

Optical Flow: A Curve Evolution Approach

Arun Kumar, Allen R. Tannenbaum, and Gary J. Balas

Abstract—A novel approach for the computation of optical flow based on an L^1 type minimization is presented. It is shown that the approach has inherent advantages since it does not smooth the flow-velocity across the edges and hence preserves edge information. A numerical approach based on computation of evolving curves is proposed for computing the optical flow field. Computations are carried out on a number of real image sequences in order to illustrate the theory as well as the numerical approach.

I. INTRODUCTION

THE computation of optical flow has proved to be an important tool for problems arising in active vision. The optical flow field is defined as the velocity vector field of apparent motion of brightness patterns in a sequence of images [22]. It is assumed that the motion of the brightness patterns is the result of relative motion large enough to register a change in the spatial distribution of intensities on the images. Thus, relative motion between an object and a camera can give rise to optical flow. Similarly, relative motion among objects in a scene being imaged by a static camera can give rise to optical flow.

In this paper, we will consider a spatiotemporal differentiation method for optical flow (see the discussion below). Even though, in such an approach, the optical flow typically estimates only the isobrightness contours, it has been observed that if the motion gives rise to sufficiently large intensity gradients in the images, then the optical flow field can be used as an approximation to the real velocity field, and the computed optical flow can be used reliably in the solutions of a large number of problems; see [23], [60] and the references therein. Thus, optical flow computations have been used quite successfully in problems of 3-D object reconstruction and in 3-D scene analysis for computing information such as depths and surface orientation [37], [47]. In object tracking and robot navigation, optical flow has been used to track targets of interest. Discontinuities in optical flow have proved to be an important tool in approaching the problem of image-segmentation [24]–[26].

Manuscript received October 14, 1994; revised July 27, 1995. This work was supported in part by grants from the National Science Foundation DMS-8811084 and ECS-9122106, by the Air Force Office of Scientific Research AFOSR AF/F49620-94-1-00S8DEF, by the Army Research Office DAAH04-94-G-0054 and DAAH04-93-G-0332, and by Image Evolutions Limited. The associate editor coordinating the review of this paper and approving it for publication was Prof. Eric Dubois.

A. Kumar and G. Balas are with the Department of Aerospace Engineering, University of Minnesota, Minneapolis, MN 55455 USA (e-mail: arun@aem.umn.edu and balas@jette.aem.umn.edu).

A. R. Tannenbaum is with the Department of Electrical Engineering, University of Minnesota, Minneapolis, MN 55455 USA (e-mail: tannenba@ee.umn.edu).

Publisher Item Identifier S 1057-7149(96)02753-4.

The problem of computing optical flow is ill posed in the sense of Hadamard. Well-posedness has to be imposed by assuming suitable *a priori* knowledge [46]. In this paper, we concentrate on the variational formulation of imposing such *a priori* knowledge.

One constraint that has often been used in the literature is the “optical flow constraint” (OFC). The OFC is a result of the simplifying assumption of constancy of the intensity $E = E(x, y, t)$ at any point in the image [22]. It can be expressed as the following linear equation in the unknown variables u and v :

$$E_x u + E_y v + E_t = 0 \quad (1)$$

where E_x , E_y and E_t are the intensity gradients in the x , y , and the temporal directions, respectively, and u and v are the x and y velocity components of the apparent motion of brightness patterns in the images, respectively. It has been shown that the OFC holds, provided the scene has Lambertian surfaces and is illuminated by either a uniform or an isotropic light source, the 3-D motion is translational, the optical system is calibrated, and the patterns in the scene are locally rigid; see [8]. We should add (as pointed out by one of the referees) that since one typically does not have the choice of the temporal sampling rate, one does not have access to velocities but to displacements; therefore, the practical validity of the (OFC) equation when computing E_t from finite differences is subject to a slow motion assumption. In fact, this hypothesis is satisfied in the test sequences that we use in this paper; see Section VIII.

It is not difficult to see from (1) that computation of optical flow is unique only up to computation of the flow along the intensity gradient $\nabla E = (E_x, E_y)^T$ at a point in the image [22]. (The superscript T denotes “transpose.”) This is the celebrated *aperture problem*. One way of treating the aperture problem is through the use of regularization in computation of optical flow and, consequently, the choice of an appropriate constraint. A natural choice for such a constraint is the imposition of some measure of consistency on the flow vectors situated close to one another on the image.

In their pioneering work, Horn and Schunk [22] use a quadratic smoothness constraint. The immediate difficulty with this smoothness constraint is that at the object boundaries, where it is natural to expect discontinuities in the flow, such a constraint will have difficulty capturing the optical flow. For instance, in the case of a quadratic constraint in the form of the square of the norm of the gradient of the optical flow field [22], the Euler–Lagrange (partial) differential equations for the velocity components turn out to be *linear* elliptic. The corresponding parabolic equations therefore have a linear

diffusive nature and tend to blur the edges of a given image. In the past, work has been done to try to suppress such a constraint in directions orthogonal to the occluding boundaries in an effort to capture discontinuities in image intensities that arise on the edges; see Nagel *et al.* [43] and the references therein.

In this paper, we propose a novel method for computing optical flow based on the theory of the evolution of curves and surfaces. The approach employs an L^1 type minimization of the norm of the gradient of the optical flow vector rather than quadratic minimization, as has been undertaken in most previous regularization approaches. This type of approach has already been applied to derive a powerful denoising algorithm in Rudin *et al.* [49]. The equations that arise are *nonlinear* degenerate parabolic equations. The equations diffuse in a direction orthogonal to the intensity gradients i.e., in a direction along the edges. This results in the edges being preserved. The equations can be solved by following a methodology very similar to the evolution of curves based on the work of Osher and Sethian [44], [55], [56]. Proper numerical implementation of the equations leads to solutions that incorporate the nature of the discontinuities in image intensities into the optical flow.

The organization of this paper is as follows. In Section II, we give a brief survey of gradient-based approaches. In Section III, we review the relevant curve evolution theory and scale-spaces. In Section IV, we motivate the approach based on level curve evolution and discuss the more intrinsic dependency on the geometry of the optical flow that this approach allows. Section V gives a formulation for the optical flow problem whose solution will make contact with the nonlinear geometric scale-space ideas given in the previous sections. In Section VI, we explore the smoothing properties of the geometric heat equation while preserving edges. In Section VII, we discuss the numerical issues involved in the computation of the optical flow and give the algorithm used. Finally in Section VIII, we present results of experiments on images.

II. A BRIEF SURVEY OF GRADIENT-BASED APPROACHES

Numerous techniques for optical flow computation have been reported in the literature. Recently, an excellent survey of existing optical flow computation techniques has been done in Barron *et al.* [7]. They perform comparisons of eight key approaches from the category of gradient-based approaches, region-based matching, and energy- and phase-based methods.

The technique we propose in this paper belongs to the category of the classical gradient-based approaches. In this category, two main methods are discussed in the literature: *regularization* and *multiconstraint-based* approaches. The *regularization* approach involves imposing smoothness constraints on the optical flow. In the first such approach, Horn and Schunk [22] used a smoothness constraint that is quadratic in nature:

$$\iint \left((E_x u + E_y v + E_t)^2 + \alpha^2 (u_x^2 + u_y^2 + v_x^2 + v_y^2) \right) dx dy. \quad (2)$$

The first term puts a penalty on the optical flow constraint, and the second imposes a penalty on the departure from smoothness of the flow field with α as a weighting between these two terms. This approach gives rise to two equations at each pixel of the image that can be solved iteratively for the optical flow field. In an effort to capture discontinuities on object boundaries, Nagel and Enkelmann [43] studied orientation-dependent smoothness constraints and their computation. In Konrad and Dubois [33], two techniques are presented for motion estimation based on Bayesian estimators. In Schunk [53], an *extended optical flow constraint* has been derived that also involves the divergence of the optical flow field:

$$\nabla \cdot (E\vec{v}) + \frac{\partial E}{\partial t} = 0. \quad (3)$$

Bimbo *et al.* [8] study computational approaches to optical flow using the extended optical flow constraints. In Hildreth [21], optical flow is computed along contours rather than the entire object. This avoids the propagation effects from one region to another in the presence of object occlusions. We should also add that some work has recently been done in Luetggen *et al.* [38] on multiscale regularization ideas applied to optical flow.

Multiconstraint-based methods make use of the stationarity condition for other image properties like contrast, entropy, color spectrum, etc. Such constraints are imposed and a system of equations is obtained for u and v , which can be solved using traditional numerical methods. Such approaches have been explored in Markandey and Flichbaugh [40] and Mitiche *et al.* [41]. Other approaches in this category employ constraints with second-order partial derivatives of the image brightness; see Haralick and Lee [20], Tretiak and Pastor [59], and the references therein. In other related approaches to the computation of optical flow, researchers have used images illuminated with different sources of light; see Woodham [63]. An overdetermined set of equations is developed by using one optical flow constraint for each illumination and solved for the optical flow using standard techniques. Markandey and Flichbaugh [40] take images at different spectra and use one optical flow constraint equation for each spectrum.

A third set of approaches are called *multipoint*. The multipoint methods assume that the flow field is smooth and that constraints can be evaluated in a neighborhood of a pixel. This results in an overdetermined system of equations with constraints evaluated at more than one point; see Cafforio and Rocca [9] and Campani and Verri [10] for details on these techniques.

III. CURVE EVOLUTION THEORY AND SCALE-SPACES

The numerical approach we will use for the implementation of our optical flow algorithm is very closely related to the theory of evolution of level curves and, hence, a certain nonlinear scale-space [4], [3], [44]. Moreover, by putting optical flow in such a scale-space framework, we believe that one can glean new insight into its geometric structure. Thus, in this section, we briefly review work on curve evolution and the corresponding theory of scale-spaces of planar images.

We follow quite closely the treatment given in Sapiro and Tannenbaum [51], [50].

Multiscale representations have been the object of much research effort since the work of Witkin [62]. The basic concept of scale-space filtering is to filter a signal with a kernel that depends on a positive parameter representing the scale. Different values of the scale parameter result in a scale-space. More precisely, let $\Psi_0(\vec{X}) : \mathbf{R}^n \rightarrow \mathbf{R}^m$ be an initial signal, and let $\Upsilon(\vec{X}, t) : \mathbf{R}^n \times \mathbf{R}^+ \rightarrow \mathbf{R}^m$ be a kernel, where $t \in \mathbf{R}^+$ (i.e., $t \geq 0$) is the scale parameter. If $\Lambda_{\Upsilon(\vec{X}, t)}[\Psi_0(\vec{X})]$ represents the action of the filter $\Upsilon(\cdot)$, then the scale-space can be defined as

$$\Psi(\vec{X}, t) = \Lambda_{\Upsilon(\vec{X}, t)}[\Psi_0(\vec{X})]. \quad (4)$$

In the context of image processing, increased values of the parameter t can be thought of as representing an image at a coarser resolution. There are restrictions to be imposed on the kernels used to define a scale-space, a key one being the restriction of *causality*, i.e., that information cannot be created when passing from finer to coarser scales. For an extensive discussion of such issues, see [31], [45], [16]. Our aim, in this paper, is to examine the problem of optical flow determination from a perspective based on scale-spaces and the theory of curve evolution. This work provides good insight as to how the problem formulation should be carried out. We will restrict our attention to the classical and the geometric heat equations that are discussed below.

The simplest kernel that generates a (nontrivial) scale-space is the Gaussian kernel [32]. The resulting scale-space is linear, and the filter is defined by convolution. One of the properties of the Gaussian kernel is that the filtered signal Ψ can also be obtained as the solution of the heat equation with the initial condition Ψ_0

$$\frac{\partial \Psi}{\partial t} = \Delta \Psi \quad \text{with} \quad \Psi(\vec{X}, 0) = \Psi_0 \quad (5)$$

where $\Delta(\cdot) = \frac{\partial^2(\cdot)}{\partial x^2} + \frac{\partial^2(\cdot)}{\partial y^2}$ denotes the Laplacian. The Gaussian kernel has been classically used in image processing for smoothing images, in spite of the fact that it blurs the image, resulting in a loss of edge information. Moreover, mathematically, it possesses properties that are undesirable when applied to planar shapes, e.g., it is a filter that is not *intrinsic* to the geometry of the shape. Let us elucidate this point more precisely by relating it to the theory of planar curve evolution.

Consider a family of simple closed planar curves depending on a parameter t , which can be thought of as time or as scale. Denote the family of such curves by $C(p, t) : S^1 \times [0, \tau) \rightarrow \mathbf{R}^2$, where p is independent of t . Let the evolution of this family of curves be governed by the following evolution equation:

$$\frac{\partial C}{\partial t} = \frac{\partial^2 C}{\partial p^2} \quad \text{with} \quad C(p, 0) = C_0(p) \\ C(p, t) = [x(p, t), y(p, t)]^T. \quad (6)$$

(Here, the superscript T denotes "transpose.") As above, the solution vector of this heat equation $[x(p, t), y(p, t)]^T$ can be

obtained by convolution with the Gaussian $G(p, t)$, which is given by

$$G(p, t) = \frac{1}{\sqrt{4\pi t}} \exp\left\{\frac{-p^2}{4t}\right\} \quad (7)$$

where t is the variance of the Gaussian filter. In this case, this results in the classical linear Gaussian scale-space with t as the scale-space parameter.

In order to define an intrinsic geometric scale-space for planar shapes, we must first introduce the *arc-length parametrization*. This parametrization is the one in which the curve is traversed at unit speed. More precisely, define the *Euclidean arc length* as

$$s(p) = \int_0^p \left\| \frac{\partial C(\xi)}{\partial \xi} \right\| d\xi. \quad (8)$$

Then

$$\|C_s\| = \left\| \frac{\partial C}{\partial s} \right\| = 1 \quad (9)$$

and hence, as required, the curve is traversed at constant velocity of unity. Reparametrization of curves by using the Euclidean arc length forms the basis of the *Euclidean* or *geometric scale-space*.

Consider the following general evolution equation:

$$\frac{\partial C}{\partial t} = \alpha \vec{T} + \beta \vec{N} \\ C(p, 0) = C_0(p) \quad (10)$$

where

- \vec{N} inward Euclidean unit normal,
- \vec{T} unit tangent,
- α tangential component of the evolution velocity,
- β normal component of the evolution velocity [58].

It has been shown in [13] that if β is chosen as a geometric quantity for a curve, i.e., a function independent of the particular parametrization, then the tangential velocity does not affect the geometry of the solutions of (10) in the sense that the trace of the solutions of the following two equations—(11) and (12)—are the same:

$$C_t = \alpha \vec{T} + \beta \vec{N} \quad (11)$$

$$\dot{C}_t = \beta \vec{N}. \quad (12)$$

(Of course, we assume the same initial curve.) Thus, from the point of view of shape, it is sufficient to consider evolution equations of the form

$$\frac{\partial C}{\partial t} = \beta \vec{N} \quad (13)$$

where β is a geometric quantity and can be viewed as the projection of the velocity vector in the normal direction. Evolution of curves with different normal velocities has been the subject of research in a number of papers. The theory has been well developed in diverse areas such as computational physics [44], [55], [57], differential geometry [17], [18], [61], numerical analysis [56], parabolic equations [5], [6], and viscosity solutions [12], [15]. In addition, evolution equations have been used to model different physical phenomena such

as crystal growth [57], the Huygen's principle [55], and curve shortening processes [5], [17]–[19], [52]. Using these ideas, Kimia *et al.* have constructed a scale-space—the *reaction-diffusion scale-space*—that can be used in formulating a theory of shape. We will be concerned with the scale-space obtained by the evolution equation when the normal velocity $\beta = \kappa$, where κ is the Euclidean curvature defined by (see [58])

$$\kappa = \left\| \frac{\partial^2 C}{\partial s^2} \right\|. \quad (14)$$

(Note that as above, ds denotes Euclidean arc length.) The resulting equation is

$$\frac{\partial C}{\partial t} = \kappa \vec{N}. \quad (15)$$

The flow in (15) has been called the *Euclidean shortening flow*. It has been shown in Gage and Hamilton [17] and Grayson [18] that a planar embedded curve shrinks to a round point when evolving according to (15). Discussions on the Euclidean curve shortening flow can be found in [1], [5], [6], [13], [17], and [61]. Now, it is a classical fact [58] that

$$\kappa \vec{N} = \frac{\partial^2 C}{\partial s^2}. \quad (16)$$

Hence, (15) results in the *geometric heat flow*

$$C_t = C_{ss}. \quad (17)$$

Notice that the equation above is not linear since the arc length s is a function of time for an evolving curve. A complete proof that the scale-space obtained satisfies all the basic properties for a scale-space is given in [15], [17]–[19], and the references therein. This nonlinear scale-space has been studied in relation to shape theory in [28]–[30] and 42]. It will appear naturally as well in our approach to optical flow.

IV. QUADRATIC REGULARIZATION REVISITED

In the next two sections, we wish to examine the problem of the determination of optical flow from a slightly different point of view. The approach we give to computing optical flow is motivated using ideas from curve evolution theory and scale-spaces. Consider the classical regularization of the optical flow problem with a quadratic functional [22]:

$$\min_{(u,v)} \int_{\Omega} (u_x^2 + u_y^2 + v_x^2 + v_y^2) dx dy \quad (18)$$

subject to

$$E_x u + E_y v + E_t = 0 \quad (19)$$

where $E = E(x, y, t)$ denotes the intensity.

This type of constraint seeks to improve some measure of consistency on the flow vectors that are close to each other. The domain Ω for the minimization problem is the 2-D image. Define the *smoothing error*

$$\epsilon_s = \int_{\Omega} (u_x^2 + u_y^2 + v_x^2 + v_y^2) dx dy \quad (20)$$

and the *image motion error*

$$\epsilon_c = \int_{\Omega} (E_x u + E_y v + E_t)^2 dx dy. \quad (21)$$

In the classical Horn and Schunk [22] approach, the following problem is solved where the minimization is performed over the entire image Ω :

$$\min_{(u,v)} (\epsilon_s + \alpha^2 \epsilon_c). \quad (22)$$

The parameter α has the important role of weighting the error in the image motion relative to the departure from smoothness. The value of α is usually specified based on the given problem. The Euler–Lagrange equations for this problem can be written as follows:

$$F_u - \frac{\partial F_{u_x}}{\partial x} - \frac{\partial F_{u_y}}{\partial y} = 0 \quad (23)$$

$$F_v - \frac{\partial F_{v_x}}{\partial x} - \frac{\partial F_{v_y}}{\partial y} = 0 \quad (24)$$

with

$$F = (u_x^2 + u_y^2 + v_x^2 + v_y^2) + \alpha^2 (E_x u + E_y v + E_t)^2. \quad (25)$$

The boundary conditions are the Neumann boundary conditions $\frac{\partial u}{\partial n} = 0$ and $\frac{\partial v}{\partial n} = 0$. This leads to the following equations in Horn and Schunk [22]:

$$\Delta u = \alpha^2 (E_x u + E_y v + E_t) E_x \quad (26)$$

$$\Delta v = \alpha^2 (E_x u + E_y v + E_t) E_y \quad (27)$$

where $\Delta(\cdot) = \frac{\partial^2(\cdot)}{\partial x^2} + \frac{\partial^2(\cdot)}{\partial y^2}$ denotes the Laplacian.

To derive some insight into the nature of the solutions of this set of equations, we can consider the associated set of (linear) parabolic equations; in other words, we try to solve (26) and (27) using *gradient descent*. Accordingly, we introduce a new “scale-space” variable t' (independent of t), and we consider linear scale-space type equations given by

$$\hat{u}_{t'} = \Delta \hat{u} - \alpha^2 (E_x \hat{u} + E_y \hat{v} + E_t) E_x \quad (28)$$

$$\hat{v}_{t'} = \Delta \hat{v} - \alpha^2 (E_x \hat{u} + E_y \hat{v} + E_t) E_y \quad (29)$$

where $\hat{u} = \hat{u}(x, y, t')$ and similarly for \hat{v} . Note that the steady-state solution (that is, the limiting solution as $t' \rightarrow \infty$) of the coupled pair of linear parabolic equations (28) and (29) is the desired solution of the Euler–Lagrange equations (26) and (27). (This is exactly the same relationship that the steady-state solution of the classical heat equation has with that of the Laplace equation. See [34] for the precise technical conditions guaranteeing convergence.) Recall that α is the factor that weights the relative importance of the error in the *image motion* (i.e., a measure of how well the optical flow constraint is satisfied) against departure from smoothness.

Clearly, as $\alpha \rightarrow \infty$, we approach the optical flow constraint (19). For $\alpha = 0$, we derive

$$\hat{u}_{t'} = \Delta \hat{u} \quad (30)$$

$$\hat{v}_{t'} = \Delta \hat{v} \quad (31)$$

which is, of course, a pair of ordinary heat equations. This generates a linear scale-space and is equivalent to convolution with a Gaussian filter. However, this inherits all the undesirable properties of such a scale-space. The solution would tend to fill areas inwards from the boundaries. For any finite value of α , the solution will have similar behavior. It is clear that in this framework, regardless of the value of α , important edge information will be lost.

V. L^1 MINIMIZATION

We are now ready to give a formulation for the optical flow problem whose solution will make contact with the nonlinear geometric scale-space ideas given above. Intuitively, it is desired that the "geometry" of the optical flow dictated by the sharp intensity gradients in the images should be captured by the regularization. An L^1 -type optimization problem is successful in satisfying this intuition and is directly related to the geometric scale-space developed in Section III. It has been argued that L^1 approximations give better visual features than the classical L^2 . Indeed, the L^1 norm of the gradient has been found to be very useful for image denoising; see Rudin *et al.* [44]. For further mathematical details about the geometry of optimization problems relative to various L^p norms, see Evans [14].

We will therefore use the following L^1 norm smoothing error to develop insights into the optical flow problem:

$$\epsilon_s = \int \int_{\Omega} \left(\sqrt{u_x^2 + u_y^2} + \sqrt{v_x^2 + v_y^2} \right) dx dy. \quad (32)$$

Notice that in the case of a single variable, the smoothing error in this form gives the L^1 norm of the gradient of the optical flow

$$\epsilon_s = \int (|u_x| + |v_x|) dx. \quad (33)$$

Consider the regularization of the optical flow using the following cost functional:

$$\min_{(u,v)} \int \int_{\Omega} \left(\sqrt{u_x^2 + u_y^2} + \sqrt{v_x^2 + v_y^2} + \alpha^2 (E_x u + E_y v + E_t)^2 \right) dx dy.$$

It is easy to see that in this case, we get the following Euler-Lagrange equations

$$\begin{aligned} -\alpha^2 E_x (E_x u + E_y v + E_t) + \frac{\partial}{\partial x} \left(\frac{u_x}{\sqrt{u_x^2 + u_y^2}} \right) \\ + \frac{\partial}{\partial y} \left(\frac{u_y}{\sqrt{u_x^2 + u_y^2}} \right) = 0, \end{aligned} \quad (34)$$

$$\begin{aligned} -\alpha^2 E_y (E_x u + E_y v + E_t) \frac{\partial}{\partial x} \left(\frac{v_x}{\sqrt{v_x^2 + v_y^2}} \right) \\ + \frac{\partial}{\partial y} \left(\frac{v_y}{\sqrt{v_x^2 + v_y^2}} \right) = 0. \end{aligned} \quad (35)$$

As we show below, we can identify the "curvature terms" in these equations very easily and rewrite the equations as

$$\kappa_u - \alpha^2 E_x (E_x u + E_y v + E_t) = 0, \quad (36)$$

$$\kappa_v - \alpha^2 E_y (E_x u + E_y v + E_t) = 0 \quad (37)$$

where the curvatures κ_u and κ_v are given by (see [58])

$$\kappa_u = \frac{u_x^2 u_{yy} - 2u_x u_y u_{xy} + u_y^2 u_{xx}}{(u_x^2 + u_y^2)^{3/2}} = \operatorname{div} \left(\frac{\nabla u}{\|\nabla u\|} \right) \quad (38)$$

and

$$\kappa_v = \frac{v_x^2 v_{yy} - 2v_x v_y v_{xy} + v_y^2 v_{xx}}{(v_x^2 + v_y^2)^{3/2}} = \operatorname{div} \left(\frac{\nabla v}{\|\nabla v\|} \right). \quad (39)$$

κ_u and κ_v given in these equations are precisely the curvatures of level sets as we show now; see also [2]–[4], [11], [44], [55], [56].

Consider the level set of the differentiable function $\Psi(x, y)$

$$C_k = \{(x, y) : \Psi(x, y) = k\} \quad (40)$$

where k is a constant. An arc-length parametrization for C_k is chosen by

$$C_k = \{(x(s), y(s)) : s \in [0, \mathbf{L}(C_k)]\} \quad (41)$$

where $\mathbf{L}(\cdot)$ denotes the length of the curve. Since the level set is given as

$$\Psi(x(s), y(s)) = k = \text{constant} \quad (42)$$

differentiating gives

$$\Psi_x x_s + \Psi_y y_s = 0. \quad (43)$$

Hence, we have

$$y_s = \lambda \Psi_x \text{ and } x_s = -\lambda \Psi_y. \quad (44)$$

In addition,

$$x_{ss} = -\lambda^2 (\Psi_{yy} \Psi_x - \Psi_{xy} \Psi_y) \text{ and } y_{ss} = \lambda^2 (\Psi_{xy} \Psi_x - \Psi_{xx} \Psi_y). \quad (45)$$

The curvature κ of a curve $(x(p), y(p))^T$ at a point is given by

$$\kappa = \frac{x_p y_{pp} - y_p x_{pp}}{(x_p^2 + y_p^2)^{3/2}}. \quad (46)$$

Hence, via a simple computation, we may write the curvature of the implicitly defined curve C_k as follows:

$$\kappa_{\Psi} = \frac{\Psi_{xx} \Psi_y^2 - 2\Psi_{xy} \Psi_x \Psi_y + \Psi_{yy} \Psi_x^2}{(\Psi_x^2 + \Psi_y^2)^{3/2}} = \operatorname{div} \left(\frac{\nabla \Psi}{\|\nabla \Psi\|} \right). \quad (47)$$

Now, as in the quadratic case, we want to use gradient descent to solve equations (36) and (37). Thus, once again, we introduce a new scale-space variable t' and derive the following nonlinear parabolic evolution equations:

$$\hat{u}_{t'} = \kappa_u - \alpha^2 E_x (E_x \hat{u} + E_y \hat{v} + E_t) \quad (48)$$

$$\hat{v}_{t'} = \kappa_v - \alpha^2 E_y (E_x \hat{u} + E_y \hat{v} + E_t) \quad (49)$$

for $\hat{u} = \hat{u}(x, y, t')$ and similarly for \hat{v} . As before, the steady-state solution of the system (48) and (49) is precisely the

solution of the Euler–Lagrange equations (36) and (37). Note that in the limit that $\alpha \rightarrow 0$, we get the equations:

$$\hat{u}_t = \kappa_{\hat{u}} = \frac{\kappa_{\hat{u}}}{\|\nabla \hat{u}\|} \|\nabla \hat{u}\| \quad (50)$$

$$\hat{v}_t = \kappa_{\hat{v}} = \frac{\kappa_{\hat{v}}}{\|\nabla \hat{v}\|} \|\nabla \hat{v}\|. \quad (51)$$

We have now come full circle. As we will see in Section VII, the numerical implementation of the geometric heat equation (17) is derived by embedding the evolution as a level curve evolution of an associated family of surfaces defined by $\Phi(x, y, t) : \mathbf{R}^2 \times [0, \tau] \rightarrow \mathbf{R}$. In fact, the corresponding evolution of Φ that induces (17) on the level curves is precisely

$$\Phi_t = \kappa_{\Phi} \|\nabla \Phi\|, \quad (52)$$

where, as above

$$\kappa_{\Phi} = \operatorname{div} \left(\frac{\nabla \Phi}{\|\nabla \Phi\|} \right). \quad (53)$$

Thus, we will see that one can use similar numerical techniques as those developed in Osher and Sethian [44] for computing evolution of propagating curves in order to find an efficient reliable algorithm for numerically solving the system (48) and (49). Moreover, as we will see in the next section, diffusion equations such as (50) and (51) when applied to a 2-D image have been proven to smooth on both sides of an edge while preserving the edge itself. Therefore, solutions of the optical flow based on (48) and (49) are expected to pick out discontinuous edges.

Remark: We have considered another L^1 -type minimization problem

$$\min_{(u,v)} = \int \int_{\Omega} (\sqrt{u_x^2 + u_y^2 + v_x^2 + v_y^2} + \alpha^2 (E_x u + E_y v + E_t)^2) dx dy.$$

This minimization problem performs in practice similar to the one discussed in this paper.

Remark: In a very nice paper, Evans [14] considers the following set of optimization problems. For $\Omega \subset \mathbf{R}^2$ open and $\omega = (u, v) \in \Omega$, consider minimizing the energy

$$\|\nabla \omega\|_{L^p(\Omega)}$$

for $1 \leq p \leq \infty$. Then, the Euler–Lagrange equation for this variational problem leads to a p -Laplacian operator, say Δ_p . In the present paper, we have considered exactly the case $p = 1$. As we have seen, the equation

$$\Delta_1 \omega = 0$$

is geometric in the sense that it says that the level sets of ω have zero mean curvature. For an extensive discussion about all of this, see Evans [14].

VI. EDGE-PRESERVING SMOOTHING

Classically, smoothing has been done using the linear heat equation, i.e., filtering with a Gaussian kernel. However, smoothing with the Gaussian operator smoothes isotropically and, thus, tends to blur edges. To alleviate this problem, we would like a filter in which the smoothing is minimal across edges and maximal along the direction of the edges. The idea of smoothing without losing important edge information can be made precise in the following mathematical form. We want a diffusion equation that does not diffuse in the direction of the gradient $\nabla \Phi$. This can be seen to be (see [2], [45])

$$\Phi_t = \Delta \Phi - \frac{1}{\|\nabla \Phi\|^2} \langle \nabla^2 \Phi (\nabla \Phi), \nabla \Phi \rangle \quad (54)$$

where

$$\nabla^2 \Phi (\nabla \Phi) = \begin{bmatrix} \Phi_{xx} & \Phi_{xy} \\ \Phi_{xy} & \Phi_{yy} \end{bmatrix} \begin{pmatrix} \Phi_x \\ \Phi_y \end{pmatrix} \quad (55)$$

i.e., the Hessian $\nabla^2 \Phi$ is being applied to the gradient vector $\nabla \Phi$. Note that $\langle \cdot, \cdot \rangle$ denotes the usual inner product in \mathbf{R}^2 .

Via a simple computation, this equation may be written as

$$\Phi_t = \|\nabla \Phi\| \operatorname{div} \left(\frac{\nabla \Phi}{\|\nabla \Phi\|} \right). \quad (56)$$

Hence, based on (53), we can identify the curvature and write this diffusion equation as

$$\Phi_t = \kappa_{\Phi} \|\nabla \Phi\|. \quad (57)$$

Now, in fact, the (50) and (51) have the form

$$\Phi_t = \frac{\kappa_{\Phi}}{\|\nabla \Phi\|} \|\nabla \Phi\|. \quad (58)$$

Since $\|\nabla \Phi\|$ is maximal at an edge, (58) will smooth while preserving edges. (Diffusion at the edge is even slower than that of (57).) Moreover, the numerical techniques of Osher–Sethian [44] are valid for the numerical implementation of (58); see [49] and Section VII. Hence, the optical flow equations (48) and (49) we have derived in the previous section based on the L^1 norm allow us to retain edges in the computation of the optical flow.

VII. NUMERICAL IMPLEMENTATION

Unfortunately, a straightforward finite-difference approach to numerically solve for the optical flow from (48) and (49) does not work very well. The motivation for the appropriate numerical method comes from the problem of curves evolving with normal velocity as a function of the curvature. In this section, we will describe issues that are important in the numerical implementation of curve evolution. Problems can occur that are not obvious at the outset, and we shall attempt to briefly point them out. For more detailed treatments, see the work of Osher and Sethian [54], [55], [44] on which all of

the results of this section are based. In what follows, we will closely follow the discussion of [51].

It is very natural for evolving curves to develop singularities. For example, it is known that the curvature of a smooth nonconvex closed curve evolving according to equation

$$\frac{\partial C}{\partial t} = \beta(\kappa)\vec{N} \quad (59)$$

blows up in finite time for $\beta = +1$. After the singularity develops, it is not obvious how the evolution should be carried forward since the solution does not exist in the classical sense. The idea used to deal with this problem is to impose an *entropy condition* on the evolution. The entropy condition employed here is that for a propagating flame front formulated by Sethian: *A particle once burnt remains burnt*; see [54]–[56]. Kimia *et al.* [27], [29] have demonstrated the usefulness of this solution in their diffusion scale-space framework for shape, while Malladi *et al.* [39] have used it for shape modeling.

The second problem with evolution of curves is that topological changes may occur as the evolution proceeds. For example, a simply connected region enclosed by an evolving curve might get “broken up” into two or more simply connected regions. Further, since the numerical implementation of curve evolution usually proceeds with a suitable discretization of the curve, there is the issue of accurate and stable discretizations. Simple Lagrangian difference approximations do not work well since, as the computation proceeds, the marker particles on the evolving curve may come very close to one another. Thus, the constraint of stability requires taking very small time steps, making the computation impractically slow; see Sethian [56].

Osher and Sethian in [44], [55], [56] give powerful algorithms for the propagation of curves and surfaces that incorporate the above-mentioned entropy condition. They develop equations of motion for propagating fronts with curvature-dependent speed that are Hamilton–Jacobi equations with parabolic right-hand sides and then develop numerical schemes using techniques from the theory of hyperbolic conservation laws to successfully compute the evolutions. They show that by embedding the curve in a 2-D surface and then evolving the surface rather than the curve, one derives robust, stable, and reliable evolution algorithms. As we show below, this will give the correct evolution of the curve when the curve is viewed as a level set of the embedding surface. This method takes care of the topological changes that might occur as the curve evolves: The embedding surface remains smooth even though the level set representing the curve may split into several closed curves.

We shall now indicate how the evolution of a curve under equation (59) may be recovered as the level set of an evolving surface [44], [55], [2], [11]. Let us consider the curve $C(p, t)$ embedded as the zero level set of a surface defined by a smooth and Lipschitz continuous surface $\Phi : \mathbf{R}^2 \times [0, \tau) \rightarrow \mathbf{R}$. Without loss of generality, we may assume that Φ is negative in the interior and positive in the exterior of the zero level set. One way to construct this surface is to consider the distance of each point from the level set curve boundary and assign an appropriate sign to the distance, depending upon the position of the point relative to the curve’s interior. Now, consider the

zero level set $\vec{Z}(t)$ of the surface $\Phi(t)$ defined as

$$\{\vec{Z}(t) \in \mathbf{R}^2 : \Phi(\vec{Z}, t) = 0\}. \quad (60)$$

The evolution of the surface Φ will be carried out so that the evolving zero level set of the surface gives the evolution of the curve, i.e.

$$C(t) \equiv \vec{Z}(t) \forall t \geq 0. \quad (61)$$

Differentiating equation (60) with respect to t gives

$$\nabla\Phi(\vec{Z}, t) \cdot \vec{Z}_t + \Phi_t(\vec{Z}, t) = 0. \quad (62)$$

Now, for a level set, the following relation holds:

$$\frac{\nabla\Phi}{\|\nabla\Phi\|} = -\vec{N} \quad (63)$$

where \vec{N} is the unit normal. Notice that in (63), the surface is on the left side of the equation, whereas the right side is related to the curve C . Using (59) and (63), we see that

$$\Phi_t = -\nabla\Phi \cdot \vec{Z}_t = -\nabla\Phi \cdot C_t = (-\nabla\Phi \cdot \vec{N})\beta(\kappa) = \beta(\kappa)\|\nabla\Phi\|. \quad (64)$$

Thus, from (64) and (57), it is clear that the problem of smoothing while preserving edges can be solved by carrying out evolution of curves embedded in higher dimensional surfaces. In an earlier section, it had already been established that there is a connection between the optical flow problem and the evolution of curves through the geometric heat equation. Thus, it seems natural to derive the numerical algorithms to compute optical flow based on algorithms for curve evolution.

Accordingly, we apply the aforementioned numerical methods of Osher and Sethian to the optical flow equations (48) and (49) given in Section V, where the parameter t' plays the role of a gradient descent (scale-space) parameter along which the evolution is performed. The discretization of t' is, of course, dictated by the given problem, and an exercise of choice will have to be made in its selection. Similarly, we will assume that α is a fixed positive number, and the choice here is also problem dependent. Note that each of the optical flow velocity components are considered to be functions of the spatial coordinates and scale-space parameter, i.e., $\hat{u}(\cdot)$ and $\hat{v}(\cdot)$ are $\hat{u}(x, y, t')$, $\hat{v}(x, y, t')$.

The numerical algorithm we use is similar to the one developed in Rudin *et al.* [49]. As pointed out in [49], the advantage of this approach is that it allows us to compute the optical flow based on the gradient projection method presented in [48]. The 2-D image is discretized with the grid points placed at the pixel locations; Δx and Δy represent the step sizes in the two directions, which we take to be equal, i.e.,

$$\Delta x = \Delta y = h.$$

A discretization in the temporal direction is introduced, and the time step Δt is used to compute the temporal intensity gradient change at each pixel. Notice that we will also need a discretization $\Delta t'$ for the scale parameter t' . We shall briefly

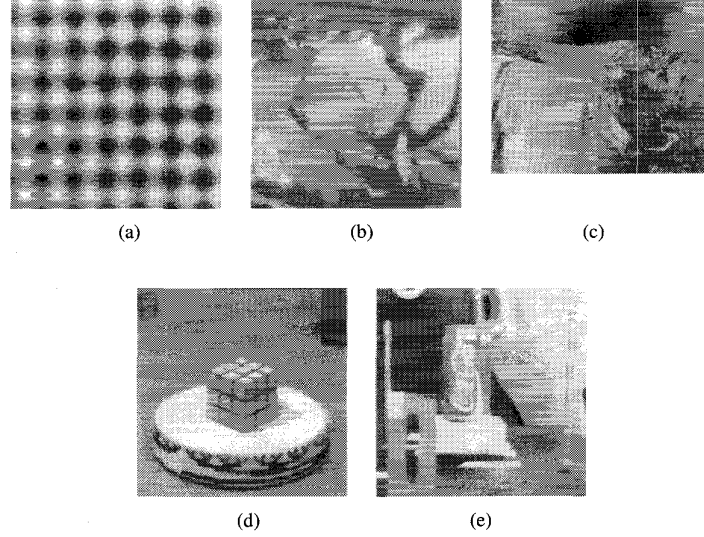


Fig. 1. Frames from (a) sinusoidal, (b) diverging tree, (c) Yosemite, (d) Rubic's cube, and (e) NASA sequences.

indicate how we use simple discretizations to compute the gradients at the point $(i\Delta x, j\Delta y, k\Delta t)$:

$$\begin{aligned} E_x(i, j, k) &\approx 0.25(E(i+1, j, k) - E(i, j, k) \\ &\quad + E(i+1, j, k+1) - E(i, j, k+1)), \\ E_y(i, j, k) &\approx 0.25(E(i, j+1, k) - E(i, j, k) \\ &\quad + E(i, j+1, k+1) - E(i, j, k+1)), \\ E_t(i, j, k) &\approx 0.5(E(i, j, k+1) - E(i, j, k)). \end{aligned}$$

(Note that we are setting

$$E_x(i, j, k) = E_x(i\Delta x, j\Delta y, k\Delta t)$$

and similarly for the other expressions.)

Following [49], we discretize our equations in the following manner. The selection of the derivatives in the numerical scheme is important to introduce proper upwinding into the equations. We now write down the discretized expression for (50). The discretization of (51) (the equation for \hat{v}) is identical. Moreover, the extension to equations (48) and (49) ($\alpha \neq 0$) is obvious. Note that we are setting

$$\hat{u}_{i,j,k'} = \hat{u}(i\Delta x, j\Delta y, k'\Delta t').$$

Accordingly, we take

$$\hat{u}_{i,j,k'+1} = \hat{u}_{i,j,k'} + \frac{\Delta t'}{h} \left[\Delta_-^x \left(\frac{\Delta_+^x \hat{u}_{i,j,k'}}{(\rho_{\hat{u}}^x)_{i,j,k'}} \right) + \Delta_-^y \left(\frac{\Delta_+^y \hat{u}_{i,j,k'}}{(\rho_{\hat{u}}^y)_{i,j,k'}} \right) \right]$$

where

$$(\rho_{\hat{u}}^y)_{i,j,k'} = [(\Delta_+^x \hat{u}_{i,j,k'})^2 + (m(\Delta_+^y \hat{u}_{i,j,k'}, \Delta_-^y \hat{u}_{i,j,k'}))^2]^{1/2}. \quad (65)$$

and

$$(\rho_{\hat{u}}^x)_{i,j,k'} = [m(\Delta_+^x \hat{u}_{i,j,k'}, \Delta_-^x \hat{u}_{i,j,k'})^2 + (\Delta_+^y \hat{u}_{i,j,k'})^2]^{1/2}.$$

Here

$$\begin{aligned} \Delta_{\mp}^x \hat{u}_{i,j,k'} &= \mp(\hat{u}_{i\mp 1, j, k'} - \hat{u}_{i, j, k'}) \\ \Delta_{\mp}^y \hat{u}_{i,j,k'} &= \mp(\hat{u}_{i, j\mp 1, k'} - \hat{u}_{i, j, k'}) \end{aligned}$$

and the function $m(\cdot, \cdot)$ is defined as

$$m(a, b) = \left(\frac{\text{sgn}(a) + \text{sgn}(b)}{2} \right) \min(|a|, |b|).$$

As is usual in the numerical solutions of hyperbolic conservation laws, a Courant-Friedrichs-Lewy (CFL) step-size restriction is imposed on the numerical schemes [36]. In [49], in order to guarantee stability, a CFL step-size restriction of the following form is suggested:

$$\frac{\Delta t'}{h^2} \leq c. \quad (66)$$

(This is quite standard with the parabolic type equations we are considering here; see [36].) In our numerical computations, we follow such a restriction.

VIII. RESULTS OF EXPERIMENTS

In this section, we present experimental simulations performed on selected sequences of images. Results are presented for five sets of image sequences, all of which are taken from Barron *et al.* [7]. Three sequences are synthetic, and they also serve as test cases for error analysis of the optical flow fields. Two sequences are real image sequences.

A. Synthetic Image Sequences and Error Analysis of Flow Fields

The following three synthetic image sequences have been used for analysis of the flow fields produced by the technique presented in this paper.

Sinusoidal Sequence: The sinusoidal sequence is a synthetic image sequence and consists of the superposition of two sinusoidal plane waves:

$$\sin(k_1 \cdot \vec{x} + \omega_1 t) + \sin(k_2 \cdot \vec{x} + \omega_2 t)$$

The spatial wavelength is taken to be 16 pixels/cycle, and the orientations of the two waves is taken as 54° and -27° . The velocity vector is $\vec{v} = (1, 1)$; see Fig. 1(a).

Diverging Tree Sequence: This is also a synthetic image sequence in which the camera is moved along the line of sight, and the focus of expansion is taken as the center of the image. The flow velocities vary from 1.29 pixels/frame on the left side to 1.86 pixels/frame on the right side; see Fig. 1(b).

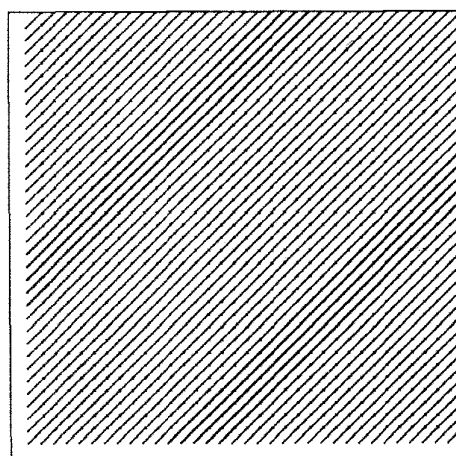
Yosemite Sequence: This synthetic sequence has divergent motion in the upper right, the clouds themselves undergo translational motion to the right with a speed of 2 pixels/frame, and speeds on the lower left are about 4–5 pixels/frame; see Fig. 1(c).

We have observed that for most cases, no presmoothing is required to produce acceptable flow fields. In fact, presmoothing the images using Gaussian filter tends to blur edges and attenuates the advantages of using curve evolution. (A key feature of the L^1 minimization is to preserve edge information.) Thresholding is done in some cases to produce visually pleasing flow fields, but otherwise, it is found that the technique works quite well without any thresholding either in the intensity values or the flow velocity values. In our simulations, we have found that simple centered differencing, both spatial and temporal, is quite adequate to produce good flow fields. In some cases where there are large number of spatial pockets of intensity gradients, it might be advisable to use averages over a small neighborhood of the computation molecule to get a satisfactory global flow field. This is because the algorithm is designed to capture intensity gradients that are local in nature.

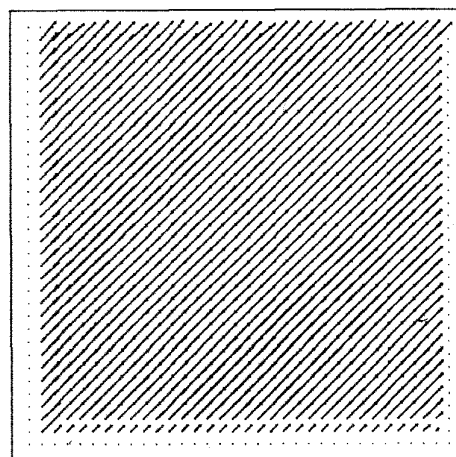
As we mentioned before, it is standard to impose the CFL restriction when evolving the equations due to stability requirements. In addition, we have the α parameter, which, we have observed, behaves like a scale-space parameter describing the optical flow field at various amounts of smoothing. The choice of α determines how sharply the resulting flow field reproduces the edge information present in the images. The optical flow computed based on curve evolution is less diffusive and quite sensitive to edge information in the images. However, recall that varying the value of α in the evolution equations gave a scale-space with varying degrees of smoothness in the flow field. We have exploited this property of the evolution in computing the flow fields for the synthetic image sequences. For the results presented for the synthetic images, the values of α used were in the range of 30–50.

This was done to successfully capture the global nature of the fields, in particular, the translational field in the sinusoidal sequence of Fig. 2 and the divergent field in the diverging tree sequence of Fig. 3. For the Yosemite sequence of Fig. 4, in areas of the image where there were substantial gradients, the global flow field is faithfully captured. In other areas, the flow field produced is quite sparse. This example also suggests that an α that varies over the image might be used to better capture the global field, which changes in nature over the image (translational and divergent in different parts of the image).

The time steps used for all the results in this paper were in the range 0.00001–0.001. Note that too large a time step may cause instability since the solution from one cell may travel into the neighboring cells. The space steps used were in the range 0.1–1.0. Again, these values were chosen on a case-by-case basis.



(a)



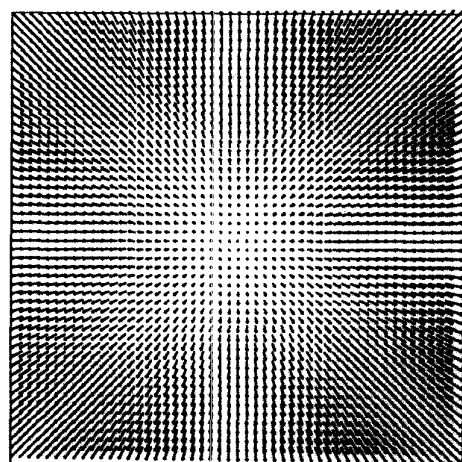
(b)

Fig. 2. Comparison of the sinusoidal synthetic sequence: (a) True flow field; (b) flow field using curve evolution.

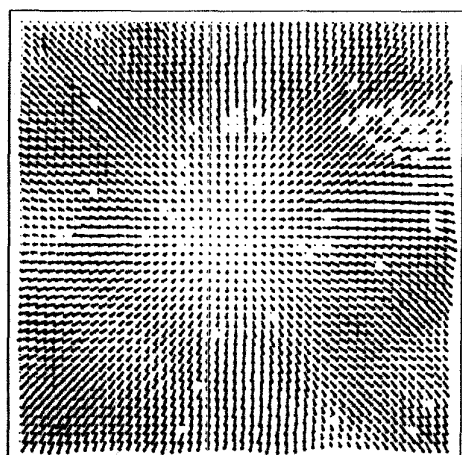
We employed a measure of accuracy based on the angular deviations of the estimated flow field from the true flow field as in [7]. If v^c and v^e denote the correct and estimated flow fields, respectively, then the measure of the angle between the augmented vectors $(v_x^c, v_y^c, 1.0)^T$ and $(v_x^e, v_y^e, 1.0)^T$ is independent of the size of the individual components of the vectors. The angle may be defined using the inner product as follows:

$$\theta = \cos^{-1} \left(\frac{v_x^e \cdot v_x^c + v_y^e \cdot v_y^c + 1.0}{\sqrt{(v_x^{c2} + v_y^{c2} + 1) \cdot (v_x^{e2} + v_y^{e2} + 1)}} \right). \quad (67)$$

However, this measure is quite limited in scope and can only be used with a sequence of frames in which the correct flow field is known. We also note here that for the technique proposed in this paper, the above measure might result in large angular errors since the flow fields generated tend to be sharply aligned with the intensity edges causing local departures from the constant global flow fields. Of course, as discussed above,



(a)

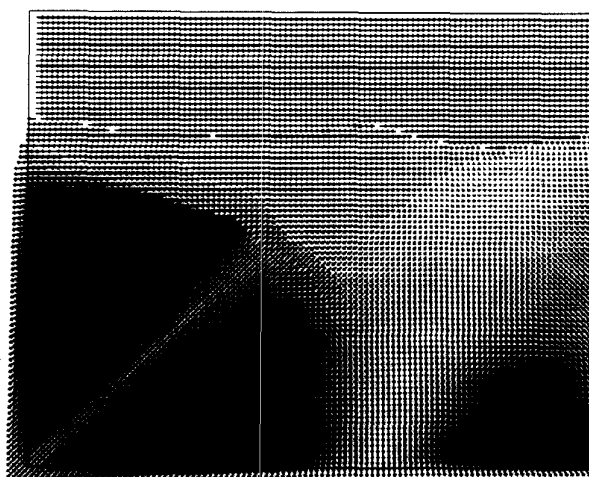


(b)

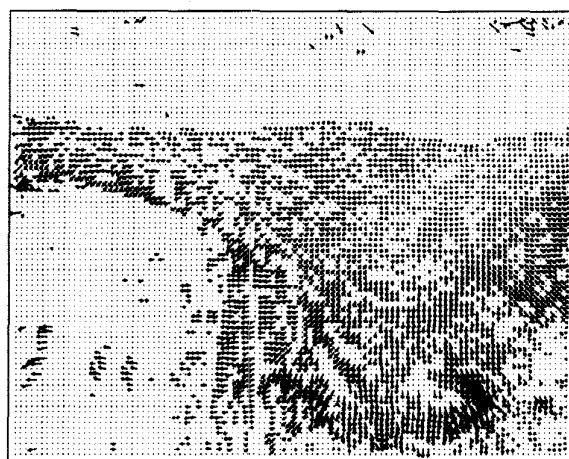
Fig. 3. Comparison of the diverging tree synthetic sequence: (a) True flow field; (b) flow field using curve evolution.

it is possible to obtain smoother flow fields by tuning the α scale-space parameter.

Table I reports angular error estimates for curve evolution and the quadratic approach of Horn and Schunk. For a complete tabulation of various optical flow techniques, see [7]. We present here error estimates for the modified Horn and Schunk. The modified Horn and Schunk approach, which involves considerable preprocessing of the images, was developed in [7] and is an improvement over the original method. In this approach, spatio-temporal smoothing is applied to the images, and then, optical flow field is computed. Curve evolution-based optical flow computation does not need spatial smoothing to be applied. In fact, the smoothing using a Gaussian filter smoothes edges and would not utilize the full potential of our method, which is to preserve the edge information in the computed field. In cases where there is large motion occurring between frames however, it might prove helpful if some amount of smoothing is performed in the temporal direction. This would allow information from the previous and future frames to be processed simultaneously. We use



(a)



(b)

Fig. 4. Comparison of the Yosemite synthetic sequence: (a) True flow field; (b) flow field using curve evolution.

temporal smoothing and simple differencing to compute the errors reported here. For the Horn and Schunk modified method, the thresholding is done at $\|\nabla\| \geq 5.0$ (as suggested in [7]). For curve evolution, the thresholding values for the results reported here were 8.0 (Yosemite) and 10.0 (diverging tree). It is seen that the curve evolution-based optical flow field is more accurate than the modified Horn and Schunk. In fact, on comparison with other techniques investigated in [7], it is found that curve evolution gives better error estimates than a majority of the techniques for optical flow computation.

B. Real Image Sequences

Rotating Rubik's Cube Sequence: This sequence shows a Rubik's cube rotating counterclockwise on a turntable; see Fig. 1(d). It is primarily a rotational motion field. Velocities on the turntable are between 1.2 to 1.4 pixels/frame, whereas on the cube, they are between 0.2 to 0.5 pixels/frame.

TABLE I
PERFORMANCE OF OPTICAL FLOW ALGORITHM BASED ON CURVE EVOLUTION

Angular Error Estimates For Synthetic Images				
Sequence	Method	Mean error	Deviation	density
SineC	Horn and Schunk (modified)	2.55°	0.59°	100.0 %
SineC	Curve Evolution	1.00°	0.5°	100.0 %
Diverging Tree	Horn and Schunk (modified, thresholded)	1.94°	3.89°	32.9 %
Diverging Tree	Curve Evolution (thresholded)	0.76°	1.12°	32.96 %
Yosemite Sequence	Horn and Schunk (modified, thresholded)	5.59°	11.52°	32.9 %
Yosemite Sequence	Curve Evolution(thresholded)	4.04°	9.50°	34.08 %

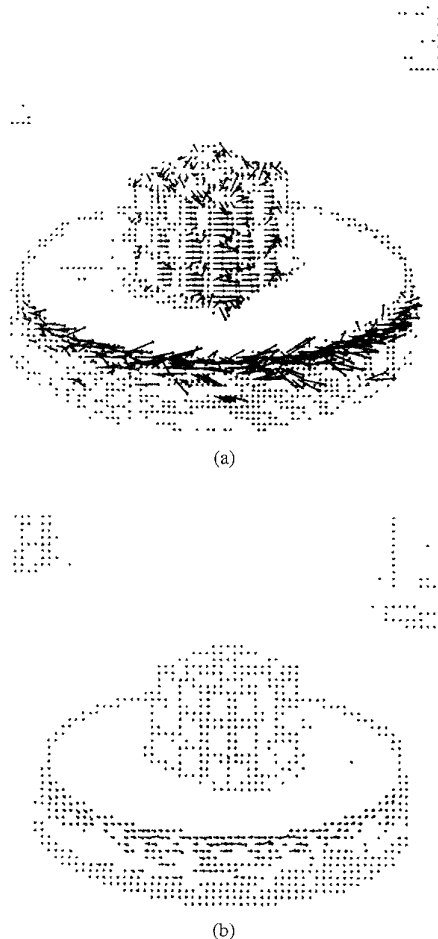


Fig. 5. Optical flow (intensity thresholded) using (a) curve evolution and (b) Horn and Schunk.

NASA Sequence: The NASA sequence is a real image sequence showing dilational motion. The camera moves along its line of sight toward the Coke can, which is kept close to the center of the image; see Fig. 1(e). Image velocities of less than 1 pixel/frame are reported.

For the flow fields computed from the real sequences, we have kept low values of α to closely follow the edges. Satisfactory values were found to be in the range of 2–5. For the two real sequences, we have performed the flow computation using the Horn and Schunck method for the sake of comparison. The flow fields for the Horn and Schunck approach have been computed using the software of Barron *et al.* [7], which is available via anonymous ftp from *ftp.csd.uwo.ca*

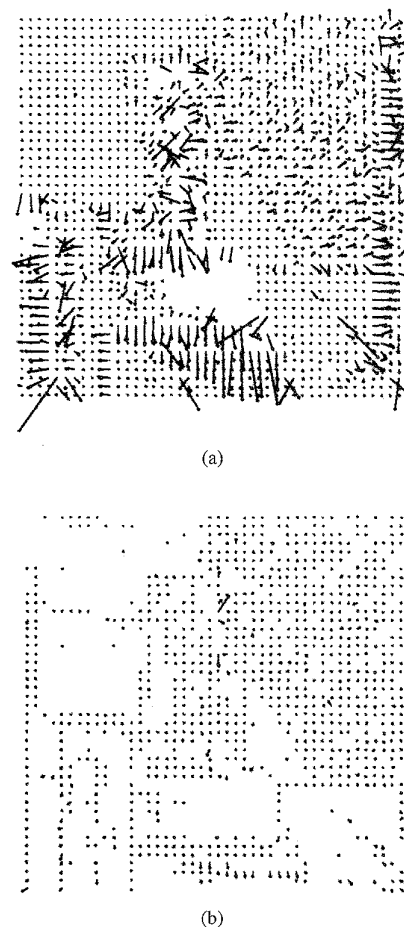


Fig. 6. Optical flow (intensity thresholded) using (a) curve evolution and (b) Horn and Schunk.

in the directory *pub/vision*. The flow fields for the Horn and Schunck approach have been computed using the original method described in [22], i.e., no presmoothing was performed on images, and simple differencing was used. The flow fields have been intensity thresholded at $\|\nabla E\| > 5.0$ for the sake of presentation.

It is observed that the L^1 minimization approach based on curve evolution is sensitive to intensity gradients and is much less diffusive in its computations of the flow fields. In the real image sequence of the Rubik's cube placed on a rotating turntable, the flow field computed shows sensitivity to the intensity gradients (due to the local patterns on the turntable's edge) while at the same time reproducing the global motion of the turntable; see Fig. 5(a). On the cube itself, the L^1 approach is able to capture the rotating faces and the top surface quite well. Notice that the flow vectors on the top face of the cube are longer toward the circumference than in the center. On the other hand, the diffusive action in the Horn and Schunck's approach gives a very smooth flow field on the base of the turntable while barely capturing the motion of the cube itself; see Fig. 5(b). In the dilational motion in the NASA sequence, the L^1 -based approach captures the global motion as well as the edges in the scene; see Fig. 6(a). The Horn and Schunck

method, on the other hand, is seen to capture the global motion but is much more diffusive, and hence, much edge information is lost; see Fig. 6(b).

IX. CONCLUSIONS

In this paper, we have developed a novel approach for the computation of optical flow from sequence of images. We motivate the use of an L^1 -norm minimization of the flow vectors based on the ideas of scale-spaces and the geometric heat equation. It has been shown that this approach has inherent advantages since it does not smooth the flow velocity across the edges and, hence, is likely to preserve edge information. The numerical approach to solving the resulting equations is implemented using numerical approaches from the theory of evolution of curves. The results of applying the approach to some standard image sequences have been presented, and a very brief comparison is made with the classical Horn and Schunk approach for the purposes of showing the advantages to be gained by such an approach. Results show that L^1 -norm minimization is successful in capturing local intensity gradient information and, hence, edges while at the same time preserving the global motion field. This, in general, is not true of the approaches based on quadratic regularization. Finally, in [35], we are now applying similar techniques to the problem of stereo disparity.

ACKNOWLEDGMENT

The authors would first like to thank Dr. G. Sapiro of Hewlett-Packard for a number of very valuable suggestions and conversations toward the writing of this paper. In particular, they thank him for pointing out to them the key reference Barron *et al.* [7], and the corresponding software. They would like to gratefully acknowledge Prof. T. Shield and R. Muno of the University of Minnesota for help with image acquisition on the Indigo video board in the early stages of this work. Finally, they would like to thank the referees for their very helpful comments.

REFERENCES

- [1] U. Abresch and J. Langer, "The normalized curve shortening flow and homothetic solutions," *J. Diff. Geom.*, vol. 23, p. 175–196, 1986.
- [2] L. Alvarez, F. Guichard, and J. M. Morel, "Image selective smoothing and edge detection by nonlinear diffusion," *SIAM J. Numerical Anal.*, vol. 29, pp. 845–866, 1992.
- [3] L. Alvarez, F. Guichard, P. L. Lions, and J. M. Morel, "Axiomatization et nouveaux operateurs de la morphologie mathematique," *C. R. Acad. Sci. Paris*, vol. 315, pp. 265–268, 1992.
- [4] ———, "Axiomes et equations fondamentales du traitement d'images," *C. R. Acad. Sci. Paris*, vol. 315, p. 135–138, 1992.
- [5] S. Angenent, "Parabolic equations for curves on surfaces, I: Curves with p -integrable curvature," *Ann. Math.*, vol. 133, pp. 451–483, 1990.
- [6] ———, "Parabolic equations for curves on surfaces, II: Intersections, blow-up, and generalized solutions," *Ann. Math.*, vol. 133, pp. 171–215, 1991.
- [7] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, "Performance of optical flow techniques," *Int. J. Comput. Vision*, vol. 12, pp. 43–77, 1994.
- [8] A. D. Bimbo, P. Nesi, and J. L. C. Sanz, "Optical flow computation using extended constraints," Tech. Rep., Dept. Syst. Informatics, Univ. of Florence, 1992.
- [9] C. Cafforio and F. Rocca, "Tracking moving objects in television images," *Signal Processing*, vol. 1, pp. 133–140, 1979.
- [10] M. Campani and A. Verri, "Computing optical flow from an overconstrained system of linear algebraic equations," in *Proc. Third IEEE Int. Conf. Comput. Vision (ICCV)*, Osaka, Japan, 1990, pp. 22–26.
- [11] V. Caselles, F. Catte, T. Coll, and F. Dibos, "A geometric model for active contours in image processing," *AMS*, vol. 29, pp. 1–10, 1993.
- [12] M. G. Crandall, H. Ishii, and P. L. Lions, "Users guide to viscosity solutions of second order partial linear differential equations," *Bull. Amer. Math. Soc.*, vol. 27, pp. 1–67, 1992.
- [13] C. L. Epstein and M. Gage, "The curve shortening flow," *Wave Motion: Theory, Modeling, and Computation*, A. Chorin and A. Majda, Eds. New York: Springer-Verlag, 1987.
- [14] L. C. Evans, "Estimates for smooth absolutely minimizing Lipschitz extensions," *Electron. J. Diff. Eq.*, vol. 1, pp. 1–9, 1993.
- [15] L. C. Evans and J. Spruck, "Motion of level sets by mean curvature, I," *J. Diff. Geom.*, vol. 33, pp. 635–681, 1991.
- [16] L. Florack, "The syntactical structure of scalar images," Ph.D. Dissertation, Univ. Utrecht, The Netherlands, Nov. 1993.
- [17] M. Gage and R. S. Hamilton, "The heat equation shrinking convex plane curves," *J. Diff. Geom.*, vol. 23, pp. 285–314, 1986.
- [18] M. Grayson, "The heat equation shrinks embedded plane curves to round points," *J. Diff. Geom.*, vol. 26, pp. 285–314, 1987.
- [19] ———, "Shortening embedded curves," *Ann. Math.*, vol. 129, pp. 71–111, 1989.
- [20] R. M. Haralick and J. S. Lee, "The facet approach to optical flow," in *Proc. Image Understanding Workshop (Science Applications)*, Arlington, VA, Dec. 1983.
- [21] E. C. Hildreth, "Computations underlying the measurement of visual motion," *Artif. Intell.*, vol. 23, pp. 309–354, 1984.
- [22] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artif. Intell.*, vol. 23, pp. 185–203, 1981.
- [23] B. K. P. Horn, *Robot Vision*. Cambridge, MA: MIT Press, 1986.
- [24] R. Jain, W. N. Martin, and J. K. Aggarwal, "Segmentation through the detection of changes due to motion," *Comput. Graphics Image Processing*, vol. 11, no. 1, pp. 13–34, 1979.
- [25] R. Jain, D. Militzer, and H.-H. Nagel, "Separating nonstationary from stationary scene components in a sequence of real world TV-images," in *Proc. 5th Int. Joint Conf. Artificial Intell.*, Aug. 1979, pp. 612–618.
- [26] R. Jain and H.-H. Nagel, "On the analysis of accumulative difference pictures from image sequences of real world scenes," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-1, no. 2, pp. 206–214, 1979.
- [27] B. B. Kimia, "Toward a computational theory of shape," Ph.D. dissertation, McGill Univ., Dept. Elect. Eng., Aug. 1990.
- [28] B. B. Kimia, A. Tannenbaum, and S. W. Zucker, "Shapes, shocks and deformations-I," *Int. J. Comput. Vision*, vol. 15, pp. 189–224, 1996.
- [29] ———, "Toward a computational theory of shape: An overview," in *Lecture Notes in Computer Science*, vol. 427. New York: Springer-Verlag, 1990, pp. 402–407.
- [30] ———, "On the evolution of curves via a function of curvature-I: The classical case," *J. Math. Anal. Applications*, vol. 163, pp. 438–458, 1992.
- [31] J. J. Koenderink, "The structure of images," *Biol. Cybern.*, vol. 50, pp. 363–370, 1984.
- [32] ———, *Solid Shape*. Cambridge, MA: MIT Press, 1990.
- [33] J. Konrad and E. Dubois, "Multigrid Bayesian estimation of the image motion fields using stochastic relaxation," in *Proc. Second IEEE Int. Conf. Comput. Vision ICCV*, Tampa, FL, 1988, pp. 354–362.
- [34] N. V. Krylov, *Nonlinear Elliptic and Parabolic Equations of the Second Order*. Boston, MA: Kluwer, 1987.
- [35] A. Kumar, A. Tannenbaum, and S. Zucker, "Stereo disparity and L^1 minimization," in preparation.
- [36] R. J. LeVeque, *Numerical Method for Conservation Laws*. Boston, MA: Birkhauser, 1992.
- [37] C. Longuet-Higgins and K. Prazdny, "The interpretation of a moving retinal image," in *Proc. Royal Soc. London B*, vol. 208, pp. 385–397, 1980.
- [38] M. Luetzgen, W. C. Karl, and A. Willsky, "Efficient multiscale regularization with applications to the computation of optical flow," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-3, pp. 41–63, 1984.
- [39] R. Malladi, J. A. Sethian, and B. C. Vemuri, "Shape modeling with front propagation: a level set approach," *IEEE Trans. Patt. Anal. Machine Intell.*, to be published.
- [40] V. Markandey and B. E. Flichbaugh, "Multispectral constraints for optical flow computation," in *Proc. Third IEEE Int. Conf. Comput. Vision ICCV*, Osaka, Japan, Dec. 1990, pp. 38–41.
- [41] A. Mitche, Y. F. Wang, and J. K. Aggarwal, "Experiments in computing optical flow with the gradient-based multiconstraint method," *Patt. Recogn.*, vol. 20, no. 2, pp. 173–179, 1987.
- [42] F. Mokhtarian and A. Mackworth, "A theory of multiscale, curvature-based shape representation for planar curves," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 14, pp. 789–805, 1992.

- [43] H.-H. Nagel and W. Enkelmann, "An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-8, no. 5, pp. 565-593, 1986.
- [44] S. J. Osher and J. A. Sethian, "Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations," *J. Comput. Phys.*, vol. 79, pp. 12-49, 1988.
- [45] P. Perona and J. Malik, "Scale space and edge detection using anisotropic diffusion," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 12, pp. 629-639, 1990.
- [46] T. Poggio, V. Torre, and C. Koch, "Computational vision and regularization theory," *Nature*, vol. 317, no. 26, pp. 314-319, Sept. 1985.
- [47] K. Prazdny, "On the information in optical flows," *Comput. Vision, Graphics, Image Processing*, vol. 23, pp. 239-259, 1983.
- [48] J. G. Rosen, "The gradient projection method for nonlinear programming, II: Nonlinear constraints," *J. Soc. Indust. Appl. Math.*, vol. 9, pp. 514-532, 1961.
- [49] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, vol. 60, pp. 259-268, 1993.
- [50] G. Sapiro and A. Tannenbaum, "Area and length preserving geometric invariant scale-spaces," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 17, pp. 67-72, 1995.
- [51] ———, "Affine invariant scale-space," *Int. J. Comput. Vision*, vol. 11, no. 1, pp. 25-44, 1993.
- [52] ———, "On affine plane curve evolution," *J. Functional Anal.*, vol. 119, no. 1, pp. 79-120, 1994.
- [53] B. G. Schunck, "The motion constraints equation for optical flow," in *Proc. Seventh IEEE Int. Conf. Patt. Recogn.*, 1984, pp. 20-22.
- [54] J. A. Sethian, "An analysis of flame propagation," Ph.D. Dissertation, Univ. of California, Berkeley, June 1982.
- [55] ———, "Curvature and the evolution of fronts," *Commun. Math. Phys.*, 101, pp. 487-499, 1985.
- [56] ———, "A review of recent numerical algorithms for hypersurfaces moving with curvature dependent speed," *J. Diff. Geom.*, vol. 31, pp. 131-161, 1989.
- [57] J. A. Sethian and J. Strain, "Crystal growth and dendritic solidification," *J. Computat. Phys.*, vol. 98, 1992.
- [58] M. Spivak, *A Comprehensive Introduction to Differential Geometry*. Berkeley, CA: Publish or Perish, 1979.
- [59] O. Tretiak and L. Pastor, "Velocity estimation from image sequences with second order differential operators," in *Proc. Seventh IEEE Int. Conf. Patt. Recogn.*, 1984, pp. 16-29, 1984.
- [60] A. Verri and T. Poggio, "Motion field and optical flow: Qualitative properties," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 11, pp. 490-498, 1989.
- [61] B. White, "Some recent developments in differential geometry," *Math. Intelligencer*, vol. 11, pp. 41-47, 1989.
- [62] A. P. Witkin, "Scale-space filtering," in *Int. Joint. Conf. Artificial Intell.*, 1983, pp. 1019-1021.
- [63] R. J. Woodham, "Multiple light source optical flow," in *Proc. Third IEEE Int. Conf. Comput. Vision (ICCV)*, Osaka, Japan, Dec. 1990, pp. 42-46.



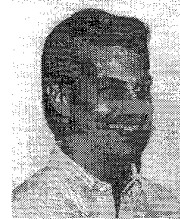
Arun Kumar received the B.Tech degree in mechanical engineering from the Indian Institute of Technology, Kanpur, in 1990 and the M.S. degree from the University of Minnesota, Minneapolis, USA, in 1993. He is currently pursuing the Ph.D. degree.

His research interests are image processing and computer vision, medical image analysis and visualization, neural network applications in machine vision, modern robust control theory, and object-oriented graphical user interface design.



Allen Tannenbaum was born in New York, NY, USA, in 1953. He received the Ph.D. degree in mathematics from Harvard University, Cambridge, MA, in 1976.

He has held faculty positions at the Weizmann Institute of Science; McGill University, Montreal, Canada; the University of Florida, Gainesville, USA; ETH, Zurich, Switzerland; Technion, Haifa, Israel; and Ben-Gurion University of the Negev, Israel. He is presently a Professor of Electrical Engineering at the University of Minnesota, Minneapolis, USA. He has authored or co-authored over 150 research papers (mostly in refereed journals) and is the author or co-author of three books: *Invariance and Systems Theory*, *Feedback Control Theory* (with J. Doyle and B. Francis), and *H[∞] of Distributed Systems* (with C. Foias and H. Ozbay). He has done research in systems and control, operator theory, algebraic geometry, robotics, image processing, computer vision, and invariant theory.



Gary J. Balas received the B.S. and M.S. degrees in civil and electrical engineering from the University of California at Irvine, USA, and the Ph.D. degree in Aeronautics from the California Institute of Technology, Pasadena, USA, in 1990.

Since 1990, he has been a faculty member with the Department of Aerospace Engineering and Mechanics at the University of Minnesota, Minneapolis, USA, and from 1993 to 1995, he held the McKnight-Land Grant Professorship. He is a co-organizer and developer of the MUSYN Robust Control Short Course and the μ -Analysis and Synthesis Toolbox used with MATLAB. He is the president of MUSYN Inc. His research interests include control of flexible structures, aircraft, model validation, and industrial applications of robust control methods.