

Computing Oriented Texture Fields

A. RAVISHANKAR RAO*

I.B.M., T. J. Watson Research Center, Yorktown Heights, New York 10598

AND

BRIAN G. SCHUNCK

Artificial Intelligence Laboratory, University of Michigan, Ann Arbor, Michigan 48109-2110

Received June 12, 1989; accepted June 20, 1990

The first step in the analysis of oriented texture consists of the extraction of an orientation field. The orientation field is comprised of the angle and coherence images, which describe at each point the dominant local orientation and degree of anisotropy, respectively. A new algorithm for computing the orientation field for a flow-like texture is presented. The basic idea behind the algorithm is to use an oriented filter, namely the gradient of Gaussian, and perform manipulations on the resulting gradient vector field. The most important aspect of the new algorithm is that it is provably optimal in estimating the local orientation of an oriented texture. An added strength of the algorithm is that it is simpler and has a better signal-to-noise ratio than previous approaches, because it employs fewer derivative operations. We also propose a new measure of coherence, which works better than previous measures. The estimates for orientation and coherence are related to measures in the statistical theory of directional data. We advocate the use of the angle and coherence images as intrinsic images. An analysis of oriented textures will require the computation of these intrinsic images as a first step. In this sense, the computation of the orientation field, resulting in the intrinsic images, is indispensable in the analysis of oriented textures. We provide results from several experiments to indicate the usefulness of the angle and coherence intrinsic images. These results show that the notion of scale plays an important role in the interpretation of textures. Further, measures defined on these intrinsic images are useful for the inspection of surfaces. © 1991 Academic Press, Inc.

1. INTRODUCTION

Texture had been an active area for research for over 2 decades [1, 2]. There are several areas like petrography [3], metallography [4], and lumber processing [5] that make extensive use of textural features such as grain

* Support provided by the Semiconductor Research Corporation under Contract 86-07-085.

shapes, sizes, and distributions for recognizing and analyzing specimens. Texture is very important in quality control since many inspection decisions are based on the appearance of the texture of some material. There are many different kinds of texture, and these have been classified in the form of a taxonomy in [6].

This paper deals with visual textures that are comprised of flow-like patterns such as wood grain. These textures are characterized by local selectivity of orientation, which can vary arbitrarily over the entire image. In other words, the texture is anisotropic. Every point in the image is associated with a dominant local orientation, and a local measure of the coherence or degree of anisotropy of the flow pattern. One way of visualizing oriented textures is to think about the image intensity surface as being comprised of ridges, whose direction and height can vary continuously. More formally, we define the *orientation field* of a texture image to be comprised of two images, called *angle image* and *coherence image*. The angle image captures the dominant local orientation at each point in the texture in terms of an angle, and the coherence image represents the degree of anisotropy at each point in the texture. The term *orientation field* is used because it is closely related to a velocity flow field, where at each point in space, a fluid element can have some velocity, which is composed of a magnitude and direction. However, the analogy is not complete, as we shall show later. The major contributions of this paper are to present optimum methods for the computation of the angle and coherence images, and to present the results of several experiments that illustrate the importance of these images in different contexts.

Kass and Witkin [7] have developed an algorithm for estimating the orientation of texture patterns. Their method can be used to decompose the oriented pattern

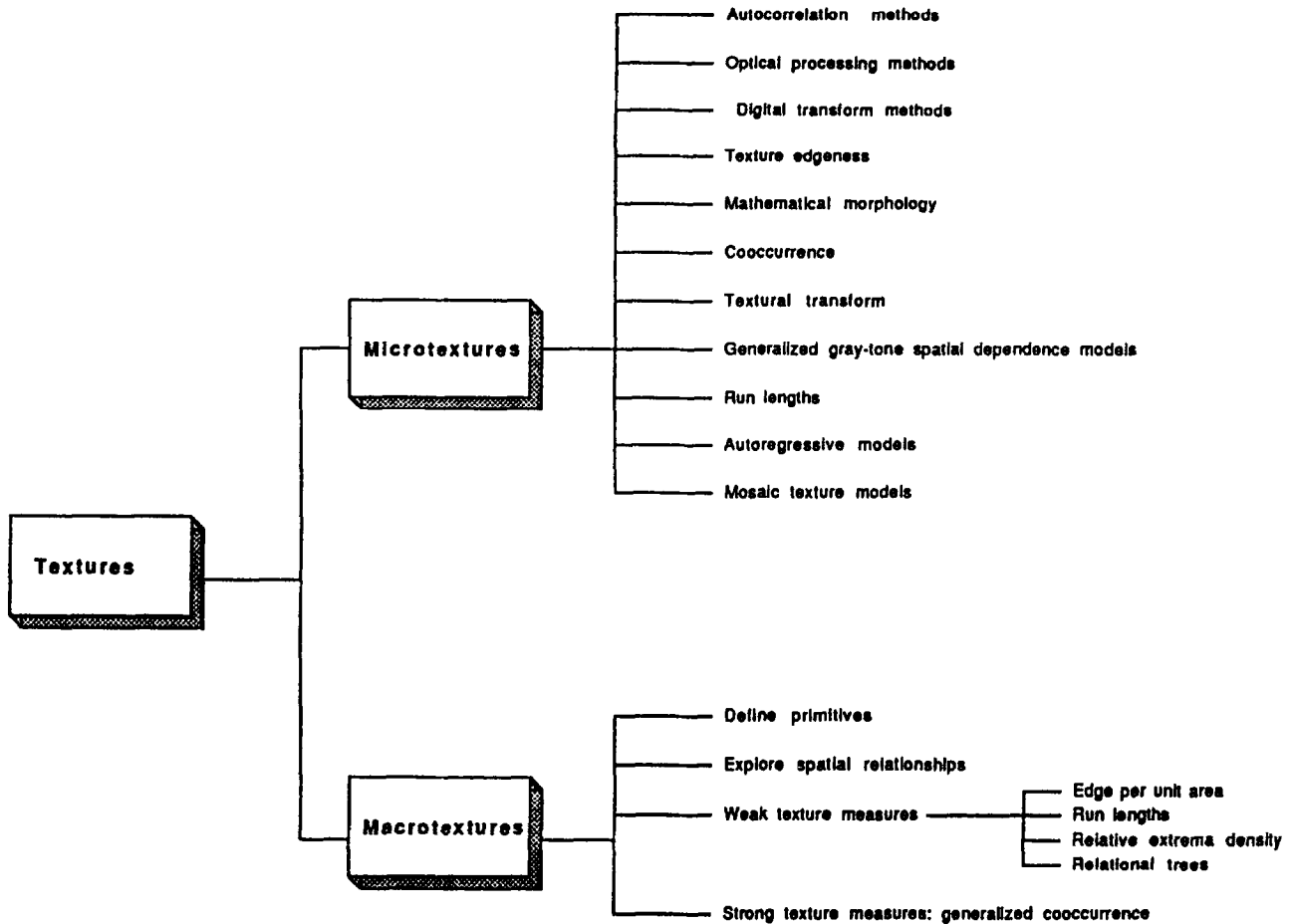


FIG. 1. Different approaches to texture description.

into a flow field, describing the direction of anisotropy, and describing the pattern independent of changing flow direction. This can be used to classify directional textures into classes such as lamellar structure. The algorithm developed by Kass and Witkin [7] for estimating the texture orientation was based on Laplacian-of-Gaussian filters. This paper presents another algorithm based on the gradient of the Gaussian. The most important aspect of the new algorithm is that it relates directly to results concerning the mean direction and dispersion¹ in statistics of directional data [8]. This result is an improvement over Kass and Witkin's scheme [7] as it makes no assumptions about the nature of the oriented texture (such as producing a zero mean filter output after convolving with the Laplacian-of-Gaussian). The new algorithm incorporates a better scheme for computing the coherence of the flow field. Furthermore, some attractive features of the new algorithm are that it requires less computation and has better signal-to-noise characteristics since it incorporates fewer derivative operations.

¹ Section 4.5 defines these terms.

2. BACKGROUND

There are three broad categories of texture problems [9, p. 424]. The first involves the identification and description of two-dimensional patterns, and has been widely addressed [1, 2]. Figure 1 summarizes the different approaches to texture description, as reviewed in [1]. The second is concerned with using texture as a means to perform segmentation of an image [10–12]. The third problem is to use texture as a cue to retrieve information about surface orientation and depth [13]. The problem we address in this paper falls in the first category, since our primary objective is to obtain the orientation field from the texture. Based on the philosophy of Barrow and Tenenbaum [14], the orientation field may be viewed as an intrinsic image, and represents the first computational stage in analyzing oriented textures. The orientation field is computed using domain-independent processing, and constitutes a more abstract description of the image than raw intensity values. In the final section, we show how predicates defined on the intrinsic image give rise to meaningful results in appropriate domains.

Many of the classic approaches to texture identification and description are based on cooccurrence matrices [1]. The cooccurrence matrix is defined as follows. Consider the second-order joint conditional probability density function, $f(i, j|d, \theta)$. For a given θ and d , $f(i, j|d, \theta)$ represents the probability of going from gray level i to gray level j , given that the intersample spacing is d and the direction of the intersample spacing is θ . For a given value of d and θ we can generate a matrix which represents the estimated second-order joint conditional probability density function, which is known as the *gray level cooccurrence matrix* [1]. There are many problems associated with the use of the cooccurrence matrices. First, a cooccurrence matrix must be computed for different values of d and θ , and then the appropriate texture measure must be computed from each cooccurrence matrix. Second, cooccurrence matrices are inappropriate to describe oriented textures since directionality can vary continuously in an oriented texture. This is because the cooccurrence matrix is defined only for *globally* discrete values of θ , and is hence incapable of estimating local variations in orientation. In this paper we present a more powerful tool to analyze oriented textures.

There is considerable literature on visual texture which indicates that there are several kinds of texture that should be considered as separate visual phenomena. Aloimonos and Swain [15, 16] presented an algorithm for determining the orientation of surfaces from the distortion of textures under perspective projection. The algorithm assumes that the scene surface is covered with uniformly repeated texture elements. It would be interesting to explore how the algorithms would work when presented with perspective distortion of texture flow patterns. Coleman and Jain [17] extend the work on shape from shading to include variations in the albedo. Essentially, the variations in albedo form a texture pattern and their shape from shading algorithm compensates for the effect of texture on the shape from shading algorithm.

Julesz and Bergen [18] introduced the notion of textons, which are features that are extracted by the preattentive visual system. Textons are visual primitives such as blobs and terminations from the primal sketch theory of Marr [19] and crossings of line segments. Textons have specific properties such as color or orientation. This paper addresses the extraction of the orientation of primitive features.

Kanatani [20] used integral geometry to develop an algorithm that determines surface orientation from the intersections the image scan lines with curves of texture. In principle, the texture orientation field could be used as a source for the curves of texture used in the work of Kanatani.

Tomita *et al.* [21] presented an algorithm for extracting texture elements and the placement rules from images.

The texture orientation field algorithm presented in this paper could be used as a preprocessing stage to extract the texture pattern. The notion of texture coherence and angle could guide the formulation of the placement rules that define the repetition pattern for the texture elements.

Vilnrotter *et al.* [22] used a similar approach based on computing the repetition of edge elements in a texture. Their work is directed toward computing an interpretation of the texture. They compute edge repetition arrays, which are similar to cooccurrence matrices, from the edge detector outputs. The algorithm presented in this paper does not require the explicit computation of the edge repetition structure as the coherence and angle intrinsic images play a similar role, but the algorithm presented in this paper is restricted to flow-like textures that can be modeled by coherence and angle alone.

Witkin [13] presents an algorithm for computing surface orientation from texture; the possible directions for the tangents of markings on the surface are equally likely, and surface orientation and tangent direction are independent. These are the assumptions of isotropy and independence, without assuming that the texture pattern is regular. The only assumptions are that all possible surface orientations are equally likely. If there is a flow-like texture, then all possible tangent directions are not equally likely. It is still important to run experiments to determine the effect of violating isotropy.

Zucker and Cavanaugh [23] have performed experiments with subjective figures in texture discrimination. The test image is called the Ehrenstein illusion. The two versions of the illusion, called the pattern and the control, should have different texture orientation field structures which could contribute to part of an explanation for the subjective effects. We present results related to the illusion in this paper.

Statistical models for texture were described by Zucker and Terzopoulos [24]. Pentland [12] introduced fractal models for texture. Work has been done to analyze the style or texture of brush strokes in Japanese paintings [25].

Bovik *et al.* [26] have developed a multichannel approach to texture analysis using Gabor filters [27]. Textures are modeled as signals of band-limited spatial frequency. Variations in the amplitude and phase response of the filter are analyzed to segment texture mosaics.

Texture plays a critical role in inspecting surfaces that are produced at various stages in all types of manufacturing [5]. For instance, in the inspection of semiconductor devices, surface texture is an important factor that is used to decide the integrity of a fabricated device [28]. Texture plays an important role in the area of machined parts. In fact, an entire standard [29] is devoted to the specifications related to surface texture. When a metal is deformed, its grains are reoriented, with certain orienta-

tions being preferred over others. This development of a preferred orientation gives rise to a texture. The analysis of such directional textures is important because many material properties, such as tensile strength, depend on the distribution of grain directionality [30, p. 14]. In this paper, we restrict our attention to flow-like patterns, which have several interesting applications.

The research presented in this paper is also part of a larger effort toward texture interpretation [6]. One way to systematize the interpretation and description of textures is to devise a taxonomy for texture. Such a taxonomy has been compiled by Rao [31]. Textures are broadly divided into ordered and disordered textures, based on the presence of repetitive primitive elements and the directionality of the texture. The class of flow-like textures falls under the category of ordered textures, and forms an important component of the taxonomy.

The analysis of flow-like textures can form the basis of a scheme for flow visualization [32]. The method of flow visualization is an important tool in experimental flow mechanics. Computer vision techniques can extract qualitative information about flow behavior which may not be readily available from conventional flow measurements [33]. Image processing techniques have been applied to quantitative flow analysis with promising results [34], and there is still plenty of room for technical improvements in these techniques. Flow visualization techniques are becoming increasingly popular in diverse areas such as biomedical engineering, tracer methods, oceanography, aerodynamics, and surface flows [33]. Subsurface microstructure, consisting largely of flow-like textures, plays a crucial role in the analysis of fractograph specimens [35, pp. 27–35]. In petrography, orientation analysis is helpful in labeling certain textures as lamellar (plate-like) [3, 36].

The orientation estimation algorithm presented in this paper has been successfully used to analyze oriented textures through the geometric theory of differential equations [37, 31]. The basic approach is to locally approximate an orientation field with a linear differential equation. This work illustrates the significance of the orientation field in deriving higher level symbolic descriptors for texture.

3. ORIENTED TEXTURE FIELDS

The algorithm for estimating the orientation of a texture field presented in this paper is based on the gradient of a Gaussian. The approach is similar to the work of Kass and Witkin [7], but involves fewer derivative operations and hence has a better signal-to-noise ratio. Figure 2 is a plot of the magnitude of the Fourier transform of the first derivative of a Gaussian. In two dimensions, the Fourier transform of the gradient of a Gaussian has two lobes oriented along a line passing through the origin in frequency space. The frequency spectrum of a sine-wave

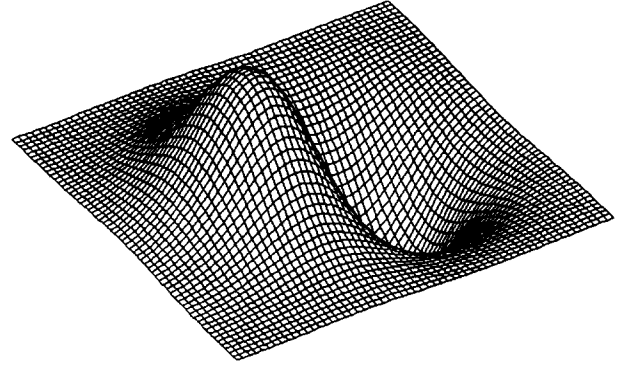


FIG. 2. The Fourier transform of the first derivative of a Gaussian filter is tuned to detect oriented textures at a particular orientation and wavelength.

grating has two δ -functions at equal distances from the origin along a line in frequency space. An oriented pattern will have a dominant frequency component and the response of a gradient of Gaussian can be tuned to this dominant component.

The Gaussian smoothing filter has the impulse response given by

$$g(x) = e^{-x^2/2\sigma^2} \quad (1)$$

and Fourier transform

$$G(\omega) = \sqrt{2\pi} e^{-\omega^2/2\nu^2}, \quad \nu = \frac{1}{\sigma}. \quad (2)$$

The magnitude of the Fourier transform of the first derivative of a Gaussian is

$$|i\omega G(\omega)| = |\omega| \sqrt{2\pi} \sigma e^{-\omega^2/2\nu^2}. \quad (3)$$

The location of the maximum response of the first derivative of a Gaussian is easily determined. Let the Fourier transform of the first derivative of a Gaussian be

$$H(\omega) = \frac{\sqrt{2\pi}}{\nu} \omega e^{-\omega^2/2\nu^2}. \quad (4)$$

The location of the maximum can be determined by setting the derivative of $H(\omega)$ to zero. From this analysis,

$$H'(\omega) = \frac{\sqrt{2\pi}}{\nu} \left[e^{-\omega^2/2\nu^2} - \omega \frac{2\omega}{2\nu^2} e^{-\omega^2/2\nu^2} \right] \quad (5)$$

$$= \frac{\sqrt{2\pi}}{\nu} \left[1 - \frac{\omega^2}{\nu^2} \right] e^{-\omega^2/2\nu^2}. \quad (6)$$

It is easy to see that the maximum response occurs at a distance of $\omega = \nu = 1/\sigma$ from the origin the frequency

space. In two dimensions, the maximum response is oriented along a line passing through the origin in frequency space. If the orientation of the line is θ , then the filter is the derivative of the Gaussian in the direction of θ ; in other words, the normal derivative

$$\frac{dg}{d\hat{n}} = \nabla g \cdot (\cos \theta, \sin \theta). \quad (7)$$

The orientation of the filter is adjusted to achieve the maximum response for the underlying visual texture. In implementation, the orientation of the maximum response is determined by first computing the gradient after smoothing with a Gaussian filter, and then processing the gradient vector field as follows. The orientation of the gradient is the direction of maximum response to within a sign reversal.

There are five steps to estimating the local orientation of the texture field:

1. Smooth the image with a Gaussian filter.
2. Compute the gradient of the smoothed image.
3. Find the local orientation angle using the inverse tangent.
4. Average the local orientation estimates over a neighborhood.
5. Compute a measure of the coherence (the degree of flow-like texture) of the pattern.

The first two steps are standard algorithms in edge detection [38, 39]. These steps are discussed in more detail in Sections 5 and 7. The gradient of Gaussian operator is near-optimal for edge detection, as shown by Canny [38]. This notion of optimality for edge detection can be carried over to the analysis of oriented textures, since the two problems have similar initial processing stages. Kass and Witkin use a Laplacian-of-Gaussian operator as the oriented filter. Clearly, the gradient of Gaussian involves one less differentiation, and will have better performance in the presence of noise.

Since the smoothing of the image must be performed at a given scale, this aspect will be explored in this paper in greater detail. As we shall show, there is no one scale that can be used to describe all textures, and in fact, the choice of scale can dramatically affect the description of a texture. Thus, the choice of scale is dependent to a large extent on what feature sizes one would like to focus on, and this is dictated by prior goals of processing.

The third step can be easily performed using the inverse tangent function. In most edge detection algorithms, the angle of the edge covers the full range of the unit circle; but in estimating the orientation field, it is necessary to reflect orientation vectors that lie along the same line to a canonical orientation. In edge detection, the gradient angle is computed using the arctangent func-

tion of two arguments [39]; but if this function were used to compute the orientation angle in this problem, then there would not be a unique angle for each texture orientation. The texture field orientations would have to be postprocessed to reflect orientation vectors into a canonical range as is apparently done by Kass and Witkin [7]. The texture orientation angle can be computed using the inverse tangent function of one argument which results in angles in the range $(-\pi/2, \pi/2)$, and the representation of texture orientation is unique. This technique is simpler than the rescaling operation that must be performed as part of the orientation estimation algorithm of Kass and Witkin [7], but does not work as well as expected for reasons explained in Section 4.4.

After computing the local orientation of the texture field, the orientation estimate must be smoothed to compute the average orientation over a neighborhood of significant size. In order to perform this, we present a best estimate for the dominant local orientation in Section 4.1. Let σ_1 be the width of the first Gaussian smoothing filter used to compute the local texture orientation and σ_2 be the width of the second smoothing filter used to average the orientation estimate. The averaging filter must be large enough to average the orientation from several of the local estimates and so $\sigma_2 \gg \sigma_1$, but the averaging filter should be smaller than the distance over which the orientation of the texture field undergoes major changes; in other words, the averaging filter should not blur changes in the texture field.

The *coherence* of the oriented texture pattern is the degree to which the local orientation estimates computed before averaging agree. The essential idea in measuring coherence is to project the orientation vectors in some neighborhood onto a representative orientation vector from the neighborhood and normalize the result. If the orientations are coherent, then the normalized projections will be close to one; but if the orientations are not coherent, then the projections will tend to cancel and produce a result close to zero.

4. EXPERIMENTAL METHODS

Test images were digitized using a CCD camera from photographs of textures available in published sources [40, 41]. The image is first smoothed with a Gaussian filter. The sizes of the filter are indicated in the figures. The algorithm for computing the filter coefficients can be found in [39]. To make this operation efficient, we have implemented the convolution in a separable manner [39]. The gradient of the smoothed image is computed using finite differences. Let $G_x(i, j)$ and $G_y(i, j)$ be the x and y components of the gradient vector at point (i, j) . In order to calculate the orientation at a point, one needs to combine the gradient orientations in the neighborhood of that

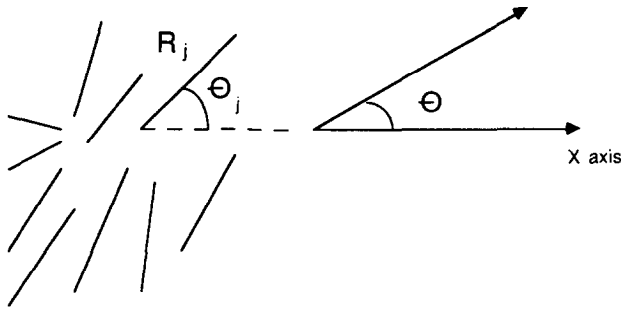


FIG. 3. Illustrating the method to compute the dominant orientation in a group of line segments.

point. As Kass and Witkin [7] pointed out, one cannot smooth the gradient vectors, as they tend to cancel each other out at intensity ridges. There are several methods that one can use to avoid such a cancellation, and we shall present a best estimate for the dominant orientation within a neighborhood.

4.1. A Best Estimate for Dominant Local Orientation

We now show how the dominant direction can be estimated in a 2-D case. A generalization of this proof for the N -dimensional case will be reviewed later. Though such techniques are well known in the statistics of directional data [8], we present considerations that are relevant when adapting the technique for image analysis.

Consider a set of line segments, as shown in Fig. 3. The problem that needs to be solved is, How does one determine the dominant orientation of this set of segments?

One may think that it is sufficient to sum up these line segments vectorially, and find the resultant direction. However, this will not work for two reasons. First, any given line segment does not have a unique direction, since it could be taken to point in either the θ or the $\theta + \pi$ direction. Second, even if the line segments were assigned directions, there is the danger that segments pointing in opposite directions will cancel each other out, instead of influencing the choice of dominant orientation as they should. One way to tackle this problem is as follows.

Assume that the segments are indexed by the subscript i , where i ranges from 1 to N , the number of segments. Consider a line oriented at an angle θ as shown. Let the j th segment subtend an angle θ_j . The next few steps will show that it does not matter what sense this angle is taken in; i.e., it is immaterial as to what vector direction one chooses for the line segment. Let R_j be the length of the j th segment. The projection of this segment onto the line is $R_j \cos(\theta_j - \theta)$. Consider the sum of the absolute value of all such projections,

$$S_1 = \sum_{j=1}^{j=N} \|R_j \cos(\theta_j - \theta)\|. \quad (8)$$

S_1 varies as the orientation of the line θ is varied. That value of θ which maximizes S_1 is the dominant orientation of the given set of line segments. Thus, one can evaluate the dominant orientation by maximizing S_1 with respect to θ . Since the absolute value function is not differentiable everywhere, one can equivalently maximize the sum S_2 , where

$$S_2 = \sum_{j=1}^{j=N} R_j^2 \cos^2(\theta_j - \theta). \quad (9)$$

Here we have taken the sum of the square of the projections. Differentiating Eq. (9) with respect to θ we get

$$\frac{dS_2}{d\theta} = -\sum_{j=1}^{j=N} 2R_j^2 \cos(\theta_j - \theta) \sin(\theta_j - \theta). \quad (10)$$

Equation $dS_2/d\theta$ to zero in order to obtain an extremum, we get from the above equation

$$\sum_{j=1}^{j=N} R_j^2 \sin 2(\theta_j - \theta) = 0 \quad (11)$$

$$\sum_{j=1}^{j=N} R_j^2 \sin 2\theta_j \cos 2\theta = \sum_{j=1}^{j=N} R_j^2 \cos 2\theta_j \sin 2\theta. \quad (12)$$

Hence

$$\tan 2\theta = \frac{\sum_{j=1}^{j=N} R_j^2 \sin 2\theta_j}{\sum_{j=1}^{j=N} R_j^2 \cos 2\theta_j}. \quad (13)$$

Let θ be the value of θ which satisfies Eq. (13). Instead of finding the sum S_2 at different orientations, Eq. (13) tells us in a single computation the orientation θ that maximizes S_2 and is hence the dominant direction of the pattern of line segments. Since θ maximizes S_2 , we call it the *best estimate for dominant local orientation*. That θ indeed maximizes S_2 will be proved shortly.

Equation (13) has an interesting interpretation. Consider the line segments to lie in the complex plane, each segment being represented by $R_j e^{i\theta_j}$, where R_j is the length of the segment and θ_j is its direction. Now square all the segments, which have been represented as complex numbers. Thus each segment will give rise to a term of the form $R_j^2 e^{2i\theta_j}$. If we sum these numbers, the resulting complex number has an orientation α , with respect to the x -axis, given by

$$\tan \alpha = \frac{\sum_{j=1}^{j=N} R_j^2 \sin 2\theta_j}{\sum_{j=1}^{j=N} R_j^2 \cos 2\theta_j}. \quad (14)$$

This equation is the same as Eq. (13).

Using the above interpretation, one can show that θ derived from Eq. (13) indeed maximizes S_2 . In order to show this, one must show that $d^2 S_2/d\theta^2$ is negative at $\theta = \hat{\theta}$. From Eq. (10) we get

$$d^2 S_2/d\theta^2 = -2 \sum_{j=1}^{j=N} R_j^2 \cos(2\theta_j - 2\theta). \quad (15)$$

If we show that the summed quantity is positive we are done. In order to do this, we must assume that the texture has only one dominant local orientation. In this case, the term $R_j^2 \cos(2\theta_j - 2\theta)$ represents the projection of the squared vector onto the line oriented at 2θ . The sum of such projections is positive when the texture has only one dominant local orientation. Thus, we have proved that Eq. (13) provides us with a best estimate of the local orientation within a neighborhood, in the sense that it maximizes the norm in Eq. (9).

Another interesting point of comparison can be made as follows. Equation (13) can be regarded as a smoothing operation on the image, where a box filter (all coefficients identical) is used for smoothing. Kass and Witkin use a Gaussian envelope to perform the smoothing, whereas we have used a box filter for smoothing. We found that the box filter gave better looking results, as shown in Fig. 12.

In order to use Eq. (13) to estimate the orientation at each point in the image, one can consider the gradient vector having components G_x and G_y to represent the line segments of Fig. 3. Consider the vector in the complex plane formed by combining G_x and G_y as $(G_x + iG_y)$. Let the gradient vector at point (m, n) in the image have the polar representation $R_{mn}e^{i\theta_{mn}}$. Thus, the estimate of the dominant orientation θ in an $N \times N$ neighborhood of the image would be given by

$$\theta = \tan^{-1} \left(\frac{\sum_{m=1}^{m=N} \sum_{n=1}^{n=N} R_{mn}^2 \sin 2\theta_{mn}}{\sum_{m=1}^{m=N} \sum_{n=1}^{n=N} R_{mn}^2 \cos 2\theta_{mn}} \right) / 2. \quad (16)$$

The estimated orientation angle at (m, n) is then $\theta_{mn} + \pi/2$, since the gradient vector is perpendicular to the direction of anisotropy. When we refer to the *scale* of the smoothing filter to combine information about the gradient vectors, we mean the size N of the neighborhood over which the estimate is obtained, as in Eq. (16). In all our results we display the estimated orientation angle overlaid on the original image.

4.2. Derivation Using the Moment Method

The derivation presented above is for the two-dimensional case, and is equivalent to the *moment method*, which holds for N dimensions.

Problem. Given vectors $\mathbf{v}_1, \dots, \mathbf{v}_K$, estimate the average orientation when the sign of \mathbf{v}_i is ignored.

Solution. The average orientation is given as the principal axis of the moment tensor

$$\mathbf{M} = \sum_{i=1}^{i=K} \mathbf{v}_i \mathbf{v}_i^T \quad (17)$$

corresponding to the maximum principal value, where $(\cdot)^T$ designates the transpose.

Proof. Let $\bar{\mathbf{v}}$ be the unit vector along the average orientation to be estimated. The projection of vector \mathbf{v}_i along this orientation has a magnitude $|\langle \mathbf{v}_i, \bar{\mathbf{v}} \rangle|$, where (\cdot, \cdot) denotes the inner product. Hence the average orientation $\bar{\mathbf{v}}$ can be estimated by maximizing

$$J = \sum_{i=1}^{i=K} (\mathbf{v}_i, \bar{\mathbf{v}})^2. \quad (18)$$

In terms of the moment tensor \mathbf{M} , this expression becomes

$$J = (\bar{\mathbf{v}}, \mathbf{M}\bar{\mathbf{v}}). \quad (19)$$

This quadratic form in $\bar{\mathbf{v}}$ is maximized under the constraint $|\bar{\mathbf{v}}| = 1$. The solution is given by the eigenvector $\bar{\mathbf{v}}$ of \mathbf{M} corresponding to the maximum eigenvalue, a well known result in linear algebra.

From this result, the following observations can be made. Both $\bar{\mathbf{v}}$ and $-\bar{\mathbf{v}}$ give rise to the same maximum value for J . Similarly, replacing any \mathbf{v}_i by $-\mathbf{v}_i$ will not alter J , as the latter is quadratic in \mathbf{v}_i . Finally, the above proof is independent of the dimensionality of the vector space corresponding to \mathbf{v}_i .

4.3. Squaring the Gradient Vectors: Kass and Witkin's Scheme

We now discuss an alternate method, used by Kass and Witkin [7] in order to smooth the gradient vector field. Consider the vector in the complex plane formed by combining G_x and G_y as $(G_x + iG_y)$. Let this vector have the polar representation $Re^{i\theta}$. The square of this vector is $R^2e^{2i\theta}$. Consider the vector $Re^{i(\theta+\pi)}$, which points opposite to $Re^{i\theta}$. The square of this vector is $R^2e^{2i\theta+2\pi} = R^2e^{2i\theta}$. Hence, squaring gradient vectors that point in opposite directions makes them reinforce each other. This is the basis of the first scheme for combining gradient orientations, and has been used by Kass and Witkin [7].

Let $J(i, j)$ denote the squared gradient vector at (i, j) . The x component of J is $J_x(i, j) = G_x(i, j)^2 - G_y(i, j)^2$ and the y component is $J_y(i, j) = 2G_x(i, j)G_y(i, j)$. J_x and J_y are

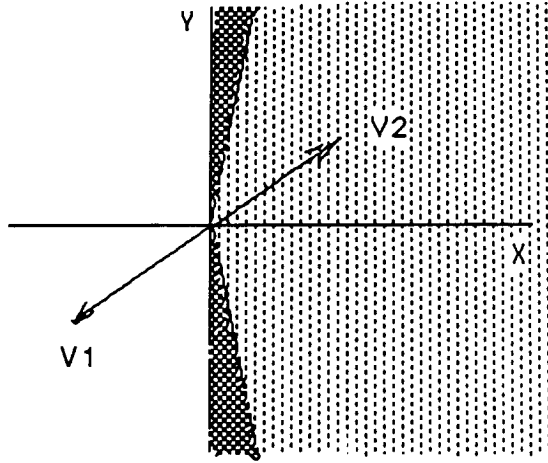


FIG. 4. This figure shows how the arc tangent of one argument can be used to map vectors in the xy -plane onto a half-plane. Thus, V1 gets mapped onto V2. The darkly shaded area represents those regions where vectors can point in opposite directions, and can still cancel each other out.

computed from the gradient of the image in this manner. The next step is to smooth J_x and J_y in order to average the orientation estimates over a neighborhood. This is again done using Gaussian filters, as in the first stage. Let $J_x^*(i, j)$ and $J_y^*(i, j)$ represent the smoothed squared gradient vector at (i, j) . Let θ_{ij} be defined by

$$\theta_{ij} = \tan^{-1} \left(\frac{J_y^*(i, j)}{J_x^*(i, j)} \right) / 2, \quad (20)$$

where the arctangent is computed using two arguments, and lies in the range $[0, 2\pi)$. The division by 2 occurs because the original gradient vector was squared. The estimated orientation angle at (i, j) is then $\theta_{ij} + \pi/2$, since the gradient vector is perpendicular to the direction of anisotropy.

Kass and Witkin arrived at the above result by making several approximating assumptions which veil its importance. First, they use the mean of the variance of the output of the directional filter as an estimate of dominant orientation. Their analysis has been done in a very intuitive manner. However, the derivation presented earlier in this paper employs the maximization of a norm, which is a more accurate representation of the problem at hand. Second, Kass and Witkin assume that the output of the directional filter is zero mean, which is unnecessary, because the derivation in Section 4.1 does not require such assumptions. Finally, Kass and Witkin use a Gaussian envelope to perform a smoothing of the squared gradient vectors with the sole justification that "Gaussian convolutions can be computed efficiently" [7, p. 366]. Again, this is an ad hoc scheme, based on intuition without rigorous justification. In fact, if efficient computation is the sole justification, then a box filter provides the most efficient filter for smoothing.

On the other hand, the only assumption made for the method presented in Section 4.1 is that there is a single dominant direction at each point in the image.

4.4. Inverse Arctangent

Instead of squaring the gradient vectors, a much simpler scheme may appear to be to use the arc tangent of one argument, which returns $\tan^{-1} x$ in the range $(-\pi/2, \pi/2)$. This effectively takes vectors that point in opposite directions and maps them onto the same direction, as is shown in Fig. 4. This is another way of ensuring that gradient vectors pointing in opposite directions actually reinforce each other instead of canceling when smoothing is performed.

Using this second scheme, we compute θ_{ij} , defined by

$$\theta_{ij} = \tan^{-1} \left(\frac{G_y(i, j)}{G_x(i, j)} \right), \quad (21)$$

where the arctangent is computed using one argument, and lies in the range $(-\pi/2, \pi/2)$. We now smooth the array of θ_{ij} values using a Gaussian filter, as was done in the first stage. However, for this smoothing, we used a larger filter size than what was used for smoothing the image. Let θ_{ij}^* denote the array of smoothed angle values. The estimated orientation angle at (i, j) is then $\theta_{ij}^* + \pi/2$.

4.5. Flow Orientation Coherence

Let $\theta(x, y)$ denote the estimated orientation angle at point (x, y) , found in the earlier step. Let $G(x, y)$ denote the gradient magnitude at point (x, y) in the image, as shown in Fig. 5. To find the coherence at point (x_0, y_0) ,

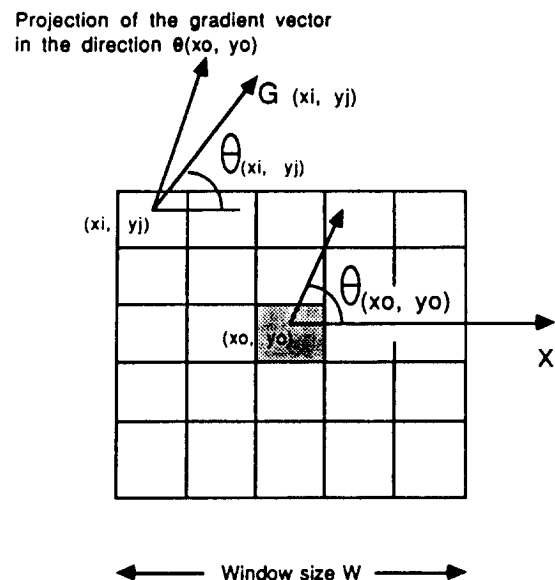


FIG. 5. Illustration of the method used to compute the coherence of the texture flow field.

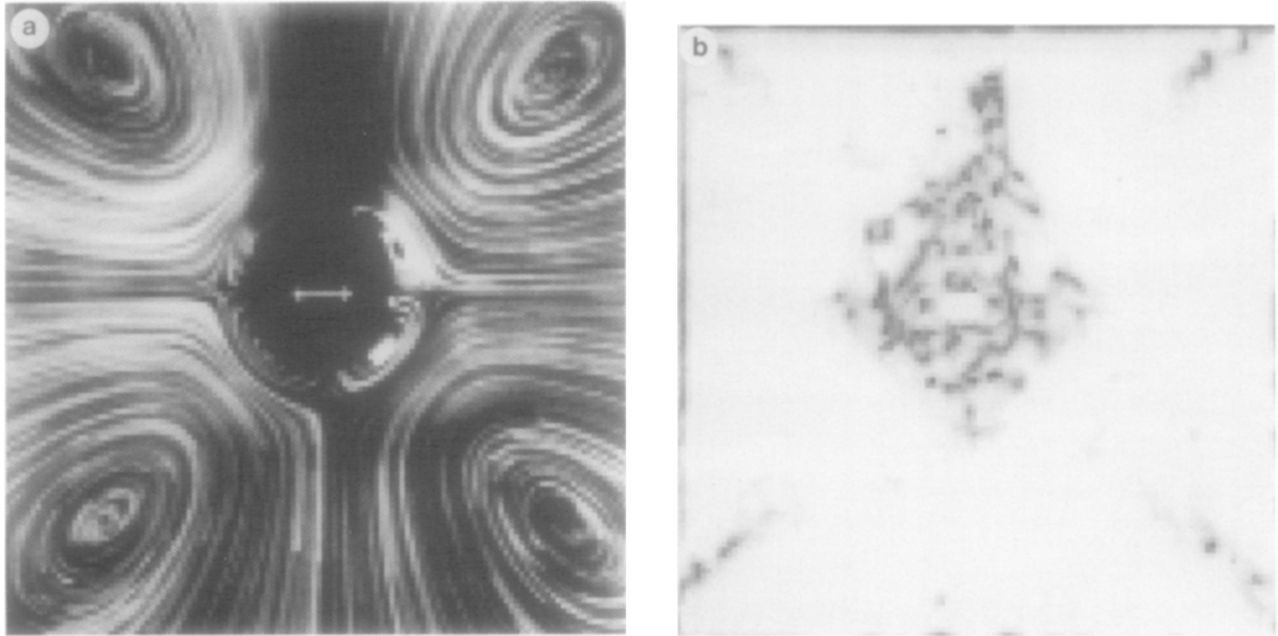


FIG. 6. (a) The original flow image (photograph courtesy of M. Van Dyke). (b) Result of applying the coherence measure in Eq. (22) to the flow image in (a).

consider the point (x_i, y_i) , where i and j are chosen so that they fall within a window W of prescribed size around the point (x_0, y_0) . Project the gradient magnitude $G(x_i, y_i)$ taken in the direction $\theta(x_i, y_i)$ onto the unit vector in the direction $\theta(x_0, y_0)$. This will be $G(x_i, y_i) \cos(\theta(x_0, y_0) - \theta(x_i, y_i))$. Compute the sum of the absolute value of such projections over all (i, j) values within the window. The

absolute value is used in order to avoid the cancellation of vectors that point in opposite directions. Consider the measure then defined by

$$\kappa = \frac{\sum_{(i,j) \in W} \|G(x_i, y_i) \cos(\theta(x_0, y_0) - \theta(x_i, y_i))\|}{\sum_{(i,j) \in W} G(x_i, y_i)}. \quad (22)$$

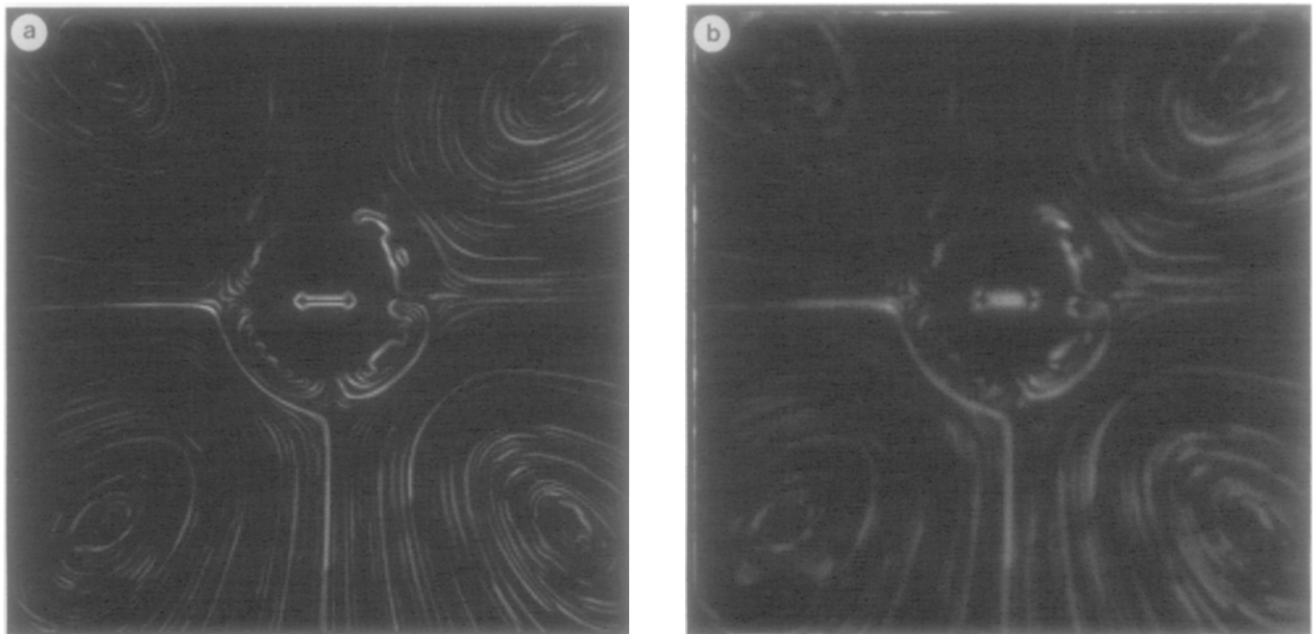


FIG. 7. (a) The coherence map obtained using Eq. (24). This gives better results than using Eq. (25), which is shown in part (b). (b) The coherence map obtained using Eq. (25).

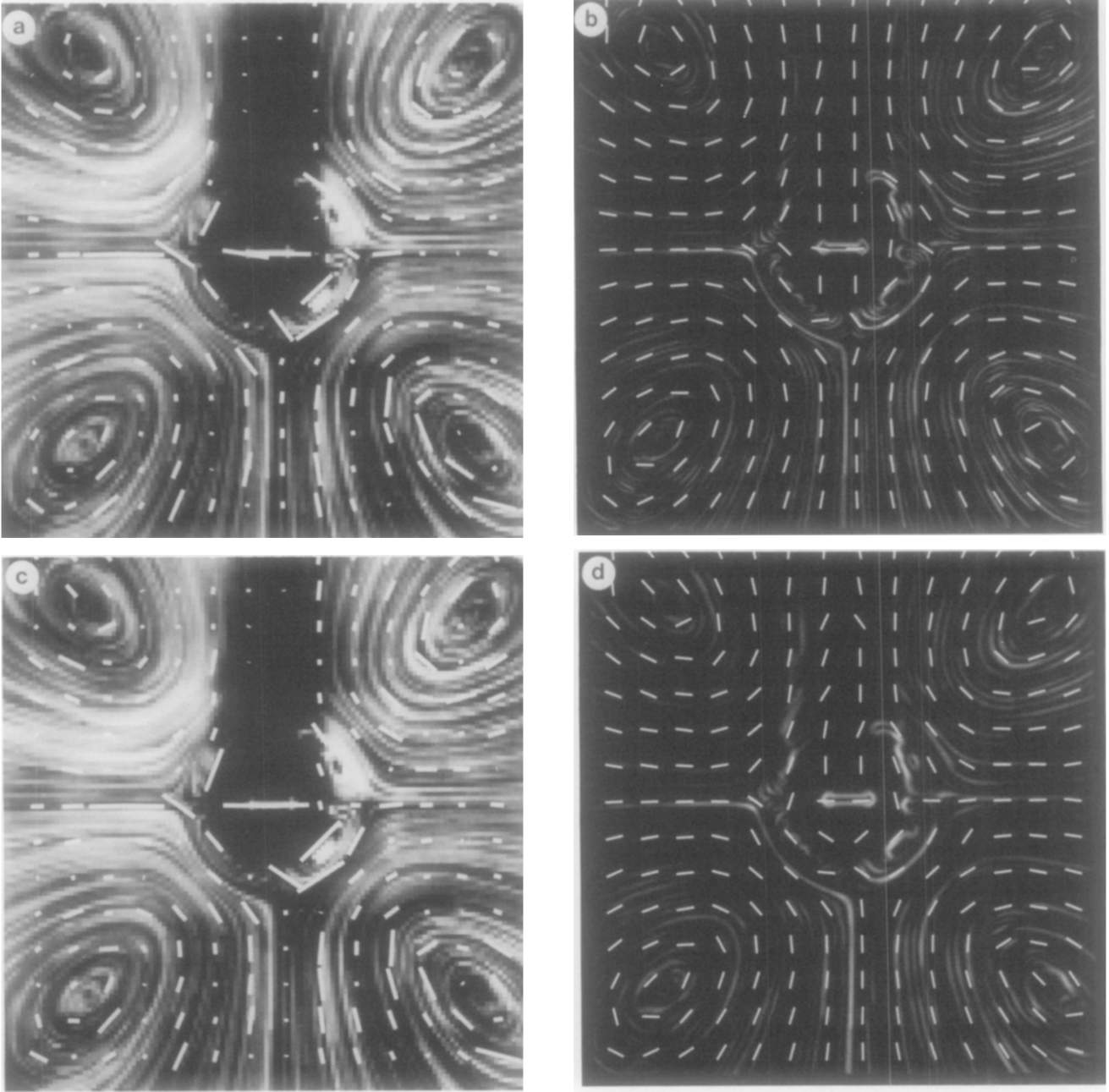


FIG. 8. (a) An image of a flow pattern induced by an oscillating cylinder (photograph courtesy of M. Van Dyke). The estimated flow directions (calculated from Eq. (16)) are represented by line segments overlaid on the original image. Filter sizes used were $\sigma_1 = 5$ and $\sigma_2 = 7$. The length of each line segment is proportional to the coherence at that point. Thus this image directly encodes the information about flow direction and flow coherence. (b) The coherence map. The coherence at each point of the original image (calculated from Eq. (24)) is encoded as an intensity value. Filter sizes used were $\sigma_1 = 5$ and $\sigma_2 = 7$. Unit vectors representing the estimated flow directions are superimposed on the coherence map. Note that the coherence is low within the cylinder. (c) The estimated flow directions (calculated from Eq. (16)) using filter sizes of $\sigma_1 = 9$ and $\sigma_2 = 13$. (d) Unit vectors representing the estimated flow directions are superimposed on the coherence map. (e) The estimated flow directions (calculated from Eq. (16)) using filter sizes of $\sigma_1 = 15$ and $\sigma_2 = 21$. (f) Unit vectors representing the estimated flow directions are superimposed on the coherence map. Note that the nature of (b), (d), and (f) has not changed even though the scale has changed significantly.

This measure is related to the *dispersion* of directional data [8]. Assume that we have n data points, where the i th point P_i is at an angle θ_i , and lies on the unit circle. Let α be a fixed direction. Then the dispersion D for the n data points about the orientation α is defined by

$$D = \frac{1}{n} \sum_{i=1}^n (1 - \cos(\theta_i - \alpha)). \quad (23)$$

Thus if the measure D is generalized to data that do not necessarily lie on the unit circle, then it follows that $\kappa =$

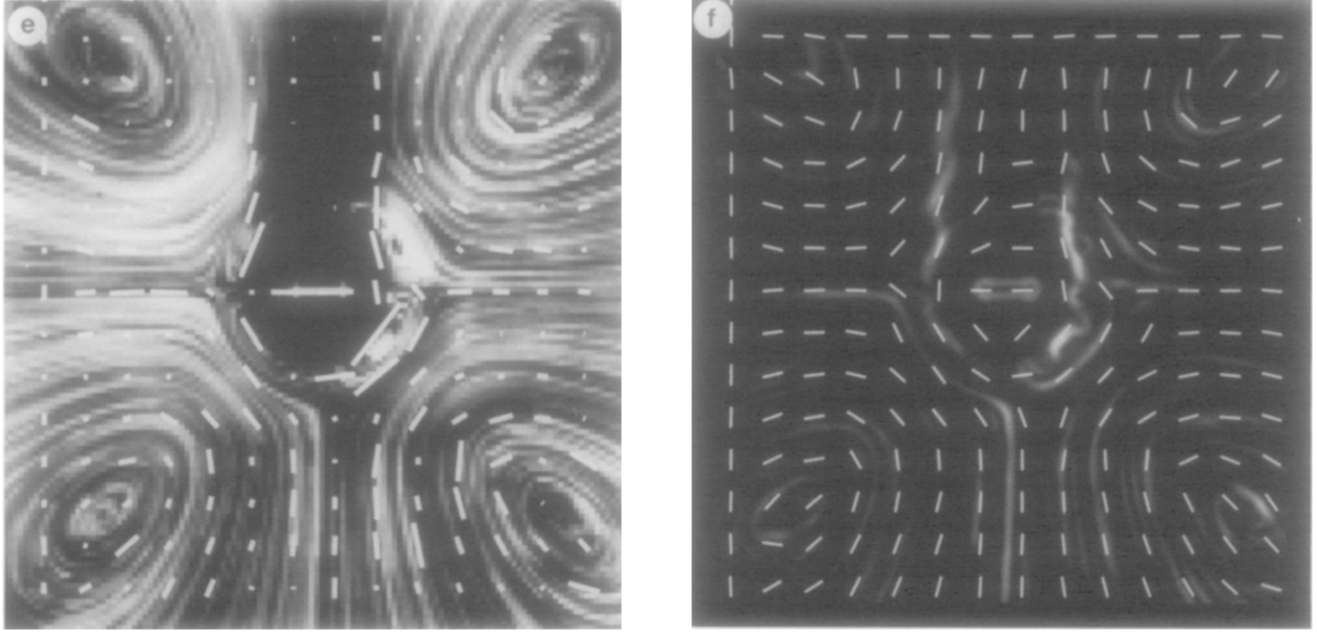


FIG. 8—Continued

1 – *D*. This shows that the measures for both orientation and coherence come from the same theory of statistics of directional data [8].

The result of applying the measure of coherence in Eq. (22) to the flow image is shown in Fig. 6. One can observe that the coherence values are low within the center of the image, and also around the four vortices in the corners. However, at other points in the image the coherence does not exhibit much variation.

One can obtain a better measure of coherence by weighting the quantity in Eq. (22) by the gradient magnitude at that point. Thus, the new measure of coherence is defined by

$$\rho = G(x_0, y_0) \frac{\sum_{(i,j) \in W} \|G(x_i, y_j) \cos(\theta(x_0, y_0) - \theta(x_i, y_j))\|}{\sum_{(i,j) \in W} G(x_i, y_j)}. \quad (24)$$

The reason for weighting by the gradient magnitude is that we want the coherence to be high at points in the image which have high visual contrast, i.e., high gradients. The result of applying this measure of coherence to the flow image is shown in Fig. 7a. The coherence measure that we propose incorporates the gradient magnitude and hence places more weight on regions that have higher visual contrast.

Kass and Witkin propose the following coherence measure

$$\rho = (J_x^*(i, j)^2 + J_y^*(i, j)^2)^{1/2} / G^*(i, j), \quad (25)$$

where $J_x^*(i, j)$ and $J_y^*(i, j)$ have been defined in Section 4.3 and $G^*(i, j)$ is the smoothed gradient magnitude, obtained by smoothing the gradient magnitude with a Gaussian filter. We have used both coherence measures and found that the coherence we propose in Eq. (24) gives better results, as is shown in Fig. 7.

4.6. The Effect of Varying σ_1 on the Estimate of Dominant Orientation

The scale corresponding to Eq. (1), σ_1 , is the size of the Gaussian filter used to derive the gradient vectors as presented in Section 4.1. We now provide a brief discussion of how varying σ_1 affects the estimate of dominant orientation in Eq. (16).

We shall restrict our analysis to the ideal case first. Consider again the example of a sine-wave grating, say $S(x, y)$, which is an ideal oriented texture. According to the method presented in Section 4.1, a best estimate of the dominant orientation at a point is provided by Eq. (16). This equation has embedded in it two smoothing operations: (1) smoothing with a Gaussian with variance σ_1 ; (2) a smoothing with a box filter of width σ_2 . Let us consider the effect of varying σ_1 on Eq. (16), when σ_2 is constant.

Let $g(x, y)$ be the gradient vector at a point (x, y) of the function $S(x, y) \otimes G(\sigma_1, x, y)$, where G is the Gaussian filter. If we use a Gaussian filter of different size $G(\sigma'_1, x, y)$, then it can be shown that the gradient vector at point (x, y) becomes $(\sigma'_1/\sigma_1)g(x, y)$; i.e., there is a linear scaling. Furthermore, all gradient vectors get scaled by the same amount. The effect of scaling all gradient vectors by

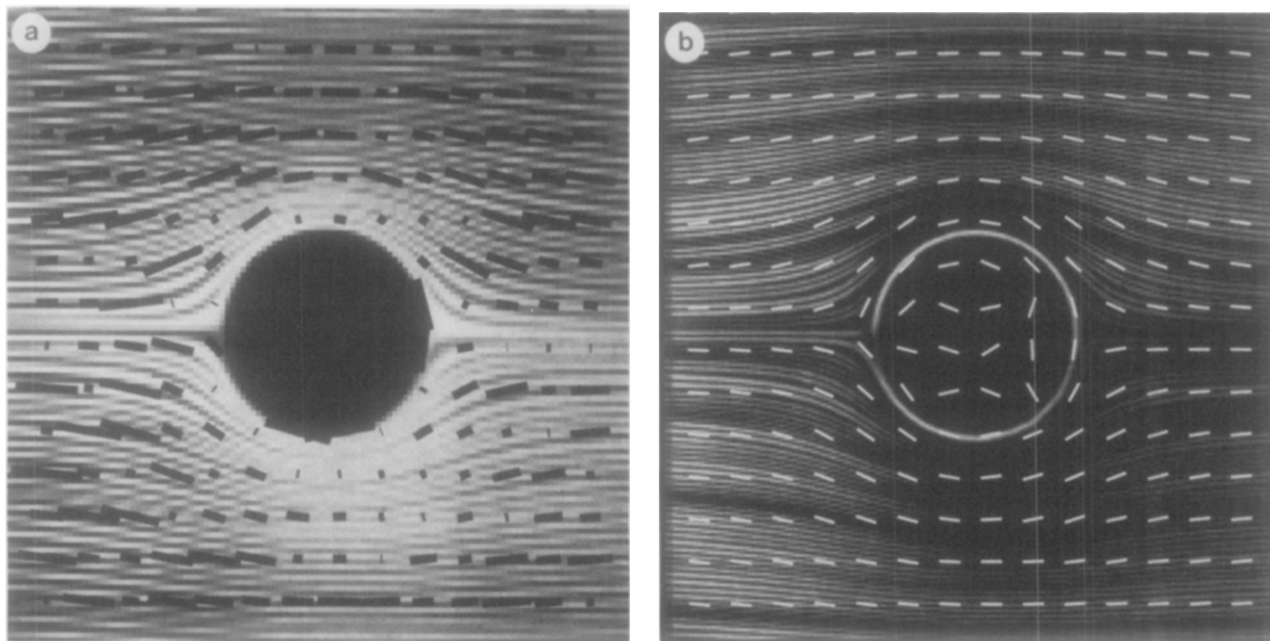


FIG. 9. (a) An image of flow past a circle, with estimated flow directions overlaid (photograph courtesy of D. H. Peregrine). Filter sizes used were $\sigma_1 = 5$ and $\sigma_2 = 7$. (b) The coherence map. Filter sizes used were $\sigma_1 = 5$ and $\sigma_2 = 7$.

the same constant leaves the best estimate in Eq. (16) unchanged. Thus, for an ideal texture pattern the choice of σ_1 to extract gradient vectors is immaterial. No matter what σ_1 is used, Eq. (16) provides the same estimate. In fact, this result holds for any texture that can be expressed as a linear combination of sine-wave patterns.

In the general case, we are faced with nonlinear textures (i.e., textures that cannot be expressed as a linear combination of sine-wave patterns), as illustrated in the next section. For such cases, the above analysis no longer holds, and the choice of σ_1 will affect Eq. (16). However, the behavior of the orientation estimation algorithm is not critically dependent on σ_1 or σ_2 . It is only when these parameters are varied *significantly* that different responses could result, as shown in the next section. If one knows a priori the sizes of features in the texture pattern, then this knowledge can be used to estimate the values of both σ_1 and σ_2 needed. Normally one would use widely spaced σ 's, as indicated in the edge detection theories of Marr and Hildreth [42] or Canny [38].

5. EXPERIMENTAL RESULTS

Figure 8a shows a 240×240 image of secondary streaming induced by an oscillating cylinder [40, p. 23]. The cylinder, which is in the center of the image, is oscillated by a loudspeaker in a fluid. This pattern results when suspended glass beads are illuminated by a stroboscope, and consists of four vortices around which there is

circulation. This texture is a useful test case because it exhibits orientation specificity at all possible angles, and is also symmetric. The results of applying the orientation estimation algorithm described in Sections 4.1 and 4.5 are displayed in Figs. 8a and 8b. The orientation field is calculated for each point in the image, but is displayed in a sampled form in order to avoid clutter. No arrowheads are drawn for the line segments because there is an inherent ambiguity in the direction of the line segment: it could point in either one direction or exactly the opposite.

Two methods of presenting the orientation field are used in order to provide easy means for interpretation. In the first method, the orientation field is overlaid on the original image to aid comparison. The orientation at each point is represented by means of a line segment, where the direction of the segment corresponds to the dominant local orientation, and the length of the line segment is proportional to the strength of orientation (or coherence). In the second method, the orientation at each point in the image is encoded as the angle of orientation of a unit vector and is superimposed on the coherence or strength of orientation which is encoded as a gray value.

Figure 8a shows the result of applying the orientation estimation algorithm at a scale of $\sigma_1 = 5$ to compute the gradient vectors, and a scale of $\sigma_2 = 7$ to smooth the gradient vectors. The oriented segments are displayed at regularly sampled points overlaid on the original image. The segments capture the flow of the texture very well at each point, and orient themselves along the direction of flow. Hence the resulting pattern looks much like the

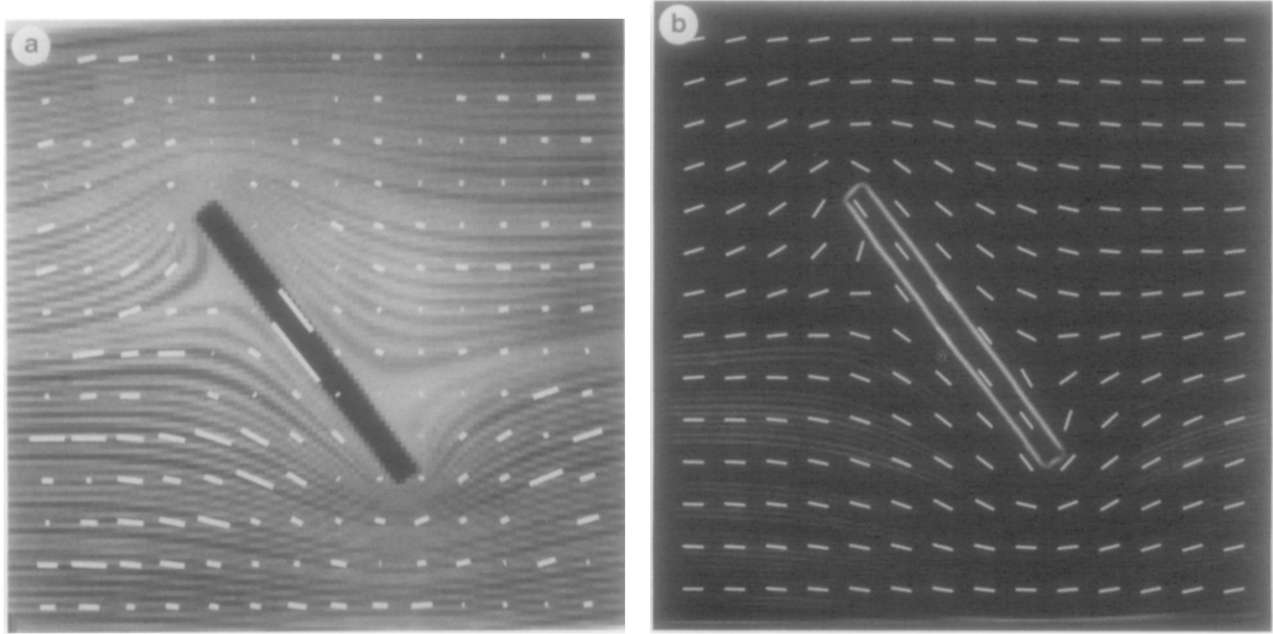


FIG. 10. (a) An image of flow past an inclined plate, with estimated flow directions overlaid. Filter sizes used were $\sigma_1 = 5$ and $\sigma_2 = 7$ (photograph courtesy of D. H. Peregrine). (b) The coherence map. Filter sizes used were $\sigma_1 = 5$ and $\sigma_2 = 7$.

original texture. Figure 8b shows the coherence of orientation (calculated using Eq. (24)) encoded as an image. The coherence at each point is quantized into one of 256 gray levels, and then displayed as an image. The angle vectors are then overlaid on the coherence image. The coherence image shows how orientation specificity varies

over the original image—the brighter points indicating strong coherence of flow. Note that the coherence at the center of the image is low, indicating that there is no dominant flow direction in that area. Thus, the coherence image combined with the overlaid angle image provides a good description of the underlying flow-like texture.

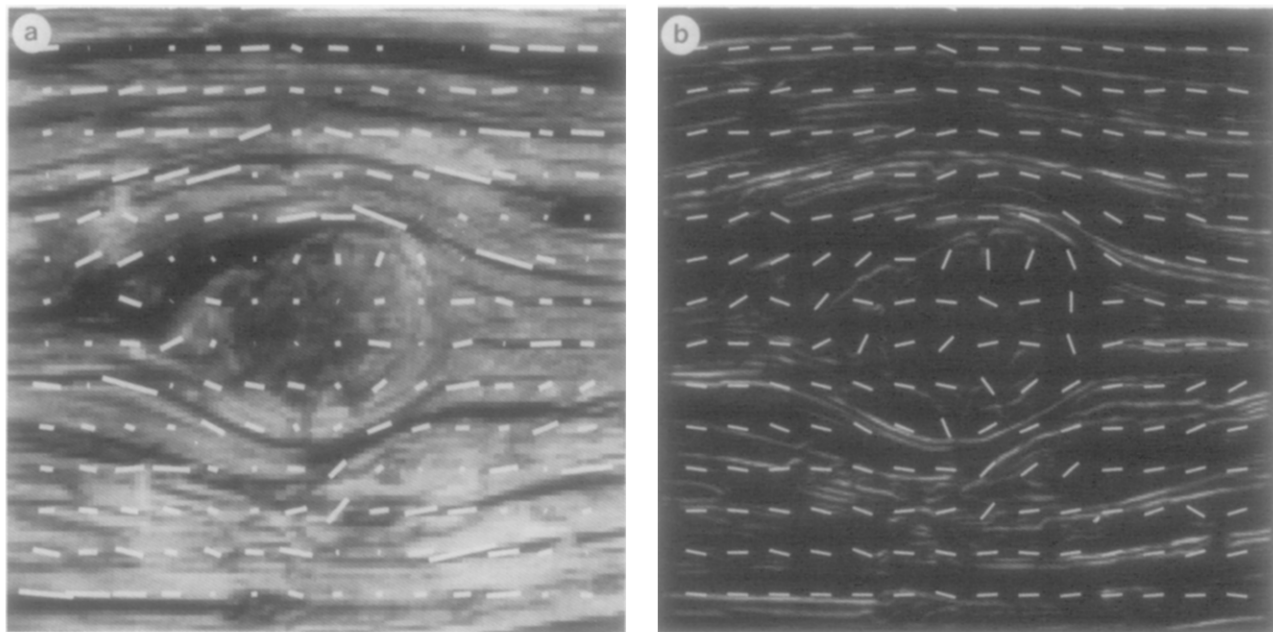


FIG. 11. (a) An image of a wood grain with a knot in the center. The estimated flow directions are overlaid on the original image. Filter sizes used were $\sigma_1 = 5$ and $\sigma_2 = 7$. (b) The coherence map. Filter sizes used were $\sigma_1 = 5$ and $\sigma_2 = 7$.

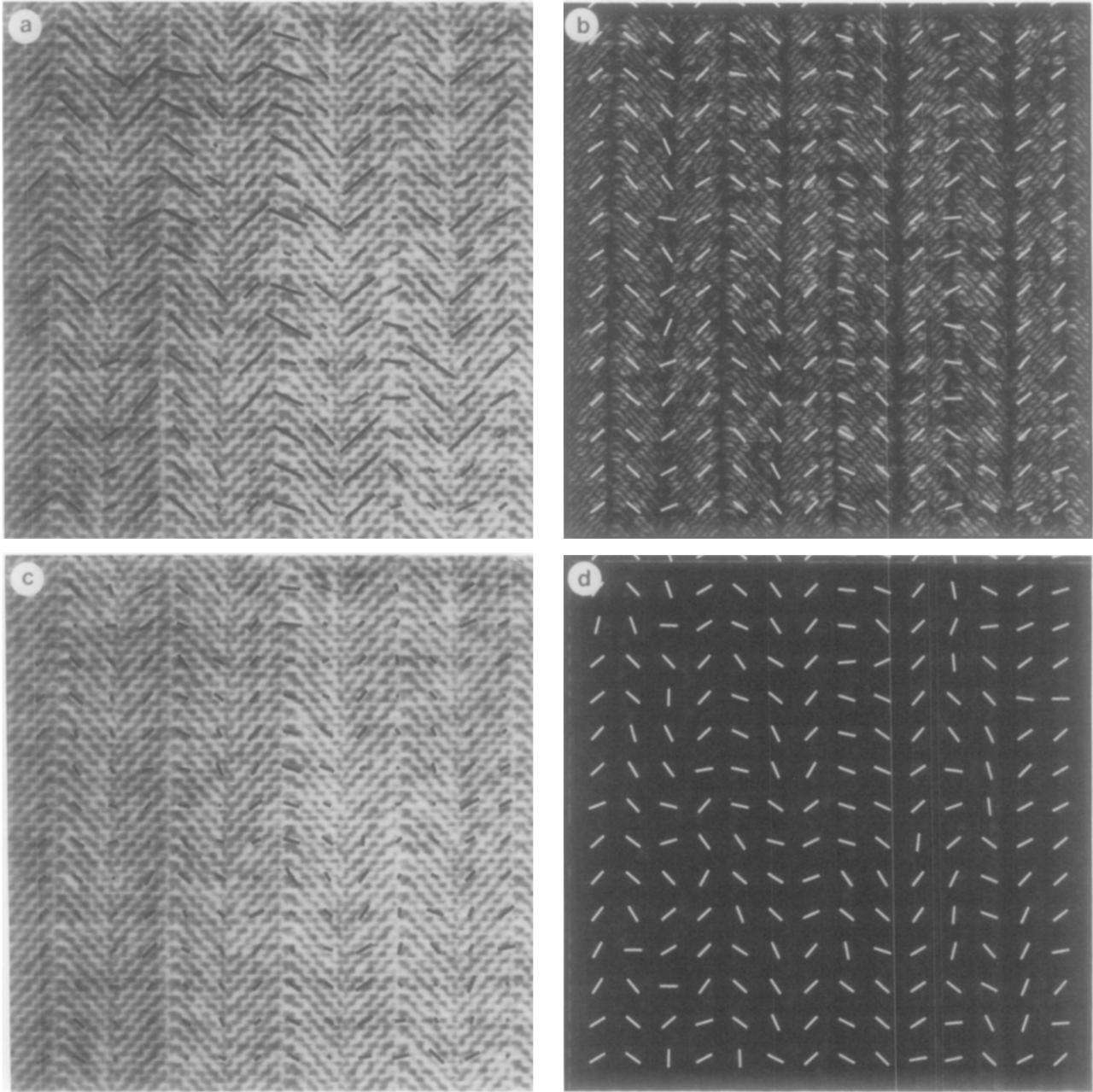


FIG. 12. Comparing the two schemes for measuring local orientations, as described by Eqs. (16) and (20). Figures (a) and (b) show the result obtained by using Eq. (16). Figures (c) and (d) show the result obtained by using Eq. (20) on the same image.

The filter sizes of 5 and 7 have been arbitrarily chosen at this point. In fact, Figs. 8c through 8f show the application of the orientation estimation algorithm at different scales to the same texture. The nature of the orientation field does not change significantly even though the filter sizes have been increased significantly. The results indicate that for the texture in question, the choice of scale does not play an important role in the description of the texture. However, this does not hold true for all textures,

and the descriptions of some textures change dramatically with changing scales, as illustrated in Section 7.

Figure 9 and 10 show more results of applying our algorithm to flow textures from an album of fluid motion [40]. The algorithm captures well the direction of anisotropy and the coherence at each point of the texture.

Figure 11a shows a knot in a piece of wood, scanned from a book of textures [41]. The orientation field swirls around the knot, though the fluid flow analogy is not

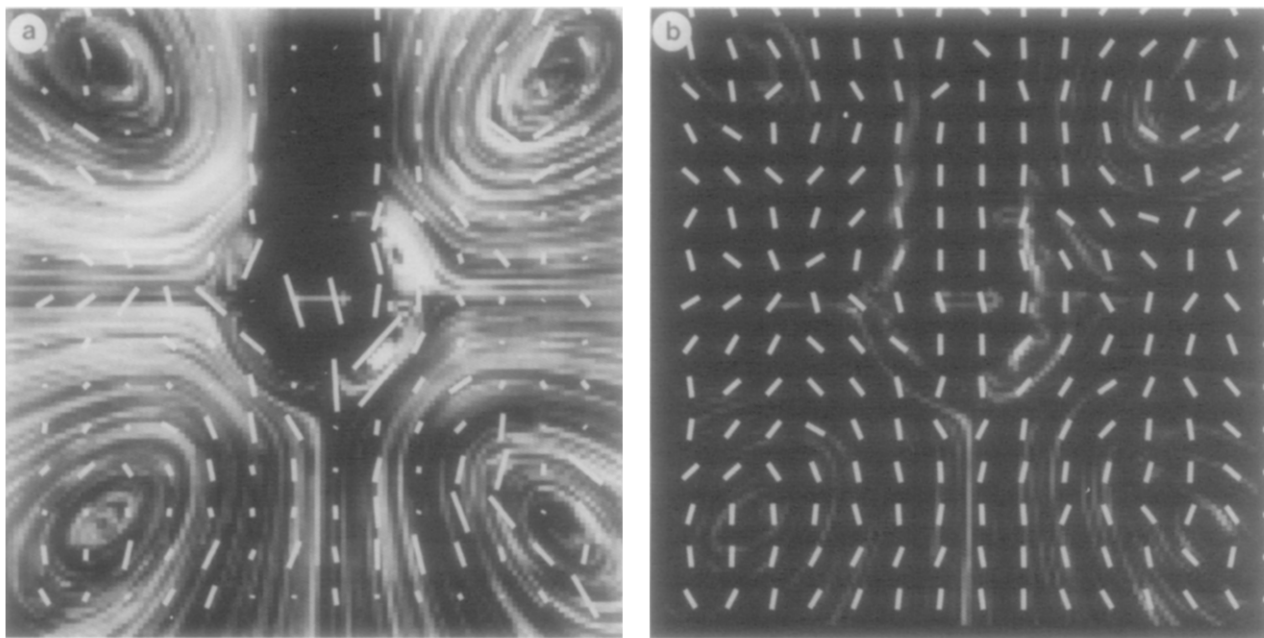


FIG. 13. Comparing two techniques for computing the angle of orientation. This figure shows results obtained by using Eq. (21). Note that the algorithm does not perform well on horizontally oriented patterns.

entirely correct as there is no directionality to the vectors: it is an orientation field, not a vector field. The results of applying the orientation estimation algorithm described in Sections 4.1 and 4.5 to the wood image are displayed in Fig. 11. The orientation vectors are drawn without arrowheads since the flow structure is nondirectional.

5.1. Comparing Calculations for Orientation

Figure 12 shows a herringbone weave texture, obtained from [41]. Figure 12 compares the two schemes for measuring local orientation, as described by Eqs. (16) and (20). Filter sizes used where $\sigma_1 = 5$ pixels and $\sigma_2 = 7$ pixels for both schemes. From a subjective estimate, Eq. (16) gives better results.

The use of Eq. (21) gives the result shown in Fig. 13a. This algorithm fails to respond well to flow patterns that are oriented horizontally. The problem with this approach is that gradient vectors having angles in the range $[\pi/2 - \varepsilon, \pi/2)$ and $(-\pi/2, -\pi/2 + \varepsilon]$ (indicated by the darkly shaded region in Fig. 4) do not get flipped around, and hence cancel each other instead of reinforce. This results in poor estimates of the orientation when the pattern has a horizontal flow direction.

5.2. Comparing Measures of Coherence

The result of using the coherence measure in Eq. (24) is shown in Fig. 7a). By comparing Figs. 7a and 7b, one can

see that Eq. (24) gives better results than Eq. (25). The coherence image corresponds closely with the original image, showing that the coherence measure that we have proposed is robust and accurate. We found this coherence measure to give better results than the measure proposed by Kass and Witkin.

6. ANALYZING TEXTURE AT DIFFERENT SCALES

There are many textures, such as the herringbone weave, which exhibit different behaviors at different scales. At finer scales, the pattern has more than one dominant orientation, but at coarser scales, the pattern has only one orientation.

This trend is clearly seen in the results shown in Fig. 14. This figure depicts a homespun woolen cloth, taken from [41]. At finer scales, the pattern appears to have alternating bands, where every other band has a texture that is diagonally oriented. At coarser scales, the pattern begins to appear vertical. This is the behavior that one would expect from the orientation estimation algorithm, and Fig. 14 proves that our algorithm indeed exhibits this behavior.

Figures 14a and 14b show the result of applying the orientation estimation algorithm at a scale of 5 to compute the gradient vectors, and a scale of 7 to smooth the gradient vectors. At these scales, the algorithm picks out the diagonal stripes in the texture. Figures 14c and 14d

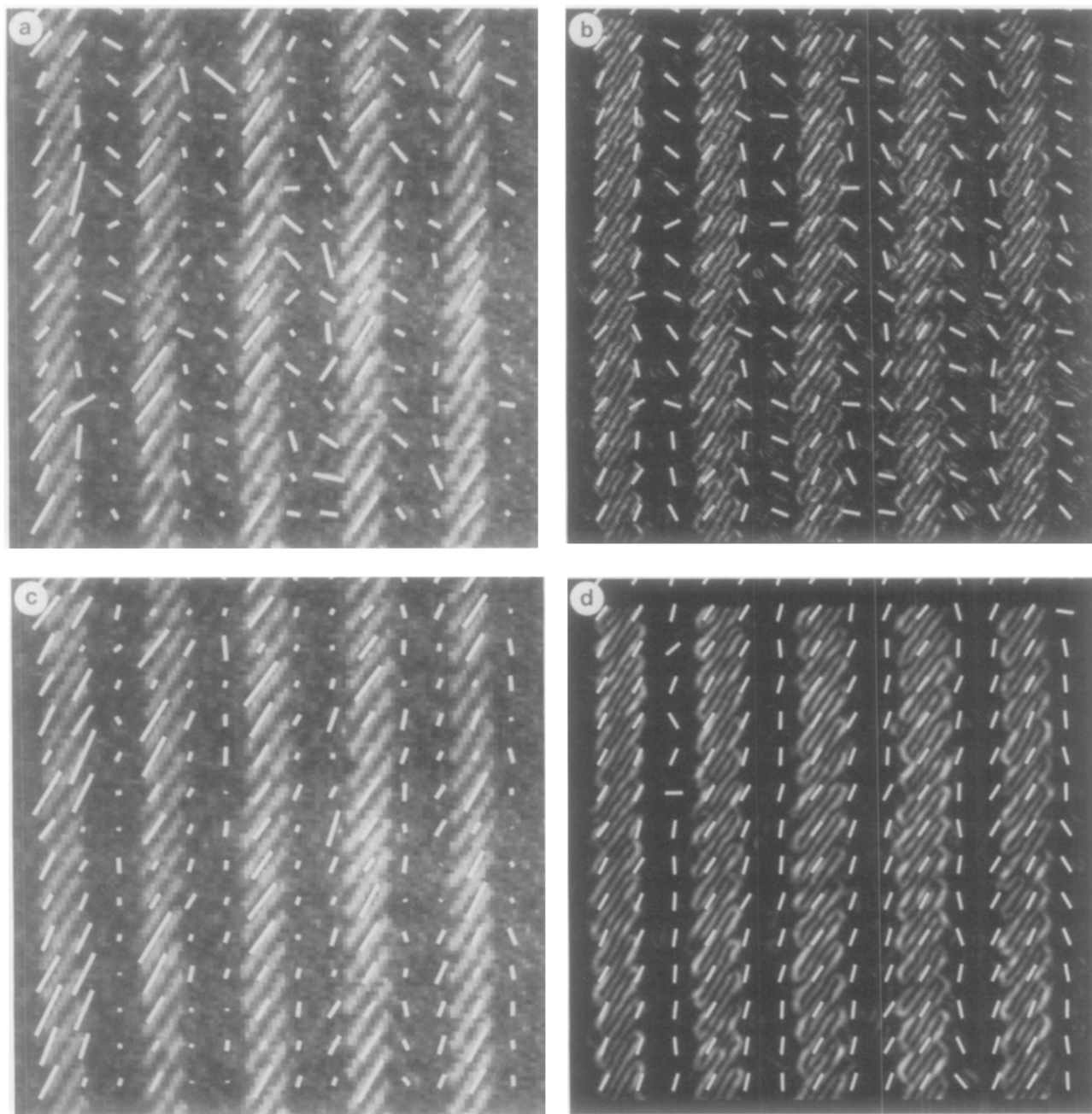


FIG. 14. Analyzing the behavior of the texture at varying scales. Figures (a) and (b) show the angle and coherence maps obtained by using a filter of window size 5 pixels to compute the gradient and a filter of window size 7 pixels to smooth the gradient vectors. Figures (c) and (d) show the angle and coherence maps obtained by using a filter of window size 11 pixels to compute the gradient and a filter of window size 17 pixels to smooth the gradient vectors. Figures (e) and (f) show the angle and coherence maps obtained by using a filter of window size 15 pixels to compute the gradient and a filter of window size 21 pixels to smooth the gradient vectors.

show the result of applying the orientation estimation algorithm at a scale of 11 to compute the gradient vectors, and a scale of 17 to smooth the gradient vectors. At these scales, the diagonal nature of the alternating bands begins to weaken, and starts to tend toward the vertical at areas close to the boundaries between the bands. Figures 14e and 14f show the result of applying the orienta-

tion estimation algorithm at a scale of 15 to compute the gradient vectors, and a scale of 21 to smooth the gradient vectors. At these scales, the texture appears predominantly vertical.

These results show that in order to analyze certain textures, one must employ different scales, because the description of the texture may vary according to the scale

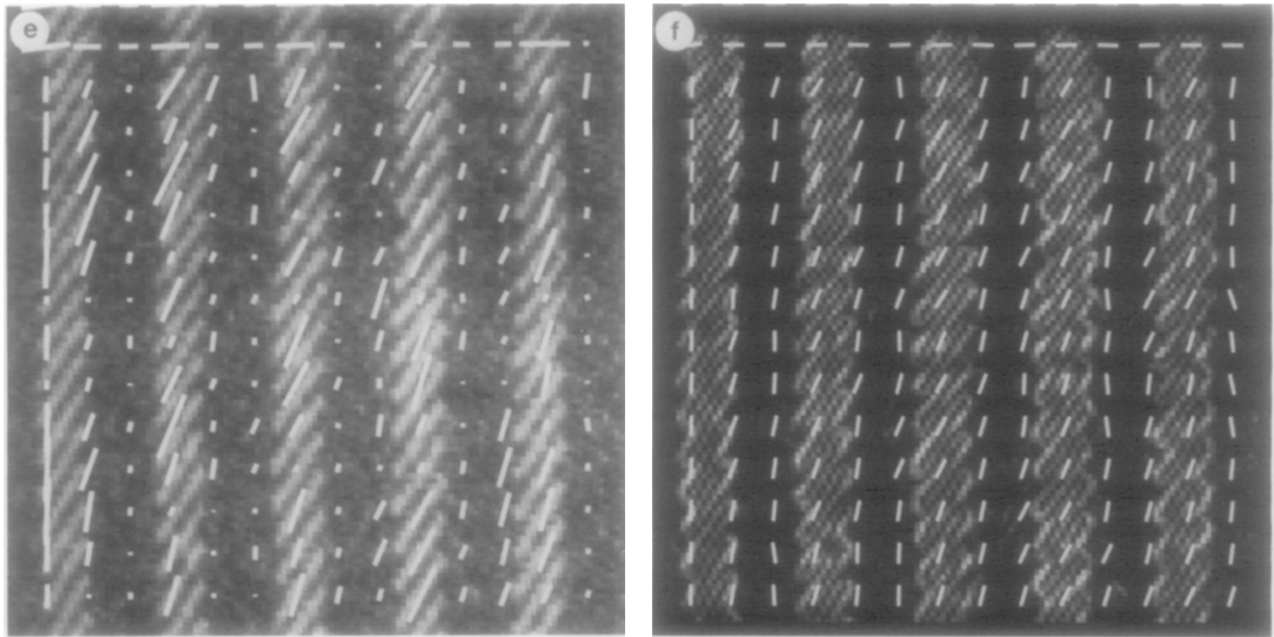


FIG. 14—Continued

employed. There are some textures, such as the homespun woolen cloth and the herringbone weave [41], which appear quite different at coarse and fine scales. We propose to extend our work in this direction in the future, where we will employ a bank of filters, tuned at different scales in order to perform orientation estimation. One can then use measures based on the coherence of the output to determine the scales at which the description of the texture is strong. In this manner, one can have rivalrous descriptions of textures such as the homespun cloth, where at one scale the texture appears to have certain orientations, and at another scale, the texture has different orientations.

Zucker and Cavanaugh [23] have performed experiments with subjective figures in texture discriminations. They investigate the Ehrenstein illusion, which involves two patterns—the Ehrenstein pattern and the control pattern. The Ehrenstein pattern consists of horizontal and vertical segments. The areas bounded by the endpoints of these segments appear as disks. Figure 15a shows the underlying Ehrenstein pattern, with the estimated orientation field overlaid. Figure 15b shows the corresponding angle and coherence images. Figures 15c and 15d show similar results for the control pattern, which is obtained from the Ehrenstein by shifting the horizontal segments by 0.5 cycles. Now let us rotate the Ehrenstein pattern by 45° about an axis perpendicular to the plane of the paper. Instead of an array of disks, subjective brightness stripes are seen, which are vertically oriented. However, the control pattern does not undergo any qualitative change as a function of orientation. Fig-

ures 15e through 15h show the orientation field for the rotated patterns. The results obtained by running the orientation estimation algorithm do not indicate the presence of subjective brightness stripes in the rotated pattern. This is because the coherence in the areas away from the line segments is very low, and hence the estimated direction of orientation within these areas has little meaning. This seems to indicate that the orientation estimation algorithm does not exhibit the subjective effects that humans do in this case. This is to be expected, because if the algorithm did exhibit the illusion, then it would not be independent of orientation of the texture, which is contrary to the fact that it is mathematically so.

7. PROCESSING OF THE INTRINSIC IMAGES

In keeping with the philosophy of Barrow and Tenenbaum [14], the angle and coherence images can be regarded as intrinsic images obtained from the original texture image. Intrinsic images were intended to capture a more invariant and more distinguishing description of surfaces than raw light intensities [14]. The recovery of these intrinsic images was based on domain-independent constraints generated through the physics of image formation. The first few intrinsic images proposed were those of range, orientation, reflectance, and incident illumination. The orientation intrinsic image, as originally used by Barrow and Tenenbaum [14], refers to surface orientation, as described by surface normals at each point in the image. However, we use the term *texture orienta-*

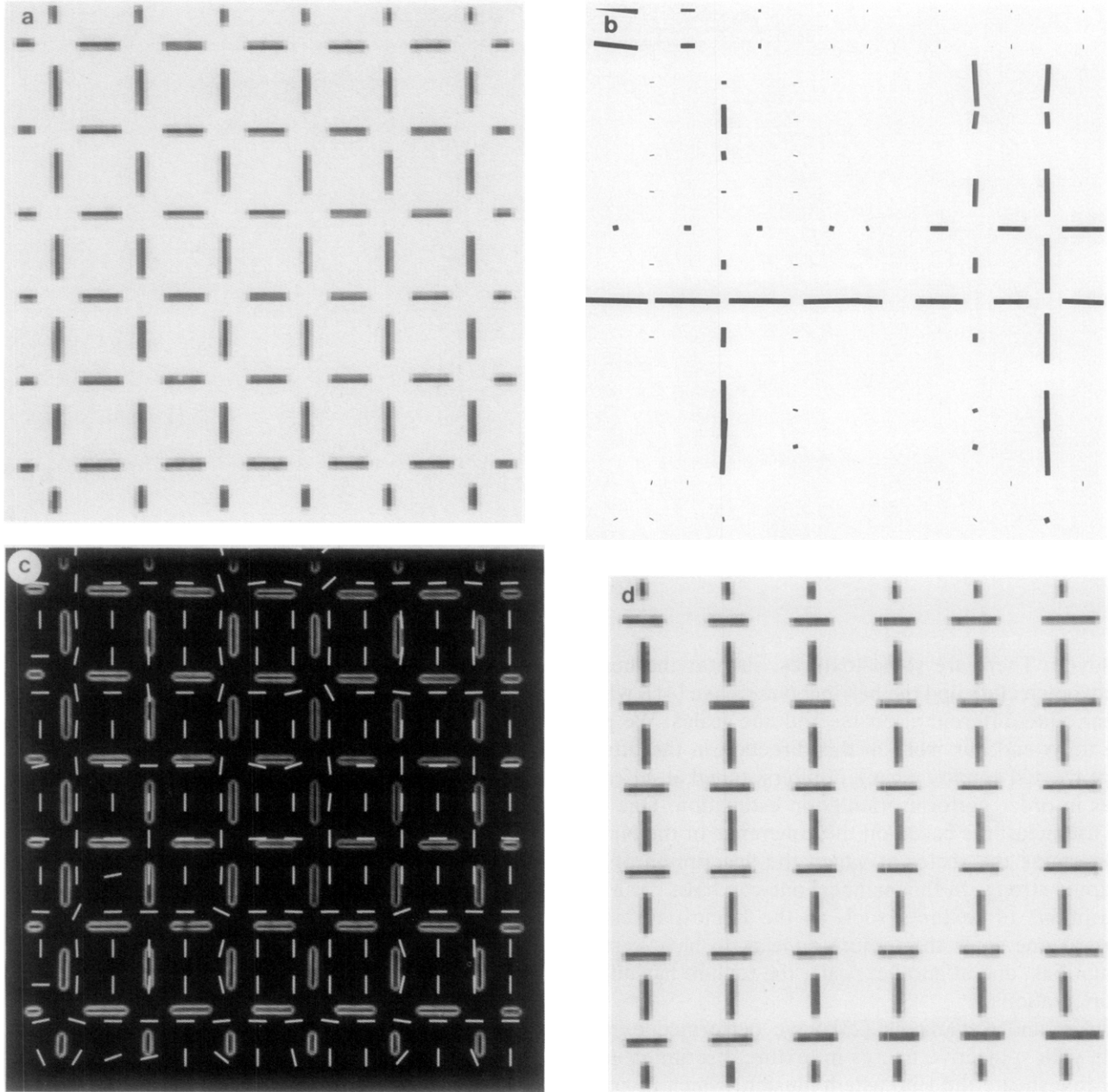


FIG. 15. (a) The original Ehrenstein pattern. (b) Its orientation field. Filter sizes $\sigma_1 = 5$ and $\sigma_2 = 7$ pixels were used. (c) The coherence map. (d) The control pattern and (e) its orientation field. Filter sizes $\sigma_1 = 5$ and $\sigma_2 = 7$ pixels were used. (f) The coherence map. (g) The Ehrenstein pattern rotated by 45° and (h) its orientation field. Filter sizes $\sigma_1 = 11$ and $\sigma_2 = 17$ pixels were used. (i) The coherence map. (j) The rotated control pattern and (k) its orientation field. Filter sizes $\sigma_1 = 11$ and $\sigma_2 = 17$ pixels were used. (l) The coherence map.

tion intrinsic images to describe local orientation specificity of patterns on a two-dimensional surface.

The use of the term “texture orientation intrinsic images” is justified because the spirit in which these images are extracted and processed echoes the principles cited above. First, these images are extracted by using domain-independent processing. Second, they are independent of lighting conditions, because of the process of averaging and normalization in computing the orientation and co-

herence. Third, domain-dependent knowledge and constraints can be imposed on the intrinsic images to yield appropriate conclusions in specific contexts. The viewpoint we have taken is also close to that of Marr [43], who proposed the extraction of a “primal sketch,” based on domain-independent processing.

We strongly advocate the use of the angle and coherence images as intrinsic images. Almost every analysis of oriented textures will require the computation of these

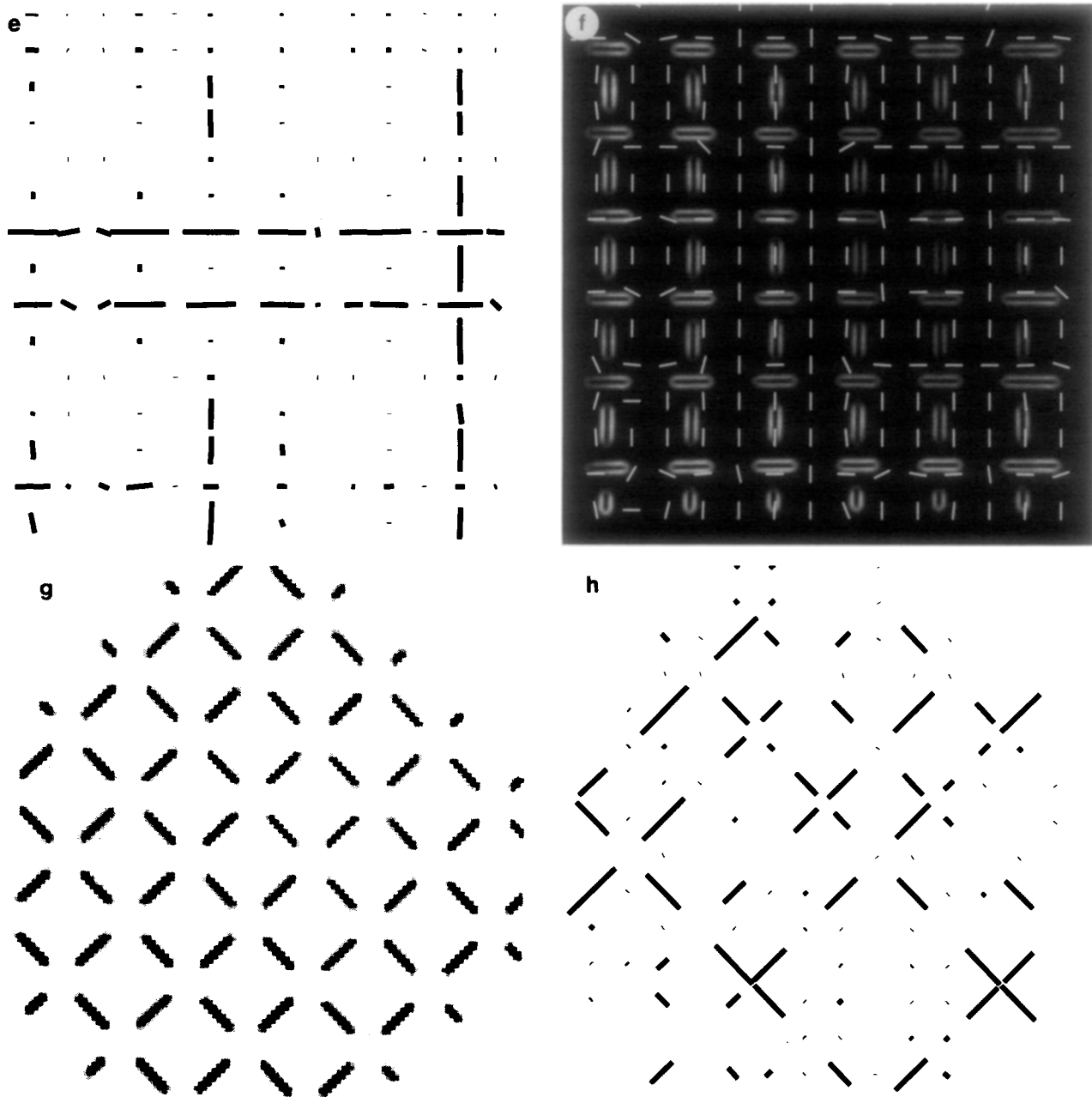


FIG. 15—Continued

intrinsic images as a first step. In this sense, the computation of the orientation field, resulting in the intrinsic images, is indispensable in the analysis of oriented textures. In order to justify this claim, we provide results from a number of experiments to indicate the usefulness of the angle and coherence intrinsic images.

Let us consider a specific domain in order to illustrate the application of our algorithm to real images. The automation of lumber defect detection is vital for the future control of lumber processing, which is an important industry [44]. Defects in wood are rich in textural content, and typically involve the identification of knots, worm

holes, checks, and grains [45–47, 5]. As observed in [5], tonal measures alone are not sufficient to isolate these defects. For instance, both checks and splits are darker than clear wood, just as knots are.

In order to illustrate the importance and usefulness of the texture orientation intrinsic images, we have used them to identify some of the defects that occur in wood, such as knots and worm holes.

7.1. The Coherence Intrinsic Image

The coherence image encodes the degree of anisotropy at each point of the original image. Coherence is low

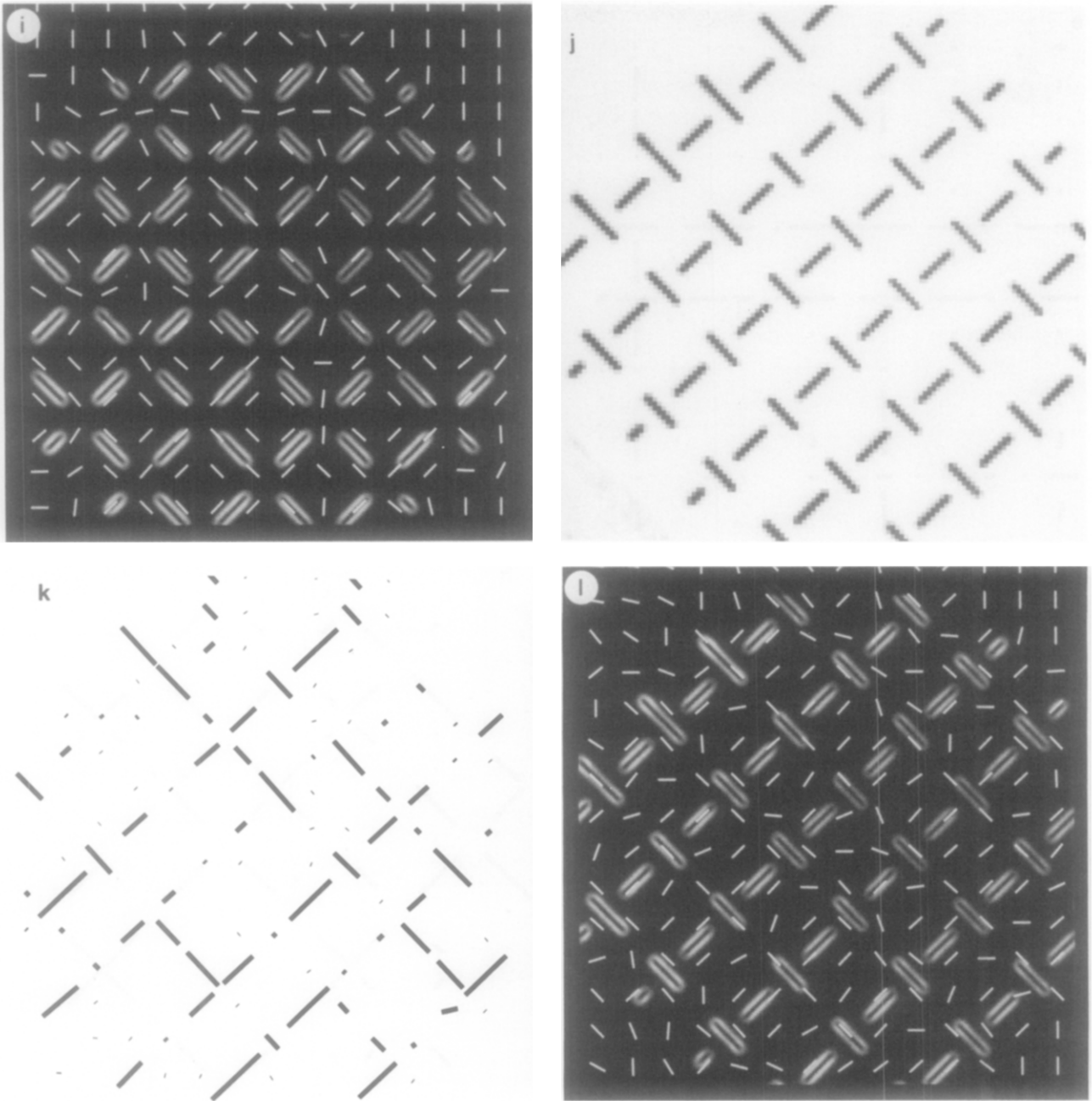


FIG. 15—Continued

within knots since there is no anisotropy of the texture within this region. Hence, the coherence image can be used to isolate knots, as these are locations in the oriented pattern that have low coherence. Figure 16a shows an image of a piece of wood containing three knots, obtained from [47, Fig. 10]. Figure 16b shows the result of computing the coherence of this image, by using Eq. (22). The coherence has been encoded as a gray value, with darker regions representing low coherence values. Recall that Eq. (22) computes a coarser measure of coherence than Eq. (24), as the latter incorporates the gradient mag-

nitude into its computation. Equation (24) would be suitable in an application where one needed a very detailed coherence map. However, Eq. (22) proves to be adequate in the isolation of knots, since it is a smoother measure of coherence. Figure 16c shows the result of thresholding the coherence image to produce a binary image. One can see that the locations of the knots have been isolated in the binary image, thus showing that simple processing of the coherence image yields meaningful results.

Figure 17a shows an image of a piece of wood contain-

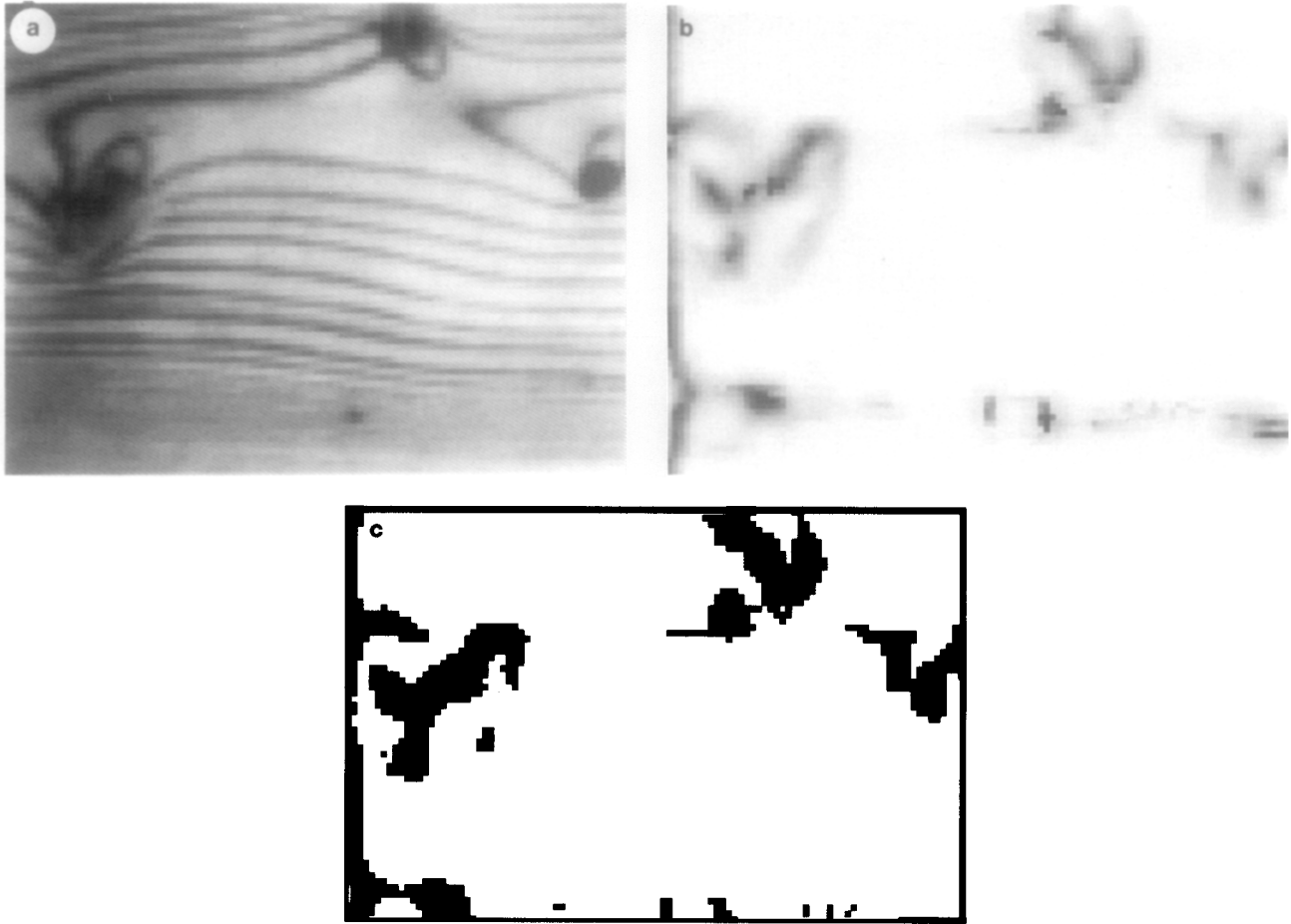


FIG. 16. (a) An image of a piece of wood containing three knots (reproduced with permission from *Defects in Wood*, by W. Erteld, W. H. Metta, and W. Acterberg, 1964 (Leonard Hill), Blackie, Glasgow/London). (b) The coherence image calculated using filter sizes $\sigma_1 = 5$ and $\sigma_2 = 7$ pixels. (c) The threshold coherence image. Note that the locations of the knots show up in the threshold image.

ing worm holes, obtained from [5]. Figure 16b shows the result of computing the coherence of this image, by using Eq. (22). The positions of the worm holes correspond to regions of low coherence, i.e., the dark regions. Figure 17c shows the result of thresholding the coherence image to produce a binary image. One can see that the locations of the worm holes have been isolated in the binary image.

An application where the detailed coherence map generated through Eq. (24) is used is illustrated in Fig. 18. Figure 18a shows the cross section of a piece of wood, obtained from [48]. The coherence measure in Eq. (24) is used to generate the coherence's image of Fig. 18b. The coherence image is then thresholded with a high threshold, so as to isolate regions having high coherence (the brighter a region, the higher its coherence). In the resulting image, shown in Fig. 18c, one can see that the cell boundaries show up. This is because the cell boundaries are regions of strong anisotropy in the texture, and hence become regions of high coherence.

It is in this sense that the coherence measure is similar to a measure of edginess. The spirit of the orientation estimation algorithm is the same as that of edge detection, as the early stages of processing are identical—they both involve smoothing and the computation of the gradient. However, the postprocessing reflects different goals. The goal of edge detection is to strictly *localize* the position of sharp gradients, while the goal of orientation estimation is to *combine* gradient information within a neighborhood to infer a dominant local orientation.

The coherence image can be processed in a variety of ways. Figure 19a shows the image of a brick wall, obtained from [41]. The coherence measure in Eq. (24) is used to generate the coherence image of Fig. 19b. The coherence image is then thresholded with a *low* threshold, so as to isolate regions having low coherence (the brighter a region, the higher its coherence). The resulting image is shown in Figure 19c. Note that the interior of the bricks are regions with low coherence, and are hence

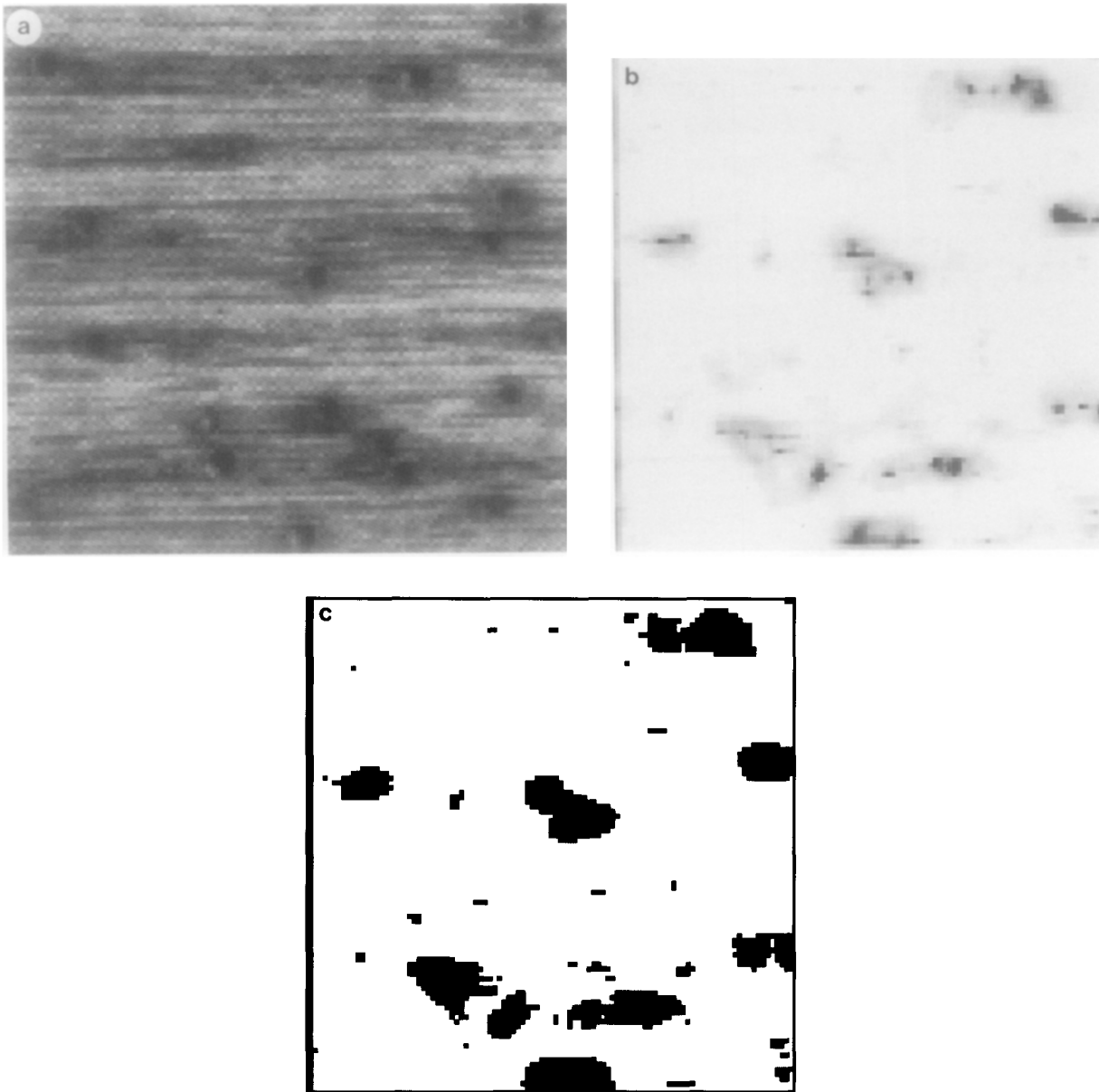


FIG. 17. (a) An image of a piece of wood containing worm holes (reproduced, by permission of the publisher, from Ref. [5], © 1983 IEE). (b) The coherence image calculated using filter sizes $\sigma_1 = 5$ and $\sigma_2 = 7$ pixels. (c) The threshold coherence image. Note that the locations of the worm holes show up in the threshold image.

isolated in the thresholding image. This can then be used to locate the positions of individual bricks within the lattice.

Another class of defects that appears in wood is that of *checks* [46, p. 284]. Checks are longitudinal openings at weak points in the wood. One way of determining the gross “checkiness” of a piece of wood is to simply compute its average coherence. The larger the number of checks, or the more severe the intensity of the checks, the larger the coherence measure will be. Hence a simple averaging of the coherence image suffices to obtain a global measure of the number of checks on the wood.

A problem that arises in semiconductor wafer inspection is determining the precise characterization of surface texture [49]. For instance, an *orange peel* defect arises when the surface texture appears wrinkled, like the skin of an orange. Such a wrinkled texture is actually an oriented texture, and the method described in this section can be used. Figure 20a shows an image of an orange peel defect. The sample consists of a silicon wafer that has undergone an isotropic etch using hydrofluoric and nitric acids.

Currently, inspection technicians use terms like “light” orange peel and “heavy” orange peel to describe

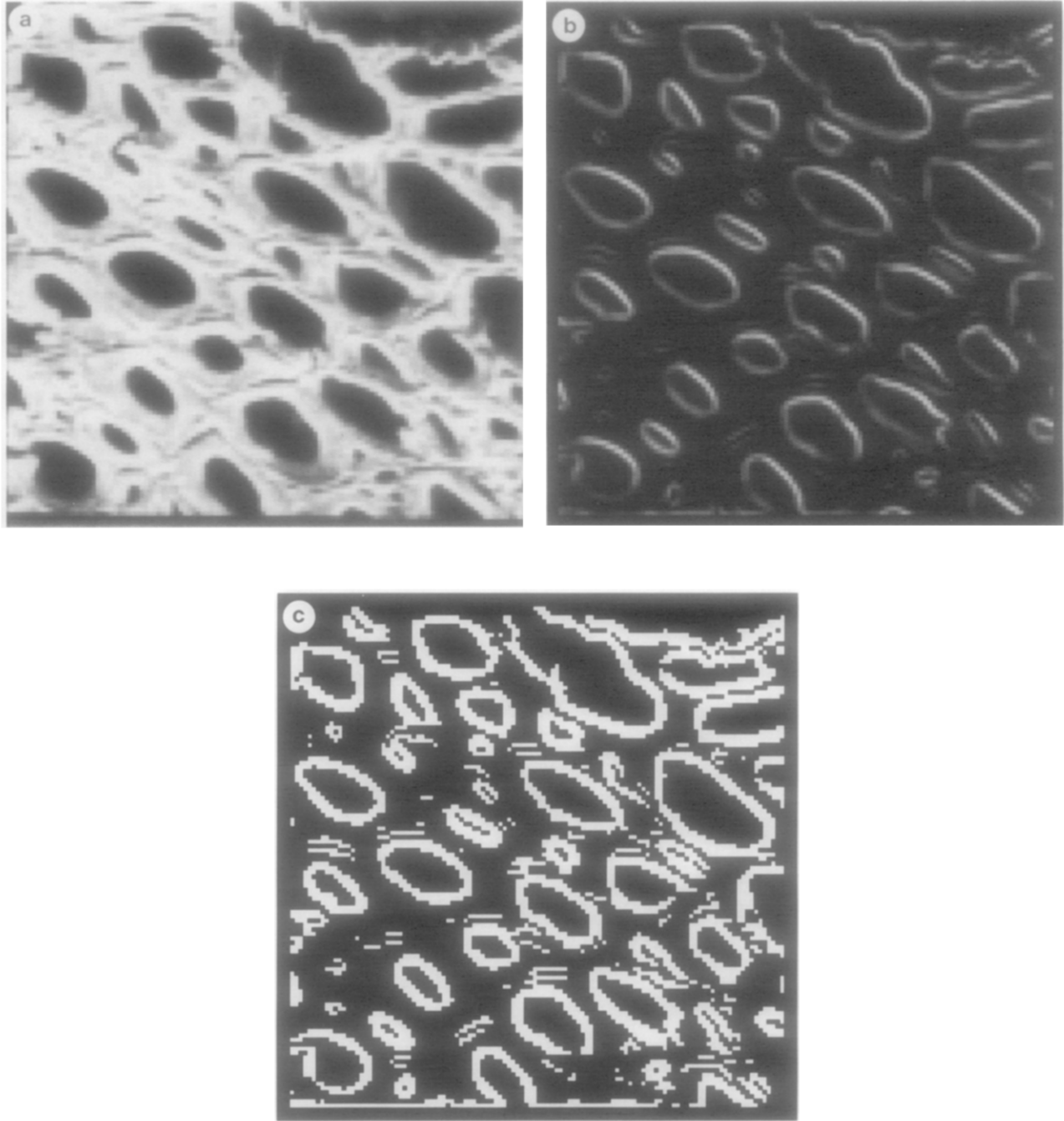


FIG. 18. (a) An image of a cross section of wood. (b) The coherence image calculated using filter sizes $\sigma_1 = 9$ and $\sigma_2 = 13$ pixels. (c) The threshold coherence image. Note that the boundaries of the cells show up in the threshold image.

the severity of the defect. However, no precise quantitative measure is available. Hence, we propose the following measure for the severity of an oriented texture defect, such as orange peel. Find the average coherence $\bar{\rho}$ over the image of the defect, where ρ is defined in (24).

In the case of the image shown in Fig. 20, the average coherence measure for the region of orange peel is 0.13, which represents the severity of the defect.

7.2. The Angle Intrinsic Image

When the fiber alignment in a piece of wood does not coincide with the longitudinal axis of the piece, the wood is said to be *cross-grained* [46, p. 251]. Any form of cross grain which occurs in structural lumber is a defect because of the reduction in the strength of the member in which it occurs. The direction of grain may be deter-

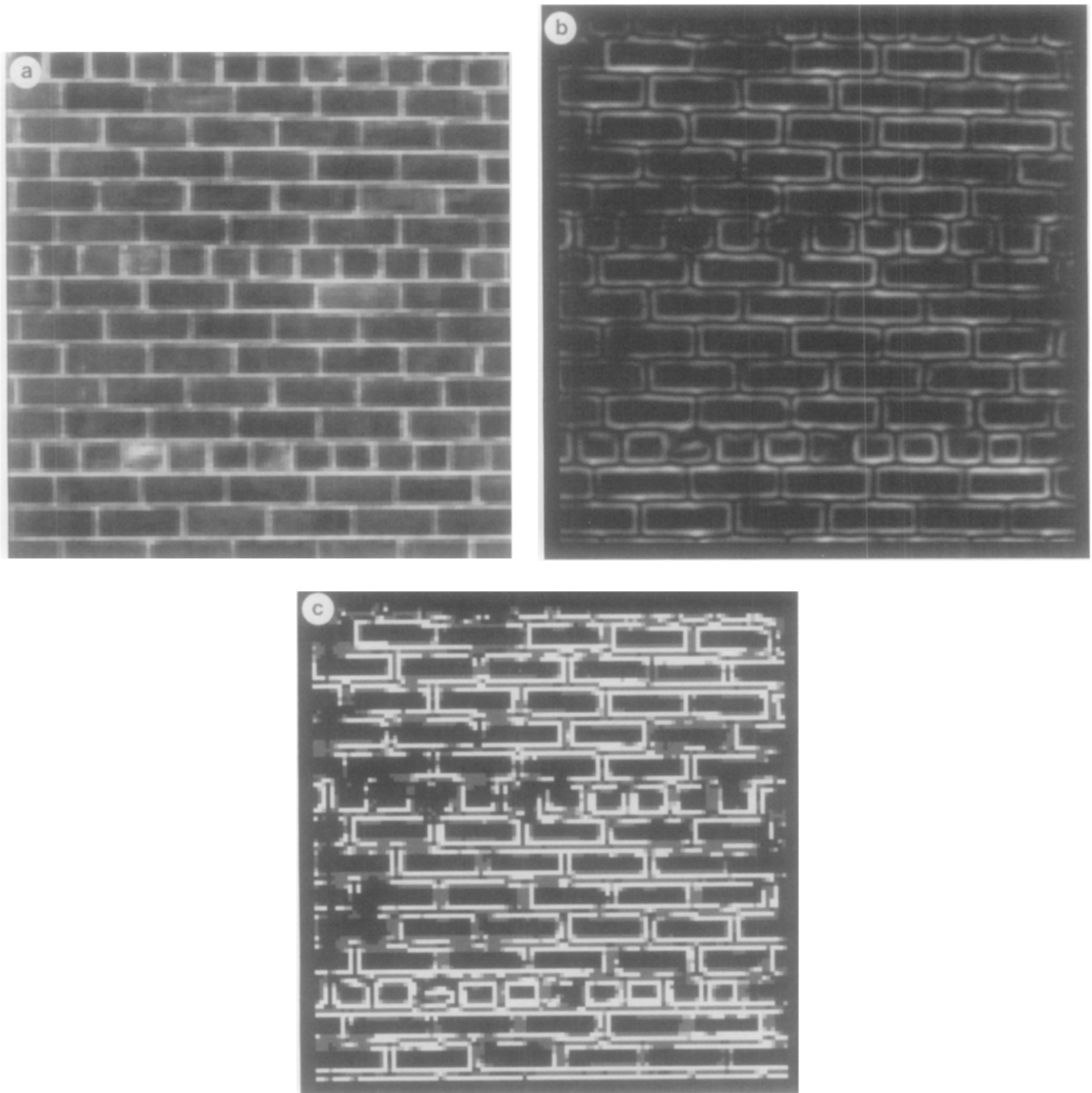


FIG. 19. (a) An image of a brick wall. (b) The coherence image calculated using filter sizes $\sigma_1 = 9$ and $\sigma_2 = 13$ pixels. (c) The threshold coherence image. Note that the interior of the bricks show up in the threshold image.

mined by observing the orientation of resin canals or vessels.

The angle intrinsic image readily yields this information. All one needs to do is histogram the angle image, and locate the peak of the histogram. This will yield the most dominant global orientation in the texture, and will coincide with the direction of the grain. In order to estimate the dominant global directions, one can compute an

angle histogram, weighted according to the coherence measure. Let us map all the angles into the range $[0, \pi)$, and discretize this range into 180 bins, each bin corresponding to the angle in degrees. Let the angle at point (i, j) be θ_{ij} deg, where θ_{ij} is quantized to be an integer in the range $[0, 180)$. Let the coherence at this point be ρ_{ij} . Let H denote the histogram, and H_k be the k th bin of the histogram. In order to compute the weighted histogram,

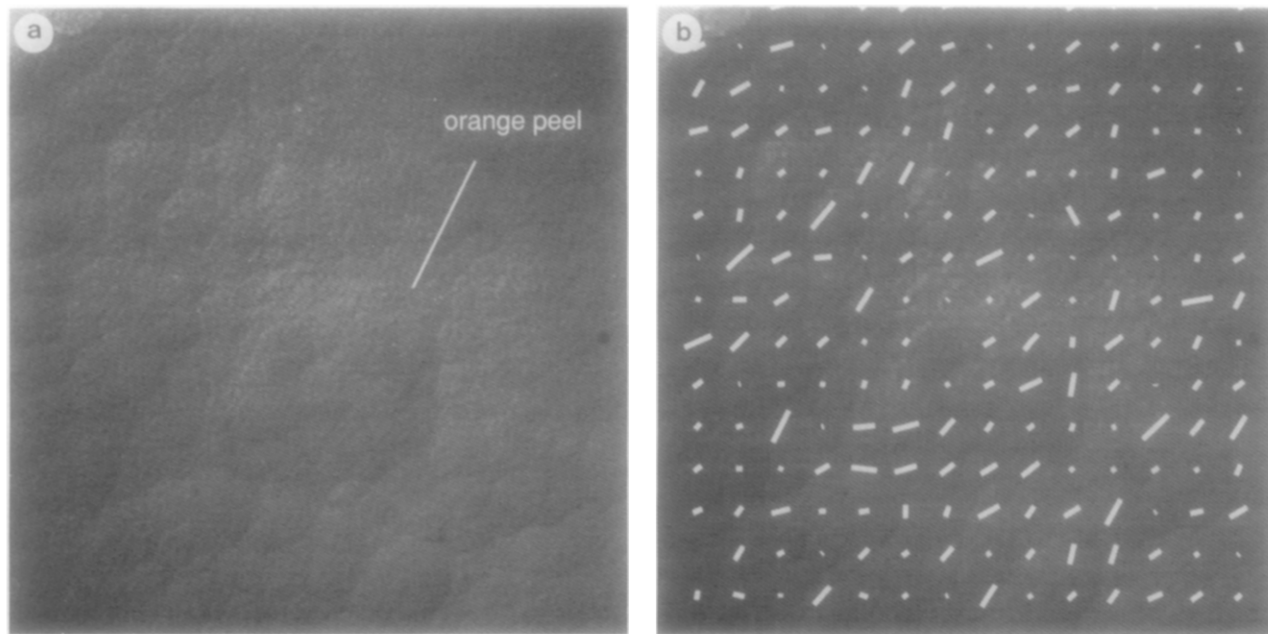


FIG. 20. (a) The original image of the orange peel defect, obtained at a magnification of 500, using Nomarski phase contrast. (b) The orientation field overlaid on the original image in the form of white segments.

the contribution of point (i, j) is then ρ_{ij} toward bin θ_{ij} . These contributions are summed over all image points, in order to yield an angle histogram. Thus,

$$H_k = \sum_{i,j=1,N} (\rho_{ij} | \theta_{ij} = k). \quad (26)$$

Figure 21a shows an image of a piece of wood containing checks. In order to find out whether there is a dominant orientation, one needs to compute the orientation field, which is shown in Fig. 21b. Figure 21c shows the weighted angle histogram, computed according to Eq. (26). One can see that there is a very sharp peak at $\theta =$

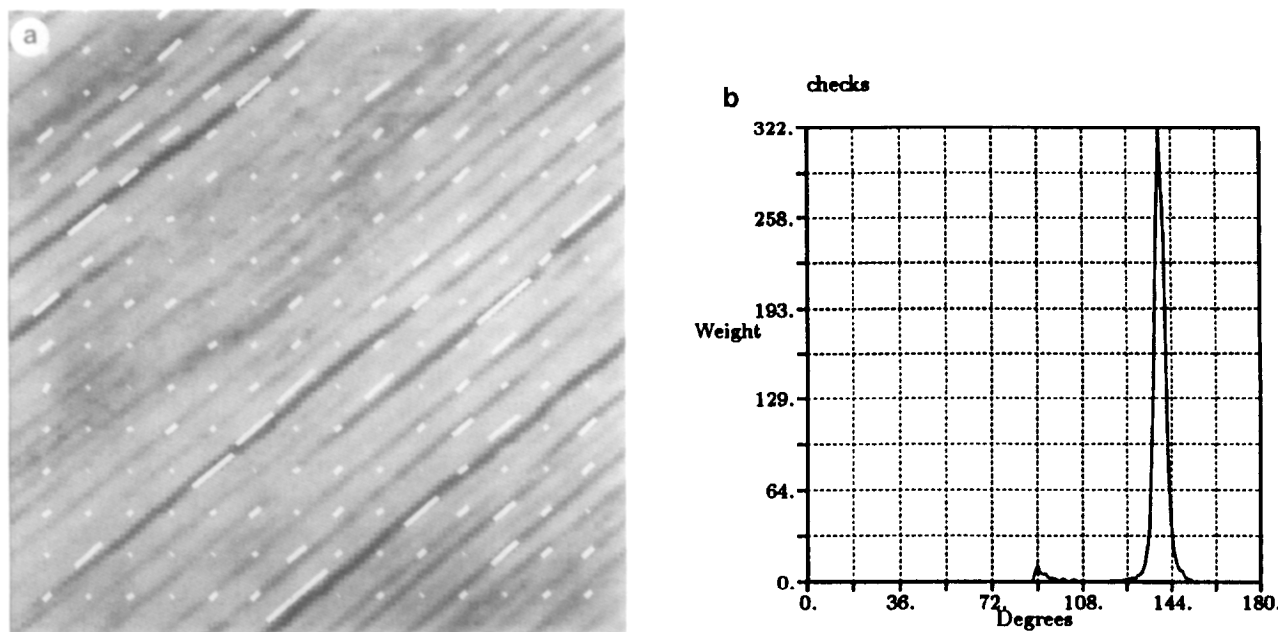


FIG. 21. (a) An image of a piece of wood with checks and its orientation field computed for filter sizes $\sigma_1 = 5$ and $\sigma_2 = 7$ pixels. (b) The weighted angle histogram, showing a sharp peak at 140° , indicating that the texture is strongly unidirectional, with an orientation of $\theta = 140^\circ$.

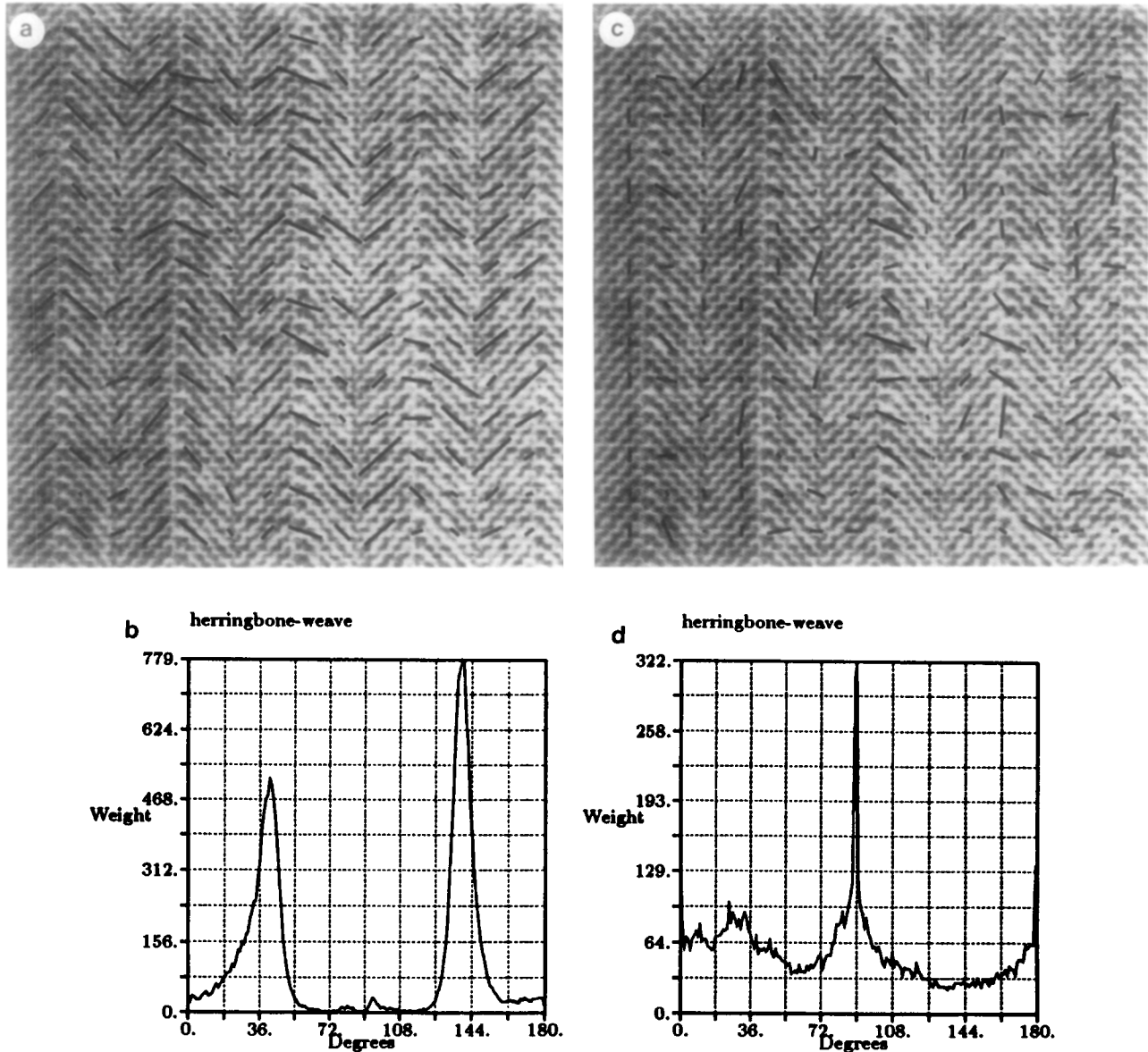


FIG. 22. (a) An image of a herringbone weave, from [41], with its orientation field computed for filter sizes $\sigma_1 = 5$ and $\sigma_2 = 7$ pixels. (b) The weighted angle histogram, showing sharp peaks at 40° and 140° . (c) The orientation field computed for filter sizes $\sigma_1 = 15$ and $\sigma_2 = 21$ pixels. (d) The weighted angle histogram, showing a sharp peak at 90° , indicating that the texture is predominantly vertical.

140° , showing that the pattern is strongly unidirectional, and the dominant direction is 140° . Thus, one can easily obtain reliable estimates of dominant orientation in such cases. (In all the figures displayed, the angle is measured clockwise with respect to the x -axis, which points horizontally to the right).

Figures 22 and 23 show another application of the angle histogramming technique, where the dominant orientation is computed over different scales. As mentioned in Section 6, the description of a texture may change over different scales. One way this can happen is that the dominant orientation of a texture can change dramatically when observed at different scales. Figure 22a shows an

image of a herringbone weave, digitized to a size of 240×240 pixels, obtained from [41]. Figure 22b shows the result of applying the orientation estimation algorithm at a scale of 5 to compute the gradient vectors, and a scale of 7 to smooth the gradient vectors. At these scales, the texture has two distinct directions, as shown by the weighted angle histogram in Fig. 22c. There are strong peaks at 40° and 140° . A dramatically different picture emerges when the orientation estimation algorithm is applied at a scale of 15 to compute the gradient vectors, and a scale of 21 to smooth the gradient vectors. Figure 22d shows the result of applying the orientation estimation algorithm, and Fig. 22e shows the weighted angle histo-

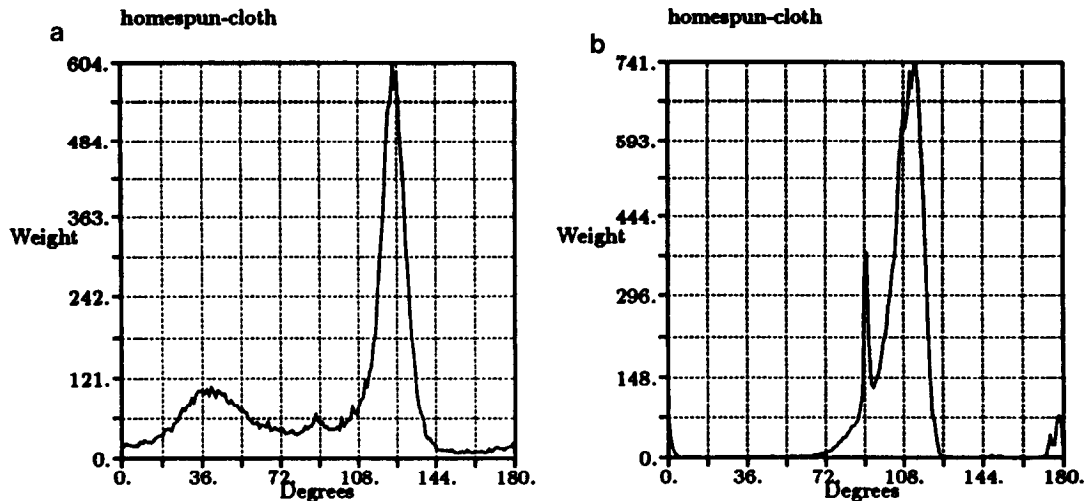


FIG. 23. The angle histograms for the image of homespun cloth, shown in Fig. 14. The orientation fields have been displayed in Figs. 14a and 14c. (a) The weighted angle histogram, showing a sharp peak at 120° . This was calculated from the orientation field. Filter sizes $\sigma_1 = 5$ and $\sigma_2 = 7$ pixels were used. (b) The weighted angle histogram, showing sharp peaks at 90° and 120° . This indicates that the texture has started acquiring a strong vertical component at this scale. The orientation field was computed for filter sizes $\sigma_1 = 15$ and $\sigma_2 = 21$ pixels.

gram. Figure 22e displays a completely different behavior from Fig. 22c, because there is only one sharp peak in the histogram at 90° . This shows that the texture appears predominantly vertical at higher scales, and distinctly bi-directional at lower scales.

Figure 23 illustrates a similar behavior in the case of an image of homespun cloth, obtained from [41]. Figure 14a shows the result of applying the orientation estimation algorithm at a scale of 5 to compute the gradient vectors, and a scale of 7 to smooth the gradient vectors. At these scales, the texture has one dominant direction, at 120° , as shown by the weighted angle histogram in Fig. 23a. Figure 14e shows the result of applying the orientation estimation algorithm at a scale of 15 to compute the gradient vectors, and a scale of 21 to smooth the gradient vectors. At these scales, the texture has two dominant directions, at 120° and 90° , as shown by the weighted angle histogram in Fig. 23e. Thus, one can see that the description of oriented textures is dependent on the notion of scale.

Another method of processing the angle intrinsic image is to obtain qualitative descriptions of the flow pattern. Thus, the flow pattern in Fig. 8 could be described in terms of the four vortices present in the corners. In fact such a description is enough to reconstruct the qualitative nature of the original flow-like image, as shown in Fig. 24. We have successfully used the geometric theory of differential equations to analyze such flow-like textures [37]. Once again, we stress the fact that the angle and coherence intrinsic images form the input to the fluid-dynamic approach to analyzing flow-like textures. The orientation field is the starting point for the process of synthesizing symbolic descriptors of different oriented textures.

8. CONCLUSIONS

In this paper we presented a new algorithm for computing the orientation field for flow-like textures. The basic idea behind the algorithm is to use an oriented filter, namely the gradient of Gaussian, and perform manipula-

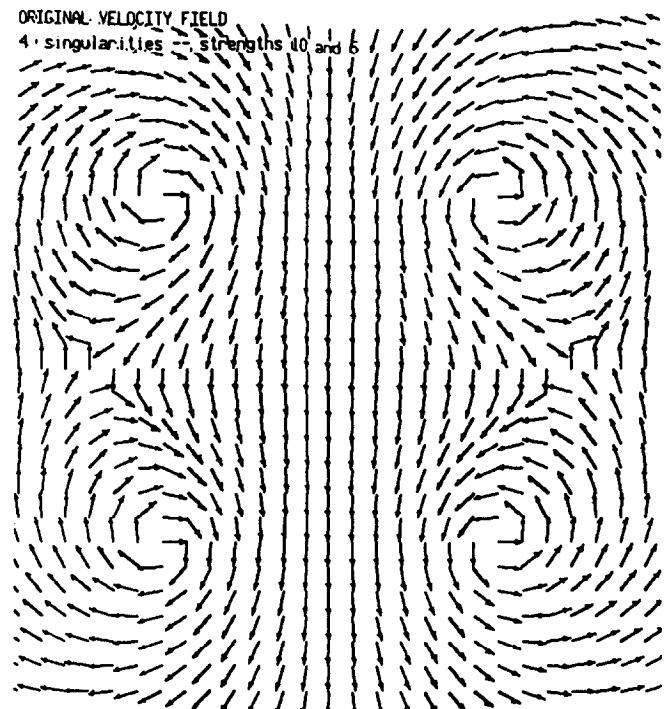


FIG. 24. A flow image created by constructing a flow field on the basis of four vortices distributed over a plane. Note how this texture appears visually similar to the texture in Fig. 8.

tions on the resulting gradient vector field. We have proved the optimality of this algorithm in estimating the orientation direction. An added strength of the algorithm is that it is simpler and has a better signal-to-noise ratio than previous approaches, because it employs fewer derivative operations. We have also proposed a new measure of coherence, which works better than previous measures.

We are convinced that the angle and coherence intrinsic images represent a necessary initial stage in the processing of oriented textures. We justified the role that the texture orientation intrinsic images play by providing results from a number of experiments. These results indicate that operations on the angle and coherence images yield useful results in many domains. Given this important role that the texture orientation intrinsic images play, our results suggest the best way to compute these images.

ACKNOWLEDGMENTS

We thank Ramesh Jain for many helpful discussions on visual texture, and Roukoz Atallah for a discussion that led to the derivation in Section 4.1. We also thank the reviewers for their comments, and especially for pointing out the proof in Section 4.2.

REFERENCES

1. R. M. Haralick, Statistical and structural approaches to texture, *Proc. IEEE* **67**, 1979, 786–804.
2. H. Wechsler, Texture analysis: A survey, *Signal Process.* **2**, 1980, 271–282.
3. W. S. MacKenzie, *Atlas of Igneous Rocks and Their Textures*, Wiley, New York, 1982.
4. V. Voort, *Applied Metallography*, Van Nostrand-Reinhold, New York, 1986.
5. R. Conners, C. W. McMillin, K. Lin, and R. E. Vasquez-Espinosa, Identifying and locating surface defects in wood: Part of an automated lumber processing system, *IEEE Trans. Pattern Anal. Mach. Intelligence* **5**, 1983, 573–582.
6. A. R. Rao and R. Jain, A visual language for interactions with computer vision systems, in *Vision* **88**, 1988.
7. M. Kass and A. Witkin, Analyzing oriented patterns, *Comput. Vision Graphics Image Process.* **37**, 1987, 362–385.
8. K. V. Mardia, *Statistics of Directional Data*, Academic Press, New York, 1972.
9. M. Levine, *Vision in Man and Machine*, McGraw-Hill, New York, 1985.
10. J. Beck, K. Pradny, and A. Rosenfeld, A theory of textural segmentation, in *Human and Machine Vision* (J. Beck, B. Hope, and A. Rosenfeld, Eds.), pp. 1–38, Academic Press, New York, 1983.
11. H. Voorhees, *Finding Texture Boundaries in Images*, MIT AI Laboratories, Technical Report 968, June 1987.
12. A. Pentland, Fractal-based description of natural scenes, *IEEE Trans. Pattern Anal. Mach. Intelligence* **6**, 1984, 661–674.
13. A. P. Witkin, Recovering surface shape and orientation from texture, *Artificial Intelligence* **17**, 1981, 17–45.
14. H. Barrow and J. Tenenbaum, Recovering intrinsic scene characterization from images, in *Computer Vision Systems* (A. Hanson and E. Riseman, Eds), pp. 3–27, Academic Press, New York, 1978.
15. J. Aloimonos and M. J. Swain, Shape from texture, in *Int. Joint Conf. Artificial Intelligence*, pp. 926–931, Aug. 1985.
16. J. Aloimonos and M. Swain, Shape from patterns: Regularization, *Int. J. Comput. Vision* **2**, 1988, 171–187.
17. E. N. Coleman and R. Jain, Obtaining 3-dimensional shape of textures and specular surfaces using four-source photometry, *Comput. Graphics Image Process.* **18**, 1982, 309–328.
18. B. Julesz and J. R. Bergen, Textons: The fundamental elements in preattentive vision and perception of textures, *Bell Syst. Tech. J.* **62**, No. 6, 1983, 1619–1645.
19. D. Marr, Early processing of visual information, *Philos. Trans. R. Soc. London Ser. B* **275**, 1976, 483–524.
20. K. Kanatani, Detection of surface orientation and motion from texture by a stereological technique, *Artificial Intelligence* **23**, 1984, 213–237.
21. F. Tomita, Y. Shirai, and S. Tsuji, Description of textures by a structural analysis, in *Int. Joint Conf. Artificial Intelligence*, pp. 884–889, 1979.
22. F. M. Vilnrotter, R. Nevatia, and K. E. Price, Structural analysis of natural textures, *IEEE Trans. Pattern Anal. Mach. Intelligence* **8**, 1986, 76–89.
23. S. W. Zucker and P. Cavanaugh, *Subjective Figures and Texture Perception*, Technical Report TR-85-2R, Computer Vision and Robotics Laboratory, Department of Electrical Engineering, McGill University, 1985.
24. S. W. Zucker and D. Terzopoulos, Finding structure in co-occurrence matrices for texture analysis, in *Image Modeling* (A. Rosenfeld, Ed.), pp. 423–445, Academic Press, New York, 1981.
25. S. Brand, *The Media Lab: Inventing the Future at MIT*, Viking, New York, 1987.
26. A. C. Bovik, M. Clark, and W. S. Geisler, Multichannel texture analysis using localized spatial filters, *IEEE Trans. Pattern Anal. Mach. Intelligence*, **12**, 1990, 55–73.
27. J. G. Daugman, Pattern and motion vision without Laplacian zero crossings, *J. Opt. Soc. Am. A* **5**, 1988, 1142–1148.
28. E. S. Meieran, P. A. Flinee, and J. R. Carruthers, Analysis technology for VLSI fabrication, *Proc. IEEE* **75**, 1987, 908–955.
29. *Surface Texture (Surface Roughness, Waviness, and Lay)*, The American National Standards Institute, 1985, ANSI/ASME B46.1-1985.
30. W. C. Leslie, *The Physical Metallurgy of Steels*, Hemisphere, New York, 1981.
31. A. R. Rao, *A Taxonomy for Texture Description and Identification*, PhD thesis, Electrical Engineering and Computer Science Department, University of Michigan, Ann Arbor, 1989; Springer-Verlag, Berlin/New York, 1990.
32. I. Iwasaki and H. Tanaka, On the flow visualization and turbulent measurement on the ripple models, in *Proceedings, Third International Symposium on Flow Visualization*, pp. 681–685, 1983.
33. W. Yang, Preface, in *Proceedings, Third International Symposium on Flow Visualization*, pp. 681–685, 1983.
34. K. Imaichi and K. Ohmi, Quantitative flow analysis aided by image processing of low visualization photographs, in *Proceedings, Third International Symposium on Flow Visualization*, pp. 365–369, 1983.
35. Fractographic features revealed by light microscopy, in *Metals Handbook, 8th ed.*, Vol. 9, *Fractography and Atlas of Fractographs*, American Society for Metals, Metals Park, OH.
36. A. Spry, *Metamorphic Textures*, Pergamon, Elmsford, NY, 1969.

37. A. R. Rao and R. Jain, The analysis of oriented textures through phase portraits, in *Int. Conf. Pattern Recognition*, June 1990.
38. J. F. Canny, A computational approach to edge detection, *IEEE Trans. Pattern Anal. Mach. Intelligence* **8**, 1986, 679–698.
39. B. G. Schunck, *Gaussian Filters and Edge Detection*, Research Publication GMR-5586, Computer Science Department, General Motors Research Laboratories, Oct. 1986.
40. M. Van Dyke, *An Album of Fluid Motion*, Parabolic Press, Stanford, CA, 1982.
41. P. Brodatz, *Textures: A Photographic Album for Artists and Designers*, Dover, New York, 1966.
42. D. Marr and E. C. Hildreth, Theory of edge detection, *Proc. R. Soc. London Ser. B* **207**, 1980, 187–217.
43. D. Marr, *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*, Freeman, San Francisco, 1982.
44. R. Szymani and K. McDonald, Defect detection in lumber: State of the art, *Forest Products J.* **31**, No. 11, 1981, 24–44.
45. H. Core, W. Cote, and A. Day, *Wood Structure and Identification*, Syracuse Univ. Press, Syracuse NY, 1979.
46. A. Panshin, C. D. Zeeuw, and H. Brown, *Textbook of Wood Technology*, McGraw-Hill, New York, 1964.
47. W. Erteld, H. Mette, and W. Achterberg, *Defects in Wood*, Hill, London, 1964.
48. B. Meylan and B. Butterfield, *The Structure of New Zealand Woods*, New Zealand Dept. of Scientific and Industrial Research DSIR Bulletin 222, 1978, Fig. 682.
49. A. R. Rao and R. Jain, Quantitative measures for surface texture description in semiconductor wafer inspection, in *SPIE Conf. Microlithography, 1990, San Jose*.
50. H. Tamura, S. Mori, and T. Yamawaki, Textural features corresponding to visual perception, *IEEE Trans. Syst. Man Cybernetics*, **8**, 1978, 460–473.
51. S. W. Zucker, Early orientation selection: Tangent fields and the dimensionality of their support, *Comput. Vision Graphics Image Process.* **32**, 1985, 74–103.