JOURNAL OF VISUAL COMMUNICATION AND IMAGE REPRESENTATION
Vol. 4, No. 4, December, pp. 309–323, 1993

Computation of Dense Optical Flow with a Parametric Smoothness Model[1]

NAVID HADDADI[2] AND C.-C. JAY KUO[2]

Signal and Image Processing Institute and Department of Electrical Engineering–Systems, University of Southern California, Los Angeles, California 90089

Received October 23, 1992

A new algorithm is presented for the computation of dense optical flow and motion boundaries from an image sequence using two or more frames. The algorithm is based on a novel parametric smoothness model; it decomposes optical flow into irrotational and solenoidal fields, and imposes the smoothness constraint on each field separately. This model implies smooth translation and rotation of the underlying motion process. In contrast, the smoothness constraints used in all previous work do not distinguish the translational and rotational components but simply combine them as a whole. The derivation of the parametric smoothness model sheds new light on the interpretation of the conventional membrane model. The problem of oversmoothing across motion boundaries can be resolved to a high degree by successively improving the estimate of the parameters of the smoothness model. Significant improvements over classical gradient based methods have been obtained for a class of test problems by the proposed new algorithm. © 1993 Academic Press, Inc.

1. INTRODUCTION

Optical flow is a 2-D vector field that measures the disparity between adjacent frames in a sequence of images. The information contained in the computed flow field can be used for image sequence compression [22, 23] or the determination of the relative depth map and/or three-dimensional motion and structure [1]. It should be noted that, in general, the shape of the underlying scene cannot be uniquely determined from optical flow without additional visual cues [2]; however, in motion compensated coding, optical flow can be considered to be a good estimate of motion vectors since only the compensation residual needs to be minimized. In this paper, we develop an algorithm for computation of a discontinuous optical flow which is not directly based on discontinuous regularization via line processes [7].

Since the flow field is usually not unique over extensive regions of its support, there have been many attempts to limit the scope of estimation to regions for which only a good estimate can be obtained. These techniques range from computation of the flow only for highly conspicuous points [4] to flow along some boundaries [8, 15, 29] or some regions obtained by thresholding an appropriate function [10, 19]. Although we do not explicitly address any application issues in this paper, our intended target is motion compensated coding. For such an application, we need an estimate of the flow over the entire region of support, which is often known as dense optical flow. In this paper, we are mainly concerned with the computation of dense optic flow from at least two frames of an image sequence. The desired algorithm should have the following properties.

- An estimate of the flow should be available for all of the pixels in the region of support with the exception of occluded regions which must be identified.
- The algorithm is able to provide a robust estimate of the flow from two frames and improved estimation quality with more frames. This requirement is needed due to practical limitation on the size of the frame storage buffer in actual implementation.
- The complexity of the algorithm should be low and, hopefully, in the order of N^2 operations per image frame of size $N \times N$.
- The algorithm is capable of estimating subpixel flow fields. This is useful for applications where only a low resolution estimate of the flow is needed so that computational time can be significantly reduced in processing the subsampled image sequence.

The algorithm presented in this paper meets all of the above requirements. It is based on a regularization approach similar to previous work in [16, 20] but with a novel parametric smoothness model consisting of two parameter functions $\rho(x, y)$ and $\omega(x, y)$—we denote these functions as (ρ, ω). Borrowing techniques from fluid mechanics, we decompose optical flow into two flow fields, the irrotational and solenoidal fields, and impose the smoothness constraint on each field separately. This im-

[1] This work was supported by a National Science Foundation Young Investigator Award (ASC-9258396).

[2] E-mail addresses: haddadi@sipi.usc.edu and cckuo@sipi.usc.edu.

1047-3203/93 $5.00

plies smooth translation and rotation of the underlying motion process. In contrast, the smoothness constraints used in all previous work do not distinguish the translational and rotational components but simply combine them as a whole. The conventional membrane model is shown to be equivalent to the choice of $(\rho, \omega) = 0$ in our model. An accurate estimate of the parameters ρ and ω can greatly enhance the computed flow field. The problem of oversmoothing across motion boundaries can be resolved to a high degree, and an estimate of occluding boundaries is obtained as a byproduct. Since the nature of motion is characterized by ρ and ω, we may incorporate a prior knowledge of motion obtained from earlier frames in an image sequence into ρ and ω by assuming that they are smooth time-varying functions. In this work, we focus on the two frame case where an iterative algorithm is proposed for estimating ρ and ω.

This paper is organized as follows. Gradient based methods for optical flow computation are reviewed in Section 2. To put various problems arising in optical flow computation into perspective, some fundamental characteristics of optical flow are discussed in Section 3. Section 4 forms the crux of our method, where a parametric smoothness model is derived. In Section 5, we apply our algorithm to a set of test problems ranging from a simple translation and rotation to combined motion along all axes. We end our presentation in Section 6 with a few concluding remarks. Numerical implementation of the algorithm is given in the Appendix. In particular, we derive equations for the Gauss–Seidel relaxation that is appropriate for multigrid implementation.

2. REVIEW OF GRADIENT BASED METHODS FOR COMPUTING OPTICAL FLOW

There have been many efforts reported in the literature for computing optical flow. Most of these methods can be classified into one of four major categories: gradient based [16], contour based [8, 15, 29], correlation based [3, 26], and spatiotemporal frequency based [14]. Since our method falls into the gradient based category, its basic idea and some related results are reviewed in this section. For notational simplicity, the argument of various functions of spatial and temporal coordinates, (x, y, t), are dropped whenever such arguments are clear from the context.

Given an image sequence $E(x, y, t)$, we assume that

$$\frac{dE}{dt} = 0. \tag{2.1}$$

Expanding this equation in partial derivatives, one obtains the so-called *optical flow constraint equation*

$$E_x u + E_y v + E_t = 0, \tag{2.2}$$

where the subscript denotes partial differentiation with respect to the given variable and the optical flow (u, v) are the velocities in the x and y directions, respectively—i.e.,

$$(u, v) = \left(\frac{dx}{dt}, \frac{dy}{dt}\right). \tag{2.3}$$

Equation (2.2) can alternatively be obtained as the first order approximation of a translation model

$$E(x, y, t + \delta t) = E(x - u\,\delta t, y - v\,\delta t, t), \tag{2.4}$$

as shown in [16]. It is evident that the optical flow constraint imposes only one constraint for two unknown variables per image pixel, so that we need at least one more constraint to uniquely determine the flow field. All gradient based methods use the optical flow constraint equation, but consider different ways to impose additional constraints.

One way to avoid the underdetermination problem is to impose a constant velocity constraint for neighboring flow fields. Several methods were proposed under this assumption. The local optimization method assumes two or more neighboring pixels to have the same displacement value so that a set of linear equations based on the optical flow constraint is obtained [18, 19, 22]. With the clustering method, one attempts to detect clusters of the intersection of optical flow constraint equations from different pixels in the velocity space [24]. With the differential method, one can obtain three more equations for each pixel by differentiating the optical flow constraint equation with respect to x, y, and t under the assumption that u and v are constant [13, 27].

Another approach to solve the underdetermination problem is via regularization. The method of Horn and Schunck [16] is perhaps the most popular technique for computing optical flow due to the simplicity of its implementation. Here, one assumes that the overall flow satisfies a certain global smoothness constraint. For example, by using the membrane model, one defines an energy function

$$\mathscr{E} = \iint (E_x u + E_y v + E_t)^2 + \lambda(u_x^2 + u_y^2 + v_x^2 + v_y^2)\, dx\, dy, \tag{2.5}$$

which is minimized for the desired estimate of (u, v). λ is known as the smoothness or the regularization parameter; it determines the degree of smoothness in the computed flow. There are two major difficulties with this method. First, it ignores the presence of occluded regions and very inaccurate flow may be concluded in these regions. Second, the global smoothness constraint tends to smooth the solution across discontinuities, so that there is a leakage of the flow into the stationary background.

An attempt was made in [9] to remedy these problems by setting the smoothness term to 0 in regions near contours of zero-crossings under the assumption that motion discontinuity is a subset of intensity zero-crossings. Another solution based on an oriented smoothness constraint was proposed by Nagel in [20]. A weighting matrix depending on gray value variations was introduced so that the smoothness requirement would be retained essentially only for the normal component of the flow. By enforcing the flow at the gray value corners, the initially proposed weighting matrix contained both first- and second-order derivatives of the intensity process. However, Nagel recommended to use only first-order derivatives in his later work [21] to make the computational algorithm simple and robust.

3. DISCUSSION ON OPTICAL FLOW CONSTRAINT EQUATION

The validity of the optical flow constraint equation was studied in [25]. In this section, we provide more insights into this equation and discuss its limitations.

Given an optical flow field (u, v), we can decompose it into the normal and tangential components denoted by α and β, respectively, along the iso-brightness contours shown in Fig. 1, where the vector (p, q) with $p = E_x$ and $q = E_y$ is the image gradient direction. We can rewrite the optical flow constraint equation, i.e. (2.2), in the product form as

$$(E_x, E_y)(u, v)^T = |(E_x, E_y)|\,|(u, v)| \cos\theta = -E_t, \quad (3.1)$$

where θ is the angle between (E_x, E_y) and (u, v). Thus, based on (3.1), the α component of the flow can be expressed as

$$\alpha = \frac{-E_t}{\sqrt{E_x^2 + E_y^2}}. \quad (3.2)$$

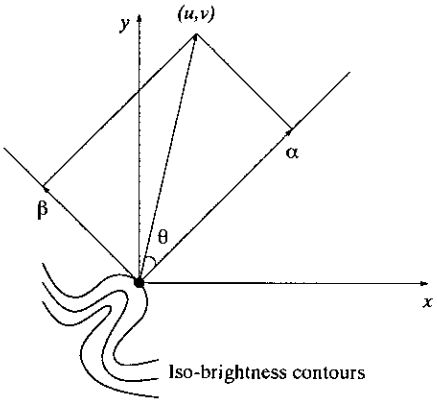

FIG. 1. Normal and tangential components of the flow, where α and β are the normal and tangential components of the flow to iso-brightness contours, respectively.

By using the geometry shown in Fig. 1, we can also express (u, v) in terms of (α, β)

$$\begin{aligned} u &= \frac{E_x}{\sqrt{E_x^2 + E_y^2}}\alpha - \frac{E_y}{\sqrt{E_x^2 + E_y^2}}\beta, \\ v &= \frac{E_y}{\sqrt{E_x^2 + E_y^2}}\alpha + \frac{E_x}{\sqrt{E_x^2 + E_y^2}}\beta. \end{aligned} \quad (3.3)$$

It is clear that the optical flow constraint or, equivalently, (3.2) provides information for the α component, but no information for the β component. Thus, the β flow can only be determined by imposing additional constraint such as the parametric smoothness constraint discussed in Section 4. Even with (3.2), there exist difficulties with the computation of α. In the following, we discuss these problems and ways to circumvent them.

3.1. *Occluded Regions*

To give an example of occluded regions, consider a simple translation of an object in the scene. There are regions in the first frame that are to be covered in the second frame, and new regions in the second frame that were previously covered in the first frame. We denote the union of *occluded regions* in both frames as Ω_{occ}. Even if we knew the location of Ω_{occ}, we still could not estimate α from local information since there is no corresponding region in the other frame. Thus, the value of α computed from (3.2) is invalid in Ω_{occ}. Note that the value of α is not necessarily zero in occluded regions because the occluded region may not be part of the stationary background. One such example is self-occlusion, where parts of the moving object occlude other moving parts of the same object occurring in, say, rotation.

It is possible to determine the occluded regions with the knowledge of exact optical flow (u, v). However, it is usually not easy to determine the exact flow (u, v) if the occluded regions are not known ahead of computation. This dilemma can be resolved to some extent by estimating Ω_{occ} with the definition of optic flow (2.4) and an estimate of (u, v) as described below. Let us denote $E(x, y, t)$ and $E(x, y, t + \delta t)$ as E_1 and E_2, respectively. Without loss of generality, we choose $\delta t = 1$. Given E_1 and an estimate $(\hat{u}, \hat{v})$ of the flow, we can obtain an estimate of E_2 with (2.4) as

$$\hat{E}_2(x, y) = E_1(x - \hat{u}, y - \hat{v}). \quad (3{:}4)$$

The quantity $(\hat{E}_2 - E_2)$ is referred to as the compensation residual and is denoted by $r(x, y)$. By comparing $\hat{E}_2(x, y)$ and $E_2(x, y)$, we have an estimate of the occluded regions as

$$\hat{\Omega}_{\text{occ}} = \{(x, y) : |\hat{E}_2(x, y) - E_2(x, y)|^2 > \tau\}, \quad (3.5)$$

where τ is a threshold value. A good choice for τ seems to be the MSE value of the residual $r(x, y)$. The estimate $\hat{\Omega}_{\text{occ}}$ can be used to improve the estimated optical flow $(\hat{u}, \hat{v})$ as detailed in Section 4.2.

It is worthwhile to mention a fine point with the computation of (3.4). Note that the range of (u, v) is the 2-D vector field of real numbers, while the domain of E is a 2-D integer lattice

$$\Omega = \{(x_i, y_j) \mid i, j = 0, 1, \ldots, N - 1\}.$$

To obtain an estimate of E_2 over Ω, we should look for the corresponding subpixel location in E_1 for each pixel in E_2. This means that the computations in (3.4) requires subpixel interpolation. One commonly used 2-D interpolation scheme is the bilinear interpolation.

3.2. *Flow Field Characterization*

We refer to regions with a small value of $E_x^2 + E_y^2$ as α-indeterminate regions since the value of α cannot be determined from (3.2). We denote the α-indeterminate regions as

$$\Omega_\alpha = \{(x, y) \mid E_x^2 + E_y^2 < \eta\}, \quad (3.6)$$

where, η is a small positive threshold value. The α component of the flow can either be interpolated from the boundaries of such regions or be considered simply to be 0. However, such a decision cannot be entirely based on local information and is not an issue that can be resolved at low-level vision. We suggest interpolation of the flow from the boundaries of these regions, and also marking them as α-indeterminate. The decision to choose between 0 and interpolated values is left to a higher level of vision and is not considered in this paper.

Evidently, the α component of the flow can be uniquely determined in $\overline{\Omega}_\alpha$ (i.e., the compliment of Ω_α) from (3.2). However, it turns out that the β component may also be recovered in such regions by adding appropriate constraints in some special cases. For example, in regions where the local gray value structure is sufficiently characteristic of the underlying motion, such as gray value corners and extremum, it was shown in [21] that the membrane model of Horn and Schunck [16] imposes sufficient constraint to fully determine both components of the flow. Another example was considered by Waxman and Wohn [28]. They showed that it is possible to recover the β component of the flow from at least 12 local measurements of the α components, if the optical flow is caused by the motion of a planar patch. Therefore, it seems reasonable to further divide $\overline{\Omega}_\alpha$ in two subregions: $\alpha\beta$-determinate, in which both components of the flow can be determined, and β-indeterminate, in which only the α component of the flow is recoverable. To summarize, we classify an image into four mutually exclusive characteristic regions.

1. $\alpha\beta$-determinate regions: Both components of the flow may be obtained from the membrane model.
2. β-indeterminate region: Only the α component of flow can be determined.
3. α-indeterminate region: Neither component of the flow can be determined from local information.
4. Occluded regions: No flow information is available.

From the above discussion, we see the limitation of the optical flow constraint equation in determining the optical flow. The importance of the smoothness constraint cannot be overlooked. Most smoothness constraints derived so far do not take the underlying motion into consideration. In Section 4, we derive a smoothness model by using concepts borrowed from fluid dynamics.

3.3. *Spatio-Temporal Filtering*

In the above discussion, we considered the use of the image sequence $E(x, y, t)$ to estimate first-order partial derivatives. However, it is sometimes more advantageous to use a smoothed version of the image sequence for two major reasons. First, noise due to quantization error and temporal aliasing present in an image sequence are magnified by differentiation so that numerical computation of partial derivatives obtained directly from $E(x, y, t)$ often gives a poor approximation of true partial derivatives. The quality of these approximations can be improved by smoothing the intensity $E(x, y, t)$ in both spatial and temporal domains by an isotropic Gaussian filter; i.e.,

$$A(x, y, t) = G(x, y, t) * E(x, y, t), \quad (3.7)$$

where

$$G(x, y, t) = c \exp\left(-\frac{x^2 + y^2}{2\sigma_1^2} - \frac{t^2}{2\sigma_2^2}\right) \quad (3.8)$$

and where σ_1 and σ_2 are the parameters that determine the amount of smoothing in spatial and temporal directions, respectively, and c is the normalization constant. Numerical approximation of partial derivatives of $E(x, y, t)$ can be computed by, say, central differences applied to $A(x, y, t)$. An alternative approach is to use the properties of convolution so that the Gaussian filter is differentiated before it is applied to the image sequence, i.e.,

$$\begin{aligned} A_x &\equiv (G * E)_x = G_x * E, \\ A_y &\equiv (G * E)_y = G_y * E, \\ A_t &\equiv (G * E)_t = G_t * E. \end{aligned} \quad (3.9)$$

The second reason to require spatiotemporal filtering has to do with the validity of the optical flow constraint equation (2.2). It was shown in [25] that the optical flow constraint is valid even across spatial discontinuities for a piecewise continuous intensity process. Temporal discontinuities are more difficult to handle. For example, the optical flow constraint equation is a poor representation of the α component of the flow over moving regions which have a regular texture, even if the exact values of (E_x, E_t, E_y) are known. In [17], it was shown that this phenomena could be quantified in terms of the interframe cross-correlation function and spatiotemporal filtering can be used to reduce its effects. The importance of spatiotemporal smoothing is further discussed in [5].

4. DERIVATION OF A PARAMETRIC SMOOTHNESS MODEL

4.1. *Flow Model Based on Fluid Dynamics*

Given a 2-D vector field $\mu(x, y)$, we can determine its divergence and curl, respectively, by

$$\nabla \cdot \mu = \rho \quad \text{and} \quad \nabla \times \mu = \omega \mathbf{i}_z,$$

where $\mathbf{i}_z = \mathbf{i}_x \times \mathbf{i}_y$ is the unit vector perpendicular to the plane. Using terminology from fluid dynamics, the field μ is called *solenoidal* if $\nabla \cdot \mu = 0$ and *irrotational* if $\nabla \times \mu = 0$. We can decompose a general velocity field μ into the solenoidal and irrotational components, i.e.

$$\mu = \mu_s + \mu_i, \tag{4.1}$$

where μ_s and μ_i satisfy the following constraints:

$$\nabla \cdot \mu_s = 0, \quad \nabla \times \mu_s = \omega \mathbf{i}_z, \tag{4.2}$$

and

$$\nabla \cdot \mu_i = \rho, \quad \nabla \times \mu_i = 0. \tag{4.3}$$

By combining the two constraints [12], we get the cost functional for the general flow case,

$$\mathscr{E}_g = \iint (u_x + v_y - \hat{\rho})^2 \, dx \, dy + \iint (v_x - u_y - \hat{\omega})^2 \, dx \, dy. \tag{4.4}$$

Evidently, the flow computed from (4.4) depends only on the estimates of ρ and ω. It is instructive to look at the Euler equations of this energy function. The overall integral is of the form

$$\iint F(u, v, u_x, v_x, u_y, v_y) \, dx \, dy,$$

with the corresponding Euler equations

$$F_u - \frac{\partial}{\partial x} F_{u_x} - \frac{\partial}{\partial y} F_{u_y} = 0,$$
$$F_v - \frac{\partial}{\partial x} F_{v_x} - \frac{\partial}{\partial y} F_{v_y} = 0.$$

We assume that $(\hat{\rho}, \hat{\omega})$ is not a function of (u, v) or its derivatives. Performing the required differentiations and simplifying, we obtain the following set of Euler equations:

$$\begin{aligned} \nabla^2 u - (\hat{\rho}_x - \hat{\omega}_y) &= 0, \\ \nabla^2 v - (\hat{\rho}_y - \hat{\omega}_x) &= 0. \end{aligned} \tag{4.5}$$

Given a pair of $(\hat{\rho}, \hat{\omega})$ and a set of boundary conditions, we can solve (4.5) for its solution that minimizes the energy function (4.4), denoted by $(\hat{u}, \hat{v})$. Therefore, under fixed boundary conditions, there is a one-to-one relationship between $(\hat{\rho}, \hat{\omega})$ and $(\hat{u}, \hat{v})$.

Next, we incorporate the smoothness model into the optical flow constraint equation with regularization. Consider the energy function

$$\mathscr{E} = \iint (A_x u + A_y v + A_t)^2 \, dx \, dy + \lambda \iint (u_x + v_y - \hat{\rho})^2 + (v_x - u_y - \hat{\omega})^2 \, dx \, dy, \tag{4.6}$$

where the first integrand is the smoothed version of the optical flow constraint equation. The corresponding Euler equations are of the form

$$\begin{aligned} \nabla^2 u - (\hat{\rho}_x - \hat{\omega}_y) &= \frac{1}{\lambda} (A_x u + A_y v + A_t) \, A_x, \\ \nabla^2 v - (\hat{\rho}_y + \hat{\omega}_x) &= \frac{1}{\lambda} (A_x u + A_y v + A_t) \, A_y. \end{aligned} \tag{4.7}$$

Comparing (4.7) to the Euler equations of the membrane model of Horn and Schunck [16], we have the following observation: *If* $(\hat{\rho}, \hat{\omega}) = 0$ *then the system* (*4.7*) *of Euler equations is the same as that derived from the membrane model given in* (*2.5*). This result sheds new light on the interpretation of the membrane model in terms of the divergence and curl of the flow field; i.e. the membrane model is equivalent to assuming a curl-free and divergence-free flow field. This observation also implies that the smoothest flow field is not always the most desired or optimal flow field. With our generalized model (4.6), any prior knowledge of either curl or divergence of the flow can be incorporated through the Euler equations in (4.7).

Furthermore, by comparing (4.5) and (4.7), we see that the optical flow constraint basically provides the right

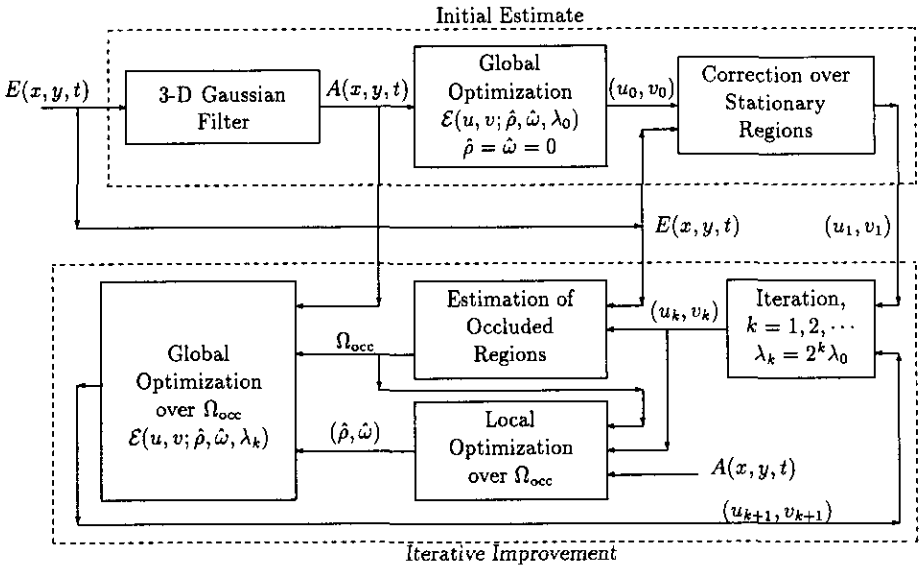
Initial Estimate
$E(x,y,t)$
3-D Gaussian Filter
$A(x,y,t)$
Global Optimization
$\mathcal{E}(u,v;\hat{\rho},\hat{\omega},\lambda_0)$
$\hat{\rho}=\hat{\omega}=0$
(u_0,v_0)
Correction over Stationary Regions
$E(x,y,t)$
(u_1,v_1)
Estimation of Occluded Regions
(u_k,v_k)
Iteration,
$k=1,2,\cdots$
$\lambda_k=2^k\lambda_0$
Global Optimization over Ω_{occ}
$\mathcal{E}(u,v;\hat{\rho},\hat{\omega},\lambda_k)$
Ω_{occ}
$(\hat{\rho},\hat{\omega})$
Local Optimization over Ω_{occ}
$A(x,y,t)$
(u_{k+1},v_{k+1})
Iterative Improvement

FIG. 2. Algorithm for iterative computation of optical flow with discontinuities.

hand side of (4.7). Even if the right-hand-side of (4.7) becomes singular, i.e. $A_xu + A_yv + A_t \approx 0$, $A_x \approx 0$, or $A_y \approx 0$, the solution to (4.7) will not be oversmoothed due to the existence of ρ and ω. Thus, the nature of motion characterized by the functions ρ and ω can play an important role in optical flow computation. Given a sequence of images, it is possible to estimate the ρ and ω by assuming that they are smooth time-varying functions. We focus on the case consisting of only two image frames in this work, and describe a procedure to estimate ρ and ω in the next section.

4.2. *Estimation of Flow Using Two Image Frames*

In this section, we present an iterative algorithm for estimation of the optical flow from the image sequence $E(x, y, t)$. A block diagram of the algorithm is depicted in Fig. 2. The algorithm is based on estimation of curl and divergence, $(\hat{\rho}, \hat{\omega})$, from local intensity values and then using these as parameters in the optimization of energy function in (4.6). We assume the availability of at least two image frames $E_1(x, y)$ and $E_2(x, y)$ at times t and $t + \delta t$, respectively.

In the initialization step, the image sequence $E(x, y, t)$ is first filtered by the 3-D Gaussian filter given in (3.8) to produce the required parameters (A_x, A_y, A_t). Assuming a membrane mode (i.e., $(\hat{\rho}, \hat{\omega}) = 0$), an initial estimate of the flow (u_0, v_0) is computed by minimizing (4.6). Because of the excessive smoothness induced by the membrane model, this estimate can be improved by zeroing the flow over an estimate of stationary regions. Let

$$r'(x, y) = E_2(x, y) - E_1(x, y),$$
$$r_1(x, y) = E_2(x, y) - E_1(x - u_1, y - v_1),$$

and let $g(x, y)$ denote a 2-D Gaussian filter. The improved initial estimate can be written as

$$(u_1, v_1) = \begin{cases} 0 & r'(x, y) * g(x, y) < r_1(x, y) * g(x, y), \\ (u_0, v_0) & \text{otherwise,} \end{cases} \tag{4.8}$$

where $*$ denotes convolution.

In the iterative improvement step, the initial estimate (u_1, v_1) is first used to obtain an estimate of occluded regions from (3.5) as

$$\hat{\Omega}_{occ} = \{(x, y) : |E_2(x, y) - E_1(x - u_1, y - v_1)|^2 > \tau\}. \tag{4.9}$$

Evidently, the initial estimate of the flow is a poor representation of the true flow in $\hat{\Omega}_{occ}$. This can be caused by one of two factors. First, it may be the case that $\hat{\Omega}_{occ}$ is a good representation of the true location of the occluded regions, in which case no satisfactory estimate of the flow can be computed from the information available in the two image frames. Second, it may be the case that $\hat{\Omega}_{occ}$ is estimated incorrectly due to errors in the initial estimate of the flow. The latter case seems to be the dominant factor during the initial iterations of the algorithm. Hence, for each $(x_0, y_0) \in \hat{\Omega}_{occ}$, it is possible to use local information to improve the quality of the flow. This can be accomplished by a local optimization as described below.

Let $W(x_0, y_0)$ denote a window (e.g., 2×2) centered at (x_0, y_0). Using the translational model in (2.4), we perform a local search in $W(x_0, y_0)$ to obtain the optimal local value of the flow, (u_{min}, v_{min}), such that

$$(u_{\min}, v_{\min}) = \{(u_i, v_i) : |E_1(x - u_i, y - v_i) - E_2(x, y)| < |E_1(x - u_j, y - v_j) - E_2(x, y)|, \quad j \in W(x_0, y_0)\}.$$

A new (nonsmooth) estimate may thus be obtained as

$$(u_1', v_1') = \begin{cases} (u_{\min}, v_{\min}) & (x, y) \in \Omega_{\text{occ}} \\ (u_1, v_1) & \text{otherwise.} \end{cases} \tag{4.10}$$

To summarize, we have obtained two estimates of the flow: (u_1, v_1), which is optimally smooth over the moving regions, and (u_1', v_1'), which may exhibit sharp discontinuities over estimates of occluded regions. We wish to combine these two estimates so that a smooth solution is obtained with possible discontinuities along motion boundaries. One way to accomplish this is to formulate a constraint optimization problem using the parametric energy function $\mathscr{E}(u, v; \hat{\rho}, \hat{\omega}, \lambda)$ in (4.6), i.e.,

$$\begin{aligned} &\text{minimize:} && \mathscr{E}(u, v; \hat{\rho}, \hat{\omega}, \lambda) \\ &\text{subject to:} && (u, v) = (u_1, v_1) \quad (x, y) \notin \Omega_{\text{occ}}, \end{aligned} \tag{4.11}$$

where $\hat{\rho}$ and $\hat{\omega}$ are the curl and divergence of $(u_1'\ v_1')$, respectively. The constraint in (4.11) ensures that we do not over-smooth the solution in nonoccluded regions, while the parameters $(\hat{\rho}, \hat{\omega})$ favor a solution with discontinuities along motion boundaries. Numerical solution of the above optimization is discussed in the Appendix and it is shown that the explicit computation of $(\hat{\rho}, \hat{\omega})$ is not necessary.

The new estimate, (u_2, v_2), obtained by solving the constraint optimization in (4.11) may be used as the initial guess for the next iteration of the algorithm. This iterative process may be terminated after a predetermined number of iterations or when there is no substantial improvement in the MSE of the compensation residual. Also, note that the choice of the regularization parameter λ_k, $k = 1, 2, \ldots$, affects only the value of the flow over the estimate of occluded regions. Hence, we increase the value of λ_k at each iteration so that the computation of the flow becomes less dependent on the intensity process $A(x, y, t)$ over occluded regions.

5. EXPERIMENTAL RESULTS

In all our experiments, we assume that only two frames of a test image sequence $E(x, y, t)$ are available as $E_1(x, y)$ and $E_2(x, y)$. Although additional frames can be used to compute a more accurate estimate of partial derivatives according to (3.9), the following results show that the estimate of the flow obtained by the algorithm of the previous section from only two frames is comparable to other computationally more intensive techniques that use as many as 21 frames. Of course, using two frames to obtain an accurate estimate of the flow is not only computationally more efficient than using a large number of frames, but also implementation of the algorithm in real systems with limited video buffer necessitates a small number of frames. We also note that by applying the same technique to every two consecutive frames, we are able to calculate the optical flow of the entire image sequence.

We perform experiments on both synthetic and real image data. For synthetic images where the true flow field is known, the error of the estimated flow field is evaluated using phase error used in [6], i.e.,

$$\text{AAPE} = \text{Average absolute phase error} \tag{5.1}$$

$$\equiv \frac{1}{n^2} \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} \left| \arccos \left(\frac{u_{ij}\hat{u}_{ij} + v_{ij}\hat{v}_{ij} + 1}{((\hat{u}_{ij}^2 + \hat{v}_{ij}^2 + 1)(u_{ij}^2 + v_{ij}^2 + 1))^{1/2}} \right) \right|,$$

where (u_{ij}, v_{ij}) are the true values of the flow and $(\hat{u}_{ij}, \hat{v}_{ij})$ are the estimated values of the flow. It is also useful to look at the compensation error,

$$\text{MSCE} = \text{Mean square compensation error}$$

$$= \frac{1}{n^2} \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} (E_2(x_{ij}, y_{ij}) - E_1(x_{ij} - \hat{u}_{ij}, y_{ij} - \hat{v}_{ij}))^2. \tag{5.2}$$

MSCE is the only type of error that we report for natural images where the true optical flow is unknown. Also, in all of our experiments, we set the value of the threshold τ in (4.9) equal to MSCE obtained by the most recent estimate of the flow.

5.1. *Synthetic Images*

A set of 64×64 pixel images are synthesized with known optical flow fields. The first frame, shown in Fig. 3, is a sphere with radius of 20 pixels on a sinusoidal background. We consider three types of motion.

- Experiment 1: Irrotational flow.

The radius of the sphere in Fig. 3 is increased from 20 pixels to 21 pixels. This corresponds to translational motion perpendicular to the plane of view. The true optic flow (Fig. 4a) has sharp discontinuities on the motion boundaries.

- Experiment 2: Solenoidal flow.

The sphere is rotated 5° along the axis passing through the center of the sphere and perpendicular to the plane of view.

- Experiment 3: General motion.

The motion of the sphere is the combination of the above two types of motions.

In each experiment, we first presmooth the input images by a spatiotemporal Gaussian filter, where 3×3

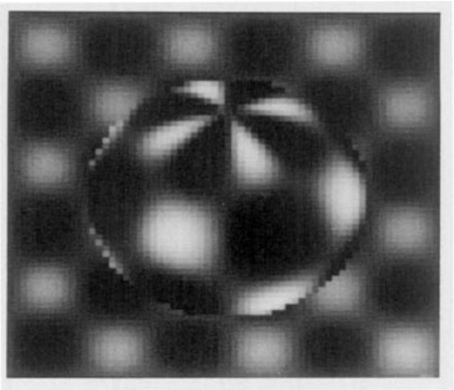

FIG. 3. First frame of a synthetic sphere images. Both the pattern on the sphere and the background are sinusoidal functions.

TABLE 1
Comparison of Results for Experiments 1–3

	AAPE	MSCE	Density
	Experiment 1		
Horn and Schunck	6.01°	21.85	100%
Enhanced Estimate	1.21°	1.05	100%
Estimate over Ω_{nor}	0.97°	0.09	96%
	Experiment 2		
Horn and Schunck	4.80°	9.28	100%
Enhanced Estimate	2.05°	0.82	100%
Estimate over Ω_{nor}	1.50°	0.08	94%
	Experiment 3		
Horn and Schunck	6.92°	28.01	100%
Enhanced Estimate	2.38°	3.10	100%
Estimate over Ω_{nor}	1.72°	0.23	96%

spatial windows are used to maintain a good resolution of discontinuities in the flow field. Based on the preprocessed images, partial derivatives A_x and A_y are computed using 2-point central differences and A_t is computed by first order forward differences, since we assume only two frames are available. The bilinear interpolation is used for getting the gray values at subpixel locations. In all experiments, we use the regularization parameter $\lambda_0 = 250$. The relaxation equations are iteratively updated until the residual drops below 10^{-3}.

Table 1 summarizes the results of the experiments. In this table, the first row of each experiment corresponds to the optical flow obtained by assuming $\hat{\rho} = 0$ and $\hat{\omega} = 0$, which we know to be equivalent to the Horn and Schunck solution. The second row in this table corresponds to the enhanced solution obtained by our algorithm after 10 iterations of updating ρ and ω (further iterations lead to only marginal improvements). The enhanced flow exhibits consistent improvement over the Horn and Schunck solution. The third row of each experiment in the table lists the measures of error of the enhanced estimate considered only over the estimate of non-occluded regions $\overline{\Omega}_{occ}$, as defined in Section 3.1 and shown in Fig. 7. The density values in the last column represent the percentage of the pixels for which the error is computed.

While the two types of error measure defined above are useful for global evaluation of the quality of the optical flow, a more detailed analysis of the flow field requires local estimates of the error. Figures 4a–6a show the computed optical flow. The flow shown in these figures has been obtained by averaging the computed optical flow over 2 × 2 blocks. The true flow is virtually identical to the computed values and hence is not shown. We define the local squared error as

$$\varepsilon^2(i, j) = |u_{ij} - \hat{u}_{ij}|^2 + |v_{ij} - \hat{v}_{ij}|^2. \tag{5.3}$$

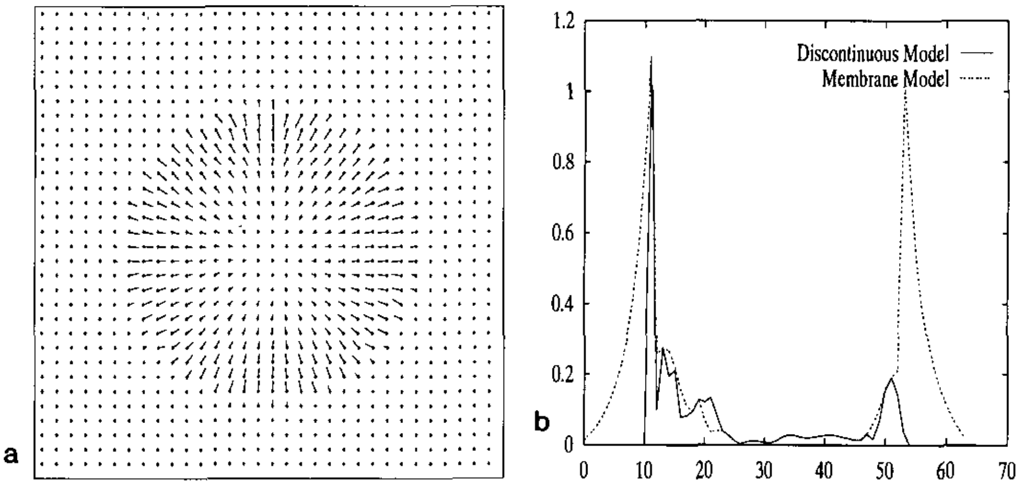

FIG. 4. (a) Computed optical flow for irrotational flow. (b) Cross-section of error between estimated and true optical flow.

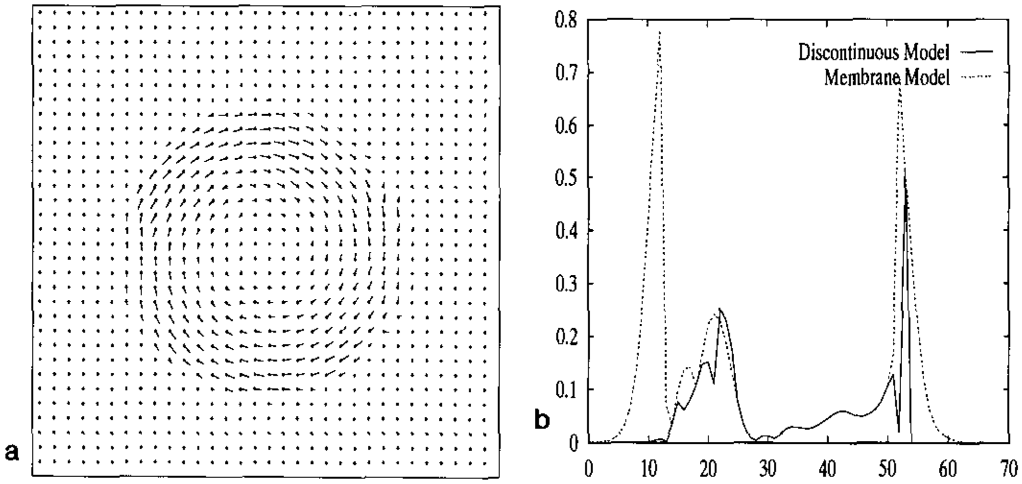

FIG. 5. (a) Computed optical flow for solenoidal flow. (b) Cross-section of error between estimated and true optical flow.

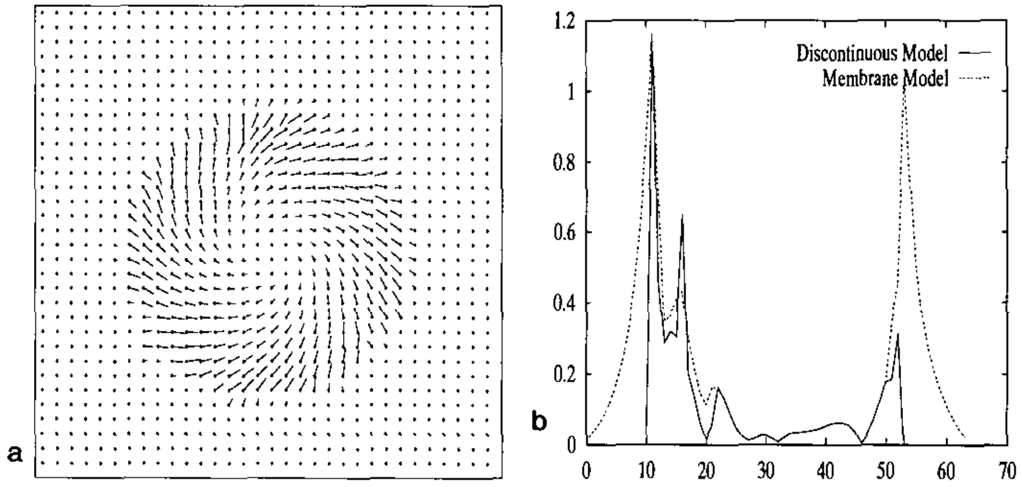

FIG. 6. (a) Computed optical flow for general flow. (b) Cross-section of error between estimated and true optical flow.

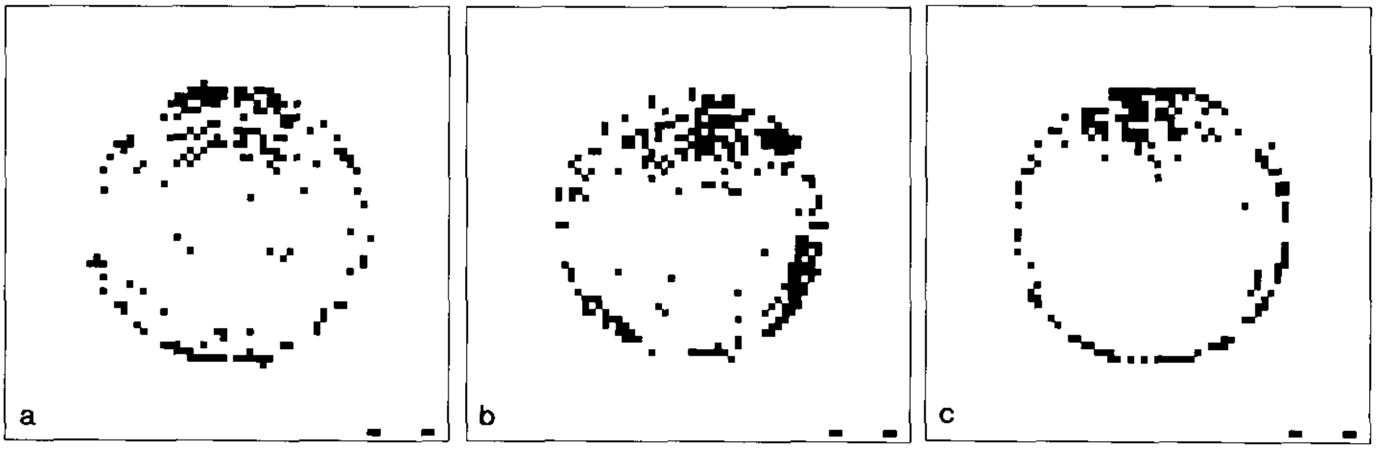

FIG. 7. (a)–(c) Estimate of occluded regions for synthetic images in experiments 1–3, respectively.

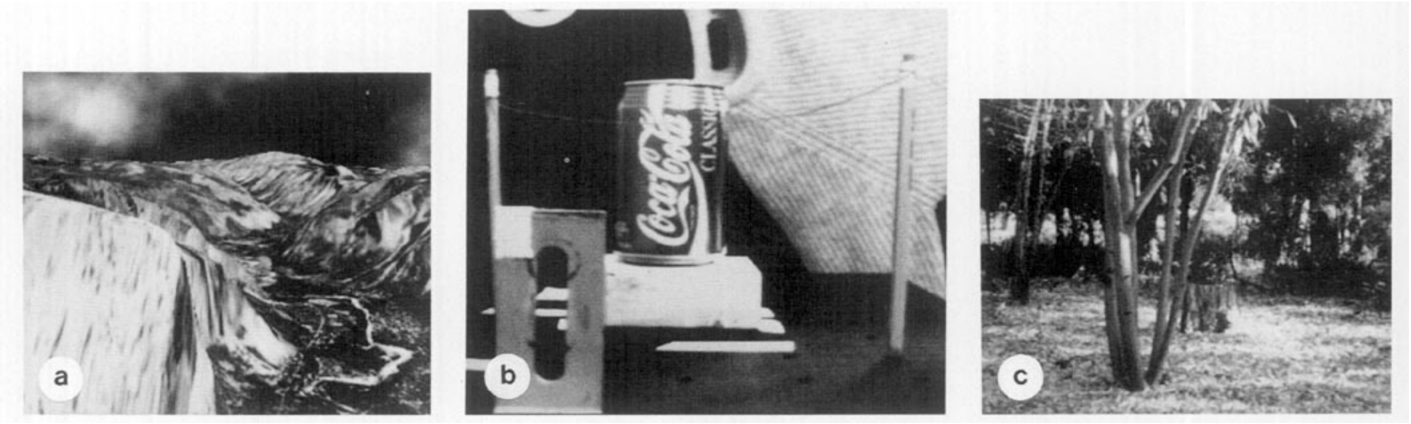

FIG. 8. First frames of real image data. (a) Yosemite sequence. (b) NASA-Ames sequence. (c) SRI trees sequence.

Cross-section plots of ε are shown in Figs. 4b–6b for the Horn and Schunck and the enhanced estimates of the flow field—cross-sections are taken along ε(*i*, *n*/2). In all cases, we observe that maximum error occurs on the boundary of motion. The main reason for this is that the estimate of A_t is unreliable in these regions. However, we note that ε drops off rapidly from its maximum for the enhanced estimate of the flow. Indeed, the main disadvantage of the membrane model is the fact that the solution is smoothly connected across the motion bound-

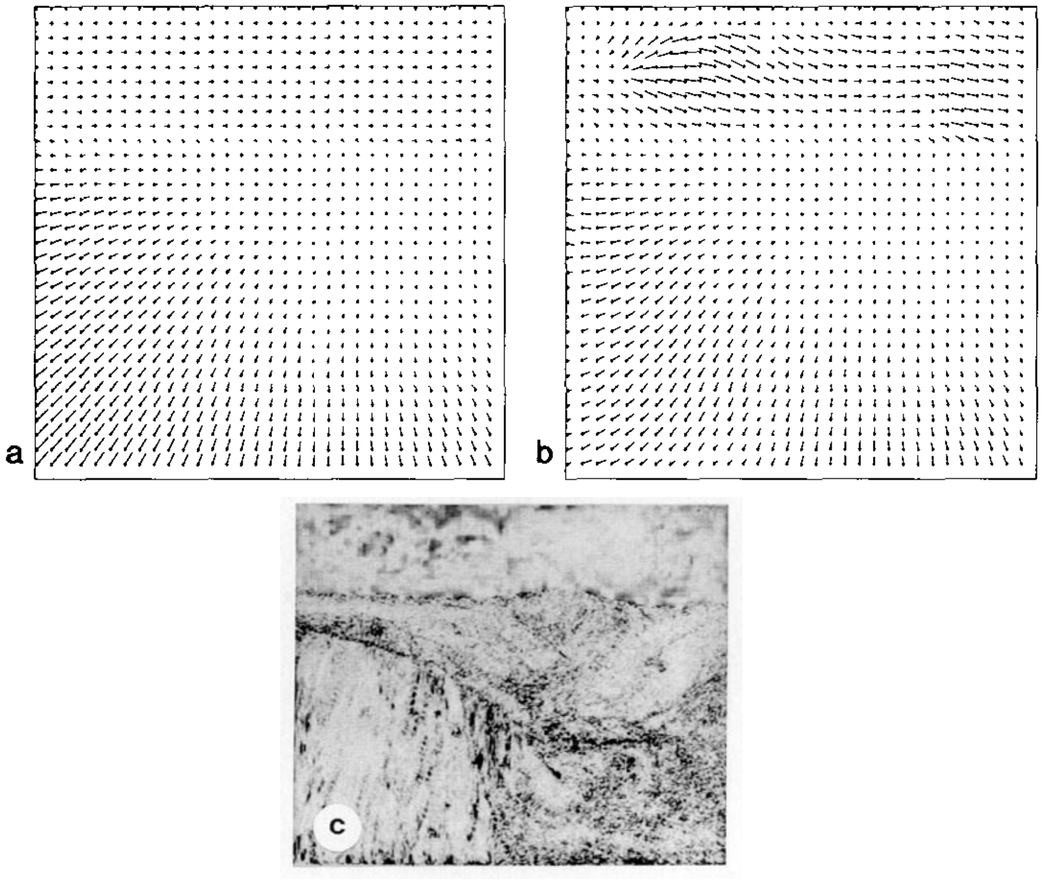

FIG. 9. Result of applying our algorithm to Yosemite sequence. (a) True optical flow. (b) Estimated optical flow. (c) Compensation residual obtained by optical flow in (b).

aries. The cross-section plots clearly show that our algorithm provides an effective method to overcome this problem.

5.2. *Real Images*

We test our algorithm on three sets of real images that were also used in the comparatively study of various techniques for computation of optical flow reported in Barron *et al.* [6]. For comparison of our results to other techniques for computation of optical flow, we refer the reader to that reference. The first frame of each sequence is shown in Fig. 8.

The Yosemite sequence shown in Figure 8a is actually a complex set of synthetic images with known optical flow shown in Figure 9a. Motions are as large as 4 pixels per frame. The frames are clipped to 256 × 256 pixels. Because of the large interframe motion, we first decimate the sequence to obtain 128 × 128 frames and then apply our algorithm to the decimated frames with $\lambda_0 = 250$. Then we linearly interpolate the computed flow to obtain a 256 × 256 estimate of the flow. We use this estimate to compensate $E_1(x, y)$ according to (2.4) and again apply our algorithm to the compensated frames with $\lambda_0 = 1000$. Finally, we obtain the estimate shown in Fig. 9b. Table 2 lists the error between the estimated and true optical flow.

Comparing Figs. 9a and 9b, we note that phase error seems to be large over the regions corresponding to the clouds in the image. This observation is verified by the third row of Table 2 where the phase error over cloud regions is excluded from the total error. The source of this error may be partially explained by considering the compensation residual, $r(x, y)$, as was defined in Section 3.1. Since $r(x, y)$ is the difference between two images with $L = 256$ gray scales, the range of $r(x, y)$ is $(-L, L)$. However, in practice, the histogram of $r(x, y)$ is mostly centered about zero. A useful way to represent $r(x, y)$ is as a negative image $r^{\perp}(x, y)$, where

$$r^{\perp}(x, y) = \max\{L - 10|r(x, y)|, 0\}. \tag{5.4}$$

Figure 9c shows $r^{\perp}(x, y)$ obtained by the estimated optical flow. The darker a region in this figure, the higher the value of residual error. The negative residual image shows that the estimated optical flow over the cloud regions provides a reasonably good compensation.

TABLE 2

Average Phase Error between the True and Estimated Optical Flow Obtained by Our Algorithm for the Yosemite Sequence

	AAPE	MSCE	Density
Estimate over the entire image	14.62°	40.17	100%
Estimate over nonoccluded regions	14.42°	7.13	83%
Estimate over lower $\frac{3}{4}$ of image	10.53°	50.43	75%

TABLE 3

Comparison of Various Algorithms for Computation of Optical Flow of the Yosemite Sequence (Adopted from Barron *et al.* [6])

	AAPE	Density
Horn and Schunck	22.58°	100%
Lucas and Kanade ($\lambda_2 \geq 1.0$)	5.20°	35.1%
Lucas and Kanade ($\lambda_2 \geq 5.0$)	3.55°	8.8%
Uras *et al.* (unthresholded)	16.45°	100%
Uras *et al.* ($\det(H) \geq 1.0$)	5.97°	23.4%
Uras *et al.* ($\det(H) \geq 2.0$)	3.75°	6.1%
Anandan	15.54°	100%
Heeger	11.74°	44.8%
Fleet and Jepson ($\tau = 1.25$)	4.95°	30.6%
Fleet and Jepson ($\tau = 2.5$)	4.29°	34.1%
The proposed method with threshold		
$\tau_\sigma = 0.256$	10.40°	70%
$\tau_\sigma = 0.128$	9.88°	65%
$\tau_\sigma = 0.064$	9.10°	60%
$\tau_\sigma = 0.032$	7.88°	52%
$\tau_\sigma = 0.016$	6.43°	43%
$\tau_\sigma = 0.008$	5.41°	32%
$\tau_\sigma = 0.004$	4.52°	19%
$\tau_\sigma = 0.002$	3.87°	8%
$\tau_\sigma = 0.001$	2.95°	3%

For comparison purposes, we have summarized the result of other techniques for computation of optical flow of the Yosemite sequence in Table 3. We refer the reader to [6] for description of various algorithms in the table. In order to compare our results with others, we should use a thresholding technique to exclude some of the computed flow values. Let $(\hat{u}, \hat{v})$ denote the optical flow estimated as described in the above, and let $\sigma_{\hat{u}}(x, y)$ and $\sigma_{\hat{v}}(x, y)$ denote the variance of $\hat{u}$ and $\hat{v}$ at point (x, y), respectively. To threshold the result, we simply exclude the estimated flow value at (x, y) if $\sigma_{\hat{u}}(x, y) + \sigma_{\hat{v}}(x, y)$ is greater than a threshold τ_σ. Results of this thresholding method are listed in Table 3 for several values of τ_σ. We note that although our computation is based on first order approximation of partial derivatives from two frames, our result is comparable to other techniques that use higher order approximation from many frames.

The NASA-Ames sequence is a nearly irrotational sequence with interframe motion of less than 1 pixel/frame. We use frame 1, shown in Fig. 8b, and frame 3 of the sequence as input to our algorithm. The frames are clipped to 300 × 300 pixels. Fig. 10a shows the computed optical flow with $\lambda_0 = 1000$. The negative compensation residual $r^{\perp}(x, y)$ corresponding to this estimate is shown in Fig. 10b with a MSCE of 9.32 over the entire region of support and 1.57 over the estimate of nonoccluded regions (89.19%).

The SRI trees sequence shows horizontal translation of

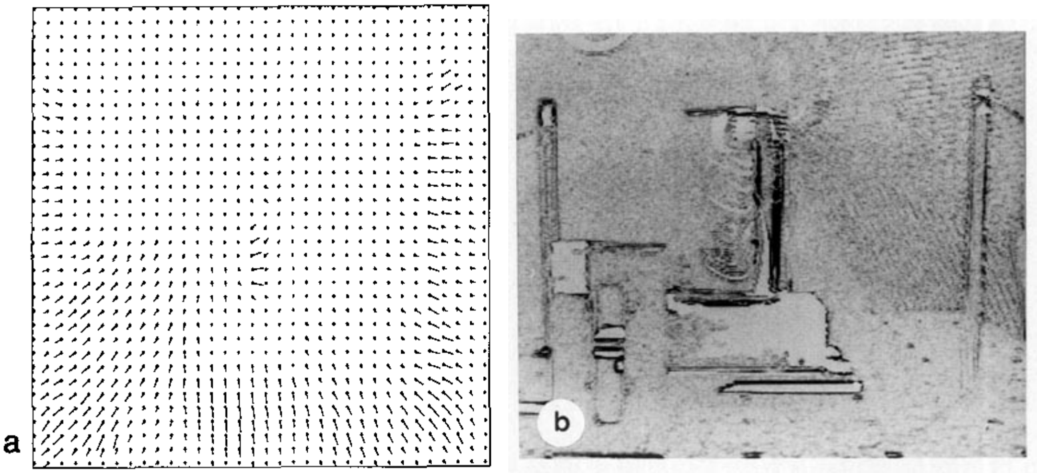

FIG. 10. Result of applying our algorithm to NASA-Ames sequence. (a) Estimated optical flow. (b) Compensation residual obtained by optical flow in (a).

a scene (foreground trees) over a nearly stationary background. The interframe motion is less than 2 pixel/frame. The first frame is shown in Fig. 8c. Fig. 11a shows the computed optical flow with $\lambda_0 = 1000$. The negative compensation residual corresponding to this estimate is shown in Fig. 10b with a MSCE of 47.35 over the entire region of support and 9.85 over the estimate of nonoccluded regions (78%).

6. CONCLUSIONS AND EXTENSIONS

In this paper, we discussed the limitation of the optical flow constraint equation by decomposing the flow into α and β components. Motivated by concepts from fluid dynamics, we derived a parametric smoothness model by decomposing the flow into irrotational and solenoidal fields, and imposing the smoothness constraint on each field separately. This implies smooth translation and rotation of the underlying motion process. We focused on the two frame case in this research and proposed an iterative procedure to improve the estimate of parameters ρ and ω and obtain a more accurate result of the computed optical flow. An estimate of occluded regions, where the motion compensation error is likely above a given threshold, can also be obtained as a byproduct of the algorithm. We showed that significant improvements can be achieved with the proposed algorithm over the classical regularization approach for a set of test problems.

We are also interested in the hierarchical representation of images, where our optical flow algorithm can be

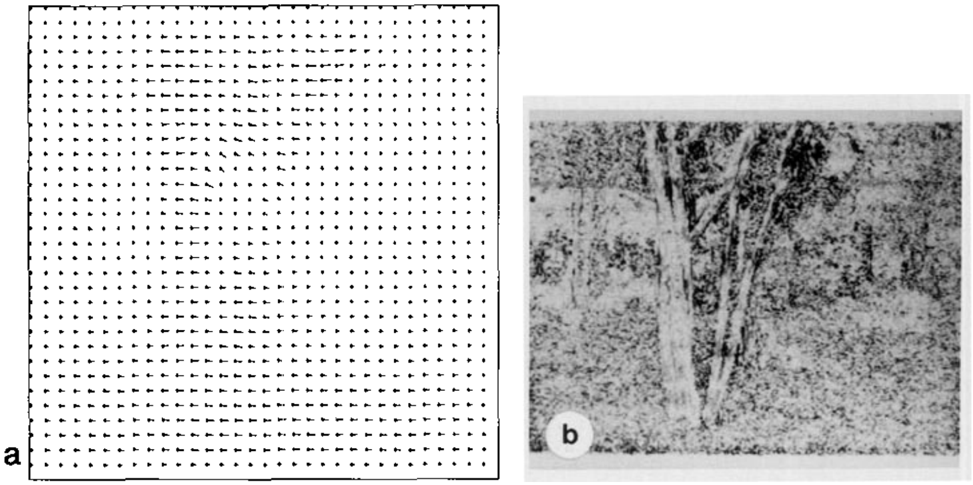

FIG. 11. Result of applying our algorithm to SRI trees sequence. (a) Estimated optical flow. (b) Compensation residual obtained by optical flow in (a).

applied to the low resolution image, so that multiple pixel motion can also be characterized by using the same algorithm—some of our preliminary results have been reported in [12, 11].

APPENDIX: NUMERICAL IMPLEMENTATION

In this appendix, we derive Gauss–Seidel relaxation equations required for the solution of the optimization problem in (4.11).

A.1. Discretization

Let the region of support, Ω, be an $N \times N$ rectangular grid,

$$\Omega = \{(i, j) \mid 1 \leq i \leq N, \quad 1 \leq j \leq N\}.$$

Any function $u(x, y)$ defined over Ω is represented as an N^2 element lexicographically ordered vector of the samples of $u(x, y)$,

$$\mathbf{u} = \{u(i, j) \mid (i, j) \in \Omega\}.$$

Partial derivatives can be approximated by forward differences, backward differences, or central differences:

$$\left(\frac{\partial \mathbf{u}}{\partial x}, \frac{\partial \mathbf{u}}{\partial y}\right)^T_{ij} = \underbrace{\begin{pmatrix} \mathbf{u}_{(i+1)j} - \mathbf{u}_{ij} \\ \mathbf{u}_{i(j+1)} - \mathbf{u}_{ij} \end{pmatrix}}_{\text{forward}}$$

$$= \underbrace{\begin{pmatrix} \mathbf{u}_{ij} - \mathbf{u}_{(i-1)j} \\ \mathbf{u}_{ij} - \mathbf{u}_{i(j-1)} \end{pmatrix}}_{\text{backward}} = \underbrace{\frac{1}{2}\begin{pmatrix} \mathbf{u}_{(i+1)j} - \mathbf{u}_{(i-1)j} \\ \mathbf{u}_{i(j+1)} - \mathbf{u}_{i(j-1)} \end{pmatrix}}_{\text{central}}.$$

Assuming zero boundary conditions, partial derivatives for backward differences can be written as matrix–vector products,

$$\frac{\partial}{\partial x}\mathbf{u} = \mathbf{B}_x\mathbf{u} \quad \text{and} \quad \frac{\partial}{\partial y}\mathbf{u} = \mathbf{B}_y\mathbf{u},$$

where

$$\mathbf{B}_x = \begin{pmatrix} I & & & \\ -I & \ddots & & \\ & \ddots & \ddots & \\ & & -I & I \end{pmatrix} \quad \mathbf{B}_y = \begin{pmatrix} \mathbf{T}_b & & \\ & \ddots & \\ & & \mathbf{T}_b \end{pmatrix}$$

$$\mathbf{T}_b = \begin{pmatrix} 1 & & & \\ -1 & \ddots & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{pmatrix}.$$

Similarly, matrix operators can be defined for backward and central differences; we denote these by $\mathbf{F}_x$, $\mathbf{F}_y$, $\mathbf{C}_x$, and $\mathbf{C}_y$, respectively. Using forward differences, (4.6) can be written as

$$\mathscr{E}_f = \|\mathbf{Pu} + \mathbf{Qv} + \mathbf{t}\|^2 + \lambda\|\mathbf{F}_x\mathbf{u} + \mathbf{F}_y\mathbf{v} - \boldsymbol{\rho}\|^2 + \lambda\|\mathbf{F}_x\mathbf{v} - \mathbf{F}_y\mathbf{u} - \boldsymbol{\omega}\|^2,$$

where $\mathbf{P}$ and $\mathbf{Q}$ are $N^2 \times N^2$ diagonal matrices with their diagonal elements equal to samples of A_x and A_y, respectively, and $\mathbf{t}$ is a vector formed by lexicographical ordering of samples of A_t. A similar expression is also obtained for backward differences, $\mathscr{E}_b$. It turns out that by estimating $\mathscr{E}$ with the average of the forward and the backward differences, $\mathscr{E} = \frac{1}{2}(\mathscr{E}_f + \mathscr{E}_b)$, some cross terms cancel which otherwise would have made the subsequent computations unnecessarily more complex. After some algebraic manipulations, $\mathscr{E}$ can be expressed as

$$\mathscr{E} = \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix}^T \begin{pmatrix} \lambda\mathbf{H} + \mathbf{P}^2 & \mathbf{PQ} \\ \mathbf{PQ} & \lambda\mathbf{H} + \mathbf{Q}^2 \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix} \tag{A.1}$$

$$+ 2\begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix}^T \begin{pmatrix} \mathbf{Pt} + \lambda\mathbf{f} \\ \mathbf{Qt} + \lambda\mathbf{g} \end{pmatrix} + (\mathbf{t}^T\mathbf{t} + \boldsymbol{\rho}^T\boldsymbol{\rho} + \boldsymbol{\omega}^T\boldsymbol{\omega}), \tag{A.2}$$

where

$$\mathbf{H} = \tfrac{1}{2}(\mathbf{F}_x^T\mathbf{F}_x + \mathbf{F}_y^T\mathbf{F}_y + \mathbf{B}_x^T\mathbf{B}_x + \mathbf{B}_y^T\mathbf{B}_y),$$
$$\mathbf{f} = (\mathbf{C}_x\boldsymbol{\rho} - \mathbf{C}_y\boldsymbol{\omega}),$$
$$\mathbf{g} = (\mathbf{C}_y\boldsymbol{\rho} + \mathbf{C}_x\boldsymbol{\omega}).$$

The iterative solution of this optimization problem requires evaluation of matrix–vector products of the form $\mathbf{Hu}$. It is convenient to express these products as a stencil of $\mathbf{H}$ operating on a 2-D array of $u(x_i, y_j)$. The stencil operators of $\mathbf{H}$ for various positions of the 2-D array are shown in Fig. 12.

A.2. Gauss–Seidel Relaxation Method

We divide (u, v) into two groups, (u^+, v^+) and (u^-, v^-), where the variables with a plus sign superscript are con-

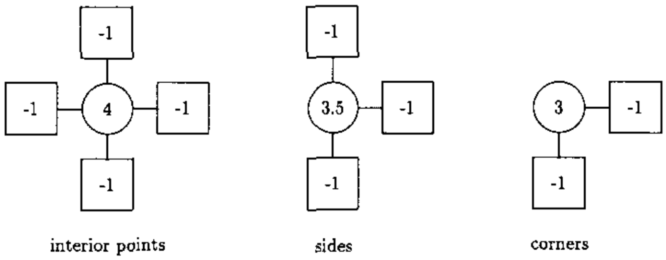

FIG. 12. Stencil operators of matrix **H**.

strained. Also, we define $\mathbf{A}^+$ and $\mathbf{A}^-$ as subsets of the identity matrix such that

$$\mathbf{u} = \mathbf{A}^+\mathbf{u}^+ + \mathbf{A}^-\mathbf{u}^- \quad \text{and} \quad \mathbf{v} = \mathbf{A}^+\mathbf{v}^+ + \mathbf{A}^-\mathbf{v}^-.$$

Substituting the above expressions for (u, v) in (A.1) and equating the partial derivatives with respect to (u^-, v^-), to zero, the following normal equations are obtained:

$$\begin{pmatrix} \mathbf{A}^- & 0 \\ 0 & \mathbf{A}^- \end{pmatrix} \begin{pmatrix} \lambda\mathbf{H} + \mathbf{P}^2 & \mathbf{PQ} \\ \mathbf{PQ} & \lambda\mathbf{H} + \mathbf{Q}^2 \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix} = -\begin{pmatrix} \mathbf{A}^- & 0 \\ 0 & \mathbf{A}^- \end{pmatrix} \begin{pmatrix} \mathbf{Pt} + \lambda\mathbf{f} \\ \mathbf{Qt} + \lambda\mathbf{g} \end{pmatrix}.$$

Using the stencil operator of $\mathbf{H}$, this can be written as

$$\begin{pmatrix} 4\lambda + p_{ij}^2 & p_{ij}q_{ij} \\ p_{ij}q_{ij} & 4\lambda + q_{ij}^2 \end{pmatrix} \begin{pmatrix} u_{ij} \\ v_{ij} \end{pmatrix} = \begin{pmatrix} 4\lambda\bar{u} - p_{ij}t_{ij} \\ 4\lambda\bar{v} - q_{ij}t_{ij} \end{pmatrix}, \qquad (x_i, y_j) \notin \Omega_{\text{occ}},$$

where

$$\bar{u} = \tfrac{1}{4}(u_{i(j+1)} + u_{i(j-1)} + u_{(i+1)j} + u_{(i-1)j} - f_{ij}),$$
$$\bar{v} = \tfrac{1}{4}(v_{i(j+1)} + v_{i(j-1)} + v_{(i+1)j} + v_{(i-1)j} - g_{ij}).$$

Solving for (u_{ij}, v_{ij}), we obtain the following update equations:

$$\begin{aligned} u_{ij} &\leftarrow \bar{u} - \frac{p_{ij}\bar{u} + q_{ij}\bar{v} + t_{ij}}{4\lambda + p_{ij}^2 + q_{ij}^2}\, p_{ij}, \\ v_{ij} &\leftarrow \bar{v} - \frac{p_{ij}\bar{u} + q_{ij}\bar{v} + t_{ij}}{4\lambda + p_{ij}^2 + q_{ij}^2}\, q_{ij}, \end{aligned} \qquad (x_i, y_j) \notin \Omega_{\text{occ}}. \tag{A.3}$$

REFERENCES

1. G. Adive, Determining three-dimensional motion and structure from optical flow generated by several moving objects, *IEEE Trans. Pattern Recognition Mach. Intelligence* **7**, 1985, 384–401.
2. J. Aloimonos and D. Shulman, *Integration of Visual Modules*, Academic Press, San Diego, 1989.
3. P. Anandan, *Measuring Visual Motion from Image Sequences*, COINS Department, Univ. of Massachusetts, Amherst, 1987.
4. S. T. Barnard and W. B. Thombson, Disparity analysis of images, *IEEE Trans. Pattern Recognition Mach. Intelligence* **2**, 1980, 333–340.
5. J. L. Barron, D. J. Fleet, and S. S. Beauchemin, "Performance of Optical Flow Techniques," TR229, Dept. of Computer Science, The University of Western Ontario, July 1993.
6. J. L. Barron, D. J. Fleet, S. S. Beauchemin, and T. A. Burkitt, Performance of optical flow techniques, *in* "IEEE Conference on Computer Vision and Pattern Recognition, 1992," pp. 236–242.
7. A. Blake and A. Zisserman, *Visual Reconstruction*, The MIT Press, Cambridge, MA, 1987.
8. B. F. Buxton and H. Buxton, Computation of optical flow from motion of edge features in image sequences," *Image Vision Comput.* **2**, 1984, 59–74.
9. N. Cornelius and T. Kanade, Adaptive optical flow to measure object motion in reflectance and X-ray image sequence, in *ACM Siggraph/Sigart Interdisciplinary Workshop on Motion, Toronto, 1983*, pp. 50–58.
10. D. Fleet and A. Jepson, Computation of component image velocity from local phase information, *Int. Comput. Vision* **5**, 1990, 77–104.
11. N. Haddadi and C.-C. J. Kuo, Fast computation of motion vectors for MPEG, in *Visual Communication and Image Processing, Cambridge, MA, February 1993.*
12. N. Haddadi and C.-C. J. Kuo, Video image coding via multiresolution computation of motion vectors with discontinuities, in *SPIE Annual Conference on Applications of Digital Image Processing XVI, San Diego, December 1993.*
13. R. Haralick and J. Lee, The facet approach to optical flow, in *Image Understanding Workshop, Arlington, VA*, L. Baumann, Ed.), pp. 84–93, Science Applications, 1983.
14. D. J. Heeger, Optical flow using spatiotemporal filters, *Int. J. Comput. Vision* **1**, No. 4, 1988, 279–302.
15. E. Hildreth, Computing the velocity field along the contours, in *ACM Siggraph/Sigart Interdisciplinary Workshop on Motion: Representation and Perception, Toronto, 1983*, pp. 26–32.
16. B. Horn and B. Schunck, Determining optical flow, *Artificial Intelligence* **17**, 1981, 185–203.
17. D. Kang, K. Jung, and C. Lee, Effects of image smoothing on the reliability of motion estimation, in *Proceedings ISCAS-93, Chicago, Illinois, 1993*, pp. 5–8.
18. J. K. Kearney, W. B. Thompson, and D. L. Boley, Optical flow estimation: An error analysis of gradient-based methods with local optimization, *IEEE Trans. Pattern Recognition Mach. Intelligence*, **9**, 1987, 229–244.
19. B. Lucas and T. Kanade, An iterative image registration technique with an application to stereo vision," in *Proceedings, IJCAI, 1981*, pp. 674–679.
20. H.-H. Nagel, Displacement vectors derived from second-order intensity variations in image sequences, *Comput. Vision Graphics Image Process.* **21**, 1983, 85–117.
21. H. H. Nagel, On the estimation of optical flow: Relations between different approaches and some new results, *Artificial Intelligence* **33**, 1987, 299–325.
22. A. N. Netravali and J. D. Robbins, Motion compensated television coding, I, *Bell System Technical J.*, 1979, 631–670.
23. A. N. Netravali and J. D. Robbins, Motion compensated coding: Some new results, *Bell System Technical J.*, 1980, 1735–1745.
24. B. Schunck, Image flow segmentation and estimation by constant line clustering," *IEEE trans. Pattern Recognition Mach. Intelligence* **11**, 1989, 1010–1027.
25. B. G. Schunck, The image flow constraint equation, *Comput. Vision Graphics Image Process.* **35**, 1986, 20–46.
26. A. Singh, *Optic Flow Computation: A Unified Perspective*, IEEE Comput. Soc. Press, Los Alamitos, CA, 1991.
27. O. Tretiak and L. Pastor, Velocity estimation from image sequences with second order differential operators, in *International Conference on Pattern Recognition, Montreal, 1984*, pp. 16–19.
28. A. M. Waxman and K. Wohn, Contour evolution, neighborhood deformation and global image flow: Planar surface in motion, *Int. J. Robotics* **4**, 1985, 95–108.

29. A. M. Waxman, J. Wu, and F. Bergholm, Convected activation profiles and measurement of visual motion, in *IEEE Conference on Computer Vision and Pattern Recognition, Ann Arbor, MI,* 1988, pp. 717–722.

NAVID HADDADI received the B.S. and M.S. degrees in electrical engineering from the University of Evansville, Indiana, and the University of Colorado at Boulder in 1984 and 1988, respectively. He is currently pursuing the Ph.D. degree in electrical engineering at the University of Southern California.

C.-C. JAY KUO received the M.S. and Ph.D. degrees in electrical engineering at the Massachusetts Institute of Technology in 1985 and 1987, respectively. He is currently Assistant Professor in the Department of Electrical Engineering–Systems at the University of Southern California. He received the National Science Foundation Young Investigator (NYI) Award and the Presidential Faculty Fellow (PFF) Award in 1992 and 1993, respectively.