Fourier descriptor pairs (M_p). Note that the coefficients (and thus the reconstructions) are very similar, although considerably fewer samples were used for the trigonometric approach.

[15] Note the analogy to the elliptical reconstruction in Eqns. (26.47) and (26.48).

[16] Most test images used in this chapter were taken from the Kimia dataset [134]. A selected subset of modified images taken from this dataset is available on the book's website.



(a) $M_{\mathrm{p}} = 1$

(b) $M_{\mathrm{p}} = 2$

(c) $M_{\mathrm{p}} = 5$

(d) $M_{\mathrm{p}} = 10$

(e) $M_{\mathrm{p}} = 20$

(f) $M_{\mathrm{p}} = 50$

Fig. 26.14
Fourier descriptors from uniformly sampled vs. non-uniformly sampled (trigonometric) contours. Partial constructions from Fourier descriptors obtained from uniformly sampled contours (rows 1, 3) and non-uniformly sampled contours (rows 2, 4), for different numbers of Fourier descriptor pairs (M_{p}).

26.4 Effects of Geometric Transformations

To be useful for comparing shapes, a representation should be invariant against a certain set of geometric transformations. Typically, a minimal requirement for robust 2D shape matching is invariance to translation, scale changes, and rotation. Fourier shape descriptors in their basic form are *not* invariant under any of these transformations but they can be modified to satisfy these requirements. In this section, we discuss the effects of such transformations upon the corresponding Fourier descriptors. The steps involved for making Fourier descriptors invariant are discussed subsequently in Sec. 26.5.

26.4.1 Translation

As described in Sec. 26.3.1, the coefficient G_0 of a Fourier descriptor $\boldsymbol{G}$ corresponds to the centroid of the encoded contour. Moving the points g_k of a shape $\boldsymbol{g}$ in the complex plane by some constant $z \in \mathbb{C}$,

$$g'_k = g_k + z, \tag{26.79}$$

for $k = 0, \ldots, M-1$, only affects Fourier coefficient G_0, that is,

$$G'_m = \begin{cases} G_m + z & \text{for } m = 0, \\ G_m & \text{for } m \neq 0. \end{cases} \tag{26.80}$$

To make an FD invariant against translation, it is thus sufficient to zero its G_0 coefficient, thereby shifting the shape's center to the origin of the coordinate system. Alternatively, translation invariant matching of Fourier descriptors is achieved by simply ignoring coefficient G_0.

26.4.2 Scale Change

Since the Fourier transform is a linear operation, scaling a 2D shape $\boldsymbol{g}$ uniformly by a real-valued factor s,

$$g'_k = s \cdot g_k, \tag{26.81}$$

also scales the corresponding Fourier spectrum by the same factor, that is,

$$G'_m = s \cdot G_m, \tag{26.82}$$

for $m = 1, \ldots, M-1$. Note that scaling by $s = -1$ (or any other negative factor) corresponds to *reversing* the ordering of the samples along the contour (see also Sec. 26.4.6). Given the fact that the DFT coefficient G_1 represents a circle whose radius $r_1 = |G_1|$ is proportional to the size of the original shape (see Sec. 26.3.2), the Fourier descriptor $\boldsymbol{G}$ could be normalized for scale by setting

$$G^{\mathrm{S}}_m = \frac{1}{|G_1|} \cdot G_m, \tag{26.83}$$

for $m = 1, \ldots, M-1$, such that $\left|G^{\mathrm{S}}_1\right| = 1$. Although it is common to use only G_1 for scale normalization, this coefficient may be relatively small (and thus unreliable) for certain shapes. We therefore prefer to normalize the complete Fourier coefficient vector to achieve scale invariance (see Sec. 26.5.1).

26.4.3 Rotation

If a given shape is rotated about the origin by some angle β, then each contour point $\boldsymbol{v}_k = (x_k, y_k)$ moves to a new position

$$\boldsymbol{v}'_k = \begin{pmatrix} x'_k \\ y'_k \end{pmatrix} = \begin{pmatrix} \cos(\beta) & -\sin(\beta) \\ \sin(\beta) & \cos(\beta) \end{pmatrix} \cdot \begin{pmatrix} x_k \\ y_k \end{pmatrix}. \tag{26.84}$$

If the 2D contour samples are represented as complex values $g_k = x_k + \mathrm{i} \cdot y_k$, this rotation can be expressed as a multiplication

$$g'_k = e^{\mathrm{i}\beta} \cdot g_k, \tag{26.85}$$

with the complex factor $e^{\mathrm{i}\beta} = \cos(\beta) + \mathrm{i} \cdot \sin(\beta)$. As in Eqn. (26.82), we can use the linearity of the DFT to predict the effects of rotating the shape $\boldsymbol{g}$ by angle β as

$$G'_m = e^{\mathrm{i}\beta} \cdot G_m, \tag{26.86}$$

for $m = 0, \ldots, M-1$. Thus, the spatial rotation in Eqn. (26.85) multiplies each DFT coefficient G_m by the *same* complex factor $e^{\mathrm{i}\beta}$, which has unit magnitude. Since

$$e^{\mathrm{i}\beta} \cdot G_m = e^{\mathrm{i}(\theta_m + \beta)} \cdot |G_m|, \tag{26.87}$$

this only rotates the *phase* $\theta_m = \sphericalangle G_m$ of each coefficient by the *same* angle β, without changing its *magnitude* $|G_m|$.

26.4.4 Shifting the Sampling Start Position

Despite the implicit periodicity of the boundary sequence and the corresponding DFT spectrum, Fourier descriptors are generally not the same if sampling starts at different positions along the contour. Given a periodic sequence of M discrete contour samples $\boldsymbol{g} = (g_0, g_1, \ldots, g_{M-1})$, we select another sequence $\boldsymbol{g}' = (g'_0, g'_1, \ldots) = (g_{k_\mathrm{s}}, g_{k_\mathrm{s}+1}, \ldots)$, again of length M, from the same set of samples but starting at point k_s, that is,

$$g'_k = g_{(k+k_\mathrm{s}) \bmod M} \, . \tag{26.88}$$

This is equivalent to *shifting* the original signal $\boldsymbol{g}$ circularly by $-k_\mathrm{s}$ positions. The well-known "shift property" of the Fourier transform[17] states that such a change to the "signal" $\boldsymbol{g}$ modifies the corresponding DFT coefficients G_m (for the original contour sequence) to

$$G'_m = e^{\mathrm{i} \cdot m \cdot \frac{2\pi k_\mathrm{s}}{M}} \cdot G_m = e^{\mathrm{i} \cdot m \cdot \varphi_\mathrm{s}} \cdot G_m, \tag{26.89}$$

where $\varphi_\mathrm{s} = \frac{2\pi k_\mathrm{s}}{M}$ is a constant phase angle that is obviously proportional to the chosen start position k_s. Note that, in Eqn. (26.89), each DFT coefficient G_m is multiplied by a *different* complex quantity $e^{\mathrm{i} \cdot m \cdot \varphi_\mathrm{s}}$, which is of unit magnitude and varies with the frequency index m. In other words, the *magnitude* of any DFT coefficient G_m is again preserved but its *phase* changes individually. The coefficients of any Fourier descriptor pair $\mathrm{FP}_m = (G_{-m}, G_{+m})$ thus become

$$G'_{-m} = e^{-\mathrm{i} \cdot m\varphi_\mathrm{s}} \cdot G_{-m} \quad \text{and} \quad G'_{+m} = e^{\mathrm{i} \cdot m\varphi_\mathrm{s}} \cdot G_{+m}, \tag{26.90}$$

that is, coefficient G_{-m} is rotated by the angle $-m \cdot \varphi_\mathrm{s}$ and G_{+m} is rotated by $m \cdot \varphi_\mathrm{s}$. In other words, a circular shift of the signal by $-k_\mathrm{s}$ samples rotates the coefficients G_{-m}, G_{+m} by the same angle $m \cdot \varphi_\mathrm{s}$ but in *opposite* directions. Therefore, the sum of both angles stays the same, that is,

$$\sphericalangle G'_{-m} + \sphericalangle G'_{+m} \;\equiv\; \sphericalangle G_{-m} + \sphericalangle G_{+m}. \tag{26.91}$$

[17] See Chapter 18, Sec. 18.1.6.

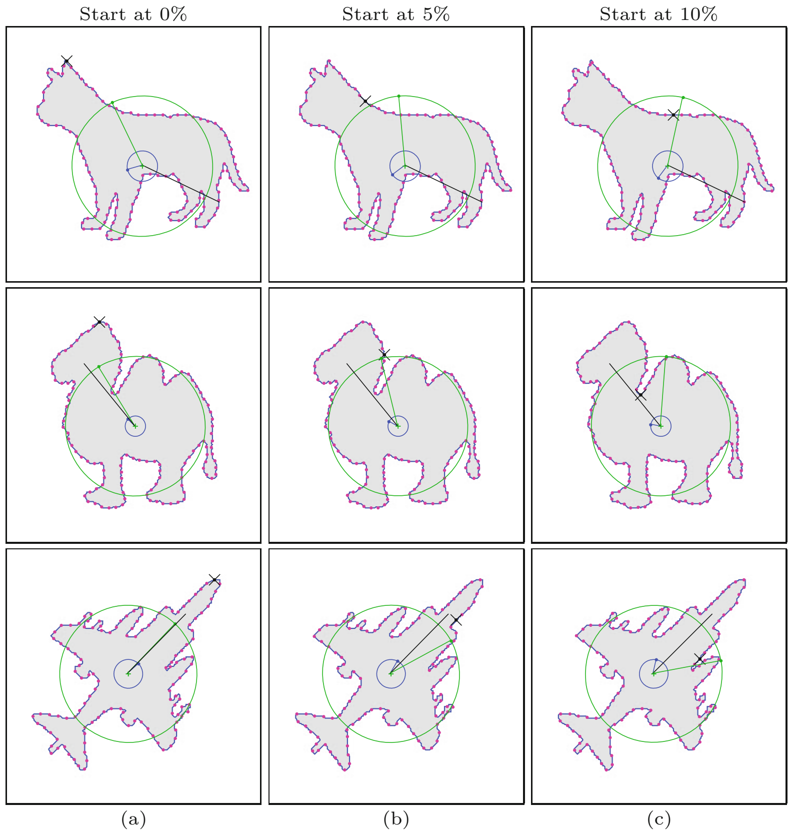


Fig. 26.15
Effects of choosing different start points for contour sampling. The start point (marked × on the contour) is set to 0%, 5%, 10% of the contour path length. The blue and green circles represent the partial reconstruction from single DFT coefficients G_{-1} and G_{+1}, respectively. The dot on each circle and the associated radial line shows the phase of the corresponding coefficient. The black line indicates the average orientation $(\sphericalangle G_{-1} + \sphericalangle G_{+1})/2$. It can be seen that the phase difference of G_{-1} and G_{+1} is directly related to the start position, but the average *orientation* (black line) remains unchanged.

In particular, we see from Eqn. (26.90) that shifting the start position modifies the coefficients of the *first* descriptor pair $\mathrm{FP}_1 = (G_{-1}, G_{+1})$ to

$$G'_{-1} = e^{-\mathrm{i}\cdot\varphi_\mathrm{s}} \cdot G_{-1} \qquad \text{and} \qquad G'_{+1} = e^{\mathrm{i}\cdot\varphi_\mathrm{s}} \cdot G_{+1}. \tag{26.92}$$

The resulting *absolute* phase change of the coefficients G_{-1}, G_{+1} is $-\varphi_\mathrm{s}, +\varphi_\mathrm{s}$, respectively, and thus the change in phase *difference* is $2 \cdot \varphi_\mathrm{s}$, that is, the phase difference between the coefficients G_{-1}, G_{+1} is proportional to the chosen start position k_s (see Fig. 26.15).

26.4.5 Effects of Phase Removal

As described in the two previous sections, shape rotation (Sec. 26.4.3) and shift of start point (Sec. 26.4.4) both affect the phase of the Fourier coefficients but not their magnitude. The fact that magnitude is preserved suggests a simple solution for rotation invariant shape matching by simply ignoring the phase of the coefficients and comparing only their magnitude (see Sec. 26.6). Although this comes at the price of losing shape descriptiveness, magnitude-only descriptors are often used for shape matching. Clearly, the original shape cannot be reconstructed from a magnitude-only Fourier descriptor, as demonstrated in Fig. 26.16. It shows the reconstruction of shapes from Fourier descriptors with the phase of all coefficients set to zero, except for G_{-1}, G_0 and G_{+1} (to preserve the shape's center and main orientation).

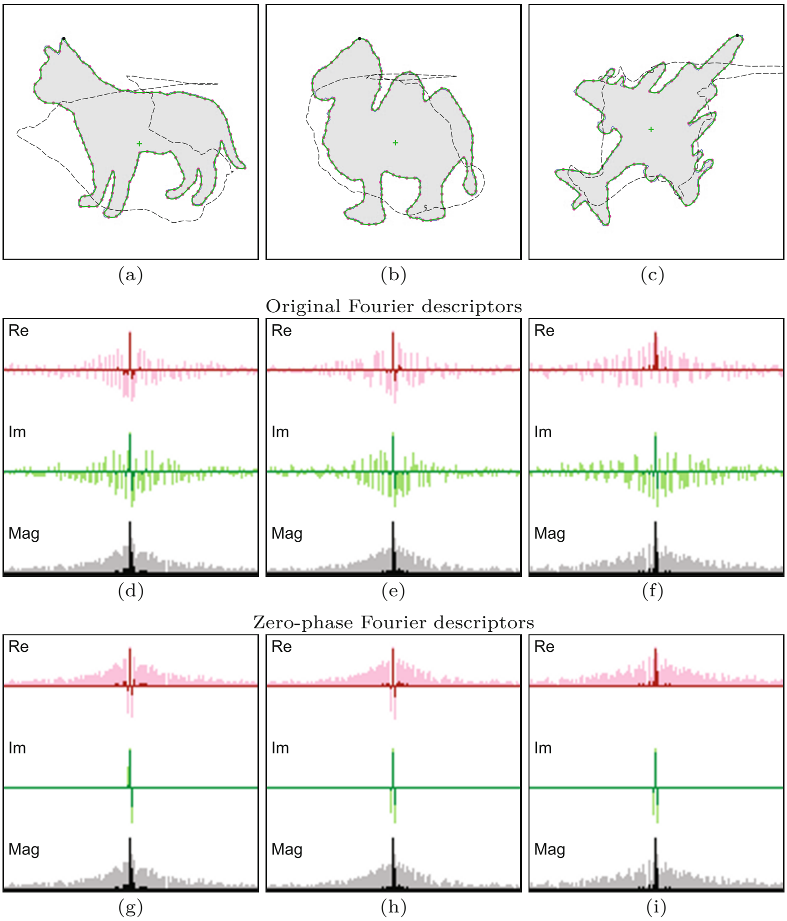


Fig. 26.16
Effects of removing phase information. Original shapes and reconstruction after phase removal (a–c). Original Fourier coefficients (d–f) and zero-phase coefficients (g–i). The red and green plots in (d–i) show the real and imaginary components, respectively; gray plots show the coefficient magnitude. Dark-shaded bars correspond to the actual values, light-shaded bars are logarithmic values. The magnitude of the coefficients in (d–f) is the same as in (g–i).

26.4.6 Direction of Contour Traversal

If the traversal direction of the contour samples is reversed, the coefficients of all Fourier descriptor pairs are exchanged, that is,

$$G'_m = G_{-m \bmod M}. \tag{26.93}$$

This is equivalent to scaling the original shape by $s = -1$, as pointed out in Section 26.4.2. However, this is typically of no relevance in matching, since we can specify all contours to be sampled in either clockwise or counter-clockwise direction.

26.4.7 Reflection (Symmetry)

Mirroring or reflecting a contour about the x-axis is equivalent to replacing each complex-valued point $g_k = x_k + \mathrm{i} \cdot y_k$ by its *complex conjugate* g^*_k, that is,

$$g'_k = g^*_k = x_k - \mathrm{i} \cdot y_k. \tag{26.94}$$

This change to the "signal" results in a modified DFT spectrum with coefficients

$$G'_m = G^*_{-m \bmod M}, \tag{26.95}$$

Table 26.1
Effects of spatial transformations upon the corresponding DFT spectrum. The original contour samples are denoted g_k, the DFT coefficients are G_m.

Operation	Contour samples	DFT coefficients
Forward transformation	g_k, for $k = 0, \ldots, M-1$	$G_m = \frac{1}{M} \cdot \sum_{k=0}^{M-1} g_k \cdot e^{-\mathrm{i} 2\pi m \frac{k}{M}}$
Inverse transformation	$g_k = \sum_{m=0}^{M-1} G_m \cdot e^{\mathrm{i} 2\pi m \frac{k}{M}}$	G_m, for $m = 0, \ldots, M-1$
Translation (by $z \in \mathbb{C}$)	$g'_k = g_k + z$	$G'_m = \begin{cases} G_m + z & \text{for } m = 0 \\ G_m & \text{otherwise} \end{cases}$
Uniform scaling (by $s \in \mathbb{R}$)	$g'_k = s \cdot g_k$	$G'_m = s \cdot G_m$
Rotation about the origin (by β)	$g'_k = e^{\mathrm{i} \cdot \beta} \cdot g_k$	$G'_m = e^{\mathrm{i} \cdot \beta} \cdot G_m$
Shift of start position (by k_{s})	$g'_k = g_{(k+k_{\mathrm{s}}) \bmod M}$	$G'_m = e^{\mathrm{i} \cdot m \cdot \frac{2\pi k_{\mathrm{s}}}{M}} \cdot G_m$
Direction of contour traversal	$g'_k = g_{-k \bmod M}$	$G'_m = G_{-m \bmod M}$
Reflection about the x-axis	$g'_k = g^*_k$	$G'_m = G^*_{-m \bmod M}$

where G^* denotes the complex conjugate of the original DFT coefficients. Reflections about arbitrary axes can be described in the same way with additional rotations. Fourier descriptors can be made invariant against reflections, such that symmetric contours map to equivalent descriptors [245]. Note, however, that invariance to symmetry is not always desirable, for example, for distinguishing the silhouettes of left and right hands.

The relations between 2D point coordinates and the Fourier spectrum, as well as the effects of the aforementioned geometric shape transformations upon the DFT coefficients are compactly summarized in Table 26.1.

26.5 Transformation-Invariant Fourier Descriptors

As mentioned already, making a Fourier descriptor invariant to *translation* or absolute shape position is easy because the only affected spectral coefficient is G_0. Thus, setting coefficient G_0 to zero implicitly moves the center of the corresponding shape to the coordinate origin and thus creates a descriptor that is invariant to shape translation.

Invariance against a change in *scale* is also a simple issue because it only multiplies the magnitude of all Fourier coefficients by the same real-valued scale factor, which can be easily normalized.

A more challenging task is to make Fourier descriptors invariant against shape *rotation* and shift of the contour *starting point*, because they jointly affect the phase of the Fourier coefficients. If matching is to be based on the complex-valued Fourier descriptors (not on coefficient magnitude only) to achieve better shape discrimination, the phase changes introduced by shape rotation and start point shifts must be eliminated first. However, due to noise and possible ambiguities, this is not a trivial problem (see also [183, 184, 189, 245]).

26.5.1 Scale Invariance

As mentioned in Section 26.4.2, the magnitude G_{+1} is often used as a reference to normalize for scale, since G_{+1} is typically (though not always) the Fourier coefficient with the largest magnitude. Alternatively, one could use the size of the fundamental ellipse, defined by the Fourier descriptor pair FP_1, to measure the overall scale, for example, by normalizing to

$$G_m^{\mathrm{S}} \leftarrow \frac{1}{|G_{-1}| + |G_{+1}|} \cdot G_m, \tag{26.96}$$

which normalizes the *length* of the major axis $a_1 = |G_{-1}|+|G_{+1}|$ (see Eqn. (26.57)) of the fundamental ellipse to unity. Another alternative is

$$G_m^{\mathrm{S}} \leftarrow \frac{1}{(|G_{-1}| \cdot |G_{+1}|)^{1/2}} \cdot G_m, \tag{26.97}$$

which normalizes the *area* of the fundamental ellipse. Since all variants in Eqns. (26.83), (26.96) and (26.97) scale the coefficients G_m by a fixed (real-valued) factor, the shape information contained in the Fourier descriptor remains unchanged.

There are shapes, however, where coefficients G_{+1} and/or G_{-1} are small or almost vanish to zero, such that they are not always a reliable reference for scale. An obvious solution is to include the complete set of Fourier coefficients by standardizing the *norm* of the coefficient vector $\boldsymbol{G}$ to unity in the form

$$G_m^{\mathrm{S}} \leftarrow \frac{1}{\|\boldsymbol{G}\|} \cdot G_m, \tag{26.98}$$

(assuming that $G_0 = 0$). In general, the L_2 norm of a complex-valued vector $Z = (z_0, z_1, \ldots, z_{M-1})$, $z_i \in \mathbb{C}$, is defined as

$$\|Z\| = \Bigl(\sum_{i=1}^{M-1} |z_i|^2\Bigr)^{1/2} = \Bigl(\sum_{i=1}^{M-1} \mathrm{Re}(z_i)^2 + \mathrm{Im}(z_i)^2\Bigr)^{1/2}. \tag{26.99}$$

Scaling the vector Z by the reciprocal of its norm yields a vector with unit norm, that is,

$$\Bigl\| \frac{1}{\|Z\|} \cdot Z \Bigr\| = 1. \tag{26.100}$$

To normalize a given Fourier descriptor $\boldsymbol{G}$, we use all elements except G_0 (which relates to the absolute position of the shape and is not relevant for its shape). The following substitution makes $\boldsymbol{G}$ scale invariant by normalizing the remaining sub-vector $(G_1, G_2, \ldots, G_{M-1})$ to

$$G_m^{\mathrm{S}} \leftarrow \begin{cases} G_m & \text{for } m = 0, \\ \frac{1}{\sqrt{\nu}} \cdot G_m & \text{for } 1 \leq m < M, \end{cases} \qquad \text{with } \nu = \sum_{m=1}^{M-1} |G_m|^2. \tag{26.101}$$

See procedure MakeScaleInvariant($\boldsymbol{G}$) in Alg. 26.6 (lines 7–15) for a summary of this step.

26.5.2 Start Point Invariance

As discussed in Sections 26.4.3 and 26.4.4, respectively, shape rotation and shift of start point both affect the phase of the Fourier coefficients in a combined manner, without altering their magnitude. In particular, if the shape is rotated by some angle β (see Eqn. (26.89)) and the start position is shifted by k_s samples (see Eqn. (26.86)), then each Fourier coefficient G_m is modified to

$$G'_m = e^{\mathrm{i}\cdot\beta} \cdot e^{\mathrm{i}\cdot m\cdot\varphi_{\mathrm{s}}} \cdot G_m = e^{\mathrm{i}\cdot(\beta + m\cdot\varphi_{\mathrm{s}})} \cdot G_m, \tag{26.102}$$

where $\varphi_{\mathrm{s}} = 2\pi k_{\mathrm{s}}/M$ is the corresponding *start point phase*. Thus, the incurred phase shift is not only different for each coefficient but simultaneously depends on the rotation angle β and the start point phase φ_{s}. Normalization in this case means to remove these phase shifts, which would be straightforward if β and φ_{s} were known. We derive these two parameters one after the other, starting with the calculation of the start point phase φ_{s}, which we describe in this section, followed by the estimation of the rotation β, shown subsequently in Section 26.5.3.

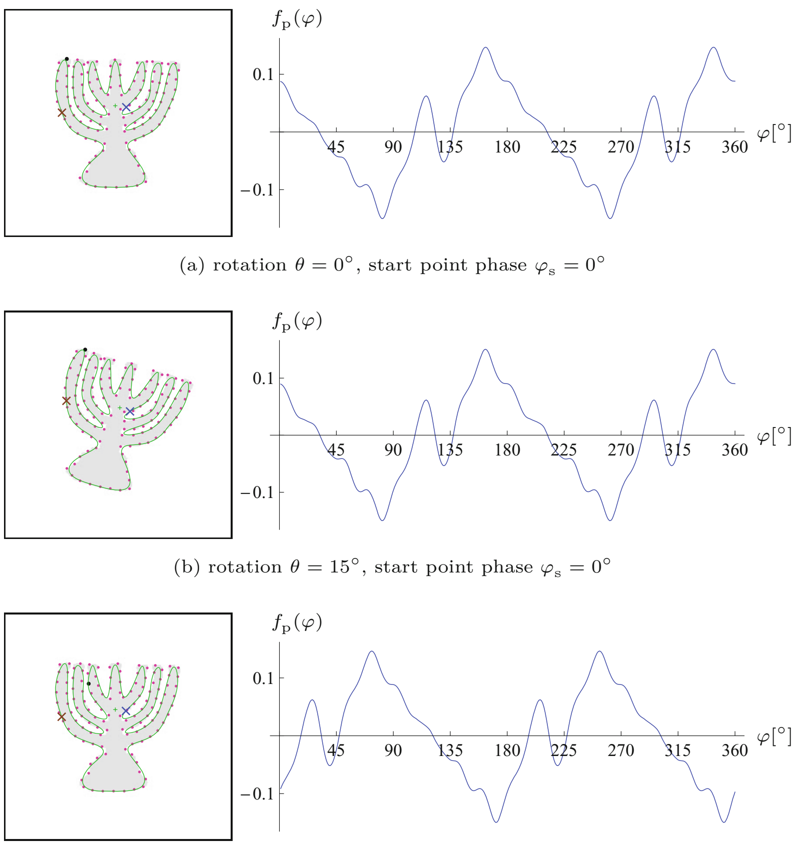
To normalize the Fourier descriptor of a particular shape to a "canonical" start point, we need a quantity that can be calculated from the Fourier spectrum and only depends on the start point phase φ_{s} but is independent of the rotation β. From Eqn. (26.90) and Fig. 26.15 we see that the phase *difference* within any Fourier descriptor pair (G_{-m}, G_{+m}) is proportional to the start point phase φ_{s} and independent to shape rotation β, since the latter rotates all coefficients by the same angle. Thus, we look for a quantity that depends only on the phase *differences* within Fourier descriptor pairs. This is accomplished, for example, by the function

$$f_{\mathrm{p}}(\varphi) = \sum_{m=1}^{M_{\mathrm{p}}} \left[e^{-\mathrm{i}\cdot m\cdot\varphi} \cdot G_{-m}\right] \otimes \left[e^{\mathrm{i}\cdot m\cdot\varphi} \cdot G_m\right], \tag{26.103}$$

where parameter φ is an arbitrary start point phase, M_{p} is the number of coefficient pairs, and $\otimes$ denotes the "cross product" between two Fourier coefficients.[18] Given a particular start point phase φ, the function in Eqn. (26.103) yields the sum of the cross products of each coefficient pair (G_{-m}, G_m), for $m = 1, \ldots, M_{\mathrm{p}}$. If each of the complex-valued coefficients is interpreted as a vector in the 2D plane, the magnitude of their cross product is proportional to the *area* of the enclosed parallelogram. The enclosed area is potentially large only if *both* vectors are of significant length, which means that the corresponding ellipse has a distinct eccentricity and orientation. Note that the sign of the cross product may be positive or negative and depends on the relative orientation or "handedness" of the two vectors.

Since the function $f_{\mathrm{p}}(\varphi)$ is based only on the *relative* orientation (phase) of the involved coefficients, it is invariant to a shape rotation

[18] In analogy to 2D vector notation, we define the "cross product" of two complex quantities $z_1 = (a_1, b_1)$ and $z_2 = (a_2, b_2)$ as $z_1 \otimes z_2 = a_1 \cdot b_2 - b_1 \cdot a_2 = |z_1| \cdot |z_2| \cdot \sin(\theta_2 - \theta_1)$. See also Sec. B.3.3 in the Appendix.



(a) rotation $\theta = 0°$, start point phase $\varphi_s = 0°$

(b) rotation $\theta = 15°$, start point phase $\varphi_s = 0°$

(c) rotation $\theta = 0°$, start point phase $\varphi_s = 90°$

Table 26.2
Plot of the function $f_p(\varphi)$ used for start point normalization. In the figures on the left, the real start point is marked by a black dot. The normalized start points φ_A and $\varphi_B = \varphi_A + \pi$ are marked by a blue and a brown cross, respectively. They correspond to the two peak positions of the function $f_p(\varphi)$, as defined in Eqn. (26.103), separated by a fixed phase shift of $\pi = 180°$ (right). The function is invariant under shape rotation, as demonstrated in (b), where the shape is rotated by 15° but sampled from the same start point as in (a). However, the phase of $f_p(\varphi)$ is proportional to the start point shift, as shown in (c), where the start point is chosen at 25% ($\varphi_s = 90°$) of the boundary path length. The functions were calculated after scale normalization, using $M_p = 25$ Fourier coefficient pairs.

β, which shifts all coefficients by the same angle (see Eqn. (26.86)). As shown in Fig. 26.2, $f_p(\varphi)$ is periodic with π and its phase is proportional to the actual start point shift. We choose the angle φ that *maximizes* $f_p(\varphi)$ as the "canonical" start point phase φ_A, that is,

$$\varphi_A = \operatorname*{argmax}_{0 \leq \varphi < \pi} f_p(\varphi). \tag{26.104}$$

However, since $f_p(\varphi) = f_p(\varphi + \pi)$, there is also a *second* candidate phase

$$\varphi_B = \varphi_A + \pi, \tag{26.105}$$

displaced by $\pi = 180°$. The two "canonical" start points corresponding to φ_A and φ_B, respectively, are marked on the reconstructed shapes in Fig. 26.2. Although it might seem easy at first to resolve this 180° ambiguity of the start point phase, this turns out to be difficult to achieve in general from the Fourier coefficients alone. Several functions have been proposed for this purpose that work well for certain shapes but fail on others, including the "positive real energy" function suggested in [245]. In particular, any decision based on the magnitude or phase of a *single* coefficient (or a single coefficient pair) must eventually fail, since none of the coefficients is guaranteed to have a significant magnitude. With vanishing coefficient magnitude, phase measurements become unreliable and may be very susceptible to noise.

The complete process of start point normalization is summarized in Alg. 26.7. The start point phase φ_{A} is found numerically by evaluating the function $f_{\mathrm{p}}(\varphi)$ at 400 discrete steps for $\varphi = 0, \ldots, \pi$ (lines 6–16). For practical use, this exhaustive method should be substituted by a more efficient and accurate optimization technique (for example, using Brent's method [190, Ch. 10]).[19] Given the estimated start point phase φ_{A} for the Fourier descriptor $\boldsymbol{G}$, *two* normalized versions $\boldsymbol{G}^{\mathrm{A}}, \boldsymbol{G}^{\mathrm{B}}$ are calculated as

$$\begin{aligned} \boldsymbol{G}^{\mathrm{A}}:\ & G^{\mathrm{A}}_m \leftarrow G_m \cdot e^{\mathrm{i}\cdot m\cdot\varphi_{\mathrm{A}}}, \\ \boldsymbol{G}^{\mathrm{B}}:\ & G^{\mathrm{B}}_m \leftarrow G_m \cdot e^{\mathrm{i}\cdot m\cdot(\varphi_{\mathrm{A}}+\pi)}, \end{aligned} \tag{26.106}$$

for $m = -M_{\mathrm{p}}, \ldots, M_{\mathrm{p}}, m \neq 0$. Note that start point normalization does not require the Fourier descriptor $\boldsymbol{G}$ to be normalized for translation and scale (see Sec. 26.5.1).

26.5.3 Rotation Invariance

After normalizing for starting point, the orientation of the fundamental ellipse (formed by the descriptor pair (G_{-1}, G_{+1})) could be assumed to be a reliable reference for global shape rotation. However, for certain shapes (e.g., regular polyhedra with an even number of faces), G_{-1} may vanish. Therefore, we recover the overall shape orientation from the vector obtained as the weighted sum of *all* Fourier coefficients, that is,

$$z = \sum_{m=1}^{M_{\mathrm{p}}} \frac{1}{m} \cdot (G_{-m} + G_{+m}), \tag{26.107}$$

where the $1/m$ serves as a weighting factor, giving stronger emphasis to the low-frequency coefficients and attenuating the influence of the high-frequency coefficients. The resulting shape orientation estimate is

$$\beta = \sphericalangle z = \tan^{-1}\left(\frac{\mathrm{Im}(z)}{\mathrm{Re}(z)}\right). \tag{26.108}$$

To normalize $\boldsymbol{G}^{\mathrm{A}}, \boldsymbol{G}^{\mathrm{B}}$ (obtained in Eqn. (26.106)) for shape orientation, we rotate each coefficient (except G_0) by $-\beta$, that is,

$$\begin{aligned} \boldsymbol{G}^{\mathrm{A}}:\ & G^{\mathrm{A}}_m \leftarrow G^{\mathrm{A}}_m \cdot e^{-\mathrm{i}\cdot\beta}, \\ \boldsymbol{G}^{\mathrm{B}}:\ & G^{\mathrm{B}}_m \leftarrow G^{\mathrm{B}}_m \cdot e^{-\mathrm{i}\cdot\beta}, \end{aligned} \tag{26.109}$$

for $m = -M_{\mathrm{p}}, \ldots, M_{\mathrm{p}}, m \neq 0$. For a summary of these steps, see procedure MakeRotationInvariant($\boldsymbol{G}$) in Alg. 26.6 (lines 16–24).

[19] The accompanying Java implementation uses the class `BrentOptimizer` from the *Apache Commons Math library* [4] for this purpose.

```
MakeInvariant($\boldsymbol{G}$)
    Input: $\boldsymbol{G}$, Fourier descriptor with $M_\mathrm{p}$ coefficient pairs.
    Returns a pair of normalized Fourier descriptors $\boldsymbol{G}^\mathrm{A}$, $\boldsymbol{G}^\mathrm{B}$, with
    a start point phase offset by 180°.

    MakeScaleInvariant($\boldsymbol{G}$)                                  ▷ see below
    $(\boldsymbol{G}^\mathrm{A}, \boldsymbol{G}^\mathrm{B}) \leftarrow$ MakeStartPointInvariant($\boldsymbol{G}$)          ▷ see Alg. 26.7
    MakeRotationInvariant($\boldsymbol{G}^\mathrm{A}$)                             ▷ see below
    MakeRotationInvariant($\boldsymbol{G}^\mathrm{B}$)
    return $(\boldsymbol{G}^\mathrm{A}, \boldsymbol{G}^\mathrm{B})$.

MakeScaleInvariant($\boldsymbol{G}$)
    Modifies $\boldsymbol{G}$ by unifying its norm and returns the scale factor $\nu$.
    $s \leftarrow 0$                                                ▷ $s \in \mathbb{R}$
    for $m \leftarrow 1, \ldots, M_\mathrm{p}$ do
        $s \leftarrow s + |\boldsymbol{G}(-m)|^2 + |\boldsymbol{G}(m)|^2$
    $\nu \leftarrow 1/\sqrt{s}$
    for $m \leftarrow 1, \ldots, M_\mathrm{p}$ do
        $\boldsymbol{G}(-m) \leftarrow \nu \cdot \boldsymbol{G}(-m)$
        $\boldsymbol{G}(m) \;\;\leftarrow \nu \cdot \boldsymbol{G}(m)$
    return $\nu$.

MakeRotationInvariant($\boldsymbol{G}$)
    Modifies $\boldsymbol{G}$ and returns the estimated rotation angle $\beta$.
    $z \leftarrow 0 + \mathrm{i} \cdot 0$                                          ▷ $z \in \mathbb{C}$
    for $m \leftarrow 1, \ldots, M_\mathrm{p}$ do
        $z \leftarrow z + \frac{1}{m} \cdot (\boldsymbol{G}(-m) + \boldsymbol{G}(m))$          ▷ complex addition!
    $\beta \leftarrow \measuredangle z$
    for $m \leftarrow 1, \ldots, M_\mathrm{p}$ do               ▷ rotate all coefficients by $-\beta$
        $\boldsymbol{G}(-m) \leftarrow e^{-\mathrm{i} \cdot \beta} \cdot \boldsymbol{G}(-m)$
        $\boldsymbol{G}(m) \;\;\leftarrow e^{-\mathrm{i} \cdot \beta} \cdot \boldsymbol{G}(m)$
    return $\beta$.
```

Alg. 26.6
Making Fourier descriptors invariant against scale, shift of start point, and shape rotation. For a given Fourier descriptor $\boldsymbol{G}$, procedure MakeStartPointInvariant($\boldsymbol{G}$) returns a pair of normalized Fourier descriptors $(\boldsymbol{G}^\mathrm{A}, \boldsymbol{G}^\mathrm{B})$, one for each normalized start point phase φ_A and $\varphi_\mathrm{B} = \varphi_\mathrm{A} + \pi$.

26.5.4 Other Approaches

The aforementioned normalization for making Fourier descriptors invariant to geometric transformations deviates from the published "classic" techniques in certain ways, but also adopts some common elements. As representative examples, we briefly discuss two of these techniques (already referenced earlier) in the following.

Persoon and Fu [183,184] proposed (in what they call the "suboptimal" approach) to choose the parameters s (common scale factor), β (shape rotation), and φ_s (start point phase) such that the modified coefficients G'_{-1}, G'_{+1} are both imaginary and $|G_{-1} + G_{+1}| = 1$. As argued in [245], this method leaves a ±180° ambiguity for the shape orientation. Also, it requires that both G_{-1}, G_{+1} have significant magnitude, which may not be true for G_{-1} in case of shapes that are circularly symmetric (e.g., equilateral triangles, squares, pentagons etc.).

Wallace and Wintz [245] use $|G_{+1}|$ as the common scale factor, because the coefficient G_{+1} typically has the largest magnitude. The phase of G_{+1}, denoted $\phi_1 = \measuredangle G_{+1}$, and the phase of another coefficient G_k ($k > 0$) with the second-largest magnitude and phase $\phi_k = \measuredangle G_k$ are used to compensate for rotation and starting point. Coefficients are phase shifted such that both G'_{+1} and G'_k have zero

Alg. 26.7
Making Fourier descriptors invariant to the shift of start point. Since the result is ambiguous by 180°, two normalized descriptors $(\boldsymbol{G}^{\mathrm{A}}, \boldsymbol{G}^{\mathrm{B}})$ are returned, with the start point phase set to φ_{A} and $\varphi_{\mathrm{A}} + \pi$, respectively.

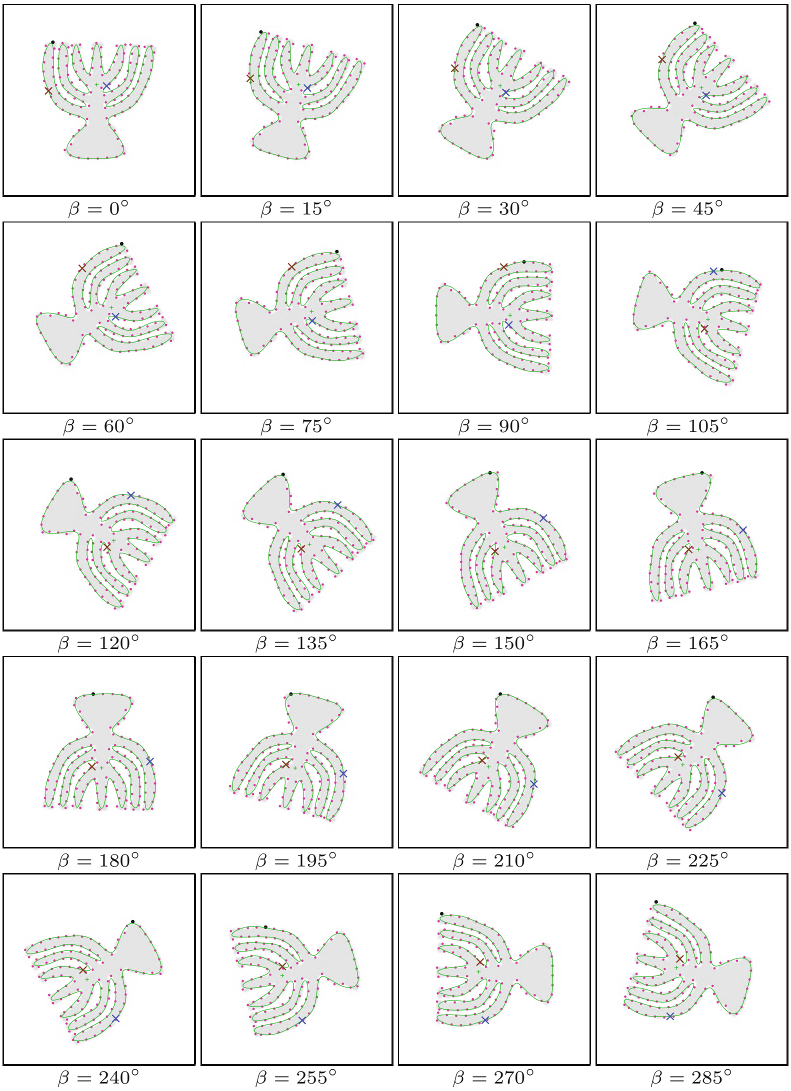
```
MakeStartPointInvariant($\boldsymbol{G}$)
    Input: $\boldsymbol{G}$, Fourier descriptor with $M_{\mathrm{p}}$ coefficient pairs.
    Returns a pair of new Fourier descriptors $\boldsymbol{G}^{\mathrm{A}}$, $\boldsymbol{G}^{\mathrm{B}}$, normalized
    to the start point phase $\varphi_{\mathrm{A}}$ and $\varphi_{\mathrm{A}} + \pi$, respectively.
    $\varphi_{\mathrm{A}} \leftarrow$ GetStartPointPhase($\boldsymbol{G}$)                ▷ see below
    $\boldsymbol{G}^{\mathrm{A}} \leftarrow$ ShiftStartPointPhase($\boldsymbol{G}, \varphi_{\mathrm{A}}$)          ▷ see below
    $\boldsymbol{G}^{\mathrm{B}} \leftarrow$ ShiftStartPointPhase($\boldsymbol{G}, \varphi_{\mathrm{A}} + \pi$)
    return $(\boldsymbol{G}^{\mathrm{A}}, \boldsymbol{G}^{\mathrm{B}})$.

GetStartPointPhase($\boldsymbol{G}$)
    Returns $\varphi$ maximizing $f_{\mathrm{p}}(\boldsymbol{G}, \varphi)$, with $\varphi \in [0, \pi)$. The maximum
    is found by simple brute-force search (for illustration only).
    $c_{\max} \leftarrow -\infty$
    $\varphi_{\max} \leftarrow 0$
    $K \leftarrow 400$                               ▷ do $K$ search steps over $0, \ldots, \pi$
    for $k \leftarrow 0, \ldots, K-1$ do                    ▷ find $\varphi$ maximizing $f_{\mathrm{p}}(\boldsymbol{G}, \varphi)$
        $\varphi \leftarrow \pi \cdot \frac{k}{K}$
        $c \leftarrow f_{\mathrm{p}}(\boldsymbol{G}, \varphi)$
        if $c > c_{\max}$ then
            $c_{\max} \leftarrow c$
            $\varphi_{\max} \leftarrow \varphi$
    return $\varphi_{\max}$.

$f_{\mathrm{p}}(\boldsymbol{G}, \varphi)$                                       ▷ see Eq. 26.103
    $s \leftarrow 0$
    for $m \leftarrow 1, \ldots, M_{\mathrm{p}}$ do
        $z_1 \leftarrow \boldsymbol{G}(-m) \cdot e^{-\mathrm{i} \cdot m \cdot \varphi}$
        $z_2 \leftarrow \boldsymbol{G}(m) \cdot e^{\mathrm{i} \cdot m \cdot \varphi}$
        $s \leftarrow s + \mathrm{Re}(z_1) \cdot \mathrm{Im}(z_2) - \mathrm{Im}(z_1) \cdot \mathrm{Re}(z_2)$   ▷ $= s + (z_1 \otimes z_2)$
    return $s$.

ShiftStartPointPhase($\boldsymbol{G}, \varphi$)                ▷ start-point normalize $\boldsymbol{G}$ by $\varphi$
    $\boldsymbol{G}' \leftarrow$ Duplicate($\boldsymbol{G}$)
    for $m \leftarrow 1, \ldots, M_{\mathrm{p}}$ do
        $\boldsymbol{G}'(-m) \leftarrow \boldsymbol{G}(-m) \cdot e^{-\mathrm{i} \cdot m \cdot \varphi}$
        $\boldsymbol{G}'(m) \leftarrow \boldsymbol{G}(m) \cdot e^{\mathrm{i} \cdot m \cdot \varphi}$
    return $\boldsymbol{G}'$.
```

phase. This is accomplished by multiplying all coefficients in the form

$$G'_m = G_m \cdot e^{\mathrm{i} \cdot [(m-k) \cdot \phi_1 + (1-m) \cdot \phi_k] \cdot (k-1)}, \tag{26.110}$$

for $-\frac{M}{2} + 1 \leq m \leq \frac{M}{2}$ (also used in [189]). Depending on the index k of the second-largest coefficient, there exist $|k-1|$ different orientation/start point combinations to obtain zero-phase in G'_{+1} and G'_k. If $k = 2$, then $|k-1| = 1$, thus the solution is unique and Eqn. (26.110) simplifies to

$$G'_m = G_m \cdot e^{\mathrm{i} \cdot [(m-2) \cdot \phi_1 + (1-m) \cdot \phi_2]}, \tag{26.111}$$



$\beta = 0°$ $\beta = 15°$ $\beta = 30°$ $\beta = 45°$

$\beta = 60°$ $\beta = 75°$ $\beta = 90°$ $\beta = 105°$

$\beta = 120°$ $\beta = 135°$ $\beta = 150°$ $\beta = 165°$

$\beta = 180°$ $\beta = 195°$ $\beta = 210°$ $\beta = 225°$

$\beta = 240°$ $\beta = 255°$ $\beta = 270°$ $\beta = 285°$

Fig. 26.17
Start point normalization under varying shape rotation (β). The real start point (which varies with shape rotation) is marked by a black dot. The two normalized start points φ_{A} and $\varphi_{\mathrm{B}} = \varphi_{\mathrm{A}} + \pi$ (calculated with the procedure in Alg. 26.7) are marked by a blue and a brown ×, respectively. Twenty-five Fourier coefficient pairs are used for the normalization and shape reconstruction. Inaccuracies are due to shape variations caused by the use of nearest-neighbor interpolation for the image rotation.

with $\phi_2 = \sphericalangle G_2$.[20] Otherwise, the ambiguity is resolved by calculating an “ambiguity-resolving” criterion for each of the $|k-1|$ solutions, for example, the amount of “positive real energy”,

$$\sum_{m=1}^{N-1} \operatorname{Re}(G'_m) \cdot |\operatorname{Re}(G'_m)|\,,$$

as defined in [245] (other functions were suggested in [189]). This leaves the problem that, for matching, the normalization of the investigated shape descriptor must be based on the same set of dominant coefficients as the reference descriptor. Alternatively, one could memorize the relevant coefficient indexes for every reference descrip-

[20] Unfortunately, the general use of coefficient G_2 as a phase reference is critical, because the magnitude of G_2 may be small or even zero for certain symmetrical shapes (including all regular polygons with an even number of faces).

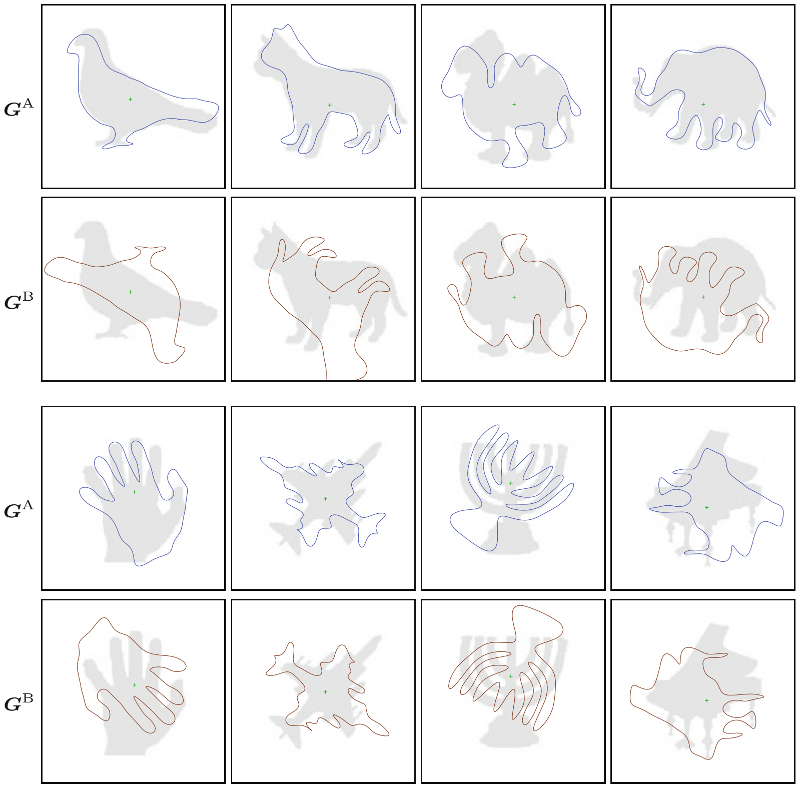


Fig. 26.18
Reconstruction of various shapes from Fourier descriptors normalized for start point shift and shape rotation. The blue shapes (rows $1, 3$) correspond to the normalized Fourier descriptors G^A with start point phase φ_A. The brown shapes (rows $2, 4$) correspond to the normalized Fourier descriptors G^B with start point phase $\varphi_B = \varphi_A + \pi$. No scale normalization was applied for better visualization.

tor, but then different normalizations must be applied for matching against multiple models in a database.

26.6 Shape Matching with Fourier Descriptors

A typical use of Fourier descriptors is to see if a given shape is identical or similar to an exemplar contained in a database of reference shapes. For this purpose, we need to define a distance measure that quantifies the difference between two Fourier shape descriptors $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$. In the following, we assume that the Fourier descriptors $\boldsymbol{G}_1, \boldsymbol{G}_2$ are at least scale-normalized (as described in Alg. 26.6) and of identical length, each with M_p coefficient pairs.

26.6.1 Magnitude-Only Matching

In the simplest case, we only use the *magnitude* of the Fourier coefficients for comparison and entirely ignore their phase, using the distance function

$$\operatorname{dist}_M(\boldsymbol{G}_1, \boldsymbol{G}_2) = \Bigl[\sum_{\substack{m=-M_p,\\ m\neq 0}}^{M_p} \bigl(|\boldsymbol{G}_1(m)| - |\boldsymbol{G}_2(m)| \bigr)^2 \Bigr]^{1/2} \tag{26.112}$$

$$= \Bigl[\sum_{m=1}^{M_p} \bigl(|\boldsymbol{G}_1(-m)| - |\boldsymbol{G}_2(-m)| \bigr)^2 + \bigl(|\boldsymbol{G}_1(m)| - |\boldsymbol{G}_2(m)| \bigr)^2 \Bigr]^{1/2},$$

where M_{p} denotes the number of FD pairs used for matching. Note that Eqn. (26.112) is simply the L_2 norm of the magnitude difference vector, and of course other norms (such as L_1 or L_∞) could be used as well. The advantage of the magnitude-only approach is that no normalization (except for scale) is required. Its drawback is that even highly dissimilar shapes might be mistakenly matched, since the removal of phase naturally eliminates shape information that is possibly essential for discrimination. As demonstrated in Fig. 26.19, a given Fourier magnitude vector may correspond to a great diversity of shapes, and thus the subspace of "equivalent" shapes defined by the magnitude-only distance $\mathrm{dist}_{\mathrm{M}}$ is quite large.

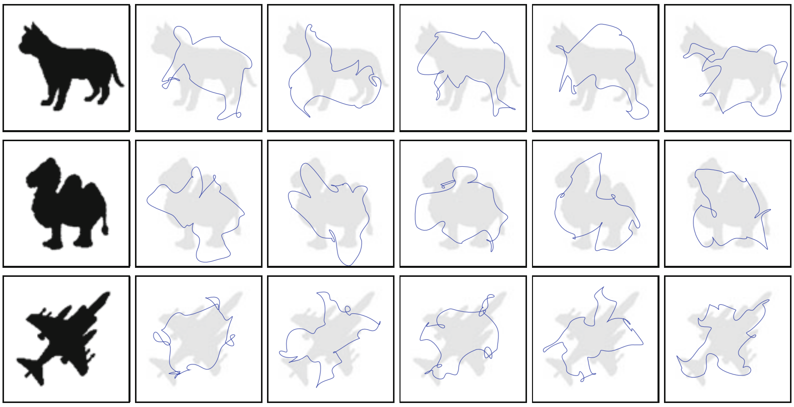


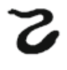
Fig. 26.19
Magnitude-only reconstruction (randomized phase). Reconstruction of shapes from Fourier descriptors with the phase of all coefficients (except G_{-1}, G_0, and G_{+1}) individually randomized. Note that the magnitude of the coefficients is exactly the same for each shape category, so all blue shapes would be considered "equivalent" to the original shape (first column) by a magnitude-only matcher.

Nevertheless, magnitude-only matching may be sufficient in situations where the reference shapes are not too similar. In a sense, the operation of reducing the complex-valued Fourier descriptors to their magnitude vectors can be viewed as a *hash* function. While potentially many different shapes may produce (i.e., "hash to") similar Fourier magnitude vectors, the chance of two real shapes mapping to the same vector (and thus being confused) may be relatively small. Thus, particularly considering its simplicity (only scale-normalization of descriptors is required), magnitude-based matching can be quite effective in practice.

Figure 26.20 shows the pair-wise magnitude-only distances (blue cells, values are $10 \times \mathrm{dist}_{\mathrm{M}}$) between various sample shapes. The corresponding intra-class distances, given in Fig. 26.21, are typically more than one order of magnitude smaller, indicating that shape discrimination based on this measure should be fairly reliable.

26.6.2 Complex (Phase-Preserving) Matching

Assuming that the Fourier descriptors $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$ have been normalized for scale, start point shift, and shape rotation (see Alg. 26.6), we can use the following function to measure their mutual distance:

	bird	cat	camel	elephant	hand	harrier	menora	piano	creature
	0.000	4.529	4.482	5.007	5.525	4.314	7.554	5.174	7.076
	3.156	0.000	5.788	4.708	5.711	5.701	7.181	5.543	7.677
	2.648	3.005	0.000	4.429	5.573	3.726	7.014	4.013	8.480
	3.487	1.933	2.549	0.000	6.100	4.618	5.338	4.369	8.743
	4.627	3.146	3.132	2.372	0.000	6.079	8.540	5.580	7.136
	3.712	3.707	2.687	3.553	4.294	0.000	6.818	4.958	8.284
	5.835	4.893	4.563	4.162	3.788	5.775	0.000	6.826	11.072
	4.037	2.426	2.610	1.876	1.848	3.405	4.315	0.000	7.666
	6.030	6.261	5.554	5.492	5.955	5.914	5.190	6.049	0.000

$\mathrm{dist}_{\mathrm{M}}(\boldsymbol{G}_1, \boldsymbol{G}_2)$

$\mathrm{dist}_{\mathrm{C}}(\boldsymbol{G}_1, \boldsymbol{G}_2)$

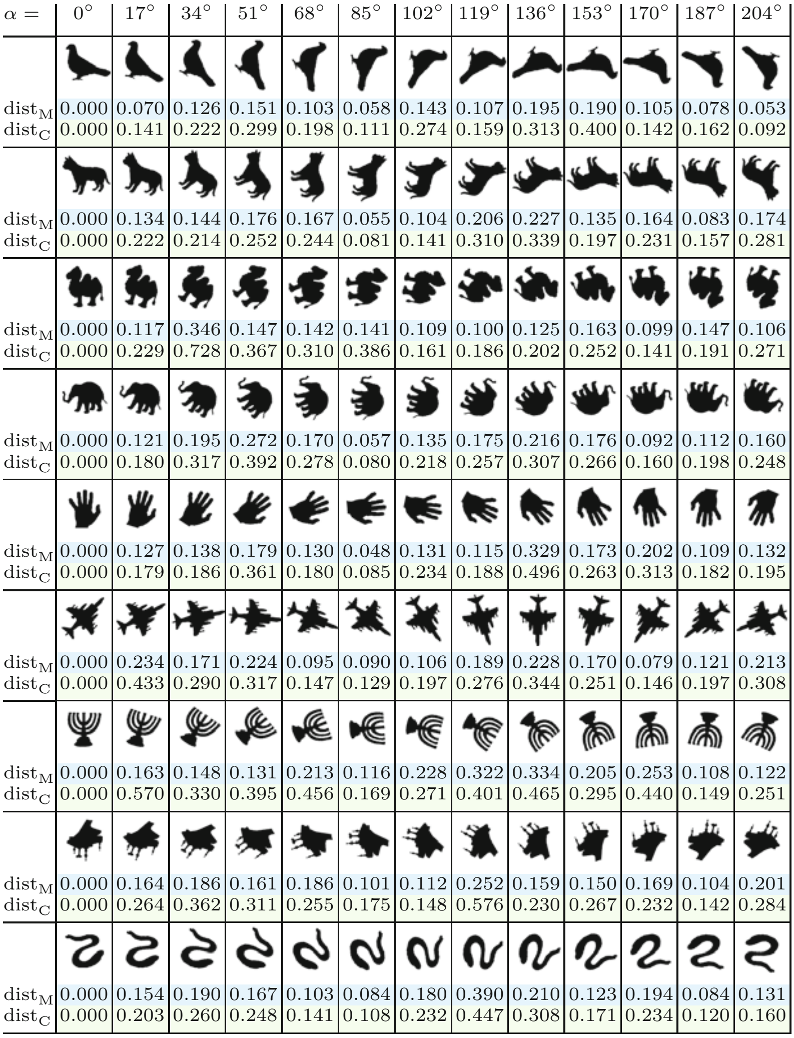
Fig. 26.20
Inter-class Fourier descriptor distances (magnitude-only and complex-valued). Numbers inside the green fields (lower-left half of the matrix) are the magnitude-only distances $\mathrm{dist}_{\mathrm{M}}$ (see Eqn. (26.112)). Numbers in blue fields (upper-right half of the matrix) are the complex-valued distances $\mathrm{dist}_{\mathrm{C}}$ (see Eqn. (26.114)). Shapes were sampled uniformly at 125 contour positions, with 25 coefficient pairs. Fourier descriptors were normalized for scale, start point and rotation. All distance values are multiplied by 10.

$$\mathrm{dist}_{\mathrm{C}}(\boldsymbol{G}_1, \boldsymbol{G}_2) = \Big(\sum_{\substack{m=-M_{\mathrm{p}},\\ m\neq 0}}^{M_{\mathrm{p}}} \big|\boldsymbol{G}_1(m) - \boldsymbol{G}_2(m)\big|^2 \Big)^{1/2} \tag{26.113}$$

$$= \Big(\sum_{m=1}^{M_{\mathrm{p}}} \big|\boldsymbol{G}_1(-m) - \boldsymbol{G}_2(-m)\big|^2 + \big|\boldsymbol{G}_1(m) - \boldsymbol{G}_2(m)\big|^2 \Big)^{1/2} \tag{26.114}$$

$$= \Big(\sum_{\substack{m=-M_{\mathrm{p}},\\ m\neq 0}}^{M_{\mathrm{p}}} \big[\mathrm{Re}(\boldsymbol{G}_1(m)) - \mathrm{Re}(\boldsymbol{G}_2(m))\big]^2 + \big[\mathrm{Im}(\boldsymbol{G}_1(m)) - \mathrm{Im}(\boldsymbol{G}_2(m))\big]^2 \Big)^{1/2}. \tag{26.115}$$

Again, this is simply the L_2 norm of the complex-valued difference vector $\boldsymbol{G}_1 - \boldsymbol{G}_2$ (ignoring the coefficients at $m = 0$), which could be substituted by some other norm. Since the phase of the involved coefficients is fully preserved, a zero distance between two Fourier descriptors means that they represent the very same shape. Thus the set of equivalent shapes defined by the distance function in Eqn. (26.114) is much smaller than the one defined by the magnitude-only distance in Eqn. (26.112). Consequently, the probability of two different shapes being confused for the same is also significantly smaller with this distance measure.



$\alpha =$	0°	17°	34°	51°	68°	85°	102°	119°	136°	153°	170°	187°	204°
dist_M	0.000	0.070	0.126	0.151	0.103	0.058	0.143	0.107	0.195	0.190	0.105	0.078	0.053
dist_C	0.000	0.141	0.222	0.299	0.198	0.111	0.274	0.159	0.313	0.400	0.142	0.162	0.092
dist_M	0.000	0.134	0.144	0.176	0.167	0.055	0.104	0.206	0.227	0.135	0.164	0.083	0.174
dist_C	0.000	0.222	0.214	0.252	0.244	0.081	0.141	0.310	0.339	0.197	0.231	0.157	0.281
dist_M	0.000	0.117	0.346	0.147	0.142	0.141	0.109	0.100	0.125	0.163	0.099	0.147	0.106
dist_C	0.000	0.229	0.728	0.367	0.310	0.386	0.161	0.186	0.202	0.252	0.141	0.191	0.271
dist_M	0.000	0.121	0.195	0.272	0.170	0.057	0.135	0.175	0.216	0.176	0.092	0.112	0.160
dist_C	0.000	0.180	0.317	0.392	0.278	0.080	0.218	0.257	0.307	0.266	0.160	0.198	0.248
dist_M	0.000	0.127	0.138	0.179	0.130	0.048	0.131	0.115	0.329	0.173	0.202	0.109	0.132
dist_C	0.000	0.179	0.186	0.361	0.180	0.085	0.234	0.188	0.496	0.263	0.313	0.182	0.195
dist_M	0.000	0.234	0.171	0.224	0.095	0.090	0.106	0.189	0.228	0.170	0.079	0.121	0.213
dist_C	0.000	0.433	0.290	0.317	0.147	0.129	0.197	0.276	0.344	0.251	0.146	0.197	0.308
dist_M	0.000	0.163	0.148	0.131	0.213	0.116	0.228	0.322	0.334	0.205	0.253	0.108	0.122
dist_C	0.000	0.570	0.330	0.395	0.456	0.169	0.271	0.401	0.465	0.295	0.440	0.149	0.251
dist_M	0.000	0.164	0.186	0.161	0.186	0.101	0.112	0.252	0.159	0.150	0.169	0.104	0.201
dist_C	0.000	0.264	0.362	0.311	0.255	0.175	0.148	0.576	0.230	0.267	0.232	0.142	0.284
dist_M	0.000	0.154	0.190	0.167	0.103	0.084	0.180	0.390	0.210	0.123	0.194	0.084	0.131
dist_C	0.000	0.203	0.260	0.248	0.141	0.108	0.232	0.447	0.308	0.171	0.234	0.120	0.160

Fig. 26.21
Intra-class Fourier descriptor distances (magnitude-only and complex-valued). The reference images (0° column) were rotated by angle α (multiples of 17°), using no (i.e., nearest-neighbor) interpolation. Numbers inside the blue fields are the magnitude-only distances dist_M (see Eqn. (26.112)). Numbers inside the green fields are the complex-valued distances dist_C (see Eqn. (26.114)). Shapes were sampled uniformly at 125 contour positions, with 25 coefficient pairs. Fourier descriptors were normalized for scale, start point shift and shape rotation. All distance values are multiplied by 10. Note that all *intra*-class distances are roughly one order of magnitude smaller than the *inter*-class distances shown in Fig. 26.20.

Complex inter-class and intra-class distance values for the set of sample shapes are listed in Figs. 26.20 and 26.21. Notice that, with the normalization described in Alg. 26.6, the complex intra-class distance values in Fig. 26.21 (which should be as small as possible) are typically about twice as large as the corresponding magnitude-only distance values, but still an order of magnitude smaller than comparable inter-class values in Fig. 26.20, so reliable shape discrimination should be possible.

The price paid for the increased discriminative power is the extra work necessary for normalizing the Fourier descriptors for start point and shape rotation (in addition to scale), as described in Alg. 26.6. Note that this involves the comparison with *two* normalized descriptors to cope with the unresolved 180° ambiguity of the start point normalization (see Eqns. (26.104) and (26.105)). For example, assume we wish to compare two shapes V_1, V_2 with Fourier descriptors $\boldsymbol{G}_1, \boldsymbol{G}_2$, respectively. We first calculate the corresponding invariant descriptors (as described in Alg. 26.6),

$$\begin{aligned}(\boldsymbol{G}_1^A, \boldsymbol{G}_1^{\mathrm{B}}) &\leftarrow \mathsf{MakeInvariant}(\boldsymbol{G}_1)\ ,\\ (\boldsymbol{G}_2^A, \boldsymbol{G}_2^{\mathrm{B}}) &\leftarrow \mathsf{MakeInvariant}(\boldsymbol{G}_2)\ .\end{aligned} \tag{26.116}$$

Now we use Eqn. (26.114) to calculate the complex-valued distance as

$$d_{\min} = \min\bigl(\mathrm{dist}_{\mathrm{C}}(\boldsymbol{G}_1^{\mathrm{A}}, \boldsymbol{G}_2^{\mathrm{A}}), \mathrm{dist}_{\mathrm{C}}(\boldsymbol{G}_1^{\mathrm{A}}, \boldsymbol{G}_2^{\mathrm{B}})\bigr) \tag{26.117}$$

or, alternatively, as

$$d_{\min} = \min\bigl(\mathrm{dist}_{\mathrm{C}}(\boldsymbol{G}_1^{\mathrm{A}}, \boldsymbol{G}_2^{\mathrm{A}}), \mathrm{dist}_{\mathrm{C}}(\boldsymbol{G}_1^{\mathrm{B}}, \boldsymbol{G}_2^{\mathrm{A}})\bigr). \tag{26.118}$$

Note that, in any case, the resulting distance $d_{\min}$ will be small only if the two shapes V_1, V_2 are really similar. This also means that we only need to store *one* of the two normalized Fourier descriptors—for example, $\boldsymbol{G}_{\mathrm{ref}}^{\mathrm{A}}$—for each reference shape V_{ref} and then (following Eqn. (26.117)) compare it to *both* normalized descriptors $\boldsymbol{G}_{\mathrm{new}}^{\mathrm{A}}$ and $\boldsymbol{G}_{\mathrm{new}}^{\mathrm{B}}$ of any new shape V_{new}.[21]

To illustrate this idea, Alg. 26.8 shows the construction of a simple Fourier descriptor database from a set of reference shapes and its subsequent use for classifying unknown shapes. First, procedure $\mathsf{MakeFdDataBase}(\boldsymbol{V})$ returns a map D holding a normalized Fourier descriptor for each of the reference shapes given in $\boldsymbol{V}$. Matching a new shape V_{new} to the entries in the database D is accomplished by procedure $\mathsf{FindBestMatch}(V_{\mathrm{new}}, \mathsf{D}, d_{\max})$, which returns the index of the best-fitting shape in D, or nil if the distance of the closest match exceeds the predefined threshold $d_{\max}$. As common in this situation, we use *squared* distance values (i.e., $\mathrm{dist}_{\mathrm{C}}^2$) for matching in Alg. 26.8 (lines 15–18), thereby avoiding the square root operations in Eqns. (26.112) and (26.114).

26.7 Java Implementation

The algorithms described in this chapter have been implemented as part of the open `imagingbook` library,[22] which is available at the book's accompanying website. As usual, most Java methods are named and structured identically to the procedures defined in the various algorithms for easy identification.

`FourierDescriptor` (class)

This is the main class of this package; it holds all data structures and implements the functionality common to all Fourier descriptors, including methods for shape reconstruction, invariance, and matching, as will be described here.

[21] The justification for keeping only *one* of the two normalized descriptors $\boldsymbol{G}_{\mathrm{ref}}^{\mathrm{A}}$, $\boldsymbol{G}_{\mathrm{ref}}^{\mathrm{B}}$ of each reference shape V_{ref} is that if two candidate shapes V_1, V_2 are similar, then the normalization will produce pairs of Fourier descriptors $(\boldsymbol{G}_1^{\mathrm{A}}, \boldsymbol{G}_1^{\mathrm{B}})$ and $(\boldsymbol{G}_2^{\mathrm{A}}, \boldsymbol{G}_2^{\mathrm{B}})$ that are also similar but not necessarily in the same order. Therefore $\boldsymbol{G}_1^{\mathrm{A}}$ must only match with *either* $\boldsymbol{G}_2^{\mathrm{A}}$ *or* $\boldsymbol{G}_2^{\mathrm{B}}$ to detect the similarity of V_1 and V_2.

[22] Package `imagingbook.pub.fd`.

Alg. 26.8
Simple shape matching with a database of Fourier descriptors. $\mathsf{MakeFdDataBase}(\mathsf{V}_{\mathrm{ref}}, M')$ creates and returns a new database (map) R from a sequence of reference shapes $\mathsf{V}_{\mathrm{ref}}$. R can then be passed to $\mathsf{FindBestMatch}(V_{\mathrm{new}}, M', \mathsf{R}, d_{\max})$ for classifying a new shape $\mathsf{V}_{\mathrm{new}}$, where $d_{\max}$ is a predefined distance threshold.

```
MakeFdDataBase($\mathsf{V}_{\mathrm{ref}}, M'$)
    Input: $\mathsf{V}_{\mathrm{ref}} = (V_0, V_1, \ldots, V_{N_\mathrm{R}})$, a sequence of reference shapes;
    $M'$, the number of Fourier coefficients. Returns a sequence of
    model Fourier descriptors for the reference shapes in $\mathsf{V}_{\mathrm{ref}}$.
    $N_\mathrm{R} \leftarrow |\mathcal{V}_{\mathrm{ref}}|$
    $\mathsf{R} \leftarrow$ new map of Fourier descriptors over $[0, N_\mathrm{R}-1]$
    for $i \leftarrow 0, \ldots, N_\mathrm{R}-1$ do
        $\boldsymbol{G} \leftarrow \mathsf{FourierDescriptorUniform}(\mathsf{V}_{\mathrm{ref}}(i), M')$      ▷ Alg. 26.3
        $(\boldsymbol{G}^\mathrm{A}, \boldsymbol{G}^\mathrm{B}) \leftarrow \mathsf{MakeInvariant}(\boldsymbol{G})$      ▷ Alg. 26.6
        $\mathsf{R}(i) \leftarrow \boldsymbol{G}^\mathrm{A}$      ▷ store only one normalized descriptor ($\boldsymbol{G}^\mathrm{A}$)
    return R.

FindBestMatch($V_{\mathrm{new}}, M', \mathsf{R}, d_{\max}$)
    Input: $V_{\mathrm{new}}$, a new shape; $M'$, the number of Fourier coefficients;
    R, a sequence of reference Fourier descriptors; $d_{\max}$, maximum
    squared distance acceptable for a positive match. Returns the
    best-matching shape index $i_{\min}$ or nil if no acceptable match was
    found.
    $\boldsymbol{G}_{\mathrm{new}} \leftarrow \mathsf{FourierDescriptorUniform}(V_{\mathrm{new}}, M')$      ▷ Alg. 26.3
    $(\boldsymbol{G}^\mathrm{A}_{\mathrm{new}}, \boldsymbol{G}^\mathrm{B}_{\mathrm{new}}) \leftarrow \mathsf{MakeInvariant}(\boldsymbol{G}_{\mathrm{new}})$      ▷ Alg. 26.6
    $d_{\min} \leftarrow \infty$,      $i_{\min} \leftarrow -1$
    for $i \leftarrow 0, \ldots, |\mathsf{R}|-1$ do
        $\boldsymbol{G}^\mathrm{A}_{\mathrm{ref}} \leftarrow \mathsf{R}(i)$
        $d_2 \leftarrow \min\big(\mathsf{D2}(\boldsymbol{G}^\mathrm{A}_{\mathrm{new}}, \boldsymbol{G}^\mathrm{A}_{\mathrm{ref}}), \mathsf{D2}(\boldsymbol{G}^\mathrm{B}_{\mathrm{new}}, \boldsymbol{G}^\mathrm{A}_{\mathrm{ref}})\big)$      ▷ Eq. 26.118
        if $d_2 < d_{\min}$ then
            $d_{\min} \leftarrow d_2$
            $i_{\min} \leftarrow i$
    if $d_{\min} \leq d_{\max}$ then
        return $i_{\min}$      ▷ best match index is $i_{\min}$
    else
        return nil.      ▷ no matching shape found in R

D2($\boldsymbol{G}_1, \boldsymbol{G}_2$)
    Returns the squared complex distance $\mathrm{dist}^2_\mathrm{C}(\boldsymbol{G}_1, \boldsymbol{G}_2)$ between the
    Fourier descriptors $\boldsymbol{G}_1$,$\boldsymbol{G}_2$ (see Eq. 26.114).
    $d \leftarrow 0$,   $M_\mathrm{p} \leftarrow (\min(|\boldsymbol{G}_1|, |\boldsymbol{G}_2|) - 1) \div 2$
    for $m \leftarrow -M_\mathrm{p}, \ldots, M_\mathrm{p}, m \neq 0$ do
        $d \leftarrow d + [\mathrm{Re}(\boldsymbol{G}_1(m)) - \mathrm{Re}(\boldsymbol{G}_2(m))]^2 +$
                  $[\mathrm{Im}(\boldsymbol{G}_1(m)) - \mathrm{Im}(\boldsymbol{G}_2(m))]^2$
    return $d$.      ▷ $d \equiv (\mathrm{dist}_\mathrm{C}(\boldsymbol{G}_1, \boldsymbol{G}_2))^2$
```

Class `FourierDescriptor` is abstract and thus cannot be instantiated. To create Fourier descriptor objects, one of the concrete subclasses `FourierDescriptorUniform` or `FourierDescriptorFromPolygon` (discussed later in this section) may be used, which provide the appropriate constructors. `FourierDescriptor` provides the following methods for both types of Fourier descriptors.

Access to Fourier coefficients

`Complex[] getCoefficients ()`
Returns the complete vector of complex-valued Fourier coefficients.[23]

[23] The class `Complex` is defined in package `imagingbook.lib.math`.

`Complex getCoefficient (int m)`
Returns the value of the Fourier coefficient $\boldsymbol{G}(\mathtt{m} \bmod M)$, with $M = |\boldsymbol{G}|$ as above.

`Complex setCoefficient (int m, Complex z)`
Replaces the Fourier coefficient $\boldsymbol{G}(\mathtt{m} \bmod M)$ by the complex value `z`, with $M = |\boldsymbol{G}|$ as above.

`Complex setCoefficient (int m, double a, double b)`
Replaces the Fourier coefficient $\boldsymbol{G}(\mathtt{m} \bmod M)$ by the complex value $z = \mathtt{a} + \mathrm{i} \cdot \mathtt{b}$, with $M = |\boldsymbol{G}|$ as above.

`int size ()`
Returns the length (M) of the Fourier descriptor.

`int getMaxNegHarmonic ()`
Returns the max. negative harmonic $m = -(M-1) \div 2$ for this Fourier descriptor (of length M).

`int getMaxPosHarmonic ()`
Returns the max. positive harmonic $m = M \div 2$ for this Fourier descriptor (of length M).

`int getMaxCoefficientPairs ()`
Returns the maximum number of coefficient pairs, $(M-1) \div 2$, for this Fourier descriptor (of length M).

`void truncate (int Mp)`
Truncates this Fourier descriptor to the `Mp` lowest-frequency coefficients (see Eqn. (26.23)).

Comparing Fourier descriptors

`double distanceComplex (FourierDescriptor fd2)`
Returns the complex-valued distance ($\mathrm{dist}_{\mathrm{C}}(\boldsymbol{G}_1, \boldsymbol{G}_2)$, see Eqn. (26.114)) between *this* Fourier descriptor ($\boldsymbol{G}_1$) and another Fourier descriptor `fd2` ($\boldsymbol{G}_2$). The zero-coefficients are ignored.

`double distanceComplex (FourierDescriptor fd2, int Mp)`
As above, but using only `Mp` coefficient pairs (see Eqn. (26.114)).

`double distanceMagnitude (FourierDescriptor fd2)`
Returns the magnitude-only distance ($\mathrm{dist}_{\mathrm{M}}(\boldsymbol{G}_1, \boldsymbol{G}_2)$, see Eqn. (26.112)) between *this* Fourier descriptor ($\boldsymbol{G}_1$) and another Fourier descriptor `fd2` ($\boldsymbol{G}_2$). The zero-coefficients are ignored.

`double distanceMagnitude (FourierDescriptor fd2, int Mp)`
As above, but using only `Mp` coefficient pairs (see Eqn. (26.112)).

Shape reconstruction

`Complex[] getReconstruction (int N)`
Returns the shape reconstructed from the complete Fourier descriptor as a sequence of `N` complex-valued contour points. The contour points are obtained by evaluating `getReconstructionPoint(t)` at uniformly spaced positions $\mathtt{t} \in [0, 1)$.

`Complex[] getReconstruction (int N, int Mp)`
Returns a partial shape reconstruction from `Mp` Fourier coefficient pairs as a sequence of `N` complex-valued contour points.

`Complex getReconstructionPoint (double t)`
: Returns a single point (as a complex value) on the continuous contour for path parameter `t` $\in [0, 1)$, reconstructed from the complete Fourier descriptor (see Eqn. (26.20)).

`Complex getReconstructionPoint (double t, int Mp)`
: Returns a single point (as a complex value) on the continuous contour for path parameter `t` $\in [0, 1)$, reconstructed from `Mp` Fourier coefficient pairs.

Normalization

`FourierDescriptor[] makeInvariant ()`
: Returns a pair of Fourier descriptors $(\boldsymbol{G}^{\mathrm{A}}, \boldsymbol{G}^{\mathrm{B}})$ that are normalized for scale, start point shift and shape rotation (see Alg. 26.6).

`double makeRotationInvariant ()`
: Normalizes the Fourier descriptor for shape rotation by phase-shifting all coefficients (see Alg. 26.6). Returns the estimated rotation angle β.

`double makeScaleInvariant ()`
: Normalizes the Fourier descriptor for scale by multiplying with a common factor, such that the L_2 norm of the resulting vector is 1. Returns the scale factor that was applied for normalization.

`FourierDescriptor[] makeStartPointInvariant ()`
: Returns a pair of normalized Fourier descriptors $(\boldsymbol{G}^{\mathrm{A}}, \boldsymbol{G}^{\mathrm{B}})$, one for each start point normalization angles φ_{A} and $\varphi_{\mathrm{B}} = \varphi_{\mathrm{A}}$ $+ \pi$, respectively (see Alg. 26.7).

`void makeTranslationInvariant ()`
: Modifies this Fourier descriptor by setting the coefficient $\boldsymbol{G}(0)$ to zero. This method is rarely needed because $\boldsymbol{G}(0)$ is ignored for matching.

`FourierDescriptorUniform` (class)

This sub-class of `FourierDescriptor` represents Fourier descriptors obtained from uniformly sampled contours, as described in Alg. 26.2. It provides the constructor methods

`FourierDescriptorUniform (Point2D[] V)`,

`FourierDescriptorUniform (Point2D[] V, int Mp)`,

where `V` is a sequence of M contour points (`Point2D`), assumed to be uniformly sampled. The first constructor creates a full Fourier descriptor with M coefficients (see Alg. 26.2). The second constructor creates a Fourier descriptor with `Mp` coefficient *pairs* (i.e., $2 \cdot \texttt{Mp} + 1$ coefficients), as described in Alg. 26.3

`FourierDescriptorFromPolygon` (class)

This sub-class of `FourierDescriptor` represents Fourier descriptors obtained directly from polygons (without contour sampling, see Alg. 26.5). It provides the single constructor method

`FourierDescriptorFromPolygon (Point2D[] V, int Mp)`, where `V` is a sequence of polygon vertices and `Mp` specifies the number of Fourier coefficient pairs.

`PolygonSampler` (class)

Instances of this utility class can be used to produce uniformly sampled polygons.

`Point2D[] samplePolygonUniformly (Point2D[] V, int M)`
Samples the closed polygon path specified by the vertices in `V` at `M` equi-distant positions and returns the resulting point sequence (see Alg. 26.1).

Example

The code example in Prog. 26.1 demonstrates the use of the Fourier descriptor API. It assumes that the binary input image (`ip`) contains at least one connected foreground region. Region labeling and contour extraction is applied first, using methods provided by the `imagingbook.regions` and `imagingbook.contours` packages.[24] Subsequently, the longest region contour (`C`) is used to create a Fourier descriptor (`fd`) with $M_{\mathrm{P}} = 15$ coefficient pairs. A partial reconstruction is calculated from the original Fourier descriptor with 100 sample points along the contour. The last lines show how a pair of invariant descriptors ($\boldsymbol{G}^{\mathrm{A}}, \boldsymbol{G}^{\mathrm{B}}$) is obtained by applying the `makeInvariant()` method. Note that the code fragment in Prog. 26.1 is not complete but would typically be part of the `run()` method in an ImageJ plugin. The full version and additional code examples can be found on the book's website.

26.8 Discussion and Further Reading

The use of Fourier descriptors for shape description and matching dates back to the early 1960's [55, 81], advanced by the work of Zahn and Roskies [262], Granlund [93], Richard and Hemami [196], and Persoon and Fu [183, 184] in the 1970s, particularly in the context of character recognition and aircraft identification. Making Fourier descriptors invariant against various geometric transformations was a key issue from the very beginning, and several relevant contributions were published in the 1980s, including [245], [57] [143], and [189]. Unfortunately, as illustrated in this chapter, to achieve robust invariance and uniqueness of representation in practice is not as easy as sometimes suggested in the literature, despite the simplicity and elegance of the underlying theory. In practice, normalization for descriptor invariance is quite difficult for arbitrary shapes because of possibly vanishing Fourier coefficients and the resulting sensitivity to noise.

Fourier descriptors have nevertheless become popular in a wide range of applications, including geology and, in particular, biological imaging, as documented by the work of Lestrel and others in [146].

[24] See also Chapter 10.

```
 1 ...
 2 import imagingbook.lib.math.Complex;
 3 import imagingbook.pub.fd.*;
 4 import imagingbook.pub.regions.*;
 5
 6 ByteProcessor ip ...;   // assumed to contain a binary image
 7
 8  // segment ip and select the longest outer region contour:
 9 RegionContourLabeling labeling =
10       new RegionContourLabeling(ip);
11 List<Contour> outerContours =
12       labeling.getAllOuterContours(true);
13 Contour contr = outerContours.get(0);   // get the longest contour
14 Point2D[] V = contr.getPointArray();
15
16  // create the Fourier descriptor for V with 15 coefficient pairs:
17 FourierDescriptor fd = new FourierDescriptorUniform(V, 15);
18
19  // reconstruct the corresponding shape with 100 contour points:
20 Complex[] R = fd.getReconstruction(100);
21
22  // create a pair of invariant descriptors (G^A, G^B):
23 FourierDescriptor[] fdAB = fd.makeInvariant();
24 FourierDescriptor fdA = fdAB[0];   // = G^A
25 FourierDescriptor fdB = fdAB[1];   // = G^B
26 ...
```

Prog. 26.1
Fourier descriptor code example. The input image `ip` is assumed to contain a binary image (line 6). The class `RegionContourLabeling` is used to find connected regions (line 10). Then the list of outer contours is retrieved (line 12) and the longest contour is assigned to `V` as an array of type `Point2D` (lines 13–14). In line 17, the contour `V` is used to create a Fourier descriptor with 15 coefficient pairs. Alternatively, we could have created a Fourier descriptor of the same length (number of coefficients) as the contour and then truncated it (using the `truncate()` method) to the specified number of coefficient pairs. A partial reconstruction of the contour (with 100 sample points) is calculated from the Fourier descriptor `fd` in line 20. Finally, a pair of invariant descriptors (contained in the array `fdAB`) is calculated in line 23.

Fourier descriptors have been extended to accommodate affine transformations and applied to 3D object identification [5] and stereo matching [257].

Although Fourier descriptors have been investigated to handle open contours and partial shapes [148], they are naturally best suited to dealing with closed contours, as we have described. Of course, this is a limitation if shapes are only partially visible or occluded. The presentation in this chapter was limited to what are frequently called "elliptical" Fourier descriptors [93], since they are most popular and well known. Other types of Fourier descriptors have been proposed, which are not covered here but can be found elsewhere in the literature (see, e.g., [126, p. 534] and [174, Ch. 7]).

26.9 Exercises

Exercise 26.1. Verify that the DFT spectrum is periodic, that is, that $\boldsymbol{G}(-m) = \boldsymbol{G}(M-m)$ holds for arbitrary $m \in \mathbb{Z}$ (as claimed in Eqn. (26.22)).

Exercise 26.2. Algorithm 26.9 shows an alternative solution to uniform polygon sampling. Implement this algorithm and verify that it is equivalent to Alg. 26.1 (implemented as method `samplePolygonUniformly()` in class `PolygonSampler`, see Sec. 26.7).

Exercise 26.3. Assume that the complete outer contour of a binary region is given as a sequence of P boundary pixels with coordinates

Alg. 26.9
Uniform sampling of a polygon path (alternative to Alg. 26.1, proposed by J. Heinzelreiter).

```
SamplePolygonUniformly$(V, M)$
    Input: $V = (\boldsymbol{v}_0, \ldots, \boldsymbol{v}_{N-1})$, a sequence of $N$ points representing
    the vertices of a closed 2D polygon; $M$, number of desired sample
    points. Returns a new sequence $\boldsymbol{g} = (g_0, \ldots, g_{M-1})$ of complex
    values representing sample points sampled uniformly along the
    path of the input polygon $V$.

    $N \leftarrow |V|$
    $\Delta \leftarrow \frac{1}{M} \cdot \mathsf{PathLength}(V)$        ▷ segment length $\Delta$, see Alg. 26.1
    Create map $\boldsymbol{g}\colon [0, M-1] \to \mathbb{C}$        ▷ complex point sequence $\boldsymbol{g}$
    $\boldsymbol{g}(0) \leftarrow \mathsf{Complex}(V(0))$
    $i \leftarrow 0$        ▷ index of path segment $\langle V_i, V_{i+1} \rangle$
    $k \leftarrow 1$        ▷ index of first unassigned point in $\boldsymbol{g}$
    $d_\mathrm{p} \leftarrow 0$        ▷ path distance between $V(i)$ and $V(k-1)$
    while $(i < N) \wedge (k < M)$ do
        $\boldsymbol{v}_\mathrm{A} \leftarrow V(i)$
        $\boldsymbol{v}_\mathrm{B} \leftarrow V((i+1) \bmod N)$
        $\delta \leftarrow \|\boldsymbol{v}_\mathrm{B} - \boldsymbol{v}_\mathrm{A}\|$        ▷ Euclidean distance
        if $(\Delta - d_\mathrm{p}) \leq \delta)$ then
            $\boldsymbol{x} \leftarrow \boldsymbol{v}_\mathrm{A} + \frac{\Delta - d_\mathrm{p}}{\delta} \cdot (\boldsymbol{v}_\mathrm{B} - \boldsymbol{v}_\mathrm{A})$        ▷ $\boldsymbol{x}_k$ by lin. interpolation
            $\boldsymbol{g}(k) \leftarrow \mathsf{Complex}(\boldsymbol{x})$
            $d_\mathrm{p} \leftarrow d_\mathrm{p} - \Delta$
            $k \leftarrow k + 1$
        else
            $d_\mathrm{p} \leftarrow d_\mathrm{p} + \delta$
            $i \leftarrow i + 1$
    return $\boldsymbol{g}$.
```

$V = (\boldsymbol{p}_0, \ldots, \boldsymbol{p}_{P-1})$. To produce a Fourier descriptor of length $M < P$ there are several options:

1. Sample the original contour V at M uniformly-spaced positions (see Alg. 26.1) and then calculate the Fourier descriptor of length M using Alg. 26.2.
2. Calculate a partial Fourier descriptor of length M' from the original contour V using Alg. 26.3.
3. Calculate the full Fourier descriptor (of length M) from the original contour V (using Alg. 26.2) and subsequently truncate[25] the Fourier descriptor to length M', as described in Eqns. (26.23) and (26.24).
4. Treat the original boundary coordinates V as the vertices of a closed polygon and calculate a Fourier descriptor with $M_\mathrm{P} = M \div 2$ coefficient pairs, using the trigonometric method described in Alg. 26.5.

Compare these approaches and discuss their individual merits or disadvantages in terms of efficiency and accuracy.

Exercise 26.4. Test the Fourier descriptor normalization described in Algs. 26.6 and 26.7 (implemented by method `makeInvariant()` in the Java API) for changes in scale, start point shift, and shape rotation on a suitable set of binary shapes (e.g., images from the

[25] See method `truncate(int Mp)` in Sec. 26.7.

KIMIA dataset [134]). See the examples for shape rotation and (implicit) start point shifts in Fig. 26.21. How reliably do the normalized Fourier descriptors of the modified shapes match to their corresponding originals?

Exercise 26.5. Magnitude-only matching (see Sec. 26.6.1) is much simpler than complex-valued matching (see Sec. 26.6.2) of Fourier descriptors, since no normalization for phase (start point shift and shape rotation) is required. However, it can be assumed that different shapes are more likely to be confused if the phase information is ignored. Test this hypothesis on a large number and variety of different shapes. Compare the confusion probability for magnitude-only vs. complex-valued matching.