Recursive Filters for Optical Flow

David J. Fleet, Keith Langley

***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, less computation, and shorter delays.**

I. Introduction

Many methods exist for computing optic flow, but few currently run at frame rates on available hardware. This paper discusses the development of efficient implementations of differential and phase–based methods. We concentrate on these because of their simplicity, and because of their performance compared to other methods [3].

One problem with these methods is their temporal filtering and differentiation, which involve large amounts of storage and computation, and a temporal delay. Low–pass prefilters are required by differential techniques for reliable numerical differentiation, while band–pass filters are an essential component of energy–based and phase–based approaches [1], [2], [8], [11], [12].

This paper examines a class of 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 [18]. We also examine a simple form of recursion in solving for optical flow from normal velocity constraints.

II. Local Gradient–Based Method

Gradient–based methods compute velocity from first–order derivatives of image intensity, or from filtered versions of the image, using the motion constraint equation

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

Manuscript received May 11, 1993; Revised January 20, 1994. Recommended for acceptance by Dr. Shmuel Peleg.

This work was financially supported by NSERC Canada and the Ontario Government under the ITRC centers.

D. J. Fleet is with the Department of Computing and Information Science, Queen's University, Kingston, Ontario, Canada K7L 3N6.

K. Langley is with the Department of Psychology, University College London, UK.

IEEE Log Number P95001

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 $\nabla I(\mathbf{x},t) \cdot \mathbf{v}(\mathbf{x},t)$ denotes the usual dot product.[1] The space–time gradient effectively measures the instantaneous velocity of level intensity contours, which can be derived from a conservation assumption (i.e. $dI(\mathbf{x},t)/dt = 0$). This measures the speed $s(\mathbf{x},t)$ in the direction of the image gradient $\mathbf{n}(\mathbf{x},t)$, given by

$$s(\mathbf{x},t) = \frac{-I_t(\mathbf{x},t)}{\|\nabla I(\mathbf{x},t)\|}, \; \mathbf{n}(\mathbf{x},t) = \frac{\nabla I(\mathbf{x},t)}{\|\nabla I(\mathbf{x},t)\|}. \tag{2}$$

But (1) represents only one constraint on the two unknowns in $\mathbf{v}(\mathbf{x},t)$, and therefore further constraints are necessary to solve for both. Three possibilities are well–known: (*i*) to use higher–order derivatives with additional conservation assumptions (*e.g.* [10], [24]); (*ii*) to impose global smoothness constraints (regularization) on the velocity field (*e.g.* [13], [20]); and (*iii*) to impose a parametric model (*e.g.* constant or linear variation) on the velocity field locally (*e.g.* [1], [4], [9], [17], [21], [22]).

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 $\mathbf{v}(\mathbf{x},t)$ in each small spatiotemporal neighborhood Ω by minimizing

$$\sum_{(\mathbf{x},t)\in\Omega} w(\mathbf{x},t)[\nabla I(\mathbf{x},t).\mathbf{v} + I_t(\mathbf{x},t)]^2 . \tag{3}$$

$w(\mathbf{x},t)$ is a window that gives more influence to constraints near the center of the neighborhood. The constant model for $\mathbf{v}$, and the support $w(\mathbf{x},t)$, reflect assumptions of spatial and temporal coherence (or smoothness) in the local behavior of $\mathbf{v}$ (*e.g.* [5], [22], [23]).

The minimization in (3) leads to a linear system $WA\mathbf{v} = W\mathbf{b}$, the solution of which is given by

$$\mathbf{v} = [A^T W A]^{-1} A^T W \mathbf{b}, \tag{4}$$

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

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

When $(A^T W A)^{-1}$ exists we can solve for $\mathbf{v}$, which is easy because $A^T W A$ is a 2 by 2 matrix:

$$A^T WA = \begin{bmatrix} \Sigma w I_x^2 & \Sigma w I_x I_y \\ \Sigma w I_y I_x & \Sigma w I_y^2 \end{bmatrix}, \; A^T W \mathbf{b} = \begin{bmatrix} \Sigma w I_x I_t \\ \Sigma w I_y I_t \end{bmatrix} \tag{5}$$

where sums are taken over points in the neighborhood Ω.

Finally, treating $[A^TWA]^{-1}$ as the covariance matrix for $\mathbf{v}$, unreliable estimates may be identified using its inverse eigenvalues (*e.g.* [17], [23]). 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^TWA [22], or the magnitude of its smallest eigenvalue λ_2 [3]. Such measures are important to the success of this technique.

As an example of the method, an implementation 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 (3) involved a purely spatial domain Ω at each frame to avoid further delays and computational expense.

III. Causal Recursive Temporal Filters

We now examine the use of recursive filters to reduce computation and storage requirements. First, let the smoothing 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 temporal filters we consider are derived from the truncated exponential [6] [7],

$$E(t) \equiv H(t)\, \tau e^{-\tau t} = \begin{cases} \tau \exp[-\tau t], & t \geq 0 \\ 0, & t < 0 \end{cases} \tag{6}$$

where $H(t)$ is a Heaviside step function. $E(t)$ in (6) is a low–pass causal filter where τ^{-1}, the time constant, determines the duration of temporal support. Its Fourier transform is

$$\hat{E}(\omega) = \frac{\tau}{\tau + i\omega}, \tag{7}$$

where $i^2 = -1$. Its amplitude and phase spectra are

$$|\hat{E}(\omega)| = \frac{\tau}{\sqrt{\tau^2 + \omega^2}}, \qquad \arg[\hat{E}(\omega)] = \tan^{-1}\left(\frac{\omega}{\tau}\right). \tag{8}$$

The amplitude spectrum is low–pass, symmetric, and broad (decaying like $1/\omega$ at high frequencies). The phase spectrum is nearly linear, especially at the low frequencies; 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 [6], it can be shown that repeated convolution of $E(t)$ is equivalent to convolution with a single kernel that has a particularly simple form:

$$En(t) \equiv [E(t)]^{*n} = \frac{(t\tau)^{(n-1)}}{(n-1)!} E(t). \tag{9}$$

From the central limit theorem it also follows that this cascade tends to a Gaussian [6]. The mode (peak) of the impulse response $E_n(t)$ is easily shown to be $(n-1)/\tau$. The corresponding mean (center of mass) and standard deviation are given by n/τ and $\sqrt{n}/\tau$. When using (9) 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 between the mean and the mode. As the number of cascades increases, the filter becomes more low–pass, and its phase spectrum remains nearly linear. The amplitude and phase spectra of $E_n(t)$ are

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

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

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

IV. Temporal Differentiation

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

$$\frac{\partial R_n(\mathbf{x},t)}{\partial t} = \frac{dE_n(t)}{dt} * \left[B(\mathbf{x}) * I(\mathbf{x},t)\right]\,, \tag{10}$$

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

Interestingly the temporal derivative can be computed using the same exponential filters discussed above. In particular, the temporal derivative of $E_n(t)$ has the form

$$\frac{dE_n(t)}{dt} = \delta(t)E_n(t) + \tau E_{n-1}(t) - \tau E_n(t)\,, \tag{11}$$

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

$$\frac{dE_n(t)}{dt} = \tau\left[E_{n-1}(t) - E_n(t)\right]\,, \text{ for } n \geq 2\,. \tag{12}$$

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.

V. Digital Filter Design

To realize discrete IIR implementations of (9) and (12), we use a bilinear transformation to map the Laplace transform (the s–domain) of the continuous filter onto the z–domain [14]: i.e.,

$$H(z) = H_c(s)\Big|_{s=\frac{2}{T}\left(\frac{1-z^{-1}}{1+z^{-1}}\right)} \tag{13}$$

The impulse invariant transformation is simpler [14], but it is susceptible to aliasing problems, which distorts the differentiation when computed according to (12).

The Laplace transform of the truncated exponential (6) is $\mathcal{L}[E(t)] = \tau/(s+\tau)$. Using the fact that convolution in time is equivalent to multiplication in the s–domain, the Laplace transform of $E_n(t)$ is easily shown to be

$$\mathcal{L}[E_n(t)] = \left[\frac{\tau}{s+\tau}\right]^n\,. \tag{14}$$

Rewriting the low–pass filter (14) as a cascade of an $(n-1)^{th}$–order filter and a first–order filter (used for (12)), we apply the bilinear transformation to obtain the z–transform

$$H_n(z) = \left(\frac{\tau}{s+\tau}\right)^{(n-1)}\left(\frac{\tau}{s+\tau}\right)\Bigg|_{s=2\left(\frac{1-z^{-1}}{1+z^{-1}}\right)} \tag{15}$$

For example, $H_3(z)$ is given by

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

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

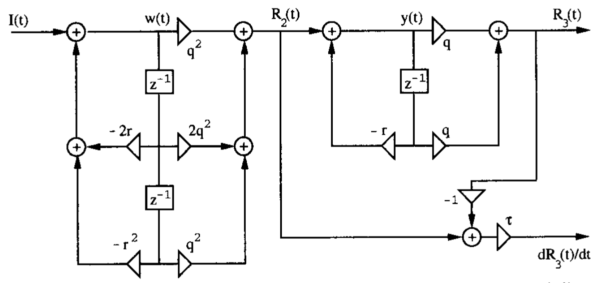


Fig. 1. This direct–form–II structure shows the delays, adders and multipliers used to implement the cascaded low–pass filter $E_3(t)$ and differential filter $dE_3(t)/dt$.

Discretizations then follow from the z–transform, using a direct–form–II structure [14]. Fig. 1 shows a cascaded implementation of the low–pass filter and the differentiator when $n = 3$. The difference equations (with t now a discrete variable) for the first stage of the cascade are

$$\begin{aligned} w(t) &= I(t) - 2r\,w(t-1) - r^2\,w(t-2) \\ R_2(t) &= q^2\,w(t) + 2q^2\,w(t-1) + q^2\,w(t-2) \end{aligned} \tag{16}$$

while the second stage is given by

$$\begin{aligned} y(t) &= R_2(t) - ry(t-1) \\ R_3(t) &= qy(t) + qy(t-1) \end{aligned} \tag{17}$$

and the temporal derivative is given by

$$\frac{dR_3(t)}{dt} = \tau\left[R_2(t) - R_3(t)\right]\,. \tag{18}$$

The bilinear transform warps frequency space in a nonlinear manner, so the difference of low–pass filters in (12) will no longer be a perfect differentiator. It is therefore interesting to examine the accuracy of this IIR scheme compared to common forward and central–difference schemes. One measure of the accuracy of a numerical differentiation scheme is the difference between its transfer function (i.e., its Fourier transform) and $i\omega$, since differentiation is equivalent to multiplication with $i\omega$ in frequency space, i.e., $\mathcal{F}[df(t)/dt] = i\omega\mathcal{F}[f(t)]$. Given the Fourier transform of the differentiation scheme $\hat{D}\omega)$, our measure of error is

$$E(\omega) = \left|\,\hat{D}(\omega) - i\,\omega\,\right|\,. \tag{19}$$

Table I gives the convolution masks for several common FIR filters for numerical differentiation, and their respective transfer functions (see [16] for more details). It also gives a RMS measure of error, computed as the integral of $E(\omega)^2$ over radian frequencies $0 \le \omega \le 1.0$ (the lowest third of the frequency spectrum). With all these schemes, accuracy degrades and noise becomes more severe at higher frequencies, which is why it is common to smooth the signal before differentiation. For example, Gaussian filters with standard deviations $\sigma = 1.5$ and $\sigma = 2$ have half–height band–limits of $\omega = 0.78$ and $\omega = 0.58$ radians.

The transfer function for the IIR method can also be derived without much trouble: Remember that the relationship between the derivative filter and its corresponding low–pass components (12) is exact. It is also straightforward to show that this relationship holds in

TABLE I

THREE FIR DIFFERENTIATION SCHEMES ARE SHOWN WITH THEIR CONVOLUTIONS MASKS AND FOURIER TRANSFORMS: A 2–POINT FORWARD–DIFFERENCE FORMULA (FD2PT), A 2–POINT CENTRAL–DIFFERENCE (CD2PT), AND A 4–POINT CENTRAL DIFFERENCE (CD4PT).

Scheme	Convolution Mask	Transfer Function	RMS Error ($0 \le \omega \le 1.0$)
fd2pt	$(1, -1)$	$1 - \cos(\omega) + i\sin(\omega)$	0.219
cd2pt	$\frac{1}{2}(1, 0, -1)$	$i\sin(\omega)$	0.061
cd4pt	$\frac{1}{12}(-1, 8, 0, -8, 1)$	$i\frac{1}{6}[8\sin(\omega) - \sin(2\omega)]$	0.0091

the Laplace domain since the Laplace transform is linear; that is, from (12) and (14),

$$\frac{\mathrm{L}[dE_n(t)/dt]}{\mathrm{L}[E_n(t)]} = \frac{\tau\left(\mathrm{L}[E_{(n-1)}(t)] - \mathrm{L}[E_n(t)]\right)}{\mathrm{L}[E_n(t)]} = s \ , \quad (20)$$

Using the bilinear transformation $s = 2(1 - z^{-1}) / (1 + z^{-1})$ to transform this to the Z–domain, and substituting for $z = e^{i\omega}$, we obtain the transfer function

$$\hat{D}_{iir}(\omega) = 2\left(\frac{1-z^{-1}}{1+z^{-1}}\right)\Bigg|_{z=e^{i\omega}} = \frac{i2\sin(\omega)}{1+\cos(\omega)} \ . \quad (21)$$

Fig. 2 shows the errors (19) for the FIR differentiators compared to this IIR scheme based on the bilinear transformation. It shows that the IIR scheme should perform better than 2–point schemes for reasonably low frequencies, but worse than a 4–point central difference. The RMS error for the IIR scheme, compared to those in Table I, is 0.034.

VI. Estimation of Optical Flow

We now consider the weighted least–squares estimation of image velocity from the normal constraints (3). For convenience, we assume separability of the window w(x, t) in (3) that determines the support of the minimization so that its spatial and temporal components can be examined separately. Ignoring the temporal component of the window, like the implementations described by others [3] [22], we first note that the spatial collection of squared constraints (the components of (5) at each pixel) can be expressed as convolution. That is, from the partial derivatives of the low–pass filtered image R(x, t), we form the five images

$$R_x^2(\mathbf{x},t) \ , \ R_y^2(\mathbf{x},t) \ , \ R_x(\mathbf{x},t)\, R_y(\mathbf{x},t) \ ,$$
$$R_x(\mathbf{x},t)\, R_t(\mathbf{x},t) \ , \ R_y(\mathbf{x},t)\, R_t(\mathbf{x},t) \quad (22)$$

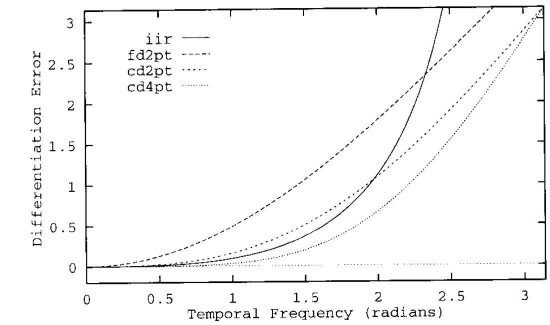


Fig. 2. The accuracy of differentiation in frequency space is shown for the IIR method and the three FIR schemes in Table I. Remember the effective band–limits of the low–pass prefilters are assumed to be below w = 1.0, which determines the critical passband.

The entries in the linear system in (5), at each pixel, are then given by the convolution of these intermediate images with spatial component of the window, for which we use a Gaussian $G(\mathbf{x})$.

The use of temporal support is somewhat more problematic because of the computational expense and storage needed to accumulate normal constraints and solve the minimization problem at each time. To alleviate this problem, yet exploit some degree of temporal coherence, it is natural to consider incremental methods. For example, Singh [23] uses recursive estimation based on the Kalman filter, while Black and Anandan [5] describe an MRF–based method. Although simplifications are made to the optimization procedures in each of these cases, they still involve considerable computational expense and storage.[2] 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 temporal window. As stated above, the accumulation of constraints through space–time is a form of low–pass filtering of the intermediate images (22). In space we use an FIR Gaussian filter. In time we use a causal IIR filter with exponential support (6), the implementation of which requires only a first–order temporal difference. More precisely, let $\bar{A}(\mathbf{x}, t)$ and $\bar{\mathbf{b}}(\mathbf{x}, t)$ represent the system of normal equations formed by convolution of the intermediate images at time t with a Gaussian $G(\mathbf{x})$. Then, the temporal accumulation of constraints within an exponential window with time constant τ_2^{-1} is given by

$$A(\mathbf{x}, t) = \alpha\, A(\mathbf{x}, t-1) + (1-\alpha)\, \bar{A}(\mathbf{x}, t) \ , \quad (23)$$
$$\mathbf{b}(\mathbf{x}, t) = \alpha\, \mathbf{b}(\mathbf{x}, t-1) + (1-\alpha)\, \bar{\mathbf{b}}(\mathbf{x}, t) \ ,$$

where $\alpha = \exp(-\tau_2)$. The solution to the resultant normal equations is then given by $\mathbf{v} = [A(\mathbf{x}, t)]^{-1}\, \mathbf{b}(\mathbf{x}, t)$. The recursion involves the normal equations rather than the final velocity estimates. Although this is perhaps the simplest form of temporal coherence, it comes at a very small cost.[3]

2 They also solve for occlusion boundaries as well as optical flow and therefore address a somewhat broader problem than that addressed here.

3 For example, 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. Also, the exponentially decaying temporal window can be regarded as a form of process noise, where the relative amounts of measurement and process noise remain fixed.

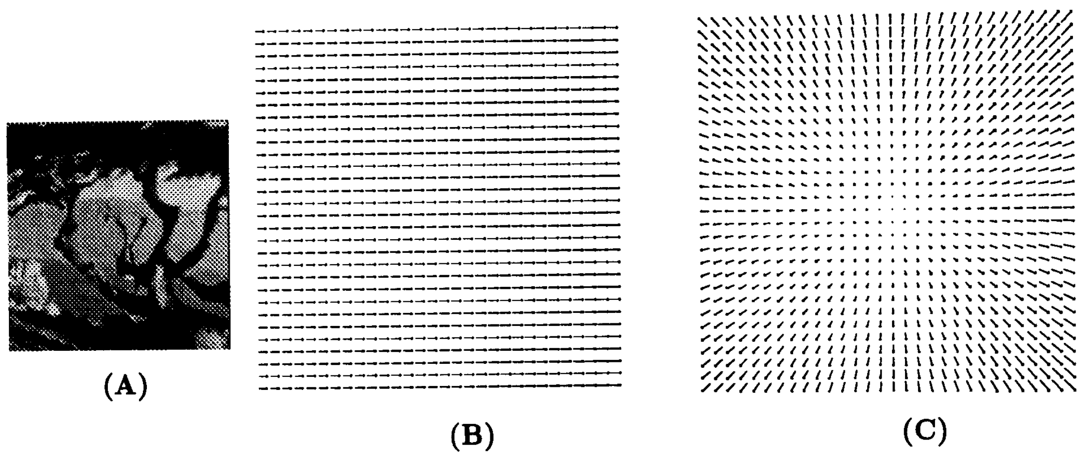


Fig. 3. (A) shows the texture on the moving surface at a single frame. (B) 2D motion field for the translating tree sequence, where the camera moves along its x-axis normal to its line of sight with image velocities between (1.73, 0.0) and (2.3, 0.0) pixels per frame. (C) 2D motion field for the diverging tree sequence, where the camera moves along its line of sight. The focus of expansion is at the image center, and speeds vary from 1.4 pixels per frame on the side to 2.0 pixels per frame on the other.

VII. Experimental Results

Our implementation involves two stages, namely, low–pass smoothing and differentiation of $I(\mathbf{x}, t)$, followed by the computation of 2d velocity from the normal constraints. With the computation of $\mathbf{v}(\mathbf{x}, t)$ we use a confidence measure based on the smallest eigenvalue of $A(\mathbf{x}, t)$, λ_2 [3], with a simple threshold $\lambda_2 < 1.0$ to detect situations in which the aperture problem prevails or the gradient magnitudes are too small. We do this for convenience in comparing our results to those of others; we acknowledge that confidence measures should be maintained with the velocity estimates, using a conservative threshold only to remove outliers.

The entire method can be viewed as a sequence of image operations, suitable for SIMD or pipeline hardware. It also lends itself to fast implementation on serial machines. The main parameters 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 temporal delay in velocity estimation is due only to the first stage of filtering, and depends on the order of the IIR filter and the time–constant. Although our implementation uses only one spatial scale, the method generalizes to multiple spatial scales for which we expect improved results.

A. Error Measure

Following [3] [9], we view velocity as orientation in space–time and use an angular measure of error. If velocity $\mathbf{v} = (v_1, v_2)^T$ is represented as a unit direction vector,

$$\vec{\mathbf{v}} \equiv \frac{1}{\sqrt{v_1^2 + v_2^2 + 1}} \left(v_1, v_2, 1\right)^T ,$$

then the error between the correct velocity $\vec{\mathbf{v}}_c$ and an estimate $\vec{\mathbf{v}}_e$ is

$$\psi_E = \arccos\left(\vec{\mathbf{v}}_c \cdot \vec{\mathbf{v}}_e\right) . \tag{24}$$

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% [9].

B. Results

Our quantitative results are derived from two synthetic sequences with known 2d motion fields. Each sequence depicts the 3d translation of a textured plane (see Fig. 3). Table II shows results reported by others on the same sequences. The gradient–based method by Barron et al. [3], described in Section 2, uses spatiotemporal Gaussian smoothing, and a spatial domain for the minimization (3) with Gaussian weights (a standard deviation of 1.1 pixels). Our implementations retain these parameters where appropriate. The energy–based technique of Haglund [2] [11] produced accurate results, but the measurement density was lower than the other methods. This technique also had a longer duration of temporal support, and hence a longer delay. The phase–based technique [8] [9] produced more accurate results, with a higher measurement density. It also has large temporal support and a long delay.

By comparison, Table III shows results from our implementation with different IIR prefilters, all with the same spatial prefiltering and the same domain of minimization. We have used relatively low–orders of IIR filters, and a small time–constant to keep the storage,

TABLE II

RESULTS OF OTHER IMPLEMENTATIONS ON THE SAME DATA.

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 [11]	21	10	0.85°	0.42°	21%	1.77°	1.14°	17%
Fleet & Jepson [8] [9]	21	10	0.23°	0.19°	49.7%	0.8°	0.73°	46.5%

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 (9). In fact, errors are often smallest with a delay near the mean, but there is not much difference between the estimates at the mode and the mean (see Fig. 4). For Table III the parameters were chosen so that an integer delay was close to the mode.

Our main experimental concern is the quantitative accuracy of these methods. Accordingly, it is evident from Table III that our results are similar to those in Table II, with the greatest differences being the poorer results obtained for the translating tree sequence. Even

TABLE III

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 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 WERE FIXED AT Σ_1=1.5, Σ_2=1.2, AND $\tau_2^{-1} = 0.83$ (A = 0.3).

Params. n, τ_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%

the third–order filter with a delay of only 2 frames produces good results which is encouraging. These results also agree with the analysis in Section 5 that shows the IIR differentiation scheme is less accurate than 4–point central differences (used in [3]).

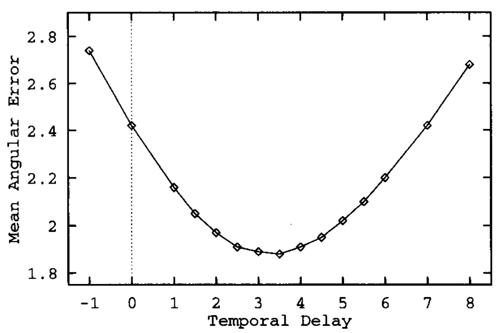


Fig. 4. Mean error is shown as a function of delay. Although the minimum error occurs at a delay close to the mean, errors are generally small everywhere between the mode and the mean. We use the mode instead of the mean as it involves a shorter delay.

Interestingly, performance tends to degrade when the time constant gets much smaller than 1.0. In part this may be due to the increased temporal band–limit in these cases, where noise is a greater problem for differentiation. The effective temporal extent of the filters and their temporal band–limits (at half amplitude) are given in Table III. As mentioned in Section 5 they are comparable to those obtained with Gaussian smoothing with standard deviations between 1.5 and 2.0. However, note that Gaussian spectra fall off more sharply at half–height compared to cascaded exponential filters.

TABLE IV

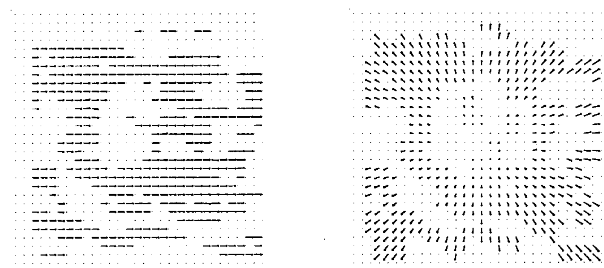
THIS SHOWS RESULTS OBTAINED WITH DIFFERENT TIME CONSTANTS FOR THE EXPONENTIAL WINDOW ON THE DOMAIN OF LEAST–SQUARES MINIMIZATION (WITH Σ_1=1.5, N=3, τ_1^{-1} = 1.25, AND Σ_2=1.2).

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 IV shows results for different temporal extents in the least–squares minimization (23) with the same prefiltering. It shows improvement with increasing temporal support. However, we also find that with sufficient spatial neighborhoods, 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. The temporal coherence does however 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 [15]). Finally, Fig. 5 shows the flow fields computed from the synthetic sequences using a third–order prefilter with τ_1^{-1} = 1.25, the quantitative results for which are given in the second row of Table III. We have also run this method on the real sequences used in the study by Barron et al. [3] with similarly good results.

VIII. Discussion

This paper reports results that may be useful in developing efficient implementations of gradient and phase–based optical flow methods. 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 simple form of recursion in solving for optical flow from normal velocity constraints. 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 delay. Using synthetic sequences it was shown that methods based on these recursive filters can approach the accuracy of explicit methods.



(A) (B)

Fig. 5. These are estimated flow fields for the translating and diverging tree sequences using third–order IIR prefilters with a time–constant of τ_1^{-1} = 1.25 Other parameters were similar to those used for Table III. The dots indicate pixels where velocity estimates failed the eigenvalue threshold.

Acknowledgments

We are grateful to Paul Hodgins for helping to implement the IIR filters, and to Allan Jepson and Hong Jiang for comments on drafts of this paper.

References

[1] E. Adelson and J. Bergen, "The extraction of spatiotemporal energy in human and machine vision," *Proc. IEEE Workshop on Visual Motion*, Charleston, pp. 151–156, 1986.

[2] H. Bårman, L. Haglund, H. Knutsson, G. Granlund, "Estimation of velocity, acceleration and disparity in time sequences" *Proc. IEEE Motion Workshop*, Princeton, pp. 44–51, 1991.

[3] J. Barron, D. Fleet, and S. Beauchemin, "Performance of optical flow techniques," *Int. J. Comp. Vision* 12, pp. 43–77, 1994.

[4] J. Bergen, P. Anandan, K. Hanna, and R. Hingorani, "Hierarchical model–based motion estimation." *Proc. ECCV*, Santa Margherita, Italy, pp. 237–252, Springer–Verlag, 1992.

[5] M. Black and P. Anandan, "Robust dynamic motion estimation over time," *IEEE Proc. CVPR*, Maui, pp. 296–302, 1991.

[6] R. Bracewell, *The Fourier Transform and Its Applications*. McGraw–Hill, New York, 1978.

[7] D. Fleet and A. Jepson, "Hierarchical construction of orientation and velocity selective filters," *IEEE Trans. PAMI* 11, pp. 315–325, 1989.

[8] D. Fleet and A. Jepson, "Computation of component image velocity from local phase information," *Int. J. Comp. Vision* 5, pp. 77–104, 1990.

[9] D. Fleet, *Measurement of Image Velocity*. Kluwer Academic Publishers, Norwell, 1992.

[10] F. Girosi, A. Verri, and V. Torre, "Constraints for the computation of optical flow," *Proc. IEEE Motion Workshop*, Irvine, pp. 116–124, 1989.

[11] L. Haglund, *Adaptive Multidimensional Filtering*. PhD Dissertation, Dept. Electrical Engineering, Univ. of Linkoping (ISSN 0345–7524), 1992.

[12] D. Heeger, "Optical flow using spatiotemporal filters," *Int. J. Comp. Vision* 1, pp. 279–302, 1988.

[13] B. Horn and B. Schunck, "Determining optical flow," *AI* 17, pp. 185–204, 1981.

[14] L. Jackson, *Digital Filters and Signal Processing*. Kluwer Academic Publishers, Norwell, 1989.

[15] A. Jepson and D. Heeger, "A fast subspace algorithm for recovering rigid motion," *Proc. IEEE Motion Workshop*, Princeton, pp. 124–131, 1991.

[16] E. Karabassis and M. Spetsakis, "Analysis of fundamental image operations," (submitted to CVGIP), 1993.

[17] J. Kearney, W. Thompson, and D. Boley, "Optical flow estimation: An error analysis of gradient–based methods with local optimization," *IEEE Trans. PAMI* 9, pp. 229–244, 1987.

[18] K. Langley and D. Fleet, "Recursive filters for phase–based optical flow," *Israeli Conf. on Vision and AI*, Ramat Gan, pp. 255–264, 1992.

[19] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," *Proc. DARPA IU Workshop*," pp. 121–130, 1981.

[20] H. Nagel, "On the estimation of optical flow: Relations between different approaches and some new results," *AI* 33, pp. 299–324, 1987.

[21] R. Schalkoff and E. McVey, "A model and tracking algorithm for a class of video targets," *IEEE Trans. PAMI* 4, pp. 2–10, 1982.

[22] E. Simoncelli, E. Adelson, and D. Heeger, "Probability distributions of optical flow," *IEEE Proc. CVPR*, Maui pp. 310–315, 1991.

[23] A. Singh, "Incremental estimation of image flow using a Kalman filter," *J. Vis. Comm. Image Repres.* 3, pp. 39–57, 1992.

[24] S. Uras, F. Girosi, A. Verri, and V. Torre, "A computational approach to motion perception," *Biological Cybernetics* 60, pp. 79–97, 1988.

0162–8828/95$04.