Motion Parameter Estimation from Global Flow Field Data

Robert Hummel, *Member, IEEE*, and V. Sundareswaran, *Student Member, IEEE*

***Abstract*—We present two methods for the determination of the parameters of motion of a sensor, given the vector flow field induced by an imaging system governed by a perspective transformation of a rigid scene. We assume that the flow field $V = (u(x,y), v(x,y))$ is given. Both algorithms are new, and both integrate global data to determine motion parameters. The first algorithm (the *flow circulation algorithm*) determines the rotational parameters. It uses the curl of the flow field (curl (V)), which under many conditions is approximately a linear function of the form $g(x,y) = ax + by + c$. The coefficients of the linear function, $a, b,$ and c, which may be determined by simple regression, are proportional to the desired rotational parameters of motion. Circulation values may be used in place of curl values, resulting in less noise. The second algorithm (the *FOE search algorithm*) determines the translational parameters of the motion independently of the first algorithm. This algorithm extends a recent method of Heeger and Jepson, giving a method for searching for the image focus of expansion. For every location (x_0, y_0) in the image plane, we compute a function $u \cdot (-y + y_0) + v \cdot (x - x_0)$. When (x_0, y_0) is located at the focus of expansion, this function will be a quadratic polynomial (of a special form). We suggest several methods for determining when the function has the appropriate form; one method involves filtering the function by a collection of circular-surround zero-mean receptive fields. The other methods project the function onto a linear space of quadratic polynomials and measures the distance between the two functions. The error function for the first two methods is a quadratic polynomial of the candidate position, yielding a very rapid search strategy.**

***Index Terms*—Egomotion, flow field analysis, focus of expansion, motion estimation, rotational velocity.**

I. Introduction

We CONSIDER the problem of determining the parameters of motion of a sensor moving about a fixed environment imaging the scene by means of perspective projection onto an image plane. We assume the existence of a collection of image feature sensors that are sensitive to values that depend on the induced image vector flow field. In particular, our algorithms will depend on feature values obtained by measuring the circulation (or vorticity) of the flow field about collections of closed curves in the image plane or by measuring the component of velocity about circular paths in the image. The desired instantaneous motion parameters are the translational components of motion $\boldsymbol{v} = (v_1, v_2, v_3)$ and the instantaneous rotational parameters of motion $\boldsymbol{\omega} = (\omega_1, \omega_2, \omega_3)$, as depicted in Fig. 1. The induced vector flow field on the image will also depend on the depths to the observed surfaces; if the point (X, Y, Z) is observed at the image location (x, y), then the function $Z(x, y)$ is also unknown and must be determined from the feature data. Although our algorithms will yield approximate depth values $Z(x, y)$, our emphasis is on the determination of the motion parameters $\boldsymbol{v}$ and $\boldsymbol{\omega}$, rather than on the determination of the structure of the observed scene.

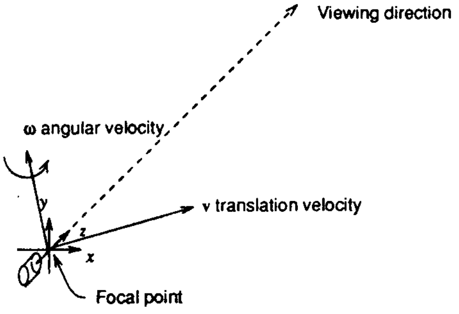


Fig. 1. Translation and rotation of the imaging sensor.

Our theme is that algorithmic methods for extraction of values must be based on global data and must integrate, through regression analysis, values obtained throughout the image. Local image intensity values, local flow field values, and especially derivatives of these values are pointwise meaningless due to noise and inaccuracies. Therefore, for example, the only hope we have of obtaining structural information from the flow field is to first extract the motion parameters, which can be obtained from global integrative processing.

The algorithms that we will present are extremely simple and qualify as *connectionist* approaches in that the motion parameters are obtained through linear and simple nonlinear operations on the feature data without sophisticated algorithmic processing. More importantly, they use *global information*, i.e., vector flow field information wherever it is available throughout the image, to determine the motion parameters. Ideally, they would operate on large field-of-view images.

We present two algorithms: one is useful for determining rotational parameters of motion, and the other is useful for determining the location of the focus of expansion, which in turn is related to the translational parameters of motion. Once one has the rotational parameters, it is a relatively

Manuscript received March 6, 1991; revised November 11, 1991. This work was supported under Air Force contract F33615-89-C-1087 reference 87-02-PMRE through AFAL and ONR grant N00014-91-J-1967. Recommended for acceptance by Associate Editor E. Hildreth.

R. Hummel is with the Courant Institute of Mathematical Sciences, Department of Computer Science, and the NYU Center for Modeling and Simulation, New York University, New York, NY 10012.

V. Sundareswaran was with the Courant Institute of Mathematical Sciences, Department of Computer Science, New York University, New York, NY 10012. He is now with IRISA, Rennes, France.

IEEE Log Number 9206554.

simple matter to determine the focus of expansion; conversely, given the focus of expansion, the rotational parameters and inverse depths may be obtained as the solution to a linear set of equations, given flow velocities. However, our two algorithms operate independently, supplying complementary ways of deriving all unknowns from the flow field.

We begin by presenting summaries of the two algorithms (the first called the *flow circulation algorithm*) used to determine rotational parameters and the second (the *FOE search algorithm*) to determine the focus of expansion. In the next section, we briefly discuss the history of treatments of the problem. Section III presents the formulation and notation, and Section IV gives the derivations of the algorithms. Section V provides a synopsis and commentary on the algorithms, discussing their applicability. The next two sections present analytical and experimental analyses of the methods. Section VIII provides some conclusions.

The **flow circulation algorithm**, which determines the parameters of rotation, is an approximate method and requires that one of a number of conditions holds true. For example, if the translational velocity v is zero, then the algorithm becomes an exact method. Alternatively, the method works if there is an identifiable visible surface whose depth is constant from the sensor (planar frontal). Other conditions lead to exact results. In any case, the method is approximately correct, providing that the preponderance of visible image points arise from surface elements distant from the sensor and are oriented so that the normal is mostly toward the sensor or if the tilts of surfaces in the scene are evenly distributed. We will later specify all alternative conditions and analyze the effects of violations. The algorithm assumes that measurements of the circulation of the flow field are given around a number of cycles in the image. The circulation of the flow field is a contour integral of the vector field about a closed curve and measures the amount of "swirling" of the field about the circuit. It is equivalent to the average vorticity of the field in the region enclosed by the closed curve. Each circulation measurement is associated with a location in the image, which is the centroid of the region enclosed by the corresponding closed curve. The circulation data is used by a regression algorithm to fit a linear function of the image coordinates of the form $g(x,y) = ax + by + c$; we note that the coefficients a, b, and c will simply be weighted sums of the available circulation data. The desired rotational parameters ω_1, ω_2, and ω_3 will be directly proportional to the coefficients a, b, and c, respectively.

For the **FOE search algorithm**, we define the focus of expansion to be the point $(v_1/v_3, v_2/v_3)$ in the image domain (assuming the focal length $f = 1$) if v_3 is nonzero and a direction toward infinity (in the direction (v_1, v_2)) if v_3 is zero. Each location in the image domain and every candidate point outside of the image domain independently attempts to determine if it is located at the focus of expansion. It does this by first computing a scalar function, which is obtained from the component of flow in a circular direction about the point within the image domain. We call this the circular-component flow velocity function. It turns out that this function is always precisely a quadratic polynomial when the candidate point is at the focus of expansion. There are several ways by which the algorithm may determine whether the function is a quadratic polynomial. One method observes that certain center-surround receptive fields always yield zero results when the scalar function is of the desired form. Another method fits a quadratic polynomial to the observed data and then computes the error. At the focus of expansion, the error will be zero. A third method computes the distance between the observed data and the 3-D linear space of quadratic polynomials that can arise as a circular-component flow velocity function at the candidate point. All three methods can be formulated as a quadratic functional of the observed circular-component data. Once again, the focus of expansion is located at the point where the functional gives a zero result.

In each of the methods, a value measures the quality of the proposed focus of expansion based on the distance of the circular component function from an ideal function. In the case of the first two methods, this error value, which should be zero at the focus of expansion, varies quadratically with the position of the candidate point, that is, the error function is itself a quadratic function of position and is zero at the focus of expansion. Thus, only a few points need to independently determine whether they are located at the focus of expansion, rather than requiring an exhaustive search. After obtaining values of the error surface at a handful of points, the true location can be easily determined by fitting a quadratic surface to the error values.

II. History

There is a large body of literature on the motion parameter estimation problem. We can only begin to summarize aspects of approaches to this problem. A more complete and excellent summary is provided by Heeger and Jepson [1]. The emphasis of much prior work on motion parameter estimation has been to recover the structure of surfaces as opposed to accurate calculation of the motion parameters. An exception is the recently introduced Heeger/Jepson algorithm [2], for which the FOE search algorithm is a straightforward extension but is formulated for continuous domains and flow fields defined over a range of image points rather than for five image points. The Heeger/Jepson algorithm, in turn, extends and simplifies aspects of an earlier algorithm by Adiv [3]. The FOE search algorithm given here introduces the notion of locating the focus of expansion by determining whether the circular-component flow data is well approximated by a quadratic polynomial, which provides more insight into the workings of the Heeger/Jepson algorithm.

Each algorithm is typically driven by some fundamental calculation or observation. The Heeger/Jepson algorithm, for example, uses the fact that once the focus of expansion is fixed, the equations become linear in all other unknowns. The flow circulation algorithm, which will be given here, is based on the observation that the curl of the flow field is approximately a linear function. The FOE search algorithm uses the Heeger/Jepson observation but is more fundamentally based on the fact that the circular-component flow function about the focus of expansion is a quadratic polynomial. None of these facts are totally new observations. However, the observations have not been exploited to give simple algorithms heretofore.

There are two stages to the motion parameter determination. In the first stage, the image sequence is analyzed, and the image flow field is determined, that is, the temporally varying image sequence is converted to a time-dependent image vector field, where the vector field at any instant indicates the velocity of motion of each point on the image plane at that time. Typically, the degree of uncertainty in the derived flow velocity will vary from point to point—accurate information often requires texture or edge features in order to perform matching. In the second stage, the vector fields are used to determine the motion parameters, through a process of solving an inverse problem, where the depths to the imaged points are typically unknown parameters that must also be determined. We are concerned with the second stage.

An alternate formulation of the motion parameter estimation problem uses pairs or sets of images in sequence, but does not attempt to establish instantaneous motion fields, but rather attempts to determine correspondences and then relative 3-D rotations and transformations between the viewpoints. This approach, which is akin to stereopsis, is the approach taken by, for example, Ullman [4], Tsai and Huang [5], and more recently, Faugeras *et al.* [6]. Our attention is on the instantaneous velocity field formulation.

Both stages of the first formulation are well-studied problems. For the first stage, numerous methods have been suggested for determining the vector flow field from an image sequence, including the methods of Horn and Schunk [7], Barnard and Thompson [8], Hildreth [9], and the more recent methods of Anandan [10] and Heeger [11], which is based on fundamental work by Adelson and Bergen [12]. These methods have varying degrees of success, and it is customary to unrealistically assume that flow-field determination is a solved problem.

In the second stage, the flow parameters must be determined from the optical field. Analysis of the field, which is also called the motion parallax field, has received considerable research attention. In 1950, Gibson discussed the motion parallax field and defined and discussed the importance of the focus of expansion [13]. Even earlier, Helmholtz had noted that the image flow field contained information about the depths to the objects [14]. Subsequent work by Gibson considered further the extraction of structural information of the scene from the flow field; for example, for the case of a pilot landing a plane, there is essentially no rotational component, and therefore, the extraction of structure from the flow field due to translation is possible [15]. In a series of papers, Koenderink and van Doorn study properties of the image flow field [16], [17]. Their work begins the practice of analyzing the flow field induced on imaged surfaces of particular form, such as planar surfaces or Gaussian protuberances. The goal is to produce local measurements that are invariantly related to properties of the surface shape.

In a famous paper, Longuet-Higgins and Prazdny [18] show that in principle, the motion parameters and local surface structure of an imaged surface may be determined from the local flow field values and values of the spatial derivatives of the local flow field up to second order. The unknown parameters are the translation and rotation velocities and the surface normal at any given point. The focus of expansion is located using the residual flow field after subtracting the flow field due to rotation. There are special difficulties with the solution method when the imaged surface is planar, and it is clear that measurements of the second-order derivative, in particular, will be noisy. However, since the computations at each point should yield the same motion parameters (and varying surface normals), the computations are redundant, and thus, one can hope that stabilized algorithms are possible.

In a series of subsequent papers, Prazdny offered a variety of other algorithms for motion determination. In [19], a precursor to the FOE search algorithm given here is presented. In this algorithm, the best rotation parameters are sought such that the flow field that remains after subtracting the vector field corresponding to the rotation parameters yields a pure expansion or pure contraction field, as will occur for the flow field due to translation only. The algorithm then simply involves a nonlinear minimization. More recent work by Burger and Bhanu [20] extends the Prazdny search algorithm to search for a focus of expansion "fuzzy" region. In [21], Prazdny shows how the computation of the translational and rotational velocities can be obtained from the flow velocities at a collection of distinct points (five of them are required) by solving a system of three cubic polynomial equations in three unknowns. An iterative method is used, and a good initial guess is required. The intention is that the processing should be local, although the equations hold for any distinct set of five points. A good survey and history of results until that point is provided in a separate paper by Prazdny [22].

The idea of searching over the possible focus of expansion points, instead of searching over the rotation parameters, as in our FOE search algorithm, was introduced by Adiv [3]. Using the same error function, Heeger and Jepson have recently introduced a related algorithm for locating the focus of expansion [1], [2], [23], [24]. Since this method is directly related to our FOE search algorithm, we discuss its workings in more detail in Section III.

Nearly all researchers in motion parameter estimation realize that once one has *some* information, such as the location of the focus of expansion or the value of the rotation parameters, all other parameters are easily obtained. Hence, there is considerable motivation for separating the flow field due to rotational parameters from the flow field due to translational parameters. The idea of making use of either depth discontinuities or motion parallax of a translucent surface (such as a dusty window) has already been noted by Helmholtz [14] and mentioned by Longuet-Higgins/Prazdny [18]. In these cases, the difference or jump in the flow velocities cancels the flow component due to the rotational parameters, leaving a flow dependent only on translational parameters. Provided that there are enough such points, then the focus of expansion, and hence all other parameters, may be determined. Lawton and Rieger exploit this idea to build a system based on differences of neighboring flow velocities [25]. Unfortunately, noise tends to make this method rather unreliable. Lawton built another system that assumes that the rotational parameters are nearly zero and, thus, finds the focus of expansion by means of a "Hough transform" technique [26]. A solution method that assumes that the translational parameters are zero would be quite easy; our flow circulation method provides an exact method, for example, in this case, and other methods are straightforward.

In a series of papers, Waxman and collaborators revisited the problem of motion parameter estimation and local surface structure determination from local flow parameters (i.e., values of the flow velocities and derivatives of the flow velocities through second order). A solution method is presented for the case of planar surface patches and quadratic flow velocity fields [27], together with an analysis of the ambiguities, followed by a new method for quadric surfaces and quadratic velocity fields [28] that improves on an earlier method of Waxman and Ullman [29]. More details and extensions to binocular image flows are given in [30]. In all of these works, the structure of the surface and the motion parameters are considered to be the unknowns relative to a single image point, that is, the analysis is local. Measurements of derivatives of flow velocity values through second order must be given, in which case, it is possible to solve for local surface structure up to a second-order Taylor expansion. The curl of the flow field is one of the 12 "deformation parameters" (D_6 to be exact), and the "kinematic relation" for this parameter is precisely the equation that we require for the flow circulation algorithm. However, since the problem that Waxman addresses is the exact computation of motion parameters coupled with surface parameters based on local data (deformation data at a single point), the approximations and the global method leading to the "flow circulation algorithm" is missed. Using instead correspondences of curves, the local quadratic nature of the velocity field can be obtained, provided that a sufficient number of curves are matched, as studied by Wohn [31], [32] (who also makes use of some temporal smoothing) and more recently studied by Faugeras [33]. Solution methods based on the use of a sufficient number of correspondences of points, without involving explicit derivatives of the flow field and without explicit representation of surface parameters, are provided by Jerian and Jain [34]; their work, like that of Tsai and Huang [5], is directed for the case of determining rotation and translation parameters from correspondences, as opposed to rotation and translation *velocities* from an image velocity *flow field*.

Clearly, the surface parameters, to the extent that they can be recovered, must be based on local measurements. The motion parameters, however, are global. Most researchers have noted that their methods provide redundant computation of motion parameters, providing a test for the rigidity assumption. Unfortunately, many of the algorithms forgo the stability that can be obtained by deriving the motion parameters from an integrative approach—i.e., by making use of the constancy over the image. Of course, if one's focus is on surface parameter reconstruction, then local processing is essential. However, if one first derives the motion parameters and then uses knowledge of the motion parameters to assist in surface depth estimates, then global methods may be used for motion parameter estimation. Methods that can potentially make use of distributed information include Prazdny's [21], Adiv's [3], and Heeger/Jepson's [2].

The curl of the flow field, which is the basis of our flow circulation algorithm, has long been recognized as an important property of the motion field. However, the observation that the curl of the field is approximated by a linear function, and the use of that approximation to determine the rotational parameters, appears to be new. Koenderink and Van Doorn [16] calculate explicitly the curl (and other functionals). We begin with their computation of the curl (converted to planar image coordinates as opposed to polar coordinates) but make use of the function to solve for the rotational parameters. They instead studied the properties of these elementary fields in the case of an observer moving with respect to a plane [17], [35]. The existence of receptive fields sensitive to the curl (and also to the divergence) was hypothesised by Koenderink and Van Doorn [16] and Longuet-Higgins and Prazdny [18], but the motivation is for surface structure determination and not for global synthesis of motion parameters. Regan and Beverly [36] followed up on the hypothesis of Longuet-Higgins and Prazdny by conducting psychophysical experiments and concluded that the existence of vorticity receptors is plausible. Cell recordings in the dorsal part of the MST of macaque monkeys suggest cells that are tuned to expansion/contraction and other cells that are sensitive to rotation [37]. More recently, Werkhoven and Koenderink [38] have considered methods for directly computing flow field invariants, including the curl, from time-varying image irradiance data. The considerable interest and evidence for the importance of the curl of the flow field (or, equivalently, the circulation values) lend credence to the flow circulation algorithm presented here.

III. Formulation

In this section, we recall the equations governing the optical flow field on an image and establish notation.

A. Optic Flow Equations

For the case of perspective projection of a scene on a planar imaging sensor, the optical flow equations are well known and understood. We assume that the sensor is moving with translational velocity $\boldsymbol{v} = (v_1, v_2, v_3)$ and with rotational (angular) velocity $\boldsymbol{\omega} = (\omega_1, \omega_2, \omega_3)$ in a fixed environment. A point $\boldsymbol{X}$ in three space attached to the coordinate frame of the sensor but measured in the coordinates of the fixed coordinate frame of the sensor at a particular instant (say, time $t = 0$) satisfies $d\boldsymbol{X}/dt = \boldsymbol{v} + \boldsymbol{\omega} \times \boldsymbol{X}$ at time $t = 0$. (This formalizes the notation of Fig. 1.) We assume that the camera system has a focal length of f; therefore, the transformation from spatial coordinates to image coordinates is governed by the equations

$$x = fX/Z, y = fY/Z$$

where $(X, Y, Z) = (X(x,y), Y(x,y), Z(x,y))$ is the position in three space of the surface element imaged at the point (x, y). We assume that the objects in the scene are fixed and rigid. Derivations of the induced vector flow field may be found, for example, in [2], [18], and [39] and yield the image optical flow (u, v) at the image point (x, y) as

$$\begin{aligned} u(x,y) &= \frac{1}{Z(x,y)}[-fv_1 + x \cdot v_3] \\ &\quad + \omega_1\left(\frac{xy}{f}\right) - \omega_2\left(f + \frac{x^2}{f}\right) + \omega_3 y, \\ v(x,y) &= \frac{1}{Z(x,y)}[-fv_2 + y \cdot v_3] \\ &\quad + \omega_1\left(f + \frac{y^2}{f}\right) - \omega_2\left(\frac{xy}{f}\right) - \omega_3 x. \end{aligned} \tag{3.1}$$

As is common, we note that the flow equations may be grouped into the sum of two terms: the first term gives a flow field due to the translational components and is modulated by the inverse depths, and the second term is a flow field due to the rotational components and is independent of the depths:

$$V(x,y) = \begin{pmatrix} u(x,y) \\ v(x,y) \end{pmatrix} = V_{\boldsymbol{v}}(x,y) + V_{\boldsymbol{\omega}}(x,y) \tag{3.2}$$

with (for the case $v_3 \neq 0$)

$$\begin{aligned} V_{\boldsymbol{v}}(x,y) &= v_3 \rho(x,y) \cdot \begin{pmatrix} x - \tau \\ y - \eta \end{pmatrix}, \\ \tau &= \frac{fv_1}{v_3}, \quad \eta = \frac{fv_2}{v_3}, \\ \rho(x,y) &= \frac{1}{Z}(x,y) \end{aligned}$$

and

$$V_{\boldsymbol{\omega}}(x,y) = \omega_1 \cdot \begin{pmatrix} \frac{xy}{f} \\ f + \frac{y^2}{f} \end{pmatrix} + \omega_2 \cdot \begin{pmatrix} -f - \frac{x^2}{f} \\ \frac{-xy}{f} \end{pmatrix} + \omega_3 \cdot \begin{pmatrix} y \\ -x \end{pmatrix}.$$

Note that the flow due to the translational components has a radial structure, expanding or contracting about a "focus of expansion" at location (τ, η) and with a magnitude modulated by the distance from the focus of expansion, the component of translation in the viewing direction (v_3), and the inverse depth to each pixel $\rho(x,y) = 1/Z(x,y)$. In the case $v_3 = 0$, the situation is nearly the same, except that $V_{\boldsymbol{v}}$ is a parallel vector field:

$$V_{\boldsymbol{v}}(x,y) = -f\rho(x,y) \cdot \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \qquad (v_3 = 0)$$

in the direction of the translational velocity, modulated again by the inverse depths. In both cases, the flow due to the rotational components is the linear sum of three fixed vector fields, with the coefficient of each field taken from the respective angular velocity components.

B. Problem Statement

In the motion parameter determination problem, we are given a collection of points $\{(x_i, y_i)\}_{i=1}^N$, where the flow velocities $(u(x_i, y_i), v(x_i, y_i))$ are known. It is desired to determine the motion parameters $\boldsymbol{v}$ and $\boldsymbol{\omega}$ and the inverse depths to the points $\rho(x_i, y_i)$. It can easily be seen that there are certain ambiguities and that it suffices to determine the focus of expansion (τ, η), the rotational parameters $\boldsymbol{\omega}$, and proportional inverse depths $c \cdot \rho(x_i, y_i)$. Without additional information, this is all that can be derived. It can happen that the focus of expansion lies at infinity, in which case, its direction (toward infinity) can be determined.

C. The Heeger/Jepson algorithm

We briefly outline the method suggested by Heeger and Jepson [1]. They observe that for fixed (τ, η), the equations are linear in the remaining collection of unknowns $\boldsymbol{\omega}$, $\{v_3 \cdot \rho(x_i, y_i)\}_{i=1}^N$. We thus have $2N$ linear equations in $N+3$ unknowns in the case when (τ, η) is fixed. As long as $N \geq 4$, these are easily solved to give the least mean square error, and that error is also easily determined. The residual error in the least mean square solution can be used as a measure of the quality of the estimate (τ, η) for the focus of expansion—in noise-free circumstances, there should be zero residual error if τ and η are the correct values. Equivalently, we see that the data $\{(u(x_i, y_i), v(x_i, y_i))\}_{i=1}^N$, which is regarded as a vector in $2N$ space, should lie on an $(N + 3)$-D hyperplane defined by the fixed (τ, η); the extent that the vector lies off this hyperplane measures the noise in the data and the inexactness of the (τ, η) estimate. Heeger and Jepson then simply check a large array of possible (τ, η) values, computing the error at each such position. Where the error is minimized, they declare the correct (τ, η) to be found and the least mean square solution to the other variables to give the solution.

Our FOE search algorithm is the same algorithm, but it is derived under the assumption that the data is given on a continuum of points, rather than at N distinct points. In a certain sense, the Heeger/Jepson assumption, of a discrete collection of data, is more realistic in image processing applications. However, the projection method is then dependent on the locations of the distinct points and would have to be computed for each new collection of points. The FOE search algorithm presented here makes clearer the analytical structure of the problem and provides a method that is independent of the sampling locations, assuming a sufficient density of values are obtained. When there are only a few discrete sampling locations, the FOE search algorithm may still be used, in a modified form, and will essentially be equivalent to the Heeger/Jepson algorithm. Specifically, the circular-component functions may still be computed, but they will be defined only at the collection of sample points. The test to determine the FOE then checks whether the discrete collection of data located at the sample points corresponds to point evaluates of a quadratic polynomial.

D. The Curl and Circulation Values

It is usual to decompose the gradient of the flow field ∇V (which is a two-by-two matrix) [17]. We will not need to make use of this decomposition; however, one of the usual coefficients in that decomposition is the curl of the flow field, which is defined by

$$\operatorname{curl}(V) = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}.$$

The curl of V is also denoted $\nabla \times V$. The curl is a scalar function (or, in the case of the flow field induced on a spherical retina, the curl is a vector field lying normal to the sphere) and is easily computed. Applying the curl to (3.2) yields

$$\nabla \times V(x,y) = v_3 \cdot \left[\frac{\partial \rho}{\partial x} \cdot (y-\eta) - \frac{\partial \rho}{\partial y} \cdot (x-\tau)\right] - \frac{1}{f}[x\omega_1 + y\omega_2 + 2f\omega_3]. \tag{3.3}$$

An important property of the curl of a vector field is provided by Stoke's theorem:

$$\iint_D \operatorname{curl}(V)\,dx\,dy = \oint_{\partial D} V \cdot ds$$

where the right-hand side, which we will call the *circulation* of the flow about the circuit ∂D, denotes the contour integral of the vector field about the boundary of D. The theorem is quite general, requiring only certain differentiability conditions on V and the boundary of the domain D. We note that the contour integral is independent of the parameterization of the contour. For example, when D is a disk of radius r about an image point (x_0, y_0), the circulation may be calculated from

$$\oint_{\partial D} V \cdot ds = \int_0^{2\pi} V(x_0 + r\cos\theta, y_0 + r\sin\theta) \cdot (-r\sin\theta, r\cos\theta)d\theta. \tag{3.4}$$

Mitiche *et al.* [40] provide a discrete version of Stoke's theorem and discuss its use for smoothing flow field data given time-varying intensity image data; our needs are only for the continuous version and for smoothing *the curl* of the flow field information.

Interestingly, by using Stoke's theorem, we will not lose any precision, nor do we require any approximations, in order to take advantage of the smoothing. Although the circulation feature values are obtained more stably than would be curl (vorticity) values of the flow field, they are just as useful to the flow circulation algorithm.

IV. Derivations

In this section, we provide the technical derivations of the algorithms.

A. The Flow Circulation Algorithm

The basic idea is to show that the first term on the right-hand side of (3.3) can be neglected, and thus, the curl values lie on a linear surface defined by the function $g(x,y) = -(x\omega_1 + y\omega_2 + 2f\omega_3)/f$. We begin by computing the curl that results at a point where an analytic surface element is imaged. Suppose that at the point (x_0, y_0), the surface element (X_0, Y_0, Z_0) is imaged, and suppose that the surface is analytic and described (implicitly) by the equation

$$n_1(X - X_0) + n_2(Y - Y_0) + n_3(Z - Z_0) + \sum_{|\boldsymbol{\alpha}| \geq 2} c_{\boldsymbol{\alpha}}(X - X_0)^{\alpha_1}(Y - Y_0)^{\alpha_2}(Z - Z_0)^{\alpha_3} = 0.$$

Here, $\boldsymbol{\alpha}$ is a multi-index with integer components α_i, and $|\boldsymbol{\alpha}|$ is the order of the multi-index, i.e., the sum of the components. Any analytic surface can be described locally in this way. The tangent plane to this surface has normal $\boldsymbol{n} = (n_1, n_2, n_3)$, which we may assume is a unit normal. We set

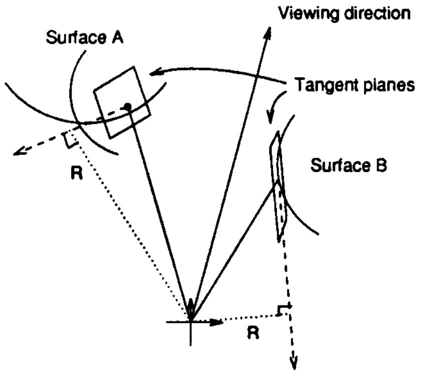


Fig. 2. Tangent planes to points on imaged surfaces.

$$R = n_1 X_0 + n_2 Y_0 + n_3 Z_0$$

which is the distance, at the point of closest approach, between the tangent plane to the surface and the focal point of the imaging system (see Fig. 2).

Then, using the imaging equations and the definition that $\rho(x,y) = 1/Z(x,y)$, it is not hard to show that

$$\frac{\partial \rho}{\partial x}(x_0, y_0) = \frac{n_1}{fR}, \frac{\partial \rho}{\partial y}(x_0, y_0) = \frac{n_2}{fR}.$$

After substitution into (3.3), we obtain

$$\nabla \times V(x_0, y_0) = v_3 \cdot \left(\frac{n_1(y_0 - \eta) - n_2(x_0 - \tau)}{fR}\right) - \frac{1}{f}(x_0\omega_1 + y_0\omega_2 + 2f\omega_3). \tag{4.1}$$

After regrouping terms and using the definitions of τ and η, we see that this is the same as

$$\nabla \times V(x_0, y_0) = \frac{-1}{f}\left[x_0\left(\omega_1 + \frac{n_2 v_3}{R}\right) + y_0\left(\omega_2 - \frac{n_1 v_3}{R}\right) + \left(2f\omega_3 + \frac{n_1 v_2 - n_2 v_1}{R}\right)\right]. \tag{4.2}$$

If we now allow (x_0, y_0) to vary and denote the point by (x,y), regarding (n_1, n_2, n_3) as a function of (x,y), we conclude from (4.1) and the equivalent (4.2) that the curl

$$\nabla \times V(x,y) \approx \frac{-1}{f}(x\omega_1 + y\omega_2 + 2f\omega_3) \tag{4.3}$$

whenever any of the following conditions holds true:

1. If $v_1 = v_2 = v_3 = 0$, then (4.3) holds with equality everywhere.
2. At any point (x,y) such that $n_1 = n_2 = 0$, (4.1) clearly becomes (4.3), with equality. The condition that $n_1 = n_2 = 0$ is equivalent to saying that the tangent plane to the surface imaged at (x,y) lies normal to the viewing direction. In particular, the condition holds true on any frontal planar surface.
3. At any point where the vector to the focus of expansion $(\tau - x, \eta - y)$ is proportional to the tilt of the tangent plane to the surface imaged at (x,y) (the tilt is defined as

(n_1, n_2)), then clearly, the first term on the right-hand side of (4.1) vanishes, and (4.3) holds with equality.

4. If the distance R from the tangent plane to the focal point is large relative to the translational velocity $||\boldsymbol{v}||$, then (4.3) holds approximately. Note that the components of $\boldsymbol{n}$ satisfy $|n_i| \leq 1$ since $\boldsymbol{n}$ is a unit normal.
5. If the rotational components of velocity $\boldsymbol{\omega}$ are quite large compared with the translational components $\boldsymbol{v}$ and assuming that the R value is bounded below, then (4.3) will hold approximately.

It is interesting to note that the R value becomes small whenever a surface lies nearby, as well as whenever a surface is oriented so that the tangent plane passes close to the focal point. However, the second situation occurs precisely when a surface element is appreciably foreshortened and will thus occupy relatively little area in the image domain.

Even when the errors are such that (4.3) does not hold approximately pointwise, it is still possible that (4.3) holds in a globally approximate sense. This will happen, for example, if the surface tilts n_1 and n_2 are random and well distributed. Because the algorithm is global, the global approximation condition suffices.

The algorithm will use feature data to fit a linear function of the form $g(x, y) = ax + by + c$ to the data, and then, ω_1, ω_2, and ω_3 may be determined directly from a, b, and c, respectively. However, using Stoke's theorem, we show that we do not have to rely on samples of the curl of the flow field; we may instead use circulation values.

Consider a region of the image domain D (not necessarily circular) such that at most of the points (x, y) in D, one of the above conditions holds true. We do not require that D be a small region, we do not require that the same condition hold true over all of D, nor are we concerned about substantial violations—as long as the average of the violations are small. We will average the values of the curl over D:

$$\frac{1}{|D|}\iint_D \mathrm{curl}(V)dxdy \approx \frac{1}{|D|}\iint_D \frac{-1}{f}(x\omega_1 + y\omega_2 + 2f\omega_3)dxdy$$
$$= \frac{-1}{f}(x_0\omega_1 + y_0\omega_2 + 2f\omega_3) \qquad (4.4)$$

where (x_0, y_0) is now the centroid in the image plane of the region D. Here, $|D|$ refers to the measure of the area of D (in the image domain). Applying Stoke's theorem to the left side of (4.4), we see that the circulation value of V about the boundary of D satisfies

$$\frac{1}{|D|}\oint_{\partial D} V \cdot d\boldsymbol{s} \approx \frac{-1}{f}(x_0\omega_1 + y_0\omega_2 + 2f\omega_3). \qquad (4.5)$$

We now state the algorithm. Suppose we have a collection of image domains D_i, $i = 1, \ldots N$. Suppose that we have normalized circulation values measured for each such a domain:

$$\gamma_i = \frac{1}{|D_i|}\oint_{\partial D_i} V \cdot d\boldsymbol{s}.$$

Let us suppose that the centroid of each region D_i is known and is denoted by (x_i, y_i). (See Fig. 3.) From (4.5), we know that the data (x_i, y_i, γ_i) for $i = 1, \ldots N$ approximately satisfies

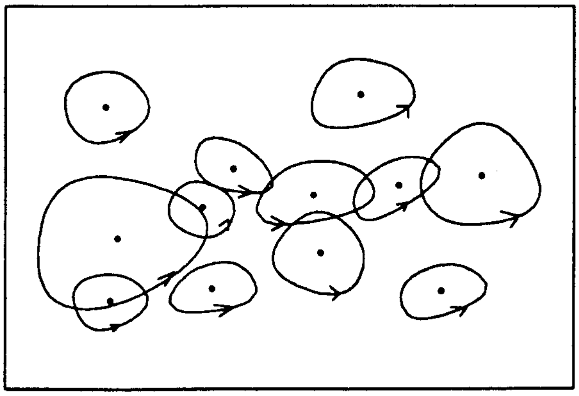


Fig. 3. Collections of cycles and their centroids, where data for the flow circulation algorithm is collected. The algorithm will simply fit a linear function to the circulation values given at each centroid.

$$\gamma_i = ax_i + by_i + c$$

where $a = -\omega_1/f$, $b = -\omega_2/f$, and $c = -2\omega_3$. We can determine the coefficients a, b, c by fitting a linear function to the available data. Outliers, of course, can be discarded by standard regression methods so that violations of the conditions are not consequential. Clearly, the rotational parameters are determined from a, b, and c.

If the collection of centroids of regions $\{(x_i, y_i)\}$ is sufficiently dense and symmetrically distributed about the image plane origin, then the determination of the rotational parameters from the circulation data becomes especially simple since then, the parameters are coefficients of mutually orthogonal functions. Specifically, a will be proportional to $\mathrm{Avg}_{i=1\ldots N} x_i \cdot \gamma_i$, b will be proportional to $\mathrm{Avg}_{i=1\ldots N} y_i \cdot \gamma_i$, and c will be equal to $\mathrm{Avg}_{i=1\ldots N}\gamma_i$. Then $\omega_1 = -fa$, $\omega_2 = -fb$, and $\omega_3 = -c/2$. Once again, outliers may be discarded from the averages to improve the quality of the estimates.

B. The FOE Search Algorithm

For each point (x_0, y_0), we consider the circular component flow field about (x_0, y_0) (see Fig. 4) defined by

$$U_{(x_0,y_0)}(x, y) = V(x, y) \cdot (-y + y_0, x - x_0). \qquad (4.6)$$

Since $V = V_{\boldsymbol{v}} + V_{\boldsymbol{\omega}}$, we further define

$$U^V_{(x_o,y_o)} = V_{\boldsymbol{v}}(x, y) \cdot (-y + y_0, x - x_0),$$
$$U^{\omega}_{(x_o,y_o)} = V_{\boldsymbol{\omega}}(x, y) \cdot (-y + y_0, x - x_0)$$

so that

$$U_{(x_0,y_0)}(x, y) = U^{\boldsymbol{v}}_{(x_0,y_0)}(x, y) + U^{\boldsymbol{\omega}}_{(x_0,y_0)}(x, y).$$

We calculate the second term first:

$$U^{\boldsymbol{\omega}}_{(x_0,y_0)}(x, y) = \omega_1\left(-\frac{x_0}{f}y^2 + \frac{y_0}{f}xy + fx - fx_0\right)$$
$$+ \omega_2\left(-\frac{y_0}{f}x^2 + \frac{x_0}{f}xy + fy - fy_0\right)$$
$$+ \omega_3\left(-y^2 - x^2 + x_0x + y_0y\right). \qquad (4.7)$$

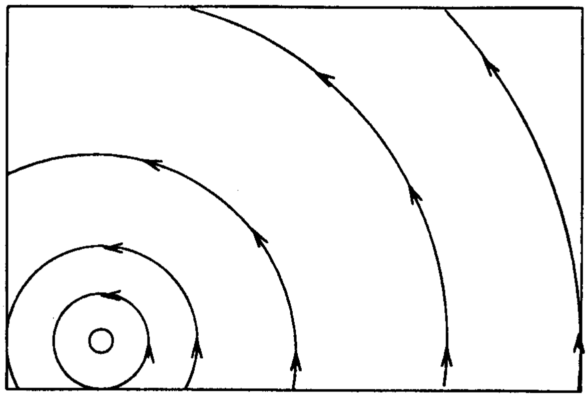


Fig. 4. Circular-component velocity function at a point (x, y) is obtained from the vector dot product of the velocity flow field and the circular vector field, as depicted, centered on (x_0, y_0), where the magnitude of the vector field increases with the radius.

The important thing to note is that for fixed $\boldsymbol{\omega}$ and (x_0, y_0), this function is a quadratic polynomial in x and y. As for the first term, we have

$$U^{\boldsymbol{v}}_{(x_0,y_0)}(x,y) = v_3\rho(x,y)\cdot((y_0-\eta)x + (-x_0+\tau)y + \eta x_0 - \tau y_0). \tag{4.8}$$

Equation (4.8) is not, in general, a quadratic polynomial since the inverse depth function $\rho(x, y)$ is not restricted. However, at the FOE, when $(x_0, y_0) = (\tau, \eta)$,

$$U^{\boldsymbol{v}}_{(x_0,y_0)}(x,y) = v_3\rho(x,y)\cdot\begin{pmatrix}x-\tau\\ y-\eta\end{pmatrix}\cdot(-y+\eta, x-\tau) \equiv 0$$

so that $U_{(x_0,y_0)} = U^{\boldsymbol{\omega}}_{(x_0,y_0)}$ for $(x_0, y_0) = (\tau, \eta)$, that is, $U_{(x_0,y_0)}$ will be a quadratic polynomial when (x_0, y_0) is at the FOE (τ, η). At other points, $U_{(x_0,y_0)}$ is composed of the sum of a quadratic polynomial (see (4.7)) and the function $U^{\boldsymbol{v}}_{(x_0,y_0)}$, which is generally not a quadratic polynomial.

The definition of the circular-component flow field function can be extended to the case where the candidate FOE is a point at infinity (corresponding to forward velocity v_3 equal to zero). In this case, the candidate FOE is a direction (v_1, v_2), and the constant vector field $(-v_2, v_1)$ may replace the circular field $(-y + y_0, x - x_0)$ in the definition of the circular component function.

The algorithm is thus the following: Every point (x_0, y_0) and potentially every direction toward infinity computes the circular component function $U_{(x_0,y_0)}$ and then determines whether the resulting function is a quadratic polynomial of the form (4.7). We suggest three methods for determining if $U_{(x_0,y_0)}$ is a quadratic polynomial of the appropriate form. The methods increase in order of discriminability of the correct FOE but lead to successively decreasing simplicity and efficiency.

1) Center-Surround Kernel Method: We first note that the Laplacian of (4.7) is a constant:

$$\nabla^2 U^{\boldsymbol{\omega}}_{(x_0,y_0)}(x,y) = \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right)U^{\boldsymbol{\omega}}_{(x_0,y_0)}(x,y) \equiv \frac{-2x_0}{f}\omega_1 - \frac{2y_0}{f}\omega_2 - 4\omega_3.$$

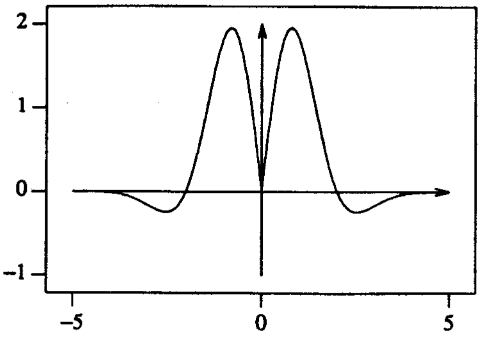


Fig. 5. Slice of the kernel K, as defined by (4.9), used in the center-surround kernel method. The 2-D kernel is a rotationally symmetric version of the displayed function. The 2-D kernel is convolved with the circular-component flow field function and yields a zero resulting function when the candidate position is located at the focus of expansion.

Thus, any derivative of the Laplacian of $U_{(x_0,y_0)}$ will give zero when $(x_0, y_0) = (\tau, \eta)$. Rather than taking derivatives, we advocate filtering $U_{(x_0,y_0)}$ by a convolution kernel. We suggest three possibilities. The first suggested kernel is $(\partial/\partial r)\Delta G_\sigma$, where G_σ is a Gaussian kernel with standard deviation σ, and r is a radial variable (we may regard G_σ as defined in polar coordinates). For fixed σ in (x, y) coordinates, this kernel is proportional to

$$K(x,y) = (x^2+y^2)^{1/2}\left(4 - \frac{x^2+y^2}{\sigma^2}\right)\cdot e^{-(x^2+y^2)/2\sigma^2}. \tag{4.9}$$

A cross section of this kernel is shown in Fig. 5.

Another possibility is $(\partial/\partial x)\Delta G_\sigma$. This kernel will not be circularly symmetric but can be implemented by differencing two horizontally displaced center-surround receptive fields, as might be found in a stereo imaging system. Convolution by this kernel is reminiscent of the receptive fields proposed by Nakayama and Loomis [41]. For this kernel, we have the (proportional) formula

$$K(x,y) = x\left(4 - \frac{r^2}{\sigma^2}\right)\cdot e^{-r^2/2\sigma^2}. \tag{4.10}$$

A third possibility is to filter $U(x_0, y_0)$ by ΔG_σ, without any derivatives. This kernel is given by

$$K(x,y) = \left(2 - \frac{r^2}{\sigma^2}\right)\cdot e^{-r^2/2\sigma^2} \tag{4.11}$$

which is the well-known Mexican-hat function. In this case, we search for a location (x_0, y_0), where the result of the filtering operation is a constant function.

By filtering, we mean that a convolution is desired:

$$\Phi_{(x_0,y_0)}(x,y) = K * U_{(x_0,y_0)}(x,y) = \iint K(x',y')U_{(x_0,y_0)}(x-x', y-y')dx'dy'.$$

We are assured that when $(x_0, y_0) = (\tau, \eta)$, $\Phi_{(x_0,y_0)}(x, y)$ is identically zero (the zero function) for the kernels (4.9) and (4.10) discussed above and is constant for the kernel (4.11).

For kernels (4.9) and (4.10), we are assured that when $(x_0, y_0) = (\tau, \eta)$, $\Phi_{(x_0,y_0)}(x, y)$ is identically zero (the zero function) no matter what value of σ has been chosen. For

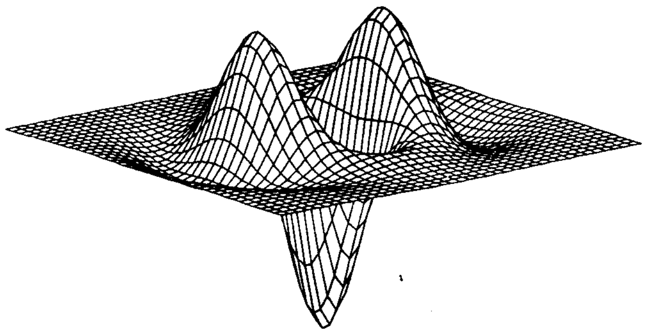


Fig. 6. Autoconvolution kernel $K * K$ that may be used as the quadratic functional kernel in the center-surround kernel method to detect whether $U_{(x_0,y_0)}$ is a quadratic polynomial and, thus, if (x_0, y_0) is the FOE.

kernel (4.11), $\Phi_{(x_0,y_0)}$ will be constant at $(x_0, y_0) = (\tau, \eta)$ and thus has zero variance at this location.

Therefore, the center-surround kernel method asserts that a candidate FOE is found at a location (x_0, y_0) where the function $\Phi_{(x_0,y_0)}$ is identically zero for the first two candidate kernels, or has zero variance, in the case of the third candidate kernel. A reasonable criterion would be to search for (x_0, y_0) such that

$$E(x_0, y_0) = \iint |\Phi_{(x_0,y_0)}(x, y)|^2 dx dy$$

is zero in the case of kernels (4.9) and (4.10) and where

$$E(x_0, y_0) = \mathrm{Var}(\Phi_{(x_0,y_0)})$$

is zero in the case of kernel (4.11).

In all cases, the error surface $E(x_0, y_0)$ can be rewritten as a quadratic functional of the function $U_{(x_0,y_0)}(x, y)$. The quadratic functional has form

$$E(x_0, y_0) = \iiiint R(x, y, x', y') \cdot U_{(x_0,y_0)}(x, y) \cdot U_{(x_0,y_0)}(x', y') dx dy dx' dy'. \tag{4.12}$$

Formulas may be derived for R in each of the three cases. For example, for the kernel $(\partial/\partial x)\Delta G_\sigma$ (kernel (4.10), it can be shown that

$$R(x, y, x', y') = S(x - x', y - y')$$

where

$$S(x, y) = \frac{1}{64\sigma^{12}}\big(-192\sigma^6 + 192\sigma^4 x^2 - 32r^2\sigma^2 x^2 + (x^2 - 2\sigma^2)r^4 + 48\sigma^4 r^2\big) \cdot e^{-r^2/4\sigma^2}$$

for $r = \sqrt{x^2 + y^2}$. In fact, in this case, S is simply $K * K$, which is the autoconvolution of our kernel K. In turn, $K * K$ equals $(\partial^2/\partial r^2)\Delta^2 G_{\sqrt{2}\sigma}$. More manipulations show that (4.12) may be rewritten in this case as

$$E(x_0, y_0) = \iint U_{(x_0,y_0)}(x, y) \cdot \big(K * K * U_{(x_0,y_0)}\big)(x, y) dx dy. \tag{4.13}$$

A plot of the $K * K$ kernel is shown in Fig. 6.

Of utmost importance is the fact that $E(x_0, y_0)$ is a quadratic polynomial in (x_0, y_0) for all three kernels for any σ. In fact, the result is true for any K for both the norm and variance definitions of E. A proof of this may be found in [42]. As a consequence, it suffices to compute $E(x_0, y_0)$ for six different potential FOE points, whence the entire structure of E will be known, and the true FOE can then be determined immediately without a search.

Although this provides a particularly attractive test for the FOE, the problem with the center-surround kernel is that many functions other than quadratic polynomials will pass the test. For example, if ρ is such that $U_{(x_0,y_0)}$ happens to be a harmonic function for some (x_0, y_0) other than (τ, η), then E will be identically zero. Although it is not likely that ρ will yield a harmonic $U_{(x_0,y_0)}$ for general surface shapes, the test using the center-surround kernel is not specific to the form of quadratic polynomial that arises at the FOE and, thus, can be considerably improved.

2) The Quadratic Polynomial Projection Method: The following are the Hermite polynomials in two variables up to degree two:

$$\phi_1(x, y) = \frac{1}{\sqrt{2\pi}}, \quad \phi_2(x, y) = \frac{x}{\sqrt{2\pi}}, \quad \phi_3(x, y) = \frac{y}{\sqrt{2\pi}},$$

$$\phi_4(x, y) = \frac{xy}{\sqrt{2\pi}}, \quad \phi_5(x, y) = \frac{x^2 - 1}{2\sqrt{\pi}}, \quad \phi_6(x, y) = \frac{y^2 - 1}{2\sqrt{\pi}}.$$

These polynomials have unit norm and are mutually orthogonal with respect to the Hermite inner product

$$\langle f, g\rangle = \iint f(x, y) g(x, y) e^{(-x^2-y^2)/2} dx dy. \tag{4.14}$$

The function $U_{(x_0,y_0)}$ will thus be a quadratic polynomial if and only if the following function vanishes identically:

$$U_{(x_0,y_0)}(x, y) - \sum_{i=1}^{6}\langle U_{(x_0,y_0)}, \phi_i\rangle \phi_i(x, y).$$

For the quadratic polynomial projection method, we will use the norm of this function as our error measure $E(x_0, y_0)$. The (Hermite) norm, which will be zero if and only if $U_{(x_0,y_0)}$ is a quadratic polynomial, may be written as a quadratic functional of $U_{(x_0,y_0)}$ in the form

$$E(x_0, y_0) = \iiiint R(x, y, x', y') \cdot U_{(x_0,y_0)}(x, y) U_{(x_0,y_0)}(x', y') \cdot e^{-(x^2+y^2+x'^2+y'^2)/2} dx dy dx' dy'. \tag{4.15}$$

In this case, R is the "quadratic polynomial projection kernel" and will equal

$$R(x, y, x', y') = \delta(x, y, x', y') - \sum_{i=1}^{6} \phi_i(x, y)\phi_i(x', y').$$

Here, $\delta(x, y, x', y')$ is a delta distribution whose mass lies entirely on the slice $(x, y) = (x', y')$. We thus see that the test for (x_0, y_0) depends on a quadratic functional applied to

$U_{(x_0,y_0)}$ and that the kernel of the quadratic functional in (4.15) has the form

$$\begin{aligned} &R(x,y,x',y') = \\ &\quad \delta(x,y,x',y') - \frac{1}{2\pi}(1+xx'+yy'+xyx'y') \\ &\quad - \frac{1}{4\pi}\Big((x^2-1)\Big(x'^2-1\Big) + (y^2-1)\Big(y'^2-1\Big)\Big). \end{aligned}$$

Note that the quadratic polynomial projection kernel is independent of (x_0, y_0).

Once again, it can be shown that $E(x_0, y_0)$ will itself be a quadratic polynomial in (x_0, y_0). Thus, as with the center-surround kernel method, a search over (x_0, y_0) is not necessary.

The quadratic polynomial projection method has the property that if $U_{(x_0,y_0)}$ is a quadratic polynomial, then (x_0, y_0) will be identified as the focus of expansion. Alas, if $U^{\boldsymbol{v}}_{(x_0,y_0)}$ happens to be a quadratic polynomial, then $U_{(x_0,y_0)}$ is simply the sum of two quadratic polynomials, and therefore, (x_0, y_0) will be erroneously identified. We might assert that it is highly unlikely that the scene will be such that $U^{\boldsymbol{v}}_{(x_0,y_0)}$ gives precisely a quadratic polynomial, but unfortunately, if $\rho(x, y)$ happens to be a linear function, then $U^{\boldsymbol{v}}_{(x_0,y_0)}$ is indeed a quadratic polynomial. It so happens that $\rho(x, y)$ is linear (and, thus, harmonic as well; therefore, the center-surround kernel method will also fail) when the surface $Z(x, y)$ is planar (although the entire scene will have to consist of the single planar surface).

3) The Subspace Projection Method: We note that $U^{\boldsymbol{\omega}}_{(x_0,y_0)}$ is a quadratic polynomial in (4.6) of a special form; $U^{\boldsymbol{\omega}}_{(x_0,y_0)}$ must lie in a 3-D subspace of quadratic polynomials spanned by the (redefined) basis functions

$$\begin{aligned} \phi_1(x,y) &= -\frac{x_0}{f}y^2 + \frac{y_0}{f}xy + fx - fx_0, \\ \phi_2(x,y) &= -\frac{y_0}{f}x^2 + \frac{x_0}{f}xy + fy - fy_0, \\ \phi_3(x,y) &= -y^2 - x^2 + x_0x + y_0y. \end{aligned} \tag{4.16}$$

Note that unlike the Hermite basis functions from the previous section, these basis functions depend on (x_0, y_0).

For the subspace projection method, we define the error $E(x_0, y_0)$ to be

$$E(x_0,y_0) = \min_{a_1,a_2,a_3} ||U_{(x_0,y_0)} - \sum_{i=1}^{3} a_i\phi_i||^2.$$

The norm is based on an inner product, and we use the Hermite inner product defined in (4.14). This is a standard least-squares minimization problem, and the a_i may be found as the solution to the "normal equations"

$$Q_{(x_0,y_0)} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} \langle U_{(x_0,y_0)}, \phi_1 \rangle \\ \langle U_{(x_0,y_0)}, \phi_2 \rangle \\ \langle U_{(x_0,y_0)}, \phi_3 \rangle \end{pmatrix} \tag{4.17}$$

where $Q_{(x_0,y_0)}$ is the three-by-three matrix of inner products, i.e., the i, j^{th} component contains the value $\langle \phi_i, \phi_j \rangle$, which depends on (x_0, y_0). It is not hard to show that for a solution set a_1, a_2, and a_3, the minimum distance $E(x_0, y_0)$ is given by the quadratic functional formula (4.15), where R, which is the "subspace projection kernel," is now redefined as

$$\begin{aligned} &R_{(x_0,y_0)}(x,y,x',y') = \\ &\quad \delta(x,y,x',y') \\ &\quad - (\phi_1(x,y), \phi_2(x,y), \phi_3(x,y)) Q^{-1}_{(x_0,y_0)} \begin{pmatrix} \phi_1(x',y') \\ \phi_2(x',y') \\ \phi_3(x',y') \end{pmatrix}. \end{aligned}$$

The triple product (the second term on the right-hand side) means the same thing as

$$\sum_{i=1}^{3}\sum_{j=1}^{3} q_{i,j}\phi_i(x,y)\phi_j(x',y')$$

where $q_{i,j}$ is the i, jth component of the inverse matrix $Q^{-1}_{(x_0,y_0)}$. Note that Q and, thus, Q^{-1} can be precomputed; they are matrices of constants that depend on (x_0, y_0). In fact, closed-form formulas for the coefficients can be derived as functions of (x_0, y_0); details appear in [43]. Using the resulting kernel R provides the exact test, and $E(x_0, y_0)$ will be zero if and only if $U_{(x_0,y_0)}$ is a quadratic polynomial of exactly the correct form.

However, the disadvantage of the subspace projection method is that the error surface $E(x_0, y_0)$ is not quadratic in this case, and thus, a search over many possible FOE points may be necessary.

V. Commentary

We may now summarize the two algorithms and comment on their applicability. An analysis of their viability, based on numerical analysis and empirical evidence, is given in subsequent sections. Although the algorithms are not directly motivated by biological concerns, there are implications for possible neuronal implementations. These implications are not stressed here.

The *flow circulation algorithm* is easily understood. Circulation values are obtained for a number of cycles in the image domain. A circulation value is simply a contour integral of the vector flow field around the cycle and measures the "swirling" of the field about the cycle. Mathematically, a circulation value is proportional to the average of the curl of the vector field in the region enclosed by the cycle. Our analysis has shown that these values, taken as data points at the centroids of the regions, will approximately lie on a linear surface: $g(x, y) = ax + by + c$. By fitting such a surface to the given data, the parameters a, b, and c are determined. These are proportional to the rotational parameters of motion ω_1, ω_2, and ω_3, respectively. Clearly, fitting a linear surface to the data is a global process, which we believe will lead to more stable determination of the rotational parameters than local methods based on local deformation parameters (i.e., higher order derivatives of the vector flow field). Further, fitting a linear surface is a particularly simple global process; the coefficients can be determined by a regression analysis that will essentially use weighted sums of the data points.

The fact that the circulation values only *approximately* lie on a linear surface makes the flow circulation algorithm an approximate method. In the next section, we consider the magnitude of errors that can result pointwise. However, the real issue is the extent of the errors in an average sense, and there are ways that the errors can be managed to improve the accuracy of the globally obtained rotational parameters. We mention three techniques here: 1) discarding of outliers, 2) depth filtering, and 3) surface normal balancing. For discarding outliers, an iterative approach may be used, which first fits a linear surface and then improves on the parameters defining the surface by discarding data that lies far from the norm. For depth filtering, we might intentionally discard cycles that enclose points that lie on nearby surfaces. Such surfaces are frequently located near the periphery of the visual field; other nearby surfaces (that do not lie normal to the line of sight) might be identified and discarded from the flow circulation algorithm by independent depth sensing mechanisms. For surface normal balancing, we note from (4.2) that perturbations to the estimates of ω_1, ω_2, and ω_3 are mediated by the values of the surface component normals n_2, n_1, and $n_1 v_2 - n_2 v_1$, respectively. Thus, if the circulation values obtained for the flow circulation algorithm encompass elements such that the tilts (n_1, n_2) are balanced, then errors will cancel. Thus, when surfaces with large tilts are present, we should chose circulation values such that the union of the enclosed regions contains surfaces that have an even balance of tilts with respect to the horizontal axis, the vertical axis, and the axis defined by the direction to the FOE. Again, independent surface structure methods such as stereo and shading clues may be used to make the selection.

The *FOE search algorithm* is an exact algorithm in the sense that barring noise, the FOE will be found. This algorithm is also quite easily stated. For every candidate point, the circular-component flow velocity function is computed according to (4.6). The resulting function is used in a quadratic functional computation, which is used to determine if the function is a quadratic polynomial of the appropriate form. The functional is zero at the FOE.

Three methods were suggested: the center-surround kernel method, the quadratic polynomial projection method, and the subspace projection method. The methods determine, respectively, whether the circular-component flow function has constant Laplacian, is a quadratic polynomial, or is a quadratic polynomial in the proper 3-D subspace. Since the error surface for the first two methods is itself a quadratic function, that surface is either identically zero, or there will be at most one zero of the error surface, which in the noise-free case must lie at the FOE. Of the three methods suggested, the center-surround kernel method is the simplest for determining whether the circular-component flow function is a quadratic polynomial.

There are alternative ways of implementing the methods. The center-surround kernel methods can be implemented by filtering the circular-component function with an appropriate kernel and then determining if the resulting *function* is the zero function (or constant, in the case of the kernel (4.10)). Alternatively, a quadratic functional computation may be made, in the form (4.12), to determine if the final *value* is zero. The computation can often be simplified by making use of the structure of the quadratic functional kernel R, implementing part of the computation as a convolution, as demonstrated by the example of (4.13). For the projection methods, a function may be determined by projecting $U_{(x_0,y_0)}$ onto a subspace, or a value is obtained from a quadratic functional applied to $U_{(x_0,y_0)}$. For a discrete version of the quadratic polynomial projection method, for example, we could fit a quadratic polynomial to the discrete collection of data obtained as point evaluates of the circular-component flow function at the locations where flow field values are available and then determine the quality of the fit. If a sum-of-squares norm is used, the error surface will once again be quadratic. Accordingly, the best-fit quadratic polynomial yields a simpler version of the Heeger/Jepson algorithm. For the subspace projection method, the kernel of the functional depends on (x_0, y_0) and, thus, is the most complicated method.

If the viewed scene consists of a single flat plane, with constant orientation over the entire visual field, then the circular component function $U_{(x_0,y_0)}$ will be a quadratic polynomial for any (x_0, y_0), and therefore, the center-surround kernel method and the polynomial projection method will fail. For this case, the full complexity of the subspace projection method is required.

The fact that the algorithms use global information to obtain the motion parameters is their great advantage. As was noted, the history of motion parameter determination is largely based on local measurements of the vector flow field (and local measurements of the derivatives of that flow field) to determine both the motion parameters and the local surface structure (the surface normal and/or surface curvatures). Although there is plenty of psychophysical evidence showing that humans are capable of deducing local surface structure from (potentially impoverished) local flow field data, there is no certainty as to the mechanisms underlying the determination. Our contention is that stable and accurate determination of translation and rotation motion parameters (due to egomotion) is possible using global information and that local surface structure determination might conceivably make advantageous use of the globally determined motion parameters.

As a consequence of the global use of information, both algorithms make use of regression methods. The flow circulation algorithm fits a linear function to observed data, whereas the FOE search algorithm essentially fits quadratic polynomials to a collection of functions. The two algorithms are, in a sense, complementary. The flow circulation algorithm works best if visible surfaces are frontal planar; the FOE search algorithm works best when there is a lot of variation in surface orientations. The flow circulation algorithm uses as input the sum of flow components about cycles, whereas the FOE search algorithm requires circular-component flow velocity values evaluated at points (which are then filtered). The prefiltered circular-component values may be used to compose the circulation values, but it is entirely possible that independent feature detection mechanisms can be devised for the two algorithms.

On the other hand, each algorithm may be used to verify the other. If the flow circulation algorithm gives an estimate of the rotational parameters ω, then the rotational vector

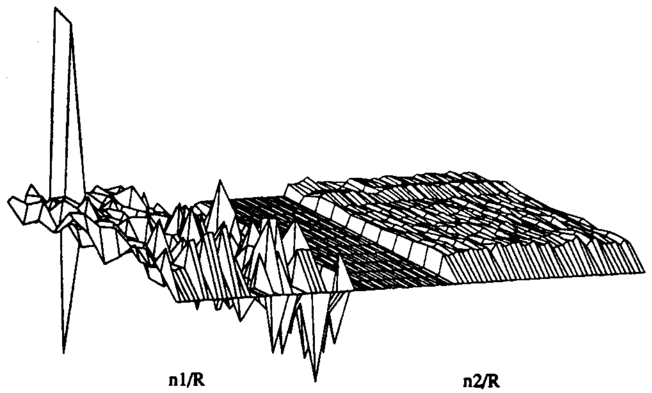


Fig. 7. Plot of n_1/R is seen in the left, and a plot of n_2/R of the same image domain is in the right for a typical scene imaged by a depth sensor consisting of a road. The depth data is used to compute the surface normals. The values of n_2/R are mostly positive and average to 0.0103, whereas the values of n_1/R are better distributed around zero, averaging to 0.0040.

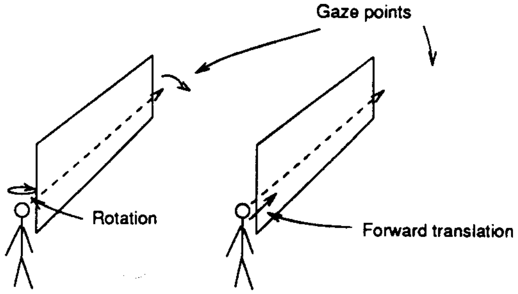


Fig. 8. Geometric and motion parameters for the flow fields shown in Fig. 9. In the case on the left, the person's translational velocity is zero, and the rotational velocity is (0,-0.125,0), indicating an instantaneous rotation about the vertical axis of 0.125 rad/s. In the case on the right, the person's rotational velocity is zero, and the forward translational velocity is 5 focal units/s. The wall lies 40 focal units to the left and extends from 20 to 80 focal distances in range.

field V_ω is determined and, therefore, can be subtracted from the observed vector field V. The result should be a pure expansion or contraction vector field centered around the FOE, as reported by the FOE search algorithm. On the other hand, given the FOE, the rotational parameters can then be found independently of the flow circulation algorithm from the coefficients of the best linear fit using the basis functions (4.16) to the circular-component flow function about the FOE. This should give the same rotational parameters as the flow circulation algorithm.

VI. An Analysis of the Sensitivity

In this section, we calculate the range of errors that might be expected for the proposed algorithms by numerically investigating typical values of the parameters.

A. Rotational Parameter Estimation

For the flow circulation algorithm, the important issue is the validity of the linearity assumption for the flow circulation values. We first consider the situation with pointwise evaluations of the curl. Recalling (4.2)

$$\begin{aligned}\operatorname{curl}(V)(x,y) = -x\cdot\left(\omega_1+\frac{n_2v_3}{R}\right)-y\cdot\left(\omega_2-\frac{n_1v_3}{R}\right)\\ -\left(2\omega_3+\frac{n_1v_2-n_2v_1}{R}\right).\end{aligned}$$

Here, we have assumed that units of measurement are in terms of the focal length of the system so that $f=1$.

Clearly, our ability to estimate the parameters ω_1, ω_2, and ω_3 will depend on the size of the remaining terms in the corresponding coefficients, that is, we need the terms

$$\frac{n_2\cdot v_3}{R},\quad \frac{n_1\cdot v_3}{R},\quad \frac{(n_1v_2-n_2v_1)}{2R}$$

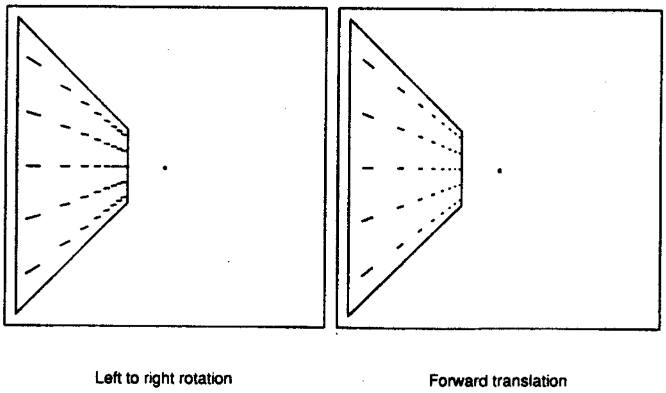
to be small in magnitude, respectively, relative to the expected sizes of ω_1, ω_2, and ω_3. In the worst case, the magnitudes of any of these three terms can attain the respective values of

$$\frac{|v_3|}{R},\quad \frac{|v_3|}{R},\quad \frac{\sqrt{v_1^2+v_2^2}}{2R}.$$

If this is all the information that is given, then we will need large values of R. Specifically, suppose that a reasonable value for a rotational velocity of a rotational parameter is on the order of 0.1 or 0.2 rad/s. Then, as long as R is greater than $|v_3|$ by a factor of 50 or so, then the rotational parameters ω_1 and ω_2 should be deducible to within an accuracy of 0.02 rad/s. Likewise, R should be larger than the magnitude of the lateral velocity in the (X,Y) plane by a factor of 25 or so. Recalling that R is the distance from the nearest approach of the tangent plane to the focal point, we observe that tangent planes to surfaces visible in the scene should, for the most part, stay outside of a sphere whose radius is the distance to be traversed at the current velocity in the next 50 s. This is a quite large bound and hardly ever true in practice unless the translational velocity is zero. However, if the forward velocity component is zero, then accurate estimation of ω_1 and ω_2 is assured by this method, whereas if the lateral translational velocity is zero, then ω_3 will be precise.

More realistically, it is the *average* values of n_1 and n_2 throughout the image that influences the accuracy of the linear regression that is used to estimate the components of $\boldsymbol{\omega}$. For example, if n_1 and n_2 average to 0.1, then the previous bounds may be reduced by a factor of 10. Although typical values of the average surface normal tilts must be determined empirically, many scenes are composed of a variety of tilt directions. Fig. 7 shows plots of n_1/R and n_2/R for a typical scene of a road computed using depth data. The values of n_1/R are well distributed about zero and average to 0.0040. The values of n_2/R, on the other hand, are predominantly positive and average to 0.0103. The result is that using this scene, the value of the horizontal rotational component ω_1 is likely to be estimated with greater error when using the flow circulation algorithm.

We observe with the above example that the rotation component about the horizontal axis ω_1 is confounded by forward velocity and horizontal surfaces with large components of n_2/R. Likewise, rotation about the vertical axis ω_2, which would arise as a sensor rotates from left to right, is confounded by a forward velocity and a surface patch whose tilt lies horizontally. Consider, for instance, a person walking next to a vertical wall (see Fig. 8). Instantaneously, the motion field



Left to right rotation

Forward translation

Fig. 9. Image flow fields induced by the motions described in Fig. 8 on a vertical wall. Qualitatively, the flow field due to rotation is similar to the flow field due to forward translation locally. The curl of the two flow fields are identical since $-\omega_2$ in the rotational case equals $n_1 v_3/R$ in the translational case. However, the magnitude of the vectors decrease with range in the case of the translational motion.

induced by the wall can indicate either forward velocity or rotation of the head away from the wall. As shown in Fig. 9, the two instantaneous flow fields are similar but not equal. The curl of these flow fields along the wall are equal (equal curls do not imply equal vector fields). In any given vertical slice along the wall, the flow fields are qualitatively extremely similar. The sense in which they differ is that the magnitude of the vector field due to forward translation decreases with the range; therefore, the velocity vectors are quite small at the far end of the wall. Since the curls are identical, the flow circulation algorithm will not distinguish between the two cases. The ambiguity may easily be resolved by higher level processing, such as analyzing over time the scene at the gaze point. A similar analysis shows that rotation about the line of sight is readily confused with lateral motion with respect to a surface such that $-n_2 v_1 + n_1 v_2$ is large.

Our conclusion is that for some scenes, the flow circulation algorithm should be able to estimate the rotational parameters correctly; for other scenes, accurate estimation of rotational parameters from the velocity field using the flow circulation algorithm is an unstable process and that other processing, such as scene analysis of the gaze point or global analysis of some feature of the flow field in addition to the curl, will be necessary.

B. FOE Estimation

For the FOE search algorithm, we already know, since the algorithm is exact, that $E(\tau, \eta)$ will always be zero, assuming there is no noise. Since $U^{\boldsymbol{\omega}}_{(x_0,y_0)}$ (see (4.7)) projects to zero (under any of the methods), errors arise solely due to noise in the flow field. The noise is balanced against the error surface E that results from projection of $U^{\boldsymbol{v}}_{(x_0,y_0)}$ (see (4.8)). If the scene consists solely of a plane that is tilted in any direction, then $\rho(x, y)$ will be linear, and both the center-surround kernel method and the quadratic polynomial projection methods will yield a function $E(x, y)$ that is zero everywhere. Thus, the algorithm fails in this case. However, if there exist two or more planar surfaces (as in the case of a corridor scene; see Section VII), then E will generally be nonzero except at the FOE.

Except for these degenerate cases, the center-surround kernel method works well in our experiments. If the convolution kernel used is the radial derivative of the Laplacian of a Gaussian (see (4.9), then $E(x_0, y_0)$ is the norm of a smoothed version of a function of the following form:

$$\frac{1}{\sqrt{x^2+y^2}}\left(x\frac{\partial}{\partial x}+y\frac{\partial}{\partial y}\right)[L(x,y)\cdot\Delta\rho + c\cdot D_{\boldsymbol{w}}\rho]$$

where $L(x, y)$ is a linear function (depending on (x_0, y_0)), and $D_{\boldsymbol{w}}$ is a directional derivative in a fixed direction (which depends on (x_0, y_0)). Clearly, discontinuities and sudden jumps in ρ will lead to large Laplacian and gradient values, which will contribute to the residual and, thus, to $E(x_0, y_0)$. We thus see that the center-surround kernel method benefits from scenes with depth variations. The amount of error allowed in the flow field determination for typical scenes, however, must be determined by further empirical analysis.

The quadratic polynomial and subspace projection methods have the advantage that they can be used with sparse data, if necessary, by fitting the appropriate quadratic polynomials to the distributed data.

VII. Experimental Results

In order to assess the performance empirically, we apply the algorithms to synthetically generated vector flow fields of both synthetic scenes and scenes of actual depth data.

A. Flow Circulation Algorithm

Consider the image of an ellipsoid that is defined by

$$\left(\frac{X-X_0}{a}\right)^2+\left(\frac{Y-Y_0}{b}\right)^2+\left(\frac{Z-Z_0}{c}\right)^2=1.$$

We assume that the ellipsoid center is 150 focal units away (along the Z axis) and that the semiradii are 50, 30, and 70 focal units, respectively. We assume the ellipsoid is in front of a flat planar background. With the observer moving with a velocity of (0.3, 0,2) focal units/s and rotating with an angular velocity of (0.2, 0.1, 0.5) rad/s, the curl of the vector flow field will have values as graphed in Fig. 10. The true linear surface is seen in the background region, and the distortion in the other regions is caused by the translational velocity in conjunction with the surface tilts. If there is error in the determination of the curl of the vector field, then the resulting surface shown in Fig. 10 will be similarly perturbed. In addition, a larger translational velocity will cause a larger perturbation of the surface. Recall that estimates of the linear surface will most likely be based on averaged values over regions and not on local gradients of the displayed surface. Clearly, the best-fit linear surface to the surface shown in Fig. 10 will give an accurate estimate of the parameters of the unperturbed linear surface.

We next apply the algorithm to a synthetically generated vector flow field using a scene obtained from actual depth data. The depth data was obtained from publically available scanned data obtained from a "White Scanner" by researchers

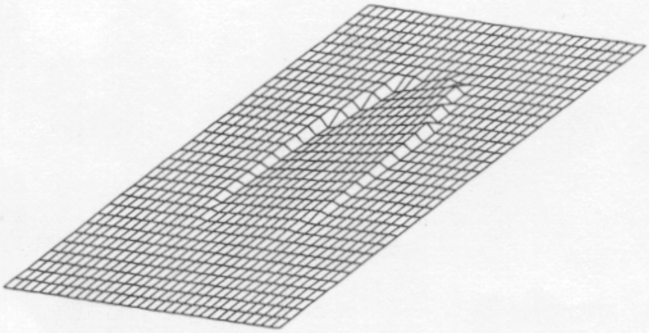


Fig. 10. Curl of the flow field due to an imaged ellipsoid, with a sensor velocity of $\boldsymbol{v} = (0.3, 0, 2)$ and rotational velocity $\boldsymbol{\omega} = (0.2, 0.1, 0.5)$.

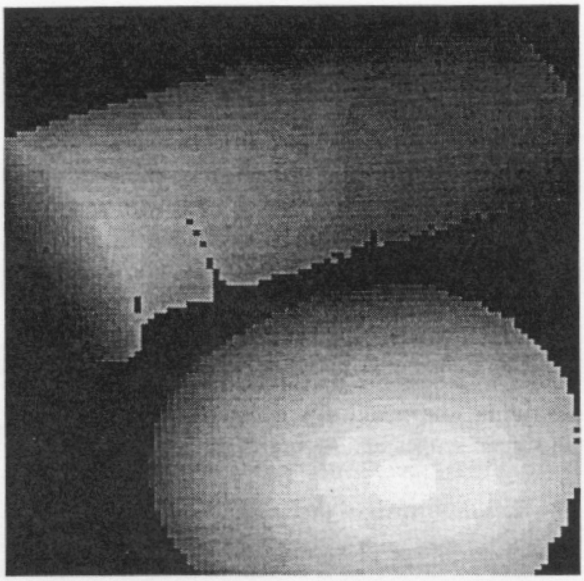


Fig. 11. Depth image, where lighter values represent points with smaller depth. The scene consists of a cylinder and a sphere sitting in front of a flat plane. Only depth values are depicted—the sampling rate with respect to x and y vary somewhat throughout the image.

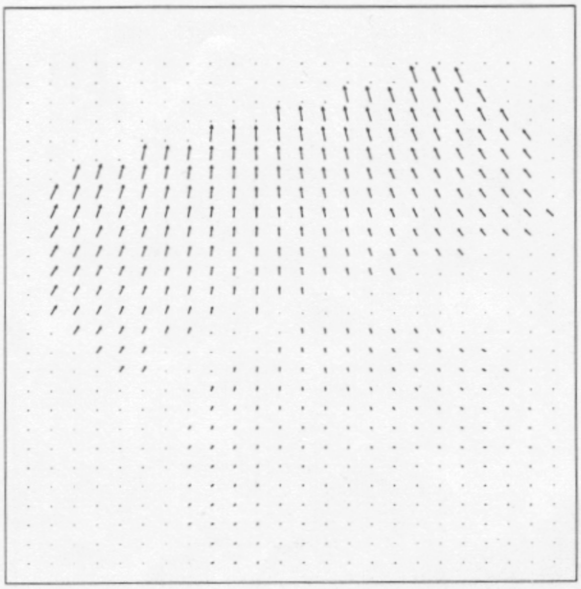


Fig. 12. Flow field computed using the depth values given by the depth sceneof Fig. 11 computed analytically using a camera motion of$\boldsymbol{v} = (5, 2, 20)$ focal units/s, and$\boldsymbol{\omega} = (0.2, 0.1, 0.5)$ rad/s.

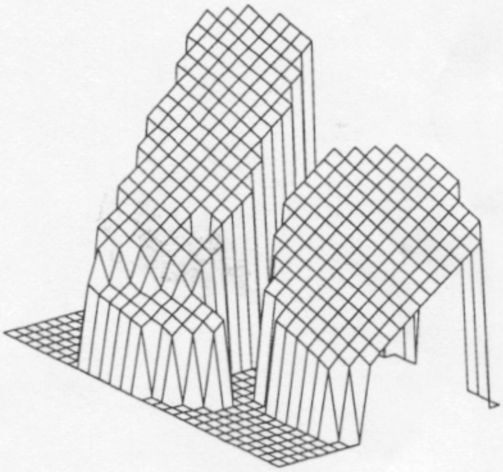


Fig. 13. Approximate curl values of the vector flow field that result when the scene whose depth values are displayed in Fig. 11 is imaged by a sensor whose translational velocity is $\boldsymbol{v} = (5, 2, 20)$and rotational velocity is $\boldsymbol{\omega} = (0.2, 0.1, 0.5)$. Data is only shown in regions where depth values are defined in Fig. 11. Partial derivatives of the flow components are computed using a local regression, and the location of the curl values are moved to approximately the correct image location. The resulting best-fit planar surface predicts rotational parameters of (0.2126, 0.1023, 0.4982).

at Michigan State University. Fig. 11 shows a gray-scale interpretation of the raw range data of the scene.

The scene contains a sphere and a cylinder; the sphere, for example, has radius of about four focal units, and its center is located 50 focal units from the sensor. The scene, it should be noted, is a ''foveal region,'' occupying an image size of 0.15 by 0.25 focal units. A simulated flow field is then computed using the depth values, as well as the projected image coordinate values, in conjunction with a camera motion of $\boldsymbol{v} = (5, 2, 20)$ focal units/s and $\boldsymbol{\omega} = (0.2, 0.1, 0.5)$ rad/s (relatively large motion parameters). An indication of the flow field is shown in Fig. 12.

An approximation to the curl of the flow field is computed using the u, v data at five points in the neighborhood of each discrete sample, and the result is displayed in Fig. 13. Data is plotted only at (x, y) positions, where meaningful depth data is derived from Fig. 11. Once again, it can be seen that the curl is approximately linear, as expected. Finally, the best-fit plane (without using any benefit of discarding outliers) is computed to the approximated curl data over the regions where curl data has been computed and is used to estimate the rotational parameters. The procedure yields the estimate $\boldsymbol{\omega} \approx (0.2126, 0.1023, 0.4982)$, which quite accurately reflects the true rotational velocity.

None of these experiments have made use of circulation values. Instead, we have computed the full field of curl values analytically or numerically over the image. If circulation values were used instead, then the curl values would be averaged over the corresponding regions. Preliminary experiments using sequences of images with known motion parameters suggest that errors can be substantial using existing flow field extraction methods. Ideally, circulation values would be provided directly by the sequence analysis process, as opposed

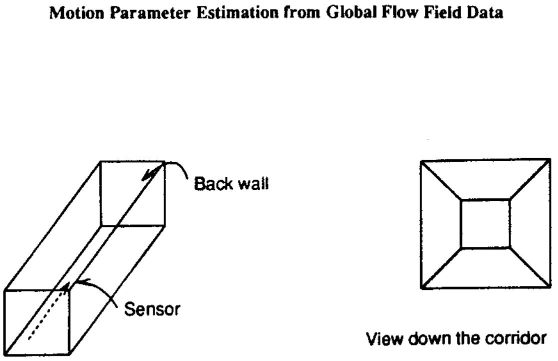


Fig. 14. Corridor and view from the sensor. The camera is located in the center of the corridor, viewed from down the corridor. The back end of the corridor is located 500 focal units away, and the width of the corridor is 50 focal units. The camera has translational velocity $\boldsymbol{v}$ angular velocity ω, and a flow field is thus induced on the image domain.

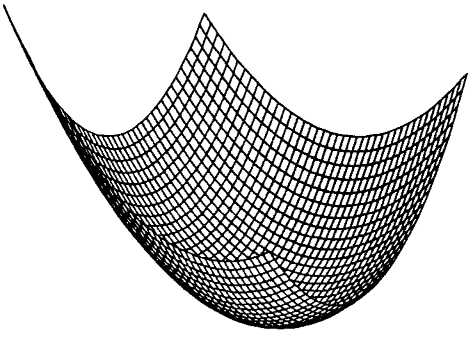


Fig. 15. Plot of the values $E(x_0, y_0)$ obtained from the norm of the function $\Phi_{(x_0,y_0)}$, which in turn is obtained from a discrete convolution of the circular component function $U_{(x_0,y_0)}$ using the horizontal difference of the discrete Laplacian as the kernel. The minimum value occurs, as desired, at the FOE.

to providing full-field vector flow fields. Once a continuous flow field is available, there is no advantage in computing circulation values. In addition, the extraction will ideally make use of large-field data, including peripheral fields as present in normal human visual fields.

B. FOE Search algorithm

Some of the experiments use synthetic flow fields created when the camera moves along a corridor with translational and rotational velocity (see Fig. 14).

Recall that the FOE search method involves computing for each candidate FOE (x_0, y_0) the circular component function $U_{(x_0,y_0)}(x, y)$ of the flow field, as given in (4.6). We test all three methods to determine whether the resulting function $U_{(x_0,y_0)}$ is a quadratic polynomial. All computations are done over a finite discrete grid.

We also test the first method (the center-surround kernel method) with the vector flow field induced by the depth scene shown in Fig. 11.

1) The Center-Surround Kernel Method:

We use a first difference (along the horizontal direction) of a discrete Laplacian as the kernel K (see Section IV-B). It can be shown that if this kernel is applied to point evaluates on a discrete grid of the circular-component function $U_{(x_0,y_0)}$ with (x_0, y_0) at the FOE, then, despite the fact that the discrete Laplacian is an approximation of the continuous Laplacian, the result should be a zero function. The actual kernel that was used is 3 by 4, with entries

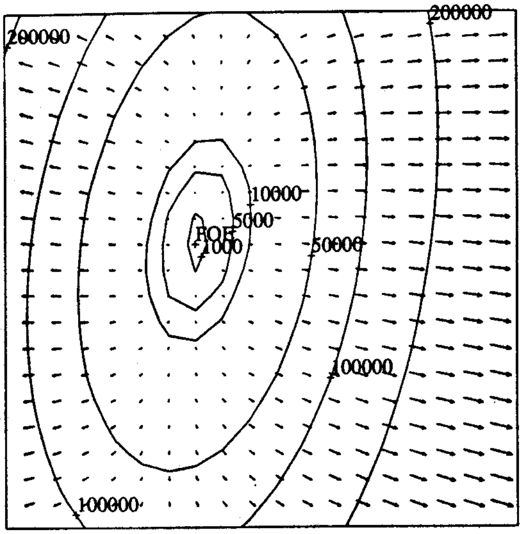


Fig. 16. Contour and gradient of the error function $E(x_0, y_0)$ using the center-surround kernel method, as described in Fig. 15, computed for the flow field induced when the depth image of Fig. 11 is imaged by a sensor with translational velocity of (5, 2, 20) and rotational velocity of (0.2, 0.1, 0.5). The function has the minimum at the FOE at (0.25, 0.1).

$$K = \begin{bmatrix} 0 & 1 & -1 & 0 \\ 1 & -5 & 5 & -1 \\ 0 & 1 & -1 & 0 \end{bmatrix}.$$

For every candidate (x_0, y_0), we compute $U_{(x_0,y_0)}(i, j)$ and then convolve it with the filter K, yielding $\Phi_{(x_0,y_0)}(i, j)$. We know that for the correct FOE, this function $\Phi_{(x_0,y_0)}$ is identically zero. To test this, we compute

$$E(x_0, y_0) = \sum_i \sum_j \left(\Phi_{(x_0,y_0)}(i, j)\right)^2.$$

A plot of the function $E(x_0, y_0)$ is shown in Fig. 15 for the case where the camera is moving along the corridor with a translational velocity $\boldsymbol{v} = (-4, -2, 16)$ and with rotational parameters $\omega = (0.1, 0.2, 0.035)$. The minimum of the function, which is precisely zero, occurs at the true FOE $(-0.25, -0.125)$. As can be seen, the error function is nonzero elsewhere and has a quadratic behavior.

Fig. 16 shows the error function that results when the translational and rotational motion of (5, 2, 20) and (0.2, 0.1, 0.5) are used to compute a vector flow field of the depth scene of Fig. 11, and the same center-surround kernel method is applied to obtain the $\Phi_{(x_0,y_0)}$ functions. We present the error function as a contour plot together with a field denoting the gradient of E in order to suggest the possibility of a gradient descent search for the FOE. In this case, we assume that there is a fixed-depth background behind the objects in Fig. 11 so that circular-component functions can be defined over the entire image. Once again, the error function correctly zeros out at the true FOE at (0.25, 0.1).

2) The Quadratic Polynomial Projection Method: The center-surround kernel method will fail to work in certain pathological situations; however, since the method works adequately for the examples given in the previous section, we would expect the other two methods to work as well. The main issue is whether the accuracy of the localization of the FOE suffers due to the enhanced specificity of the determination of the quadratic polynomial nature of the circular component functions.

For the quadratic polynomial projection method, we use orthogonal polynomials in two variables up to degree two on a discrete domain of finite size. Rather than using a discrete approximation to the continuous theory presented in Section IV-B-2, we use basis functions that are defined and orthonormal on a discrete grid of size $2n + 1$ by $2n + 1$, with interpixel sampling distance d. We assume that the grid is indexed by (i, j), where $-n \leq i, j \leq n$. The discrete basis functions can be derived quite easily:

$$\phi_1(i,j) = \frac{1}{D}, \quad \phi_2(i,j) = \frac{i}{D}, \quad \phi_3(i,j) = \frac{j}{D},$$

$$\phi_4(i,j) = \frac{i \cdot j}{D}, \quad \phi_5(i,j) = \frac{i^2 - C}{K}, \quad \phi_6(i,j) = \frac{j^2 - C}{K}$$

where D, C, and K are constants that depend only on n and d. The constants may be chosen so that the ϕ_i functions have unit norm and are orthogonal with respect to the inner product

$$\langle f, g \rangle = \sum_{i=-n}^{n} \sum_{j=-n}^{n} f(i,j) g(i,j). \tag{7.1}$$

Note that we use a standard inner product as opposed to a Hermite inner product. We compute

$$U_{(x_0,y_0)}(i,j) - \sum_{k=1}^{6} \langle U_{(x_0,y_0)}, \phi_k \rangle \phi_k(i,j).$$

The norm squared of this function, based on the inner product in (7.1) is the error measure $E(x_0, y_0)$. We correctly determine the FOE as the zero of this error function for the case where the camera is moving along a corridor with a translational velocity of $(5, 2, 20)$ and with rotational parameters $(0.1, 0.025, -0.05)$. The function $E(x_0, y_0)$ is shown in Fig. 17.

3) The Subspace Projection Method: We solve for a_1, a_2 and a_3 by solving the "normal equations" (4.14) of Section IV-B-3 using the inner product definition given in (7.1). The basis functions ϕ_i for $i = 1, 2, 3$ are simply point evaluates (on the discrete grid) of the functions defined by (4.13). We compute the inner product matrix Q explicitly instead of using the analytical expressions given in Section IV-B-3, again using the discrete inner product (7.1). Once we have a_1, a_2 and a_3, we can compute the error function

$$E(x_0, y_0) = ||U_{(x_0,y_0)} - \sum_{i=1}^{3} a_i \phi_i||^2.$$

Thus, $E(x_0, y_0)$ gives the norm of the residual after $U_{(x_0,y_0)}$ is differenced with its projection onto the subspace spanned by the basis functions (which are quadratic polynomials) defined at (x_0, y_0). Again, the norm is based on the inner product defined in (7.1). We determine the correct FOE for the case of a camera whose motion is the same as that described in Fig. 17. A contour plot of $E(x_0, y_0)$ is shown in Fig. 18.

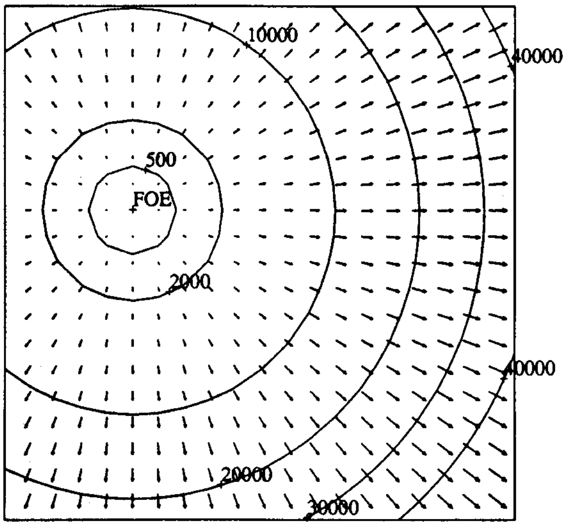


Fig. 17. Contour and gradient plot of the error function $E(x_0, y_0)$ obtained using the quadratic polynomial projection method for the image flow field induced by a translational velocity of (5, 2, 20) and rotational velocity of (0.1, 0.025, -0.05) for the corridor scene (Fig. 14).

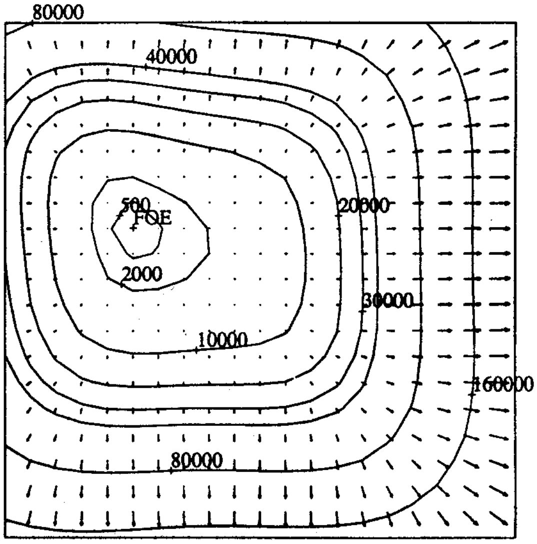


Fig. 18. Contour and gradient plot of the error function obtained using the subspace projection method using the same scene and motion described in Fig. 17.

We see that although the contours are less regular, the error increases rapidly as we move away from the true FOE, suggesting that the subspace projection method will yield the greatest accuracy.

VIII. Conclusions

We have presented two independent algorithms for the determination of motion parameters given the global optical flow field. The *flow circulation algorithm* determines rotational parameters of motion and works best when there is substantial rotational velocity of the sensor, and the scene consists of a lot of distant surfaces that either lie approximately normal to the viewing direction or whose tilts are well-distributed with respect to the direction of motion. The *FOE search algorithm*, on the other hand, determines the location of the FOE and works best when the translational velocity of the sensor dominates, and the scene contains plenty of depth variations.

Since both algorithms are global, they would ideally use large-field information. Further, the algorithms have been formulated for dense flow fields but, in practice, will have to deal with sparse data because accurate image motion data is available only where corners or other identifiable features can be tracked.

Our experimental results are hampered by the use of simulated data or by the limitations of actual data, which includes limited field of view and potentially inaccurate ground truth information provided for motion parameters only (and not depths). Nonetheless, the experiments validate the algorithms but fail to establish their stability with respect to noise, inaccuracies in the flow-field data, sparsity of the data, or other real-world exigencies. We are confident, however, that the global nature of the algorithms makes them more stable than motion parameter determination methods that rely on flow field data, and sometimes flow field derivatives, at a few image points.

Acknowledgment

We thank P. Anandan, R. Wildes, A. Dobbins, D. Heeger, and K. Wohn for comments and suggestions.

References

[1] D. Heeger and A Jepson, "Subspace methods for recovering rigid motion I: Algorithm and implementation," *Int. J. Comput. Vision*, vol. 7, no. 2, pp. 95–117, 1992.

[2] ——, "Simple method for computing 3D motion and depth," in *Proc. 3rd Int. Conf. Comput. Vision* (Osaka, Japan), Dec. 1990, pp. 96-100.

[3] G. Adiv, "Determining three-dimensional motion and structure from optical flow generated by several moving objects," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-7, pp. 384-401, 1985.

[4] S. Ullman, *The Interpretation of Visual Motion*. Cambridge, MA: MIT Press, 1979.

[5] R. Tsai and T. Huang, "Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-6, pp. 13-27, 1984

[6] O. Faugeras, F. Lustman, and G. Toscani, "Motion and structure from motion from point and line matches," in *Proc. First Int. Conf. Comput. Vision* (London), 1987, pp. 25-34.

[7] B. K. P. Horn and B. Schunk, "Determining optical flow," *Artificial Intell.*, vol. 17, pp. 185-203, 1981.

[8] S. Barnard and W. Thompson, "Disparity analysis of images," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 2, pp. 333–340, 1980.

[9] E. C. Hildreth, "Computations underlying the measurement of visual motion," *Artificial Intell.*, vol. 23, pp. 309-354, 1984.

[10] P. Anandan, "A computational framework and an algorithm for the measurement of visual motion," *Int. J. Comput. Vision*, vol. 2, pp. 283-310, 1989.

[11] D. J. Heeger, "Optical flow using spatiotemporal filters," *Int. J. Comput. Vision*, vol. 1, pp. 279-302, 1988.

[12] J. R. Bergen and E. H. Adelson, "Spatiotemporal energy models for the perception of motion," *J. Opt. Soc. Amer.*, vol. A2, pp. 284-299, 1985.

[13] J. J. Gibson, *The Perception of the Visual World*. Cambridge, MA: Riverside Press, 1950.

[14] H. von Helmholtz, *Handbuck der Physiologischen Optik*. Hamburg: Verlag von Leopold Voss, 1896; translation (J. P. C. Southall, Ed.). New York: Dover, 1925.

[15] J. J. Gibson, P. Olum, and F. Rosenblatt, "Parallax and perspective during aircraft landings", *Amer. J. Psych.*, vol. 68, pp. 372-385, 1955.

[16] J.J. Koenderink and A.J. Van Doorn, "Invariant properties of the motion parallax field due to the movement of rigid bodies relative to an observer," *Optica Acta*, vol. 22, no. 9, pp. 773-791, 1975.

[17] ——, "Local structure of movement parallax of the plane," *J. Opt. Soc. Amer.*, vol. 66, no. 7, 1976.

[18] H. C. Longuet-Higgins and K. Prazdny, "The interpretation of a moving retinal image," *Proc. Royal Soc. London, B*, vol. 208, pp. 385-397, 1980.

[19] K. Prazdny, "Determining the instantaneous direction of motion from optical flow generated by a curvilinearly moving observer," *Comput. Graphics Image Processing*, vol. 17, pp. 238-248, 1981.

[20] W. Burger and B. Bhanu, "Estimating 3-D egomotion from perspective image sequences," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 12, pp. 1040-1058, 1990.

[21] K. Prazdny, "Egomotion and relative depth map from optical flow," *Biolog. Cybern.*, vol. 36, pp. 87-102, 1980.

[22] ——, "On the information in optical flows," *Comput. Vision Graphics Image Processing*, vol. 22, pp. 239-259, 1983.

[23] D. Heeger and A. Jepson, "Visual perception of 3D motion," *Neural Computation*, vol. 2, pp. 127-135, 1990.

[24] A. Jepson and D. Heeger, "Subspace methods for recovering rigid motion II: Theory," submitted to *Int. J. Comput. Vision*, Univ. of Toronto, Res. Biolog. Comput. Vision Tech. Rep., RBCV-TR-90-36.

[25] J. H. Rieger and D. T. Lawton, "Determining the instantaneous axis of translation from optical flow generated by arbitrary sensor motion," in *Proc. ACM Interdisciplinary Workshop Motion* (Toronto), 1983, pp. 33-41.

[26] D. Lawton, "Processing translational image sequences," *Comput. Vision Graphics Image Processing*, vol. 22, pp. 116-144, 1983.

[27] M. Subbarao and A. Waxman, "On the uniqueness of image flow solutions for planar surfaces in motion," *Comput. Vision GraphicsImage Processing*, vol. 36, pp. 208-228, 1986.

[28] A. Waxman, B. Kamgar-Parsi, and M. Subbarao, "Closed form solutions to image flow equations," in *Proc. First Int. Conf. Comput. Vision* (London), 1987, pp. 12–24.

[29] A. Waxman and S. Ullman, "Surface structure and 3-D motion from image flow: A kinematic analysis," *Int. J. Robotics Res.*, vol. 4, no. 3, pp. 72-94, 1985.

[30] A. Waxman and K. Wohn, "Image flow theory: A framework for 3-D inference from time varying imagery," in *Advances in Computer Vision, vol. 1*. New York: Erlbaum, 1988, pp. 165–224.

[31] ——, "Contour evolution, neighborhood deformation, and global image flow: Planar surfaces in motion," *Int. J. Robotics Res.*, vol. 4, pp. 95–108, 1985.

[32] K. Wohn and A. Waxman, "The analytic structure of image flows: deformations and segmentation," *Comput. Vision Graphic Image Processing*, vol. 49, pp. 127-151, 1990.

[33] O. Faugeras, "On the motion of 3D curves and its relationship to optical flow," in *Proc. 1st ECCV*, 1990, pp. 105–117.

[34] C. Jerian and R. Jain, "Polynomial methods for structure from motion," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 12, pp. 1150-1166, 1990.

[35] J. J. Koenderink and A.J. Van Doorn, "Exterospecific component of the motion parallax field," *J. Opt. Soc. Amer.* , vol. 71, no. 8, pp. 953-957, 1981.

[36] D. Regan and K.I. Beverley, "Visual responses to vorticity and the neural analysis of optic flow," *J. Opt. Soc. Amer. A*, vol. 2, no. 2, pp. 280-283, Feb. 1985.

[37] K. Tanaka and H. Saito, "Analysis of motion of the visual field by direction, expansion/contraction, and rotation cells clustered in the dorsal part of the medial superior temporal area of the Macaque monkey," *J. Neurophys.*, vol. 62, no. 3, pp. 626 and 642, 1989.

[38] P. Werkhoven and J. J. Koenderink, "Extraction of motion parallax structure in the visual system I," *Biolog. Cybern.*, vol. 63, pp. 185-191, 1990.

[39] B. K. P. Horn, *Robot Vision*. Cambridge, MA: MIT Press, 1986.

[40] A. Mitiche, R. Grisell, and J. K. Aggarwal, "On the smoothness of a vector field-application to optical flow," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 10, no. 6, pp. 943-949, 1988.

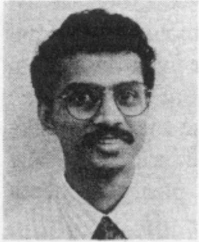
[41] K. Nakayama and J. Loomis, "Optical velocity patterns, velocity-sensitive neurons, and space perception: a hypothesis," *Perception*, vol. 3, pp. 63-80, 1974.

[42] V. Sundareswaran, "Egomotion from global flow field data," in *Proc. IEEE Workshop Visual Motion* (Princeton, NJ).

[43] ——, "Motion parameter estimation from global flow field data," Ph.D thesis, New York Univ., Oct. 1992 (available by anonymous ftp at cs. nyu. edu).

Robert Hummel (M'82) received the B.A. degree in mathematics from the University of Chicago and the Ph.D. degree in mathematics from the University of Minnesota.

He is an associate professor of computer science at New York University. Since 1980, he has been at the Courant Institute of Mathematical Sciences at New York University. His interests include computer vision, artificial intelligence and connectionism, and parallel computing. He is a member of the NYU Center for Neural Science and is involved in the planning for the NYU Center for Modeling and Simulation. In 1989, he was a visiting faculty member at Vrije Universiteit in Amsterdam and was on sabbatical visiting Project Epidaure at INRIA Roquencourt in France from 1991 to 1992.



V. Sundareswaran (S'90) received the Bachelor of Engineering degree from the University of Madras in 1984 and the Master of Engineering degree from the Indian Institute of Science, Bangalore, in 1987. He was a doctoral candidate in the Computer Science program in the Courant Institute at New York University when this work was performed.

He is now a postdoctural fellow at IRISA, Rennes, France. He has worked in the Indian software industry and served as a consultant in machine vision applications during the summers 1988-1990. His research interests include computer vision, visual psychophysics, neural networks, and artificial intelligence.