Performance of Optical Flow Techniques

J.L. Barron
Dept. Computer Science
Univ. Western Ontario,
Canada

D.J. Fleet
Dept. Computing Science
Queen's University,
Canada

S.S. Beauchemin, T.A. Burkitt
Dept. Computer Science
Univ. Western Ontario
Canada

Abstract

While different optical flow methods continue to appear, there has been a lack of quantitative evaluation of existing methods. For a common set of image sequences, we report the results of regularly cited techniques, including instances of differential, matching, energy-based, and phase-based approaches.

1 Introduction

A fundamental problem in image sequence analysis is the measurement of optical flow (or image velocity). The goal is an approximation to the 2-d motion field (the projection of 3-d velocities onto the imaging surface) from image intensity [18, 32]. Image velocity can be used for many tasks including passive scene interpretation and autonomous, active exploration. However, tasks such as computation of egomotion and surface structure require accurate and dense velocity measurements; current techniques require that relative errors in the optical flow be less than 10% [6, 21]. Verri and Poggio [32] suggest that accurate estimates of the motion field are generally inaccessible.

Many methods for computing optical flow have been proposed - others continue to appear. Lacking however, is quantitative evaluation of existing methods, and direct comparisons. Kearney et al. [22] discussed sources of error with gradient-based methods. Little and Verri [23] compared properties of differential and matching methods, and reported some quantitative comparisons, but only on two relatively simple, synthetic test cases; the accuracy they reported was not encouraging, with average relative errors of 10%–20%, and average angular errors of 7°–12° in the best cases. Some of the methods reported here produce significantly better results.

This paper reports a comparison of widely cited optical flow methods. We implemented nine techniques (see [7]), of which six are reported here. They include instances of differential methods, region-based matching, energy-based and phase-based techniques: Horn and Schunck [19], Lucas and Kanade [24, 25], Uras et al. [31], Anandan [3, 4], Heeger [17], and Fleet and Jepson [12, 13]. Details on the implementations, in addition to those reported below can be found in [7]. The programs are available to those interested.

Most of these techniques can be viewed as several stages of processing. Typically there exists *1)* some degree of prefiltering or smoothing, *2)* the extraction of basic measurements, such as spatiotemporal derivatives or local correlation surfaces, and *3)* the integration of these measurements to produce a 2-d flow field, which often involves assumptions about the smoothness of the underlying flow field. It is important to test the performance of optical flow techniques with respect to each of these stages separately (where possible) [23]. Our selection of techniques for comparison was motivated in part by the desire to examine differences in initial measurement process, or in the method used to integrate measurements.

2 Optical Flow Techniques

We begin with a brief description of the techniques, and several of the implementation specifics.

2.1 Differential Techniques

Differential techniques compute velocity from spatiotemporal derivatives of image intensity, or filtered versions of the image (using low-pass or band-pass filters). The first instances used first-order derivatives, and were based on image translation [11, 19, 26], i.e.

$$I(\mathbf{x}, t) \;=\; I(\mathbf{x} - \mathbf{v}\,t, 0)\,, \tag{1}$$

where $\mathbf{v} = (v_1, v_2)^T$. From a Taylor expansion of (1) [19], or more generally from an assumption of conservation of intensity, $dI(\mathbf{x}, t)/dt = 0$, the *gradient constraint equation* is easily derived:

$$(\nabla I(\mathbf{x}, t))^T\mathbf{v} \;+\; I_t(\mathbf{x}, t) \;=\; 0\,, \tag{2}$$

where $\nabla I(\mathbf{x}, t) = (I_x(\mathbf{x}, t), I_y(\mathbf{x}, t))^T$. In effect, (2) yields the orientation and normal speed of spatial contours of constant intensity. But the two components of $\mathbf{v}$ in (2) are constrained by only one linear equation. Further constraints are therefore necessary.

Second-order differential methods [26, 30, 31], use second-order derivatives to constrain 2-d velocity:

$$\begin{aligned} I_{xx}(\mathbf{x}, t)v_1 + I_{yx}(\mathbf{x}, t)v_2 + I_{tx}(\mathbf{x}, t) \;&=\; 0 \\ I_{xy}(\mathbf{x}, t)v_1 + I_{yy}(\mathbf{x}, t)v_2 + I_{ty}(\mathbf{x}, t) \;&=\; 0 \end{aligned} \tag{3}$$

Equation (3) can be derived from (1), or from the conservation of $\nabla I(\mathbf{x}, t)$, $d\nabla I(\mathbf{x}, t)/dt = \mathbf{0}$. Strictly, this means that no first-order deformations of intensity (e.g., rotation or dilation) are permitted. To measure image velocity, assuming $d\nabla I(\mathbf{x}, t)/dt = \mathbf{0}$, the constraints in (3) may be used in isolation, or together with (2) to yield an over-determined system of linear equations. However, if the aperture problem prevails in a local neighbourhood (i.e. if intensity is effectively one-dimensional), then because of the sensitivity of numerical differentiation, 2^{nd}-order derivatives cannot be measured accurately enough to determine the tangential component of $\mathbf{v}$. Velocity estimates from 2^{nd}-order

0-8186-2855-3/92 $3.00 ©1992 IEEE

methods are therefore usually sparse and somewhat less accurate than estimates from 1^{st}-order methods.

Local estimates of component (normal) velocity can also be combined through space and time. In one approach, the local measurements in each neighbourhood are fit a single 2-d velocity field, e.g., a low-order polynomial model in v_1 and v_2, using least-squares fit or Hough transform [11, 22, 25, 29, 34]. Usually $\mathbf{v}(\mathbf{x})$ is taken to be constant, although linear models for $\mathbf{v}(\mathbf{x})$ have been used successfully. A second approach uses global smoothness constraints, in which the velocity field is defined implicitly in terms of the minimum of an energy functional defined over the image [19, 26].

These techniques assume that $I(\mathbf{x}, t)$ is differentiable. This suggests that temporal aliasing should be avoided, and that numerical differentiation must be done carefully. The often stated restrictions that gradient-based techniques require image velocities less than 1 pixel/frame and linear intensity, arise from the use of 2 frames, poor numerical differentiation, or input signals corrupted by temporal aliasing. With 2 frames, numerical differentiation is accomplished with a 1^{st}-order difference, which is accurate only when *1)* the input is highly over-sampled, or *2)* intensity structure is nearly linear. Assuming that aliasing cannot be avoided in image acquisition, one way to circumvent the problem is to apply differential techniques within a coarse-fine manner. Such extensions (e.g., [15]) are discussed in [7], but are beyond the scope of this paper.

Horn and Schunck: Horn and Schunck [19] combined the gradient constraint (2) with a global smoothness term to constrain the velocity field, minimizing

$$\int_D (I_x v_1 + I_y v_2 + I_t)^2 + \lambda^2(\| \nabla v_1 \|^2 + \| \nabla v_2 \|^2)\, d\mathbf{x} \qquad (4)$$

defined over a domain of interest D, where velocity $\mathbf{v}$ is a functions of $\mathbf{x}$, and λ reflects the influence of the smoothness term. Our implementation follows [19], with $\lambda = 100$.

Lucas and Kanade: Following Lucas and Kanade [25, 24], and others [2, 22, 28], we implemented a weighted least-squares fit of local measurements (2) to a constant model for $\mathbf{v}$ in each local region Ω by minimizing

$$\sum_{\mathbf{x}\in\Omega} W^2(\mathbf{x})\, [(\nabla I(\mathbf{x}, t))^T \mathbf{v} + I_t(\mathbf{x}, t)]^2 , \qquad (5)$$

where $W(\mathbf{x})$ denotes a window function (it gives more influence to constraints at the centre of the window). The solution to (5) is given by

$$A^T W^2 A\, \mathbf{v} \;=\; A^T W^2 \mathbf{b} \;. \qquad (6)$$

For points $\mathbf{x}_i \in \Omega$, $A = [\nabla I(\mathbf{x}_1), ..., \nabla I(\mathbf{x}_n)]^T$, $W = \mathrm{diag}[W(\mathbf{x}_1), ..., W(\mathbf{x}_n)]$, $\mathbf{b} = -(I_t(\mathbf{x}_1), ..., I_t(\mathbf{x}_n))^T$. Because $A^T W^2 A \in \mathcal{R}^{2\times 2}$, the solution to (6) can be given in closed form. Equations (5) and (6) may also be viewed as weighted least-squares estimates of $\mathbf{v}$ from estimates of normal velocities $v_n \mathbf{n}$; (5) is equivalent to

$$\sum_{\mathbf{x}\in\Omega} W^2(\mathbf{x})\, w^2(\mathbf{x})\, [\mathbf{v}^T \mathbf{n}(\mathbf{x}) + v_n(\mathbf{x})]^2 \qquad (7)$$

where coefficients $w^2(\mathbf{x})$ reflect our confidence in the normal velocity estimates; here, $w(\mathbf{x}) = \| \nabla I(\mathbf{x}, t) \|$.

Our implementation first smooths the image sequence with an isotropic spatiotemporal Gaussian with a standard deviation of 1.5 pixels-frames. This is necessary to attenuate the temporal aliasing and quantization present in image sequences. Derivatives were computed with 4-point central differences: the mask coefficients were $\frac{1}{12}(-1, 8, 0, -8, 1)$. Spatial neighbourhoods Ω were 5×5 pixels, and the window function $W(\mathbf{x})$ was separable and isotropic; its 1-d weights in the horizontal and vertical directions were $(0.0625, 0.25, 0.375, 0.25, 0.0625)$ as in [28]. The temporal support for the entire process is 15 frames.

Simoncelli et al. [28] present a Bayesian perspective of (5). Their *MAP* solution is similar to (6), and yields confidence measures for the velocity measurements. This result did alter the velocity estimates significantly, but it does suggest that unreliable estimates be identified using the eigenvalues of $A^T W^2 A$, $\lambda_1 \geq \lambda_2$, which depend on the magnitudes and range of orientations of the spatial gradients. If λ_1 and λ_2 are greater than a threshold τ, then $\mathbf{v}$ is computed from (6). If $\lambda_1 \geq \tau$ but $\lambda_2 < \tau$, then a normal velocity estimate is computed, and if $\lambda_1 < \tau$ no velocity is computed. Here we used $\tau = 1.0$. Confidence measures are discussed further in [7].

Uras, Girosi, Verri and Torre: The 2^{nd}-order technique considered here[1] is based on a local solution to (3). Following Uras et al. [31], (3) may be solved for $\mathbf{v}$ wherever the Hessian H of $I(\mathbf{x}, t)$ is non-singular. In practice, for reliability, they divide the image into regions 8×8 pixels wide. Within each region they select the 8 estimates that best satisfy the constraint $\| M \nabla I \| \ll \| \nabla I_t \|$, where $M \equiv (\nabla \mathbf{v})^T$. Of these they choose the one with the smallest condition number $\kappa(H)$ of the Hessian (3).

Our implementation presmooths the image sequence with a Gaussian kernel with a standard deviation of 3 in space and 1.5 in time. Beaudet operators [8] are used to compute derivatives. We found that without further constraints the results are inaccurate; the determinant $det(H)$ was therefore used to threshold the estimates, accepting estimates with $det(H) \geq 1.0$ or 2.0. This threshold is examined in [7].

2.2 Region-Based Matching

Accurate numerical differentiation may be impractical because of noise, because a small number of frames exist, or because of aliasing in the image acquisition process. Region-based matching [4, 10, 23] is then more appropriate than differential methods. Such approaches define velocity $\tilde{\mathbf{v}}$ as the shift $\mathbf{s}$ that yields the

[1] The technique of Nagel [26] is also discussed in [7].

best fit between image regions about $\mathbf{x}_0$ at time t_0, and at $\mathbf{x}_0 + \mathbf{s}$ at time t_1; that is, between image patches

$$\begin{aligned} I_0(\mathbf{x}) &\boxminus W(\mathbf{x}-\mathbf{x}_0)\, I(\mathbf{x}, t_0)\ , \\ I_1(\mathbf{x}; \mathbf{s}) &\equiv W(\mathbf{x}-\mathbf{x}_0)\, I(\mathbf{x}+\mathbf{s}, t_1)\ . \end{aligned} \qquad (8)$$

where $W(\mathbf{x})$ denotes a 2-d window function. To find the best match one might maximize a similarity measure (over $\mathbf{s}$) such as the normalized cross-correlation of $I_0(\mathbf{x})$ and $I_1(\mathbf{x}; \mathbf{s})$. Or one might minimize a distance measure, such as the sum-of-squared difference (SSD): $\| I_0(\mathbf{x}) - I_1(\mathbf{x}; \mathbf{s}) \|^2$. There is a close relationship between the SSD measure, the cross-correlation measure, and differential techniques [4, 13].

Anandan: The method considered here, based on a Laplacian pyramid and a coarse-to-fine SSD-based matching strategy, was reported by Anandan [3, 4]. Beginning at the coarsest level, displacements are computed to subpixel accuracy by finding the minimum of a quadratic approximation to the SSD surface (about the point yielding the lowest SSD value for integer shifts). Beaudet operators [8] were used for numerical differentiation to estimate the quadratic surface. Confidence measures, c_{min} and c_{max}, are derived from the principle curvatures of the SSD surface at the minimum. Anandan then uses a smoothness constraint on the resulting velocity estimates, taking c_{min} and c_{max} into account. Using an overlapped projection strategy, the smoothed displacement field is projected to the next finer level in the pyramid, and used as initial values for the matching process at this level. Matching and smoothing are performed at this and subsequent levels of the pyramid until level 0 (the image) has been processed, producing the final flow field. We used a Laplacian pyramid with two levels.

2.3 Energy-Based Methods

A third class of optical flow technique is based on the output of energy of velocity-tuned filters [2, 5, 17]. These methods are also called frequency-based owing the design of velocity-tuned filters in the Fourier domain [1, 13, 27, 33]. The Fourier transform of a translating 2-d pattern (1) is

$$\hat{I}(\mathbf{k}, \omega) \;=\; \hat{I}_0(\mathbf{k})\ \delta(\omega + \mathbf{v}^T\mathbf{k})\ , \qquad (9)$$

where $\hat{I}_0(\mathbf{k})$ is the Fourier transform of $I(\mathbf{x}, 0)$, and $\delta(k)$ is a Dirac delta function. This shows that all nonzero power associated with a translating 2-d pattern lies on a plane in frequency space. It has been shown that certain energy-based methods are equivalent to correlation-based methods [1, 27] and to the gradient-based approach of Lucas and Kanade [2].

Heeger: Here we consider the method developed by Heeger [17]. Using 12 Gabor filters at each spatial scale, the computation of image velocity is formulated as a least-squares fit of the filter energies to a plane in frequency space (based on an input model of white noise). Heeger first uses a Gaussian pyramid, each level of which is then band-pass filtered. Gabor filters are then applied to the scale-specific channels, from which velocities are measured. Level 0 (the image) should be used for speeds between 0–1.25 pixels/frame. Similarly, levels 1 and 2 should be used for speeds between 1.25–2.5 and 2.5–5 pixels/frame.

Our implementation uses three levels of the pyramid and chooses the $\mathbf{v}$ value from the pyramid level that best satisfies expected range of speeds for that level. The computation of $\mathbf{v}$ used Heeger's parallel method: a functional (relating the input model to expected filter energies) defined over some range of $\mathbf{v}$ is maximized to determine $\mathbf{v}$. A peak signifies a 2-d velocity estimate. A ridge, rather than a well-defined peak, signifies a normal component of velocity. Our implementation here is *ad hoc*.

2.4 Phase-Based Methods

We refer to our fourth class of methods as phase-based, because velocity is defined in terms of the phase behaviour of band-pass filter outputs. Zero-crossing techniques [16, 35] may be viewed as phase-based [13].

Fleet and Jepson: The use of phase was first proposed by Fleet and Jepson [12, 13]. The method defines component velocity in terms of the instantaneous motion of level phase contours in the output of band-pass velocity-tuned filters. Band-pass filters are used to decompose the input signal according to scale, speed and orientation. Each filter output is complex-valued and may be written as

$$R(\mathbf{x}, t) \;=\; \rho(\mathbf{x}, t)\ \exp[i\phi(\mathbf{x}, t)]\ , \qquad (10)$$

where $\rho(\mathbf{x}, t)$ and $\phi(\mathbf{x}, t)$ are the amplitude and phase parts of R. The component of $\mathbf{v}$ in the direction normal to level phase contours is given by $\mathbf{v}_n = v_n\mathbf{n}$, where normal speed and direction are $v_n = -\phi_t(\mathbf{x}, t)/\| \nabla\phi(\mathbf{x}, t) \|$, and $\mathbf{n} = \nabla\phi(\mathbf{x}, t)/\| \nabla\phi(\mathbf{x}, t) \|$, where $\nabla\phi(\mathbf{x}, t) = (\phi_x(\mathbf{x}, t), \phi_y(\mathbf{x}, t))^T$. In effect, this is a differential technique applied to phase rather than intensity. The phase derivatives are computed using the identity

$$\phi_x(\mathbf{x}, t) \;=\; \frac{\mathrm{Im}[R^*(\mathbf{x}, t)\ R_x(\mathbf{x}, t)]}{|R(\mathbf{x}, t)|^2}\ , \qquad (11)$$

where R^* is the complex conjugate of R. The full 2-d image velocity is then recovered locally by fitting a linear velocity field to the component velocities.

They compute component velocity from the output of each velocity-tuned channel, on the condition that the phase behaviour is stable. The key to detecting instability is the detection of singularity neighbourhoods with a constraint on instantaneous frequency and amplitude derivatives [20, 13]; we refer to this stability threshold as τ. A second constraint is also needed on amplitude to ensure a reasonable signal-to-noise ratio. With respect to the computation of 2-d velocity, constraints are placed on the conditioning of the linear system, and on the residual error. Like [12, 13], our implementation uses only a single scale tuned to a spatiotemporal wavelength of 4.25 pixels-frames. The entire temporal support is 21 frames, and we used the *same threshold values as those in* [12, 13].

3 Experimental Technique

Before reporting the results, we describe the image sequences, and the way that errors are measured.

3.1 Synthetic Image Sequences

The advantage of synthetic sequences is that we know the true motion field, and can therefore quantify performance. However, they are usually clean signals, with little occlusion, specularity, shadowing, transparency, etc.; and therefore these results should be taken as optimistic bounds on the expected errors with real inputs. We have collected results from several synthetic sequences [7]. Here we report results from:

3D Camera Motion and Planar Surface: Following [12] we used two sequences that simulate translational camera motion with respect to a textured planar surface (see Figure 1 (top)):

- In the **translating tree** sequence, the camera moves normal to its line of sight, with image velocities between (1.73, 0.0) and (2.3, 0.0);
- In the **diverging tree** sequence, 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.

Yosemite Fly-Through Sequence: The Yosemite sequence (courtesy of Lynn Quam) is a complex test case (see Figure 1 (middle)). The motion in the upper left is mainly divergent, the clouds translate to the right with at pixel/frame, while velocities in the lower left are about 4 pixels/frame. This sequence is challenging because of the range of velocities and the occluding edges between the mountains and at the horizon. There is severe aliasing in the lower left portion of the images however, causing most methods to produce poorer velocity measurements.

3.2 Real Image Sequences

In [7] we show results from several real image sequences, of which two (see Figure 1 (bottom)) are discussed here (obtained from the Database at Sarnoff Research Centre, courtesy of NASA-Ames and SRI International.). The **NASA** sequence is mainly dilational – the camera moves along it's line of sight toward the pop can. Image velocities are typically less than 1 pixel/frame. In the **SRI** sequence the camera translates perpendicular to its line of sight in front of clusters of trees. This is challenging because of the relatively poor resolution, the amount of occlusion, and the low contrast. Velocities are as large as 2 pixels/frame.

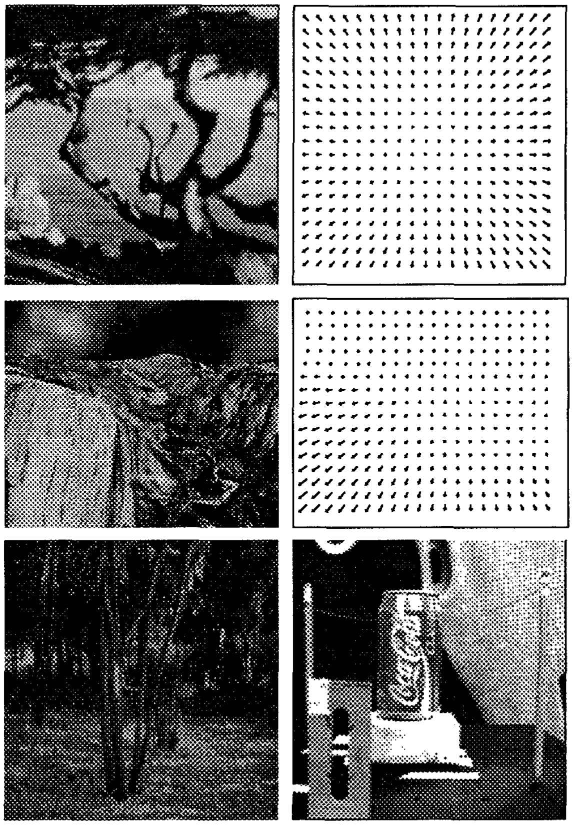

Figure 1: *(top) Image used for the* **translating** *and* **diverging** *tree sequences, and the motion field for the* **diverging** *tree sequence. (middle) A frame of the* **Yosemite** *sequence, and its motion field. (bottom) Frames from the* **SRI** *and the* **Nasa** *sequences.*

3.3 Error Measurement

Following [12] we use an angular measure of error: Let velocities $\mathbf{v} = (v_1, v_2)^T$ be written as 3-d direction vectors, $\vec{\mathbf{v}} \equiv (v_1^2 + v_2^2 + 1)^{-1/2} (v_1, v_2, 1)^T$. The error between the correct velocity $\vec{\mathbf{v}}_c$ and an estimate $\vec{\mathbf{v}}_e$ is

$$\psi_E = \arccos(\vec{\mathbf{v}}_c^T \, \vec{\mathbf{v}}_e) \ . \qquad (12)$$

This error measure is convenient because it handles large and very small speeds without the amplification inherent in a relative measure of vector differences. The 10% bound on acceptable velocity errors for egomotion and structure from 2-d motion corresponds to angular errors of roughly 2.5°. A similar measure is available for component velocity errors.

There are several ways in which error behaviour may be reported. Below we concentrate on angular error statistics (for synthetic sequences), and the computed flow fields (for the real sequences). In [7] we provide more details, and errors in component velocity.

4 Experimental Results

4.1 Synthetic Image Sequences

For the synthetic data, for which the correct motion fields are known, we report average angular error, standard deviation, the density of measurements, and the effects of confidence constraints. Although most of the results are self-evident from the tables, many of them deserve comments.

For Horn and Schunck's method, it is clear that the accuracy is generally poor. In order to test whether this was due to the crude numerical differentiation (1^{st}-order forward differences) in [19] we also we also implemented the approach with spatiotemporal Gaussian presmoothing and 4-point central-differences for numerical differentiation, like our implementation of Lucas and Kanade. This improved the results in most cases, however the accuracy remained noticably poorer than other techniques, presumably owing to the amount of smoothing imposed by the smoothness term in (4). The smoothing produced attractive flow fields, but appears to degrade the measurement accuracy.

By contrast, the estimates produced by the 1^{st}-order method of Lucas and Kanade are encouraging. Moreover, as discussed below, we also find that the eigenvalues of the normal matrix in (6) provide good measures of accuracy of the estimates. We find that changing this threshold allows us to select more accurate subsets of these estimates, accompanied of course by a reduction in the density of measurements.

Interestingly, the matching technique of Anandan produced reasonable results for the **translating** tree sequence, but poor results for the **diverging** tree sequence. We found in general that this technique was sensitive to dilation, in part owing to the smoothness constraint. However, we also ran the technique without the smoothing process and the results did not improve dramatically. Our attempts to threshold the results to obtain an accurate subset (based on values of c_{min} and c_{max}) were also unsuccessful. Accordingly we do not report thresholded results here.

Results for Uras et al. significantly improve with a threshold on $det(H)$. Other results in [7] show that $\kappa(H)$ also provides a measure of confidence. Although this technique is capable of reasonable accuracy, the results tend to be very sparse.

Heeger's results for the translating tree sequence used level 1 of the pyramid as the input speeds coincided with its velocity range of 1.25–2.5 pixels/frame. Level 0 was used for diverging tree sequence as most input speeds were below 1.25 pixel/frame. For the Yosemite sequence velocity estimates were computed at the three levels of the pyramid and then combined. Of the three, that velocity estimate from the level of the pyramid whose speed range was consistent with the true motion field, was chosen. These results are in the table. We also combined the pyramid levels without using the correct motion fields, choosing the estimate from the lowest pyramid level whose speed range was consistent with the estimate. This produced poorer results – errors of 13.75° ± 23.06°.

Technique	Average Error	Standard Deviation	Density
Horn and Schunck	33.40°	16.46°	100%
Lucas and Kanade ($\lambda_2 \geq 1.0$)	1.75°	1.43°	40.8%
Lucas and Kanade ($\lambda_2 \geq 5.0$)	1.12°	0.82°	13.6%
Uras et al. (unthresholded)	12.48°	17.52°	100%
Uras et al. ($det(H) \geq 2.0$)	6.49°	5.00°	23.6%
Anandan	4.54°	2.98°	100%
Heeger	4.79°	2.39°	13.8%
Fleet and Jepson ($\tau = 1.25$)	0.23°	0.20°	50.7%
Fleet and Jepson ($\tau = 2.5$)	0.36°	0.41°	76.0%

Technique	Average Error	Standard Deviation	Density
Horn and Schunck	9.85°	8.86°	100%
Lucas and Kanade ($\lambda_2 \geq 1.0$)	3.05°	2.53°	49.4%
Lucas and Kanade ($\lambda_2 \geq 5.0$)	2.32°	1.84°	24.8%
Uras et al. (unthresholded)	6.51°	7.00°	100%
Uras et al. ($det(H) \geq 2.0$)	4.00°	2.19°	38.6%
Anandan	8.23°	6.17°	100%
Heeger	4.95°	3.09°	73.8%
Fleet and Jepson ($\tau = 1.25$)	1.08°	0.52°	49.4%
Fleet and Jepson ($\tau = 2.5$)	1.24°	0.72°	64.3%

Technique	Average Error	Standard Deviation	Density
Horn and Schunck	22.58°	19.73°	100%
Lucas and Kanade ($\lambda_2 \geq 1.0$)	5.20°	9.45°	35.1%
Lucas and Kanade ($\lambda_2 \geq 5.0$)	3.55°	7.11°	8.8%
Uras et al. (unthresholded)	16.45°	21.02°	100.0%
Uras et al. ($det(H) \geq 1.0$)	5.97°	11.74°	23.4%
Uras et al. ($det(H) \geq 2.0$)	3.75°	3.44°	6.1%
Anandan	15.54°	13.46°	100%
Heeger	11.74°	19.0°	44.8%
Fleet and Jepson ($\tau = 1.25$)	4.95°	12.39°	30.6%
Fleet and Jepson ($\tau = 2.5$)	4.29°	11.24°	34.1%

Tables: *(top) Translating Tree Results; (middle) Diverging Tree Results; (bottom) Yosemite Results*

The phase-based method of Fleet and Jepson [12] produced the most accurate results. This is clear for the first two synthetic sequences, but not so clear for the Yosemite sequence. Interestingly, because only 15 frames were available in this case, we had to to increase the tuning frequency of the filters to reduce the width of support (from 21 to 15 frames). But this also pushes the pass-band region of the filters over the fold-over rate, causing greater sensitivity to aliasing and corruption at high frequencies. As a consequence, component velocity estimates were considerably worse, and the stability constraint [20] was not as effective at separating reliable from unreliable estimates. In fact, 2-d velocity estimates were improved slightly by relaxing this constraint, thereby creating larger systems of equations to average out some of the noise.

4.2 Real Image Data

The remaining figures show (miniature versions of) the flow fields for produced by five techniques, excluding the method of Horn and Schunck when applied to the two real image sequences. The Lucas and Kanade method was applied with a threshold of $\lambda_2 \geq 1$. Despite our questionable success with confidence thresh-

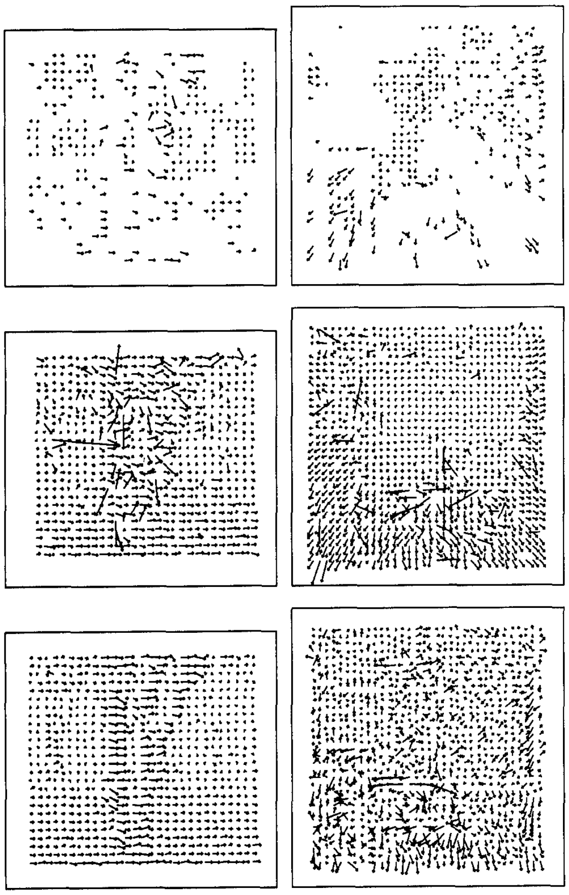

Figure 2: The left and right columns show computed flow fields from the **SRI** and **Nasa** sequences for the methods of: (top) **Lucas and Kanade** (threshold: $\lambda_2 \geq 1.0$); (middle) **Uras et al.** (no thresholding); and (bottom) **Anandan** (no thresholding).

olds for the methods of Uras et al., the result shown in Figure 2 are thresholded with $det(H) > 1.5$. For Anandan, the results are shown with no threshold used to separate more accurate subsets of the results. The results on the **NASA** sequence reflect this. The results shown for Heeger's method were based on 3 levels combined as discussed above. The method of Fleet and Jepson (Figure 3) was applied with the stability constraint $\tau = 1.25$.

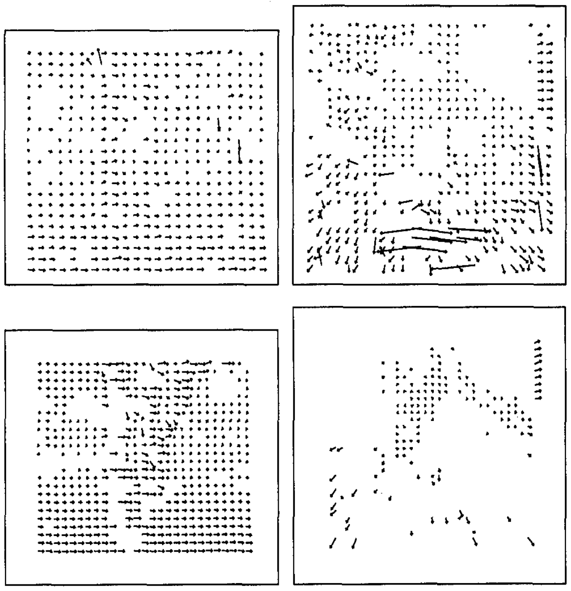

Figure 3: The left and right columns show computed flow fields from the **SRI** and **Nasa** sequences for the methods of: (top) **Heeger**; (bottom) **Fleet and Jepson** (stability constraint: $\tau = 1.25$).

5 Discussion

This paper compares the performance of optical flow techniques, emphasising measurement accuracy. The most accurate methods are the local differential approaches, where **v** is computed explicitly in terms of a locally constant or linear model. The phase-based method of Fleet and Jepson [12], and the first-order differential technique of Lucas and Kanade [25] were the most accurate. Techniques using global smoothness constraints appear to produce visually attractive flow fields, but in general they appear to be accurate enough for qualitative use only, and insufficient as precursors to the computation of egomotion and 3-d structure. Furthermore, we find that the sensitivities involved in solving Heeger's nonlinear minimization problem, finding SSD minima, or computing second derivatives are somewhat prohibitive.

One of the important aspects of this work, reported more fully in [7], concerns the use of confidence measures and thresholds. While many authors make no mention of confidence measures, we found that some form of confidence measure/threshold was crucial for all techniques in order to separate the inaccurate from the accurate. Not surprisingly, given the results above, we were most successful establishing reliable confidence measures for the methods of Fleet and Jepson, and Lucas and Kanade. In particular, they performed consistently well over all inputs. Although thresholds were used here to extract error statistics, in practice we imagine the confidence measure is not used to remove estimates, but rather is passed with the estimate to subsequent stages of processing.

But the accuracy of the measurements does not tell the entire story. Other factors, such as the computational efficiency, storage requirements, temporal duration of measurement support, and measurement density are also important. For example, it is clear from results above that although some methods appear to produce accurate estimates, their density of measurements is much lower than other methods with similar accuracy. This is evident with the 2^{nd}-order differential technique in relation to other methods.

Furthermore, each method comes with a price. The simplest method, conceptually and computationally, is that of Lucas and Kanade [25]. The most expensive methods were those requiring a large number of filters, notably the methods of Fleet and Jepson [12] and Heeger [17]. On the other hand, it is reasonable to expect that with the appropriate hardware, the filtering should cease to be a severe limitation, and all techniques could be implemented in real-time. Furthermore, all our convolution results were stored in floating point, and were not subsampled. More efficient encodings of the filter output should be possible with subsampling and quantization of the filter outputs as in [12] with only slightly less accurate measurements.

Similar comment apply to the six methods with respect to their the temporal support. We find that smoothing in time and space is crucial to most methods, especially those using numerical differentiation. Here, the numbers of frames required the methods of Fleet and Jepson [12], Lucas and Kanade [25], Uras et al. [31], and Heeger [17] were 21, 15, 12, and 7 respectively. The matching approach of Anandan requires only 2 frames. From this perspective it appears to do quite well, but it is difficult to compete with other techniques that exploit coherent structure of image intensity through time as well as space.

Finally, it is important to remember the conditions under which these tests were performed. We assumed that temporal aliasing was not be a severe problem, and that intensity (or filtered versions) were differentiable. As discussed earlier, if temporal aliasing is serious, then other approaches must be considered. Other important issues include occlusion and multiple velocities. All techniques had problems at occlusion boundaries, not well reflected in the confidence measures.

Acknowledgements: This work was supported in part by NSERC Canada and the Government of Ontario (through ITRC centres).

References

[1] Adelson, E. and Bergen, J.: *JOSA* A2: 284-299, 1985

[2] Adelson, E. and Bergen, J.: *Proc. IEEE Motion Workshop*, Charleston, pp. 151-156, 1986

[3] Anandan P.: PhD Dissertation, COINS TR 87-21, Univ. Massachusetts, Amherst, 1987

[4] Anandan P.: *IJCV* 2:238-310, 1989

[5] Barman, H., Haglund, L., Knutsson, H., Granlund, G.: *IEEE Motion Workshop*, Princeton, pp44-51, 1991

[6] Barron J., Jepson A. and Tsotsos J.: *IJCV* 5:239-269, 1990

[7] Barron J., Fleet D., Beauchemin S. and Burkitt T.: TR 299, Dept. Comp. Sci., Univ. Western Ontario, March, 1992

[8] Beaudet P.: *Proc. ICPR*, pp. 579-583, 1978

[9] Burt P. and Adelson E.: *IEEE Trans. Comm.*, 31:532-540, 1983

[10] Burt P., Yen C. and Xu X.: *Proc. IEEE CVPR*, Washington, pp. 246-252, 1983

[11] Fennema C. and Thompson W.: *CGIP* 9:301-315, 1979

[12] Fleet D. and Jepson A.: *IJCV* 5:77-104, 1990

[13] Fleet D., *Measurement of Image Velocity*, Kluwer Academic Publ., Norwell, 1992

[14] Girosi F. Verri A. and Torre V.: *Proc. IEEE Motion Workshop*, Irvine, pp. 116-124, 1989

[15] Glazer F., PhD Dissertation, COINS TR 87-02, Univ. Massachusetts, Amherst, MA, 1987

[16] Hildreth E.: *Proc. R. Soc. Lond.* B 221:189-220, 1984

[17] Heeger D.: *IJCV* 1:279-302, 1988

[18] Horn B.: *Robot Vision*, MIT Press, Cambridge, 1986

[19] Horn B. and Schunck B.: *AI* 17:185-204, 1981

[20] Jepson A. and Fleet D.: *IVC* 9: 338-343, 1991

[21] Jepson A. and Heeger D.: RBCV-TR-90-36, Dept. Computer Science, U. Toronto, 1990

[22] Kearney J., Thompson W. and Boley D.: *IEEE Trans. PAMI* 9:229-244, 1987

[23] Little J. and Verri A.: *IEEE Motion Workshop*, Irvine, pp. 173-180, 1989

[24] Lucas B.: PhD Dissertation, Dept. Computer Science, Carnegie-Mellon Univ., 1984

[25] Lucas, B. and Kanade, T.: *Proc. DARPA IU Workshop*, pp. 121-130, 1981

[26] Nagel H.: *CGIP* 21:85-117, 1983

[27] Santen J. van and Sperling G.: *JOSA* A2:300-321, 1985

[28] Simoncelli, E., Adelson E. and Heeger D.: *Proc. IEEE CVPR*, Maui, pp. 310-315, 1991

[29] Singh A.: *Proc. IEEE ICCV*, Osaka, pp. 168-177, 1990

[30] Tretiak O. and Pastor L.: *Proc. ICPR*, Montreal, pp. 20-22, 1984

[31] Uras S., Girosi F., Verri A. and Torre V.: *Biol. Cybern.* 60:79-97, 1988

[32] Verri A. and Poggio T.: *Proc. IEEE ICCV*, London, pp. 171-180, 1987

[33] Watson, A. and Ahumada, A.: *JOSA* A2:322-342, 1985

[34] Waxman A. and Wohn K.: *Int. J. Rob. Res.* 4:95-108, 1985

[35] Waxman A., Wu J. and Bergholm F.: *IEEE Proc. CVPR*, Ann Arbor, pp. 717-723, 1988