Recursive Filters for Optical Flow[1]

David J. Fleet
Department of Computing and Information Science,
Queen's University,
Kingston, Canada, 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, 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 reduce computational expense 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 [3]. A problem with these methods is their spatiotemporal filtering and differentiation, which involve 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, 8, 9, 11, 12]. 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 shorter 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.

[1]Published in *IEEE Trans. PAMI*, 17(1): 61–67, 1995

2 Local Gradient-Based Method

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

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

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 $\nabla I(\mathbf{x},\, t) \cdot \mathbf{v}(\mathbf{x},\, t)$ 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

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

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 neighbourhood Ω by minimizing

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

$W(\mathbf{x},\, t)$ is a window that gives more influence to constraints near the centre of the neighbourhood. The constant model for $\mathbf{v}$, and the support $W(\mathbf{x},\, t)$, reflect assumptions of spatial and temporal coherence (or smoothness) in the local behaviour 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 $(\mathbf{x}_i, t_i) \in \Omega$,

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

[2] 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). This is not addressed here in detail.

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:

$$A^T W A \;=\; \begin{bmatrix} \sum W I_x^2 & \sum W I_x I_y \\ \sum W I_y I_x & \sum W I_y^2 \end{bmatrix} \quad , \qquad A^T W \mathbf{b} \;=\; \begin{bmatrix} \sum W I_x I_t \\ \sum W I_y I_t \end{bmatrix} \tag{5}$$

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) (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 $A^T W A$, $\lambda_1 + \lambda_2$ [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 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 (3) 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 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 class of temporal filters we consider is 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. Equation (6) is a low-pass causal filter where τ^{-1}, the time constant, determines the duration of temporal support. The Fourier transform of $E(t)$ is

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

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

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

The amplitude spectrum is low-pass, symmetric about the origin where it takes a value of 1, 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:

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

From the Central Limit Theorem it also follows that this cascaded convolution tends to a Gaussian function [6]. 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 (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 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

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

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 involves 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

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

where $R_n(\mathbf{x},\,t) = E_n(t) * [B(\mathbf{x}) * I(\mathbf{x},\,t)]$. However, it is expensive to apply a different temporal filter for differentiation, either 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 (9) 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

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

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.

5 Digital Filter Design

In order 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.,

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

Although the impulse invariant transformation is somewhat simpler,[14] it is susceptible to severe aliasing problems, which in this case distort the differentiation when computed according to (12).

The Laplace transform of the truncated exponential (6) 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

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

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

Discrete implementations then follow directly from the z-transform, using a direct-form-II structure [14]. Figure 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

$$\begin{aligned} y(t) &= R_2(t) \;-\; r\, y(t-1) \\ R_3(t) &= q\, y(t) \;+\; q\, y(t-1)\;, \end{aligned} \tag{17}$$

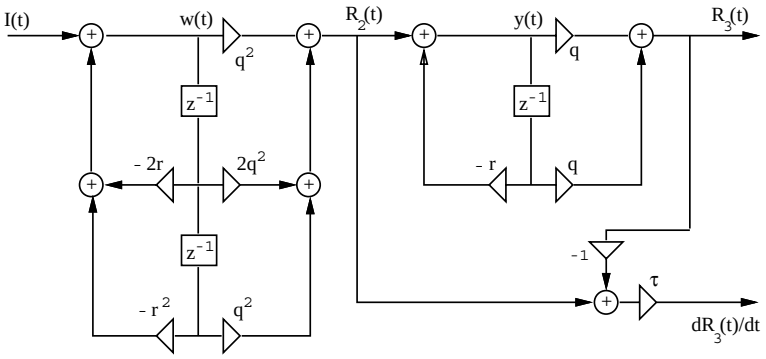


Figure 1. *This direct-form-II structure shows the combination of delays, adders and multipliers used to implement the cascaded low-pass filter $E_3(t)$ and differential filter $dE_3(t)/dt$.*

and the temporal derivative is given by

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

In order to reduce aliasing, the bilinear transform effectively warps frequency space in a nonlinear manner, so the difference of low-pass filters in (12) will no longer be a perfect differentiator. As a consequence it is interesting to examine the accuracy of this IIR scheme in comparison to the 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, $\mathcal{F}[df(t)/dt] = i\,\omega\,\mathcal{F}[f(t)]$. 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 1 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 our derivative filter and its corresponding low-pass components (12) is exact. It is also straightforward to show that this relationship holds in the Laplace domain since the Laplace transform is linear; that is, from (12) and (14),

$$\frac{\mathcal{L}[dE_n(t)/dt]}{\mathcal{L}[E_n(t)]} \;=\; \frac{\tau(\mathcal{L}[E_{(n-1)}(t)] - \mathcal{L}[E_n(t)])}{\mathcal{L}[E_n(t)]} \;=\; s\;, \tag{20}$$

Scheme	Convolution Mask	Transfer Function	RMS Error ($0 \leq \omega \leq 1.0$)
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

Table 1. *FIR differentiation schemes are shown with their associated convolutions masks and Fourier transforms, including a 2-point forward-difference formula (fd2pt), a 2-point central-difference formula (cd2pt), and a 4-point central difference (cd4pt).*

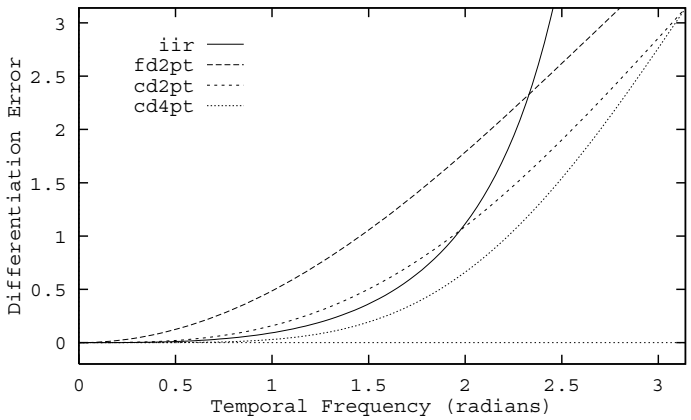


Figure 2. *The accuracy of differentiation in frequency space is shown for the IIR method and the three FIR schemes given in Table 1. Remember that the effective band-limits of the low-pass prefilters are assumed to be below $\omega = 1.0$, which determines the critical band for which errors must be small.*

Applying 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

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

Figure 2 shows the errors (19) for the FIR differentiators compared to this IIR scheme based on the bilinear transformation. It can be seen that the IIR scheme should perform better than 2-point schemes for reasonably low frequencies and worse than a 4-point central difference scheme. The RMS error for the IIR scheme, in comparison to those listed in Table 1, is 0.034.

6 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 $W(\mathbf{x}, t)$ 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. In other words, given the partial derivatives of the low-pass filtered image $R(\mathbf{x},\, t)$, we first form the five images

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

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 which we take to be 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.[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 (22). In space we apply an FIR Gaussian filter. In time we use a causal IIR filter with exponential support (6), 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 (22) at time t with the Gaussian weights $G(\mathbf{x})$. Then, the temporal accumulation of constraints within an exponential window of support (6), with time constant τ_2^{-1}, is computed as

$$\begin{aligned} A(\mathbf{x},\, t) &= \alpha\, A(\mathbf{x},\, t-1) \,+\, (1-\alpha)\, \bar{A}(\mathbf{x},\, t)\;, \\ \mathbf{b}(\mathbf{x},\, t) &= \alpha\, \mathbf{b}(\mathbf{x},\, t-1) \,+\, (1-\alpha)\, \bar{\mathbf{b}}(\mathbf{x},\, t)\;, \end{aligned} \tag{23}$$

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

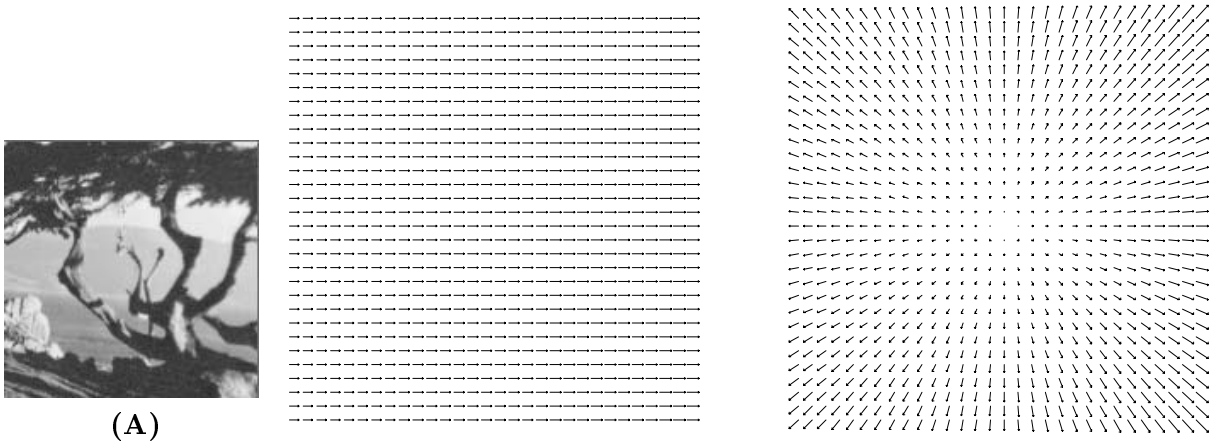
$$\mathbf{v} \;=\; [A(\mathbf{x},\, t)]^{-1}\, \mathbf{b}(\mathbf{x},\, t)\;. \tag{24}$$

Observe that with $\alpha = 0$ in (24) we remove the temporal coherence in this computation.

This recursion directly 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.[4]

[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.

[4] 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 might be regarded as a form of the process noise, where the relative amounts of measurement and process noise remain fixed.



(A) **(B)** **(C)**

Figure 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/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 image centre, and speeds vary from 1.4 pixels/frame on one side to 2.0 pixels/frame on the other.*

7 Experimental Results

In summary, the implementation described above can be broken into two main stages of processing:

1. Spatiotemporal low-pass smoothing and differentiation of $I(\mathbf{x}, t)$, accomplished in three stages:
   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 (16)–(18), 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 (22);
   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 (24);
   c) solution of the normal equations (24).

Following the computation of $\mathbf{v}(\mathbf{x}, t)$ we compute 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 (24); 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 $\lambda_2 < 1.0$ to detect situations in which the aperture problem prevails or the gradient magnitudes are too small. 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 [22]).

In summary, the entire method (with the confidence threshold) 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 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 use an angular measure of error. If velocity $\mathbf{v} = (v_1, v_2)^T$ is represented as a 3-d unit direction vector, $\vec{\mathbf{v}} \equiv \frac{1}{\sqrt{v_1^2+v_2^2+1}}(v_1,\ v_2,\ 1)^T$, then the error between the correct velocity $\vec{\mathbf{v}}_c$ and an estimate $\vec{\mathbf{v}}_e$ is

$$\psi_E = \arccos(\vec{\mathbf{v}}_c \cdot \vec{\mathbf{v}}_e) \ . \tag{25}$$

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].

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., [11]). 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 3.

Table 2 shows results reported by others on the same sequences. The gradient-based implementation by Barron et al., described above in Section 2, uses spatiotemporal Gaussian low-pass filtering, and a spatial domain for the minimization (3) with Gaussian weights (with 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 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.[5] The phase-based technique [8, 9] 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.

By comparison, Table 3 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.

[5] Haglund reports results with smaller temporal extents but the accuracy in these cases is somewhat poorer.

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%

Table 2. *This shows results of other implementations on the same data.*

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%

Table 3. *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 4. *This shows results obtained with different time constants for the exponential window on the domain of least-squares minimization (with $\sigma_1 = 1.5$, $n = 3$, $\tau_1^{-1} = 1.25$, and $\sigma_2 = 1.2$).*

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 (9). We find that the errors are typically minimal with a delay close to the mean, but there is not much difference between the estimates at the mode and the mean. For example, Figure 4 shows the mean error as a function of the temporal delay for the diverging tree sequence (the standard deviation is roughly constant because of the smooth change in the motion field from frame to frame, and is not shown). In most instances reported in Table 3, the parameters were chosen so that an integer delay was reasonably close to the mode.

Our main experimental concern is of course the quantitative accuracy of these methods. Accordingly, it is evident from Table 3 that the results are comparable to those in Table 2, with the greatest differences in the somewhat poorer results obtained for the translating tree sequence. Even 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 to be some-

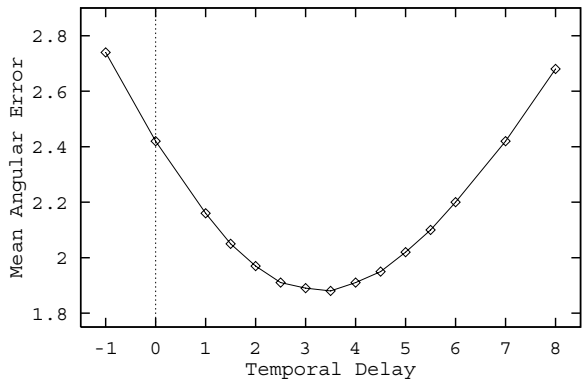


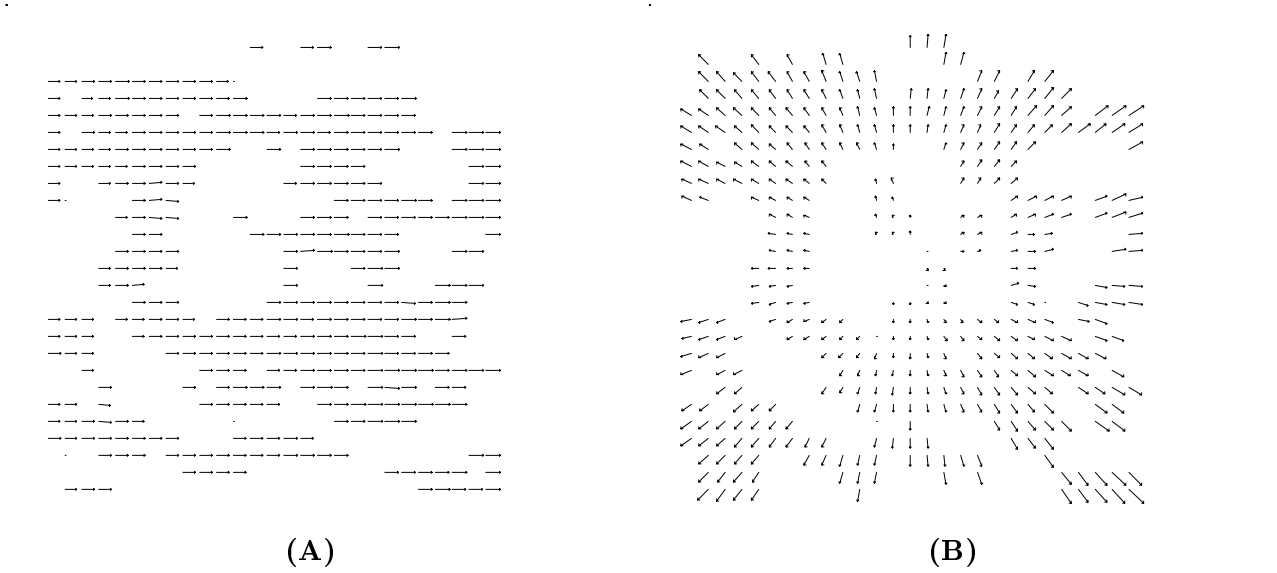
Figure 4. *Mean error is shown as a function of delay. Although the minimum error occurs at a delay close to the mean, errors are generally small everywhere between the mode and the mean. We use the mode instead of the mean as it involves a shorter delay.*

what less accurate than a 4-point central difference (used by Barron et al.). Although not reported in detail, we found that the IIR method produces significantly better results than the 2-point FIR formulas in Table 1 for numerical differentiation.

Interestingly, performance tends to degrade when the time constant gets much smaller than 1.0. In part this may be due to the the increased temporal band-limit in these cases, where noise is a greater problem for differentiation schemes. The effective temporal extent of the filters and their temporal band-limits (at half amplitude) are given in Table 3. As mentioned in Section 5 they are comparable to those obtained with FIR Gaussian smoothing with standard deviations between 1.5 and 2.0. However, note that Gaussian spectra fall off more sharply about their half-height band-limit compared to cascaded exponential filters, the band-limits of which are given in Table 3.

Table 4 shows results for different temporal extents in the least-squares minimization (24) 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. 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, Figure 5 shows the flow fields computed from the synthetic sequences using a third-order prefilter with $\tau_1^{-1} = 1.25$, the quantitative results for which are given in the second row of Table 3. Figure 6 shows results of the same filters 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].



(A) **(B)**

Figure 5. *These are estimated flow fields for the translating and diverging tree sequences using third-order IIR prefilters with a time-constant of* $\tau_1^{-1} = 1.25$ *Other parameters were similar to those used for the results reported in Table 3. The tiny dots represent pixels at which the velocity estimate did not satisfy the eigenvalue threshold.*

8 Discussion

This paper reports results that may be useful in developing more efficient 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 temporal delay. Using synthetic and real sequences it was shown that optical flow methods based on these recursive filters can approach the accuracy and reliability of explicit methods.

Acknowledgements: 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. We are also grateful to NSERC Canada, and the Ontario Government under the ITRC centres for their financial support.

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] Bårman H. Haglund L., Knutsson H., Granlund G. (1991) Estimation of velocity, acceleration and disparity in time sequences *Proc. IEEE Workshop on Visual Workshop*, Princeton, pp. 44-51

[3] Barron J.L., Fleet D.J., and Beauchemin S.S. (1994) Performance of optical flow techniques, *Int. J. Comp. Vision* 12, pp. 43-77 (also see RPL-TR-9207, Dept. Computing Science, Queen's University)

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

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

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

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

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

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

[10] 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

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

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

[13] Horn B.K.P. and Schunck B.G. (1981) Determining optical flow, *AI* 17, pp. 185-204

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

[15] Jepson, A. and Heeger, D. (1991) A fast subspace algorithm for recovering rigid motion. *Proc. IEEE Workshop on Visual Motion*, Princeton, pp. 124-131

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

[17] 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. PAMI* 9, pp. 229-244

[18] 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

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

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

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

[22] Simoncelli E.P., Adelson E.H. and Heeger D.J. (1991) Probability distributions of optical flow, *IEEE Proc. CVPR*, Maui pp. 310-315

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

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

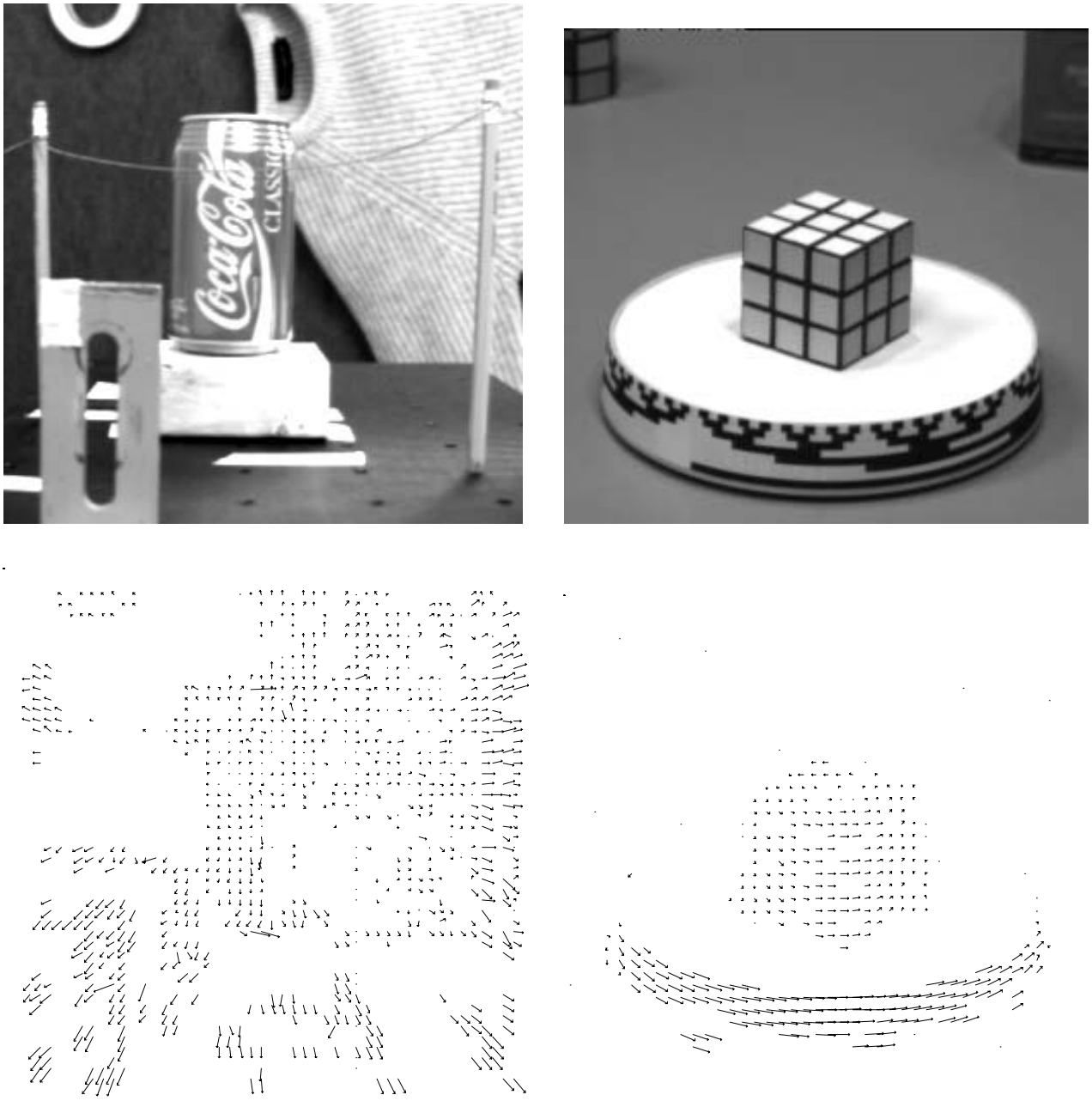


Figure 6. **Real Sequences:** *These show a single frame and a flow field estimated for that frame using the same technique as that for Figure 5. The tiny dots represent pixels at which the velocity estimate did not satisfy the eigenvalue threshold.*