10

Direction in 2D

Directional processing of visual signals is the largest single analysis toolbox of *mammalian visual* system: it feeds other specialized visual processing areas [114, 173, 235], e.g., face recognition. Directional analysis is gaining increased traction even in computer vision, as it moves from single-problem-solving systems towards multi-problem-solving platforms. Nearly all applications of image analysis now have alternatives using direction tensor fields. The necessary tools are more modern and offer advanced low-level signal processing that was hitherto reserved to processing of high-level tokens, such as binarized or skeletonized edge maps. In 2D, the earliest solutions to the problem of finding the direction of an image patch, e.g., [51, 116], consisted in projecting the image onto a number of fixed orthogonal functions. The projection coefficients were then used to evaluate the orientation parameter of the model. When the number of filters used is increased, the local image is described better and better, but the inverse function, mapping the coefficients to the optimal orientation, increase greatly in complexity. A generalization of the inverse projection approach to higher dimensions becomes therefore computationally prohibitive. Here we will follow a different approach by modeling the shapes of isocurves via tensors.

10.1 Linearly Symmetric Images

We will refer to a small 2D image patch around a point as an *image*, to the effect that we will treat the local image patches in the same way as the global image. Let the scalar function f, taking a two-dimensional vector $\mathbf{r} = (x, y)^T$ as argument, represent an image. Assume that f$\mathbf{r} = (x, y)^T$ is a two-dimensional real vector that represents the coordinates of a point in a plane on which an image $f(x, y)$ is defined. Furthermore, assume that $\mathbf{k} = (k_x, k_y)^T$ is a two-dimensional unit vector representing a constant direction in the plane of the image.

Definition 10.1. *The function f is called a* linearly symmetric image *if its isocurves have a common direction, i.e., there exists a scalar function of one variable g such that*

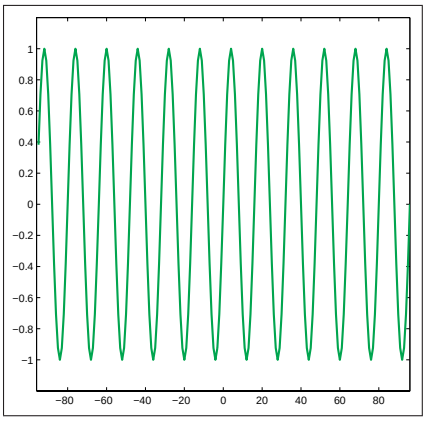

Fig. 10.1. The graph represents the 1D function $g(t) = \sin(\omega t)$ that will be used to construct a linearly symmetric 2D function

$$f(x, y) = g(\mathbf{k}^T \mathbf{r}) = g(k_x x + k_y y) \tag{10.1}$$

The direction of the linear symmetry is $\pm\mathbf{k}$.

The term is justified in that F, the 2D Fourier transform of f, is concentrated to a line, as will be shown below. In addition, all isocurves of linearly symmetric functions are lines that have a common direction $\mathbf{k}$, i.e., they are parallel to each other. Note that the term *isocurves* refers to the fact that the values of g and thereby f are invariant when one moves along certain curves in the argument domain. For linearly symmetric images, these curves are lines.

It should be noted that while $g(t)$ is a function of one free variable, $g(\mathbf{k}^t \mathbf{r})$ is a function of two free variables, $(x, y)^T$ since $\mathbf{k}$ is constant. In the rest of this section we will assume that the argument domain of f is two-dimensional, whereas that of g is one-dimensional and g is a "constructor" of f via Eq. (10.1) and $\mathbf{k}$ whenever f is linearly symmetric. Therefore $g(\mathbf{k}^T \mathbf{r})$ generates an image despite that g by itself is a one-dimensional function. By definition, images with the linear symmetry property have the same gray value at all points $\mathbf{r}$ satisfying $\mathbf{k}^T \mathbf{r} = C$ for a given value C. Because $\mathbf{k}^T \mathbf{r} = C$ describes a line in the (x, y)-plane, it follows that along this line the gray values of the image do not change and this gray value equals to $g(C)$. In such images, the only occasion when g can change is when the argument of g changes, i.e., when the constant C assumes another value. However, the curves $\mathbf{k}^T \mathbf{r} = C_1$, $\mathbf{k}^T \mathbf{r} = C_2$, ... $\mathbf{k}^T \mathbf{r} = C_n$, with C_i being different constants, represent lines that are shifted versions of each other, all having the same direction, $\mathbf{k}$. Consequently, the one-dimensional function g is a profile of the two-dimensional function $g(\mathbf{k}^T \mathbf{r})$ along any line perpendicular to the line $\mathbf{k}^T \mathbf{r} = C$.

Local images, which can be extracted by multiplying the original image with an appropriate window function, are, from mathematical viewpoint, no different than the larger original image from which these local images are "cut". For notational

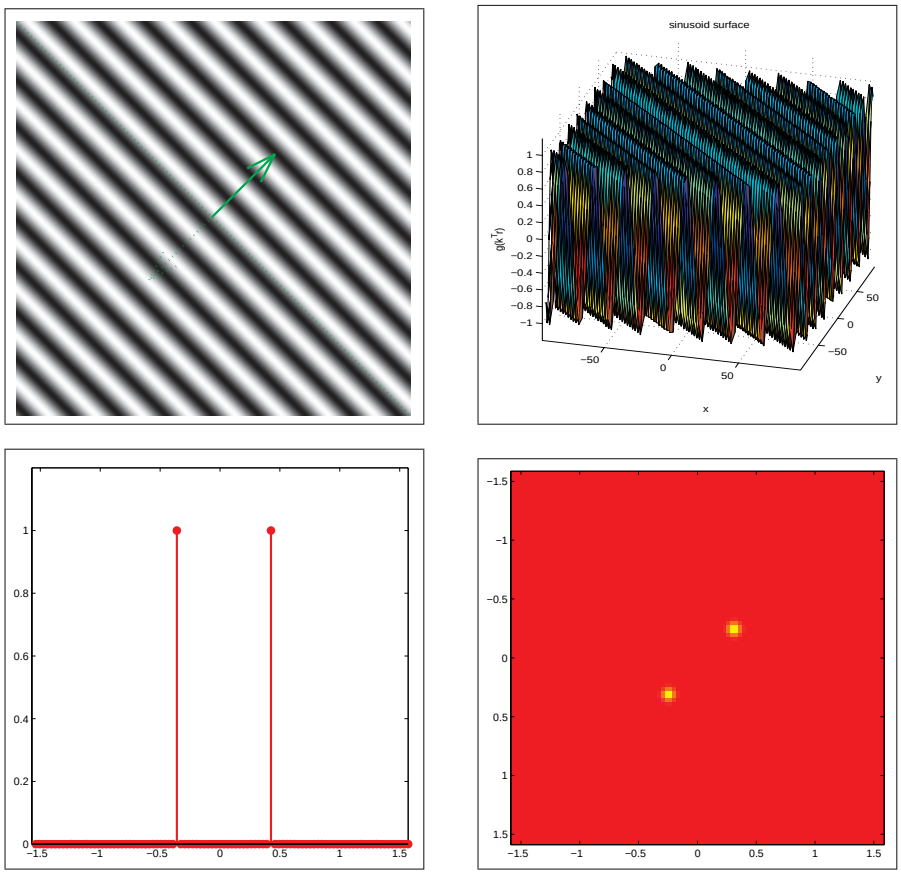

Fig. 10.2. In *top, left*, the image generated by $g(t) = \sin(\omega t)$ with the argument $t = \mathbf{k}^T\mathbf{r}$ is given. The *solid* and *dashed* vectors represent $\mathbf{k}$ and $-\mathbf{k}$ respectively. The 3D graph in *top, right* represents $g(\mathbf{k}^T\mathbf{r})$ as a surface. The FT magnitudes of $g(t)$ and $g(\mathbf{k}^T\mathbf{r})$ are shown in *bottom, left* and *bottom, right*. The FT coordinates are in the angular frequencies ω and $(\omega_x, \omega_y)^T$

simplicity, we will therefore not make a distinction between an image and a local neighborhood of it. Both variants will be referred to as an image here, unless an ambiguity calls for further precision.

Below we detail the process of constructing linearly symmetric images from 1D functions first by three examples of 1D functions g, that are continuous. The last one of these will model an ideal line. Then we will study a discontinuous g which will be a model of ideal edges. Both ideal lines and ideal edges have been used to model and to detect discontinuities in image processing.

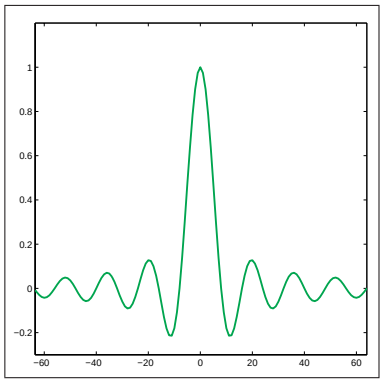

Fig. 10.3. The graph represents the sinc function, Eq. (10.4)

Example 10.1. *The function*

$$g(t) = \sin(\omega t), \qquad \textit{with} \qquad \omega = \frac{2\pi}{12}, \tag{10.2}$$

rendered in Fig. 10.1, oscillates around zero. This sinusoid is used to construct the linearly symmetric 2D function, shown in Fig. 10.2 (top, left). To render the negative values of g we added the constant 0.5 and rescaled the gray values to the effect that white represents 1 and black represents -1 by using 256 gray tones and linear mapping. The dashed green line through origin shows an example isocurve. We recall that an isocurve is the curve that joins all points having a certain gray value. The example isocurve in the constructed 2D image is clearly a line, along which the gray value of the image is the same. In fact, every other isocurve is also a line and all isocurves are parallel, just as they should be, because of the way the image is constructed.

By construction, the gray value of the image cannot change unless the argument of g changes. For any given image location $\mathbf{r}$, the argument of g will equal a constant value C

$$\mathbf{k}^T\mathbf{r} = C \tag{10.3}$$

to the effect that it is possible to restrict the changes of $\mathbf{r}$ to a curve so that C is invariant. By virtue of Eq. (10.3), this path is a line, and the equation represents parallel lines when $\mathbf{k}$ is fixed. We can move in the image, i.e., change $\mathbf{r}$, along a line which is perpendicular to the line shown as the dashed diagonal, i.e along $\mathbf{k}$, and obtain changes in the value of C, which in turn changes the values of g. In this path, the obtained function or gray values are identical to the values of the original 1D function, $g(t) = \cos(\omega t)$. The solid green arrow illustrates the vector $\mathbf{k} = (\cos(\frac{\pi}{4}), \sin(\frac{\pi}{4}))^T$ used to build the image. The representation is not unique because $-\mathbf{k}$ (dashed) would have generated the same image. In other words, the $\mathbf{k}$ used to construct the linearly symmetric image is unique only up to a sign factor ± 1. The color surface in 3D shows

the same image as a landscape, illustrating that the gray variations are identical to those in Fig. 10.1 across the isocurves.

The magnitudes of the Fourier transforms of $g(t)$ and $g(\mathbf{k}^T\mathbf{x})$ are also given in Fig. 10.2 (bottom). The red color represents the value zero in the image. The bright (yellowish) spots represent the largest values. In the image, the 2D Fourier transform magnitudes clearly equal zero outside of the two bright points.

Example 10.2. *The 1D sinc function*

$$g(t) = \frac{\sin(\omega t)}{\omega t}, \qquad \textit{with} \qquad \omega = \frac{2\pi}{12}, \tag{10.4}$$

is plotted in Fig. 10.3. The synthetic image represented by the function $g(\mathbf{k}^T\mathbf{r})$ is linearly symmetric. Its image is illustrated by the gray image in Fig. 10.4 (top) which differs from the one in Fig. 10.2 only by the choice of g. The function values are scaled and shifted to be rendered by the available 256 gray tones. The solid and the dashed vectors represent $\mathbf{k}$ and $-\mathbf{k}$ respectively. Both this image and the gray image in Fig. 10.2 have the same vector $\mathbf{k} = (\cos(\frac{\pi}{4}), \sin(\frac{\pi}{4}))^T$, which represents the direction orthogonal to the isocurve direction. In the direction of $\mathbf{k}$, any cross section of the image is identical to the sinc function of Fig. 10.3, as illustrated by the 3D graph in Fig. 10.4, which shows $g(\mathbf{k}^T\mathbf{r})$ as a surface.

In the (2D) color image, we note that the magnitudes of the Fourier transformed function equal zero (red color) outside of a line passing through the origin, indicated by bright yellow in Fig. 10.4 (bottom, right). The line has the direction $\mathbf{k}$. Along the line, the Fourier transform magnitude has the same shape as the (1D) Fourier transform magnitude (bottom, left).

We will bring further precision to the relationship between the 1D and 2D Fourier transforms of the linearly symmetric functions below. For now we note the result of this example, as illustrated by Fig. 10.5 bottom, right, is consistent with that of the previous example, Fig. 10.2 bottom, right. Both Fourier transform magnitudes vanish outside a central line having the direction $\mathbf{k}$, whereas on the line itself both magnitudes have at least the same magnitude variations as their 1D counterparts shown on the respective left. From the magnitudes of Fourier transformed functions, we can in general not deduce the underlying complex values. However, there is one exception to that which is a result of the null property of norms, i.e., the magnitudes of complex numbers are zero if and only if the complex numbers are zero. The Fourier transforms of the two illustrated example images possessing linear symmetry must consequently have not only magnitudes but also complex values that equal zero outside the referenced line.

The second example actually showed the same sinusoid as in the first one, with the difference that in the second example, the sinusoid attenuates gradually as $1/t$. The Fourier transform magnitude of the Sinc example is therefore more spread as compared to that of the pure sinusoid, which consists of a pair of Dirac pulses.

The sinusoid is neither a pure line nor a pure edge, but yet it has a direction. The classical edge and line detection techniques in image processing model and detect

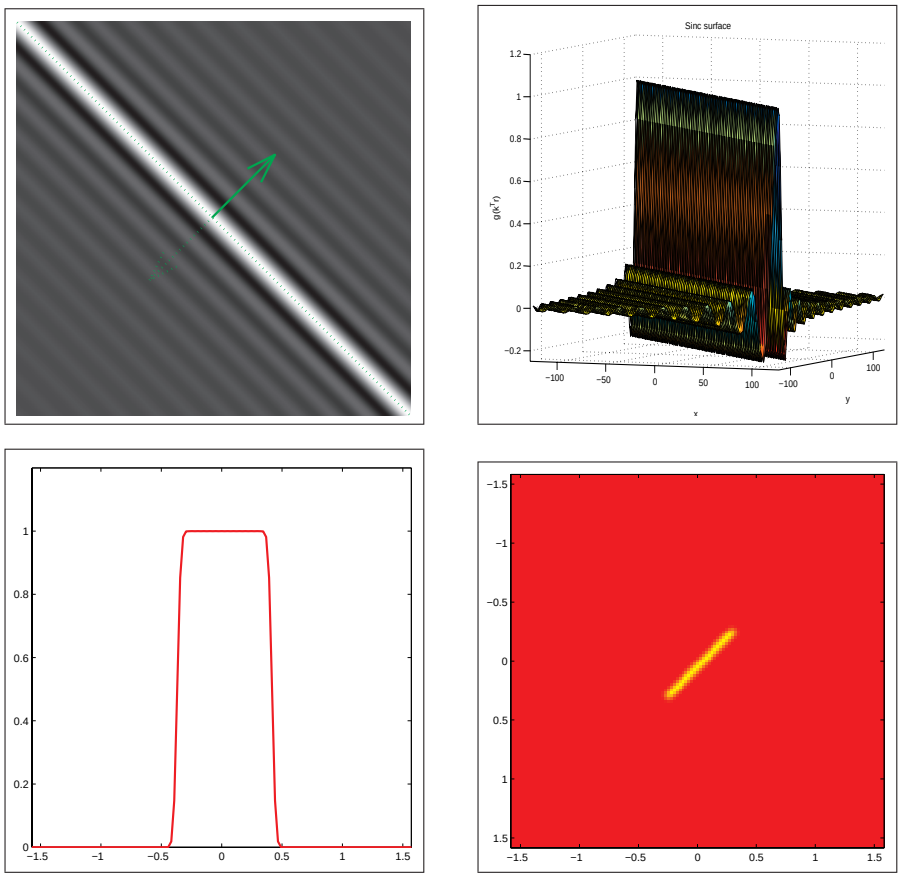

Fig. 10.4. (*Top*) The gray image is generated by substituting $t = \mathbf{k}^T\mathbf{r}$ in Eq. (10.4). The 3D graph shows $g(\mathbf{k}^T\mathbf{r})$. (*Bottom*) The 1D and 2D FT magnitudes of $g(t)$ and $g(\mathbf{k}^T\mathbf{r})$, respectively

pure lines and edges, without a provision for other types of patterns that have well-defined directions. In the next example, we show that pure lines can be modeled as a linearly symmetric function generated by means of an analytic function, a Gaussian.

Example 10.3. *The* 1*D Gaussian*

$$g(t) = \exp\left(-\frac{t^2}{2\sigma^2}\right), \qquad \textit{with} \qquad \sigma = 3, \tag{10.5}$$

is plotted as the green curve in Fig. 10.5. The synthetic image represented by the function $g(\mathbf{k}^T\mathbf{r})$ *is linearly symmetric and is illustrated by the gray image in Fig. 10.5. The function values are scaled and linearly mapped to 256 gray tones, with 0 corresponding to black, and 1 corresponding to white. In the direction of* $\mathbf{k}$*, any cross section of the image is identical to the 1D Gaussian of Fig. 10.5.*

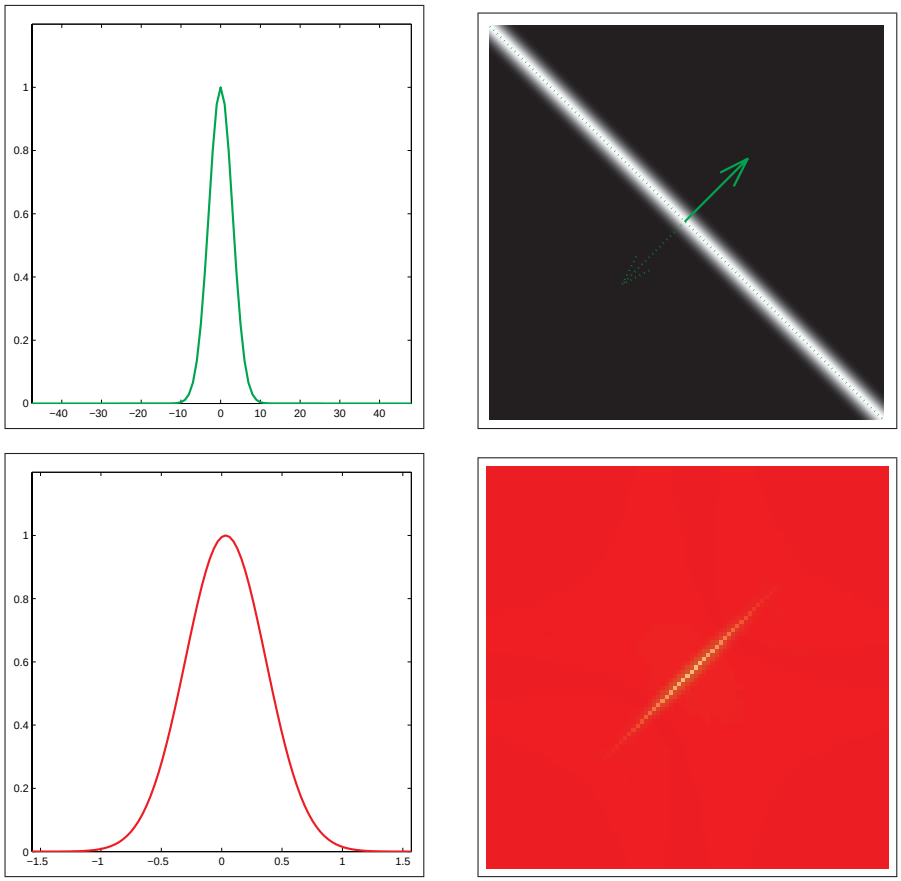

Fig. 10.5. (*Top*) The graph represents the 1D Gaussian $g(t)$ in Eq. (10.5) and the 2D function generated by the substitution $t = \mathbf{k}^T\mathbf{r}$. The *solid* and *dashed* vectors represent $\mathbf{k}$ and $-\mathbf{k}$ respectively. (*Bottom*) The 1D FT magnitude of $g(t)$ and the 2D FT magnitude of $g(\mathbf{k}^T\mathbf{r})$ are illustrated by the (*left*) graphics and the (*right*) image, respectively

When we study the 2D Fourier transform magnitudes of this linearly symmetric image, Fig. 10.5 bottom, right, we note that it too equals to zero (red) outside of the same central line (bright yellow) as in the previous two examples. The line has a profile matching the 1D version of the Fourier transform magnitude, shown in the bottom, left graph.

Example 10.4. *The 1D step function*

$$g(t) = \begin{cases} 1, & \text{if } t \geq 0, \\ 0, & \text{otherwise}, \end{cases} \tag{10.6}$$

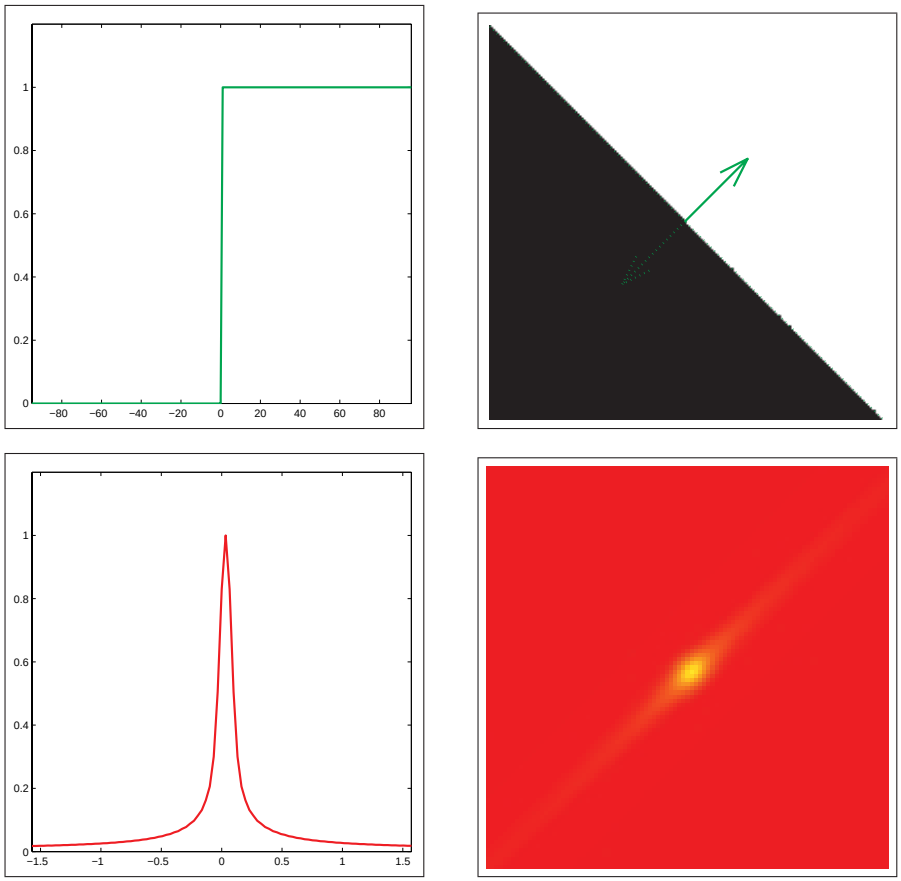

Fig. 10.6. (*Top*) The graph represents the 1D step function $g(t)$ in Eq. (10.6) and the 2D function generated by the substitution $t = \mathbf{k}^T\mathbf{r}$. The *solid* and *dashed* vectors represent $\mathbf{k}$ and $-\mathbf{k}$, respectively. (*Bottom*) The 1D FT magnitude of $g(t)$ and the 2D FT magnitude of $g(\mathbf{k}^T\mathbf{r})$ are illustrated by the (*left*) graphics and the (*right*) image, respectively

is discontinuous. It is shown as the green curve in Fig. 10.6 with its corresponding linearly symmetric gray image, which is obtained by sampling

$$g(\mathbf{k}^T\mathbf{r}) = \begin{cases} 1, & \textit{if} k_x x + k_y y \geq 0, \\ 0, & \textit{otherwise}. \end{cases} \tag{10.7}$$

The gray image models what came to be known as the ideal edge in image processing. Despite the fact that it is discontinuous, it too is a linearly symmetric function, with Fourier transform magnitudes concentrated to the same central line as in the previous examples.

Fig. 10.7. In real images, linear symmetries often resemble "lines" or "edges" but they appear even as object or texture boundaries. Some of these are illustrated by colored lines in this photograph

Example 10.5. *The linearly symmetric images appear frequently as local images, both in images of human-made environments and in images of the nature. They are often perceived as lines or edges (Fig. 10.7), or as repetitive patterns called textures (Fig. 10.8). Although the cross sections of their isocurves are seldom like the ideal lines and edges, they usually have a well-distinguishable direction, with Fourier transform magnitudes concentrated to a line, as Fig. 10.8 shows.*

We give precision to the example indications regarding the Fourier transform of the linearly symmetric images in the following lemma. We recall that the argument domain of g is one-dimensional, whereas that of f is two-dimensional by the adopted convention.

Lemma 10.1. *A linearly symmetric image $f(\mathbf{r}) = g(\mathbf{k}^T\mathbf{r})$ has a 2D Fourier transform concentrated to a line through the origin:*

$$F(\omega_x, \omega_y) = G(\mathbf{k}^T\boldsymbol{\omega})\delta(\mathbf{u}^T\boldsymbol{\omega}) \tag{10.8}$$

where $\mathbf{k}$, $\mathbf{u}$ are orthogonal vectors and δ is the Dirac distribution. The vector $\boldsymbol{\omega}$ is the angular frequency vector $\boldsymbol{\omega} = (\omega_x, \omega_y)^T$. The function G is the one-dimensional Fourier transform of g.

♦

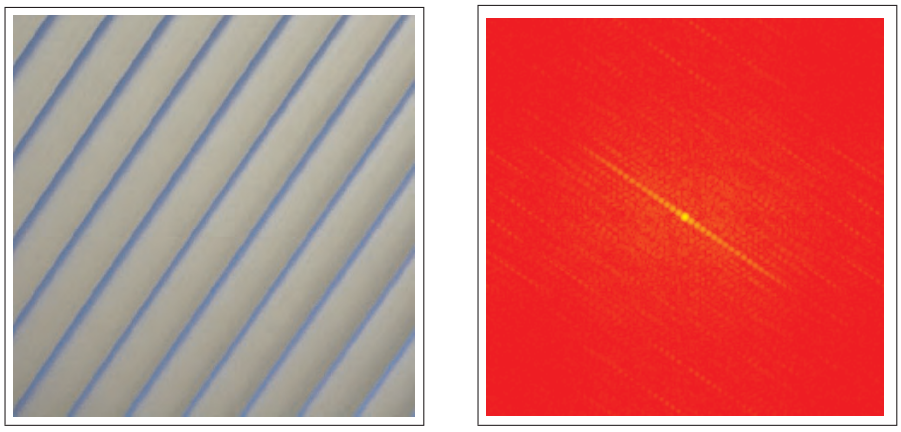

Fig. 10.8. *Left*: a real image that is linearly symmetric. It shows a close-up view of the blinds. *Right*: The Fourier transform magnitude of a neighborhood in the central part of the original (brightness) image. Notice that the power is concentrated to a line orthogonal to isocurves in the original

Lemma 10.1 is proven in the Appendix of this chapter, Sect 10.17. It states that the function $g(\mathbf{k}^T\mathbf{r})$, which is in general a "spread" function such as a sinusoid or an edge, is compressed to a line having the direction $\mathbf{k}$ in the Fourier domain. Even more important, it says that as far as $\mathbf{k}$ is concerned the choice of G, and thereby g, has no significance. This is because, $\mathbf{k}$, the angle at which all nonzero F reside, remains the same no matter what G is. This is achieved by the Dirac pulse δ, which becomes a line pulse along the infinite line $\mathbf{u}^T\boldsymbol{\omega} = 0$ by the expression $\delta(\mathbf{u}^T\boldsymbol{\omega})$. Because $\mathbf{u}$ is the normal direction of this line and $\mathbf{u}$ is orthogonal to $\mathbf{k}$, the direction of the spectral line $\mathbf{u}^T\boldsymbol{\omega} = 0$ coincides with the vector $\mathbf{k}$. We have already observed this line in red-colored images of Examples 10.1–10.5, as a concentration of the magnitudes to the central line, in the same direction as the same (green) $\mathbf{k}$ vector shown in the gray images, regardless g.

Along this central spectral line, not only the magnitude but also the complex values conform to that of the 1D Fourier transform. This is because G is the 1D Fourier transform of g, and Eq. (10.8) is a formula for how to produce the 2D Fourier transform of the linearly symmetric functions only from the 1D Fourier transform G and the isocurve normal $\mathbf{k}$. According to the lemma, the vector $\mathbf{u}$ can always be deduced from $\mathbf{k}$ up to a sign factor, because it is orthogonal to $\mathbf{k}$. Due to limitations of the illustration methods, Examples 10.1–10.5 could only be indicative about this more powerful result, and this only as far as the Fourier transform magnitudes are concerned.

Consequently, to determine whether or not an image is linearly symmetric is the same thing as to quantitate to what extent the Fourier transform vanishes outside of a line, a property which will be exploited to construct computer algorithms below. Such algorithms can be constructed conveniently to describe textures possessing a

direction, or by detecting the lack of linear symmetry, to describe textures lacking direction. Measuring the lack of linear symmetries has been frequently used as a way of detecting corners in image processing.

10.2 Real and Complex Moments in 2D

In image analysis there are a variety of occasions when we need to quantitate functions by comparing them to other functions. Assuming that the integral exists, the quantity

$$m_{pq}(\kappa) = \langle x^p y^q, \kappa \rangle = \iint x^p y^q \kappa(x,y) dxdy \tag{10.9}$$

with p and q being nonnegative integers, is the *real moment* p, q of the function κ. If κ has a finite extension, then the real moments defined as above are projections of an integrable function onto the vector space of polynomials. It follows from the Weierstrass theorem [193, 209], that the vector space of the polynomials is powerful enough to approximate a finite extension function κ to a desired degree of accuracy. In that, the approximation property of moments is comparable to the FCs, although the polynomial basis of moments is not orthogonal, whereas the Fourier basis is. Nonetheless, moments are widely used in applications. If κ is a positive function then it is possible to view it as a probability distribution, after a normalization with m_{00}. Accordingly,

$$\bar{c} = (\bar{x}, \bar{y})^T = \frac{1}{m_{00}} (m_{10}, m_{01})^T \tag{10.10}$$

represents the centroid or the mean vector of the function κ. The quantity

$$\begin{aligned} \tilde{m}_{pq}(\kappa) &= \langle \left(x - \frac{m_{10}}{m_{00}} \right)^p \left(y - \frac{m_{01}}{m_{00}} \right)^q, \kappa \rangle \\ &= \iint \left(x - \frac{m_{10}}{m_{00}} \right)^p \left(y - \frac{m_{01}}{m_{00}} \right)^q \kappa(x,y) dxdy \end{aligned} \tag{10.11}$$

related to real moments, is called the central moment p, q of the function κ. Both real moments and central moments have been utilized as tools to quantitate the shape of a finite extension image region. We will discuss this further in Sect. 17.4.

Another type of moment, which we will favor over real moments in what follows, is

$$I_{pq}(\kappa) = \langle (x - iy)^p (x + iy)^q, \kappa \rangle = \iint (x + iy)^p (x - iy)^q \kappa(x,y) dxdy \tag{10.12}$$

with p and q being nonnegative integers. This is the *complex moment* p, q of the function κ. Notice that complex moments are linear combinations of the real moments, e.g.,

$$\begin{aligned} I_{20} &= m_{20} - m_{02} + i2m_{11} \\ I_{11} &= m_{20} + m_{02} \end{aligned}$$

The order number and the symmetry number of a complex moment refer to $p+q$ and $p-q$, respectively. The integrals above should be interpreted as summations when the complex moments of discrete functions are to be computed. In the following sections we will make use of (real and complex) moments as a spectral regression tool, i.e., to fit a line to the FT of a function.

10.3 The Structure Tensor in 2D

The *structure tensor*[1] or the *direction tensor* models linearly symmetric structures that are frequently found in images. To represent certain geometric properties of images, it associates 2×2 symmetric matrices that are tensors to them. This is not different from the fact that in a color image there are several color components per image point, e.g., HSB, to every point. Typically, however, the structure tensor is used to quantify shape properties of local images. As such, structure tensors are assigned to every image point to represent properties of neighborhoods.

Let the scalar function $f(\mathbf{r})$, taking the two-dimensional vector $\mathbf{r} = (x, y)^T$ as argument, represent an image, which is usually a neighborhood around an image point. As before, the (capitalized) letter F is the Fourier transform of f. We denote with $|F(\boldsymbol{\omega})|$ the magnitude spectrum of f, where $\boldsymbol{\omega} = (\omega_x, \omega_y)^T$ is the Fourier transform coordinates in angular frequencies, and with $|F(\boldsymbol{\omega})|^2$ we denote the power spectrum of f. We will use the power spectrum rather than $|F|$ to measure the significance of a given frequency in the signal because it will turn out that the average values of $|F|^2$ are easier to measure in practice than $|F|$.

The direction of a linearly symmetric function $f(\mathbf{r}) = g(\mathbf{k}^T\mathbf{r})$ is well-defined by the vector $\mathbf{k}$, but only up to a sign factor. According to lemma 10.1, if and only if f is linearly symmetric is its power spectrum, $|F|^2$ concentrated to a central line with the direction $\mathbf{k}$. The direction of this line represents the direction of the linear symmetry. We will approach estimating $\mathbf{k}$ by fitting the image power spectrum, $|F|^2$, a line in the total least square TLS sense. Consequently, it will be possible to "measure" if f is linearly symmetric by studying the error of the fit. If the error is "small" in the sense that has been defined, then our method will take this as a provision that the fit was successful and that the image approximates a linearly symmetric image $g(\mathbf{k}^T\mathbf{r})$ well. It turns out that, in this procedure, g need not be known beforehand. It will be automatically determined when the error of the fit is near zero, because we will then obtain a reliable direction along which to "cut" the image. In turn the 1D function obtained by cutting the image is g only if f is linearly symmetric. Owing to the continuity of the TLS error function, the decrease or the increase of the error will be graceful when f approaches to a linearly symmetric function or departs from one. We discuss the details of the line-fitting next.

We wish to fit an axis through the origin of the Fourier transform of an image, f, which *may or may not* be linearly symmetric. Fitting an axis to a finite set of points is classically performed by minimizing the error function:

[1] Other names of this tensor include the "second order moment tensor", "inertia tensor", "outer product tensor", and "covariance matrix".

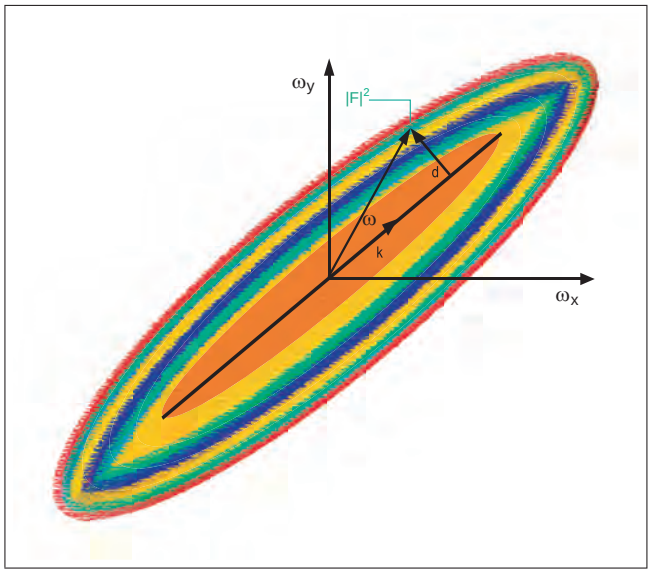

Fig. 10.9. The line-fitting process is illustrated by the linear symmetry direction vector $\mathbf{k}$, the angular frequency vector $\boldsymbol{\omega}$, and the distance vector $\mathbf{d}$. The function values $|F(\boldsymbol{\omega})|^2$ are represented by color. The frequency coordinate vector is $\boldsymbol{\omega}$

$$e(\mathbf{k}) = \sum_{\boldsymbol{\omega}} d^2(\boldsymbol{\omega}, \mathbf{k}) \tag{10.13}$$

where $d(\boldsymbol{\omega}, \mathbf{k})$ is the shortest distance between a point $\boldsymbol{\omega}$ and a candidate axis $\mathbf{k}$. This is the TLS error function for a discrete data set. Noting that $\|\mathbf{k}\| = 1$, then the projection of $\boldsymbol{\omega}$ on the vector $\mathbf{k}$ is $(\boldsymbol{\omega}^t\mathbf{k})\mathbf{k}$. As illustrated by Fig. 10.9, the vector $\mathbf{d}$ represents the difference between $\boldsymbol{\omega}$ and the projection of $\boldsymbol{\omega}$. This difference vector is orthogonal to $\mathbf{k}$

$$\mathbf{d} = \boldsymbol{\omega} - (\boldsymbol{\omega}^t\mathbf{k})\mathbf{k} \tag{10.14}$$

with its norm being equal to the shortest distance, i.e., $\|\mathbf{d}\| = d(\boldsymbol{\omega}, \mathbf{k})$. Consequently, the square of the norm of $\mathbf{d}$ provides:

$$d^2(\boldsymbol{\omega}, \mathbf{k}) = \|\boldsymbol{\omega} - (\boldsymbol{\omega}^t\mathbf{k})\mathbf{k}\|^2 \tag{10.15}$$

$$= \big(\boldsymbol{\omega} - (\boldsymbol{\omega}^t\mathbf{k})\mathbf{k}\big)^T \big(\boldsymbol{\omega} - (\boldsymbol{\omega}^t\mathbf{k})\mathbf{k}\big) \tag{10.16}$$

Since we have a Fourier transform function, F, defined on dense angular frequency coordinates in E_2, instead of a sparse point set, Eq. (10.13) needs to be modified. The following error function is a generalization of Eq. (10.13) to dense point sets. It weights the squared distance contribution at an angular frequency point $\boldsymbol{\omega}$ with the energy $|F(\boldsymbol{\omega})|^2$ and integrates all error contributions

$$e(\mathbf{k}) = \int d^2(\boldsymbol{\omega}, \mathbf{k}) |F(\boldsymbol{\omega})|^2 d\boldsymbol{\omega} \tag{10.17}$$

where $d\boldsymbol{\omega}$ equal to $d\omega_x d\omega_y$, and the integral is a double integral over E_2. The expression defines the TLS error function for a continuous data set. Interpreting the integral as a summation, Eq. (10.13) is a special case of Eq. (10.17). By construction, the contribution of $F(\boldsymbol{\omega})$ to the error is zero for points $\boldsymbol{\omega}$ along the line $t\mathbf{k}$. With increased distance of $\boldsymbol{\omega}$ to the latter axis, the contribution of $F(\boldsymbol{\omega})$ will be amplified.

If $|F(\boldsymbol{\omega})|^2$, which is the spectral energy, is interpreted as the mass density, then the error $e(\mathbf{k})$ corresponds to the inertia of a mass with respect to the axis $\mathbf{k}$ in mechanics [83]. Besides this function being continuous in $\mathbf{k}$, it has also a screening effect; that is, F is concentrated to a line if and only if $e(\mathbf{k})$ vanishes for some $\mathbf{k}$. This is to be expected because the resulting quantity when substituting Eq. (10.16) in Eq. (10.17) is the square norm of a vector-valued function, i.e.,

$$e(\mathbf{k}) = \|\big(\boldsymbol{\omega} - (\boldsymbol{\omega}^T\mathbf{k})\mathbf{k}\big)F\|^2 = \int \|\big(\boldsymbol{\omega} - (\boldsymbol{\omega}^T\mathbf{k})\mathbf{k}\big)\|^2 |F(\boldsymbol{\omega})|^2 d\boldsymbol{\omega} \tag{10.18}$$

Because $e(\mathbf{k})$ is the square norm of a function in a Hilbert space,

$$e(\mathbf{k}) = \|\big(\boldsymbol{\omega} - (\boldsymbol{\omega}^T\mathbf{k})\mathbf{k}\big)F\|^2 = 0 \quad \Rightarrow \quad \big(\boldsymbol{\omega} - (\boldsymbol{\omega}^T\mathbf{k})\mathbf{k}\big)F = \mathbf{0} \tag{10.19}$$

Consequently, when $e(\mathbf{k}) = 0$, the expression $\big(\boldsymbol{\omega} - (\boldsymbol{\omega}^T\mathbf{k})\mathbf{k}\big)F$ equals (the vector-valued function) zero. Assuming $F \neq 0$, this will happen if and only if F equals zero outside the line $t\mathbf{k}$. A nontrivial F can thus be nonzero only on the line $t\mathbf{k}$, which, according to lemma 10.1, means that f must be linearly symmetric. Conversely, if F is zero except on the line $t\mathbf{k}$, the corresponding $e(\mathbf{k})$ vanishes. To summarize, if and only if $e(\mathbf{k}) = 0$, all spectral energy, $|F|^2$, is concentrated to a line and f is linearly symmetric with the direction $\pm\mathbf{k}$. Leaving the question whether or not $\pm\mathbf{k}$ always translates to an unambiguous direction aside (we will discuss this further below), we proceed to the more urgent question of how to calculate $\mathbf{k}$.

Noting that $\boldsymbol{\omega}^T\mathbf{k}$ is a scalar that is indistinguishable from the scalar $\mathbf{k}^T\boldsymbol{\omega}$, and $\|\mathbf{k}\|^2 = \mathbf{k}^T\mathbf{k} = 1$, the square magnitude distance can be written in quadratic form:

$$d^2(\boldsymbol{\omega}, \mathbf{k}) = \mathbf{k}^T\big(\mathbf{I}\boldsymbol{\omega}^T\boldsymbol{\omega} - \boldsymbol{\omega}\boldsymbol{\omega}^T\big)\mathbf{k} \tag{10.20}$$

$$= \mathbf{k}^T\left[\begin{pmatrix} \omega_x^2 + \omega_y^2 & 0 \\ 0 & \omega_x^2 + \omega_y^2 \end{pmatrix} - \begin{pmatrix} \omega_x^2 & \omega_x\omega_y \\ \omega_x\omega_y & \omega_y^2 \end{pmatrix}\right]\mathbf{k} \tag{10.21}$$

Defining the components of $\boldsymbol{\omega} = (\omega_x, \omega_y)^T$ as

$$\omega_x = \omega_1, \qquad \text{and} \qquad \omega_y = \omega_2, \tag{10.22}$$

for notational convenience, Eq. (10.17) is expressed as

$$e(\mathbf{k}) = \mathbf{k}^T\mathbf{J}\mathbf{k} = \mathbf{k}^T(\mathbf{I} \cdot \text{Trace}(\mathbf{S}) - \mathbf{S})\mathbf{k} \tag{10.23}$$

where $\mathbf{I}$ is the identity matrix,

$$\mathbf{J} = \mathbf{I} \cdot \text{Trace}(\mathbf{S}) - \mathbf{S} \tag{10.24}$$

and

$$\mathbf{S} = \int \boldsymbol{\omega}\boldsymbol{\omega}^T |F|^2 d\boldsymbol{\omega} = \begin{pmatrix} \mathbf{S}(1,1) & \mathbf{S}(1,2) \\ \mathbf{S}(2,1) & \mathbf{S}(2,2) \end{pmatrix}, \quad \text{with} \quad \mathbf{S}(i,j) = \int \omega_i \omega_j |F(\boldsymbol{\omega})|^2 d\boldsymbol{\omega} \tag{10.25}$$

Definition 10.2. *The matrix* $\mathbf{S}$ *in Eq. (10.25), which consists of the second-order moments of the power spectrum,* $|F|^2$*, is called the* structure tensor *of the image* f*.*

The matrix $\mathbf{J}$ is the *inertia tensor* of the power spectrum using a term of mechanics. The matrix $\mathbf{S}$ is also called the scatter tensor of the power spectrum in statistics. The structure tensor can be readily obtained from $\mathbf{J}$, and vice versa via Eq. (10.24). There is another related tensor called the *covariance tensor* or the covariance matrix in statistics, $\mathbf{C} = \mathbf{S} - \mathbf{m} \cdot \mathbf{m}^T$, where $\mathbf{m} = \int \boldsymbol{\omega} |F|^2 d\boldsymbol{\omega}$. However, for real images $\mathbf{m} = 0$, since $|F|$ is even when f is real. Because of the tight relationship between the notions inertia, scatter, and covariance, they are used in an interchangeable manner in many contexts. Since different notions of the structure tensor coexist, the following lemma, which establishes the equivalence of $\mathbf{J}$ and $\mathbf{S}$ (and of $\mathbf{C}$), is useful to remember.

Lemma 10.2. *With eigenvalue, eigenvector pairs of* $\mathbf{J}$ *being* $\{\lambda_1, \mathbf{u}_1\}$ *and* $\{\lambda_2, \mathbf{u}_2\}$*, and those of* $\mathbf{S}$ *being* $\{\lambda_1', \mathbf{u}_1'\}$ *and* $\{\lambda_2', \mathbf{u}_2'\}$*, we have*

$$\{\lambda_1', \mathbf{u}_1'\} = \{\lambda_1, \mathbf{u}_2\}, \qquad \textit{and} \qquad \{\lambda_2', \mathbf{u}_2'\} = \{\lambda_2, \mathbf{u}_1\}. \tag{10.26}$$

♦

The eigenvector with a certain eigenvalue in the first matrix is an eigenvector with the *other* eigenvalue in the second matrix. The lemma can be proven by utilizing Eq. (10.24) and operating with $\mathbf{J}$ on $\mathbf{u}_i'$, which is assumed to be an eigenvector of $\mathbf{S}$:

$$\mathbf{J}\mathbf{u}_i' = (\mathbf{I} \cdot \text{Trace}(\mathbf{S}) - \mathbf{S})\mathbf{u}_i' = (\lambda_1 + \lambda_2)\mathbf{u}_i' - \lambda_i \mathbf{u}_i' \qquad i = 1, 2 \tag{10.27}$$

The error minimization problem formulated in Eq. (10.17) is reduced to a minimization of a quadratic form, $\mathbf{k}^T \mathbf{J} \mathbf{k}$ with the matrix $\mathbf{J}$ given by Eq. (10.24). This is in turn minimized by choosing $\mathbf{k}$ as the least eigenvector of the inertia matrix, $\mathbf{J}$ [231]. All eigenvalues of $\mathbf{J}$ are real and nonnegative because the error expression Eq. (10.17) is real and nonnegative. Calling the eigenvalue and eigenvector pairs of $\mathbf{J}$ $\{\lambda_{\min}, \mathbf{k}_{\min}\}$ and $\{\lambda_{\max}, \mathbf{k}_{\max}\}$, the minimum of $e(\mathbf{k})$ will occur at $e(\mathbf{k}_{\min}) = \lambda_{\min}$. In other words, the matrix $\mathbf{J}$, or equivalently $\mathbf{S}$, contains sufficient information to allow the computation of the optimal $\mathbf{k}$ in the TLS error sense. We will discuss the motivation behind this choice of error in some detail in Sect. 10.10.

The matrix $\mathbf{S}$ is defined in the frequency domain, which is inconvenient, particularly if $\mathbf{S}$ must be estimated numerous times. For example, when computing the direction for all local patches of an image, we would need to perform numerous Fourier transformations if we attempt to directly estimate the structure tensor from its definition. We can, however, eliminate the need for a Fourier transformation by utilizing (Parseval–Plancherel) theorem 7.2 which states that the scalar products are

conserved under the Fourier transform. Applying it to Eq. (10.28), the computation of the matrix elements will be lifted from the Fourier domain to the spatial domain:

$$\mathbf{S}(i,j) = \int \omega_i \omega_j |F(\boldsymbol{\omega})|^2 d\boldsymbol{\omega} = \frac{1}{4\pi^2} \int \frac{\partial f}{\partial x_i} \frac{\partial f}{\partial x_j} d\mathbf{x} \qquad i,j:1,2 \tag{10.28}$$

where $x_1 = x$, $x_2 = y$, $d\mathbf{x} = dxdy$, and the integral is a double integral over the entire 2D plane. This is rephrased in matrix form,

$$\mathbf{S} = \int \boldsymbol{\omega}\boldsymbol{\omega}^T |F|^2 d\boldsymbol{\omega} = \frac{1}{4\pi^2} \int (\nabla f)(\nabla^T f) d\mathbf{x} \tag{10.29}$$

where

$$\nabla f = \Big(D_x f, D_y f \Big)^T = \Big(\frac{\partial f(\mathbf{r})}{\partial x}, \frac{\partial f(\mathbf{r})}{\partial y} \Big)^T. \tag{10.30}$$

We summarize our finding on 2D direction estimation via the following theorem, where the integrals are double integrals taken over the 2D spatial domain.

Theorem 10.1 (Structure tensor I). *The extremal inertia axes of the power spectrum, $|F|^2$ are determined by the eigenvectors of the* structure tensor*:*

$$\mathbf{S} = \frac{1}{4\pi^2} \int (\nabla f)(\nabla^T f) d\mathbf{x} \tag{10.31}$$

$$= \frac{1}{4\pi^2} \begin{pmatrix} \int (D_x f)^2 d\mathbf{x} & \int (D_x f)(D_y f) d\mathbf{x} \\ \int (D_x f)(D_y f) d\mathbf{x} & \int (D_y f)^2 d\mathbf{x} \end{pmatrix} \tag{10.32}$$

The eigenvalues $\lambda_{\min}$, $\lambda_{\max}$ and the corresponding eigenvectors $\mathbf{k}_{\min}$, $\mathbf{k}_{\max}$ of the tensor represent the minimum inertia, the maximum inertia, the axis of the maximum inertia, and the axis of the minimum inertia of the power spectrum, respectively.

♦

We note that $\mathbf{k}_{\min}$ is the least eigenvector, but it represents the axis of the maximum inertia. This is because the inertia tensor $\mathbf{J}$ is tightly related to the scatter tensor $\mathbf{S}$ according to lemma 10.2. The two tensors share eigenvalues in 2D, although the correspondence between the eigenvalues and the eigenvectors is reversed.

While the major eigenvector of $\mathbf{S}$ fits the minimum inertia axis to the power spectrum, the image itself does not need to be Fourier transformed according to the theorem. The eigenvalue $\lambda_{\max}$ represents the largest inertia or error, which is achieved with the inertia axis having the direction $\mathbf{k}_{\min}$. The worst error is useful too, because it indicates the scale of the error when judging the size of the smallest error, $\lambda_{\min}$ (the *range problem*). By contrast, the axis of the maximum inertia provides no additional information, because it is always orthogonal to the minimum inertia axis as a consequence of the matrix $\mathbf{S}$ being symmetric and positive semidefinite.

10.4 The Complex Representation of the Structure Tensor

Estimating the structure tensor, and thereby the direction of an image, can be simplified further by utilizing the algebraic properties of the complex $\mathbf{z}$-plane. Multiplication and division are well-defined in complex numbers, whereas conventional

vectors cannot be multiplied or divided with each other to yield new vectors. As a result, an explicit computation of the matrix eigenvalues will become superfluous, and the structure tensor will be automatically decomposed into a directional and a nondirectional part.

Central to the structure tensor theory is the maximization of the scatter $\mathbf{k}^T\mathbf{S}\mathbf{k}$. Because $\mathbf{S}$ is a positive semidefinite matrix with real elements, i.e., $\mathbf{S} = \mathbf{A}^T\mathbf{A}$ for some $\mathbf{A}$ having real coefficients, the scatter is $(\mathbf{A}\mathbf{k})^T(\mathbf{A}\mathbf{k})$ and is either zero or positive for *any* real vector $\mathbf{k}$. By incorporating the complex conjugation into transposition, i.e., $\mathbf{B}^H = (\mathbf{B}^*)^T$, the Hermitian transposition, the expression $(\mathbf{A}\mathbf{k})^H(\mathbf{A}\mathbf{k}) = \mathbf{k}^H\mathbf{S}\mathbf{k}$ will be either zero or positive even if the vector $\mathbf{k}$ has been expressed in a basis that has complex elements. Here, we will maximize $\mathbf{k}^H\mathbf{S}\mathbf{k}$, assuming $\mathbf{k}$ may have complex elements. First, we introduce a new basis that has complex vector elements, using the unitary matrix: $\mathbf{U}^H$

$$\mathbf{k}' = \mathbf{U}^H\mathbf{k}, \quad \text{where} \quad \mathbf{U}^H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & i \\ i & 1 \end{pmatrix}, \quad \text{and} \quad U = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & -i \\ -i & 1 \end{pmatrix}. \tag{10.33}$$

Unitary matrices generalize orthogonal matrices in that a unitary matrix has in general complex elements, and it obeys the relationship $\mathbf{U}^H\mathbf{U} = \mathbf{U}\mathbf{U}^H = \mathbf{I}$. Consequently,

$$\mathbf{k}^H\mathbf{S}\mathbf{k} = (\mathbf{U}\mathbf{k}')^H\mathbf{S}\mathbf{U}\mathbf{k}' = \mathbf{k}'^H\mathbf{Z}\mathbf{k}' \tag{10.34}$$

where

$$\begin{aligned} \mathbf{Z} &= \mathbf{U}^H\mathbf{S}\mathbf{U} \qquad (10.35)\\ &= \frac{1}{2}\begin{pmatrix} \mathbf{S}(1,1)+\mathbf{S}(2,2) & -i(\mathbf{S}(1,1)-\mathbf{S}(2,2)+i2\mathbf{S}(1,2)) \\ i(\mathbf{S}(1,1)-\mathbf{S}(2,2)+i2\mathbf{S}(1,2))^* & \mathbf{S}(1,1)+\mathbf{S}(2,2) \end{pmatrix} \end{aligned}$$

We call $\mathbf{Z}$ the *complex structure tensor*, and we can conclude that it represents the same tensor as $\mathbf{S}$, except for a basis change, and that both matrices have common eigenvalues. They share also eigenvectors, but only up to the unitary transformation, so that $\mathbf{k}_\mathbf{Z}$ representing an eigenvector of $\mathbf{Z}$ is given by $\mathbf{k}_\mathbf{Z} = \mathbf{U}^H\mathbf{k}_\mathbf{S}$, with $\mathbf{k}_\mathbf{S}$ being an eigenvector of $\mathbf{S}$. We define the elements of $\mathbf{Z}$ via the complex quantities I_{20} and I_{11} as follows.

Definition 10.3. *The matrix*

$$\mathbf{Z} = \frac{1}{2}\begin{pmatrix} I_{11} & -iI_{20} \\ iI_{20}^* & I_{11} \end{pmatrix}, \quad \textit{where} \quad \begin{aligned} I_{20} &= \mathbf{S}(1,1)-\mathbf{S}(2,2)+i2\mathbf{S}(1,2) \\ I_{11} &= \mathbf{S}(1,1)+\mathbf{S}(2,2) \end{aligned} \tag{10.36}$$

is the complex representation of the structure tensor.

A matrix $\mathbf{Z}$ is called Hermitian if $\mathbf{Z}^H = \mathbf{Z}$, and if additionally $\mathbf{k}^H\mathbf{Z}\mathbf{k} \geq 0$, which is the case by definition for the complex structure tensor, is called Hermitian positive semidefinite. With the above representation, the elements of $\mathbf{Z}$ encode the $\lambda_{\max}$, $\lambda_{\min}$ as well as $\mathbf{k}_{\max}$ more explicitly than $\mathbf{S}$. This is summarized in the following theorem [28], the proof of which is found in Sect. 10.17.

Theorem 10.2 (Structure tensor II). *The minimum and the maximum inertia as well as the axis of minimum inertia of the power spectrum are given by*

$$I_{20} = (\lambda_{\max} - \lambda_{\min}) e^{i2\varphi_{\min}} = \frac{1}{4\pi^2} \int \overline{\neg}(f)(x,y) d\mathbf{x} \tag{10.37}$$

$$I_{11} = \lambda_{\max} + \lambda_{\min} = \frac{1}{4\pi^2} \int |\overline{\neg}(f)(x,y)| d\mathbf{x} \tag{10.38}$$

with the infinitesimal linear symmetry tensor *(ILST) defined as*[2]

$$\overline{\neg}_{x,y}(f)(x,y) = \left(\frac{\partial f}{\partial x} + i \frac{\partial f}{\partial y} \right)^2 \tag{10.39}$$

The quantities $\lambda_{\min}$, $\lambda_{\max}$, and $\varphi_{\min}$ are, respectively, the minimum inertia, the maximum inertia, and the axis of the minimum inertia of the power spectrum.

♦

The eigenvalues of the tensor in theorem 10.1 and the λ's appearing in this theorem are identical. Likewise, the direction of the major eigenvector $\mathbf{k}_{\max}$ and the $\varphi_{\min}$, of theorem 10.2 coincide. Accordingly, the eigenvector information is encoded explicitly in an offdiagonal element of $\mathbf{Z}$, i.e., I_{20} whereas the sum and the difference of the eigenvalues are encoded in the diagonal element as I_{11} and in the offdiagonal element as $|I_{20}|$, respectively.

For completeness, we provide the eigenvectors of $\mathbf{Z}$ as well. Because the argument angle of I_{20} is twice the direction angle of $\mathbf{k}_{\max}$, the direction of latter is obtained by the direction of the square root of I_{20}. The eigenvectors of $\mathbf{S}$ are related to the eigenvectors of $\mathbf{Z}$ via Eq. (10.33), so that we obtain

$$\mathbf{k}'_{\max} = \gamma(\sqrt{I_{20}}, i\sqrt{I^*_{20}})^T \tag{10.40}$$

$$\mathbf{k}'_{\min} = \gamma(\sqrt{-I_{20}}, i\sqrt{-I^*_{20}})^T \tag{10.41}$$

where γ is a scalar that normalizes the norms of the vectors to 1.

Summary of the complex structure tensor

- **Independence under merging.** Averaging the "square" of the complex field $(D_x f + iD_y f)^2$ and its magnitude (scalar) field $|D_x f + iD_y f|^2$, automatically fits an optimal axis to the spectrum,
- **Schwartz inequality.** The inequality $|I_{20}| \leq I_{11}$ holds with equality if and only if the image is linearly symmetric,
- **Rotation-invariance and covariance.** If the image is rotated, the absolute values of the elements of $\mathbf{Z}$, i.e., $|I_{20}|$ as well as I_{11}, will be invariant to the rotation, while the argument of I_{20} will change linearly with the rotation.

In the next two sections we discuss two simpler tensors that will be used as basis tensors for decomposing the structure tensor.

[2] The symbol $\overline{\neg}$ is pronounced as "doleth" or "daleth", which is intended to be a mnemonic for the fact that it is not an ordinary gradient delivering a vector constisting of derivatives but is a (complex) *scalar* comprised of derivatives.

10.5 Linear Symmetry Tensor: Directional Dominance

In this section we will mean the complex structure tensor when we refer to the structure tensor. An "ideal" linear symmetry is present in the image, when $\lambda_{\max} >> 0$ and $\lambda_{\min} = 0$. Such images have a *directional dominance* in that there is a single and well-identified direction of isocurves. One way to quantitate this property is by measuring $\lambda_{\max} - \lambda_{\min}$ and $\mathbf{k}_{\max}$, which are jointly given by the complex scalar I_{20} [28]. When $\lambda_{\max} - \lambda_{\min}$ increases, so does the evidence for the image being linearly symmetric, and hence we have a crisp direction in the image. For Hermitian[3] positive semi definite matrices, which includes the structure tensor $\mathbf{Z}$, the dimension of the eigenvector space is equal to the multiplicity of the corresponding eigenvalue, which is the number of times the latter is repeated. The eigenvectors are orthogonal if they belong to two different eigenvalues. Because there are at most two different eigenvalues in 2D, there is no need to encode both eigenvectors. The linear symmetry quality of a 2D image has also been called the "line" property [225], and the "stick" property [161]. The *linear symmetry tensor* is a special type of structure tensor such that

$$\mathbf{Z}_L = \frac{1}{2}\begin{pmatrix} |I_{20}| & -iI_{20} \\ iI_{20}^* & |I_{20}| \end{pmatrix} \tag{10.42}$$

The tensor is fully equivalent to the scalar quantity I_{20}, which determines $\mathbf{Z}_L$ uniquely, which is, in turn, a spatial average of $\urcorner(f)$:

10.6 Balanced Direction Tensor: Directional Equilibrium

Certain images lack direction, i.e., when a direction is attempted to be fit to the power spectrum there is not one optimal axis but there are an infinite (uncountable) number of them, e.g., the image of sand in Fig. 10.13 or the image in Fig. 10.10. This property is captured by the structure tensor via the condition $\lambda_{\max} = \lambda_{\min}$, i.e., the smallest (or the largest) eigenvalue is repeated twice making its multiplicity 2. The condition actually does not describe the presence of a property but the lack of it. It describes the lack of linear symmetry. An image with a structure tensor having $\lambda_{\max} = \lambda_{\min}$, has previously been called "perfectly balanced", in analogy with the terminology used in mechanics [28]. The term expresses that there is a *directional equilibrium* in that no single direction dominates over the others. Such images lack a single direction[4] that is more significant than other directions, a property that justifies the use of the term "balanced directions" or "balancedness", both referring to an equilibrium of directions. Balancedness is quantitated by $\lambda_{\min}$ because, for a fixed $\lambda_{\max}$, the larger $\lambda_{\min}$, the closer it gets to $\lambda_{\max}$. When $\lambda_{\min}$ reaches its upper bound, which is $\lambda_{\max}$, then the structure tensor has one eigenvalue which has a multiplicity 2. The least eigenvalue $\lambda_{\min}$ can be used to signal the presence of a balanced image,

[3] We recall that a Hermitian matrix $\mathbf{Z}$ fulfills $\mathbf{Z} = \mathbf{Z}^H$.

[4] The pattern may still have a group direction, although it may lack a single direction dominating others, e.g., see Fig. 10.10.

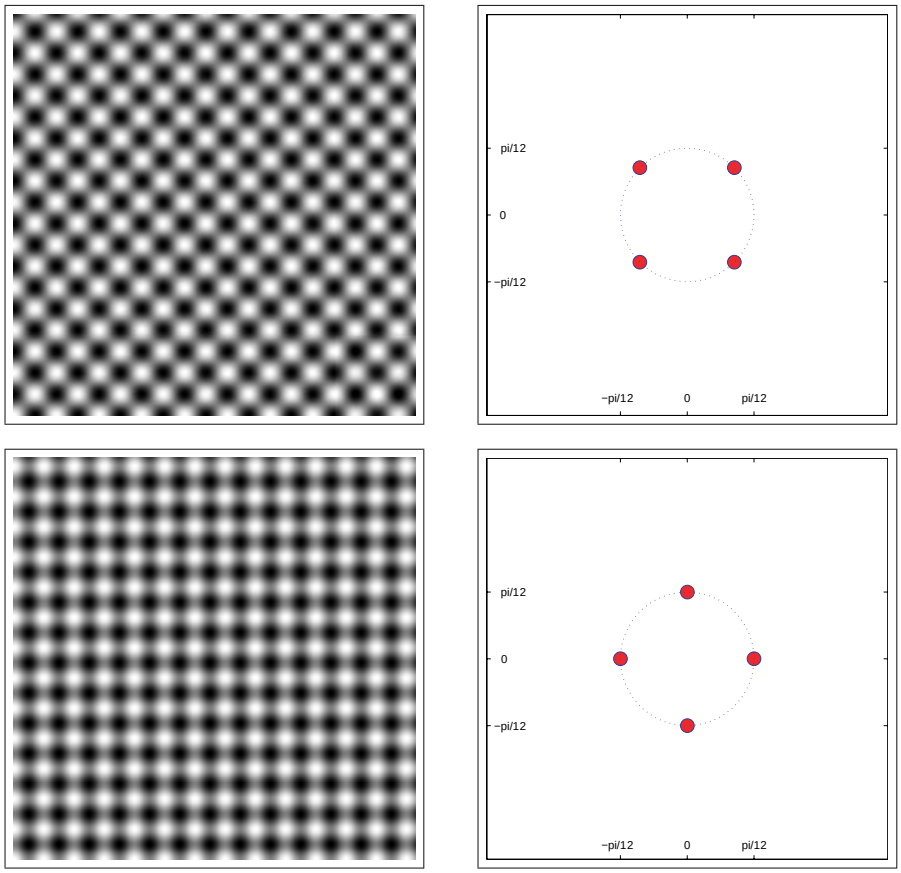

Fig. 10.10. (*Top*) A perfectly balanced image f, and its power spectrum, $|F|^2$. The *red circles* represent a concentration of the power. (*Bottom*) The same as on top, except that the image is rotated with the angle $\frac{\pi}{4}$

the state of equilibrium of directions when both eigenvalues are equal. Accordingly, the *balanced direction tensor* is given by

$$\mathbf{Z}_B = \frac{1}{2}\begin{pmatrix} I_{11} & 0 \\ 0 & I_{11} \end{pmatrix} \tag{10.43}$$

which is a special case of the complex structure tensor. The tensor is completely equivalent to the scalar quantity I_{11}, which determines $\mathbf{Z}_B$ uniquely which is in turn an average of the magnitude of $\neg(f)$. This tensor has also been called "isotropic" [225] and being "ball-like" [161] in other studies. The term isotropic should not be interpreted as an existence of all directions in the image is a necessity. Figure 10.10 shows two images that contain only two directions and yet they are qualified for the term. Likewise the term ball tensor should not be interpreted too restrictively such

as the image must look like a ball, or a junction. The images in Fig. 10.10 as well as in Fig. 10.13 represent textures in which at every point there is a "ball" tensor of approximately the same magnitude. We will discuss the use of balanced direction tensor as a corner detector in Sect. 10.9.

10.7 Decomposing the Complex Structure Tensor

In general, an image is neither perfectly linearly symmetric, e.g., Figs. 10.2-10.6, nor does it totally lack it, e.g., Fig. 10.13. Instead it has the qualities of both types. The amount of evidence for the respective type can be obtained from the structure tensor. The *structure tensor decomposition* can always be achieved into its linear symmetry and balanced direction components easily using its complex form:

$$\mathbf{Z} = \frac{1}{2}\begin{pmatrix} I_{11} & -iI_{20} \\ iI_{20}^* & I_{11} \end{pmatrix} = \mathbf{Z}_L + \mathbf{Z}_B \tag{10.44}$$

where

$$\mathbf{Z}_L = \frac{1}{2}\begin{pmatrix} |I_{20}| & -iI_{20} \\ iI_{20}^* & |I_{20}| \end{pmatrix}, \quad \text{and} \quad \mathbf{Z}_B = \frac{1}{2}\begin{pmatrix} I_{11} - |I_{20}| & 0 \\ 0 & I_{11} - |I_{20}| \end{pmatrix}. \tag{10.45}$$

Conversely, we also wish to study what happens when joining regions having different structure tensors. Without loss of generality, we consider a composition of a region consisting of two subregions, each having a different structure tensor, $\mathbf{Z}'$ and $\mathbf{Z}''$, respectively. This is a realistic scenario since two neighboring regions in an image might differ in their local structure tensors, and the local structure tensor at a border point between the two regions is needed. Because the components of the structure tensor are integrals, they can be computed as the sum of two integrals, each taken over the respective regions. Accordingly, the structure tensor, $\mathbf{Z}$, of a boundary point is obtained by the addition

$$\mathbf{Z} = p\mathbf{Z}' + q\mathbf{Z}'' \tag{10.46}$$

where p, q are two real positive scalars, with $p + q = 1$, that are proportional to the areas of the two constituent regions. Following the definition of $\mathbf{Z}$, we obtain

$$I_{20} = pI'_{20} + qI''_{20} \tag{10.47}$$
$$I_{11} = pI'_{11} + qI''_{11} \tag{10.48}$$

where $I'_{..}$, $I''_{..}$, and $I_{..}$ are the structure tensor parameters of the first, the second and the joint patches.

Example 10.6. *In Fig. 10.11 we have two regions labelled A, and B. There are four local images, called images here, and these are marked as $1, \cdots, 4$ with their borders shown as (color) circles. Let images 2 and 4 have the (complex) structure tensors $\mathbf{Z}'$ and $\mathbf{Z}''$, respectively. The corresponding tensor components are therefore*

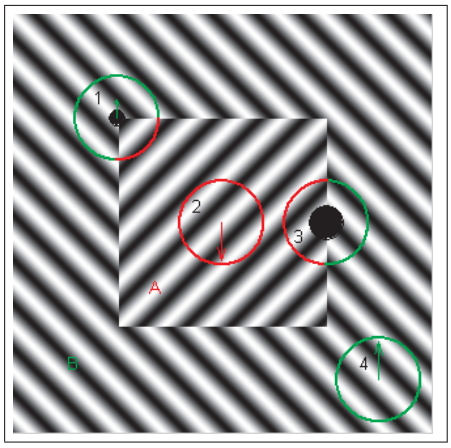

Fig. 10.11. Illustration of addition using the complex structure tensor. The linear symmetry tensor components I_{20} are shown as vectors for four local images. The balanced direction components, I_{11}, are shown as *circles filled with black*. The limits of images are marked by *color circles*

$$I'_{20} = \mathbf{k}'^2 = \exp(-i\tfrac{\pi}{2}) \qquad I''_{20} = \mathbf{k}''^2 = \exp(i\tfrac{\pi}{2})$$
$$I'_{11} = 1 \qquad I''_{11} = 1$$

In these images, the linear symmetry components point at directions given by $\mathbf{k}^2$, *where,* $\mathbf{k}$ *is the direction of the respective gradient (Fig. 10.12, left). The balanced direction components are equal to zero, because both images are linearly symmetric so that* $I_{11} = |I_{20}|$. *In image 3 we have the structure tensor* $\mathbf{Z} = \frac{1}{2}\mathbf{Z}' + \frac{1}{2}\mathbf{Z}''$, *having the components*

$$I'_{20} = \tfrac{1}{2}\exp(-i\tfrac{\pi}{2}) + \tfrac{1}{2}\exp(i\tfrac{\pi}{2}) = 0$$

$$I'_{11} = \tfrac{1}{2} + \tfrac{1}{2} = 1$$

The linear symmetry component is zero, as it should be. The image is a perfectly balanced image because none of its constituent directions dominates the others. The balanced direction tensor element is, by contrast, $I_{11} - |I_{20}| = 1$, *which indicates that all spectral power is distributed in such a way that the directions balance each other perfectly. Conceptually, balanced image phenomenon is present also when the gradient directions are random (Fig. 10.12, right). Likewise in image 1 we have the structure tensor* $\mathbf{Z} = \frac{1}{4}\mathbf{Z}' + \frac{3}{4}\mathbf{Z}''$ *having the components*

$$I'_{20} = \tfrac{1}{4}\exp(-i\tfrac{\pi}{2}) + \tfrac{3}{4}\exp(i\tfrac{\pi}{2}) = \tfrac{1}{2}\exp(i\tfrac{\pi}{2})$$

$$I'_{11} = \tfrac{1}{4} + \tfrac{3}{4} = 1$$

In particular, the balanced direction component, $I_{11} - |I_{20}| = \frac{1}{2}$, *should be contrasted to the magnitude of the linear symmetry component* $|I_{20}| = \frac{1}{2}$. *The argument of* I_{20}

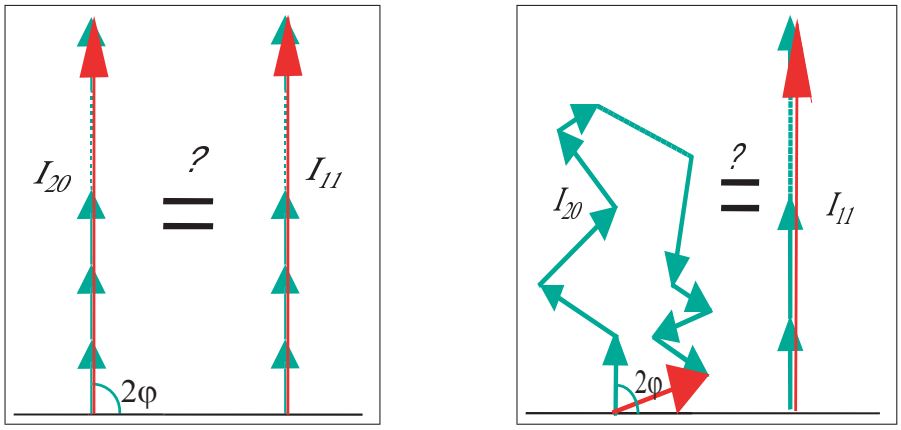

Fig. 10.12. The linear symmetry component, I_{20}, and the balanced directions component, I_{11}, of the structure tensor add as vectors and as scalars, independently. The addition of numerous structure tensors (*left*) for a merging of local images sharing a common direction, (*right*) for a merging that largely lacks such a common direction .

shows the dominant direction of the image. We paraphrase this result as follows. When the account of directions is finalized there is an excess of a single direction that is not balanced by other directions. The net excess of this dominant direction is as significant as the directions that are balanced.

10.8 Decomposing the Real-Valued Structure Tensor

Using its real form, $\mathbf{S}$, the *structure tensor decomposition* in terms of linear symmetry and balanced direction components is also possible [161, 225]. This is done by the spectral decomposition of the tensor $\mathbf{S}$

$$\mathbf{S} = \lambda_{\max}\mathbf{k}_{\max}\mathbf{k}_{\max}^T + \lambda_{\min}\mathbf{k}_{\min}\mathbf{k}_{\min}^T \tag{10.49}$$

followed by the rearrangement:

$$\begin{aligned}\mathbf{S} &= (\lambda_{\max} - \lambda_{\min})\mathbf{k}_{\max}\mathbf{k}_{\max}^T + \lambda_{\min}(\mathbf{k}_{\max}\mathbf{k}_{\max}^T + \mathbf{k}_{\min}\mathbf{k}_{\min}^T) \\ &= (\lambda_{\max} - \lambda_{\min})\mathbf{k}_{\max}\mathbf{k}_{\max}^T + \lambda_{\min}\mathbf{I}\end{aligned} \tag{10.50}$$

where the fact that $\mathbf{k}_{\max}\mathbf{k}_{\max}^T + \mathbf{k}_{\min}\mathbf{k}_{\min}^T = \mathbf{I}$, when $\mathbf{k}_{\max}$ and $\mathbf{k}_{\min}$ are orthogonal, has been used. The first term of Eq. (10.50) is the linear symmetry tensor, and the second term is the balanced direction tensor in real matrix representation. When merging two images, each consisting of balanced directions, the result is an image that is perfectly balanced. Accordingly, adding the balanced direction components of two arbitrary images will not result in a change of the linear symmetry components. However, if two linearly symmetric images are merged the result usually has

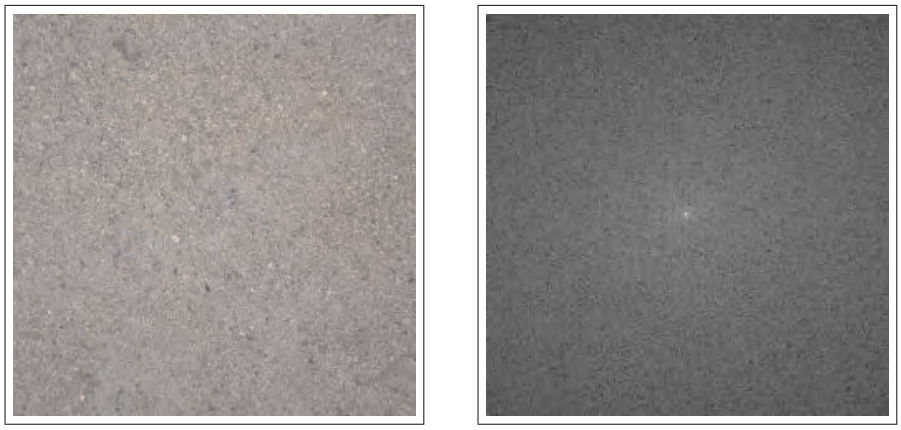

Fig. 10.13. *Left*: a real image that is not linearly symmetric. It shows a close up view of sand. Right: The Fourier transform magnitude of a neighborhood in the central part of the original image. Notice that the power is isotropic i.e., far from being concentrated to a line

a structure tensor that has a balanced direction component unless both images have the same direction. Accordingly, the linear symmetry components do not add in a straightforward fashion in the real matrix representation, causing an "interaction" contribution to the balanced direction component. In effect, when merging two arbitrary images already decomposed into their component tensors, the linear symmetry components add (matrix addition) first according to Eq. (10.50), possibly producing a balanced direction term. This additional term should then be added to the sum of the ordinary balanced direction tensor components of the two images.

The decomposition of the complex structure tensor is conserved and closed under averaging whereas that of the real structure tensor is not. Paraphrasing, *averaging linear symmetry tensors* in their complex form yields linear symmetry tensors, whereas averaging linear symmetry tensors in their real form may produce tensors that are not linearly symmetric. This is because (i) the two components I_{20} and I_{11} add separately when merging or smoothing images, and (ii) these components are explicitly linked to eigenvalues and optimal directions.

10.9 Conventional Corners and Balanced Directions

By using other algebraic functions of $\lambda_{\max}$ and $\lambda_{\min}$, numerous measures to quantitate the amount of linear symmetry can be obtained. Likewise, we can also measure the lack of symmetry, which is the balanced directions property of an image.

Example measures include the energy invariant measure, C_{f2} [28], for linear symmetry,

$$C_{f2} = \frac{|I_{20}|}{I_{11}} = \frac{\lambda_{\max} - \lambda_{\min}}{\lambda_{\max} + \lambda_{\min}} \qquad \text{(linear symmetry)} \tag{10.51}$$

which also immediately defines the energy-invariant measure for the *lack of linear symmetry* C_{f3}:

$$C_{f3} = 1 - \frac{|I_{20}|}{I_{11}} \qquad \text{(balanced directions)} \tag{10.52}$$

At the heart of such measures is how well the Schwartz inequality, $|I_{20}| \leq I_{11}$, is fulfilled. The case of equality happens if and only if one has linear symmetry ($|I_{20}| = I_{11}$). The left-hand side of it vanishes and the right-hand side becomes as large as the energy permits it, if and only we have balanced directions ($0 = |I_{20}|$). Having this in mind, then other functionals that measure the distance $I_{11} - |I_{20}|$ can be used to quantitate the balancedness of the directions in the image. The popular detector of Harris and Stephen [97] (a similar measure is that of Forstner and Gulch [74]) used to measure cornerness, quantitates this distance as well

$$\begin{aligned} C_{\text{hs}} &= \lambda_{\max}\lambda_{\min} - 0.04(\lambda_{\max} + \lambda_{\min})^2 && (10.53)\\ &= (I_{11} + |I_{20}|)(I_{11} - |I_{20}|)/4 - 0.04I_{11}^2 \\ &= (0.84I_{11}^2 - |I_{20}|^2)/4 && (10.54) \end{aligned}$$

albeit in the quadratic scale, which is most obvious if the empirical constant 0.84 is replaced by 1. Because of the constant, the measure C_{hs} must be combined with a threshold to reject the negative values. This will happen at (local) images that have only the linear symmetry component (e.g., on lines and edges) where $I_{11} = |I_{20}|$, yielding $C_{\text{hs}} = -0.04|I_{20}| < 0$. The measure C_{hs} responds strongly to many corner types, including a corner that consists of the junction of two orthogonal directions, or a corner that consists of the intersection of several lines. A word of caution is in place because C_{hs} will also respond strongly to other patterns, including at every point in a texture image that lacks direction. This may be a desirable property for an application at hand. However, it is also possible that the application is actually unintentionally accepting (false acceptence) many patterns as corners by using C_{f3} or C_{hs}. The texture images shown in Fig. 10.10 are perfectly balanced everywhere, meaning that every point is a "*balanced directions corner*" or "*Stephen–Harris corner*". Likewise, all boundary points, except the boundary corners, between region A and region B in Fig. 10.11 are the strongest corners in either of the two corner senses above. All points of these four lines are, in fact, stronger "corners" than the four boundary corner points, as discussed in Sect. 10.7!

10.10 The Total Least Squares Direction and Tensors

It is in place to ask what makes the matrices $\mathbf{J}$, $\mathbf{S}$, or even $\mathbf{Z}$ (second-order) tensors. We recall that the basic difference between a second-order tensor and a matrix is subtle and lies in that a tensor represents a physical quantity on which the coordinate system has no real influence except for a numerical representation. The numerical representation of a tensor is then a matrix that corresponds to physical measurements

in a specific basis. A representation of the same tensor in a different basis can only be obtained by a similarity transformation using the basis transformation matrix. For first-order tensors and vectors, a similar subtle difference exists. The first-order tensor is represented as a vector in a specific basis. Another representation of it can be obtained by a linear transformation corresponding to a basis change. The zero-order tensors represent physical quantities that are scalars. Their numerical representations do not change with basis transformations.

As illustrated by Fig. 10.9, the error function $e(\mathbf{k})$ employed by the total least square (TLS) error represents the spectral power weighted by its shortest (orthogonal) distance to the estimated axis $\mathbf{k}$. This makes e a zero-order tensor and $\mathbf{k}$ a first-order tensor. If we apply a basis change e.g., rotate the coordinates of the power spectrum, $e(\mathbf{k})$ will not change at all and only the numerical representation of $\mathbf{k}$ will change. The new direction vector, $\mathbf{k}'$, will be coupled to the old $\mathbf{k}$ linearly using the inverse of the matrix that caused the basis change.

To appreciate the TLS error in this context we compare it to the *mean square* (MS) error which is extensively used in applications where one has a black box controlled by known inputs resulting in a measurable output. In such applications there is thus a *response measurement*, $\mathbf{y}$, that may contain measurement errors and that is to be explained by means of another set of known (error-free) variables, called *explanatory variables* $\mathbf{X}$, via a linear model

$$\mathbf{y} = \mathbf{X}\mathbf{k} \tag{10.55}$$

Here $\mathbf{k}$ is the unknown regression parameter, which will be estimated by minimizing the following residual:

$$\min_{\mathbf{k}} \|\mathbf{y} - \mathbf{X}\mathbf{k}\|^2 \tag{10.56}$$

Adapted to our 2D direction estimation problem, the MS error yields:

$$\min_{\gamma} e(\gamma) = \int \|\omega_y - \gamma\omega_x\|^2 |F(\omega_x, \omega_y)|^2 d\boldsymbol{\omega} \tag{10.57}$$

This is the classical *regression problem*. Here, the direction coefficient γ is unknown and will be estimated from the data $F(\omega_x, \omega_y)$. The unknown γ is related to the direction vector $\mathbf{k} = (\cos\theta, \sin\theta)^T$ as $\gamma = -\frac{\cos\theta}{\sin\theta}$. Notice that the integrand measures the distance between the data point $\boldsymbol{\omega}$ and a point on the $\mathbf{k}$-axis to be fitted. This distance is in general not the shortest, distance as illustrated by the vector $\mathbf{d}$ in Fig. 10.14. The MS error would accordingly depend on the coordinate axis directions to the effect that after a basis change, the new error using the same data will be different. Likewise, the new direction $\mathbf{k}'$ will not be given by multiplying the inverse of the basis transformation matrix with $\mathbf{k}$, the estimated direction before the basis change. In consequence, neither the MS error nor γ are tensors. One can associate $\mathbf{k}$ and $\mathbf{k}\mathbf{k}^T$ to every $\gamma = -\frac{\cos(\theta)}{\sin(\theta)}$. These quantites are not tensors, although they are conventional vectors and matrices. A more detailed discussion of the TLS error can be found in [115] and [59].

Through Taylor expansion a spatial interpretation of $e(\mathbf{k})$, as an alternative to its original interpretation, the spectral inertia, can be obtained. In this view, the structure

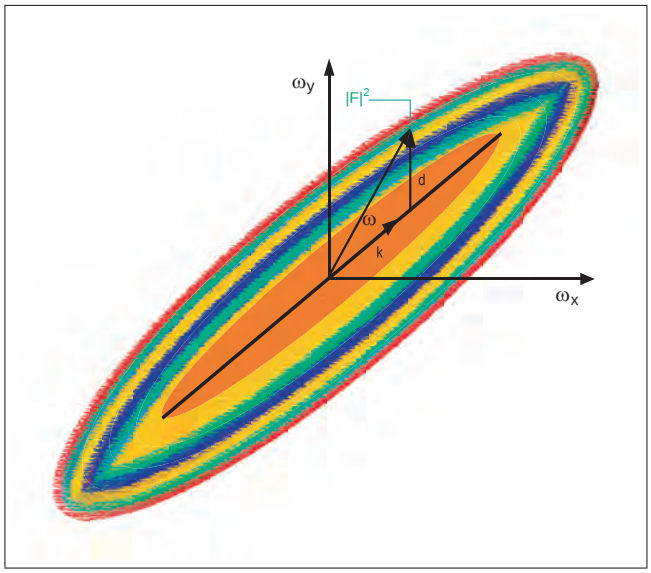

Fig. 10.14. The error, $\|\mathbf{d}\|^2$, used in the MS estimate. The error is not measured as the shortest distance between a frequency coordinate $\boldsymbol{\omega}$ and the $\mathbf{k}$-axis. This should be contrasted with the TLS error, which does measure the shortest distance, as shown in Fig. 10.9

tensor, via its minor eigenvector, encodes the direction in which a small translation of the image causes it the least departure from the original. To see this, we perform the expansion, i.e., we express the image f at $\mathbf{r}+\epsilon\mathbf{k}$, where the direction $\mathbf{k} = (k_x, k_y)^T$ has the unit length, by using the partial derivatives of f at $\mathbf{r}$

$$f(\mathbf{r}+\epsilon\mathbf{k}) = f(\mathbf{r})+\epsilon\big(k_xD_x+k_yD_y\big)f(\mathbf{r})+\frac{\epsilon^2}{2}\big(k_xD_x+k_yD_y\big)^2f(\mathbf{r})+\cdots \quad (10.58)$$

Accordingly,

$$\epsilon\big(k_xD_x + k_yD_y\big)f(\mathbf{r}) = f(\mathbf{r}+\epsilon\mathbf{k}) - f(\mathbf{r}) \quad (10.59)$$

is the linear approximation of the difference between the function $f(\mathbf{r})$ and its translated version, $f(\mathbf{r}+\epsilon\mathbf{k})$. In consequence,

$$\big(\big(k_xD_x + k_yD_y\big)f(\mathbf{r})\big)^2 \quad (10.60)$$

is the magnitude of the rate of the change in the direction of $\mathbf{k}$, which can be viewed as the error rate or resistance rate when translating the image in the direction $\mathbf{k}$. Integrating this function and using the (Parseval–Plancherel) theorem 7.2, yields

$$\iint \big(\mathbf{k}^T\nabla f(x,y)\big)^2\,dxdy = \mathbf{k}^T\left(\iint \nabla f(x,y)\nabla^T f(x,y)dxdy\right)\mathbf{k} = \mathbf{k}^T\mathbf{S}\mathbf{k} \quad (10.61)$$

which is the dynamic part of our original error function, $e(\mathbf{k})$, because

$$e(\mathbf{k}) = \text{Trace}(\mathbf{S}) - \mathbf{k}^T\mathbf{S}\mathbf{k} \tag{10.62}$$

Accordingly, minimizing Eq. (10.62) yields the direction of *minimum translation error*. Evidently maximizing Eq. (10.62) yields the direction of *maximum translation error*. Both minimum and maximum error directions are given by the eigenvectors of $\mathbf{S}$. Paraphrased, the structure tensor encodes the *minimum resistance direction* in the spatial domain, which is identical to the direction of the line fit to the power spectrum in the TLS sense.

10.11 Discrete Structure Tensor by Direct Tensor Sampling

Until this section, the theory for detection of the orientation of a scalar function in 2D space has been based on continuous signals. One such technique was summarized by theorem 10.1 which we will attempt to approximate by use of discrete functions. We call this approach *direct tensor sampling* since the suggested method examines whether or not the spectrum of an image consists of a line, by directly estimating the matrix $\mathbf{S}$, with the elements given by Eq. (10.28) without first estimating the power spectrum by a discrete local spectrum.

We need to approximate the continuous integrand of Eq. (10.28) from a discrete image. To that end, we need the approximation of

$$\frac{\partial f(\mathbf{r})}{\partial x_i}\frac{\partial f(\mathbf{r})}{\partial x_j,} \qquad \text{with} \qquad i,j = 1,2, \quad x_1 = x, \quad \text{and} \quad x_2 = y, \tag{10.63}$$

on a Cartesian grid i.e., $\mathbf{r} = \mathbf{r}_l$, where $\mathbf{r}_l$ is the coordinates of the grid nodes. In analogy with the theory presented in Sects. 8.2, 8.3, and 9.2, we can do this by filtering the original image linearly:

$$\frac{\partial f(\mathbf{r}_l)}{\partial x_i} = \sum_k f(\mathbf{r}_l + \mathbf{r}_k)\frac{\partial \mu(\mathbf{r}_l)}{\partial x_i} \qquad \text{with} \qquad i = 1,2 \tag{10.64}$$

where $\frac{\partial \mu(\mathbf{r}_l)}{\partial x_i} = \frac{\partial \mu(\mathbf{r})}{\partial x_i}|_{\mathbf{r}=\mathbf{r}_l}$, $\frac{\partial f(\mathbf{r}_l)}{\partial x_i} = \frac{\partial f(\mathbf{r})}{\partial x_i}|_{\mathbf{r}=\mathbf{r}_l}$, and then applying pointwise multiplication between the two thus-obtained discrete partial derivative images:

$$\frac{\partial f(\mathbf{r}_l)}{\partial x_i}\frac{\partial f(\mathbf{r}_l)}{\partial x_j}, \qquad \text{with} \qquad i,j = 1,2, \quad x_1 = x, \quad \text{and} \quad x_2 = y. \tag{10.65}$$

The latter is an estimate of Eq. (10.63) on a Cartesian grid. Note that the continuous form of Eq. (10.63) is not known, but we estimated nevertheless its discrete version by applying a linear discrete filtering to the discrete $f(\mathbf{r}_l)$, followed by a pointwise multiplication on the grid $\mathbf{r}_l$.

To estimate the structure tensor elements, Eq. (10.28), we first reconstruct (10.63) from its samples (10.65):

$$\frac{\partial f(\mathbf{r})}{\partial x_i}\frac{\partial f(\mathbf{r})}{\partial x_j} = \sum_l \frac{\partial f(\mathbf{r}_l)}{\partial x_i}\frac{\partial f(\mathbf{r}_l)}{\partial x_j}\mu(\mathbf{r} - \mathbf{r}_l) \qquad i,j : 1,2 \tag{10.66}$$

where $x_1 = x$, $x_2 = y$. The vectors $\mathbf{r}_l$ represent points on a grid as before, and $\mu(\mathbf{r})$ is the continuous interpolation function, assumed to be a Gaussian with the variance σ_p^2:

$$\mu(\mathbf{r} - \mathbf{r}_l) = \exp\big(-\frac{1}{2\sigma_p^2}\|\mathbf{r} - \mathbf{r}_l\|^2\big) \tag{10.67}$$

We proceed by substituting (10.66) in Eq. (10.28)

$$\mathbf{S}(i,j) = \frac{1}{4\pi^2}\sum_l \frac{\partial f(\mathbf{r}_l)}{\partial x_i}\frac{\partial f(\mathbf{r}_l)}{\partial x_j}\int_{E_2}\mu(\mathbf{r} - \mathbf{r}_l)dxdy \tag{10.68}$$

$$= \frac{C}{4\pi^2}\sum_l \frac{\partial f(\mathbf{r}_l)}{\partial x_i}\frac{\partial f(\mathbf{r}_l)}{\partial x_j} \tag{10.69}$$

Here the integral evaluates to a constant C

$$\int_{E_2}\mu(\mathbf{r} - \mathbf{r}_l)dxdy = C$$

that is independent of $\mathbf{r}_l$ because the area under a shifted Gaussian is the same regardless of the amount of the shift. The summation in Eq. (10.69) is taken over all image points, $\mathbf{r}_l$, on the grid.

However, the structure tensor is most frequently needed for a local image, rather than the entire image. A simple way to achieve this goal is to approximate $\frac{\partial f}{\partial x_i}\frac{\partial}{\partial x_j}$ for the local image by multiplying its global version with a window function, $w(\mathbf{r})$, placed around the current point $\mathbf{r}_0$. Without loss of generality, we assume that the local image for which the structure tensor is to be estimated is the one around the origin. Using a Gaussian as a window function, this amounts to replacing μ in Eq. (10.66) with

$$\mu(\mathbf{r} - \mathbf{r}_l)w(\mathbf{r}) = \exp\left(-\frac{1}{2\sigma_p^2}\|\mathbf{r} - \mathbf{r}_l\|^2\right)\exp\left(-\frac{1}{2\sigma_w^2}\|\mathbf{r}\|^2\right) \tag{10.70}$$

where σ_w^2 is a constant that controls the effective width of the second Gaussian, the window defining the local image around the origin. Substituting this into Eq. (10.66) and then using it in Eq. (10.28) yields the local structure tensor elements:

$$\mathbf{S}(i,j) = \frac{1}{4\pi^2}\sum_l \frac{\partial f(\mathbf{r}_l)}{\partial x_i}\frac{\partial f(\mathbf{r}_l)}{\partial x_j}\int_{E_2}\mu(\mathbf{r} - \mathbf{r}_l)w(\mathbf{r})dxdy \tag{10.71}$$

$$= \frac{1}{4\pi^2}\sum_l \frac{\partial f(\mathbf{r}_l)}{\partial x_i}\frac{\partial f(\mathbf{r}_l)}{\partial x_j}\mu_l \tag{10.72}$$

where μ_l is defined as

$$\mu_l = \int_{E_2}\exp\left(-\frac{1}{2\sigma_p^2}\|\mathbf{r} - \mathbf{r}_l\|^2\right)\exp\left(-\frac{1}{2\sigma_w^2}\|\mathbf{r}\|^2\right)dxdy$$

$$= (\mu * w)(\mathbf{r}_l) = \frac{2\pi}{4}\frac{\sigma_p^2\sigma_w^2}{\sigma_p^2 + \sigma_w^2}\exp\big(-\frac{1}{2(\sigma_p^2 + \sigma_w^2)}\|\mathbf{r}_l\|^2\big). \tag{10.73}$$

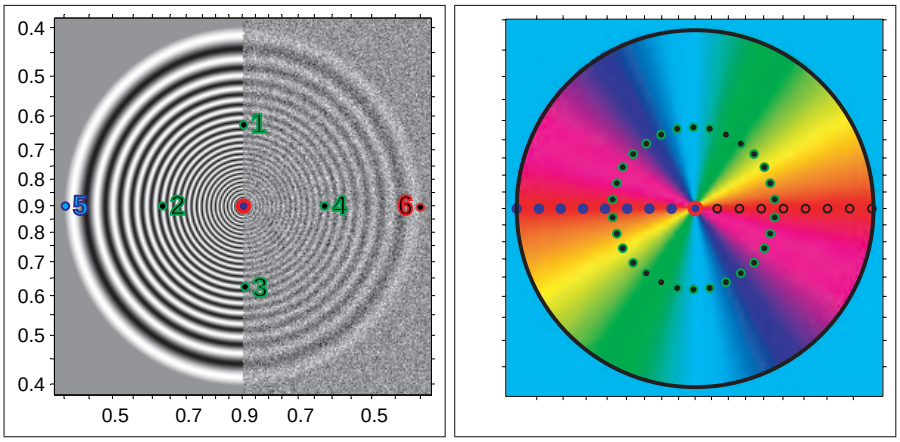

Fig. 10.15. On the *left*, the test image, the axes of which are marked with fractions of π representing the spatial frequency. On the *right* is the color code representing the directions. The marks on the axes are separated by 5 degrees when joined to the center. The *colored dots* in both images define the curves from which 1D direction measurements will be sampled

The integral represents a continuous convolution, and Eq. (10.73) is obtained by noting that both μ and w are Gaussians and that a convolution of them yields another Gaussian, with a variance that is the sum of the variances of μ and w. An easy way of seeing this is by applying the Fourier transform to $\mu * w$. Eq. (10.72), which computes the local tensor around the origin, is therefore a discrete convolution by a Gaussian if $\mathbf{S}(i,j)$ needs to be computed for local patches around *all* points of the original image grid. Since the values of μ_ls decrease rapidly outside of a circle with radius $\sqrt{\sigma_p^2 + \sigma_w^2}$, we can truncate the infinite filter when its coefficients are sufficiently small, typically when the coefficients have decreased to about 1% of the filter maximum. Thus, Eq. (10.72) implies that the local tensor (of the origin) is obtained as a window-weighted average of the gradient outer products:

$$\mathbf{S} = \frac{1}{4\pi^2} \sum_l (\nabla f_l)(\nabla f_l)^T \mu_l \tag{10.74}$$

where ∇f_l is the gradient of $f(\mathbf{r})$ at the discrete image position $\mathbf{r}_l$, and μ_l is a discrete Gaussian. Defining $D_x f_l$ and $D_y f_l$, for convenience, as the components of ∇f_l, at the mesh point $\mathbf{r}_l$:

$$\nabla f_l = (D_x f_l, D_y f_l)^T = (\frac{\partial f(\mathbf{r}_l)}{\partial x}, \frac{\partial f(\mathbf{r}_l)}{\partial y})^T, \tag{10.75}$$

We summarize our finding on tensor discretization as a theorem:

Theorem 10.3 (Discrete structure tensor I). *Assuming a Gaussian interpolator with σ_p and a Gaussian window with σ_w, the optimal* discrete structure tensor *approximation is given by*

$$\mathbf{S} = C \sum_l (\nabla f_l \nabla^T f_l) \mu_l \tag{10.76}$$

$$= C \sum_l \begin{pmatrix} (D_x f_l)^2 & (D_x f_l)(D_y f_l) \\ (D_x f_l)(D_y f_l) & (D_y f_l)^2 \end{pmatrix} \mu_l \tag{10.77}$$

where μ_l is a discrete Gaussian with $\sigma = \sqrt{\sigma_p^2 + \sigma_w^2}$.

♦

In analogy with Eqs. (10.63)–(10.65), the quantity

$$D_x f(\mathbf{r}_l) + i D_y f(\mathbf{r}_l) \tag{10.78}$$

can be obtained by two convolutions using real filters, one for $D_x f(\mathbf{r}_l)$ and one for $D_y f(\mathbf{r}_l)$. After that, the complex result depicted by Eq. (10.78) is squared to yield the ILST image:

$$\urcorner(f)(\mathbf{r}_l) = (D_x f(\mathbf{r}_l) + i D_y f(\mathbf{r}_l))^2 \tag{10.79}$$

In consequence of theorem 10.2, the following theorem then holds true:

Theorem 10.4 (Discrete structure tensor II). *Assuming a Gaussian interpolator with σ_p and a Gaussian window with σ_w, the optimal* discrete structure tensor *complex elements are given by*

$$\mathbf{Z} = \frac{1}{2} \begin{pmatrix} I_{11} & -iI_{20} \\ iI_{20}^* & I_{11} \end{pmatrix} \tag{10.80}$$

where

$$I_{20} = C \sum_l (D_x f_l + i D_y)^2 \mu_l \tag{10.81}$$

$$I_{11} = C \sum_l |D_x f_l + i D_y|^2 \mu_l \tag{10.82}$$

with μ_l being a discrete Gaussian with $\sigma = \sqrt{\sigma_p^2 + \sigma_w^2}$.

♦

Figure 10.15 shows a frequency-modulated test (FM-test) image. The test image has axes marked by the spatial frequencies of the waves in the horizontal and vertical directions from the image center. The absolute frequency of the waves decreases exponentially radially, whereas the direction of the waves changes uniformly angularly. The exponential decrease occurs between the spatial frequencies 0.4π and 0.9π. The image on the right represents the color code of the ideal orientation in double-angle representation, i.e., $\exp 2\varphi$, where φ is the polar angle coordinate of a point in the image. In half of the image, spatially uncorrelated Gaussian noise X, with mean 0.5 and variance 1/36, has been added to the image signal, f, according to $pf + (1-p)X$, where the weight coefficient $p = 0.3$, and X is the noise. On

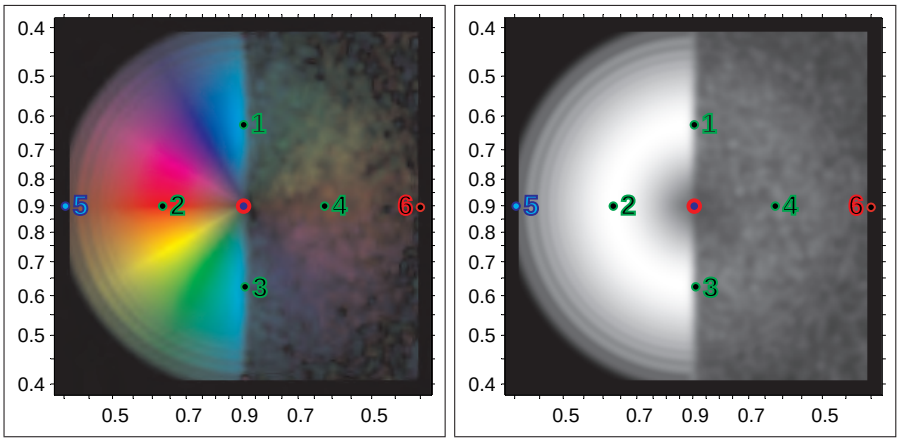

Fig. 10.16. The images illustrate the direction tensor, represented as I_{20} (*left*) and I_{11} (*right*), for Fig. 10.15 and computed by using theorem 10.4. The hue encodes the direction, whereas the brightness represents the magnitudes of complex numbers

the right the reference orientation is encoded as a color image. Using the HSB color space, the hue is modulated by 2φ, whereas the brightness and the saturation are set to the maximum at all points. The colored dots define curves along which the structure tensor measurements will be extracted and discussed in detail further below. The color coded reference image has axes marked by angles separated in 5° as seen from the center.

The images in Fig. 10.16 illustrate the structure tensor computed for all local images. The color image on the left represents the complex-valued I_{20}. The hue of an image point is modulated by the $\arg(I_{20})$ of its local neighborhood, whereas its brightness is given by $|I_{20}|$. The saturation of all points is set to the maximum. The computations are implemented according to theorem 10.4 where the derivative Gaussian filter had $\sigma_p = 0.8$ and the integrative Gaussian filter, $\sqrt{\sigma_p^2 + \sigma_w^2} = 2.5$. The hue of a point should be the same as the reference color given at the same point of the color image in Fig. 10.16. Visually, it appears that the colors are in good conformity with those of the reference image. The image on the right shows I_{11}, which, being nonnegative and real-valued, modulates the gray values. It is possible to verify that the direction measurements adhere to the theoretical values, even in the noisy part of the test image, where the signal-to-noise ratio, (SNR), was $2\log_2(\frac{0.3}{0.7}) \approx -2.4$ dB. The brightness of the points at the noisy part are lower in both images for two reasons. First, the $|I_{20}|$ should be weak or ideally zero because the unique direction is disturbed, and second, because the linear derivation and integration operations have effectively a bandpass character and the noise components are suppressed by the linear process. In the clean part of the signal, the brightness of the corresponding points in the left and the right images is the same. We will discuss these conclusions more quantitatively next.