Toward Real-Time Optical Flow *

David J. Fleet
Department of Computing Science
Queen's University,
Kingston, Ontario, K7L 3N6

Keith Langley
Department of Psychology
University College London,
London, U.K.

Abstract

Working toward efficient (real-time) implementations of optical flow methods, we have applied simple recursive filters to achieve temporal smoothing and differentiation of image intensity, and to compute 2d flow from component velocity constraints using spatiotemporal least-squares minimization. Accuracy in simulation is similar to that obtained in the study by Barron et al. [3], while requiring much less storage of image frames, less computation, and shorter delays.

1. Introduction

Many methods exist for computing optic flow, but few currently run at frame rates on reasonably priced, conventional hardware. The goal of this paper is to outline simplifications to a successful gradient-based approach that may lead to real-time performance with little degradation in accuracy. Our specific concerns include temporal smoothing and differentiation of image intensity, and temporal integration of component velocity constraints to solve for 2d velocity.

More generally, we are working toward efficient implementations of differential and phase-based methods for computing optical flow. We concentrate on these methods because of their simplicity, and because of their performance as compared to other methods[1] [3]. A problem with these methods is their spatiotemporal filtering and differentiation, which involves a significant amount of storage and computation, as well as a temporal delay. Low-pass prefilters are required by differential techniques for reliable numerical differentiation, while band-pass filters are an essential ingredient of energy-based and phase-based approaches [1, 2, 7, 8, 10, 11]. Despite the use of separable filters and reasonably efficient implementations, the filters are a major computational burden of these methods.

We concentrate on gradient-based methods here because of their lower computational demands, which make them good candidates for real-time implementations. This paper examines a class of causal temporal filters that are applied recursively for low-pass filtering and differentiation, with small storage requirements and short delays. These low-pass filters are also readily generalized to band-pass filters, and are therefore applicable to phase-based methods [16]. We also examine a simple form of recursion in solving for optical flow from normal velocity constraints.

2. Local Gradient-Based Method

Gradient-based techniques compute velocity from first-order derivatives of image intensity, or filtered versions of the image (using low-pass or band-pass filters);[2]

$$\nabla I(\mathbf{x}, t) \cdot \mathbf{v} + I_t(\mathbf{x}, t) = 0 , \qquad (1)$$

where $\nabla I(\mathbf{x}, t) = (I_x(\mathbf{x}, t), I_y(\mathbf{x}, t))^T$ denotes the spatial gradient of $I(\mathbf{x}, t)$, $I_t(\mathbf{x}, t)$ denotes the partial temporal derivative of $I(\mathbf{x}, t)$, and $\nabla I(\mathbf{x}, t) \cdot \mathbf{v}$ denotes the usual dot product. The space-time gradient effectively measures the instantaneous velocity of level intensity contours, which can be derived from a conservation assumption (e.g. $dI(\mathbf{x}, t)/dt = 0$). This measures the speed s in the direction of the image gradient $\mathbf{n}(x, t)$, given by $s(\mathbf{x}, t) = -I_t(\mathbf{x}, t)/\| \nabla I(\mathbf{x}, t) \|$, and $\mathbf{n}(\mathbf{x}, t) = \nabla I(\mathbf{x}, t)/\| \nabla I(\mathbf{x}, t) \|$.

But there are two unknown components of $\mathbf{v}$ in (1) constrained by only one equation, and therefore further constraints are necessary to solve for both. Three possibilities are well-known: *(i)* to use higher-order

*We are very grateful to Paul Hodgins for helping to implement and test the IIR filters, and to Allan Jepson for comments on a draft of this paper. We are also grateful to NSERC Canada, and ITRC (Ontario Centre of Excellence) for their financial support.

[1] Based on image sequences containing small amounts of aliasing

[2] These methods assume $I(\mathbf{x}, t)$ is differentiable, and hence aliasing should be avoided. If aliasing cannot be avoided in image acquisition, then one could apply the method in a coarse-fine manner where estimates are first produced at coarse scales where aliasing is assumed to be less severe, with velocities less than 1 pixel/frame. This issue is not addressed here in detail.

derivatives with additional conservation assumptions (e.g. [9, 21]); *(ii)* to impose global smoothness constraints (regularization) on the velocity field (e.g. [12, 18]); and *(iii)* to impose a local parametric model (e.g. constant or linear variation) on the velocity field locally (e.g. [1, 8, 15, 17, 19]).

The latter approach is adopted here, based on its performance in [3]: we compute velocity using a weighted least-squares fit of local first-order constraints (1) to a constant model for $\mathbf{v}$ in each small spatiotemporal neighbourhood Ω by minimizing

$$\sum_{(\mathbf{x},t)\in\Omega} W(\mathbf{x},t)\,[\nabla I(\mathbf{x},t)\cdot\mathbf{v} + I_t(\mathbf{x},t)]^2 . \quad (2)$$

Here, $W(\mathbf{x}, t)$ is a window that gives more influence to constraints near the centre of the neighbourhood. The constant model for $\mathbf{v}$, and the spatiotemporal support of the minimization $W(\mathbf{x}, t)$, reflect assumptions of spatial and temporal coherence (or smoothness) in the local behaviour of $\mathbf{v}$ (e.g. [4, 19, 20]).

The minimization in (2) leads to the linear system $WA\mathbf{v} = W\mathbf{b}$, the solution of which is given by

$$\mathbf{v} = [A^T W A]^{-1} A^T W \mathbf{b} , \quad (3)$$

where, for n points $(\mathbf{x}_i, t_i) \in \Omega$,

$$\begin{aligned} A &= [\nabla I(\mathbf{x}_1, t_1), ..., \nabla I(\mathbf{x}_n, t_n)]^T , \\ W &= \mathrm{diag}[W(\mathbf{x}_1, t_1), ..., W(\mathbf{x}_n, t_n)] , \\ \mathbf{b} &= -(I_t(\mathbf{x}_1, t_1), ..., I_t(\mathbf{x}_n, t_n))^T . \end{aligned}$$

When $A^T W A$ is nonsingular we may solve for $\mathbf{v}$, which is easily done because $A^T W A$ is a 2 by 2 matrix:

$$A^T W A = \begin{bmatrix} \sum W I_x^2 & \sum W I_x I_y \\ \sum W I_y I_x & \sum W I_y^2 \end{bmatrix} , \quad (4a)$$

$$A^T W \mathbf{b} = \begin{bmatrix} \sum W I_x I_t \\ \sum W I_y I_t \end{bmatrix} \quad (4b)$$

where all sums are taken over points in the spatiotemporal neighbourhood Ω.

Finally, if we treat $[A^T W A]^{-1}$ as the covariance matrix for $\mathbf{v}$, then unreliable estimates may be identified using its inverse eigenvalues (as confidence measures). That is, accuracy tends to improve with an increase in the eigenvalues of $A^T W A$, λ_1 and λ_2, which reflect the magnitudes and the range of orientations of the spatial gradients. Possible confidence measures include the trace of $A^T W A$, $\lambda_1 + \lambda_2$ [19], or the magnitude of its smallest eigenvalue λ_2 [3]. Such conditions are important to the success of this technique.

As an example of the method, an implementation described in [3] used FIR spatiotemporal filtering with Gaussian support and a standard deviation of 1.5 pixels-frames, followed by 4-pt central differences for numerical differentiation. Temporal support for the entire process was 15 frames, and therefore a delay of 7 frames was necessary. The minimization in (2) involved a purely spatial domain Ω at each frame to avoid further delays and computational expense.

3. Causal Recursive Temporal Filters

We now consider the use of recursive filters to alleviate the computation and storage requirements of FIR filters. First, let the initial spatiotemporal filter be separable in space-time, so that the response can be computed as $R(\mathbf{x}, t) = E(t) * [B(\mathbf{x}) * I(\mathbf{x}, t)]$, and we may examine its spatial and temporal components independently. The class of temporal filters we consider are derived from the truncated exponential [5, 6]:

$$E(t) \equiv H(t)\,\tau e^{-\tau t} = \begin{cases} \tau \exp[-\tau t] & , t \geq 0 \\ 0 & , t < 0 \end{cases} \quad (5)$$

where $H(t)$ is the Heaviside step function. Equation (5) is the impulse response of a low-pass causal filter, where τ^{-1} determines the duration of temporal support and the amount of smoothing; τ^{-1} is called the time constant. The Fourier transform of $E(t)$ is $\hat{E}(\omega) = \tau/(\tau + i\omega)$, with amplitude and phase spectra

$$|\hat{E}(\omega)| = \frac{\tau}{\sqrt{\tau^2 + \omega^2}} , \quad \arg[\hat{E}(\omega)] = \tan^{-1}\left(\frac{\omega}{\tau}\right) . \quad (6)$$

The amplitude spectrum is low-pass, symmetric about the origin where it takes a value of 1, and very broad (decaying like $1/\omega$ at high frequencies). The phase spectrum is nearly linear, especially at the low end of the spectrum; linear phase (a distortionless filter) is difficult to achieve with IIR causal filters.

One concern with $E(t)$ is the slow decay of its amplitude spectrum, making it susceptible to noise and temporal aliasing. To alleviate this we may cascade $E(t)$ repeatedly to achieve some useful properties. For example, following [5], it can be shown that repeated convolution of $E(t)$ is equivalent to convolution with a single kernel that has a particularly simple form:

$$E_n(t) \equiv [E(t)]^{*n} = \frac{(t\,\tau)^{(n-1)}}{(n-1)!} E(t) . \quad (7)$$

From the Central Limit Theorem it also follows that this cascaded convolution tends to a Gaussian function [5]. The mode (peak) of the impulse response $E_n(t)$ is straightforwardly shown to be $(n-1)/\tau$. The corresponding mean (centre of mass) and standard deviation are given by n/τ and $\sqrt{n}/\tau$. When using (7)

it is therefore important to note that with increasing numbers of cascades there is an implicit time delay in response, which we take to be the mode $(n-1)/\tau$.

As suggested above, as the number of cascades increases, the filter becomes more Gaussian-like, more low-pass, and its phase spectrum remains nearly linear. Its amplitude and phase spectra are given by

$$|\hat{E}_n(\omega)| = \frac{\tau^n}{\sqrt{(\tau^2+\omega^2)^n}}\ , \quad \arg[\hat{E}_n(\omega)] = n\tan^{-1}\left(\frac{\omega}{\tau}\right)$$

The amplitude spectra decays to half height, i.e. $|\hat{E}_n(\omega)| = 0.5$, when $\omega = \tau\sqrt{2^{2/n}-1}$.

4. Temporal Differentiation

The gradient-based method requires low-pass smoothing followed by differentiation. As discussed above, because of the assumed separability of the filters, we can treat the temporal and spatial components of the filter independently, in which case

$$\frac{\partial R_n(\mathbf{x},t)}{\partial t} = \frac{dE_n(t)}{dt} * [B(\mathbf{x}) * I(\mathbf{x},t)]\ , \qquad (8)$$

where $R_n(\mathbf{x},t) = E_n(t) * [B(\mathbf{x}) * I(\mathbf{x},t)]$. However, it is expensive to apply a different temporal filter for differentiation in cascade, or in parallel, with the low-pass filter.

With this in mind it is interesting to note that the temporal derivative can be computed using the same class of exponential filters discussed above. In particular, observe that the temporal derivative of $E_n(t)$ in (7) has the form

$$\frac{dE_n(t)}{dt} = \delta(t)\,E_n(t) + \tau E_{n-1}(t) - \tau E_n(t)\ , \qquad (9)$$

which uses the fact that the derivative of $H(t)$ is a Dirac delta function. Moreover, if we assume that $n > 2$, for which $E_n(0) = 0$, then the derivative becomes

$$\frac{dE_n(t)}{dt} = \tau\,[E_{n-1}(t) - E_n(t)]\ , \quad \text{for } n > 2\ . \qquad (10)$$

That is, the derivative is a weighted difference of $E_{n-1}(t)$ and $E_n(t)$. If the low-pass response is computed as a cascade of $E_{n-1}(t)$ and $E_1(t)$ then all the information necessary to compute the derivative is already available in the computation of the low-pass filter.

5. Digital Filter Design

In order to realize discrete IIR implementations of (7) and (10), we use a bilinear transformation to map the Laplace transform (the s-domain) of the continuous filter onto the z-domain [13]: i.e.,

$$H(z) = H_c(s)\big|_{s=\frac{2}{T}\left(\frac{1-z^{-1}}{1+z^{-1}}\right)} \qquad (11)$$

Although the impulse invariant transformation is somewhat simpler, it is susceptible to severe aliasing problems, which in this case distort the differentiation when computed according to (10).

The Laplace transform of the truncated exponential (5) is simply $\mathcal{L}[E(t)] = \tau/(s+\tau)$. Using the fact that convolution in the time domain is equivalent to multiplication in the s-domain, the Laplace transform of $E_n(t)$ is easily shown to be

$$\mathcal{L}[E_n(t)] = \left[\frac{\tau}{s+\tau}\right]^n\ . \qquad (12)$$

Rewriting this s-domain representation of the low-pass filter as a cascade of an $(n-1)^{th}$-order filter and a first-order filter (required for (10)), we apply the bilinear transformation to obtain the following z-transform

$$H_n(z) = \left(\frac{\tau}{s+\tau}\right)^{(n-1)}\left(\frac{\tau}{s+\tau}\right)\Bigg|_{s=2\left(\frac{1-z^{-1}}{1+z^{-1}}\right)}\ . \qquad (13)$$

For example, $H_n(z)$ for $n = 3$ and $n = 4$ is given by

$$H_3(z) = q^3\left(\frac{1+2z^{-1}+z^{-2}}{1+2rz^{-1}+r^2z^{-2}}\right)\left(\frac{1+z^{-1}}{1+rz^{-1}}\right)$$

$$H_4(z) = q^4\left(\frac{1+3z^{-1}+3z^{-2}+z^{-3}}{1+3rz^{-1}+3r^2z^{-2}+r^3z^{-3}}\right)\left(\frac{1+z^{-1}}{1+rz^{-1}}\right)$$

where $q = \tau/(\tau+2)$ and $r = (\tau-2)/(\tau+2)$.

The discrete implementations then follow directly from the z-transform, using a direct-form-II structure [13]. Figure 1 shows the implementation of the low-pass filter combined with the differentiation in the case of $n = 3$. The equivalent difference equations (with t treated as a discrete variable) for the first stage of the cascade are given by:

$$\begin{aligned} w(t) &= I(t) + a_1\,w(t-1) + a_2\,w(t-2) \\ R_2(t) &= b_0\,w(t) + b_1\,w(t-1) + b_2\,w(t-2) \end{aligned} \qquad (14)$$

while the second stage is given by

$$\begin{aligned} y(t) &= R_2(t) + c_1\,y(t-1) \\ R_3(t) &= d_0\,y(t) + d_1\,y(t-1)\ , \end{aligned} \qquad (15)$$

and the temporal derivative is given by

$$\frac{dR_3(t)}{dt} = \tau\,[R_2(t) - R_3(t)]\ . \qquad (16)$$

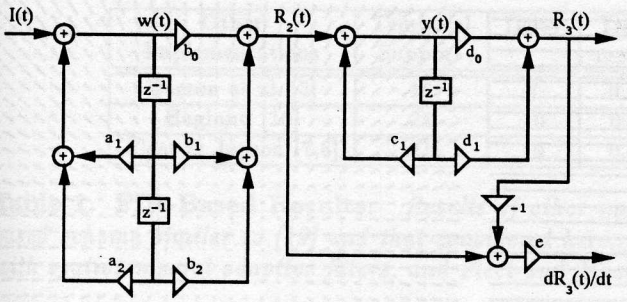


Figure 1. Direct-Form-II Structure: *Shows delays, adders and multipliers used to implement the cascaded low-pass filter $E_3(t)$ and its derivative $dE_3(t)/dt$. The coefficients are: $a_1 = -2r$, $a_2 = -r^2$, $b_0 = b_2 = q^2$, $b_1 = 2q^2$, $c_1 = -r$, $d_0 = d_1 = q$, and $e = \tau$.*

6. Estimation of Optical Flow

We now consider the weighted least-squares estimation of 2d image velocity from the normal constraints (2). For convenience, we assume separability of the window $W(\mathbf{x}, t)$ in (2) that determines the spatiotemporal extent of the minimization so that the spatial and temporal components can be examined separately.

Ignoring the temporal component of the window, like the implementations described by others [3, 19], we first note that the spatial collection of squared constraints (the components of (4) at each pixel) can be expressed as convolution. In other words, given the partial derivatives of the low-pass filtered image $R(\mathbf{x}, t)$, we first form the five images

$$R_x^2(\mathbf{x}, t), \quad R_y^2(\mathbf{x}, t), \quad R_x(\mathbf{x}, t)R_y(\mathbf{x}, t), \quad (17)$$
$$R_x(\mathbf{x}, t)R_t(\mathbf{x}, t), \quad R_y(\mathbf{x}, t)R_t(\mathbf{x}, t).$$

The entries in the linear system in (4), at each pixel, are then given by the convolution of these intermediate images with spatial component of the window $G(\mathbf{x})$, which we take to be a Gaussian. This observation can be significant for SIMD or pipeline architectures.

The use of temporal support is somewhat more problematic because of the computational expense and storage needed to accumulate constraints and solve the minimization problem from normal constraints at each time. To alleviate this problem, yet exploit some degree of temporal coherence, it is natural to consider *incremental* methods. For example, Singh [20] uses recursive estimation based on the Kalman filter, while Black and Anandan [4] describe an incremental method based on a MRF framework. Although simplifications are made to the optimization procedures in each of these cases, they still involve considerable computational expense and storage.[3] There also remain questions about the convergence of such techniques, and the choice of arbitrary parameters such as process noise in the Kalman framework.

The approach taken here returns to the original minimization but with an implicit rather than an explicit temporal window. As mentioned above, the accumulation of constraints through space and time can be viewed as low-pass filtering of the intermediate images in (17). In space we apply an FIR Gaussian filter. In time we use a causal first-order IIR filter with exponential support (5), the implementation of which requires no more than a first-order temporal difference. More precisely, let $\bar{A}(\mathbf{x}, t)$ and $\bar{\mathbf{b}}(\mathbf{x}, t)$ represent the linear system of normal equations formed by convolution of the intermediate images (17) at time t with the Gaussian weights $G(\mathbf{x})$. Then, the temporal accumulation of constraints within an exponential window of support (5), with time constant τ_2^{-1}, may be computed as

$$A(\mathbf{x}, t) = \alpha A(\mathbf{x}, t-1) + (1-\alpha)\bar{A}(\mathbf{x}, t), \quad (18a)$$

$$\mathbf{b}(\mathbf{x}, t) = \alpha \mathbf{b}(\mathbf{x}, t-1) + (1-\alpha)\bar{\mathbf{b}}(\mathbf{x}, t), \quad (18b)$$

where $\alpha = \exp(-\tau_2)$ [6]. The solution to the resultant set of normal equations is then given by

$$\mathbf{v} = [A(\mathbf{x}, t)]^{-1}\mathbf{b}(\mathbf{x}, t). \quad (19)$$

Observe that we may set $\alpha = 0$ in (18) and thereby remove any temporal coherence at this stage of computation.

This recursion directly involves the normal equations rather than the final velocity estimates. Compared to a Kalman framework, there is no need for the explicit propagation of covariance matrices, nor for the introduction of process noise, which is often difficult to do effectively without sophisticated knowledge of the underlying system. From one perspective, the exponentially decaying temporal window might be regarded as a form of the process noise, where the relative amounts of measurement and process noise remain fixed. Although this is perhaps the simplest form of temporal coherence, it comes at a very small cost.

7. Experimental Results

The implementation described above can be broken into two main stages of processing.

[3] It should be added that they also attempt to solve for occlusion boundaries as well as optical flow and therefore address a somewhat broader problem than that addressed here.

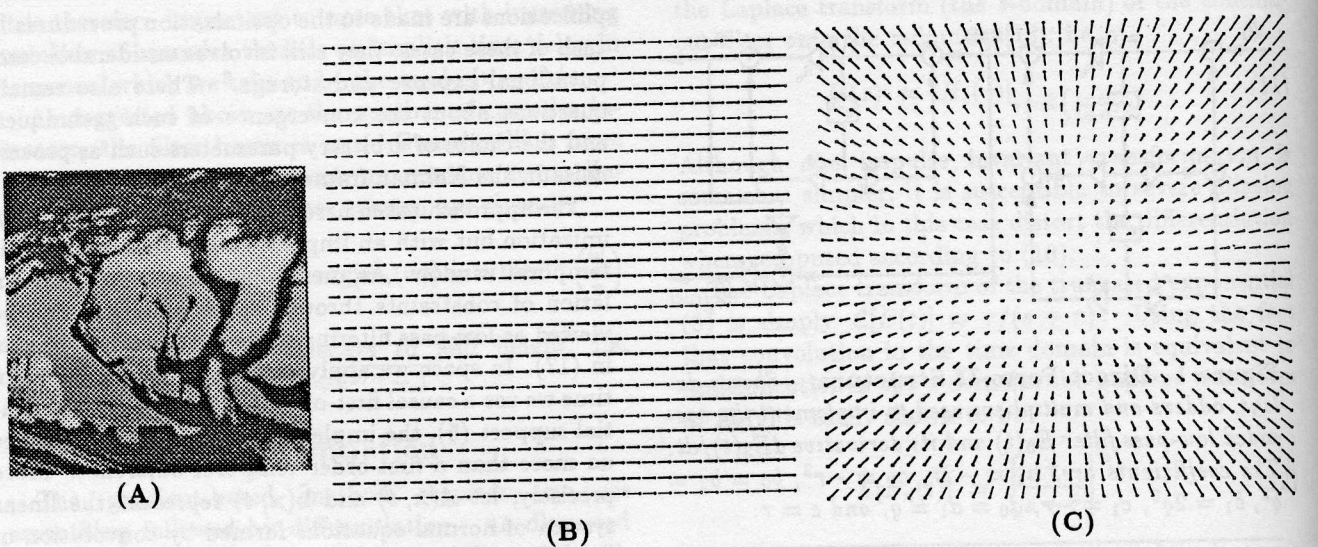


Figure 2. Image Texture and Flow Fields: **(A)** *Shows the texture on the moving surface at a single frame.* **(B)** *2D motion field for the translating tree sequence, where the camera moves along its x-axis normal to its line of sight with image velocities between* $(1.73, 0.0)$ *and* $(2.3, 0.0)$ *pixels/frame.* **(C)** *2D motion field for the diverging tree sequence, where the the camera moves along its line of sight. The focus of expansion is at the centre of the image, and image speeds vary from 1.4 pixels/frame on one side to 2.0 pixels/frame on the other.*

1. Spatiotemporal low-pass smoothing and differentiation of $I(\mathbf{x}, t)$, accomplished in three stages:
   (a) a Gaussian low-pass spatial filter $G(\mathbf{x})$ with a standard deviation σ_1;
   (b) n^{th}-order IIR temporal filters $E_n(t)$ and $dE_n(t)/dt$ with time constant τ_1 (14)–(16), yielding the low-pass output $R(\mathbf{x}, t)$ its temporal derivative $R_t(\mathbf{x}, t)$;
   (c) computation of $R_x(\mathbf{x}, t)$ and $R_y(\mathbf{x}, t)$ from $R(\mathbf{x}, t)$ using 4-pt central differences (convolutions with 1d mask $\frac{1}{12}(-1, 8, 0, -8, 1)$).
2. Computation of 2d velocity from the normal constraints in three stages:
   (a) computation of intermediate images (17);
   (b) separable low-pass smoothing of intermediate images by a spatial Gaussian $G(\mathbf{x})$ with a standard deviation σ_2, and a first-order IIR filter $E_1(t)$ with time constant τ_2 (18);
   (c) solution of the normal equations (19).

Following the computation of $\mathbf{v}(\mathbf{x}, t)$ it is also possible to derive confidence measures based on eigenvalues of $A(\mathbf{x}, t)$. In terms of computational efficiency, the simplest reasonable measure is the trace of $A(\mathbf{x}, t)$ in (18); i.e., the sum of its eigenvalues $\lambda_1 + \lambda_2$. However, in our current implementation we use only the smallest eigenvalue λ_2 [3], with a simple threshold $\lambda_2 < 1$ to detect situations in which the aperture problem prevails or the gradient magnitudes are too small. It is important to note that we do this mainly for convenience in comparing our results to those of others. In general, we acknowledge that confidence measures should be maintained with the velocity estimates, using a conservative threshold only to remove clear outliers. Other questions with the use of eigenvalues concerns their dependence on the image contrast (e.g. a lower threshold may be required on low-contrast input), and hence the relationship between the eigenvalues and *confidence measures* is unclear (however see [19]).

In summary, the entire method (including the confidence threshold) can be viewed as a sequence of image operations, suitable for special SIMD or pipeline hardware. It also lends itself to fast implementation on serial machines. The main parameters of the method include those of the initial filtering (σ_1, τ_1 and n), and those that determine the domain of the least-squares minimization (σ_2 and τ_2). The delay in velocity estimation is due only to the first stage of spatiotemporal filtering, and depends on the order of the IIR filter and the time-constant as discussed above. Although our

Other Implementations	Temporal Support	Delay	Trans. Tree Error Stats. mean, st. dev. & density			Div. Tree Error Stats. mean, st. dev. & density		
Barron et al. [3]	15	7	0.66°	0.67°	39.5%	1.94°	2.06°	48.2%
Haglund [10]	21	10	0.85°	0.42°	21%	1.77°	1.14°	17%
Fleet & Jepson [7,8]	21	10	0.23°	0.19°	49.7%	0.8°	0.73°	46.5%

Table 1. FIR-Based Results: *Results of other implementations on the same data. Barron's is a gradient-based scheme similar to [19] and that considered here, while Haglund's results are for an energy-based approach with spatiotemporal adaptive filters, and Fleet and Jepson's results are for a phase-based technique.*

implementation here uses only a single spatial scale, the method generalizes easily to multiple spatial scales for which we expect improved results.

7.1 Error Measure

Following [3, 8] we view velocity as orientation in space-time, and therefore use an angular measure of error. That is, let velocity $\mathbf{v} = (v_1, v_2)^T$ be represented as a 3-d unit direction vector, $\vec{\mathbf{v}} \equiv \frac{1}{\sqrt{v_1^2+v_2^2+1}}(v_1, v_2, 1)^T$. The error between the correct velocity $\vec{\mathbf{v}}_c$ and an estimate $\vec{\mathbf{v}}_e$ is then given by

$$\psi_E = \arccos(\vec{\mathbf{v}}_c \cdot \vec{\mathbf{v}}_e) \; . \qquad (20)$$

This error measure is useful because it handles large and very small speeds without the amplification inherent in a relative measure of vector differences. An error of two degrees when the actual speed is 1 pixel/frame corresponds to a relative error of about 6% [8].

7.2 Results

Our quantitative results are derived from two synthetic image sequences with smooth motion fields for which the true 2d motion fields are known. They are convenient since they have been used in the extensive comparative study of Barron et al. [3] and by others (e.g., [10]). Each sequence depicts a textured plane moving with respect to a camera. The textured plane and the known 2d motion fields are shown in Figure 2.

Table 1 shows results reported by others on the same sequences. The gradient-based implementation by Barron et al. is described at the end Section 2 above. It uses spatiotemporal Gaussian low-pass filtering, and a spatial domain for the minimization (2) with Gaussian weights of standard deviation of 1.1 pixels. Our implementations keep these parameters the same where appropriate.

The energy-based technique of Haglund [2, 10] produced accurate results, but the density of measurements was lower than the other methods shown. This technique also had a longer duration of temporal support, and hence a longer delay.[4] The phase-based technique [7, 8] produced more accurate results, with a higher density of measurements on most sequences. It also has a long duration of temporal support, and hence a long delay. Moreover, like energy-based methods, its computational demands that stem from spatiotemporal filters make it expensive on conventional hardware.

Common to all these methods is the use of FIR filters, requiring relatively large amounts of temporal storage and computation at each frame. They also involve a delay of approximately half the temporal duration of support for reasons of causality.

In comparison to Table 1, Table 2 shows results obtained from our implementation with different forms of IIR temporal prefiltering, all with the same spatial prefiltering and the same domain of minimization. In these cases we have used relatively low-orders of IIR filters, and a relatively small time-constant to keep the storage, computation and the delay small. In practice the delay should be between the mode $(n-1)\tau_1^{-1}$ and the mean $n\tau_1^{-1}$ of the impulse response (7); the parameters used here were chosen so that an integer delay was reasonably close to the mode. Also shown in the table are the effective standard deviations of the temporal impulse responses, and their frequency band-limits at amplitude half-height. For comparison, note that Gaussian filters with standard deviations $\sigma = 1.5$ and $\sigma = 2$ frames have half-height band-limits of $\omega = 0.78$ and $\omega = 0.58$ radians respectively, but Gaussian spectra fall off more sharply about this band-limit compared to the cascaded exponential filters.

Our main experimental concern is of course the quantitative accuracy of these methods. Accordingly, it is evident from Table 2 that the results are comparable to those in Table 1, with the greatest differences

[4] Although Haglund reports results with smaller temporal extents, the accuracy in these cases is somewhat poorer.

Params. $n,\ \tau_1^{-1}$	Extent, 1/2-height $\sqrt{n}\tau_1^{-1}$, $\tau_1\sqrt{2^{2/n}-1}$		Delay	Trans. Tree Error Stats. mean, st. dev. & density			Div. Tree Error Stats. mean, st. dev. & density		
3, 1.0	1.73	0.76	2	1.19°	0.77°	49.3%	1.99°	1.77°	51.9%
3, 1.25	2.16	0.61	3	0.97°	0.66°	45.6%	1.89°	1.63°	50.9%
4, 1.0	2.0	0.64	3	1.03°	0.63°	46.6%	1.93°	1.78°	48.8%
5, 1.0	2.2	0.56	4	0.85°	0.57°	44.3%	1.86°	1.61°	50.8%

Table 2. IIR Results I: *This shows results obtained with different forms of temporal prefiltering; i.e., different orders of IIR filters n and different time constants* τ_1^{-1}. *Also shown are the effective duration of support* $\sqrt{n}\tau_1^{-1}$, *the spectral extent at half-height* $\tau_1\sqrt{2^{2/n}-1}$ *and the temporal delay. Other parameters remained fixed at* $\sigma_1 = 1.5$, $\sigma_2 = 1.2$, *and* $\tau_2^{-1} = 0.83$ *(so that* $\alpha = 0.3$ *).*

Temporal Extent τ_2^{-1} , α		Trans. Tree Error Stats. mean, st. dev. & density			Div. Tree Error Stats. mean, st. dev. & density		
0.0	0.0	1.06°	0.8°	38.6%	2.01°	1.7°	49.0%
0.62	0.2	1.0°	0.7°	42.7%	1.92°	1.66°	50.1%
0.83	0.3	0.97°	0.66°	45.6%	1.89°	1.63°	50.9%
1.1	0.4	0.92°	0.62°	49.0%	1.85°	1.57°	52.5%

Table 3. IIR Results II: *This shows results obtained with different time constants for the exponential window on the domain of least-squares minimization. Other parameters:* $\sigma_1 = 1.5$, $n = 3$, $\tau_1^{-1} = 1.25$, *and* $\sigma_2 = 1.2$.

in the somewhat poorer results obtained for the translating tree sequence. Even the third-order filter with a delay of only two frames produces reasonably good results which is encouraging. We also find that performance tends to degrade when the time constant gets much smaller than 1.0. This appears to result from the use of a first-order filter in the cascade for computing the temporal low-pass response and its derivative.

Table 3 shows results for different temporal extents in the least-squares minimization (18) with the same prefiltering. It shows improvement with increasing temporal support. However, we also find that with sufficient spatial neighbourhoods, the temporal support of the minimization does not always improve the results significantly; spatial and temporal coherence complement one another in many cases, but not always. However, the temporal coherence does help fill in small holes where aperture problem occurs in textured patches. This may be important for egomotion methods that require dense flow (e.g., the convolution form of subspace methods [14]).

Finally, Figure 4 shows results of our method on two real image sequences that were also used by Barron et al. [3]. They show that the technique performs reasonably well, but produces occasional outliers and problems at occlusion boundaries, also shown in [3].

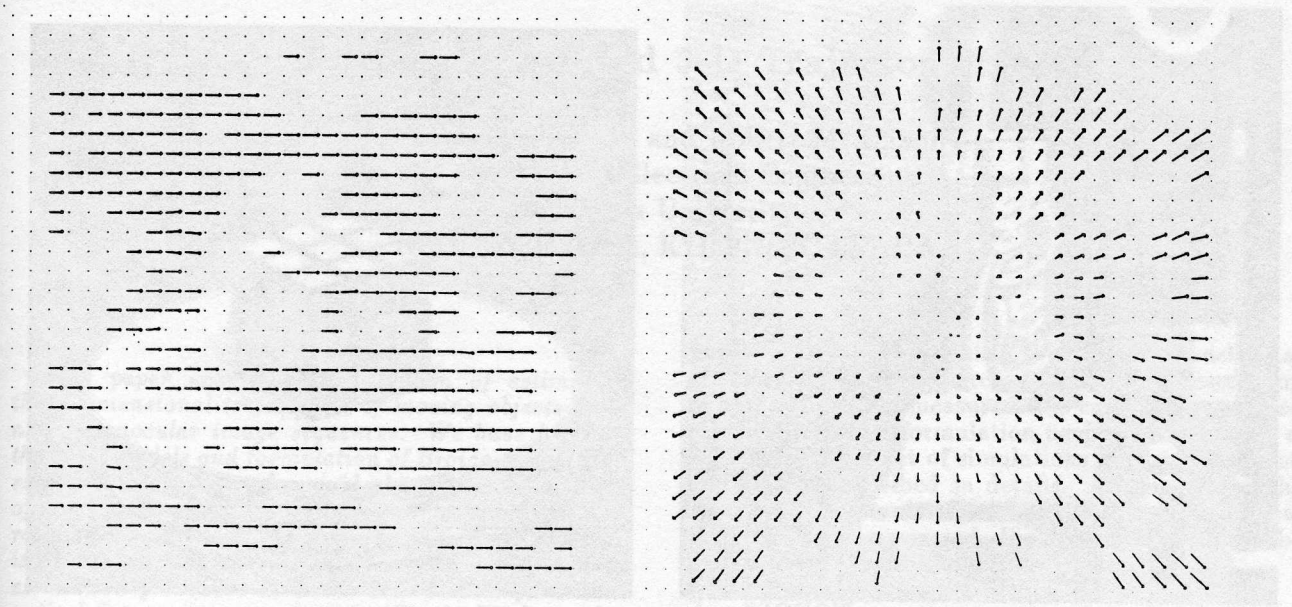
8. Discussion

This paper reports results that may be useful in developing more efficient, and potentially real-time, implementations of gradient-based and phase-based optical flow techniques. Here we concentrate on a class of low-pass IIR temporal filters that can be used efficiently to produce both the low-pass output and the temporal derivative. These filters can be applied with much shorter delays, and substantially less intermediate storage and computation compared to existing FIR-based approaches. We also examine a very simple form of recursion in solving for optical flow from normal velocity constraints. Furthermore, note that the durations of temporal support at both stages of computation can be extended with no increase in computation or storage, and only minor increases in delays. Using synthetic sequences it was shown that optical flow methods based on these recursive filters can approach the accuracy and reliability of explicit methods.

References

1. Adelson E.H. and Bergen J.R. (1986) The extraction of spatiotemporal energy in human and machine vision. *Proc. IEEE Workshop on Visual Motion*, Charleston, pp. 151-156

2. Barman H. Haglund L., Knutsson H., Granlund G. (1991) Estimation of velocity, acceleration and dis-



(A) (B)

Figure 3. **Estimated Flow Fields:** *Estimated flow fields for translating and diverging tree sequences using 3-order IIR prefilters with a time-constant of $\tau_1^{-1} = 1.25$ Other parameters were similar to those used for results in Table 2. Tiny dots represent pixels at which velocity estimates did not satisfy the eigenvalue threshold.*

parity in time sequences. *Proc. IEEE Workshop on Visual Workshop*, Princeton, pp. 44-51

3. Barron J.L., Fleet D.J., and Beauchemin S.S. (1993) Performance of optical flow techniques. *Int. J. Comp. Vis.* (to appear)
4. Black, M. and Anandan, P. (1991) Robust dynamic motion estimation over time. *IEEE Proc. CVPR*, Maui pp. 296-302
5. Bracewell, R. (1978) **The Fourier Transform and Its Applications**. McGraw-Hill, New York
6. Fleet, D.J. and Jepson, A.D. (1989) Hierarchical construction of orientation and velocity selective filters. *IEEE Trans. PAMI* 11, pp. 315-325
7. Fleet, D.J. and Jepson, A.D. (1990) Computation of component image velocity from local phase information. *Int. J. Comp. Vis.* 5, pp. 77-104
8. Fleet D.J. (1992) **Measurement of Image Velocity**. Kluwer Academic Publishers, Norwell
9. Girosi F. Verri A. and Torre V. (1989) Constraints for the computation of optical flow. *Proc. IEEE Workshop on Visual Motion*, Irvine, pp. 116-124
10. Haglund L. (1992) **Adaptive Multidimensional Filtering.** PhD Dissertation, Dept. Electrical Engineering, Univ. of Linkoping (ISSN 0345-7524)
11. Heeger D.J. (1988) Optical flow using spatiotemporal filters. *Int. J. Comp. Vis.* 1, pp. 279-302
12. Horn B.K.P. and Schunck B.G. (1981) Determining optical flow. *AI* 17, pp. 185-204
13. Jackson, L.B. (1989) **Digital Filters and Signal Processing**. Kluwer Academic Publishers, Norwell
14. Jepson, A. and Heeger, D. (1991) A fast subspace algorithm for recovering rigid motion. *Proc. IEEE Workshop on Visual Motion*, Princeton, pp. 124-131
15. Kearney J.K., Thompson W.B. and Boley D.L. (1987) Optical flow estimation: An error analysis of gradient-based methods with local optimization. *IEEE Trans. on PAMI* 9, pp. 229-244
16. Langley, K. and Fleet, D.J. (1992) Recursive filters for phase-based optical flow. *Isreali Conf. on Vision and AI*, Ramat Gan, December, pp. 255-264
17. Lucas, B. and Kanade, T. (1981) An iterative image registration technique with an application to stereo vision. *Proc. DARPA IU Workshop*, pp. 121-130
18. Nagel H.-H. (1987) On the estimation of optical flow: Relations between different approaches and some new results. *AI* 33, pp. 299-324
19. Simoncelli E.P., Adelson E.H. and Heeger D.J. (1991) Probability distributions of optical flow. *IEEE Proc. CVPR*, Maui pp. 310-315

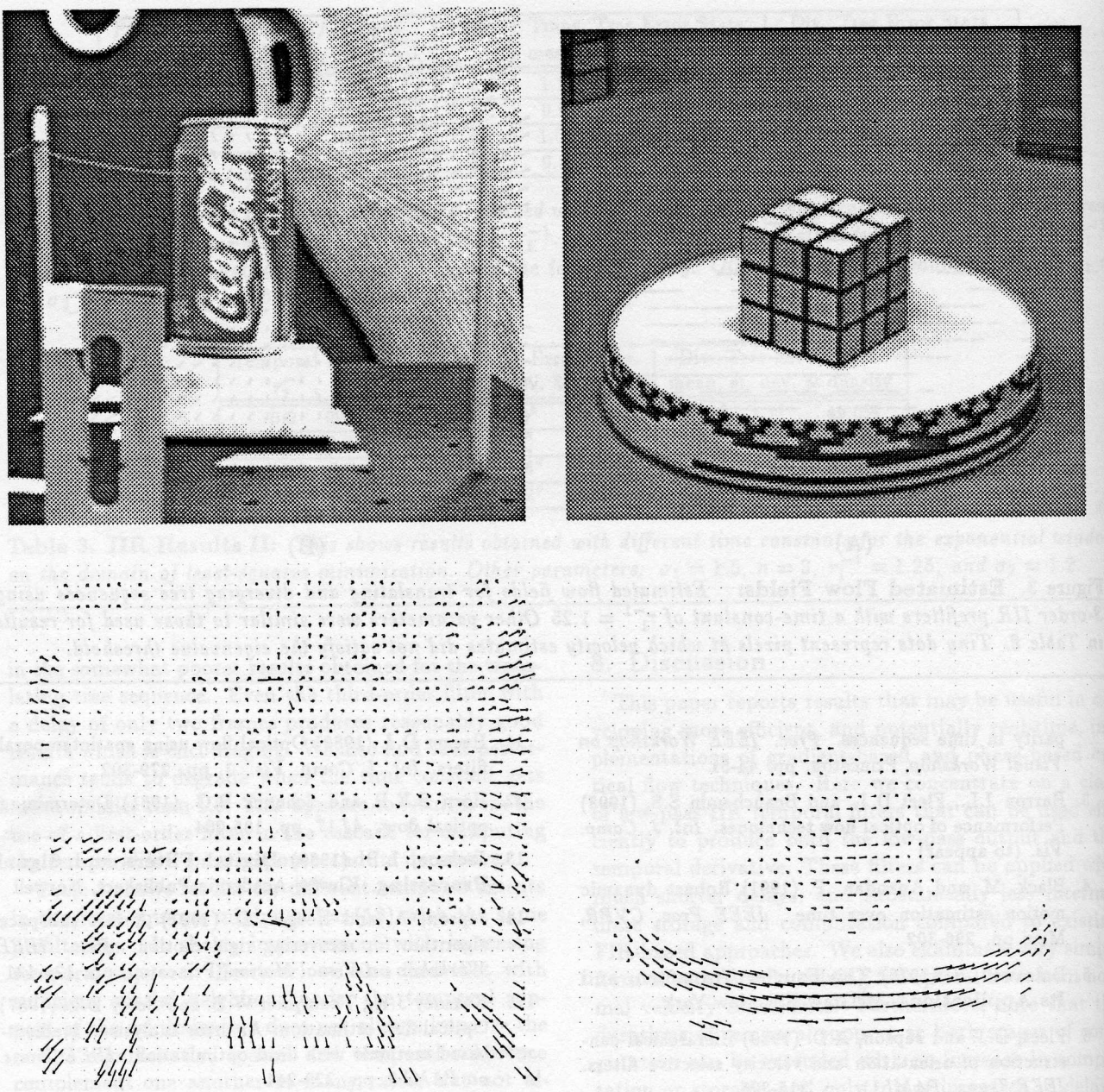


Figure 4. Real Sequences: *One frame and the flow field estimated at that frame are shown with the same parameters as those used for the results in Figure 3.*

20. Singh, A. (1992) Incremental estimation of image flow using a kalman filter. *J. Vis. Comm. Im. Rep.* 3, pp. 39-57

21. Uras S., Girosi F., Verri A. and Torre V. (1988) A computational approach to motion perception. *Biol. Cybern.* 60, pp. 79-97