CVGIP: IMAGE UNDERSTANDING
Vol. 58, No. 3, November, pp. 383–398, 1993

Direct Estimation and Error Analysis for Oriented Patterns*

CHIAO-FE SHU AND RAMESH C. JAIN

Artificial Intelligence Laboratory, Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, Michigan 48109-2110

Received December 14, 1992; accepted March 11, 1993

This paper presents an estimation algorithm and error analysis for single linear oriented pattern in images. The estimation is formulated in terms of minimizing an objective function, using the Lagrange multiplier rule. No specific noise model is assumed. The estimation algorithm uses the intensity image of a flow pattern and directly determines a symbolic description of the pattern. No preprocessing or enhancement is needed on the intensity image or any intermediate data. This results in an efficient computational algorithm. We show that it is feasible to directly compute relative divergence, curl, and deformation from the intensity image of an oriented flow pattern. These relative properties are further used for identification of the type of pattern in the intensity image. Since an oriented pattern is corrupted by noise and is distorted to some degree from a linear flow pattern, quality measures of the estimation are proposed. The sampling mean, sampling variance, and energy of noise are computed to characterize its distribution. A closed-form condition number is used to measure the vulnerability of an estimated critical point position to noise perturbation. We show results for several experiments on fluid flow images and wafer defect patterns. © 1993 Academic Press, Inc.

1. INTRODUCTION

Vector fields appear in many different contexts in computer vision. Texture, optical flow, and fluid flow analysis are common situations in which an algorithm is developed to recognize the type of orientation pattern to recover pertinent information. Zucker [21] has shown that in our visual system, the orientation field plays a very important role in inferring the physical structure information from oriented patterns. The orientation field is a kind of vector field which has ±π ambiguity [11, 6] in phase angle. Some electrophysiological [17, 18] studies have yielded evidence that there are some cells (MSTs and FSTs) in the primate visual system that are selectively sensitive to expansion, contraction, and rotation or deformation of the orientation field. Thus, these cells are sensitive to mathematical properties of an orientation field: curl, divergence, and deformation. Some psychophysical experiments [19] also show the limited ability of the primate visual system to extract two different flow fields simultaneously.

This paper presents an approach for recognizing the class of a pattern formed by an orientation field based on the characteristic parameters. Algorithms to determine the class and its parameters of an orientation field are useful in many applications. Texture is one of the most of important properties of natural objects, rigid or nonrigid. Analyzing textures [20, 6, 4, 11] and extracting properties of textures have been important issues for computer vision researchers. We pick oriented textures [11] as our study subject, which is useful for fluid flow analysis, wafer inspection, wood processing, etc.

Recently, fluid flow visualization techniques [5, 1] have been intensively studied in order to help engineers understand fluid mechanics. However, most of these studies concentrate on the representation and display techniques using known fluid flow equations. The problem of estimating fluid flow equations from the measured optical fluid flow images is not addressed. This technique will help scientists and engineers to measure the complete flow field and extract important information.

The fluid flow images, acquired from the fluid dynamic experiments mentioned above or from remotely sensed cloud or Arctic floe images [9], wood knot images, resist gel defect images for wafer inspection, etc., are considered as kinds of oriented textures [11]. The meaning of a critical point of an oriented texture is different for different applications. The critical points in images of fluid flow experiments and remotely sensed clouds represent turbulence; in wood and wafer inspection, they represent defects.

Algorithms [6, 10] for computing local flow orientations for oriented textures were developed recently. A two-dimensional first-order phase portrait model [11, 12] was proposed to aggregate those local orientation fields. Rao and Jain [11] first used a nonlinear least-squares algorithm to estimate the parameters of this model, but the algorithm

* This project is supported by the Semicodeuntor Research Corporation, under Contract 92-MC-085.

is computationally intensive. Shu, Jain, and Quek [12] developed a robust isotangent-based algorithm to deal with outliers. Since this algorithm needs to sample orientation between zero and π, it does not use all the available data points. Therefore, an optimal solution is not guaranteed. Shu and Jain [15] further developed the weighted total least-squares estimator, which compensates the sensitivity of a least-squares estimator to noise by inversely weighing data points which are sensitive to noise. Their simulated results [15] showed that this estimator is robust and can tolerate Gaussian noise of zero mean and one radian variance. The real images were also tested.

The scheme for classifying flow patterns based on the two-dimensional first-order phase portrait were studied first by Rao and Jain [11]. They have defined six elementary flow patterns: *node, saddle, star node, improper node, center*, and *spiral*, based on the two eigenvalues and Jordan form. Shu and Jain [13] (see Fig. 1) suggest using the vector field properties: curl, divergence, and deformation, to classify flow patterns. This classification scheme has the same set of flow patterns as the previous approach, but is computationally efficient. They show explicit forms for computing the three vector field properties, whereas the eigenvalue-based approach requires computation of the Jordan form. In addition to the evidence from psychophysics, the divergence, curl, and deformation of a orientation field are strongly associated with physical meaning. Therefore, the classification scheme based on these properties does yield useful applications, such as computational fluid dynamics.

The algorithms used in previous approaches have five fundamental steps: *smoothing intensity image, computing local orientation field from the smoothed image, smoothing orientation field, estimating phase portrait parameters*, and *identifying the flow pattern*. Our algorithm reduces the whole process into three steps: *computing intrinsic properties* of an intensity image without using any filtering, *estimating parameters*, and *identifying the flow pattern*. This results in an efficient computational algorithm. Furthermore, the new approach removes the hidden parameters of preprocessing the original intensity image. In addition to the issue of computational efficiency, error and sensitivity issues related to previous approaches have never been addressed.

In this paper, we first examine the intrinsic properties of intensity images of an oriented pattern. The two-dimensional first-order phase portrait models, their vector field properties, divergence, curl, and deformation, and the classification scheme [15] based on these properties are briefly reviewed. We present the design of a weighted total least-squares error estimator. The sampling mean, sampling variance, and energy of noise are computed to characterize a noise. We use a closed-form condition number to measure the vulnerability of an estimated critical point position to noise perturbation. We have conducted extensive experiments on intensity images from fluid flow images and wafer defect patterns. The results are presented here.

2. INTRINSIC PROPERTIES OF A FLOW PATTERN IMAGE

Local intensity contrast and *orientation angle* are the two features of flow patterns in intensity images that humans respond to significantly. Humans respond to the pixels which have higher local intensity contrast and start to group the orientation information to form a flow pattern. Therefore, local intensity contrast and flow orientation are the two important intrinsic properties for identifying a flow pattern.

In order to compute these two intrinsic properties of a flow pattern, we first take the gradient of its intensity image. The magnitude of this gradient represents the local intensity contrast of a flow pattern, while the phase of the gradient represents the direction of the maximum intensity change. To make the flow orientation consistent with a flow image, we need to add or subtract $\pi/2$ to or from the phase angle of gradient.

Suppose that we have a $w \times h$ pixel image with only one flow pattern. We represent this image as $I(x, y)$, where $0 \leq x \leq w - 1$ and $0 \leq y \leq h - 1$.

The gradient of the intensity image $I(x, y)$ is calculated by

$$\nabla I(x, y) = (I_x(x, y), I_y(x, y)),$$

where

$$I_x(x, y) = \frac{\partial I(x, y)}{\partial x}$$

$$I_y(x, y) = \frac{\partial I(x, y)}{\partial y}.$$

The partial differential operator can be simply implemented by convolving the intensity image $I(x, y)$ with the mask

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

For convenience in representation, we introduce two notations: mag(**V**) to indicate the magnitude of the **V** and phase(**V**) to indicate the phase of the vector. Thus, for the gradient $\nabla I(x, y)$, the two parameters of its polar form can be related to intensity derivatives by

$$\text{mag}(\nabla I(x, y)) = \sqrt{I_x^2 + I_y^2} \tag{1}$$

$$\text{phase}(\nabla I(x, y)) = \tan^{-1}\left(\frac{I_y(x, y)}{I_x(x, y)}\right). \tag{2}$$

The flow orientation which is consistent with humans perception of a flow is computed by

$$\theta(x, y) = \text{phase}(\nabla I(x, y)) + \pi/2. \tag{3}$$

Therefore, Eq. (1) represents the intensity contrast of an image $I(x, y)$, and Eq. (3) gives the flow orienation. The higher the value of the magnitude Eq. (1) the more important the flow orientation at the corresponding pixel.

3. TWO-DIMENSIONAL FIRST-ORDER PHASE PORTRAITS

In this section, we briefly review the two-dimensional first-order portrait model, its properties, and the classification scheme based on its properties. Details can be found in [15].

3.1. Phase Portrait Model

A two-dimensional first-order phase portrait can be described by the following two equations in the Cartesian coordinate system:

$$v_x(x, y) = cx + dy + f \tag{4}$$

$$v_y(x, y) = ax + by + e. \tag{5}$$

Therefore the angle,

$$\theta(x, y) = \tan^{-1}\left(\frac{v_y}{v_x}\right) = \tan^{-1}\left(\frac{ax + by + e}{cx + dy + f}\right). \tag{6}$$

Equations (4) and (5) can be represented in matrix notation as

$$\mathbf{V} = \mathbf{AX} + \mathbf{B}, \tag{7}$$

where

$$\mathbf{V} = \begin{bmatrix} v_x \\ v_y \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} x \\ y \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} c & d \\ a & b \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} f \\ e \end{bmatrix}.$$

A point at which both v_x and v_y are zero is said to be a *critical point* (x_0, y_0) of the system. Thus the critical point position is the solution of

$$\mathbf{AX} + \mathbf{B} = 0. \tag{8}$$

3.2. Vector Field Properties and Oriented Pattern Classification

An interesting way [7] of looking at a vector field (Eq. (7)) is by decomposing it into

$$\mathbf{V} = \mathbf{T}(\mathbf{V}) + \frac{1}{2}\left\{\text{curl}(\mathbf{V})\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} + \text{div}(\mathbf{V})\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \text{def}(\mathbf{V})\mathbf{S}(\mathbf{V})\right\}\mathbf{X},$$

where $\mathbf{T}(\mathbf{V})$ is the translational component of the system, curl($\mathbf{V}$), curl of $\mathbf{V}$, is the antisymmetric part of the matrix $\mathbf{A}$, div($\mathbf{V}$), divergence of $\mathbf{V}$, is the symmetric part of the matrix $\mathbf{A}$ with nonzero trace, def($\mathbf{V}$), deformation of $\mathbf{V}$, is the symmetric and traceless part of the matrix $\mathbf{A}$, and $\mathbf{S}(\mathbf{V})$ is a traceless symmetric matrix of determinant -1 and can be diagonalized into the form:

$$\mathbf{S}(\mathbf{V}) = \mathbf{Q}^{-1}\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}\mathbf{Q}.$$

The translation, curl, divergence, and deformation of a vector field $\mathbf{V}$ can be described in the following explicit forms:

$$\mathbf{T}(\mathbf{V}) = \begin{bmatrix} f \\ e \end{bmatrix}$$

$$\text{curl}(\mathbf{V}) = a - d \tag{9}$$

$$\text{div}(\mathbf{V}) = \text{trace}(\mathbf{A}) = b + c, \tag{10}$$

$$\text{def}(\mathbf{V}) = \sqrt{(c - b)^2 + (a + d)^2}. \tag{11}$$

Figure 1 shows the classification scheme suggested by Shu and Jain [13]. The divergence, curl, and deformation can be computed directly from Eqs. (9), (10), and (11).

4. ESTIMATION OF ORIENTED PATTERN

Intuitively visualizing a vector flow pattern, we can easily justify that the flow orientation (Eq. (6)) is sufficient for identifying the type of a vector flow pattern. A theorem [15] says that *the phase information of an orientation field is sufficient to classify an oriented pattern*. This theorem further supports our intuition.

Given an intensity image $I(x, y)$ of a flow pattern, its orientation field computed from Eq. (3) can be described by Eq. (6). In order to simplify the functional form, let $\tan\theta = \zeta$. The reorganized form of Eq. (6) becomes

$$(a - c\zeta)x + (b - d\zeta)y + (e - f\zeta) = 0. \tag{12}$$

We can directly estimate the parameters by using the

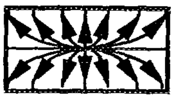
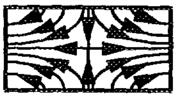
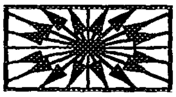
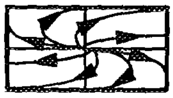
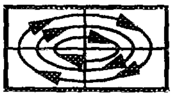
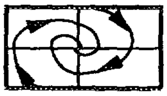
Case	Type Name	Phase Portrait
$(\mathrm{def}(\mathbf{V}))^2 > (\mathrm{curl}(\mathbf{V}))^2$ and $\lvert\mathrm{div}(\mathbf{V})\rvert > \sqrt{(\mathrm{def}(\mathbf{V}))^2 - (\mathrm{curl}(\mathbf{V}))^2}$	Node	
$(\mathrm{def}(\mathbf{V}))^2 > (\mathrm{curl}(\mathbf{V}))^2$ and $\lvert\mathrm{div}(\mathbf{V})\rvert < \sqrt{(\mathrm{def}(\mathbf{V}))^2 - (\mathrm{curl}(\mathbf{V}))^2}$	Saddle	
$(\mathrm{def}(\mathbf{V}))^2 = (\mathrm{curl}(\mathbf{V}))^2 = 0$ and $\mathrm{div}(\vec{\mathbf{V}}) \neq 0$	Star Node	
$(\mathrm{def}(\mathbf{V}))^2 = (\mathrm{curl}(\mathbf{V}))^2 \neq 0$ and $\mathrm{div}(\mathbf{V}) \neq 0$	Improper Node	
$(\mathrm{def}(\mathbf{V}))^2 < (\mathrm{curl}(\mathbf{V}))^2$ and $\mathrm{div}(\vec{\mathbf{V}}) = 0$	Center	
$(\mathrm{def}(\mathbf{V}))^2 < (\mathrm{curl}(\mathbf{V}))^2$ and $\mathrm{div}(\mathbf{V}) \neq 0$	Spiral	

FIG. 1. The classification of flow patterns based on vector field properties.

triplet data (x_i, y_i, ζ_i) and Eq. (12), where (x_i, y_i) is the coordinate of a pixel and ζ_i is the tangent of its orientation. The optimal least square estimator is one that minimizes the cost function,

$$\sum_{i=0}^{n-1} (ax_i + by_i - c\zeta_i x_i - d\zeta_i y_i + e - f\zeta_i)^2,$$

and subject to the constraint: $\sqrt{a^2 + b^2 + c^2 + d^2} = 1$, where n is the total number of triplet data used to estimate the parameter set (a, b, c, d, e, f).

In order to reliably identify an oriented pattern, we use the magnitude of the gradient of an intensity image and the sensitivity of its phase to noise to weigh data points. Since a tangent function is not uniformly sensitive to noise, we weight each observed data by the inverse of the derivative of the tangent function. Further, because higher contrast data points are usually perceived stronger by human visual system, we give importance to a data point based on the magnitude of intensity gradient. Therefore, we formulate the following weighted total least square error estimator:

$$\sum_{i=0}^{n-1} w_i^2(ax_i + by_i - c\zeta_i y_i - d\zeta_i y_i + e - f\zeta_i)^2, \qquad (13)$$

where

$$w_i(x_i, y_i) = \frac{\mathrm{mag}(\nabla I(x_i, y_i))}{255 \times \sqrt{2}} \cos\left(\mathrm{phase}(\nabla I(x, y)) + \frac{\pi}{2}\right),$$

$$\zeta_i(x_i, y_i) = \tan\left(\mathrm{phase}(\nabla I(x_i, y_i)) + \frac{\pi}{2}\right)$$

and subject to the constraint $\sqrt{a^2 + b^2 + c^2 + d^2} = 1$, where n is the total number of triplet data and $255 \times \sqrt{2}$ is the maximum gradient magnitude.

We can solve the previous constrained optimization as minimizing the cost function

$$\mathscr{C} = (\Omega_4 \mathbf{L}_4 + \Omega_2 \mathbf{L}_2)^{\mathrm{T}}(\Omega_4 \mathbf{L}_4 + \Omega_2 \mathbf{L}_2) + \lambda(\mathbf{L}_4^{\mathrm{T}} \mathbf{L}_4 - 1),$$

where

$$\mathbf{L}_4 = \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}, \quad \mathbf{L}_2 = \begin{bmatrix} e \\ f \end{bmatrix}, \quad \Omega_2 = \begin{bmatrix} w_0 & -\zeta_0 w_0 \\ w_1 & -\zeta_1 w_1 \\ \vdots & \vdots \\ w_{n-1} & -\zeta_{n-1} w_{n-1} \end{bmatrix},$$

$$\Omega_4 = \begin{bmatrix} x_0 w_0 & y_0 w_0 & -\zeta_0 x_0 w_0 & -\zeta_0 y_0 w_0 \\ x_1 w_1 & y_1 w_1 & -\zeta_1 x_1 w_1 & -\zeta_1 y_1 w_1 \\ \vdots & \vdots & \vdots & \vdots \\ x_{n-1} w_{n-1} & y_{n-1} w_{n-1} & -\zeta_{n-1} x_{n-1} w_{n-1} & -\zeta_{n-1} y_{n-1} w_{n-1} \end{bmatrix}.$$

Differentiating $\mathscr{C}$ with respect to $\mathbf{L}_4$, $\mathbf{L}_2$, and λ, and setting the derivatives to zero, we obtain.

$$\frac{\partial \mathscr{C}}{\partial \mathbf{L}_4} = 2\Omega_4^T \Omega_4 \mathbf{L}_4 + 2\Omega_4^T \Omega_2 \mathbf{L}_2 + 2\lambda \mathbf{L}_4 = 0$$

$$\frac{\partial \mathscr{C}}{\partial \mathbf{L}_2} = 2\Omega_2^T \Omega_2 \mathbf{L}_2 + 2\Omega_2^T \Omega_4 \mathbf{L}_4 = 0$$

$$\frac{\partial \mathscr{C}}{\partial \lambda} = \mathbf{L}_4^T \mathbf{L}_4 - 1 = 0$$

which yields

$$\mathbf{L}_4^T \mathbf{L}_4 = 1 \tag{14}$$

$$\mathbf{L}_2 = -(\Omega_2^T \Omega_2)^{-1} \Omega_2^T \Omega_4 \mathbf{L}_4 \tag{15}$$

$$\boldsymbol{\Psi} \mathbf{L}_4 = \lambda \mathbf{L}_4, \tag{16}$$

where

$$\boldsymbol{\Psi} = -\Omega_4^T \Omega_4 + \Omega_4^T \Omega_2 (\Omega_2^T \Omega_2)^{-1} (\Omega_2^T \Omega_4). \tag{17}$$

$\mathbf{L}_4$ is an eigenvector of the symmetric matrix $\boldsymbol{\Psi}$ and λ is its eigenvalue. From the Eqs. (14), (15), and (16), the minimal cost:

$$\mathscr{C}_{\min} = \min(|\lambda_i|). \tag{18}$$

Therefore, the eigenvector with the smallest absolute eigenvalue gives the best estimation of $\mathbf{L}_4$. We can further compute $\mathbf{L}_2$ by using Eq. (15) and the average minimal cost for each data point by using

$$\mathscr{C}_{\text{avg}} = \mathscr{C}_{\min}/n. \tag{19}$$

This value usually gives some vague sense of noise level.

5. ERROR ANALYSIS

In this section, we propose a condition number to measure the vulnerability of the estimated critical point position to noise perturbations. The sampling mean, sampling variance, and energy of the error distribution obtained by comparing the reconstructed orientation field with the original flow fields computed from the intensity image of a flow pattern are used to characterize a noise. The noise appearing in a test image is considered as a perturbation of the best estimated parameters. It affects the location of a critical point position. A larger sampling variance, sampling mean, or energy does not mean that the estimated parameter set is worse. It only means that more noise appears in the data. If an estimated parameter set has a high condition number, and the sampling mean, sampling variance, and energy are large, it means that the estimated critical point position is not reliable at all.

5.1. Sensitivity Analysis for the Location of a Critical Point Position

5.1.1. *Definition and Computation of a Condition Number*

From Eq. (8), critical point position can be computed by

$$\mathbf{X} = -\mathbf{A}^{-1}\mathbf{B}. \tag{20}$$

A condition number of the matrix $\mathbf{A}$ is defined [16] as

$$\mathbf{c} = \|\mathbf{A}\| \, \|\mathbf{A}^{-1}\|, \tag{21}$$

where $\|\mathbf{A}\|$ denotes the *norm* of a matrix $\mathbf{A}$ and

$$\|\mathbf{A}\| = \max_{\mathbf{X} \neq 0} \frac{\|\mathbf{AX}\|}{\|\mathbf{X}\|}$$

$$\|\mathbf{A}^{-1}\| = \max_{\mathbf{X} \neq 0} \frac{\|\mathbf{A}^{-1}\mathbf{X}\|}{\|\mathbf{X}\|}$$

A theorem from [16] says that *the norm of a matrix, say* $\mathbf{A}$, *is the square root of the largest eigenvalue of* $\mathbf{A}^T\mathbf{A}$, *which is a symmetric and positive definite matrix. The vector that gives the largest error is the corresponding eigenvector of the largest eigenvalue of the* $\mathbf{A}^T\mathbf{A}$, *because of the Rayleigh quotient*. This theorem provides the computational basis for computing the norm of a matrix, and thus a condition number.

Since $\mathbf{A}^T\mathbf{A}$ and $(\mathbf{A}^{-1})^T\mathbf{A}^{-1}$ are two by two matrices, which are also positive definite and symmetric; therefore a closed-form for computing their condition numbers can be obtained by the following derivations.

The two eigenvalues of $\mathbf{A}^T\mathbf{A}$ are

$$\lambda_{1,2} = \frac{(a^2 + b^2 + c^2 + d^2) \pm \sqrt{(a^2 + b^2 + c^2 + d^2) - 4(ad - bc)^2}}{2}.$$

That $\mathbf{A}^T\mathbf{A}$ is a positive definite matrix implies that λ_1 and λ_2 are both positive. Suppose $\lambda_1 \leq \lambda_2$. With the constraint:

$a^2 + b^2 + c^2 + d^2 = 1$, the largest eigenvalue (λ_2) can be obtained by

$$\lambda_2 = \frac{1 + \sqrt{1 - 4(ad - bc)^2}}{2}.$$

Therefore, the norm of matrix **A**,

$$\|\mathbf{A}\| = \sqrt{1 + \sqrt{1 - 4(ad - bc)^2}/2}.$$

Using the same derivations as the matrix **A**, we obtain the norm of matrix $\mathbf{A}^{-1}$:

$$\|\mathbf{A}^{-1}\| = \frac{1}{|bc - ad|} \times \sqrt{1 + \sqrt{1 - 4(ad - bc)^2}/2}.$$

Therefore, the closed-form for computing the condition number of matrix **A** is

$$\mathbf{c} = \frac{1}{|bc - ad|} \times \frac{1 + \sqrt{1 - 4(ad - bc)^2}}{2}.$$

5.1.2. *Sensitivity Analysis of a Critical Point Position*

If we perturb the **B** in Eq. (20) by $\delta\mathbf{B}$, as $-\mathbf{B} = \mathbf{AX}$ and $-\delta\mathbf{X} = \mathbf{A}^{-1}\delta\mathbf{B}$, from the definition of the *norm* of a matrix, we conclude that

$$\|\mathbf{B}\| \le \|\mathbf{A}\|\,\|\mathbf{X}\|$$
$$\|\delta\mathbf{X}\| \le \|\mathbf{A}^{-1}\|\,\|\delta\mathbf{B}\|.$$

By multiplying the previous two equations, the relative error of perturbing **B** satisfies

$$\frac{\|\delta\mathbf{X}\|}{\|\mathbf{X}\|} \le \mathbf{c} \times \frac{\|\delta\mathbf{B}\|}{\|\mathbf{B}\|}. \tag{22}$$

What is interesting is that the same condition number appears in Eq. (22), when we only perturb matrix **A**. If we perturb the matrix **A** by $\delta\mathbf{A}$, and $\mathbf{AX} = -\mathbf{B}$, we obtain

$$\delta\mathbf{X} = -\mathbf{A}^{-1}(\delta\mathbf{A})(\mathbf{X} + \delta\mathbf{X}).$$

From the definition of the *norm* of a matrix,

$$\|\delta\mathbf{X}\| \le \|\mathbf{A}^{-1}\|\,\|\delta\mathbf{A}\|\,\|\mathbf{X} + \delta\mathbf{X}\|$$

or

$$\frac{\|\delta\mathbf{X}\|}{\|\mathbf{X} + \delta\mathbf{X}\|} \le \|\mathbf{A}^{-1}\|\,\|\delta\mathbf{A}\| = \mathbf{c} \times \frac{\|\delta\mathbf{A}\|}{\|\mathbf{A}\|}. \tag{23}$$

From Eqs. (22) and (23), we see that the condition number is a good measure of the sensitivity of critical point position to noise perturbation. The larger the condition number, the more singular the matrix **A**, and thus, the more sensitive the solution **X** to a perturbation in the estimated parameter set [16]. Therefore, the location of a critical point is sensitive to noise if its condition number is large.

5.2. Error Distribution

Given two windows of orientation fields $\theta_1(x, y)$ and $\theta_2(x, y)$ with the same size $w \times h$, and $0 \le \theta_1(x, y), \theta_2(x, y) < \pi$, we define the phase difference at a pixel (i, j) as

$$\delta\theta(i, j) = \text{normalize}_{[-\pi/2, \pi/2]}(\theta_2(i, j) - \theta_1(i, j)),$$

where $\text{normalize}_{[-\pi/2, \pi/2]}$ is an operator which normalizes angles between $-\pi/2$ and $\pi/2$, by adding either π or $-\pi$. The reason we normalize the difference to this range is because the maximum difference between two orientation angles is $\pm\pi/2$.

Three statistical properties are useful to measure this distribution: sampling mean ($\hat{\mathbf{m}}$), sampling variance ($\hat{\mathbf{v}}$), and noise energy ($\hat{\mathbf{e}}$), where

$$\hat{\mathbf{m}} = \frac{\sum_{i=0}^{w-1}\sum_{j=0}^{h-1}(\delta\theta(i, j))}{w \times h}$$

$$\hat{\mathbf{v}} = \frac{\sum_{i=0}^{w-1}\sum_{j=0}^{h-1}(\delta\theta(i, j) - \hat{\mathbf{m}})^2}{w \times h - 1}$$

$$\hat{\mathbf{e}} = \frac{\sum_{i=0}^{w-1}\sum_{j=0}^{h-1}(\delta\theta(i, j))^2}{w \times h \times (\pi/2)^2}.$$

Because $\delta\theta$ is between $-\pi/2$ and $\pi/2$, the noise energy $\hat{\mathbf{e}}$ is always between zero and one.

Even though we do not know exactly what the noise model is, the mean tells us where its density function is located. The variance, the square of the standard deviation, tells us the spread of data about the mean. The noise energy tells us the energy of a distribution.

Applying our estimator to an intensity image, we obtain a set of parameters $(\hat{a}, \hat{b}, \hat{c}, \hat{d})$ and a critical point position $(\hat{x}_0, \hat{y}_0)$. Then the estimated phase at location (x, y) is given by

$$\hat{\theta}(x, y) = \tan^{-1}\left(\frac{\hat{a}(x - \hat{x}_0) + \hat{b}(y - \hat{y}_0)}{\hat{c}(x - \hat{x}_0) + \hat{d}(y - \hat{y}_0)}\right). \tag{24}$$

Comparing the difference between the orientation field (Eq. (3)) computed from intensity image with the reconstructed flow field $\hat{\theta}$ from Eq. (24), we obtain an error distribution. The parameter estimator of the Eq. (13) gives

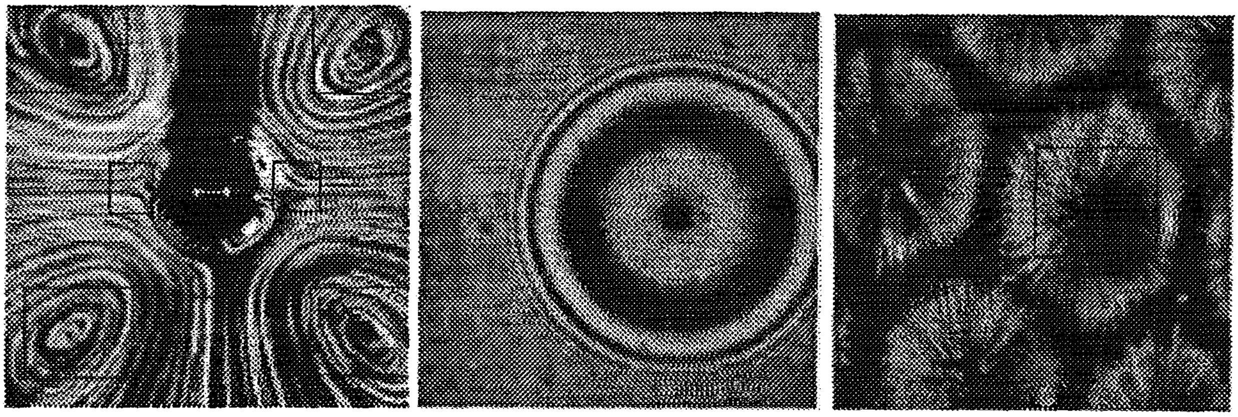


FIG. 2. Three image sets for experiments.

both the best estimate of parameters and a minimal cost. The average minimal cost from Eq. (19) is an approximate indicator of the noise level, if the noise is zero mean. However, as we are not assuming any specific noise model, this number is not sufficient to characterize any noise model. Therefore, we use the sampling mean, sampling variance, and normalized energy of the error distribution to characterize any noise distribution.

6. EXPERIMENTS

We conducted experiments on three set of images (see Fig. 2). The left image, taken from [2], in Fig. 2 is a secondary streaming induced by an oscillated cylinder; the middle image, taken from [3], in Fig. 2 is a wafer defect called a resist gel defect; the right image, taken from [2], in Fig. 2 is a surface-tension-driven convection. We use the linear part of these images to study the performance of our estimation algorithm working on real data. From observation, we found that seven linear flow patterns appear in the left image, which includes four *centers* and three *saddles*. We sample these seven flow patterns to demonstrate that our estimator is working properly for the same kind of flow pattern with different flow pattern orientations. The middle image in Fig. 2 shows a case where data points are localized to some parts of the image and other areas do not have any flow information. For this pattern, our estimator is still working properly. The right image is for demonstrating our capability to identify a *star* node uniformly corrupted with high amount of noise.

For the reader's convenience, we give names to each of these patterns. The four elliptic centers in the left image from left to right and top to down are called **ellipse.120 × 120.1**, **ellipse.120 × 120.2**, **ellipse.120 × 120.3**, and **ellipse.120 × 120.4**. The 120 × 120 means that the size is 120 by 120 pixels. The saddles in this image following the same naming rule are called **saddle.60 × 60.1**, **saddle.60 × 60.2**, and **saddle.60 × 60.3**. The middle image is called **center.120 × 120.** The only star node in the right image is called **star.120 × 120.**

Since we are using only the linear parts of these images, the whole subimage of each of these linear parts has the

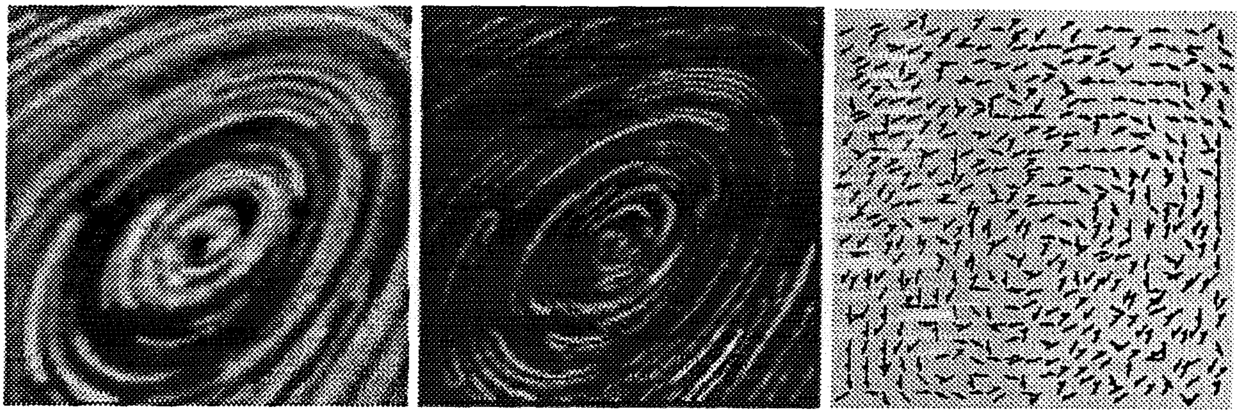


FIG. 3. Left, ellipse.120 × 120.3; middle, contrast image; right, orientation field.

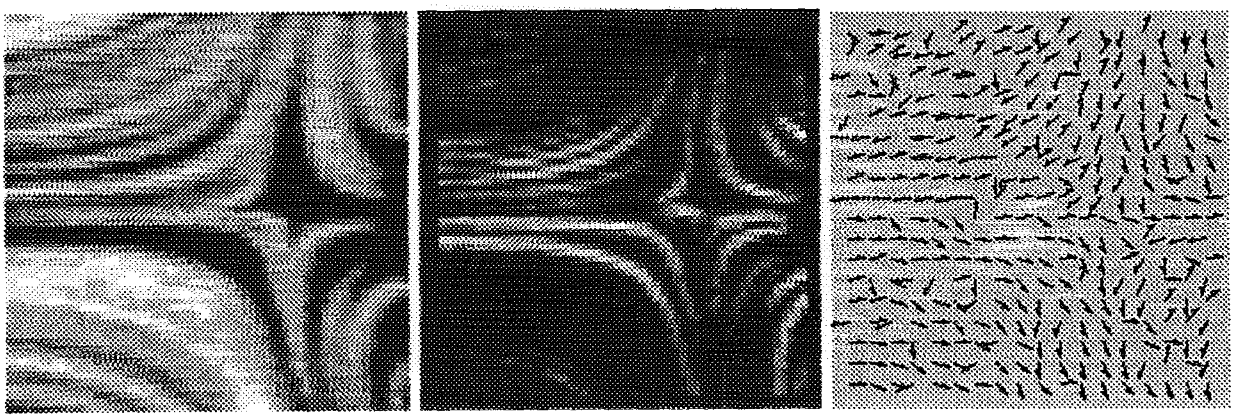


FIG. 4. Left, **saddle.60 × 60.1**; middle, contrast image; right, orientation field.

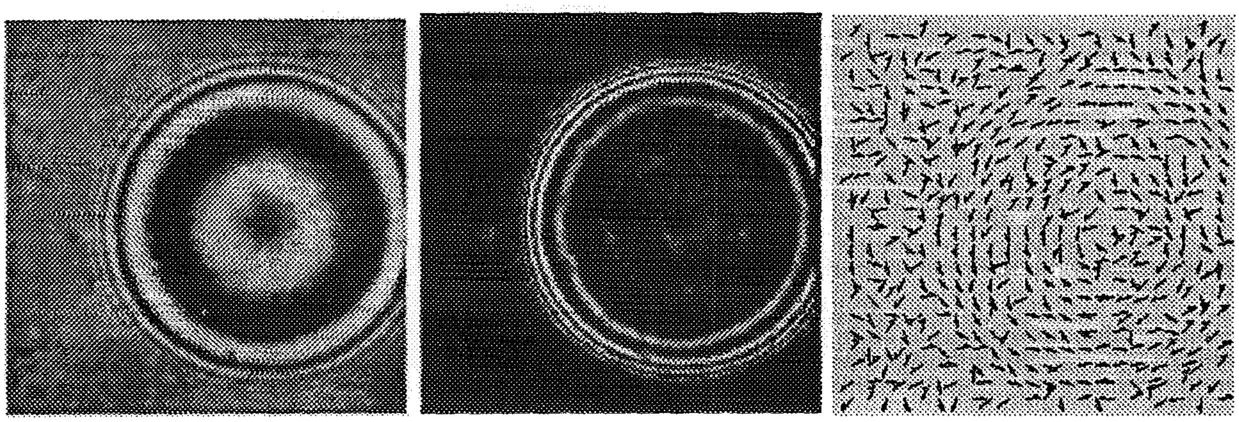


FIG. 5. Left, **center.120 × 120**; middle, contrast image; right, orientation field.

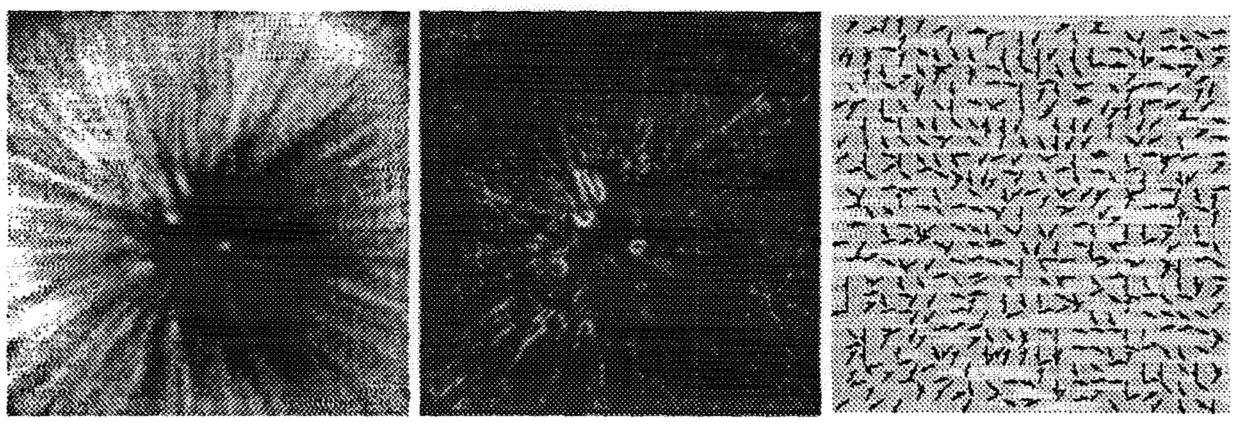


FIG. 6. Left, **star.120 × 120**; middle, contrast image; right, orientation field.

same set of parameters, which implies the same vector field properties: divergence, curl, and deformation, at every pixel in the sampled flow patterns.

In order to identify some of the flow patterns (see Fig. 1), we need to detect whether a number is zero or not. This is quite hard if there is noise and quantization errors. Therefore, a threshold (0.05) is preset for determining whether a number is equivalent to zero. If the square of a number is smaller than this threshold, we say it is *zero*.

6.1. Intrinsic Property Images of an Oriented Pattern

In this section, we show the intensity contrast image and flow orientation images of four single flow patterns sampled from Fig. 2. The left side images of Figs. 3, 4, 5, and 6 show these sample flow patterns: **ellipse.120 × 120.3, saddle.60 × 60.1, center.120 × 120,** and **star.120 × 120,** respectively. The middle images computed by Eq. (1) show the intensity contrast image, and the right images computed from Eq. (3) show the flow orientation field. No filtering has been used up to this stage. Therefore considerable amounts of noise are present in these intrinsic images, particularly **star.120 × 120.** Three pixels around the image border of each intensity contrast image are set to zero to remove the image border problem from the computing gradient of the images.

6.2. Experimental Results of the Estimation Algorithm

Tables 1–9 shows the results of estimating **ellipse.120 × 120.1, ellipse.120 × 120.2, ellipse.120 × 120.3, ellipse.120 × 120.4, saddle.60 × 60.1, saddle.60 × 60.2, saddle.60 × 60.3, center.120 × 120,** and **star.120 × 120,** respectively. The output information in these tables includes image name, image size, identified flow pattern, parameter ratio $a : b : c : d$, critical point position, relative properties: curl, divergence, and deformation, average minimal cost, a condition number, sampling mean, standard deviation, and noise energy. The reconstructed oriented field (by Eq. (24)) and the noise distribution of those images are shown in Figs. 7–15. We make the following observations regarding our experimental results:

TABLE 1

Image name	**ellipse.120 × 120.1**
Image size	120 by 120
Identified pattern type	CENTER
$a:b:c:d$	−0.6579 : 0.3337 : −0.2895 : 0.6100
Critical point	(40.033625, 51.226126)
Curl : divergence : deformation	−1.267867 : 0.044210 : 0.624927
Average minimal cost	2.247851
Condition number	2.941765
Mean (radians)	−0.048040
Std. deviation (radians)	0.555962
Noise energy (between 0 and 1)	0.126198

TABLE 2

Image name	**ellipse.120 × 120.2**
Image size	120 by 120
Identified pattern type	CENTER
$a:b:c:d$	−0.6739 : −0.3772 : 0.3060 : 0.5567
Critical point	(47.989861, 72.131528)
Curl : divergence : deformation	−1.230611 : −0.071131 : 0.693208
Average minimal cost	2.054279
Condition number	3.570028
Mean (radians)	0.123765
Std. deviation (radians)	0.561966
Noise energy (between 0 and 1)	0.134190

TABLE 3

Image name	**ellipse.120 × 120.3**
Image size	120 by 120
Identified pattern type	CENTER
$a:b:c:d$	−0.6524 : −0.3268 : 0.2970 : 0.6159
Critical point	(77.870953, 52.985933)
Curl : divergence : deformation	−1.268321 : −0.029865 : 0.624877
Average minimal cost	2.361306
Condition number	2.941228
Mean (radians)	0.107289
Std. deviation (radians)	0.573182
Noise energy (between 0 and 1)	0.137807

TABLE 4

Image name	**ellipse.120 × 120.4**
Image size	120 by 120
Identified pattern type	CENTER
$a:b:c:d$	−0.6307 : 0.3381 : −0.2883 : 0.6362
Critical point	(90.614436, 91.778658)
Curl : divergence : deformation	−1.266902 : 0.049832 : 0.626479
Average minimal cost	2.147261
Condition number	2.953464
Mean (radians)	0.000640
Std. deviation (radians)	0.553864
Noise energy (between 0 and 1)	0.124319

TABLE 5

Image name	**saddle.60 × 60.1**
Image size	60 by 60
Identified pattern type	SADDLE
$a:b:c:d$	0.0544 : −0.8215 : 0.5677 : −0.0027
Critical point	(30.092088, 44.230616)
Curl : divergence : deformation	0.057106 : −0.253812 : 1.390079
Average minimal cost	0.965492
Condition number	1.460487
Mean (radians)	0.036125
Std. deviation (radians)	0.558225
Noise energy (between 0 and 1)	0.126787

TABLE 6

Image name	**saddle.60 × 60.2**
Image size	60 by 60
Identified pattern type	SADDLE
$a:b:c:d$	0.2190 : 0.9402 : −0.2607 : −0.0002
Critical point	(28.918000, 16.411742)
Curl : divergence : deformation	0.219170 : 0.679512 : 1.220749
Average minimal cost	1.304475
Condition number	3.817806
Mean (radians)	0.159483
Std. deviation (radians)	0.647978
Noise energy (between 0 and 1)	0.180430

TABLE 7

Image name	**saddle.60 × 60.3**
Image size	60 by 60
Identified pattern type	SADDLE
$a:b:c:d$	−0.0798 : −0.4771 : 0.8710 : 0.0860
Critical point	(9.331494, 33.138916)
Curl : divergence : deformation	−0.165870 : 0.393827 : 1.348105
Average minimal cost	1.278400
Condition number	1.928203
Mean (radians)	0.088277
Std. deviation (radians)	0.513576
Noise energy (between 0 and 1)	0.110027

TABLE 8

Image name	**center.120 × 120**
Image size	120 by 120
Identified pattern type	CENTER
$a:b:c:d$	−0.7141 : 0.0000 : 0.0011 : 0.7001
Critical point	(61.183577, 75.652236)
Curl : divergence : deformation	−1.414143 : 0.001124 : 0.014061
Average minimal cost	2.059690
Condition number	1.020086
Mean (radians)	0.035038
Std. deviation (radians)	0.694068
Noise energy (between 0 and 1)	0.195722

TABLE 9

Image name	**star.120 × 120**
Image size	120 by 120
Identified pattern type	NODE
$a:b:c:d$	−0.1208 : −0.6376 : −0.7424 : −0.1663
Critical point	(62.847257, 56.850841)
Curl : divergence : deformation	0.045489 : −1.380038 : 0.305654
Average minimal cost	3.675517
Condition number	1.568589
Mean (radians)	−0.066645
Std. deviation (radians)	0.814370
Noise energy (between 0 and 1)	0.270566

- From observing the left images of Figs. 7–15, the reconstructed orientation fields are consistent with the flow patterns underlying it.
- Most of the noise distributions are like Gaussian distributions, except **saddle.60 × 60.2** and **star.120 × 120.** The error distribution of **saddle.60 × 60.2** is a sort of bimodal distribution, because some portion of its image has very little flow information and is distorted. The condition number (3.817806) is one of the highest values among our test patterns. This makes it one of the patterns which is most vulnerable to noise. Fortunately, the noise level of this image is low enough. Therefore, we can see that the reconstructed flow fields are quite consistent with the original flow pattern. The noise distribution of **star.120 × 120** is a sort of corrupted Gaussian with a significant amount of noise. This can be observed from its standard deviation (0.814370) and noise energy (0.270566), or by observing the orientation field of Fig. 6. However, its condition number (1.568589) is low enough to tolerate this amount of noise. Therefore the result is quite good for this case. One of the most stable critical point positions is **center.120 × 120,** which has the smallest condition number 1.020086 compared with the others.
- Comparing the curl, divergence, and deformation of the flow patterns **ellipse.120 × 120.1, ellipse.120 × 120.2, ellipse.120 × 120.3,** and **ellipse.120 × 120.4** with those of **center.120×120,** we found that although they are all classified as **CENTER,** the elliptic shape of a flow pattern is contributed by the deformation property. Therefore, **center.120 × 120** has very small deformation (0.014061), while the others have larger values (0.624947, 0.693208, 0.624877, 0.626479).
- The **star.120 × 120** was classified as a NODE. If we compare the reconstructed orientation field in Fig. 15 with its underlying flow pattern, they are quite consistent. From this example, we can see that STAR and NODE are quite similar to each other. Usually, STAR node is harder to obtain. For detecting a STAR, we need to detect whether deformation and curl are *zero*. Therefore, the classification problem is highly dependent on how we set the threshold value to determine whether a number is equivalent to *zero*. This threshold actually provides flexibility for a designer to partition parameter space according to his/her perception of a pattern type. Thus we are not rigidly constrained by a mathematical definition.
- The computation time starting from intensity image is less than one second for 120 by 120 images running on SUN SPARC II workstations.

7. CONCLUSION

In this paper, we first examine the intrinsic properties of intensity images of oriented patterns. We briefly review the two-dimensional first-order phase portrait model, its

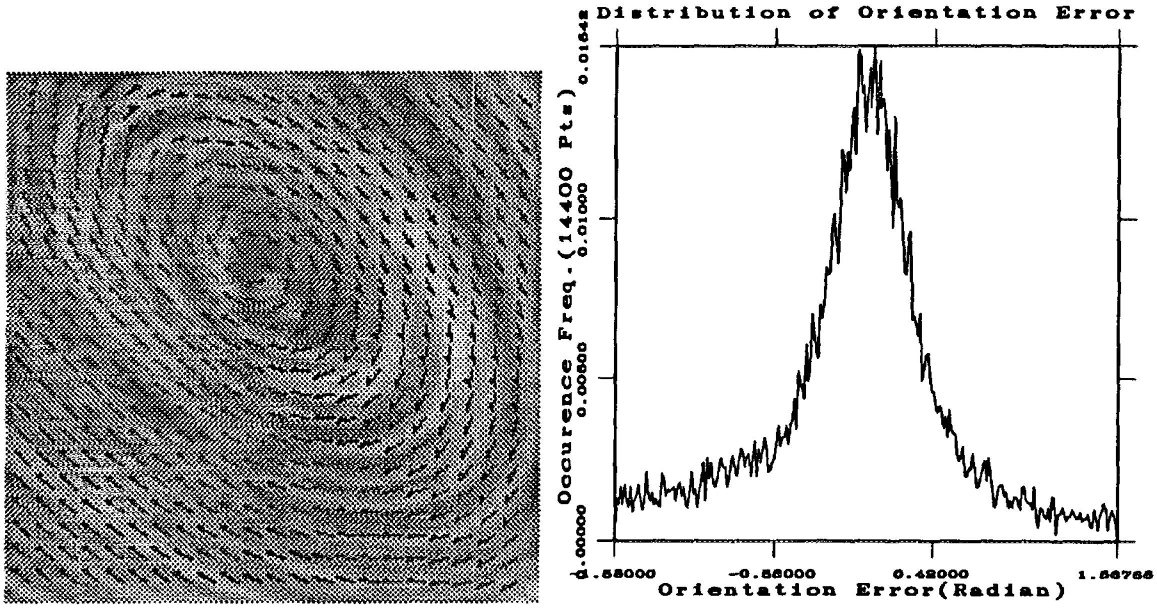


FIG. 7. Left, ellipse.120 × 120.1 with reconstructed flow field; right, error distribution.

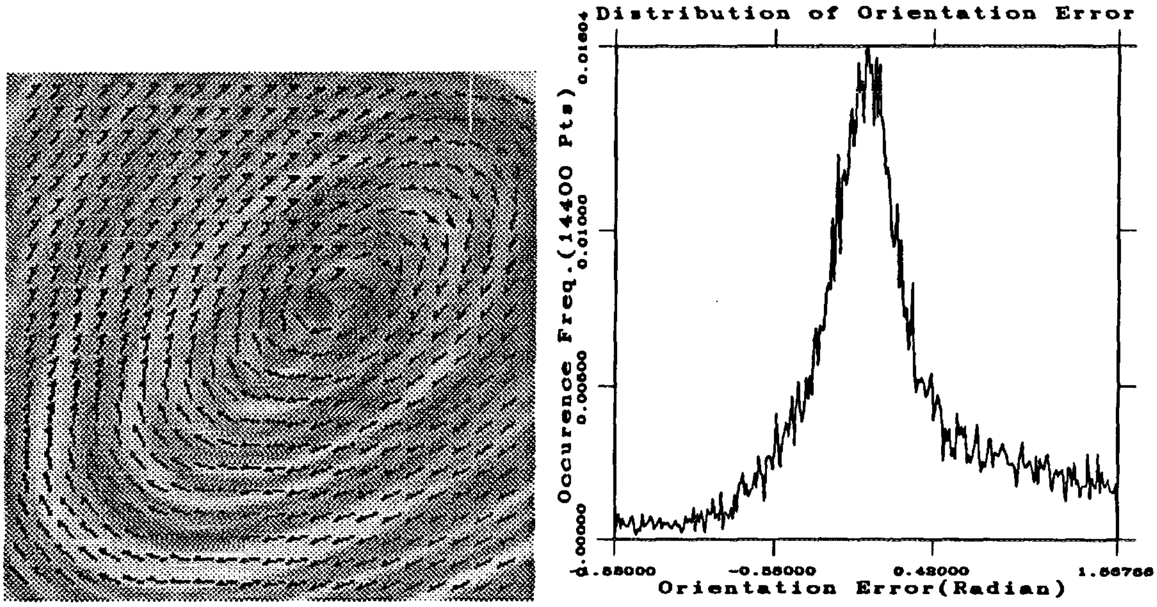


FIG. 8. Left, ellipse.120 × 120.2 with reconstructed flow field; right, error distribution.

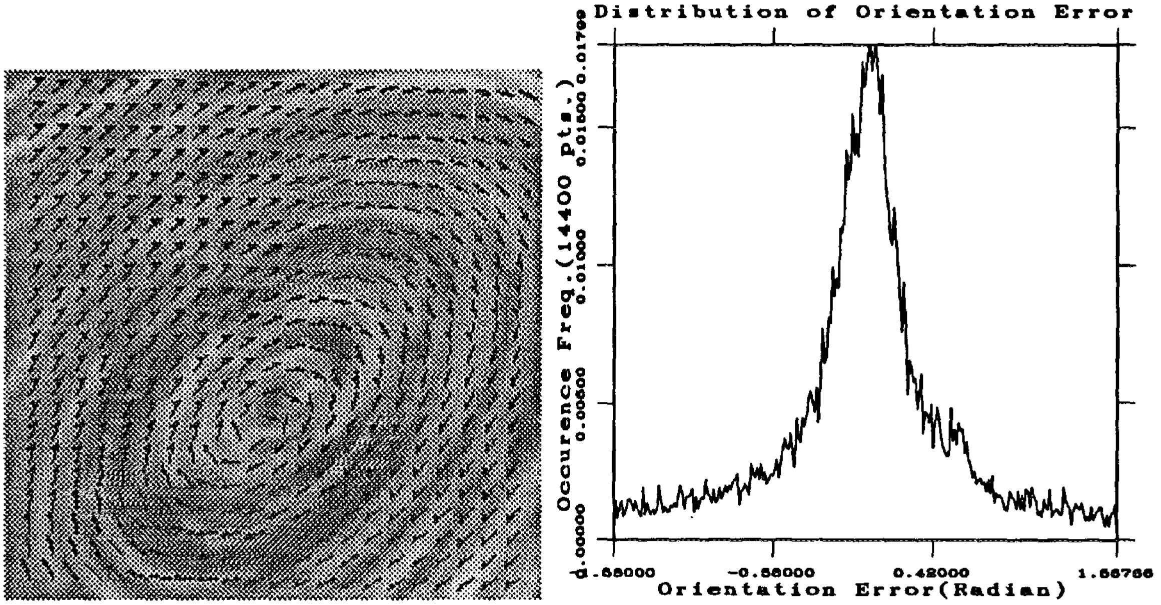


FIG. 9. Left, **ellipse.120 × 120.3** with reconstructed flow field; right, error distribution.

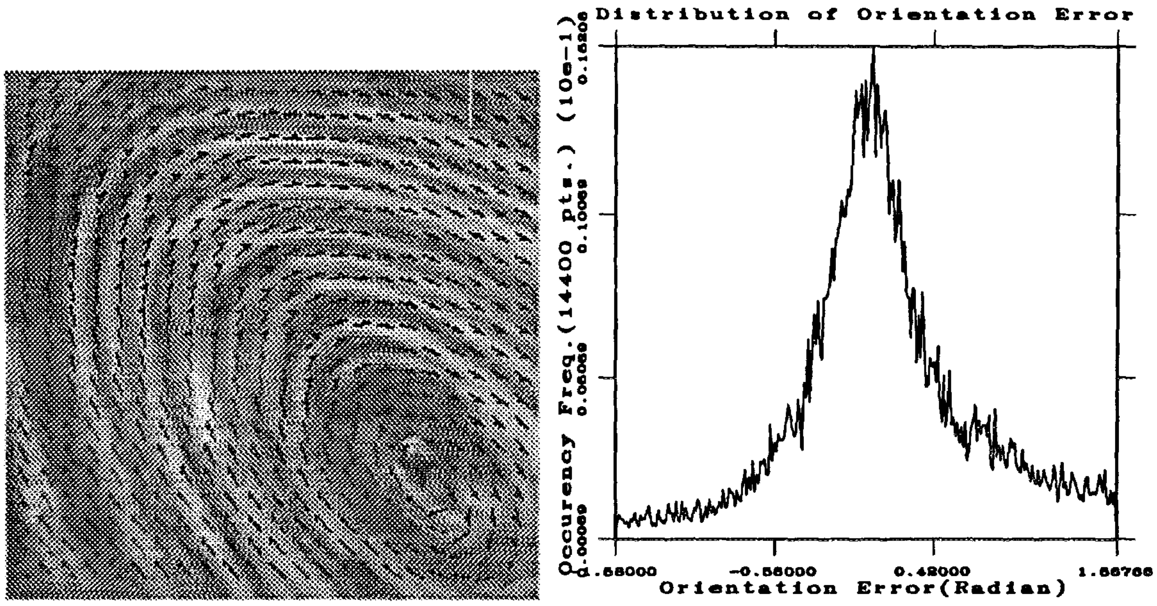


FIG. 10. Left, **ellipse.120 × 120.4** with reconstructed flow field; right, error distribution.

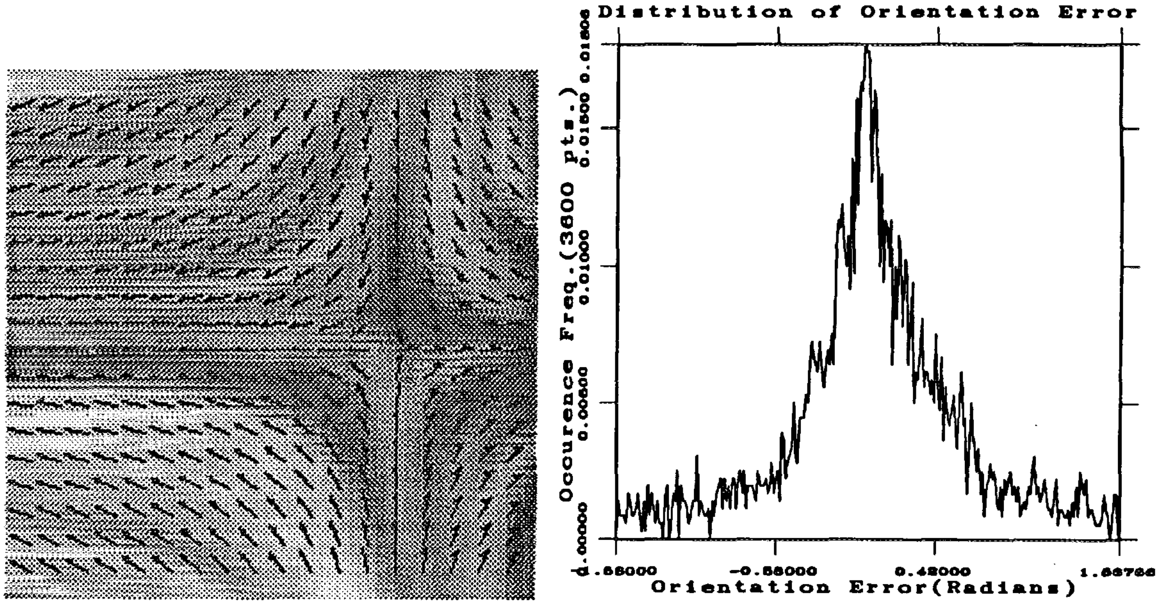


FIG. 11. Left, saddle.60 × 60.1 with reconstructed flow field; right, error distribution.

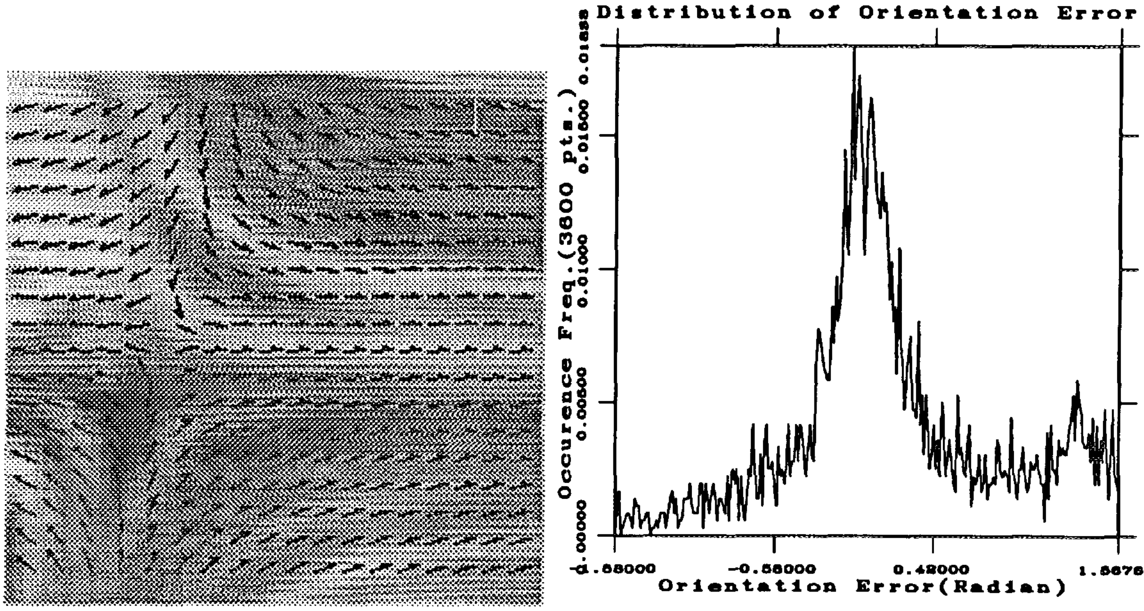


FIG. 12. Left, saddle.60 × 60.2 with reconstructed flow field; right, error distribution.

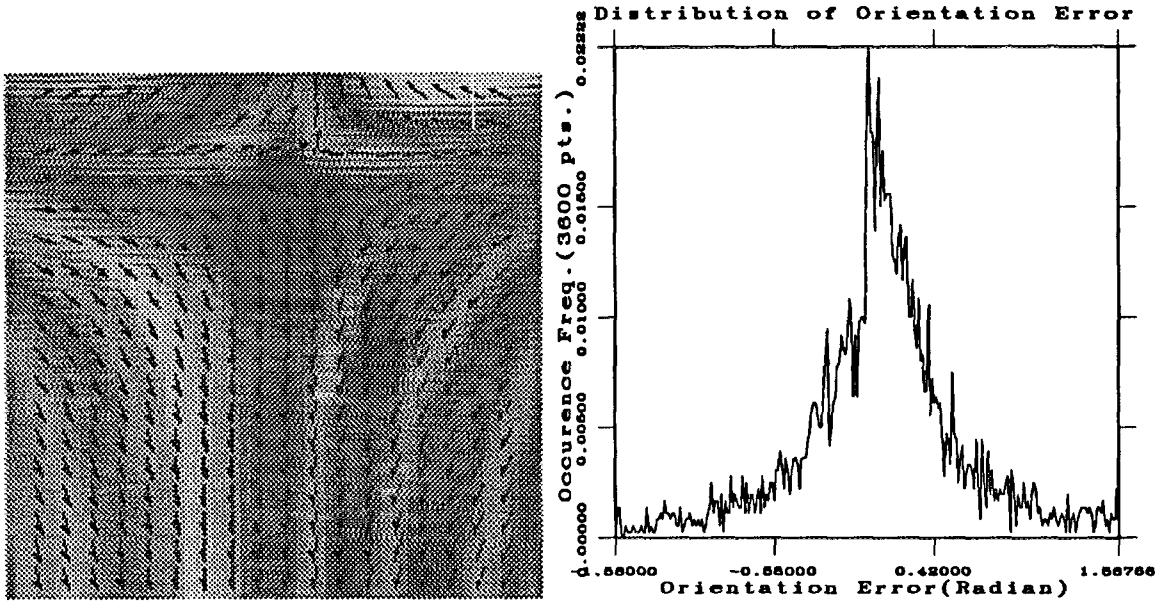


FIG. 13. Left, saddle.60 × 60.3 with reconstructed flow field; right, error distribution.

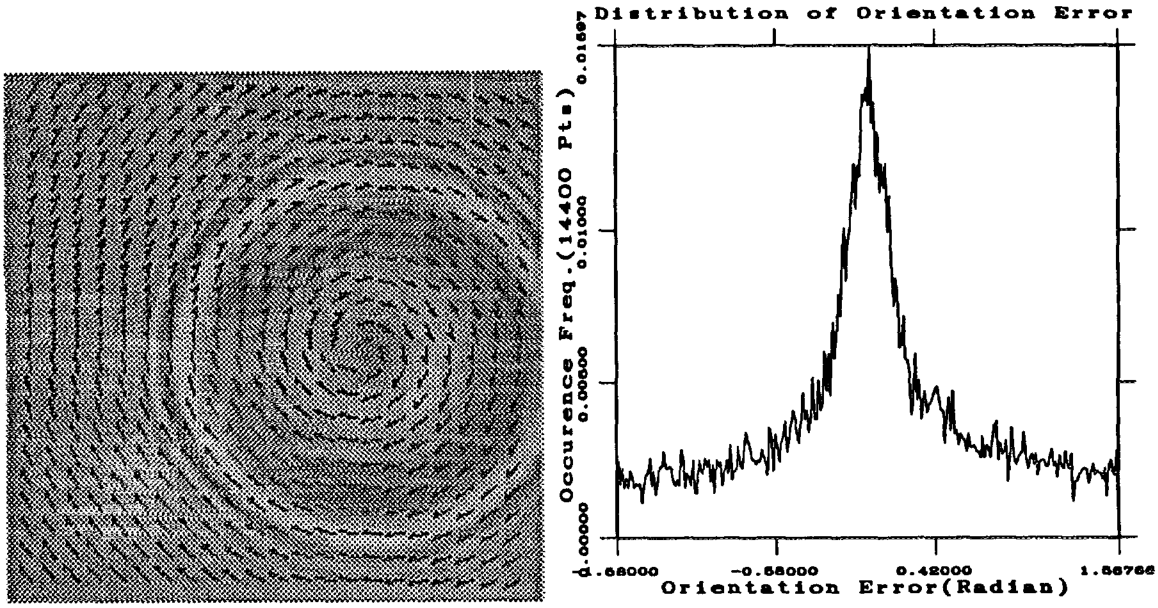


FIG. 14. Left, star.120 × 120 with reconstructed flow field; right, error distribution.

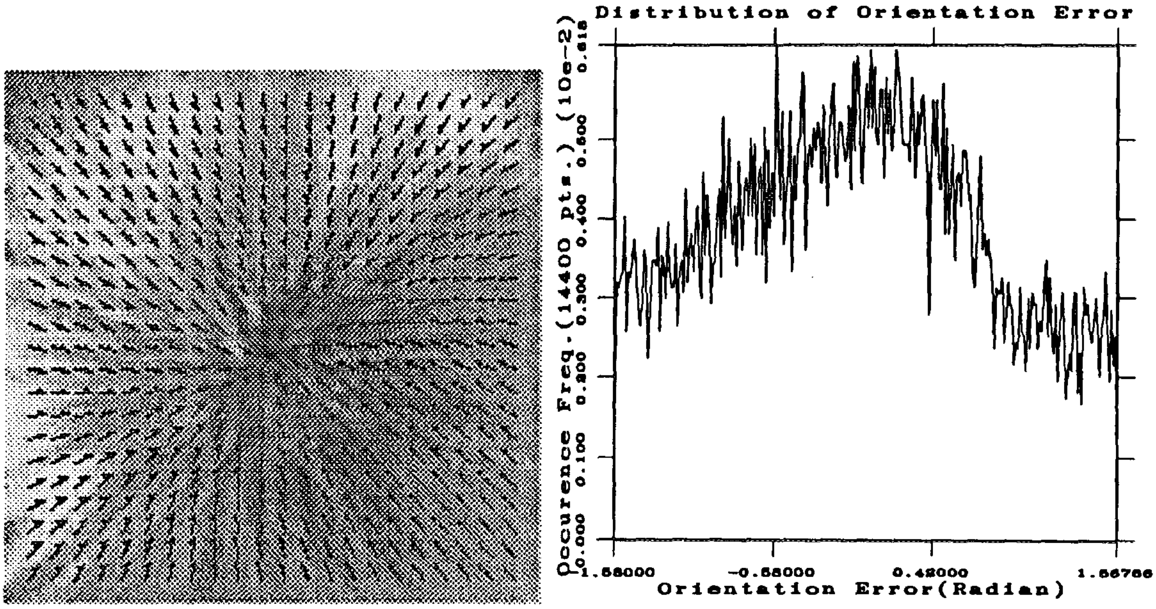


FIG. 15. Left, star.120 × 120 with reconstructed flow field; right, error distribution.

vector field properties: divergence, curl, and deformation, and the classification scheme [15] based on these properties. We minimize an objective function to estimate parameter set from the intensity image of a linear single oriented pattern, by applying the Lagrange multiplier rule. Therefore no specific noise model is assumed. The relative vector field properties are computed explicitly from the estimated parameters. These properties are further used to yield symbolic descriptions. Since an oriented pattern is corrupted by noise and is distorted to some degree from a linear flow pattern, quality measures of an estimation are proposed. The sampling mean, sampling variance, and energy of noise are computed to characterize a noise distribution. We use the condition number to measure the vulnerability of an estimated critical point position to the noise perturbation. Our experiment shows promising results. The generalization to higher order estimation is straightforward. The measured condition number and noise properties can be further used to adaptively change the window size of estimation for multiple critical point segmentation.

Vector fields appear in many different contexts in computer vision. Texture, optical flow, and fluid flow analysis are common situations in which an algorithm is developed to recognize the type of orientation pattern to recover the pertinent information. The paradigms developed in this paper are applicable to those computer vision areas which are using vector field analysis. Since the capability of our technique to tolerate a considerable amount of noise, in our laboratory, we have been using the technique developed in this paper for motion analysis from image sequences. The results are quite encouraging.

ACKNOWLEDGMENTS

We show our appreciations towards Gopal Sarma Pingali, Arun Hampapur, and David Chen for their discussion about some content. The Semiconductor Research Corporation has supported this project under Contract 92-MC-085.

REFERENCES

1. R. R. Dickinson, Interactive analysis of the topology of 4D vector fields, *IBM J. Res. Develop.* **35**(1/2), 1991, 59–66.
2. M. V. Dyke, *An Album of Fluid Motion*, Parabolic Press, Stanford, CT, 1982.
3. D. J. Elliott, *Integrated Circuit Mask Technology*, McGraw Hill, New York, 1985.
4. R. M. Haralick, Statistical and structural approaches to texture, *Proc. IEEE* **67**, 1979, 786–804.
5. J. Helman and L. Hesselink, Representation and display of vector field topology in fluid flow data sets, *Computer*, August 1989, 27–36.
6. M. Kass and A. Witkin, Analyzing oriented patterns, *Comput. Vision Graphics Image Process.* **37**, 1987, 362–385.
7. J. J. Koenderink and A. J. van Doorn, Local structure of movement parallax of the plane, *J. Opt. Soc. Am.* **66**(7), 1976, 717–723.
8. W. Merzkirch, *Flow Visualization*, Academic Press, New York, 1974.
9. S. Petterssen, *Weather Analysis and Forecasting. Vol. 1. Motion and Motion Systems*, McGraw-Hill, New York, 1956.
10. R. Rao and B. Schunck, Computing oriented texture fields, in *Proceedings, Conf. Computer Vision and Pattern Recognition, 1989.*
11. R. Rao and R. Jain, Computerized flow field analysis: Oriented texture fields, *IEEE Trans. Pattern Anal. Mach. Intell.* **14**, 1992, 693–709.
12. C. Shu, R. Jain, and F. Quek, A linear algorithm for computing the phase portraits of oriented textures, in *IEEE Conf. Proceedings, on Computer Vision and Pattern Recognition, June 3–6, 1991*, pp. 352–357.

13. C. Shu and R. Jain, Vector field analysis for oriented patterns, in *IEEE Conf. Proceedings, Computer Vision and Pattern Recognition, June 15–18, 1992*, pp. 673–676.

14. C. Shu and R. Jain, Vector field analysis for oriented patterns, in *IEEE Conf. Proceedings, Computer Vision and Pattern Recognition, June 15–18, 1992*, pp. 673–676.

15. C. Shu and R. Jain, Vector field analysis for oriented patterns, Technical Report No. CS E-TR-120-92, Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, 1992.

16. G. Strang, *Linear Algebra and Its Applications*, 2nd ed., Harcourt Brace, New York, 1988.

17. K. Tanaka, Y. Fukada, and H. Saito, Underlying mechanisms of the response specificity of expansion/contraction and rotation cells in the dorsal part of the medial superior temporal area of the macaque monkey, *J. Neurophysiol.* **62,** 1989, 643–656.

18. G. A. Orban, *The Analysis of Motion Signals and the Question of the Nature of Processing in the Primate Visual System,* Final report of the Espirit InSight project, 1992.

19. B. De Bruyn and G. A. Orban, Segregation of spatially superimposed optical flow components, *Investigat. Ophthamol. Visual Sci.* **32,** 1991, 956.

20. H. Wechsler, Texture analysis: A survey, *Signal Process.* **2,** 1980, 271–282.

21. S. W. Zucker, Early orientation selection: Tangent fields and the dimensionality of their support, *Comput. Vision Graphics Image Process.* **32**(1), 1985, 74–103.