Performance of Optical Flow Techniques

J.L. Barron[1], D.J. Fleet[2] and S.S. Beauchemin[1]

[1]Dept. of Computer Science
University of Western Ontario
London, Ontario, N6A 5B7

[2]Dept. of Computing Science
Queen's University
Kingston, Ontario, K7L 3N6

Abstract

While different optical flow techniques continue to appear, there has been a lack of quantitative evaluation of existing methods. For a common set of real and synthetic image sequences, we report the results of a number of regularly cited optical flow techniques, including instances of differential, matching, energy-based and phase-based methods. Our comparisons are primarily empirical, and concentrate on the accuracy, reliability and density of the velocity measurements; they show that performance can differ significantly among the techniques we implemented.

1 Introduction

Without doubt, a fundamental problem in the processing of image sequences is the measurement of optical flow (or image velocity). The goal is to compute an approximation to the 2-d motion field – a projection of the 3-d velocities of surface points onto the imaging surface – from spatiotemporal patterns of image intensity [30, 58]. Once computed, the measurements of image velocity can be used for a wide variety of tasks ranging from passive scene interpretation to autonomous, active exploration. Of these, tasks such as the inference of egomotion and surface structure require that velocity measurements be accurate and dense, providing a close approximation to the 2-d motion field. Current techniques require that relative errors in the optical flow be less than 10% [9, 35]. Verri and Poggio [58] have suggested that accurate estimates of the 2-d motion field are generally inaccessible due to inherent differences between the 2-d motion field and intensity variations, while others (e.g. [4]) argue that the measurement of optical flow is an ill-posed problem. For these reasons it has been suggested that only qualitative information can be extracted.

Many methods for computing optical flow have been proposed – others continue to appear. Lacking, however, are quantitative evaluations of existing methods and direct comparisons on a single set of inputs. Kearney et al. [36] discussed sources of error with gradient-based methods. Little and Verri [38] 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. More recently, Willick and Yang [62] have examined the merits of the gradient constraint used by Horn and Schunck [31] compared to the constraints suggested by Schunck [49, 50] and Nagel [44]. Of these three, they argue that the original gradient constraint is superior. This paper reports a comparison of widely cited optical flow methods. We implemented nine techniques including instances of differential methods, region-based matching, energy-based and phase-based techniques, namely those of Horn and Schunck [31], Lucas and Kanade [39, 40], Uras et al. [57], Nagel [43], Anandan [5, 6], Singh [53, 54], Heeger [29], Waxman et al. [61] and Fleet and Jepson [19, 22].

Despite their differences, many of these techniques can be viewed conceptually in terms of three stages of processing:

1. prefiltering or smoothing with low-pass/band-pass filters in order to extract signal structure of interest and to enhance the signal-to-noise ratio,
2. the extraction of basic measurements, such as spatiotemporal derivatives (to measure normal components of velocity) 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.

Our selection of techniques for comparison was motivated in part by a desire to examine properties of these individual stages; for example, we have two first-order differential techniques that differ only in the method used to integrate measurements. Where applicable, we also report results concerning the measurement of normal velocity since there is growing interest in the use of normal velocity, thereby side-stepping some of the assumptions inherent in current methods for integrating measurements to find 2-d velocity [3, 4, 9, 15, 32, 46].

We have used both real and synthetic image sequences to test the techniques. In both cases however, we have chosen sequences that are not severely corrupted by spatial or temporal aliasing. This enables us to test basic implementations of differential methods and matching methods on the same data without the complexities of hierarchical coarse-fine control and warping techniques. For example, we do not consider *stop-and-shoot* sequences [17].

This paper concentrates on the accuracy and density of velocity estimates produced by the nine methods. Confidence measures have been used to extract subsets of estimates for which we report error statistics. While confidence measures are rarely addressed in the literature, we find that they are crucial to the successful use of all techniques. Thus we have also examined the use of several different confidence measures.

2 Optical Flow Techniques

We begin with a brief description of the different techniques, and several of the implementation specifics. Although most of the important issues are addressed here, the interested reader should consult the original papers for further details. In addition, our source code and our image sequences are available via anonymous ftp from `ftp.csd.uwo.ca` in the directory `/pub/vision`.

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 [18, 31, 42], i.e.

$$I(\mathbf{x}, t) \;=\; I(\mathbf{x} - \mathbf{v}\,t, 0)\ , \tag{2.1}$$

where $\mathbf{v} = (u, v)^T$. From a Taylor expansion of (2.1) [31] or more generally from an assumption that intensity is conserved, $dI(\mathbf{x}, t)/dt = 0$, the *gradient constraint equation* is easily derived:

$$\nabla I(\mathbf{x}, t) \cdot \mathbf{v} \;+\; I_t(\mathbf{x}, t) \;=\; 0\ , \tag{2.2}$$

where $I_t(\mathbf{x}, t)$ denotes the partial time derivative of $I(\mathbf{x}, t)$, $\nabla I(\mathbf{x}, t) = (I_x(\mathbf{x}, t), I_y(\mathbf{x}, t))^T$, and $\nabla I \cdot \mathbf{v}$ denotes the usual dot product. In effect, (2.2) yields the normal component of motion of spatial contours of constant intensity, $\mathbf{v}_n = s\mathbf{n}$. The normal speed s and the normal direction $\mathbf{n}$ are given by

$$s(\mathbf{x}, t) \;=\; \frac{-\,I_t(\mathbf{x}, t)}{\|\,\nabla I(\mathbf{x}, t)\,\|}\ , \qquad \mathbf{n}(\mathbf{x}, t) \;=\; \frac{\nabla I(\mathbf{x}, t)}{\|\,\nabla I(\mathbf{x}, t)\,\|}\ . \tag{2.3}$$

There are two unknown components of $\mathbf{v}$ in (2.2), constrained by only one linear equation. Further constraints are therefore necessary to solve for both components of $\mathbf{v}$.

Second-order differential methods use second-order derivatives (the Hessian of I) to constrain 2-d velocity [42, 43, 56, 57]:

$$\begin{bmatrix} I_{xx}(\mathbf{x},\, t) & I_{yx}(\mathbf{x},\, t) \\ I_{xy}(\mathbf{x},\, t) & I_{yy}(\mathbf{x},\, t) \end{bmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} + \begin{pmatrix} I_{tx}(\mathbf{x},\, t) \\ I_{ty}(\mathbf{x},\, t) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} . \tag{2.4}$$

Equation (2.4) can be derived from (2.1) or from the conservation of $\nabla I(\mathbf{x},\, t)$, $d\nabla I(\mathbf{x},\, t)/dt = \mathbf{0}$. Strictly speaking, the conservation of $\nabla I(\mathbf{x},\, t)$ implies that first-order deformations of intensity (e.g. rotation or dilation) should not be present. This is therefore a stronger restriction than (2.2) on permissible motion fields. To measure image velocity, assuming $d\nabla I(\mathbf{x},\, t)/dt = \mathbf{0}$, the constraints in (2.4) may be used in isolation or together with (2.2) to yield an over-determined system of linear equations [23, 47]. 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 usually be measured accurately enough to determine the tangential component of $\mathbf{v}$. As a consequence, velocity estimates from 2^{nd}-order methods are often assumed to be sparser and less accurate than estimates from 1^{st}-order methods.

Another way to constrain $\mathbf{v}(\mathbf{x})$ is to combine local estimates of component velocity and/or 2-d velocity through space and time, thereby producing more robust estimates of $\mathbf{v}(\mathbf{x})$ [53]. There are two common methods to accomplish this: The first method fits the measurements in each neighbourhood to a local model for 2-d velocity (e.g. a low-order polynomial model), using least-squares minimization or a Hough transform [18, 36, 40, 53, 60]. Usually $\mathbf{v}(\mathbf{x})$ is taken to be constant, although linear models for $\mathbf{v}(\mathbf{x})$ have been used successfully [60, 19]. The second approach uses global smoothness constraints (regularization) in which the velocity field is defined implicitly in terms of the minimum of a functional defined over the image [31, 42, 43, 45].

Of course, one requirement of differential techniques is that $I(\mathbf{x},\, t)$ must be differentiable. This implies that temporal smoothing at the sensors is needed to avoid aliasing and that numerical differentiation must be done carefully. The often stated restrictions that gradient-based techniques require image intensity to be nearly linear, with velocities less than 1 pixel/frame, arise from the use of 2 frames, poor numerical differentiation or input signals corrupted by temporal aliasing. For example, with 2 frames, derivatives are estimated using 1^{st}-order backward differences, which are accurate only when *1)* the input is highly over-sampled or *2)* intensity structure is nearly linear. When aliasing cannot be avoided in image acquisition, one way to circumvent the problem is to apply differential techniques in a coarse-fine manner, for which estimates are first produced at coarse scales where aliasing is assumed to be less severe, with velocities less than 1 pixel/frame. These estimates are then used as initial guesses to warp finer scales to compensate for larger displacements. Such extensions are not examined in detail here.

This paper reports results from *four* differential techniques; they include first-order and second-order constraints, as well as local and global methods of combining the local constraints. We found that all these techniques, as described in the literature, require some confidence measure as a means of separating reliable from unreliable measurements. Although we have used such thresholds to obtain the results reported below, it is important to note that they were not taken from the original literature in all cases, but rather are a first attempt on our part to improve the accuracy of the measurements. They are discussed below and in more detail in Section 4.5, with quantitative results shown in Appendix G.

Horn and Schunck

Horn and Schunck [31] combined the gradient constraint (2.2) with a global smoothness term to constrain the estimated velocity field $\mathbf{v}(\mathbf{x}, t) = (u(\mathbf{x}, t), v(\mathbf{x}, t))$, minimizing

$$\int_D (\nabla I \cdot \mathbf{v} + I_t)^2 + \lambda^2(\| \nabla u \|_2^2 + \| \nabla v \|_2^2)\, d\mathbf{x} \tag{2.5}$$

defined over a domain D, where the magnitude of λ reflects the influence of the smoothness term. We used $\lambda = 0.5$ instead of $\lambda = 100$ as suggested in [31] because it produced better results in most of our test cases. Iterative equations are used to minimize (2.5) and obtain image velocity:

$$\begin{aligned} u^{k+1} &= \bar{u}^k - \frac{I_x[I_x\bar{u}^k + I_y\bar{v}^k + I_t]}{\alpha^2 + I_x^2 + I_y^2} \\ v^{k+1} &= \bar{v}^k - \frac{I_y[I_x\bar{u}^k + I_y\bar{v}^k + I_t]}{\alpha^2 + I_x^2 + I_y^2}\ , \end{aligned} \tag{2.6}$$

where k denotes the iteration number, u^0 and v^0 denote initial velocity estimates which are set to zero, and $\bar{u}^k$ and $\bar{v}^k$ denote neighbourhood averages of u^k and v^k. We use at most 100 iterations in all testing below.

The original method described in [31] used first-order differences to estimate intensity derivatives. Because this is a relatively crude form of numerical differentiation and can be the source of considerable error, we also implemented the method with spatiotemporal presmoothing and 4-point central differences for differentiation (with mask coefficients $\frac{1}{12}(-1, 8, 0, -8, 1)$). We used a spatiotemporal Gaussian prefilter with a standard deviation of 1.5 pixels in space and 1.5 frames in time (1.5 pixels-frames), sampled out to three standard deviations. Results from both the original and our modified method are reported below.

Lucas and Kanade

Following Lucas and Kanade [40, 39] and others [2, 36, 51, 52], we implemented a weighted least-squares (LS) fit of local first-order constraints (2.2) to a constant model for $\mathbf{v}$ in each small spatial neighbourhood Ω by minimizing

$$\sum_{\mathbf{x}\in\Omega} W^2(\mathbf{x})\,[\nabla I(\mathbf{x}, t) \cdot \mathbf{v} \,+\, I_t(\mathbf{x}, t)]^2\ , \tag{2.7}$$

where $W(\mathbf{x})$ denotes a window function that gives more influence to constraints at the centre of the neighbourhood than those at the periphery. The solution to (2.7) is given by

$$A^T W^2 A\,\mathbf{v} \;=\; A^T W^2 \mathbf{b}\ , \tag{2.8}$$

where, for n points $\mathbf{x}_i \in \Omega$ at a single time t,

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

The solution to (2.8) is $\mathbf{v} = [A^T W^2 A]^{-1}\, A^T W^2 \mathbf{b}$, which is solved in closed form when $A^T W^2 A$ is nonsingular, since it is a 2×2 matrix:

$$A^T W^2 A \;=\; \begin{bmatrix} \sum W^2(\mathbf{x}) I_x^2(\mathbf{x}) & \sum W^2(\mathbf{x}) I_x(\mathbf{x}) I_y(\mathbf{x}) \\ \sum W^2(\mathbf{x}) I_y(\mathbf{x}) I_x(\mathbf{x}) & \sum W^2(\mathbf{x}) I_y^2(\mathbf{x}) \end{bmatrix}, \tag{2.9}$$

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

Equations (2.7) and (2.8) may also be viewed as weighted least-squares estimates of $\mathbf{v}$ from estimates of normal velocities $\mathbf{v}_n = s\mathbf{n}$; i.e. (2.7) is equivalent to

$$\sum_{\mathbf{x}\in\Omega} W^2(\mathbf{x})\, w^2(\mathbf{x})\; [\mathbf{v}\cdot\mathbf{n}(\mathbf{x}) \;-\; s(\mathbf{x})]^2 \qquad (2.10)$$

where the 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 a spatiotemporal Gaussian filter with a standard deviation of 1.5 pixels-frames. This helps attenuate temporal aliasing and quantization effects in the input. Derivatives were computed with 4-point central differences with mask coefficients $\frac{1}{12}(-1, 8, 0, -8, 1)$. Spatial neighbourhoods Ω were 5×5 pixels, and the window function $W^2(\mathbf{x})$ was separable and isotropic; its effective 1-d weights are $(0.0625, 0.25, 0.375, 0.25, 0.0625)$ as in [51]. The temporal support for the entire process was 15 frames. In a more recent implementation, Fleet and Langley [21] have replaced the FIR filters with IIR recursive filters and temporally recursive estimation. This method requires only three frames of storage, delays of only 2 or 3 frames, and yields results of similar accuracy.

Simoncelli et al. [51, 52] present a Bayesian perspective of (2.7). They model the gradient constraint equation (2.2) using Gaussianly distributed errors on gradient measurements, and a Gaussianly distributed prior on velocity $\mathbf{v}$. The resulting *maximum a posteriori* solution is similar to (2.8), and yields a covariance matrix for the velocity estimates. We found that this modification does not change the accuracy 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 of the spatial gradients, and their range of orientations. Although Simoncelli et al. suggested using the sum of eigenvalues, we found that the smallest eigenvalue alone was somewhat more reliable. Therefore, in our implementation, if both λ_1 and λ_2 are greater than a threshold τ, then $\mathbf{v}$ is computed from (2.8). 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. Unless stated otherwise, we used $\tau = 1.0$. Interestingly, this also gives us two ways of computing normal velocities: *1)* from the gradient constraint (2.3) and *2)* from this LS minimization. Results from both methods are given below.

Nagel

Nagel was one of the first to use second-order derivatives to measure optical flow [42, 43, 45]. Like Horn and Schunck, the basic measurements are integrated using a global smoothness constraint. As an alternative to the constraint in (2.5), Nagel suggested an *oriented-smoothness* constraint in which smoothness is not imposed across steep intensity gradients (edges) in an attempt to handle occlusion [42, 43, 45]. The problem is formulated as the minimization of the functional

$$\int\!\!\int (\nabla I^T\mathbf{v} + I_t)^2 + \frac{\alpha^2}{\|\nabla I\|_2^2 + 2\delta}\left[(u_x I_y - u_y I_x)^2 + (v_x I_y - v_y I_x)^2 + \delta(u_x^2 + u_y^2 + v_x^2 + v_y^2)\right] dx dy. \qquad (2.11)$$

Minimizing (2.11) with respect to $\mathbf{v}$ attenuates the variation of the flow, $\nabla\mathbf{v}$, in the direction perpendicular to the gradient. As suggested in [43] we fix $\delta = 1.0$. [1] Also, unless otherwise stated we set $\alpha = 0.5$.

[1] Smaller values of δ were tested but they produced numerical instabilities unless greater blurring was used.

Appendix A shows that a solution to this minimization may be expressed with Gauss-Seidel iterations as follows:

$$\begin{aligned} u^{k+1} &= \xi(u^k) - \frac{I_x(I_x\xi(u^k) + I_y\xi(v^k) + I_t)}{{I_x}^2 + {I_y}^2 + \alpha^2} \,, \\ v^{k+1} &= \xi(v^k) - \frac{I_y(I_x\xi(u^k) + I_y\xi(v^k) + I_t)}{{I_x}^2 + {I_y}^2 + \alpha^2} \,. \end{aligned} \tag{2.12}$$

In these equations, k represents the iteration number, and $\xi(u^k)$ and $\xi(v^k)$ are given by

$$\begin{aligned} \xi(u^k) &= \bar{u}^k - 2I_xI_yu_{xy} - \mathbf{q}^T(\nabla u^k) \\ \xi(v^k) &= \bar{v}^k - 2I_xI_yv_{xy} - \mathbf{q}^T(\nabla v^k) \end{aligned}$$

where

$$\mathbf{q} = \frac{1}{I_x^2 + I_y^2 + 2\delta} \nabla I^T \left[\begin{pmatrix} I_{yy} & -I_{xy} \\ -I_{xy} & I_{xx} \end{pmatrix} + 2 \begin{pmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{pmatrix} W \right] ,$$

${u_{xy}}^k$ and ${v_{xy}}^k$ denote estimates of the partial derivatives of $\mathbf{v}^k$, $\bar{u}^k$ and $\bar{v}^k$ are local neighbourhood averages of u^k and v^k and W is the weight matrix

$$W = (I_x^2 + I_y^2 + 2\delta)^{-1} \begin{pmatrix} I_y^2 + \delta & -I_xI_y \\ -I_xI_y & I_x^2 + \delta \end{pmatrix} .$$

In our implementation, all velocities are set to zero initially. The image sequence is presmoothed with a Gaussian kernel with a standard deviation of 1.5 pixels in space and time[2] Intensity derivatives were computed using 4-point central-difference operators, cascaded in different directions to get the second derivatives. First-order velocity derivatives were computed using 2-point central-difference kernels, $\frac{1}{2}(1, 0, -1)$, and 2^{nd} order derivatives were computed as cascades of 1^{st} order derivatives. We used 100 iterations to obtain the results reported here. Further details of our implementation are given in Appendix A.

Uras, Girosi, Verri and Torre

The other 2^{nd}-order technique considered here is based on a local solution to (2.4). Following Uras et al. [57], (2.4) may be solved for $\mathbf{v}$ wherever the Hessian H of $I(\mathbf{x}, t)$ is nonsingular. In practice, for robustness, they divide the image into 8×8 pixel regions. From each region they select the 8 estimates that best satisfy the constraint $\parallel M\nabla I \parallel \ll \parallel \nabla I_t \parallel$, where $M \equiv (\nabla \mathbf{v})^T$. Of these they choose the estimate with the smallest condition number $\kappa(H)$ of the Hessian (2.4) as the velocity for the entire 8×8 region.

Our implementation presmooths the image sequence with a Gaussian kernel with a standard deviation of 3 pixels in space and 1.5 frames in time.[3] Derivatives of $I(\mathbf{x}, t)$ and $\mathbf{v}$ were computed using 4-point central-difference operators, cascaded in different directions to get the second derivatives. Although Uras et al. suggest that $\kappa(H)$ be used as a confidence measure for the velocity estimates, we found that the determinant $det(H)$ (the spatial Gaussian curvature of the smoothed input) is more reliable, as is discussed in Section 4.5. Therefore, when reporting error statistics, we extract subsets of velocity estimates using the constraint: $det(H) \geq 1.0$ (unless stated otherwise).

[2] The real image sequences required more smoothing with a standard deviation of 3.0 in space instead of 1.5 to obtain good results. The synthetic test data produced better results with less smoothing.

[3] In the original paper [57] the authors used standard deviations of 5 in space and 1 frame in time.

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. In these cases differential approaches may be inappropriate and it is natural to turn to region-based matching [24, 6, 13, 37, 38]. Such approaches define velocity $\mathbf{v}$ as the shift $\mathbf{d} = (d_x, d_y)$ that yields the best fit between image regions at different times. Finding the best match amounts to maximizing a similarity measure (over $\mathbf{d}$), such as the normalized cross-correlation or minimizing a distance measure, such as the sum-of-squared difference (SSD):

$$\begin{aligned} SSD_{1,2}(\mathbf{x};\, \mathbf{d}) &= \sum_{j=-n}^{n} \sum_{i=-n}^{n} W(i,j)\, [I_1(\mathbf{x} + (i,j)) - I_2(\mathbf{x} + \mathbf{d} + (i,j))]^2 \\ &= W(\mathbf{x}) * [I_1(\mathbf{x}) - I_2(\mathbf{x} + \mathbf{d})]^2 \ , \end{aligned} \tag{2.13}$$

where W denotes a discrete 2-d window function, and $\mathbf{d} = (d_x, d_y)$ take on integer values.

There is a close relationship between the SSD distance measure, the cross-correlation similarity measure, and differential techniques. Minimizing the SSD distance amounts to maximizing the integral of product term $I_1(\mathbf{x})I_2(\mathbf{x} + \mathbf{d})$. Also, the difference in (2.13) can be viewed as a window-weighted average of a first-order approximation to the temporal derivative of $I(\mathbf{x}, t)$.

Anandan

The first matching technique considered here, reported by Anandan [5, 6], is based on a Laplacian pyramid and a coarse-to-fine SSD-based matching strategy. The Laplacian pyramid [12] allows the computation of large displacements between frames and helps to enhance image structure, such as edges, that is often thought to be important.

We begin at the coarsest level where displacements are assumed to be 1 pixel/frame or less. SSD minima are first located to pixel accuracy by computing (i.e. sampling) SSD values in 3×3 a search space (i.e. d_x and d_y take values -1, 0 and 1 pixel/frame), using a 5×5 Gaussian for $W(\mathbf{x})$. Subpixel displacements are then computed by finding the minimum of a quadratic approximation to the SSD surface (about the minimum SSD value found with integer displacements $\mathbf{d}$). As suggested by Anandan, Beaudet operators [11] were used to estimate the quadratic surface parameters. Confidence measures, c_{min} and c_{max}, are derived from the principle curvatures, C_{min} and C_{max}, of the SSD surface at the minimum:

$$c_{max} = \frac{C_{max}}{k_1 + k_2 S_{min} + k_3 C_{max}} \ , \quad c_{min} = \frac{C_{min}}{k_1 + k_2 S_{min} + k_3 C_{min}} \ , \tag{2.14}$$

where k_1, k_2 and k_3 are normalization constants, and S_{min} is the SSD value at the minima. Anandan uses $k_1 = 150$, $k_2 = 1$ and $k_3 = 0$ (see page 130 in [5]).

Anandan also employs a smoothness constraint on the velocity estimates, taking c_{min} and c_{max} into account, by then minimizing

$$\int\!\!\int (u_x^2 + u_y^2 + v_x^2 + v_y^2) + c_{max}(\mathbf{v} \cdot \mathbf{e}_{max} - \mathbf{v}_0 \cdot \mathbf{e}_{max})^2 + c_{min}(\mathbf{v} \cdot \mathbf{e}_{min} - \mathbf{v}_0 \cdot \mathbf{e}_{min})^2 \tag{2.15}$$

where $\mathbf{e}_{max}$ and $\mathbf{e}_{min}$ are the directions of maximum and minimum curvature of the SSD surface at the minimum, and $\mathbf{v}_0$ denotes the displacements propagated from the higher level in the pyramid. Using Gauss-Seidal iterations Anandan derives the following equation

$$\mathbf{v}^{k+1} = \bar{\mathbf{v}}^k + \frac{c_{max}}{c_{max} + 1}[(\mathbf{v}_0 - \bar{\mathbf{v}}^k) \cdot \mathbf{e}_{max}]\mathbf{e}_{max} + \frac{c_{min}}{c_{min} + 1}[(\mathbf{v}_0 - \bar{\mathbf{v}}^k) \cdot \mathbf{e}_{min}]\mathbf{e}_{min} \ , \tag{2.16}$$

where $\bar{\mathbf{v}}^k$ is the neighbourhood average of $\mathbf{v}^k$ computed using mask

$$\frac{1}{4}\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

Initially, $\bar{\mathbf{v}}^0$ is set to $\mathbf{v}_0$. Anandan allows 10 iterations to achieve convergence.

Matching and smoothing are performed at each level of the Laplacian pyramid. When moving from coarser to finer levels the initial 3×3 SSD search area is determined by projecting the coarser level estimate at each pixel to all pixels in a 4×4 region at the next finer level so that each pixel at the finer level has 4 initial guesses. The SSD search area is then the union of the 3×3 areas centered at each of the 4 initial displacements. We used a Laplacian pyramid with two or three levels depending on the range of speeds in the image sequence we examine.[4] We attempted to extract subsets of estimates using a threshold on the confidence measures suggested by Anandan, i.e. c_{min} and c_{max}. However, as discussed below, we did not find such measures to be reliable.

Singh

We also implemented Singh's two-stage matching method [53, 54]. The first stage is based on the computation of SSD values with three adjacent band-pass filtered images,[5] I_{-1}, I_0 and I_{+1}:

$$SSD_0(\mathbf{x}, \mathbf{d}) \;=\; SSD_{0,1}(\mathbf{x}, \mathbf{d}) + SSD_{0,-1}(\mathbf{x}, -\mathbf{d}) \ , \tag{2.17}$$

where $SSD_{i,j}$ is given in (2.13). Adding 2-frame SSD surfaces to form SSD_0 tends to average out spurious SSD minima due to noise or periodic texture. Singh then converts SSD_0 into a probability distribution using

$$R_c(\mathbf{d}) \;=\; e^{-k\ SSD_0} \tag{2.18}$$

where $k = -\ln(0.95)/(\min(SSD_0))$.[6] The subpixel velocity $\mathbf{v_c} = (u_c, v_c)$ is then computed as the mean of this distribution (averaged over the integer displacements $\mathbf{d}$:

$$u_c \;=\; \frac{\sum R_c(\mathbf{d}) d_x}{\sum R_c(\mathbf{d})} \ , \quad \text{and} \quad v_c \;=\; \frac{\sum R_c(\mathbf{d}) d_y}{\sum R_c(\mathbf{d})} \ . \tag{2.19}$$

As this only works well when the $R_c(\mathbf{d})$ is nearly symmetrical about the true velocity, Singh suggests a coarse-to-fine strategy using a Laplacian pyramid as in [5, 6] so that the effective SSD surface is centered at the true displacement. This also allows for large speeds and produces computational savings. Finally, Singh suggests the eigenvalues of the inverse covariance matrix as measures of confidence, where the covariance matrix is given by

$$S_c \;=\; \frac{1}{\sum R_c(\mathbf{d})} \begin{pmatrix} \sum R_c(\mathbf{d})(d_x - u_c)^2 & \sum R_c(\mathbf{d})(d_x - u_c)(d_y - v_c) \\ \sum R_c(\mathbf{d})(d_x - u_c)(d_y - v_c) & \sum R_c(\mathbf{d})(d_y - v_c)^2 \end{pmatrix} . \tag{2.20}$$

In our implementation of step 1 we use a single resolution: The SSD surface is computed for a wide range of integer displacements, with $-2N \le d_x, d_y \le 2N$, where N is as large as 4 pixels. Like Singh we use a uniform window W in (2.13) of width 5 (unless specified otherwise). From this $(4N+1) \times (4N+1)$ SSD surface we extract a $(2N+1) \times (2N+1)$ subregion about the

[4] We tested our implementation of Anandan's algorithm on the same Mandrill set of images he used (page 132 in [5]). This involves a translation of the second image by $\mathbf{V} = (7, 5)$. Our results were almost identical to those reported in [5].

[5] With impulse response $\delta(\mathbf{x}) - G(\mathbf{x})$ where $\delta(\mathbf{x})$ is a Dirca delta function and $G(\mathbf{x})$ is an isotropic Gaussian with standard deviation 1.0.

[6] When $\min(SSD_0) = 0$ we choose the smallest non-zero value of SSD_0 to compute k.

minimum[7] found within the central portion of the original search window (i.e. for displacements between $-N$ and N). Our goal was to extract the SSD surface sampled symmetrically about the minimum, to better satisfy the symmetry assumption for the distribution that was mentioned above. For $N = 4$ this yields a 9×9 SSD patch about the integer velocity from within the 17×17 original SSD surface.

The second step in the algorithm propagates velocity using neighbourhood constraints. That is, it is assumed that a weighted least-squares velocity estimate $\mathbf{v}_n = (u_n, v_n)$ could be derived from velocities $\mathbf{v}_i = (u_i, v_i)$ in its local $(2w+1) \times (2w+1)$ neighbourhood as follows:

$$u_n = \frac{\sum_i R_n(\mathbf{v}_i) u_i}{\sum_i R_n(\mathbf{v}_i)} \ , \qquad v_n = \frac{\sum_i R_n(\mathbf{v}_i) v_i}{\sum_i R_n(\mathbf{v}_i)} \ . \tag{2.21}$$

where $R_n(\mathbf{v}_i)$ is a Gaussian function of the distance between the centre of the neighbourhood and the location of the estimate $\mathbf{v}_i$. Although Singh used $w = 1$, we found better results with $w = 2$. The corresponding covariance matrix is

$$S_n = \frac{1}{\sum_i R_n(\mathbf{v}_i)} \begin{pmatrix} \sum_i R_n(\mathbf{v}_i)(u_i - u_n)^2 & \sum_i R_n(\mathbf{v}_i)(u_i - u_n)(v_i - v_n) \\ \sum_i R_n(\mathbf{v}_i)(u_i - u_n)(v_i - v_n) & \sum_i R_n(\mathbf{v}_i)(v_i - v_n)^2 \end{pmatrix} . \tag{2.22}$$

The final velocity estimate, $\mathbf{v} = (u, v)$, is chosen to minimize

$$\int\int (\mathbf{v} - \mathbf{v}_n)^T S_n^{-1} (\mathbf{v} - \mathbf{v}_n) + (\mathbf{v} - \mathbf{v}_c)^T S_c^{-1} (\mathbf{v} - \mathbf{v}_c) dx dy. \tag{2.23}$$

Here, $\mathbf{v}_c$ and S_c are derived directly from intensity data in step 1, while $\mathbf{v}_n$ and S_n require the velocities to be known at each neighbouring point and cannot be computed explicitly. Singh therefore derives iterative equations using the calculus of variations:

$$\begin{aligned} \mathbf{v}_n^0 &= \mathbf{v}_c . \\ \mathbf{v}_n^{k+1} &= \left[S_c^{-1} + (S_n^k)^{-1} \right]^{-1} \left[S_c^{-1} \mathbf{v}_c + (S_n^k)^{-1} \mathbf{v}_n^k \right] . \end{aligned} \tag{2.24}$$

We use a maximun of 25 iterations (less if all velocity differences between adjacent iterations is 10^{-2} or less). Singh uses an SVD to compute the matrix inverse in (2.24), replacing singular values less than 0.1 by 0.1 to avoid singular systems.

Finally, eigenvalues of the covariance matrix $[S_c^{-1} + S_n^{-1}]^{-1}$, denoted λ_1 and λ_2, where $\lambda_1 \geq \lambda_2$, serve as confidence measures estimates for step 2. In reporting error statistics, we threshold the 2-d velocities, rejecting those velocities where $\lambda_1 \geq \tau$, for τ being some constant. We also report error statistics for subsets of the velocity estimates from step 1 (2.19), with a threshold based on the largest eigenvalue of S_c (2.20).

2.3 Energy-Based Methods

A third class of optical flow techniques is based on the output energy of velocity-tuned filters [2, 7, 10, 26, 29, 33]. These are also called frequency-based methods owing to the design of velocity-tuned filters in the Fourier domain [1, 22, 48, 59]. The Fourier transform of a translating 2-d pattern (2.1) is

$$\hat{I}(\mathbf{k}, \omega) = \hat{I}_0(\mathbf{k}) \ \delta(\omega + \mathbf{v}^T \mathbf{k}) \ , \tag{2.25}$$

where $\hat{I}_0(\mathbf{k})$ is the Fourier transform of $I(\mathbf{x}, 0)$, $\delta(k)$ is a Dirac delta function, ω denotes temporal frequency and $\mathbf{k} = (k_x, k_y)$ denotes spatial frequency. This shows that all nonzero power associated with a translating 2-d pattern lies on a plane through the origin in frequency space.

[7] In the event there are two or more SSD minima (with a small threshold) we choose the SSD minimum that corresponds to the smallest displacement.

Interestingly, it has been shown that certain energy-based methods are equivalent to correlation-based methods [1, 48] and to the gradient-based approach of Lucas and Kanade [2, 52]. Indeed, as mentioned below, results reported in [26, 52] with our image sequences are close to those for our implementation of the Lucas and Kanade gradient-based method and therefore support this claim.

Heeger

Here we consider the method developed by Heeger [28, 29], formulated as a least-squares fit of spatiotemporal energy to a plane in frequency space. Local energy is extracted using Gabor-energy filters, with 12 filters at each of several spatial scales, tuned to different spatial orientations and different temporal frequencies. Ideally, for a single translational motion, the responses of these filters are concentrated about a plane in frequency space. Heeger derives the expected response $R(u,\, v)$ of a Gabor-energy filter tuned to frequency $(k_x,\, k_y,\, \omega)$ for translating white noise as a function of velocity:

$$R(u,\, v) \;=\; \exp\left[\frac{-4\pi^2\sigma_x^2\sigma_y^2\sigma_t^2(uk_x + vk_y + \omega)}{(u\sigma_x\sigma_t)^2 + (v\sigma_y\sigma_t)^2 + (\sigma_x\sigma_y)^2}\right] \,, \tag{2.26}$$

where σ_x, σ_y and σ_t are the standard deviations of the Gaussian component of the Gabor filter.

To derive Heeger's solution, let M_i, $1 \le i \le 12$, denote the set of filters with the same orientation tuning, and let $\bar{m}_i$ and $\bar{R}_i$ be the sum of measured and predicted energies, m_j and R_j, from filters j in the set M_i:

$$\bar{m}_i \;=\; \sum_{j \in M_i} m_j \qquad \text{and} \qquad \bar{R}_i \;=\; \sum_{j \in M_i} R_j(u,v)\,. \tag{2.27}$$

A least-squares estimate for (u,v) that minimizes the difference between the predicted and measured motion energies is given by the minimum of

$$f(u,v) \;=\; \sum_{i=1}^{12}\left[m_i - \bar{m}_i\frac{R_i(u,v)}{\bar{R}_i(u,v)}\right]^2. \tag{2.28}$$

Heeger [28, 29] has outlined two ways of minimizing (2.28): We implemented the nonlinear minimization using Newton's method but the results were unsatisfactory; in addition to requiring a good initial guess we rarely obtained convergence if the measurement error was much over 10%.

For the results reported below we estimated $\mathbf{v}$ using a modified version of Heeger's parallel method: We construct a distribution $g(\mathbf{v}) = \exp^{-0.95 f(\mathbf{v})}$ for a range $-N \le (u,v) \le N$, the minima of which gives the subpixel velocity estimate, unless the aperture problem occurs in which case the minima forms a trough. To compute the sub-pixel minima we devised an *ad hoc* method that involves multi-resolution minima selection. At the coarsest resolution we compute $g(u,v)$ values in the range $-N \le u,v \le N$ in 0.2 increments. If the spread of the lowest 30 values (their average distance from the global minima denoted here as (u_M, v_M)) is within some threshold (we used a value of 3), we assume a 2-d velocity and re-compute (u,v) at a finer resolution about the minima. That is, we compute $g(u,v)$ values for $u_M - 0.2 \le u \le u_M + 0.2$ and $v_M - 0.2 \le v \le v_M + 0.2$ in 0.01 increments and determine the full velocity as the location of the resulting minima. If the spread of the smallest 30 values at the coarsest resolution is large (> 3) we assume a normal velocity and fit a straight line through the minima, determining the normal velocity as the vector from the origin to the closest point on the line.

Like Heeger, we apply the Gabor filters to each level of a Gaussian pyramid; the filter parameters were taken from [29]. Our implementation permits the use of any level of the pyramid and, as Heeger suggests, we choose the estimate of $\mathbf{v}$ from the level that best satisfies expected range of speeds for that level. Level 0 (the image) should be used for speeds between 0–1.25 pixels/frame, while levels 1 and 2 should be used for speeds between 1.25–2.5 and 2.5–5 pixels/frame.

2.4 Phase-Based Techniques

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. For this report we have classified zero-crossing techniques [14, 16, 27, 61] as phase-based methods because zero-crossings can be viewed as level phase-crossings. The generalized use of phase information for optical flow was first developed by Fleet and Jepson [19, 22].

Waxman, Wu and Bergholm

Waxman, Wu and Bergholm [61] apply spatiotemporal filters to binary edge maps to track edges in real-time. Edge maps $E(\mathbf{x}, t)$, based on DOG zero-crossings [41], are smoothed with a Gaussian filter to create a *convected activation profile* $A(\mathbf{x}, t)$:

$$A(\mathbf{x}, t) \;=\; G(\mathbf{x}, t; \sigma_x, \sigma_y, \sigma_t) * E(\mathbf{x}, t)\ . \tag{2.29}$$

Level contours of $A(\mathbf{x}, t)$ are then tracked using differential methods. However, because the spatial gradient of $A(\mathbf{x}, t)$ will be zero at edge locations, a second-order approach is adopted, applying the constraints in (2.4) to $A(\mathbf{x}, t)$. Velocity estimates at edge locations are then given by

$$\mathbf{v} \;=\; \frac{(A_{xt}A_{yy} - A_{yt}A_{xy}\ ,\ A_{yt}A_{xx} - A_{xt}A_{xy})}{A_{xx}A_{yy} - A_{xy}^2}\ , \tag{2.30}$$

where the second derivatives of $A(\mathbf{x}, t)$ are computed by convolving the appropriate Gaussian derivatives with the edge maps.

In our implementation, the central Gaussian of the DOG had a standard deviation of 1.5 pixels-frames and the ratio of surround to centre sizes was 1.6. For the activation profile we used $\sigma_x = \sigma_y = 2.0$ and $\sigma_t = 1.0$ (we require 7 frames for our implementation). Waxman et al. also proposed a *multiple* σ method which attempts to choose the best velocity at an edge location. For various $\sigma_x = \sigma_y$ values (we use 1.0, 1.5 and 2.0) we choose the velocity that maximizes

$$\max\left(\frac{2\sigma_t}{\sigma_x + \sigma_y}||\mathbf{v}||_2\right)\ . \tag{2.31}$$

Finally, as suggested by Waxman et al., the Hessian of A (i.e. the Gaussian curvature of A given in the denominator in (2.30)) provides a confidence measure for the velocities: If the Hessian is greater than or equal to a threshold τ (here we use $\tau = 0.5$), then full velocity is computed at the edge location. If it is less than τ we can proceed with a normal velocity calculation

$$(u_n,\, v_n) \;=\; -\frac{1}{\nabla^2 A}(A_{xt},\, A_{yt})\ . \tag{2.32}$$

Fleet and Jepson

The method developed by Fleet and Jepson [19] defines component velocity in terms of the instantaneous motion normal to 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

$$R(\mathbf{x},t) \;=\; \rho(\mathbf{x},t)\,\exp[i\phi(\mathbf{x},t)]\;, \tag{2.33}$$

where $\rho(\mathbf{x},t)$ and $\phi(\mathbf{x},t)$ are the amplitude and phase parts of R. The component of 2-d velocity in the direction normal to level phase contours is then given by $\mathbf{v}_n = s\mathbf{n}$, where the normal speed and direction are given by

$$s \;=\; \frac{-\,\phi_t(\mathbf{x},t)}{\|\,\nabla\phi(\mathbf{x},t)\,\|}\;, \qquad \mathbf{n} \;=\; \frac{\nabla\phi(\mathbf{x},t)}{\|\,\nabla\phi(\mathbf{x},t)\,\|}\;, \tag{2.34}$$

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

$$\phi_x(\mathbf{x},t) \;=\; \frac{\mathrm{Im}[R^*(\mathbf{x},t)\;R_x(\mathbf{x},t)\,]}{|R(\mathbf{x},t)|^2}\;, \tag{2.35}$$

where R^* is the complex conjugate of R.

The use of phase is motivated by their claim that the phase component of band-pass filter outputs is more stable than the amplitude component when small deviations from image translations that regularly occur in 3-d scenes are considered [20]. However, they show that phase can also be unstable, with instabilities occurring in the neighbourhoods about phase singularities. Such instabilities can be detected with a straightforward constraint on the instantaneous frequency of the filter output and its amplitude variation in space-time [20, 22, 34]:

$$\|\;\nabla\log R(\mathbf{x},t) \;-\; i(\mathbf{k},\omega)\;\| \;\leq\; \sigma_k\,\tau\;, \tag{2.36}$$

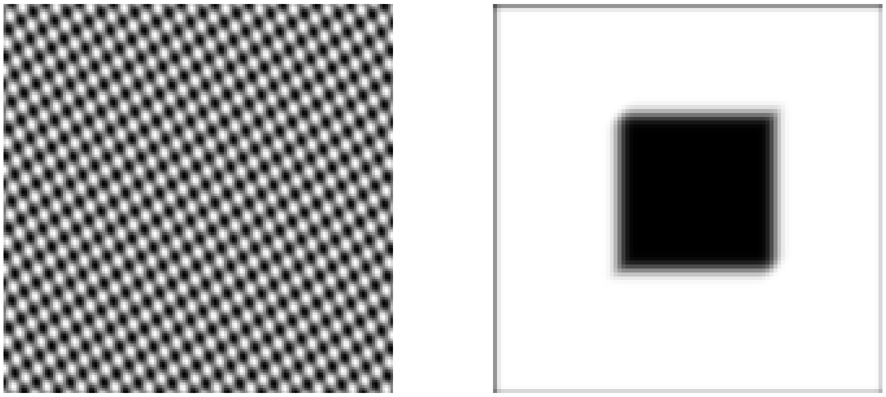
where $(\mathbf{k},\omega)$ denotes the spatiotemporal frequency to which the filter is tuned, σ_k is the standard deviation of the isotropic amplitude spectra they use and τ denotes a threshold that can be used to reject unreliable component velocity measurements. As τ decreases the filter output is more tightly constrained and therefore larger singularity neighbourhoods are detected. Like Fleet and Jepson we normally set $\tau = 1.25$. A second constraint on the amplitude of response is also used to ensure a reasonable signal-to-noise ratio.

Finally, given the component (normal) velocity estimates from the different filter channels, a linear velocity model is fit to each local region. Estimates that satisfy the stability and SNR constraints are collected from 5×5 neighbourhoods, to which the best linear velocity model, in a LS sense, is determined. To ensure that there is sufficient local information for reliable velocity estimates, they introduce further constraints on the conditioning of the linear system and on the residual LS error. To illustrate their results, Fleet and Jepson only consider 2-d velocity measurements for which the condition number is less than 10.0, and the residual error is less than 0.5.

Like [19, 22], our implementation uses only a single scale tuned to a spatiotemporal wavelength of 4.25 pixels-frames. A more complete implementation would normally have 3–5 scales in total. The entire temporal support is 21 frames, and we used the same threshold values as those in [19, 22].

3 Experimental Technique

We have examined the performance of these techniques on real sequences and synthetic sequences for which 2-d motion fields were known. Before discussing the results, it is useful to describe the image sequences used, as well as our angular measures of error.



(a) **Sinusoid1** (b) **Square2**

Figure 3.1: *Frames from the sinusoidal and square sequences.*

3.1 Synthetic Image Sequences

The main advantages of synthetic inputs are that the 2-d motion fields and scene properties can be controlled and tested in a methodical fashion. In particular, we have access to the true 2-d motion field and can therefore quantify performance. Conversely, it must be remembered that such inputs are usually clean signals (involving no occlusion, specularity, shadowing, transparency, etc.) and therefore this measure of performance should be taken as an optimistic bound on the expected errors with real image sequences. Our synthetic image sequences include:

Sinusoidal Inputs: This consists of the superposition of two sinusoidal plane-waves:

$$\sin(\mathbf{k}_1 \cdot \mathbf{x} + \omega_1 t) + \sin(\mathbf{k}_2 \cdot \mathbf{x} + \omega_2 t) \ . \tag{3.37}$$

Although we tested many different wavelengths and velocities, the results reported below are based mainly on spatial wavelengths of 6 pixels, with orientations of 54° and −27°, and speeds of 1.63 and 1.02 pixels/frame respectively. The resulting plaid pattern translates with velocity $\mathbf{v} = (1.585, 0.863)$ pixels/frame and is called **Sinusoid1** (see Figure 3.1a). We also report results on another plaid pattern with wavelengths of 16 pixels/cycle and a velocity of $\mathbf{v} = (1, 1)$, called **Sinusoid2**. This signal permits very accurate DOG edge detection and numerical differentiation.

Translating Squares: Our other simple test case involves a translating dark square (with a width of 40 pixels) over a bright background (see Figure 3.1b). We concentrate on a sequence called **Square2** which has uniform velocity $\mathbf{v}_2 = (\frac{4}{3}, \frac{4}{3})$.[8] We occasionally report results on a simpler case with velocity $\mathbf{v}_1 = (1, 1)$ called **Square1** for which some techniques produce better results. This type of input helps to illustrate the aperture problem and the inherent spatial smoothing in the different techniques. While the sinusoidal inputs can be viewed as dense in space and sparse in frequency space, the square data is concentrated in space along its edges, but richer in its frequency spectra.

[8] **Square2** was created by blurring and then down-sampling a larger version of the images which translated at an integer velocity of 4 pixels/frame.

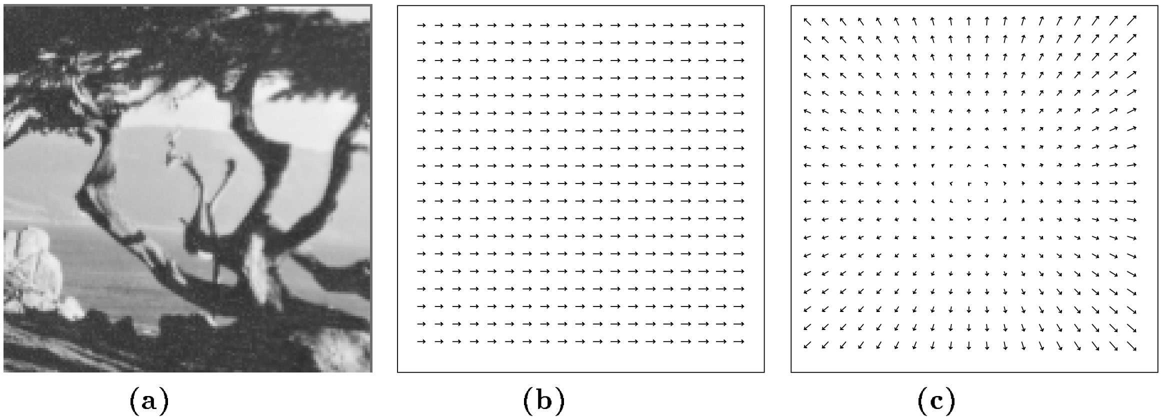


Figure 3.2: *Surface texture used for the* **Translating** *and* **Diverging Tree** *sequences, and the respective 2-d motion fields.*

3D Camera Motion and Planar Surface: Following [19] we used two sequences that simulate translational camera motion with respect to a textured planar surface (see Figure 3.2): In the **Translating Tree** sequence, the camera moves normal to its line of sight along its X-axis, with velocities all parallel with the image x-axis, with speeds between 1.73 and 2.26 pixels/frame. 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 1.29 pixels/frame on left side to 1.86 pixels/frame on the right.

Yosemite Sequence: The **Yosemite** sequence is a more complex test case (see Figure 3.3). The motion in the upper right is mainly divergent, the clouds translate to the right with a speed of 1 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 portion of the images however, causing most methods to produce poorer velocity measurements.

The sinusoidal and translating square sequences were created by the authors. The **Translating** and **Diverging Tree** sequences were created by David Fleet. The **Yosemite** sequence, created by Lynn Quam, was provided to us by David Heeger.

3.2 Real Image Sequences

Four real image sequences, shown in Figure 3.4, were also used:

SRI Sequence: In this sequence the camera translates parallel to the ground plane, perpendicular to its line of sight, in front of clusters of trees. This is a particularly challenging sequence because of the relatively poor resolution, the amount of occlusion, and the low contrast. Velocities are as large as 2 pixels/frame.

NASA Sequence: The NASA sequence is primarily dilational; the camera moves along it's line of sight toward the Coke can near the centre of the image. Image velocities are typically less than 1 pixel/frame.

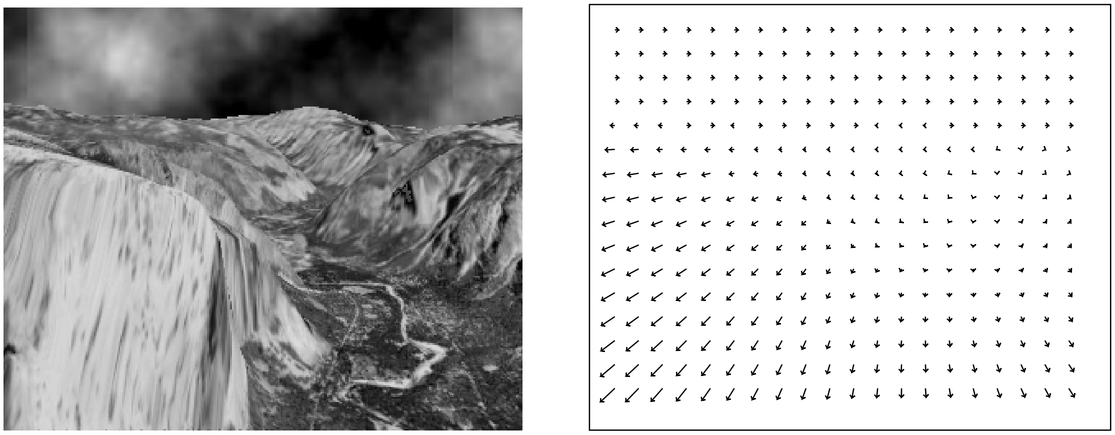


Figure 3.3: *a) left: One frame from the* **Yosemite** *sequence; b) right: Correct flow field for the* **Yosemite** *sequence.*

Rotating Rubik Cube: In this image sequence a Rubik's cube is rotating counter-clockwise on a turntable. The motion field induced by the rotation of the cube includes velocities less than 2 pixels/frame (velocities on the turntable range from 1.2 to 1.4 pixels/frame, and those on the cube are between 0.2 and 0.5 pixels/frame).

Hamburg Taxi Sequence: In this street scene there were four moving objects: *1)* the taxi turning the corner; *2)* a car in the lower left, driving from left to right; *3)* a van in the lower right driving right to left; and *4)* a pedestrian in the upper left. Image speeds of the four moving objects are approximately 1.0, 3.0, 3.0, and 0.3 pixels/frame respectively.

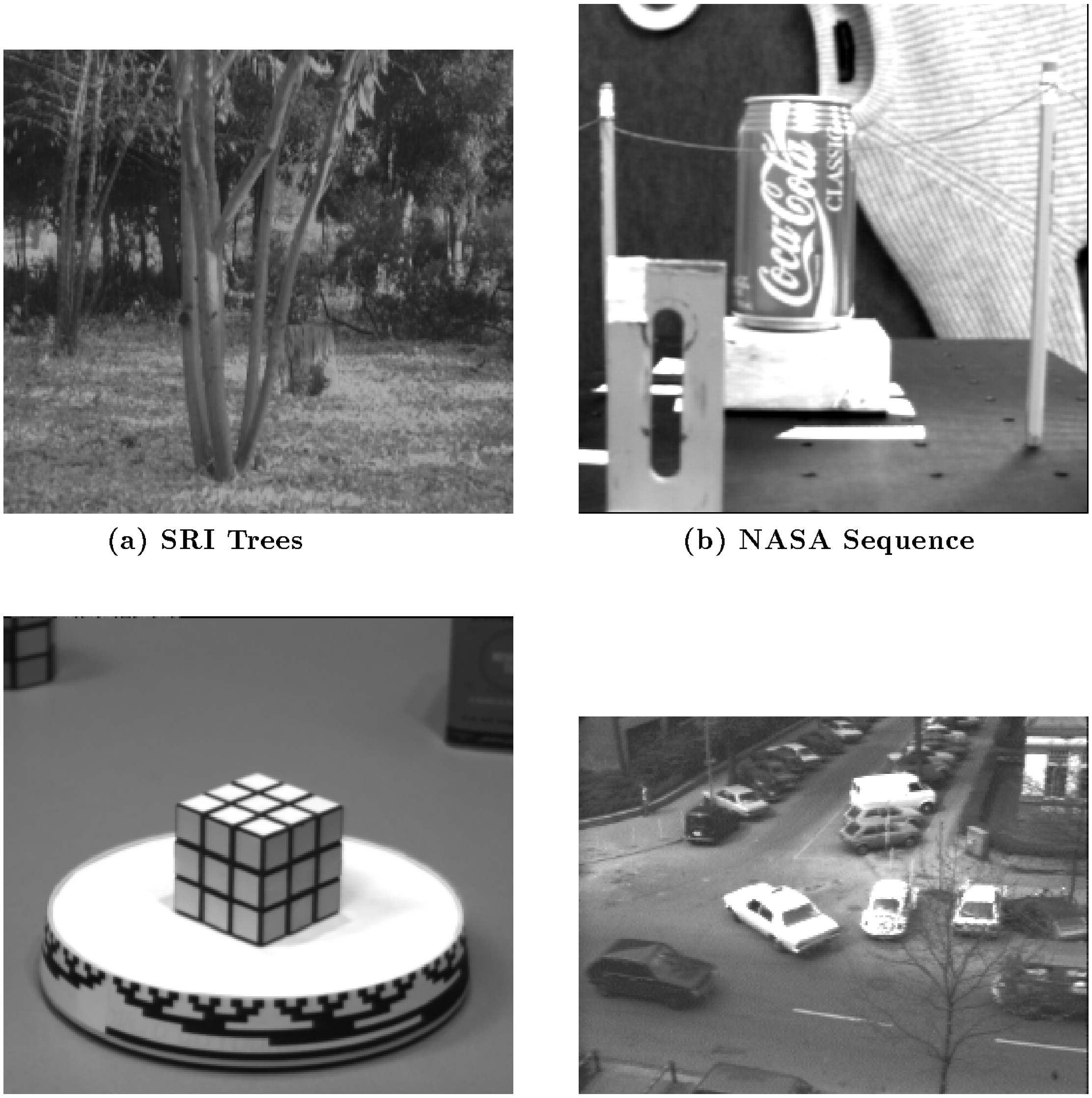
The **Nasa** and **SRI** image sequences were obtained from the IEEE Motion Workshop Database at Sarnoff Research Centre, courtesy of NASA-Ames Research Center and SRI International. The **Hamburg Taxi** sequence was provided courtesy of the University of Hamburg and the **Rubik Cube** sequence was provided by Richard Szeliski at DEC, Cambridge Research Labs.

3.3 Error Measurement

Following [19, 22] we use an angular measure of error: velocity may be written as displacement per time unit as in $\mathbf{v} = (u, v)$ pixels/frame, or as a space-time direction vector $(u, v, 1)$ in units of (pixel, pixel, frame). Of course, velocity is obtained from the direction vector by dividing by the third component. When velocity is viewed (and measured) as orientation in space-time, it is natural to measure errors as angular deviations from the correct space-time orientation. Therefore, let velocities $\mathbf{v} = (v_1, v_2)^T$ be represented as 3-d direction vectors, $\vec{\mathbf{v}} \equiv \frac{1}{\sqrt{u^2+v^2+1}}(u, v, 1)^T$. The angular error between the correct velocity $\vec{\mathbf{v}}_c$ and an estimate $\vec{\mathbf{v}}_e$ is

$$\psi_E = \arccos(\vec{\mathbf{v}}_c \cdot \vec{\mathbf{v}}_e) \ . \tag{3.38}$$

This error measure is convenient because it handles large and very small speeds without the amplification inherent in a relative measure of vector differences. It does have some bias however. For example, directional errors at small speeds do not give as large an angular error as similar



(a) SRI Trees

(b) NASA Sequence

(c) Rubik Cube

(d) Hamburg Taxi

Figure 3.4: *One frame is shown from each of the four real image sequences.*

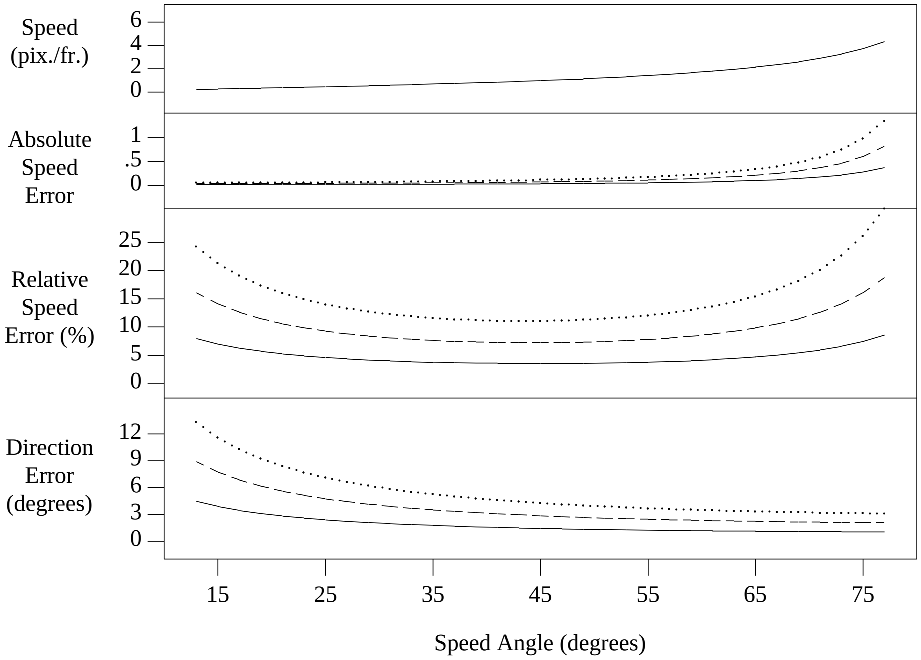


Figure 3.5: Speed in Degrees vs. Pixels/Frame (reprinted with permission from [22]). For fixed angular velocity errors ψ_E in (3.38), errors in pixels/frame depend on angular speed. With $\mathbf{v}$ represented as a unit direction vector in space-time, we can view velocity in spherical coordinates, in terms of angular speed θ_v and direction θ_x. From top to bottom in the figure, with $\psi_E = 1°$ (solid), 2° (dashed), and 3° (dotted), the four panels correspond to: (a) Speed in pixels/frame: $\tan(\theta_v)$, (b) Absolute speed errors (pixels/frame): $\tan(\theta_v) - \tan(\theta_v + \psi_E)$, (c) Relative speed errors: $100.0(\tan(\theta_v) - \tan(\theta_v + \psi_E))/\tan(\theta_v)$ and (d) Maximum error in direction of motion (in degrees): $\psi_E/\sin(\theta_v)$.

directional errors at higher speeds [22]. Relative errors of 10% correspond to angular errors of roughly 2.5° when speeds are near 1 pixel/frame. For slower and higher speeds, relative errors of 10% correspond to smaller angular errors [22]. This is illustrated in Figure 3.5.

A complementary measure is also available for errors in measurements of normal (component) velocity. There is a linear relationship between normal velocity $\mathbf{v}_n = s\mathbf{n}$ and 2-d velocity $\mathbf{v}_c$; that is, $\mathbf{n} \cdot \mathbf{v}_c - s = 0$. All component velocities generated by a translating texture pattern should ideally lie on the plane normal to $\vec{\mathbf{v}}_c$. Our error measure for component velocities is the angle between the measured component velocity and the constraint plane; that is,

$$\psi_E = \arcsin\left(\vec{\mathbf{v}}_c \cdot \vec{\mathbf{v}}_n\right) \ , \tag{3.39}$$

where $\vec{\mathbf{v}}_n \equiv \frac{1}{\sqrt{1+s^2}}(\mathbf{n}, \, -s)$.

There are many ways in which error behaviour may be reported. For the synthetic sequences we extract subsets of estimates using confidence measures and then report the densities of these sets of estimates along with their mean error and standard deviations. These are presented in tables so that different techniques can be compared on the same inputs. For the real image sequences we can only show the computed flow fields and discuss qualitative properties, leaving the reader to judge. We also refer the interested reader to Appendices B through F that contain

Technique	Average Error	Standard Deviation	Density
Horn and Schunck (original)	4.19°	0.50°	100%
Horn and Schunck (modified)	2.55°	0.59°	100%
Lucas and Kanade (no thresholding)	2.47°	0.16°	100%
Uras et al. (no thresholding)	2.59°	0.71°	100%
Nagel	2.55°	0.93°	100%
Anandan	30.80°	5.45°	100%
Singh ($n = 2$, $w = 2$, $N = 2$)	2.24°	0.02°	100%
Singh ($n = 2$, $w = 2$, $N = 4$)	91.71°	0.04°	100%
Waxman et al. $\sigma_f = 1.5$	64.26°	26.14°	12.8%
Fleet and Jepson $\tau = 1.25$	0.03°	0.01°	100%

Table 4.1: *Summary of* **Sinusoid 1** *Results. See the text for a discussion of these results and the apparent anomalies.*

more detailed results including histograms of errors, images of error as a function of image position, and proportions of estimations with errors less than 1°, 2°, and 3° degrees – these proportions provide a good indication of the percentages of estimates that may be useful for computing egomotion and 3-d structure.

4 Experimental Results

Section 4 reports the quantitative performance of the different techniques on the synthetic input sequences, discusses the use of confidence measures and shows the flow fields produced by the techniques on the natural image sequences.

4.1 Synthetic Image Sequences

In reporting the performance of the optical flow methods applied to the synthetic sequences, for which 2-d motion fields are known, we concentrate on error statistics (mean and standard deviation) and the density of measurements for subsets of the estimates extracted using confidence measures as thresholds. When reporting error statistics we use $a° \pm b°$ to denote a mean of a degrees with standard deviation b. The techniques will be discussed in the order they were described in Section 2, with differential methods followed by matching, energy-based, and then phase-based approaches.

4.2 Sinusoidal Inputs

Table 4.1 summarizes the main results of the techniques applied to **Sinusoid1**, which are generally very good. In fact, because of the relatively dense, homogeneous structure of the input, the collections of flow estimates produced by most of the techniques have not been thresholded using confidence measures. Nor have the signals been smoothed with low-pass filters since they will have little effect on performance unless subsampled, as discussed below. Many of the results are self-evident from the tables, although several deserve comments.

Beginning with differential methods, observe that our modified version of Horn and Schunck's algorithm [31], with improved numerical differentiation, performed better than the original algorithm. As one might expect, the accuracy of the original method approaches the modified method as the spatial wavelength in (3.37) is increased (for **Sinusoid2** the error was $0.97° \pm 2.62°$ for the original method and $0.86° \pm 2.39°$ for our modified version). The large standard deviations are not very significant as they are caused by directional errors near the image boundary. It is interesting to note that we found considerable variation in results as a function of the smoothness parameter λ; when $\lambda = 100$ results were noticeably worse.

Results from the gradient-based method of Lucas and Kanade are also good, with accuracy similar to that produced by the modified version of Horn and Schunck's algorithm which shares the same numerical differentiation. Interestingly, we did find with this input that the gradient-based method described in [51] produced poorer results (with error statistics of $5.23° \pm 0.70°$).

The estimates produced by Nagel's technique are also good. More accurate results can be obtained when **Sinusoid2** is used as better derivative estimation is possible (in this case we found errors of $0.04° \pm 0.23°$). We also found that the results were sensitive to certain parameters: results were significantly worse with larger values of α.

While the differential techniques performed well on sinusoidal inputs, the matching techniques did not. Anandan's technique produced consistent velocity estimates with the direction reasonably accurate but the speed usually poor. The main problem is caused by aliasing in the construction of the Laplacian pyramid: Although complete, the Laplacian pyramid described in [12] produces band-pass channels (levels) that contain substantial aliasing when considered independently of one another. Only when different levels are combined does the aliasing cancel to provide accurate reconstruction. With sinusoidal inputs and a coarse-fine control strategy on the Laplacian pyramid, aliasing causes major errors at coarse levels that are then propagated systematically to finer levels.

Similar problems would occur with Singh's technique, if implemented with a Laplacian pyramid. However, a different problem occurred with our implementation. With nearly periodic inputs (such as those due to textured inputs, sinusoidal inputs or band-pass filtered signals) there will be multiple local minima in the SSD surface (i.e. ghost matches). Furthermore, because the SSD surface is initially evaluated at a small number of integer displacements, the global minima may fall midway between integer displacements, in which case other (ghost) minima may be mistaken for global minima if they occur closer to an integer displacement. For example, as shown in Table 4.1, when the search space is limited to displacements of 2 pixels, only one minima exists within the search space. But when displacements of 4 pixels are considered, other local minima are chosen consistently. The measurement errors are all speed errors of about 6 pixels, which is the wavelength of the input components. This sampling problem occurs less frequently with natural images which lack this exact periodicity, but sampling problems will continue to occur unless finer sampling and interpolation are used.

For Heeger's technique [29] (as well as Fleet and Jepson's technique [34], see below) reasonable results can only be expected when the input frequencies match those in the pass-band to which the filters are tuned. In Heeger's case there is the additional assumption that the input has a flat amplitude spectrum, which is clearly violated by our sinusoidal inputs. Violation of this assumption is most evident when the frequencies of the component sinusoids are not close to the filter tunings, which is the case for **Sinusoid1**. Although Heeger's method did not produce any results for **Sinusoid1**, it did produce good results for others. For example, for sinusoids with orientations of 0° and 90°, speeds of 1 pixel/frame, and spatiotemporal wavelengths of 4 pixels/cycle, we obtained errors of $3.24° \pm 0.05°$ with a density of 24.3%.

To obtain good results with the zero-crossing algorithm of Waxman et al. one must choose the standard deviation of the activation kernel so that it is small enough to prevent interaction

Technique	Average Error	Standard Deviation	Density
Horn and Schunck (original)	45.80°	10.49°	100%
Horn and Schunck (original) $\|\|\nabla I\|\| \geq 1.0$	26.92°	8.13°	3.8%
Horn and Schunck (modified)	32.18°	11.57°	100%
Horn and Schunck (modified) $\|\|\nabla I\|\| \geq 1.0$	24.26°	9.24°	33.6%
Lucas and Kanade ($\lambda_2 \geq 1.0$)	0.34°	0.35°	6.2%
Lucas and Kanade ($\lambda_2 \geq 5.0$)	0.23°	0.23°	4.0%
Uras et al. (unthresholded)	19.03°	35.52°	100%
Uras et al. ($det(H) > 1.0$)	0.10°	0.14°	17.0%
Nagel	34.25°	12.17°	100%
Nagel $\|\|\nabla I\|\|_2 \geq 1.0$	25.15°	10.33°	35.4%
Anandan (unthresholded)	36.47°	17.34°	100%
Anandan ($c_{min} \geq 5.0$)	8.23°	3.86°	0.9%
Singh (Step 1, $n = 2$, $w = 2$)	42.62°	17.71°	100%
Singh (Step 1, $n = 2$, $w = 2$, $\lambda_1 \leq 5.0$)	0.11°	0.78°	6.0%
Singh (Step 2, $n = 2$, $w = 2$)	37.63°	15.24°	100%
Singh (Step 2, $n = 2$, $w = 2$, $\lambda_1 \leq 0.1$)	37.71°	15.20°	99.4%
Heeger	2.89°	3.88°	24.8%
Waxman et al. $\sigma_f = 1.5$	0.09°	0.10°	0.8%
Fleet and Jepson $\tau = 1.25$	0.07°	0.08°	3.7%

Table 4.2: *Summary of* **Square1** *2D Velocity Results.*

between adjacent edges and yet big enough to track each edge over time. Moreover, zero-crossings must be localized to sub-pixel accuracy (not done by Waxman et al.) in order to obtain good quantitative results when the underlying motion is not an integer multiple of pixels. For example, unlike **Sinusoid1**, the input **Sinusoid2** does satisfy these requirements, in which case the errors reduce to 0.04° ± 0.03° with a density of 11.94%, the low density reflecting the density of edge locations.

Finally, in Fleet and Jepson's case, the spatiotemporal wavelength of the sinusoid closely matches those to which their filters are tuned, and the results are very good. With more general input signals, we found that when input signals have local power concentrated near the boundary of a filter's amplitude spectra (far from its filter tuning), slight errors appear, as a bias in the component velocity estimates toward the velocity tuning of the filters.

4.3 Translating Square Data

The 2-d velocity estimates and the normal velocity estimates of the nine techniques for the **Square2** sequence are summarized in Tables 4.3 and 4.5. Of course, we expect normal estimates along the edges of the square and 2-d velocities only at the corners. Flow fields produced by the techniques are also shown Appendix C; these help show the distribution of measurements and hence the support of the measurement process.

From Table 4.3 it is evident that several techniques appear to produce very poor results. In several of these cases, such as the differential methods of Horn and Schunck, and Nagel, the

Technique	Average Error	Standard Deviation	Density
Horn and Schunck (original)	47.21°	14.60°	100%
Horn and Schunck (original) $\lVert\nabla I\rVert \geq 1.0$	27.61°	9.86°	18.9%
Horn and Schunck (modified)	32.81°	13.67°	100%
Horn and Schunck (modified) $\lVert\nabla I\rVert \geq 1.0$	26.46°	10.86°	42.9%
Lucas and Kanade ($\lambda_2 \geq 1.0$)	0.21°	0.16°	7.9%
Lucas and Kanade ($\lambda_2 \geq 5.0$)	0.14°	0.10°	4.6%
Uras et al. ($det(H) > 1.0$)	0.15°	0.10°	26.1%
Nagel	34.57°	14.38°	100%
Nagel $\lVert\nabla I\rVert_2 \geq 1.0$	26.67°	11.84°	44.0%
Anandan (unthresholded)	31.46°	18.31°	100%
Anandan ($c_{min} \geq 0.25$)	10.46°	5.36°	0.6%
Singh (Step 1, $n = 2$, $w = 2$)	49.03°	21.38°	100%
Singh (Step 1, $n = 2$, $w = 2$, $\lambda_1 \leq 5.0$)	9.85°	21.09°	4.2%
Singh (Step 1, $n = 2$, $w = 2$, $\lambda_1 \leq 3.0$)	2.02°	2.36°	1.6%
Singh (Step 2, $n = 2$, $w = 2$)	45.16°	21.10°	100%
Singh (Step 2, $n = 2$, $w = 2$, $\lambda_1 \leq 0.1$)	46.12°	18.64°	81.9%
Heeger	6.16°	4.02°	29.3%
Waxman et al. $\sigma_f = 1.5$	8.78°	4.71°	1.1%
Fleet and Jepson $\tau = 1.25$	0.07°	0.02°	2.2%
Fleet and Jepson $\tau = 2.5$	0.18°	0.13°	12.6%

Table 4.3: *Summary of* **Square2** *2D Velocity Results.*

problem is the lack of discrimination by the algorithm between measurements of normal velocity versus 2-d velocity. From the flow fields for Horn and Schunck and Nagel (shown in Figures C.2 and C.3) for **Square2** it is clear that these methods produce normal measurements along the edges, which blend into 2-d measurements at the corners. Although this is readily apparent, the algorithms do not provide a way of segmenting the measurements into 2-d flow, normal velocity or unreliable measurements. Furthermore, neither the magnitude of the local gradient nor the local energy defined by the objective functionals in (2.5) or (2.11) could be used as confidence measures in this case. This stands in contrast to the Lucas and Kanade gradient-based method which integrates measurements locally with a clear means of segregating normal from 2-d velocities based on the eigenvalues of the normal matrix in (2.8) (i.e. the confidence measures).

The second-order differential method of Uras et al. produced accurate results, with a confidence measure based on the (spatial) Hessian of the smoothed image sequence proving useful. The higher density of estimates for this method is a consequence of using a single estimate for each 8×8 region, which limits the spatial resolution of the flow field.

The results for the matching methods are also poor. In the case of Anandan's method, we find that the smoothing stage produces both normal and 2-d estimates of velocity, like Horn and Schunck's and Nagel's methods above (see Figures C.2 and C.3). In this case however, we do have a potential confidence measure in c_{min} as suggested by Anandan. However, although

Technique for Normal Velocity	Average Normal	Standard Deviation	Density
Lucas and Kanade (LS) ($\lambda_1 \geq 1.0$)	0.16°	3.45°	16.2%
Lucas and Kanade (LS) ($\lambda_1 \geq 5.0$)	0.03°	0.09°	10.7%
Lucas and Kanade (Raw) ($\|\|\nabla I\|\| \geq 0.0$)	5.67°	16.0°	75.8%
Lucas and Kanade (Raw) ($\|\|\nabla I\|\| \geq 5.0$)	0.19°	3.99∗°	24.2%
Heeger	1.39°	1.30°	75.2%
Waxman et al. $\sigma_f = 1.5$	0.04°	0.30°	3.7%
Fleet and Jepson $\tau = 1.25$	0.01°	0.09°	12.7% (2.4)

Table 4.4: *Summary of* **Square1** *Normal/Component Velocity Results.*

Technique for Normal Velocity	Average Normal	Standard Deviation	Density
Lucas and Kanade (LS) ($\lambda_1 \geq 1.0$)	0.07°	0.06°	25.5%
Lucas and Kanade (LS) ($\lambda_1 \geq 5.0$)	0.14°	2.76°	25.3%
Lucas and Kanade (Raw) ($\|\|\nabla I\|\| \geq 5.0$)	0.12°	2.44°	32.5%
Heeger	1.02°	4.35°	70.7%
Waxman et al. $\sigma_f = 1.5$	4.28°	5.42°	3.6%
Fleet and Jepson $\tau = 1.25$	−0.05°	0.05°	17.6% (1.1)
Fleet and Jepson $\tau = 2.5$	0.05°	0.23°	65.4% (4.2)

Table 4.5: *Summary of* **Square2** *Normal/Component Velocity Results.*

it is clear that results improve dramatically with the use of this threshold, the accuracy of the resultant 2-d velocity was still reasonably poor. It appears that subpixel measurement accuracy is poor and that the threshold is not reliable in separating normal from 2-d measurements.

Singh's algorithm produces visually pleasing but somewhat inaccurate results. We find that there is a common problem with matching methods with the aperture problem. While 2-d velocities are found with reasonably accuracy, the SSD minima will be trough-like when the aperture problem occurs, in which case, the minima found for the sampled SSD surface at integer displacements is extremely sensitive to small variations along the edge, meaning that normal velocity measurements were not trustworthy. Of course, a threshold on the eigenvalues of the inverse covariance matrix at step 1 are very useful at separating normal from 2-d velocities. Unfortunately, all velocities, including the normal velocities, are required for step 2 of Singh's algorithm. Hence, those normal estimates that are poor will corrupt step 2, in which case the covariance matrix (at step 2) is of little help.

The square sequences are clean inputs and purely translational. However, **Square1** moves an integer multiple of pixels between adjacent frames, while **Square2** has subpixel motion with vertical and horizontal and vertical speeds of 1.33 pixels/frame, and therefore a 2-d speed of 1.89 pixels/frame. While most techniques produced similar results in both cases, the zero-crossing method of Waxman et al. performs more poorly with **Square2** than **Square1** because our implementation lacks subpixel resolution. Compared to the large errors in Tables 4.3 and 4.5 for **Square2**, our results on **Square1** were $0.09° \pm 0.1°$ for 2-d velocity estimates and $0.04° \pm 0.3°$ for normal velocities.

For Heeger's technique, we found that estimates from level 1 of the Gaussian pyramid were more accurate that those from level 0. This is expected since the correct velocity $(1.33, 1.33)$ coincides with the appropriate velocity range for level 1. The flow fields in Appendix C also show the large spatial support of this method, which is caused by the cascaded convolution of the Gaussian low-pass smoothing and the band-pass Gabor filters. In this case we obtained 2-d velocity estimates near the centre of the square.

Lastly we note that the square data provides a clear way of examining the normal velocity estimates as distinct from the eventual 2-d velocity estimates. These results are reported in Table 4.5. Of the techniques we considered, those of Lucas and Kanade, Heeger, Waxman et al. and Fleet and Jepson produce both full and normal (component) velocity estimates explicitly. The method of Lucas and Kanade provides two sources of normal velocities, namely, one from the gradient constraint directly (2.3) with the gradient magnitude as an implicit confidence weighting and the second from the LS minimization in (2.8) when the aperture problem prevails (i.e. when the eigenvalues of (2.9), $\lambda_1 \geq \lambda_2$, satisfy $\lambda_1 \geq \tau$ but $\lambda_2 < \tau$ for the confidence threshold τ). Tables 4.5 report normal velocities from both sources.

The phase-based technique of Fleet and Jepson often produces several normal velocity estimates at a single image location. Table 4.5 reports density as two quantities: the first gives the density of positions where one or more component velocities is recovered and the second (in parenthesis) gives the average number of component velocities at a single point.

Many of the other techniques could be modified to produce normal flows as well: for example, with Anandan's approach we could use $c_{max} \gg c_{min}$ to indicate a normal velocity. In Singh's approach, we could use large and small eigenvalues of the covariance matrix in (2.20) to discriminate between full and normal velocity (like our implementation of the Lucas and Kanade approach). However, we have not yet made these modifications as we did not find these confidence measures to be reliable.

4.4 Realistic Synthetic Data

We now turn to the more realistic synthetic sequences, namely the **Translating** and **Diverging Tree** sequences and the **Yosemite** sequence, the results of which are presented in Tables 4.6 – 4.10. Error statistics of normal (component) velocity estimates computed from a subset of the techniques on the **Diverging Tree** sequence are given in Table 4.9. Other quantities of interest, including error histograms and flow fields, are given Appendices D through F.

The general behaviour of the differential techniques is similar to that observed above. It is especially interesting to see the improvement of our modified version of the Horn and Schunck algorithm versus the original method, which we attribute to the image presmoothing and the improved numerical differentiation. One can also see that for reasonably smooth motion fields, such as those in the **Translating** and **Diverging Tree** sequences, that the smoothness constraint used to integrate the normal constraints performs well. The constraint on gradient magnitude provides one way to identify regions within which estimates may be more reliable. Interestingly, we also found with these sequences that larger values of the smoothness parameter (e.g. $\lambda = 100$ as suggested by Horn and Schunck) yielded somewhat poorer results.

However, despite the improved performance of Horn and Schunck's method here, the results remain less accurate than those of Lucas and Kanade's method, which shares the same gradient estimates, and differs only in the method used to combine normal constraints. In particular,

Technique	Average Error	Standard Deviation	Density
Horn and Schunck (original)	38.72°	27.67°	100%
Horn and Schunck (original) $\|\|\nabla I\|\| \geq 5.0$	32.66°	24.50°	55.9%
Horn and Schunck (modified)	2.02°	2.27°	100%
Horn and Schunck (modified) $\|\|\nabla I\|\| \geq 5.0$	1.89°	2.40°	53.2%
Lucas and Kanade ($\lambda_2 \geq 1.0$)	0.66°	0.67°	39.8%
Lucas and Kanade ($\lambda_2 \geq 5.0$)	0.56°	0.58°	13.1%
Uras et al. (unthresholded)	0.62°	0.52°	100%
Uras et al. ($det(H) \geq 1.0$)	0.46°	0.35°	41.8%
Nagel	2.44°	3.06°	100%
Nagel $\|\|\nabla\|\|_2 \geq 5.0$	2.24°	3.31°	53.2%
Anandan	4.54°	3.10°	100%
Singh (Step 1, $n = 2$, $w = 2$)	1.64°	2.44°	100%
Singh (Step 1, $n = 2$, $w = 2$, $\lambda_1 \leq 5.0$)	0.72°	0.75°	41.4%
Singh (Step 2, $n = 2$, $w = 2$)	1.25°	3.29°	100%
Singh (Step 2, $n = 2$, $w = 2$, $\lambda_1 \leq 0.1$)	1.11°	0.89°	99.6%
Heeger (level 0)	8.10°	12.30°	77.9%
Heeger (level 1)	4.53°	2.41°	57.8%
Waxman et al. ($\sigma_f = 2.0$)	6.66°	10.72°	1.9%
Fleet and Jepson ($\tau = 2.5$)	0.32°	0.38°	74.5%
Fleet and Jepson ($\tau = 1.25$)	0.23°	0.19°	49.7%
Fleet and Jepson ($\tau = 1.0$)	0.25°	0.21°	26.8%

Table 4.6: *Summary of the* **Translating Tree** *2D Velocity Results.*

our confidence measure (based on the eigenvalues of the normal equations in (2.9)) appeared to perform very well, allowing us to extract subsets of accurate 2-d velocities. One can see from Tables 4.6 and 4.7 that by changing the confidence threshold from $\lambda_2 \geq 1.0$ to $\lambda_2 \geq 5.0$ we obtained better accuracy, but at the cost of a significant reduction in the measurement density.[9]

It is also worthwhile at this point to comment on another observation made during the testing of these gradient-based methods and some changes that occurred since we reported our preliminary results in [8]. Our initial implementation quantized the Gaussian smoothed image sequence with 8-bit/pixel for storage, prior to the subsequent gradient computation and least-squares minimization, causing relatively noisy derivative estimates. Compared to the results in Tables 4.6 and 4.7, which were based on a floating-point representation of the filter outputs, we found that when this quantization error is introduced the errors for Lucas and Kanade's method grew approximately 40–50%, and those produced by Horn and Schunck's method became several times larger. This suggests that Horn and Schunck's method of combining normal constraints (the global smoothness constraint) is significantly more sensitive to noise than the local least-

[9] The **Translating** and **Diverging Tree** sequences have also been used by Simoncelli [52] with his gradient-based technique and by Haglund [26] with his energy-based technique. Both get results comparable to those reported here with the Lucas and Kanade method.

Technique	Average Error	Standard Deviation	Density
Horn and Schunck (original)	12.02°	11.72°	100%
Horn and Schunck (original) $\|\|\nabla I\|\| \geq 5.0$	8.93°	7.79°	54.8%
Horn and Schunck (modified)	2.55°	3.67°	100%
Horn and Schunck (modified) $\|\|\nabla I\|\| \geq 5.0$	2.50°	3.89°	32.9%
Lucas and Kanade ($\lambda_2 \geq 1.0$)	1.94°	2.06°	48.2%
Lucas and Kanade ($\lambda_2 \geq 5.0$)	1.65°	1.48°	24.3%
Uras et al. (unthresholded)	4.64°	3.48°	100%
Uras et al. ($det(H) \geq 1.0$)	3.83°	2.19°	60.2%
Nagel	2.94°	3.23°	100.0%
Nagel $\|\|\nabla I\|\|_2 \geq 5.0$	3.21°	3.43°	53.5%
Anandan (frames 19 and 21)	7.64°	4.96°	100%
Singh (Step 1, $n = 2$, $w = 2$, $N = 4$)	17.66°	14.25°	100%
Singh (Step 1, $n = 2$, $w = 2$, $N = 4$, $\lambda_1 \leq 5.0$)	7.09°	6.59°	3.3%
Singh (Step 2, $n = 2$, $w = 2$, $N = 4$)	8.60°	5.60°	100%
Singh (Step 2, $n = 2$, $w = 2$, $N = 4$, $\lambda_1 \leq 0.1$)	8.40°	4.78°	99.0%
Heeger	4.49°	3.10°	74.2%
Waxman et al. $\sigma_f = 2.0$	11.23°	8.42°	4.9%
Fleet and Jepson ($\tau = 2.5$)	0.99°	0.78°	61.0%
Fleet and Jepson ($\tau = 1.25$)	0.80°	0.73°	46.5%
Fleet and Jepson ($\tau = 1.0$)	0.73°	0.46°	28.2%

Table 4.7: *Summary of the* **Diverging Tree** *2D Velocity Results.*

squares method used by Lucas and Kanade, since other aspects of the techniques were identical.

The second-order technique of Uras et al. produced good results (both accurate and dense) on the **Translating Tree** sequence, but its results on the next two sequences are poorer by comparison, for which we can suggest two reasons. First, as discussed in Section 2.1, while the first-order (gradient) constraint equation is valid for smooth deformations of the input (including affine deformations), the second-order constraints are based on the conservation of the intensity gradient, and are (strictly speaking) therefore invalid for rotation, dilation and shear. This is one of the main differences between the **Translating Tree** sequence and the other two. A second factor is the amount of aliasing in the **Yosemite** sequence, which makes accurate second-order differentiation difficult.

Finally, we obtained good results for the regularization approach of Nagel.[10] The use of $||\nabla I||_2$ as a confidence measure was not entirely successful here, using $||\nabla I||_2 > 1.0$ produced only slightly more accurate but considerably less dense results. Interestingly, with the **Diverging Tree** sequence this threshold actually produced poorer results. We also note that for much of our image data the 2^{nd} order derivatives of intensity and velocity are small, in which case Nagel's method yields similar results to Horn and Schunck's.

[10] This contrasts with the results reported in a previous version of this technical report where a different method of computing intensity and velocity derivatives was employed.

With respect to matching techniques, observe that although both methods produced reasonably good results on the **Translating Tree** input, Singh's results are somewhat better than Anandan's. This is true even of the first stage of Singh's algorithm that is concerned mainly with locating SSD minima. One reason for this is the larger neighbourhood support in Singh's algorithm; for example, when we used 3×3 regions ($n = 1$ and $w = 1$) instead of 5×5 regions for Singh's method the errors increased (from those reported in Table 4.6) to $2.13° \pm 5.15°$ for stage 1 and $1.35° \pm 1.68$ for stage 2.

Furthermore, we did not find Anandan's confidence measures based on c_{min} and c_{max} to be reliable. By comparison, we found for Singh's method that the inverse eigenvalues of the covariance matrix at stage 1 do provide a useful confidence measure, but the inverse eigenvalues of the covariance matrix at stage 2 were ineffective – small changes in a threshold based on the largest eigenvalue dramatically changed the density of estimates. The lack of good confidence measures makes it difficult to evaluate these methods.

It is also interesting to observe that both matching techniques produced poorer results when applied to the **Diverging Tree** sequence than with the **Translating Tree** sequence. Singh's results are about an order of magnitude worse, especially at step 1 of the algorithm. Although some of the error may be due to aliasing and the confusion between normal and 2-d velocities, we find that most of the increase in error is due to subpixel inaccuracy. The **Translating Tree** sequence has velocities very close to integer displacements, while the **Diverging Tree** sequence has a wide range of velocities. We find that velocities corresponding to noninteger displacements often have errors two to three time larger than those corresponding to integer displacements (provided the aperture problem can be overcome). In many cases, this is due to the sharpness of peaks in the mass distribution formed in (2.18); that is, they are so sharp relative to integer sampling of the SSD surface that they are sometimes missed, and the resulting sampled distribution appears very broad.

There may be several possible ways to circumvent this problem. One might use coarser temporal sampling so that subpixel errors are small relative to actual displacements, but this involves a host of additional problems for matching. Alternatively, a coarse-fine approach with warping may yield some improvement. In any case, it would be useful to have a model for the expected behaviour of such errors which may be incorporated into confidence measures.

The results reported here for Heeger's method applied to the **Translating Tree** sequence are from level 1 of the pyramid because the input speeds coincided with its velocity range of 1.25–2.5 pixels/frame. Level 0 was used for **Diverging Tree** sequence since most of its speeds were below 1.25 pixel/frame. For the **Yosemite** sequence velocity estimates were computed at all three levels of the pyramid and then combined so that, of the three, the velocity estimate from the level of the pyramid whose speed range was consistent with the true motion field was chosen. 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 (with errors of $13.75° \pm 23.06°$) than those reported in Table 4.10.

Of all the techniques we applied to the synthetic data, the phase-based method of Fleet and Jepson [19] produced the most consistently accurate results. We found that the phase stability threshold is a reliable indication of performance in most cases. Table 4.9 also shows that the normal constraints derived from phase information are often less biased than those from other methods such as gradient-based approaches.

Although, the phase-based method performs extremely well on the **Translating** and **Diverging Tree** sequences, it is clear from Table 4.10 that it is not significantly better than

Technique	Average Normal Error	Standard Deviation	Density
Lucas and Kanade (LS) ($\lambda_1 \geq 1.0$)	0.29°	0.36°	47.9%
Lucas and Kanade (LS) ($\lambda_1 \geq 5.0$)	0.24°	0.27°	61.6%
Lucas and Kanade (Raw) ($\|\|\nabla I\|\| \geq 0.0$)	0.58°	1.19°	100%
Lucas and Kanade (Raw) ($\|\|\nabla I\|\| \geq 5.0$)	0.25°	0.29°	53.2%
Heeger (level 0)	0.74°	3.86°	22.1%
Heeger (level 1)	2.21°	2.03°	9.2%
Waxman et al. $\sigma_f = 2.0$	14.77°	13.40°	12.1%
Fleet and Jepson $\tau = 1.25$	−0.06°	0.23°	71.4% (2.1)
Fleet and Jepson $\tau = 2.5$	−0.01°	0.51°	89.6% (5.8)

Table 4.8: *Summary of* **Translating Tree** *Normal/Component Velocity Results.*

Technique	Average Normal Error	Standard Deviation	Density
Lucas and Kanade (LS) ($\lambda_1 \geq 1.0$)	1.00°	0.83°	36.0%
Lucas and Kanade (LS) ($\lambda_1 \geq 5.0$)	0.86°	0.70°	49.0%
Lucas and Kanade (Raw) ($\|\|\nabla I\|\| \geq 5.0$)	0.77°	0.85°	53.5%
Heeger	1.92°	3.18°	25.8%
Waxman et al. $\sigma_f = 2.0$	8.26°	11.16°	8.8%
Fleet and Jepson $\tau = 1.25$	−0.04°	0.78°	61.0% (2.1)
Fleet and Jepson $\tau = 2.5$	−0.11°	1.30°	77.3% (5.3)

Table 4.9: *Summary of* **Diverging Tree** *Normal/Component Velocity Results.*

Technique	Average Error	Standard Deviation	Density
Horn and Schunck (original)	32.43°	30.28°	100%
Horn and Schunck (original) $\|\|\nabla I\|\| \geq 5.0$	25.41°	28.14°	59.6%
Horn and Schunck (modified)	11.26°	16.41°	100%
Horn and Schunck (modified) $\|\|\nabla I\|\| \geq 5.0$	5.48°	10.41°	32.9%
Lucas and Kanade ($\lambda_2 \geq 1.0$)	4.10°	9.58°	35.1%
Lucas and Kanade ($\lambda_2 \geq 5.0$)	3.05°	7.31°	8.7%
Uras et al. (unthresholded)	10.44°	15.00°	100%
Uras et al. ($det(H) \geq 1.0$)	6.73°	16.01°	14.7%
Nagel	11.71°	10.59°	100%
Nagel $\|\|\nabla I\|\|_2 \geq 5.0$	6.03°	11.04°	32.9%
Anandan	15.84°	13.46°	100%
Singh (Step 1, $n = 2$, $w = 2$)	18.24°	17.02°	100%
Singh (Step 1, $n = 2$, $w = 2$, $\lambda_1 \leq 5.0$)	16.29°	25.70°	2.2%
Singh (Step 2, $n = 2$, $w = 2$)	13.16°	12.07°	100%
Singh (Step 2, $n = 2$, $w = 2$, $\lambda_1 \leq 0.1$)	12.90°	11.57°	97.8%
Heeger (combined)	11.74°	19.04°	44.8%
Heeger (level 0)	20.89°	34.26°	64.2%
Heeger (level 1)	10.51°	12.11°	15.2%
Heeger (level 2)	11.51°	11.83°	2.4%
Waxman et al. $\sigma_f = 2.0$	20.32°	20.60°	7.4%
Fleet and Jepson ($\tau = 1.25$)	4.95°	12.39°	30.6%
Fleet and Jepson ($\tau = 2.5$)	4.29°	11.24°	34.1%

Table 4.10: *Summary of* **Yosemite** *2D Velocity Results*

Technique	Average Normal Error	Standard Deviation	Density
Lucas and Kanade (LS) ($\lambda_1 \geq 1.0$)	9.88°	13.73°	56.1%
Lucas and Kanade (LS) ($\lambda_1 \geq 5.0$)	5.69°	10.47°	58.6%
Lucas and Kanade (Raw) ($\|\|\nabla\|\| \geq 0.0$)	7.94°	12.75°	100%
Lucas and Kanade (Raw) ($\|\|\nabla\|\| \geq 5.0$)	2.13°	4.90°	32.9%
Heeger (level 0)	−2.78°	18.07°	35.8%
Heeger (level 1)	2.46°	6.09°	5.7%
Heeger (level 2)	5.23°	9.53°	0.8%
Waxman et al. $\sigma_f = 2.0$	−6.98°	20.60°	7.8%
Fleet and Jepson $\tau = 1.25$	−5.29°	18.56°	54.8% (2.2)
Fleet and Jepson $\tau = 2.5$	−3.67°	15.35°	65.4% (3.7)

Table 4.11: *Summary of* **Yosemite** *Normal/Component Velocity Results.*

differential methods on the **Yosemite** sequence. There are several reasons for this: First, because only 15 frames were available in this sequence, we had to increase the tuning frequency of the filters to reduce the width of support (from 21 to 15 frames) and increase the frequency tuning of the filters, thereby pushing their pass-bands closer to the Nyquist rate. Because of their narrow bandwidths, this causes greater sensitivity to aliasing and corruption at high frequencies as compared with the Gaussians used by differential techniques. To compound this problem, as already stated this sequence contains a significant amount of aliasing in certain regions of the image.

Interestingly, for the **Yosemite** sequence we found that as the phase stability threshold τ increases, the 2-d velocity errors initially increase, but then begin to decrease significantly. We attribute this to the increasing number of component velocities available for 2-d velocity computations, increasing the robustness of the minimization slightly. Furthermore, although not reported here, considerable improvement can be achieved with a tighter constraint on the condition number in the LS system as reported in [22].

In fact, most techniques perform relatively poorly on this image sequence. This is due in part to the aliasing and in part to the occlusion boundaries. The major occlusion boundary that introduces error is of course the horizon. This is evident in the flow fields produced by several of the different techniques that are shown in Appendix F. If the sky is excluded from the error analysis, most techniques show improved performance. For example, the differential methods of Lucas and Kanade and Uras et al. improved from $4.10° \pm 9.58°$ and $6.73° \pm 16.01°$ to $2.80° \pm 3.82°$ and $3.37° \pm 3.37°$ respectively, and the phase-based method of Fleet and Jepson improved from $4.29° \pm 11.24°$ to $2.97° \pm 5.76°$. In all these cases the density of estimates is effectively unchanged.

4.5 Confidence Measures

One of our major discoveries in comparing techniques has been the importance of confidence measures, i.e. some means of determining the reliability and accuracy of the computed velocities. All techniques produce velocity estimates whose accuracy varies dramatically with the structure of the underlying signal and the 2-d motion. In reporting error statistics above, we used confidence measures as thresholds to extract subsets of velocity estimates. Those techniques that appear to perform well often do so because we are able to isolate the more reliable measurements. Confidence measures also prove useful to distinguish locations at which 2-d velocity versus normal velocity is measured.

To justify the use of these confidence measures it is important to examine error behaviour and the density of estimates as functions of the confidence measures, to ensure their reliability over a wide range of confidence values.[11] Appendix G shows the plots of the performance of several confidence measures on the **Diverging Tree** sequence. In what follows here we simply summarize our main results, concentrating on the techniques that produced reasonably good results, namely, those of Fleet and Jepson [19, 22], Lucas and Kanade [40, 39], Anandan [5, 6], Uras et al. [57] and Singh [53, 54].

With respect to first-order differential methods, there are several points of interest. We first reiterate that the weighted minimization used to estimate 2-d velocity from the normal constraints involves an implicit weighting of each normal constraint by the magnitude of its spatial gradient. In most cases this was found to correlate well with accuracy. As confidence measures for the 2-d velocity estimates we have used the trace of the normal matrix (2.9) as suggested by Simoncelli et al. [51] and a measure based solely on the magnitude of the smallest eigenvalue of (2.9), λ_2. In doing so we often observed that the smallest eigenvalue alone is the better measure of confidence. There are several possible reasons for this: First note that the occurrence of the aperture problem is signalled primarily in the smallest eigenvalue; the sum of the eigenvalues can be arbitrarily large while the system remains singular due to the aperture problem. Second, although significant errors in gradient measurement are manifested in smaller eigenvalues, there are other sources of error that are not, such as differences between the 2-d motion field and the velocity of level intensity contours.

With respect to second-order differential methods, Uras et al. suggested a confidence measure based on the condition number $\kappa(H)$ of the (spatial) Hessian of $I(\mathbf{x}, t)$. We have also examined the use of the determinant of the Hessian $det(H)$ which also reflects the magnitudes of the second derivatives. Although $\kappa(H)$ is useful in certain cases, we find that $det(H)$ is more consistently reliable, producing better results on the three realistic synthetic sequences tested in Section 4.4. We also observed similar behaviour with the four natural image sequences.

Anandan suggested the use of c_{max} and c_{min} as confidence measures based on the principal SSD curvatures. However, we did not find them to be reliable. Error often appeared independent of c_{min}, and occasionally increased when the estimates were thresholded with it. We believe the problem with using c_{min} as a threshold lies in the smoothing steps after processing each level of the Laplacian pyramid. Although large c_{min} and c_{max} values should indicate image areas where there is significant local structure that permits the aperture problem to be resolved, the smoothing sometimes negates this. As well, if errors occur at coarse scales, then displacement estimates at subsequent scales are generally poor, and the SSD structure is bound to be of little help.

Singh's method involved confidence measures based on covariance matrices at both stages of computation (S_c in stage 1 (2.20), and $[S_c^{-1} + S_n^{-1}]^{-1}$ in stage 2). Because larger values of the inverse eigenvalues should indicate greater confidence, the smallest inverse eigenvalue might be taken as a single confidence measure. Interestingly we find the eigenvalues of stage 1 to be more

[11] Note that we are not proposing that these estimates be used as thresholds to extract subsets of measurements in general. Rather, we imagine that the majority of the velocity estimates will often be retained along with their respective confidence values that could then be used as weights in subsequent computation.

useful than those of stage 2. In fact, we find little if any correlation between the magnitude of inverse eigenvalues at stage 2 with the accuracy of the estimates. Moreover, we find that the resulting confidence measures are very sensitive to the choice of k in (2.18). It is also interesting to reiterate that errors in Singh's matching method appeared higher for velocities midway between integer displacements. Ideally, the confidence measure should reflect this.

For the phase-based approach of Fleet and Jepson we used confidence thresholds on both the normal velocity estimates, and on the LS system used to estimate 2-d image velocity. As suggested by Fleet and Jepson, we find that their stability constraint is important, as well as constraints on the conditioning of the LS system. Both correlate well with errors and appear to produce consistently good results across all the sequences with fixed thresholds (with the stability constraint τ between 1.0 and 2.0 and the condition number threshold between 5 and 10). One problem with the phase-based method is that several different constraints are simultaneously available, and although Fleet and Jepson used them as thresholds, it would be better if they were combined in the form of a single confidence measure, rather than a set of thresholds.

4.6 Real Image Data

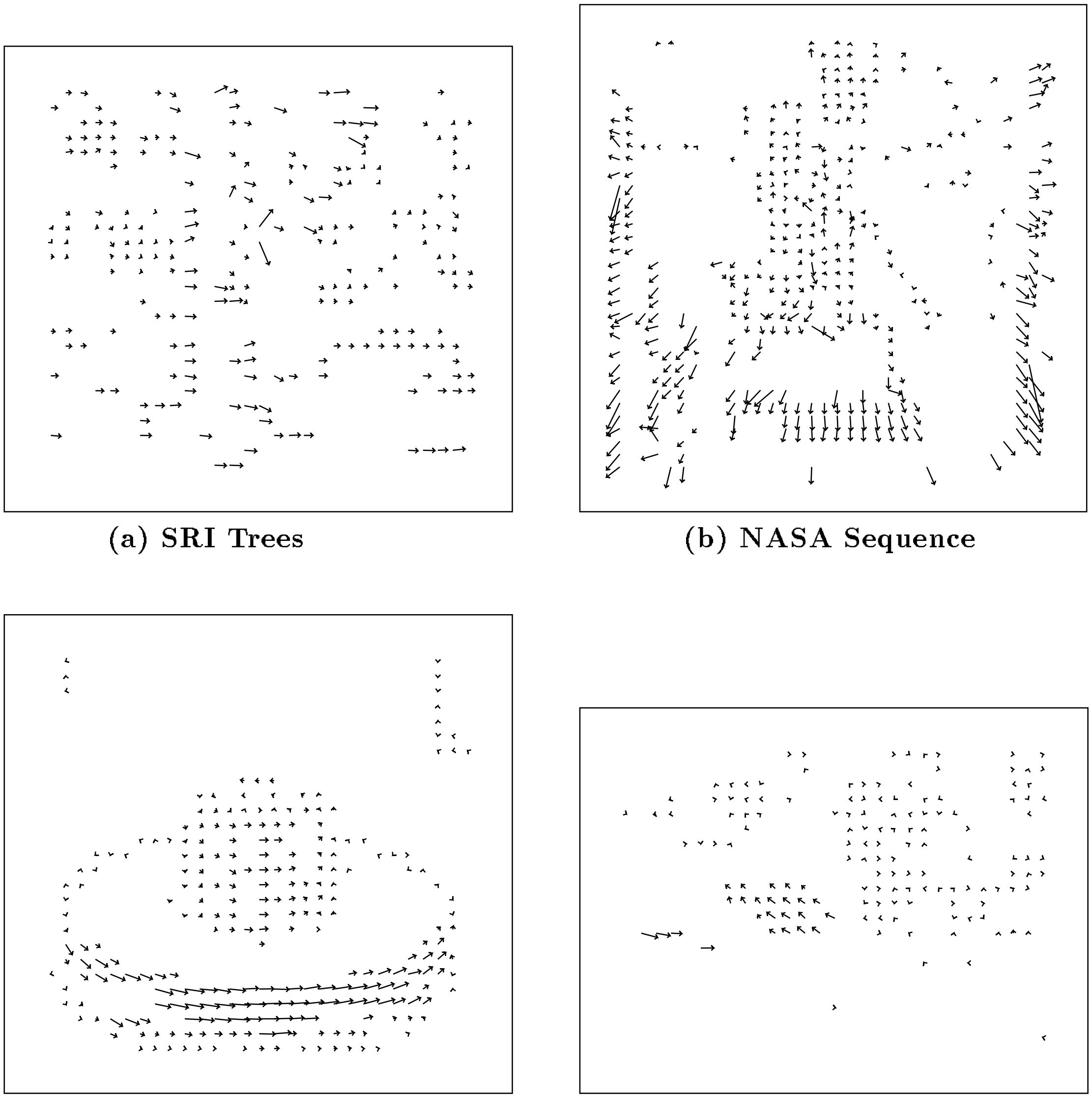
Finally, Figures 4.1 through 4.9 show subsampled versions of the flow fields produced by the various techniques when applied to the real image sequences shown in Figure 3.4. Parameters and confidence thresholds of the various methods have been kept the same as those used in the synthetic sequences above (except where noted) and are reported in the captions.

Although most of the results are self-evident, below we draw the reader's attention to several instances of behaviour already mentioned when discussing the synthetic data. With natural image sequences it is often difficult to see differences among the different techniques, since errors of 10% or 20% are not easily discerned at this resolution. Also, other errors are not always noticed, such as normal velocities mistaken for 2-d velocities.

Among the main problems outlined in Sections 4.2 – 4.5, for those methods that integrate normal constraints with global (regularization) smoothness constraints, is the lack of a confidence measure that allows one to distinguish a normal velocity estimate from 2-d velocity estimates. This point was most clear when comparing Horn and Schunck's method to the local explicit method of Lucas and Kanade. There is also clear evidence for this in the flow fields produced by these two methods in Figures 4.1 and 4.2, for example, in the **NASA** sequence just below the pop can in the bottom-middle and in the **Rubik** sequence at the bottom of the turntable). Similar errors are evident with other techniques that employ global smoothness assumptions, such as those of Nagel and Anandan.

The problems with matching methods, such as Singh's method, with slowly moving objects with subpixel velocities and some degree of dilation are evident in **NASA** sequence. Most velocities in this case were less than 1 pixel/frame, and subpixel accuracy is crucial to success on this sequence. Other problems that are evident with matching methods are the gross errors that arise from aliasing and problems choosing an incorrect local SSD minima in the first stage of processing.

The techniques that performed well, namely the differential and phase-based methods of Lucas and Kanade, Uras et al., and Fleet and Jepson, also produce good results on these sequences. In particular, note that although the method of Uras et al. produces a somewhat sparser set of estimates than other methods, the density is competitive. In the case of Fleet and Jepson, it is interesting to note the extremely good results through the ground plane toward the front of the **SRI** tree sequence compared with the problems caused by the occlusions in the trees above. In the case of the Hamburg Taxi sequence, the lower contrast moving objects



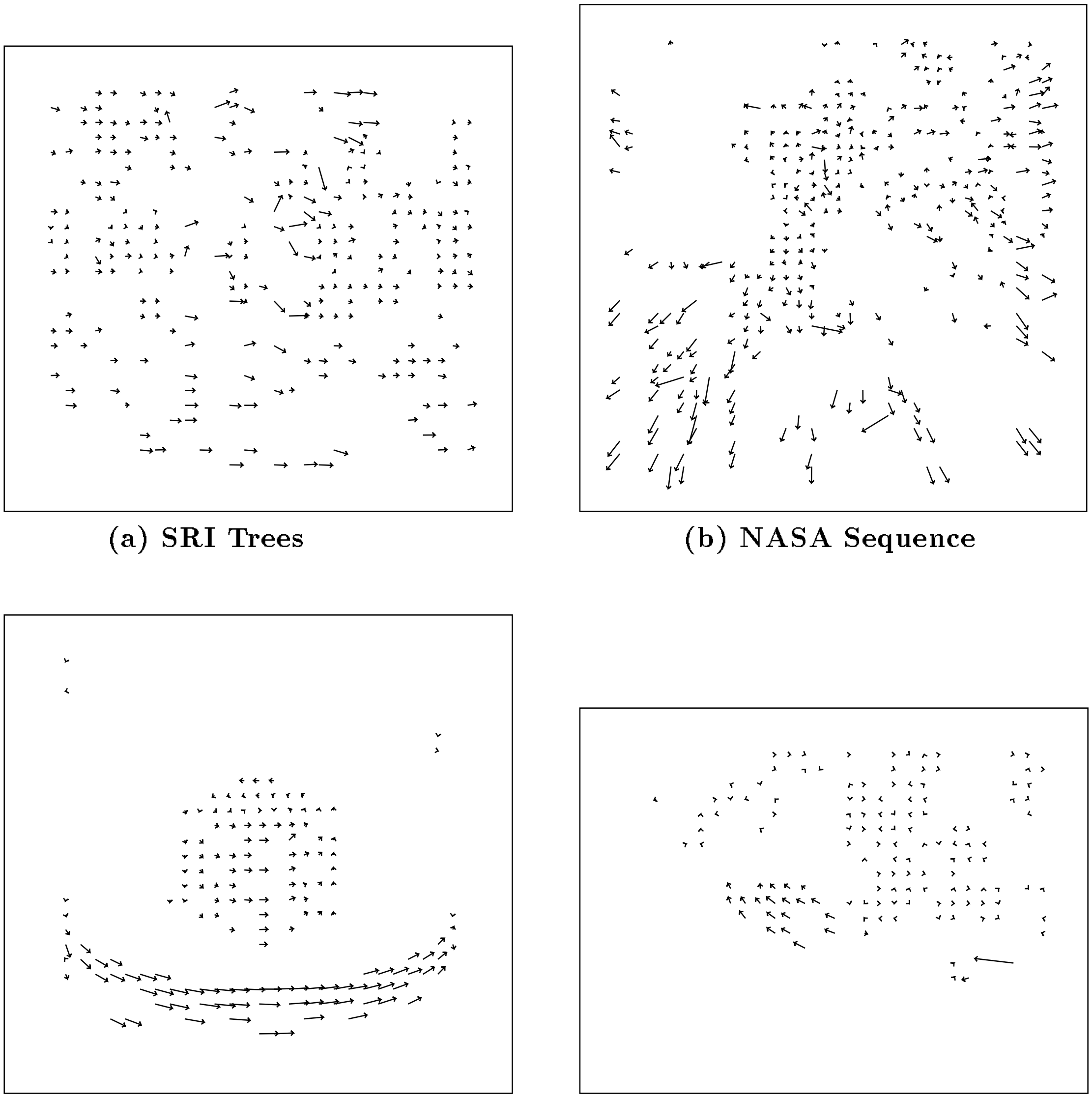
(a) SRI Trees

(b) NASA Sequence

(c) Rubik Cube

(d) Hamburg Taxi

Figure 4.1: *Flow fields for the modified* **Horn and Schunck** *technique (spatiotemporal Gaussian presmoothing and 4-point central differences) applied to real image data. The velocity estimates were thresholded using* $\| \nabla I \| \geq 5.0$.



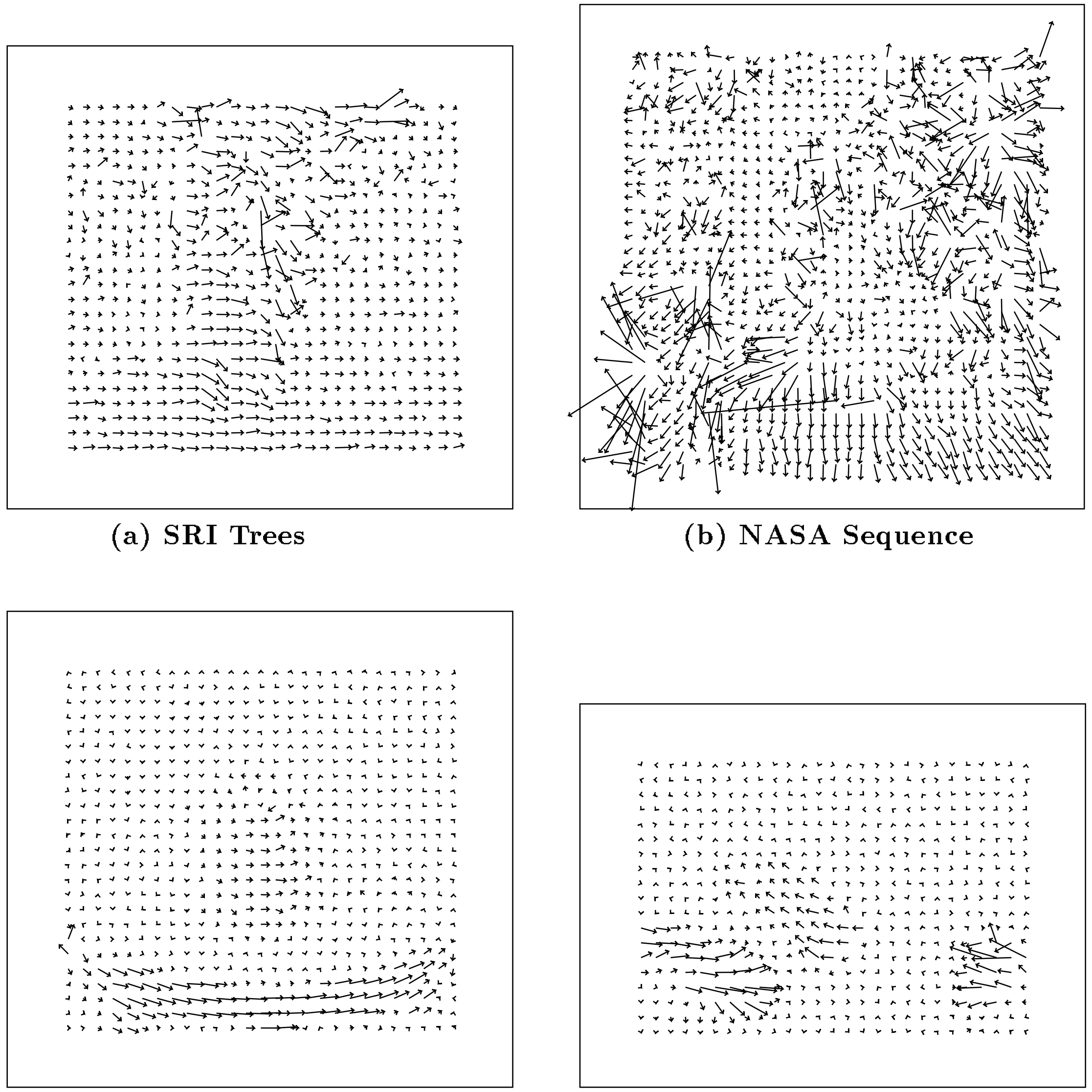
(a) SRI Trees

(b) NASA Sequence

(c) Rubik Cube

(d) Hamburg Taxi

Figure 4.2: *Flow fields for the* **Lucas and Kanade** *technique applied to real image data. All flow fields were produced with a threshold of* $\lambda_2 \geq 1.0$



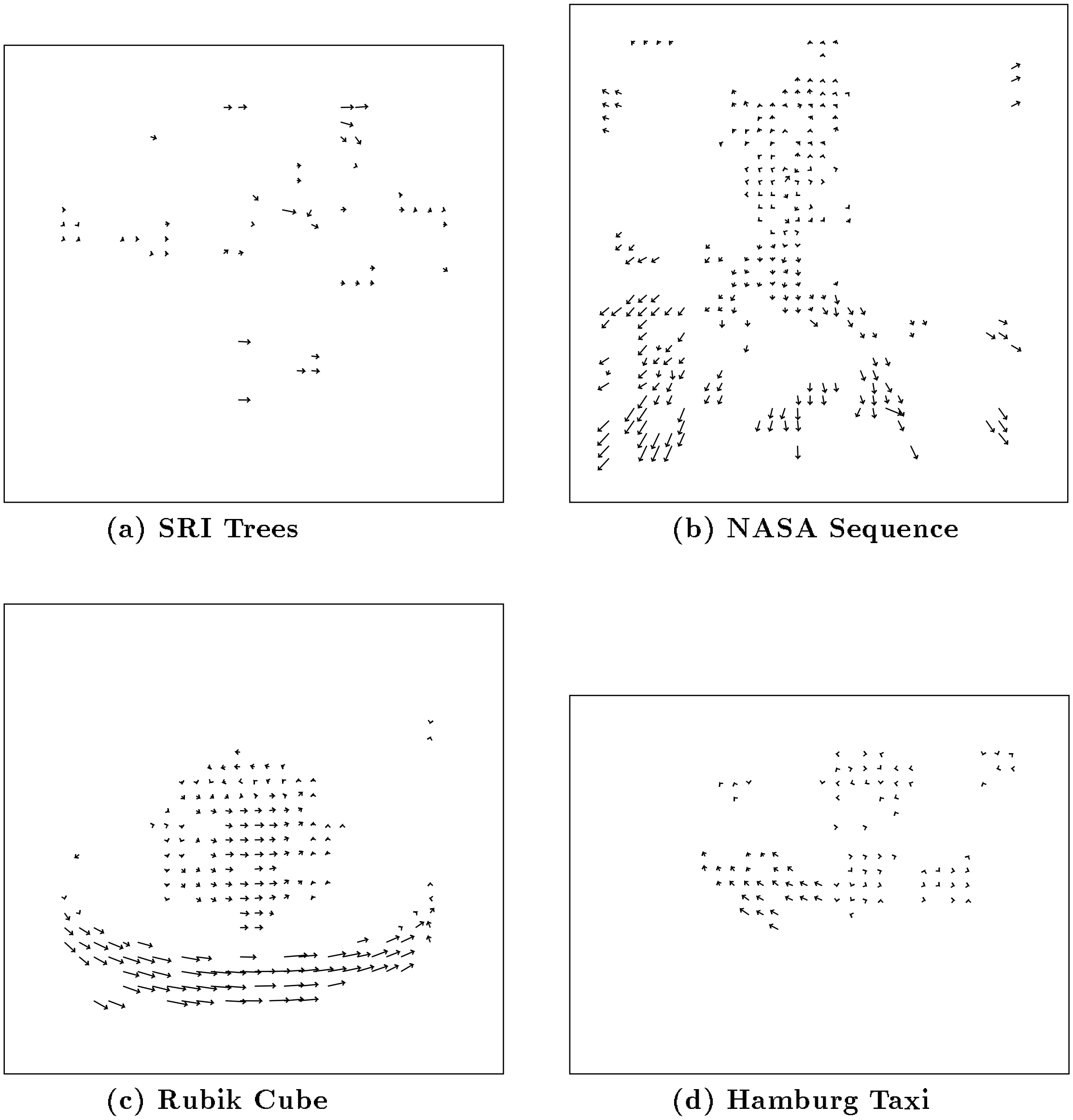
(a) SRI Trees

(b) NASA Sequence

(c) Rubik Cube

(d) Hamburg Taxi

Figure 4.3: *Flow fields for the technique of* **Nagel** *applied to real image data. With the real image sequences we found that* **Nagel's** *method required greater amounts of spatial presmoothing. Here we used a Gaussian filter with standard deviation of 3.0 in space and 1.5 in time. No thresholding was performed.*



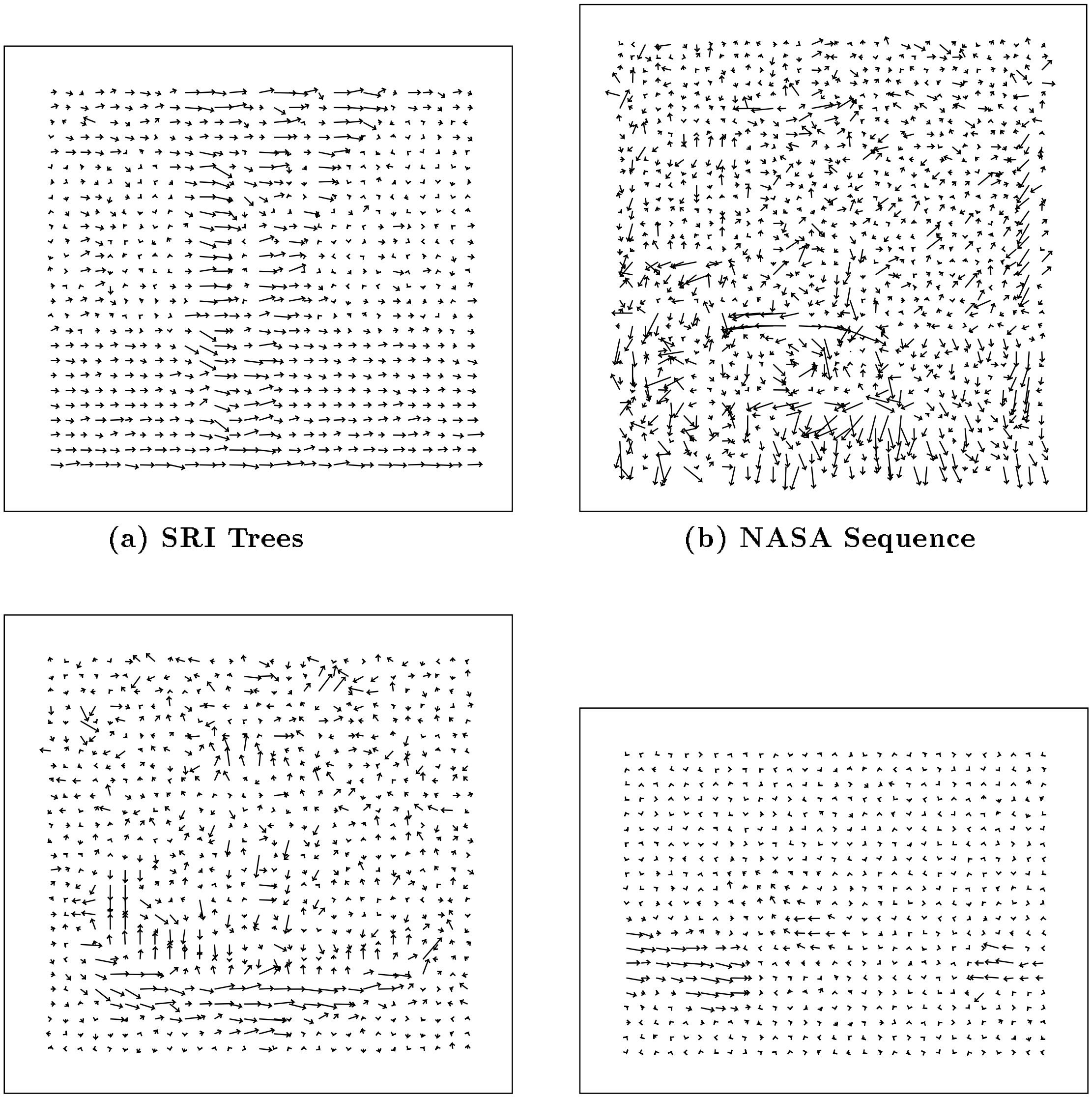
(a) SRI Trees

(b) NASA Sequence

(c) Rubik Cube

(d) Hamburg Taxi

Figure 4.4: *Flow fields for the* **Uras et al.** *technique applied to real image data. All flow fields were produced with a threshold of* $det(H) \geq 1.0$



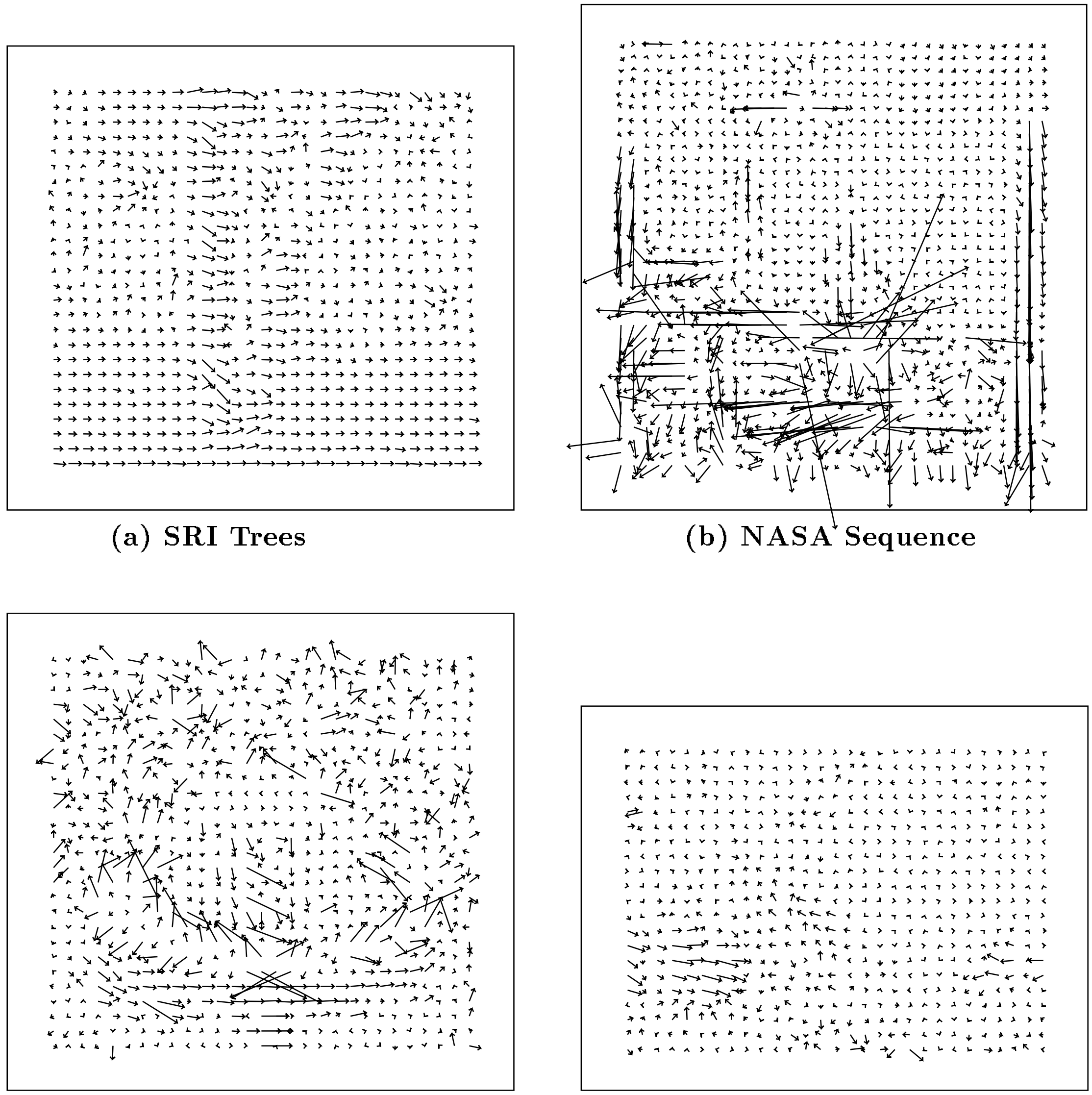
(a) SRI Trees

(b) NASA Sequence

(c) Rubik Cube

(d) Hamburg Taxi

Figure 4.5: *Flow fields for the technique of* **Anandan** *applied to real image data. The results are unthresholded.*



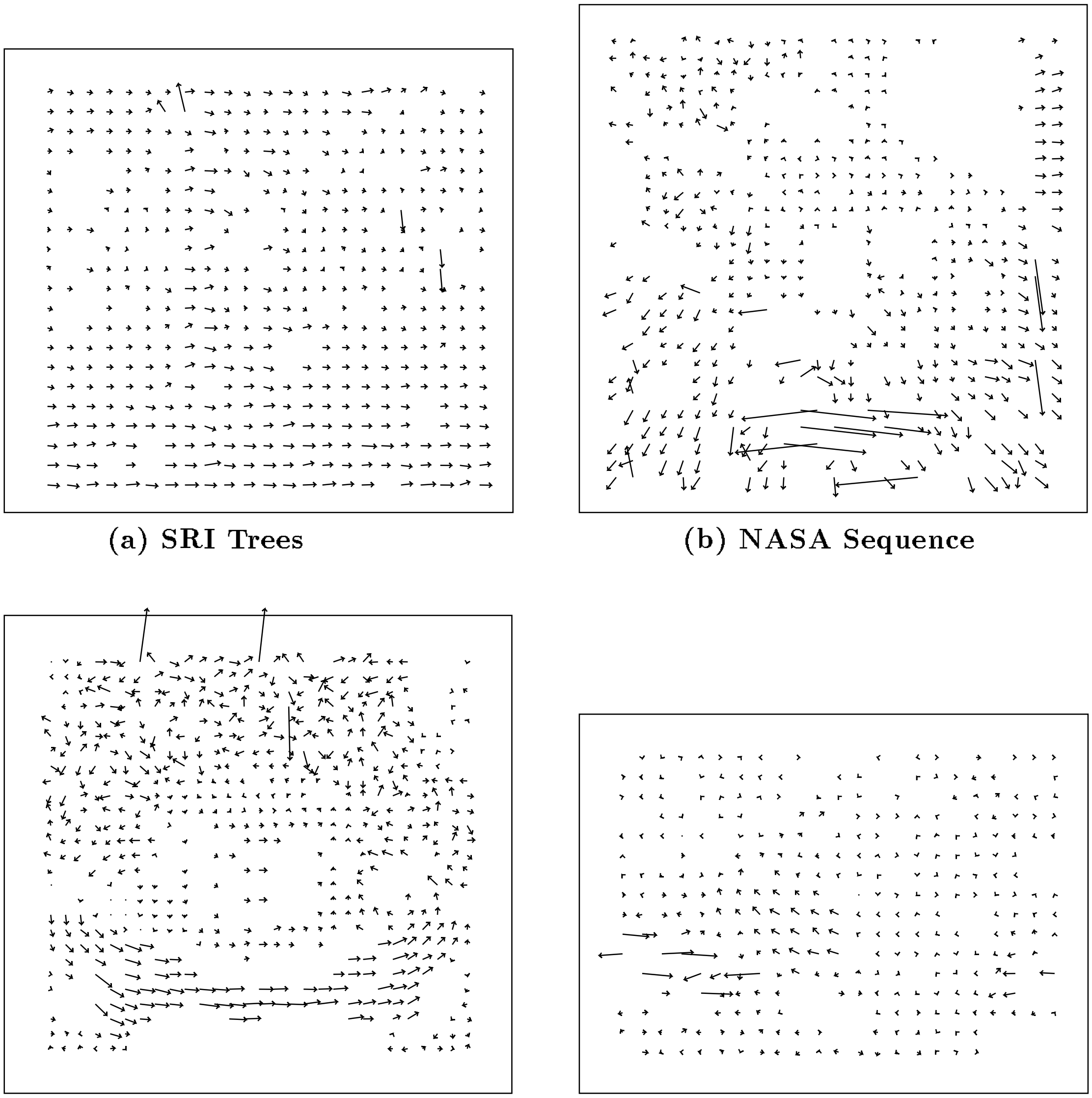
(a) SRI Trees

(b) NASA Sequence

(c) Rubik Cube

(d) Hamburg Taxi

Figure 4.6: *Flow fields produced by the technique of* **Singh** *applied to real image data. All flow fields are computed with* $n = 2$, $w = 2$ *and* $N = 4$. *No thresholding was employed.*



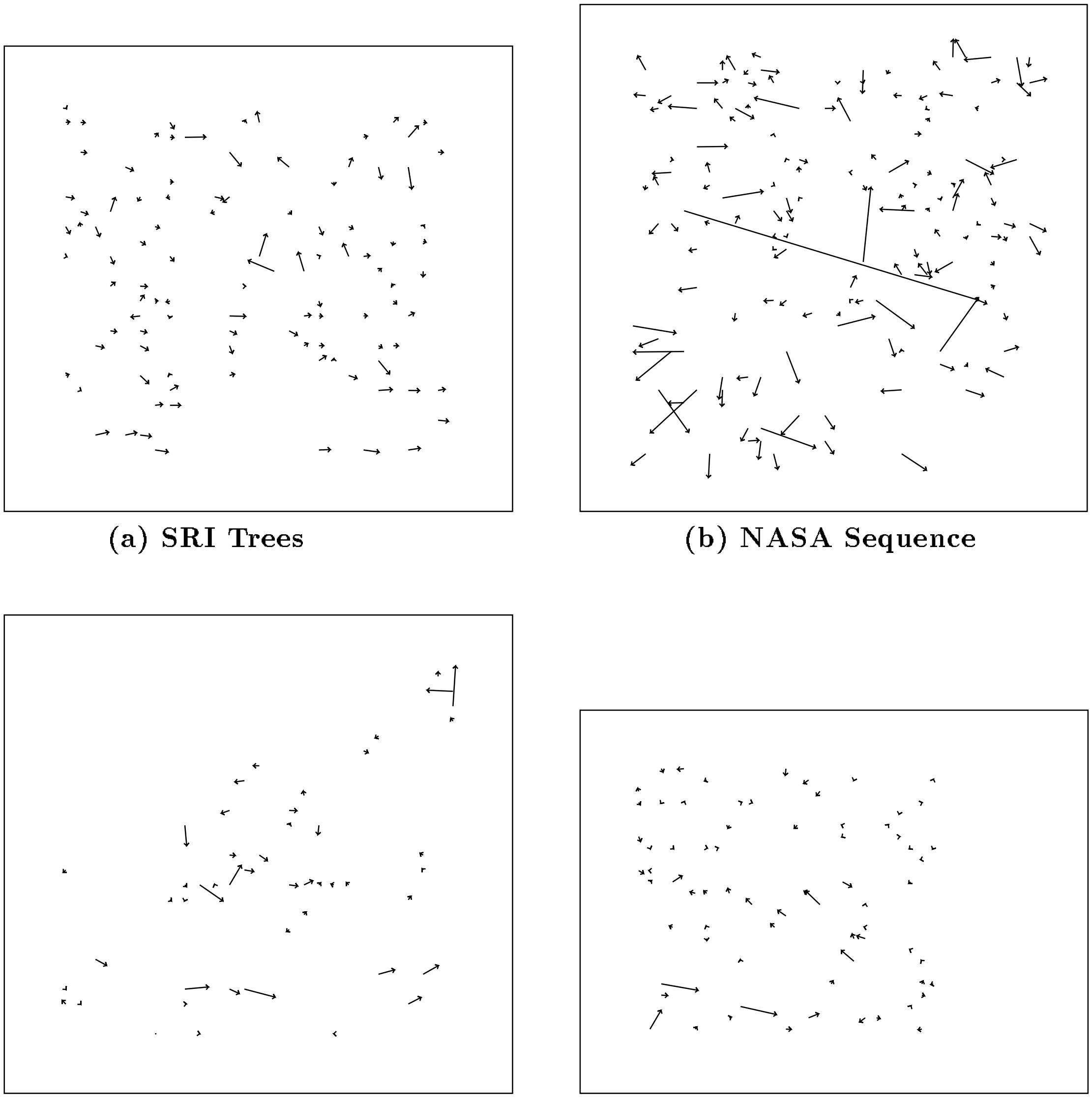
(a) SRI Trees

(b) NASA Sequence

(c) Rubik Cube

(d) Hamburg Taxi

Figure 4.7: *Flow fields for the technique of* **Heeger** *applied to real image data. The results shown for Heeger's method were based on all 3 levels of the Gaussian pyramid, choosing the estimates with speeds that are consistent from their respective levels of the pyramid (as discussed in Section 2.1). When consistent estimates are produced from more than one level, we choose the velocity estimate from the lowest level.*



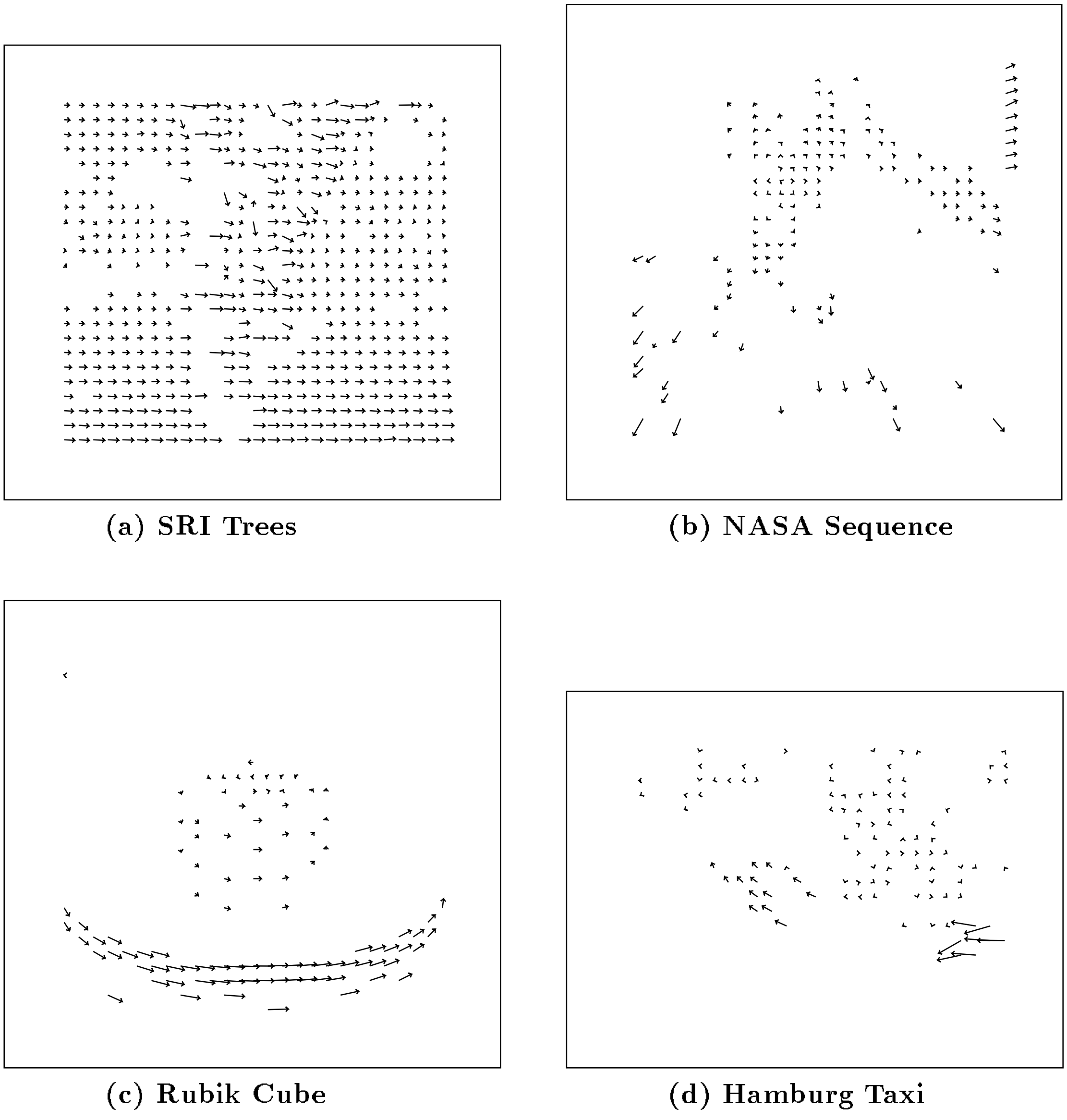
(a) SRI Trees

(b) NASA Sequence

(c) Rubik Cube

(d) Hamburg Taxi

Figure 4.8: *Flow fields for the technique of* **Waxman et al.** *applied to real image data. All flow fields were produced with a spatial standard deviation of 1.5.*



(a) SRI Trees

(b) NASA Sequence

(c) Rubik Cube

(d) Hamburg Taxi

Figure 4.9: *Flow fields for the* **Fleet and Jepson** *technique applied to real image data. All flow fields were produced with a threshold of* $\tau = 1.25$*. Other parameters were identical to those used by Fleet.*

appear quickly as the contrast threshold on the phase-based component measurements is relaxed slightly.

5 Summary

This paper compares the performance of a number of optical flow techniques, emphasizing the accuracy and density of measurements. We implemented nine techniques, including instances of differential methods, region-based matching, energy-based and phase-based techniques. They are the methods reported by Horn and Schunck [31], Lucas and Kanade [39, 40], Uras et al. [57], Nagel [43], Anandan [5, 6], Singh [53, 54], Heeger [29], Waxman et al. [61] and Fleet and Jepson [19, 22]. This allows a comparison of the performance of conceptually different techniques as well as comparisons among different instantiations of conceptually similar approaches. Both real and synthetic image sequences were used to test the techniques. In both cases, we chose sequences that are not severely corrupted by spatial or temporal aliasing.

Of these different techniques on the sequences we tested, we find that the most reliable were the first-order, local differential method of Lucas and Kanade, and the local phase-based method of Fleet and Jepson. Although not as consistent, the second-order differential method of Uras et al. also performed well. Only these approaches performed consistently well over all of the image sequences tested, with measures of confidence at the different stages of computation to detect and/or remove unreliable measurements. The lack of reliable confidence measures is a serious limitation of several of the other approaches.

With respect to the class of differential approaches tested we can draw several conclusions of general interest. The first concerns the importance of numerical differentiation and spatiotemporal smoothing. With both first and second-order differential techniques, the method of numerical differentiation is very important – differences between first-order pixel differencing and higher-order central-differences were very noticeable. Along the same lines, some degree of spatiotemporal presmoothing to remove small amounts of temporal aliasing and improve the subsequent derivative estimates had a marked effect on the quantitative accuracy of the resulting velocity estimates. The temporal smoothing was particularly useful. These factors are perhaps most evident in comparing the results obtained with Horn and Schunck's original algorithm with those of our modified version of it. For the data tested we found a spatio-temporal standard deviation of $\sigma = 1.5$ to be nearly optimal.

Another finding concerns the methods used to combine local differential constraints to obtain the 2-d velocity estimates. We found that the local explicit methods (i.e. local fits to constant or linear models of $\mathbf{v}$) were superior in both accuracy and computational efficiency to global smoothness constraints (with energy functionals that penalize a lack of smoothness), used by Horn and Schunck [31] and Nagel [42]. We also found the local methods to be more robust with respect to errors in gradient measurement caused by quantization noise. A clear example of the difference between the two approaches is apparent in the different errors produced the Lucas and Kanade method with those of our modified version of the Horn and Schunck method, since they share the same spatiotemporal derivative estimates. One of the main reasons for this distinction concerns the existence of a confidence measure to distinguish estimates of normal velocity from 2-d velocity. In the case of Lucas and Kanade's method, we found that the size of the smallest eigenvalue of the normal equations in (2.9) was one such reliable measure. By contrast, we did not find a similarly good confidence measure for Horn and Schunck's method.

Finally we found that, contrary to much of the literature, second-order differential methods (e.g. [56, 57]) are capable of producing accurate and relatively dense measurements of 2-d velocity. Moreover, the determinant of the (spatial) Hessian $I(\mathbf{x}, t)$ was a reasonably good confidence measure, and significantly more effective than its condition number (suggested by Uras et al. [57]). One problem with this technique however, appears to be its consistency. While it produced good results with predominantly translational image sequences, it appears to degrade faster than first-order techniques as the amount of higher-order geometric deformation in the input (e.g. dilation) increases. This is evident when comparing the results from the **Translating Tree** and **Diverging Tree** sequences. As discussed above, this problem is consistent with the underlying assumptions of the approach.

We now turn to the matching techniques, both of which produced results that were generally poorer than those from the better differential methods. One of the main problems we find with the SSD-based matching techniques is their ability to estimate sub-pixel displacements. With image translation and higher speeds they appear to perform well, but when the motion field involves small velocities with a significant dilational component the estimated displacements are often poor. In these cases it appears that SSD-based estimates of displacements are more accurate with integer displacements than subpixel velocities.

As a result of the relatively poor displacement estimates from the SSD minimization, the neighbourhood smoothness constraints employed by both Singh and Anandan are important to the success of these methods. At the same time, however, we found that the confidence measures suggested for both approaches were not very effective. The confidence measures suggested by Singh appeared to work somewhat better than those of Anandan's technique, in that they were generally correlated with the velocity errors. A problem in Anandan's approach, like that of Horn and Schunck was the inability to distinguish normal from 2-d estimates. In Singh's technique, they were more effective for step 1 of the computation than for the final velocity estimates of step 2, where they were largely ineffective. While matching techniques did not produce the most accurate velocity estimates among the techniques we examined, it should be restated that, as compared to the relatively large temporal duration of support used by the most successful techniques, these matching approaches used either 2 or 3 frames only.

The final techniques considered include energy-based techniques and phase-based approaches. Although there exist a number of interesting energy-based approaches, we have tested just one in this paper, namely the approach of Heeger [29]. Our results suggest that this technique is not as reliable as several of the other techniques considered. Although not reported in detail here we found that the original nonlinear optimization suggested by Heeger to solve (2.28) was extremely sensitive to initial conditions and did not produce reliable results. Our implementation of a parallel search method was better, but still left much to be desired; of course, in part this may be due to our implementation. It appears however that the effort needed to solve the optimization problem, combined with the assumptions underlying the approach (e.g. translating white noise) will make this approach difficult to employ.

The phase-based approach of Fleet and Jepson [22, 19] produced the most accurate results overall. However, there are several issues worth noting for our implementation of this technique. First, we find that this technique is sensitive to temporal aliasing in the image sequences because of the frequency tuning of the filters. A second issue concerns the potential number of confidence measures. Fleet and Jepson proposed several constraints on phase stability and signal contrast (SNR) to weed out poor normal velocity estimates. It would be useful to have these combined into a single confidence measure that would facilitate a more general weighted LS solution to the 2-d velocities. A third problem with our current implementation of the phase-based is its high computational load. Like Heeger's method and other frequency-based methods, it involves a large number of filters, which at present is the main computational expense. However, we expect that with the appropriate hardware in the near future the filtering should cease to be a severe limitation, and all these techniques could be implemented at frame-rates. It is also important to note that all our filter outputs were stored in floating point and were not subsampled (except in cases involving the Laplacian pyramid). More efficient encodings of the filter output should be possible with subsampling and quantization of the filter outputs as in [19] with only slight reductions in accuracy.

Finally, it is important to restate and qualify the conditions under which these tests were performed. First, we assumed that temporal aliasing was not a severe problem and that intensity (or filtered versions) were differentiable. As discussed earlier, if temporal aliasing is severe, then other approaches must be considered, such as coarse-to-fine control strategies. Second, we have considered relatively simple image sequences, without large amounts of occlusion, specularities, multiple motions, etc. and our quantitative measures of performance should be taken as lower bounds on the expected accuracy under more general conditions. Third, most of the implementations considered here involved only one scale of filtering, and would produce better results with multi-scale implementations. This is true of most techniques, including those of Lucas and Kanade [39, 40] and the phase-based approach of Fleet and Jepson [19, 22].

Acknowledgements: We thank the authors of the techniques examined here who shared their insights and helped get the programs running. This work has been supported in part by NSERC Canada, the Government of Ontario (through ITRC centres) and the Government of Canada (through IRIS).

References

[1] Adelson E.H. and Bergen J.R. (1985) Spatiotemporal energy models for the perception of motion. *J. Opt. Soc. Am.* A2, pp. 284-299

[2] 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

[3] Aloimonos J. and Brown C.M. (1984) Direct processing of curvilinear sensor motion from a sequence of perspective images. *Proc. Workshop on Computer Vision*, Annapolis, pp. 72-77

[4] Aloimonos Y. and Z. Duric (1992) Active egomotion estimation: a qualitative approach. *Proc. ECCV*, Ligure, Italy, pp. 497-510

[5] Anandan P. (1987) *Measuring Visual Motion from Image Sequences.* PhD dissertation, COINS TR 87-21, Univ. of Massachusetts, Amherest, MA

[6] Anandan P. (1989) A computational framework and an algorithm for the measurement of visual motion. *Int. J. Comp. Vision* 2, pp. 283-310

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

[8] Barron, J.L., Fleet, D.J., Beauchemin, S.S., and Burkitt, T. (1992) Performance of optical flow techniques. *IEEE Conf. CVPR*, Champaign, June, pp. 236-242

[9] Barron J.L. Jepson A.D. and Tsotsos J.K. (1990) The feasibility of motion and structure from noisy time-varying image velocity information. *Int. J. Comp. Vision* 5, pp. 239-269

[10] Bigun J,. Granlund, G. and Wiklund J. (1991) Multidimensional orientation estimation with applications to texture analysis and optical flow. *IEEE Trans. PAMI*, 13, pp. 775-790

[11] Beaudet P.R. (1978) Rotationally invariant image operators. *Proc. ICPR*, pp. 579-583

[12] Burt P.J. and Adelson E.H. (1983) The Laplacian pyramid as a compact image code. *IEEE Trans. on Communications* 31, pp. 532-540

[13] Burt P.J., Yen C. and Xu X. (1983) Multiresolution flow-through motion analysis. *Proc. IEEE CVPR*, Washington, pp. 246-252

[14] Buxton B. and Buxton H. (1984) Computation of optic flow from the motion of edge features in image sequences. *Image and Vision Computing* 2, pp. 59-74

[15] Cippola R. and Blake A. (1992) Surface orientation and time to contact from image divergence and deformation. *Proc. ECCV* Ligure, Italy, pp. 187-202

[16] Duncan J.H. and Chou T.C. (1988) Temporal edges: The detection of motion and the computation of optical flow. *Proc. IEEE ICCV*, Tampa, pp. 374-382

[17] Dutta, R., Manmatha, R., Williams, L., and Riseman, E.M. (1989) A data set for quantitative motion analysis. *Proc. IEEE CVPR*, San Diego, pp. 159-164

[18] Fennema C. and Thompson W. (1979) Velocity determination in scenes containing several moving objects. *CGIP* 9, pp. 301-315

[19] 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

[20] Fleet D.J. and Jepson A.D. (1993) Stability of phase information. *IEEE Trans. PAMI* (in press)

[21] Fleet D.J. and Langley K. (1993) Toward real-time optical flow. *Proc. Vision Interface*, Toronto, pp. 116-124 (also see Technical Report: RPL-TR-9308, Robotics and Perception Lab, Queen's University)

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

[23] 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

[24] Glazer F., Reynolds G. and Anandan P. (1983) Scene matching through hierarchical correlation. *Proc. IEEE CVPR*, Washington, pp. 432-441.

[25] Grzywacz N.M. and Yuille A.L. (1990) A model for the estimation of local image velocity by cells in the visual cortex. *Proc. Roy. Soc. London* B239, pp. 129-161

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

[27] Hildreth E.C. (1984) The computation of the velocity field. *Proc. Roy. Soc. London* B221, pp. 189-220

[28] Heeger D.J. (1987) Model for the extraction of image flow. *J. Opt. Soc. Am.* A4, pp. 1455-1471

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

[30] Horn B.K.P. (1986) *Robot Vision.* MIT Press, Cambridge

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

[32] Horn B.K.P. and Weldon Jr. E.J. (1988) Direct methods for recovering motion. *Int. J. Comp. Vision*, 2, pp. 51-76

[33] Jahne B. (1987) Image sequence analysis of complex physical objects: nonlinear small scale water surface waves. *Proc. IEEE ICCV* London, pp. 191-200

[34] Jepson A.D. and Fleet D.J. (1991) Phase singularities in scale-space. *Image and Vision Computing* 9, pp. 338-343

[35] Jepson A.D. and Heeger D.J. (1990) Subspace Methods for Recovering Rigid Motion, Part II: Theory, *Int. J. Comp. Vision*, (to appear)

[36] Kearney J.K., Thompson W.B. and Boley D.L. (1987) Optical flow estimation: An error analysis of gradient-based methods with local optimization. *IEEE Trans. on PAMI* 9, pp. 229-244

[37] Little J.J., Bulthoff H.H. and Poggio T.A. (1988) Parallel optical flow using local voting. *Proc. IEEE ICCV*, Tampa, pp. 454-459

[38] Little J.J. and Verri A. (1989) Analysis of differential and matching methods for optical flow. *IEEE Workshop on Visual Motion*, Irvine CA, pp. 173-180

[39] Lucas B.D. (1984) *Generalized Image Matching by the Method of Differences.* PhD Dissertation, Dept. of Computer Science, Carnegie-Mellon University

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

[41] Marr D. and Hildreth E.C. (1980) Theory of edge detection. *Proc. Roy. Soc. London*, B207, 1980, pp. 187-217

[42] Nagel H.H. (1983) Displacement vectors derived from second-order intensity variations in image sequences. *CGIP* 21, pp. 85-117

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

[44] Nagel H.-H. (1989) On a constraint equation for the estimation of displacement rates in image sequences. *IEEE Trans. PAMI* 11, pp. 13-30

[45] Nagel H.H. and Enkelmann W. (1986) An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE Trans. PAMI* 8, pp. 565-593

[46] Negahdaripour S. and Horn B.K.P. (1987) Direct passive navigation. *IEEE Trans. PAMI* 9, pp. 168-176

[47] Tistarelli M. and Sandini G. (1990) Estimation of depth from motion using anthropomorphic visual sensor. *Image and Vision Computing*, 8, pp. 271-278

[48] Santen J.P.H. van and Sperling G. (1985) Elaborated Reichardt detectors. *J. Opt. Soc. Am.* A2, pp. 300-321

[49] Schunck B.G. (1984) The motion constraint equation for optical flow. *Proc. ICPR* Montreal, pp. 20-22

[50] Schunck B.G. (1986) Image flow continuity equations for motion and density. *Proc. IEEE Workshop on Visual Motion*, Charleston, pp. 89-94

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

[52] Simoncelli E.P. (1993) *Distributed Representation and Analysis of Visual Motion.* PhD Dissertation, Dept. of Electrical Engineering and Computer Science, MIT

[53] Singh A. (1990) An estimation-theoretic framework for image-flow computation. *Proc. IEEE ICCV*, Osaka, pp. 168-177

[54] Singh A. (1992) *Optic Flow Computation: A Unified Perspective.* IEEE Computer Society Press,

[55] Schnoor C. (1991) Determing optical flow for irregular domains by minimizing quadratic functionals of a certain class. *IJCV* 6:1, pp25-38.

[56] Tretiak O. and Pastor L. (1984) Velocity estimation from image sequences with second order differential operators. *Proc. IEEE ICPR*, Montreal, pp. 20-22

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

[58] Verri A. and Poggio T. (1987) Against quantitative optical flow. *Proc. IEEE ICCV*, London, pp. 171-180

[59] Watson A.B. and Ahumada A.J. (1985) Model of human visual-motion sensing. *J. Opt. Soc. Am.* A2, pp. 322-342

[60] Waxman A.M. and Wohn K. (1985) Contour evolution, neighbourhood deformation and global image flow: Planar surfaces in motion. *Int. J. Rob. Res.* 4, pp. 95-108

[61] Waxman A.M., Wu J. and Bergholm F. (1988) Convected activation profiles and receptive fields for real time measurement of short range visual motion. *Proc. IEEE CVPR*, Ann Arbor, pp. 717-723

[62] Willick D. and Yang Y.H. (1991) Experimental evaluation of motion constraints equations. *CVGIP: Image Understanding*, 54, pp. 206-214

A Implementation of Nagel's Algorithm

The functional to be minimized from [43] is

$$\int\int (\nabla I^T \mathbf{v} + I_t)^2 + \frac{\alpha^2}{||\nabla I||_2^2 + 2\delta}\Big[(u_x I_y - u_y I_x)^2 + (v_x I_y - v_y I_x)^2 + \\ \delta(u_x^2 + u_y^2 + v_x^2 + v_y^2)\Big]\, dxdy. \tag{A.1}$$

Since the iterative equations were not in that paper, we derived them as outlined below.

The original PDEs in [43] are:

$$(\nabla I \cdot \vec{v} + I_t)I_x \ + \alpha^2 \frac{\nabla I^T \left[\begin{pmatrix} I_{yy} & -I_{xy} \\ -I_{xy} & I_{xx} \end{pmatrix} + 2\begin{pmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{pmatrix} W\right]\begin{pmatrix} u_x \\ u_y \end{pmatrix}}{{I_x}^2 + {I_y}^2 + 2\delta} \\ - \alpha^2 \text{ trace } (W\nabla\nabla u) = 0 \tag{A.2}$$

$$(\nabla I^T \cdot \vec{v} + I_t)I_y \ + \alpha^2 \frac{\nabla I^T \left[\begin{pmatrix} I_{yy} & -I_{xy} \\ -I_{xy} & I_{xx} \end{pmatrix} + 2\begin{pmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{pmatrix} W\right]\begin{pmatrix} v_x \\ v_y \end{pmatrix}}{{I_x}^2 + {I_y}^2 + 2\delta} \\ - \alpha^2 \text{ trace } (W\nabla\nabla v) = 0 \tag{A.3}$$

where

$$W = \frac{1}{I_x^2 + I_y^2 + 2\delta}\begin{bmatrix} I_y^2 + \delta & -I_x I_y \\ -I_x I_y & I_x^2 + \delta \end{bmatrix}. \tag{A.4}$$

These equations can be rewritten as:

$$\begin{aligned} (I_x u + I_y v + I_t)I_x + \alpha^2[\vec{e} \cdot \nabla u - \phi(u)] &= 0 \\ (I_x u + I_y v + I_t)I_y + \alpha^2[\vec{e} \cdot \nabla v - \phi(v)] &= 0 \end{aligned} \tag{A.5}$$

where

$$\vec{e}^T \ = \ \frac{\nabla I^T \left[\begin{pmatrix} I_{yy} & -I_{xy} \\ -I_{xy} & I_{xx} \end{pmatrix} + 2\begin{pmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{pmatrix} W\right]}{{I_x}^2 + {I_y}^2 + 2\delta} \tag{A.6}$$

and

$$\phi(u) = \text{ trace } (W\nabla\nabla u) \tag{A.7}$$

$$\phi(v) = \text{ trace } (W\nabla\nabla v) \tag{A.8}$$

By expanding (A.7) and (A.8) we find that

$$\phi(u) = \frac{({I_y}^2 + \delta)u_{xx} + ({I_x}^2 + \delta)u_{yy} - 2I_x I_y u_{xy}}{||\nabla I||_2^2 + 2\delta} \tag{A.9}$$

$$\phi(v) = \frac{({I_y}^2 + \delta)v_{xx} + ({I_x}^2 + \delta)v_{yy} - 2I_x I_y v_{xy}}{||\nabla I||_2^2 + 2\delta} \tag{A.10}$$

We numerically estimate the second derivatives of u and v using the mask $[1, -2, 1]$. In this case, the first two terms of (A.9) as follows

$$\frac{({I_y}^2 + \delta)u_{xx} + ({I_x}^2 + \delta)u_{yy}}{||\nabla I||_2^2 + 2\delta} \approx$$

$$\frac{(I_y{}^2+\delta)(u_{i,j-1}+u_{i,j+1}-2u_{i,j})+(I_x{}^2+\delta)(u_{i-1,j}+u_{i+1,j}-2u_{i,j})}{\|\nabla I\|_2^2+2\delta} =$$
$$\frac{(I_y{}^2+\delta)(u_{i,j-1}+u_{i,j+1})+(I_x{}^2+\delta)(u_{i-1,j}+u_{i+1,j})}{\|\nabla I\|_2^2+2\delta} - 2u_{i,j} = 2\bar{u}_{i,j} - 2u_{i,j} \qquad \text{(A.11)}$$

where $\bar{u}$ denotes a weighted average of u near location i,j. Note that derivatives in the x direction are computed using velocities in the j dimension and derivatives in the y direction using velocities in the i dimension.

Similarly we can approximate the first two terms of (A.10) as follows:

$$\frac{(I_y{}^2+\delta)v_{xx}+(I_x{}^2+\delta)v_{yy}}{\|\nabla I\|_2^2+2\delta} \approx 2\bar{v}_{i,j} - 2v_{i,j}\ , \qquad \text{(A.12)}$$

where

$$\bar{u}_{i,j} = \frac{(I_y{}^2+\delta)(u_{i,j-1}+u_{i,j+1})+(I_x{}^2+\delta)(u_{i-1,j}+u_{i+1,j})}{2(\|\nabla I\|_2^2+2\delta)} \qquad \text{(A.13)}$$

and

$$\bar{v}_{i,j} = \frac{(I_y{}^2+\delta)(v_{i,j-1}+v_{i,j+1})+(I_x{}^2+\delta)(v_{i-1,j}+v_{i+1,j})}{2(\|\nabla I\|_2^2+2\delta)} \qquad \text{(A.14)}$$

Since numerical estimates of other partial derivatives of $\vec{v}=(u,v)$ i.e. u_x, u_y, v_x, v_y, u_{xy} and v_{xy} do not involve $\vec{v}_{i,j}$ itself, they can be computed directly using central difference formulas, and we can rewrite $\xi(u)$ and $\xi(v)$ as:

$$\phi(u) = 2\bar{u}_{i,j} - 2u_{i,j} - \frac{2I_xI_yu_{xy}}{\|\nabla I\|_2^2+2\delta} \qquad \text{(A.15)}$$
$$\phi(v) = 2\bar{v}_{i,j} - 2v_{i,j} - \frac{2I_xI_yv_{xy}}{\|\nabla I\|_2^2+2\delta} \qquad \text{(A.16)}$$

The set of PDEs can then be written as

$$I_x(I_xu+I_yv+I_t)+\alpha^2\left[\vec{e}\cdot\nabla u - 2\bar{u} + 2u + \frac{2I_xI_yu_{xy}}{\|\nabla I\|_2^2+2\delta}\right] = 0 \qquad \text{(A.17)}$$
$$I_y(I_xu+I_yv+I_t)+\alpha^2\left[\vec{e}\cdot\nabla v - 2\bar{v} + 2v + \frac{2I_xI_yv_{xy}}{\|\nabla I\|_2^2+2\delta}\right] = 0. \qquad \text{(A.18)}$$

Rearranging yields:

$$(I_x{}^2+2\alpha^2)u+I_xI_yv = 2\alpha^2\xi(u)-I_xI_t \qquad \text{(A.19)}$$
$$I_xI_yu+(I_y{}^2+2\alpha^2)v = 2\alpha^2\xi(v)-I_yI_t \qquad \text{(A.20)}$$

where $\xi(u) = \frac{1}{2}(-\vec{e}\cdot\nabla u + 2\bar{u} - 2\frac{I_xI_yu_{xy}}{\|\nabla I\|_2^2+2\delta})$ and $\xi(v) = \frac{1}{2}(-\vec{e}\cdot\nabla v + 2\bar{v} - 2\frac{I_xI_yv_{xy}}{\|\nabla I\|_2^2+2\delta})$. Using Cramer's rule, we can write

$$(I_x{}^2+I_y{}^2+2\alpha^2)u = \xi(u)(I_y{}^2+2\alpha^2)-I_xI_y\xi(v)-I_xI_t \qquad \text{(A.21)}$$
$$(I_x{}^2+I_y{}^2+2\alpha^2)v = \xi(v)(I_x{}^2+2\alpha^2)-I_xI_y\xi(u)-I_yI_t \qquad \text{(A.22)}$$

By adding and subtracting $I_x{}^2\xi(u)$ to the right side of (A.21) and $I_y{}^2\xi(v)$ to the right side of (A.22), we obtain the following constraint:

$$u = \xi(u) - \frac{I_x(I_x\xi(u)+I_y\xi(v)+I_t)}{(I_x{}^2+I_y{}^2+2\alpha^2)} \qquad \text{(A.23)}$$
$$v = \xi(v) - \frac{I_y(I_x\xi(u)+I_y\xi(v)+2I_t)}{(I_x{}^2+I_y{}^2+2\alpha^2)}. \qquad \text{(A.24)}$$

Using the Gauss-Seidel method we obtained iterative equations in (u, v):

$$u^{k+1} = \xi(u^k) - \frac{I_x(I_x\xi(u^k) + I_y\xi(v^k) + I_t)}{{I_x}^2 + {I_y}^2 + 2\alpha^2} \tag{A.25}$$

$$v^{k+1} = \xi(v^k) - \frac{I_y(I_x\xi(u^k) + I_y\xi(v^k) + I_t)}{{I_x}^2 + {I_y}^2 + 2\alpha^2} \tag{A.26}$$

where, as above:

$$\xi(u^k) = \frac{1}{2}\left(-\vec{e}^T\begin{pmatrix} {u_x}^k \\ {u_y}^k \end{pmatrix} + 2\bar{u}^k - \frac{2I_xI_yu^k_{xy}}{\|\nabla I\|_2^2 + 2\delta}\right)$$

$$\xi(v^k) = \frac{1}{2}\left(-\vec{e}^T\begin{pmatrix} {v_x}^k \\ {v_y}^k \end{pmatrix} + 2\bar{v}^k - \frac{2I_xI_yv^k_{xy}}{\|\nabla I\|_2^2 + 2\delta}\right) .$$

The superscript k represents the iteration number, and ${u_x}^k$, ${u_y}^k$, ${v_x}^k$,${v_y}^k$,${u_{xy}}^k$ and ${v_{xy}}^k$ denote the partial derivatives of $\vec{v}$ at iteration k. The terms $\bar{u}^k$ and $\bar{v}^k$ are local weighted averages, computed according to (A.11) and (A.12). The partial derivatives of $\vec{v} = (u, v)$ are computed using the 1^{st} order central difference mask $\frac{1}{2}[1, 0, -1]$. We also compute 2^{nd} order partial derivatives by cascading this kernel with the 1^{st} order results.

Lastly, we note that if the second order spatio-temporal derivatives of intensity and velocity, i.e. I_{xx}, I_{yy}, I_{xy} and u_{xx}, u_{yy}, u_{xy}, v_{xx}, v_{yy} and v_{xy}, are all zero then $\xi(u) = \bar{u}$ and $\xi(v) = \bar{v}$ and equations (A.25) and (A.26) reduce to Horn and Schunck's [31] iterative equations (although the neighbourhood averages used by Horn and Schunck differ from those here due to the different form of differentiation):

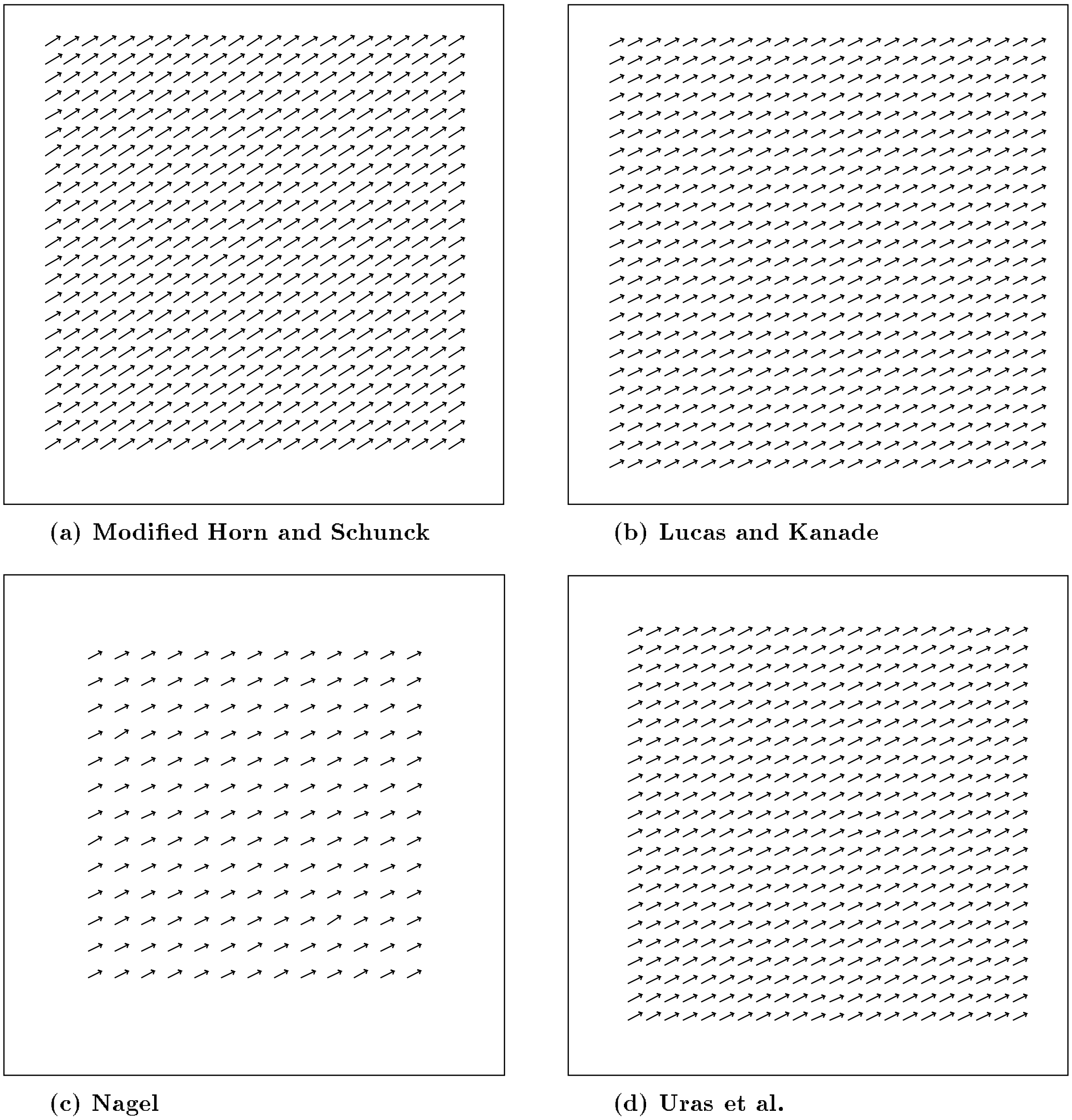
$$u^{k+1} \;=\; \bar{u}^k - \frac{I_x(I_x\bar{u}^k + I_y\bar{v}^k + I_t)}{I_x^2 + I_y^2 + 2\alpha^2} \tag{A.27}$$

and

$$v^{k+1} \;=\; \bar{v}^k - \frac{I_y(I_x\bar{u}^k + I_y\bar{v}^k + I_t)}{I_x^2 + I_y^2 + 2\alpha^2}. \tag{A.28}$$

Note that the original Horn and Schunck algorithm does not have a 2 in front of α, but α just controls smoothing and using α for Horn and Schunck's smoothing parameter is equivalent to using $\alpha/\sqrt{2}$ here.

B Sinusoidal Results



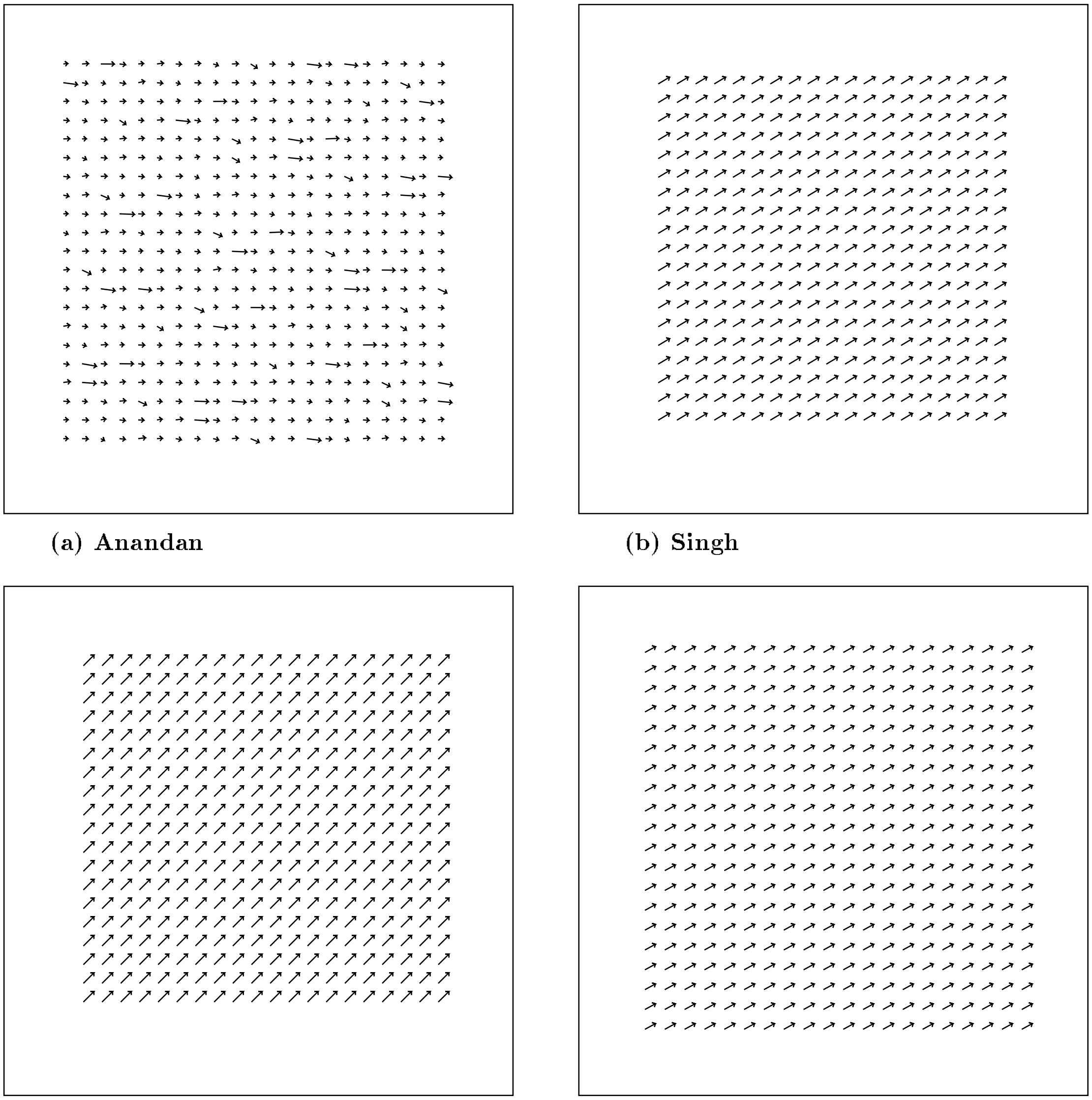
(a) Modified Horn and Schunck

(b) Lucas and Kanade

(c) Nagel

(d) Uras et al.

Figure B.1: **Sinusoidal Input:** Flow fields are shown for the **Sinusoid 1** sequence.



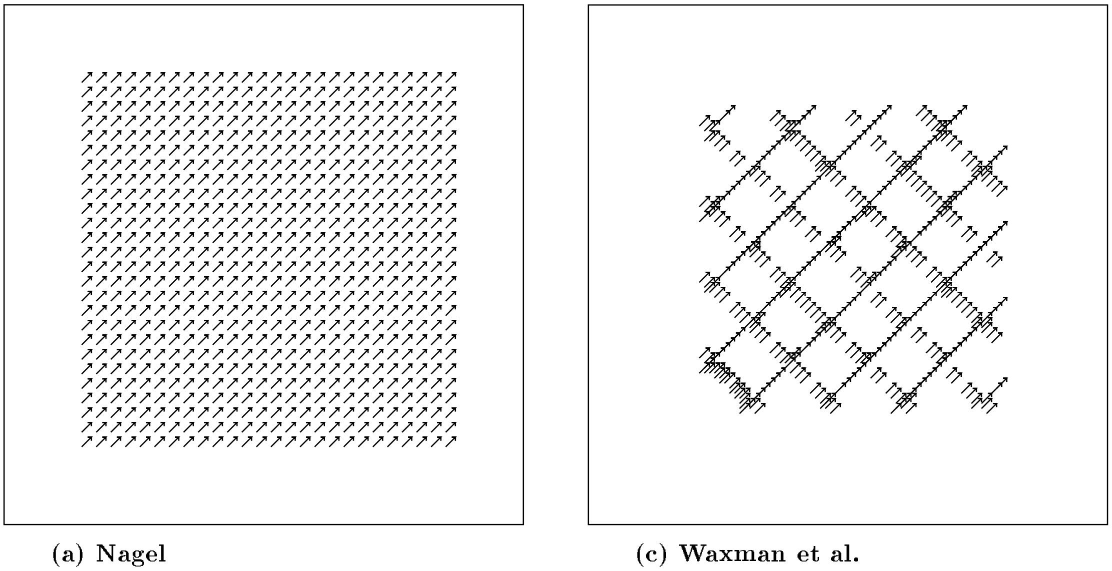
(a) Anandan

(b) Singh

(c) Heeger

(d) Fleet and Jepson

Figure B.2: **Sinusoidal Input:** Flow fields are shown for the **Sinusoid 1** sequence.

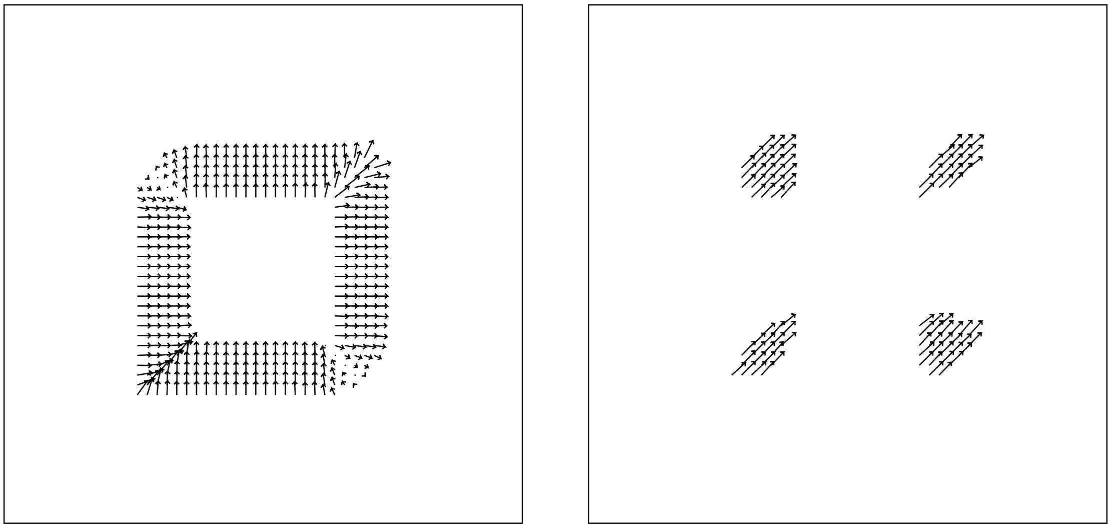


(a) Nagel

(c) Waxman et al.

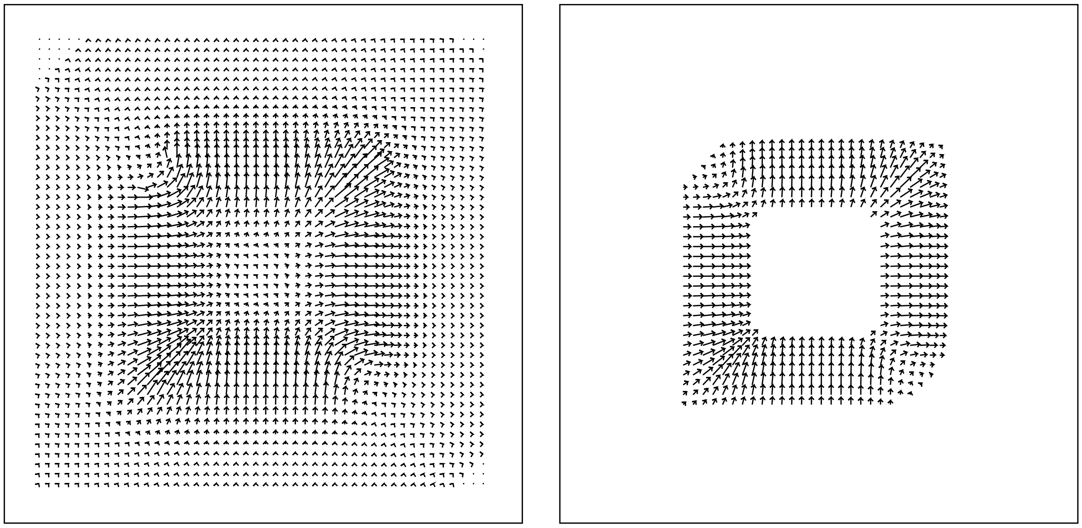
Figure B.3: **Sinusoidal Input:** The flow fields produced by the methods of **Nagel** (for $\alpha = 0.5$) and of **Waxman et al.** are shown for the **Sinusoid 2** sequence.

C Square Results



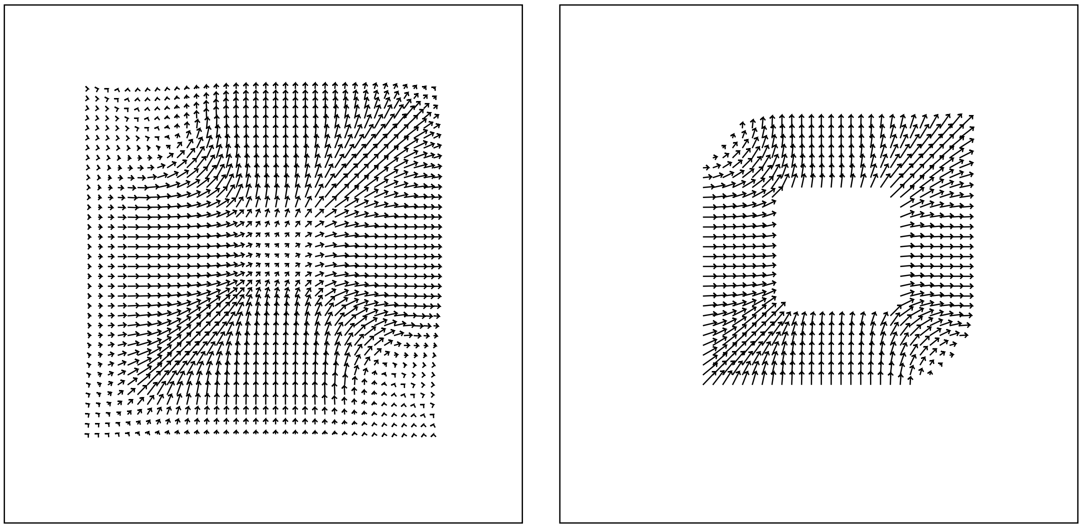
(a) **Lucas and Kanade** (raw normal velocity $||\nabla I||_2 \geq 5.0$) (b) **Lucas and Kanade** (2d velocity)

Figure C.1: **Square Input:** Velocity estimates for **Square 2**: for **Lucas and Kanade**: with spatiotemporal smoothing ($\sigma = 1.5$) and thresholding on $\lambda_2 \geq 1.0$.



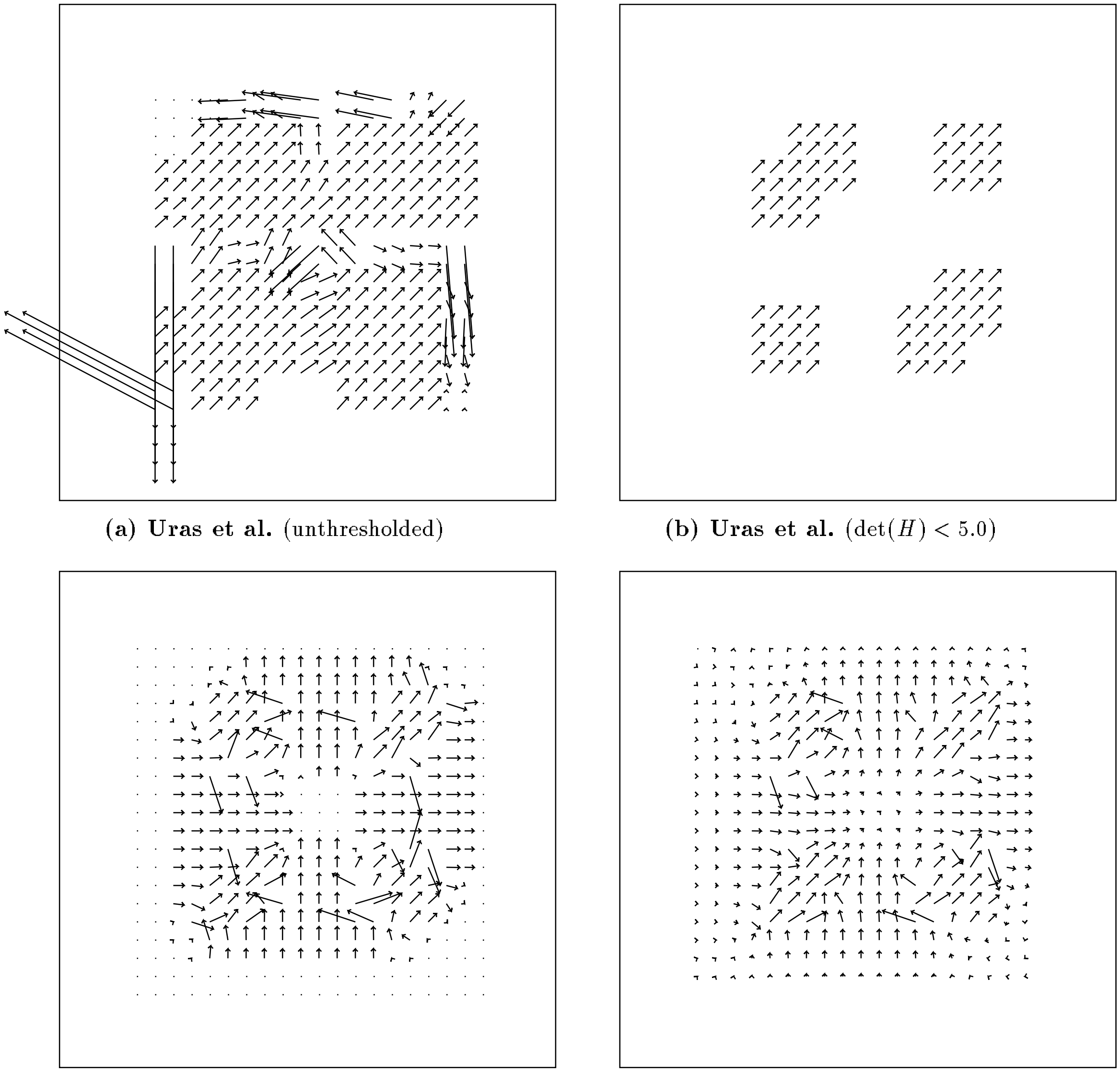
(a) unthresholded (b) $||\nabla I||_2 \geq 1.0$

Figure C.2: *Flow fields for Horn and Schunck for* **square2** *unthresholded and thresholded by* $||\nabla I||_2 \geq 1.0$.



(a) unthresholded (b) $||\nabla I||_2 \geq 1.0$

Figure C.3: *Flow fields for Nagel for* **square2** *unthresholded and thresholded by* $||\nabla I||_2 \geq 1.0$.



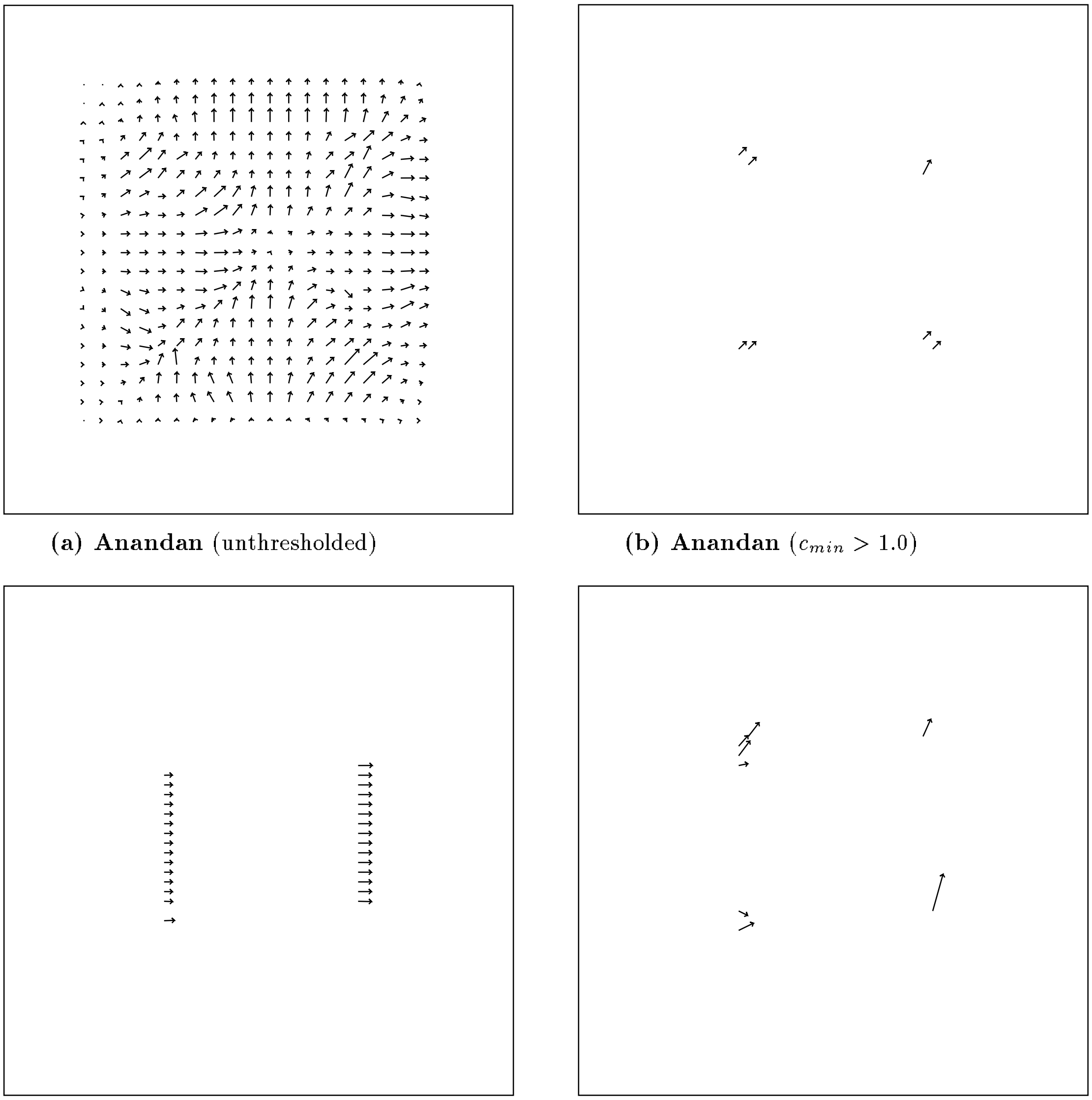
(a) Uras et al. (unthresholded)

(b) Uras et al. ($\det(H) < 5.0$)

(c) Singh (step 1, unthresholded)

(d) Singh (step 2, unthresholded)

Figure C.4: **Square Input:** Flow fields for **Square 2** input: (top) For **Uras et al.**, with and without a confidence threshold on the velocity estimates; (bottom) For steps 1 and 2 of **Singh** without thresholding.



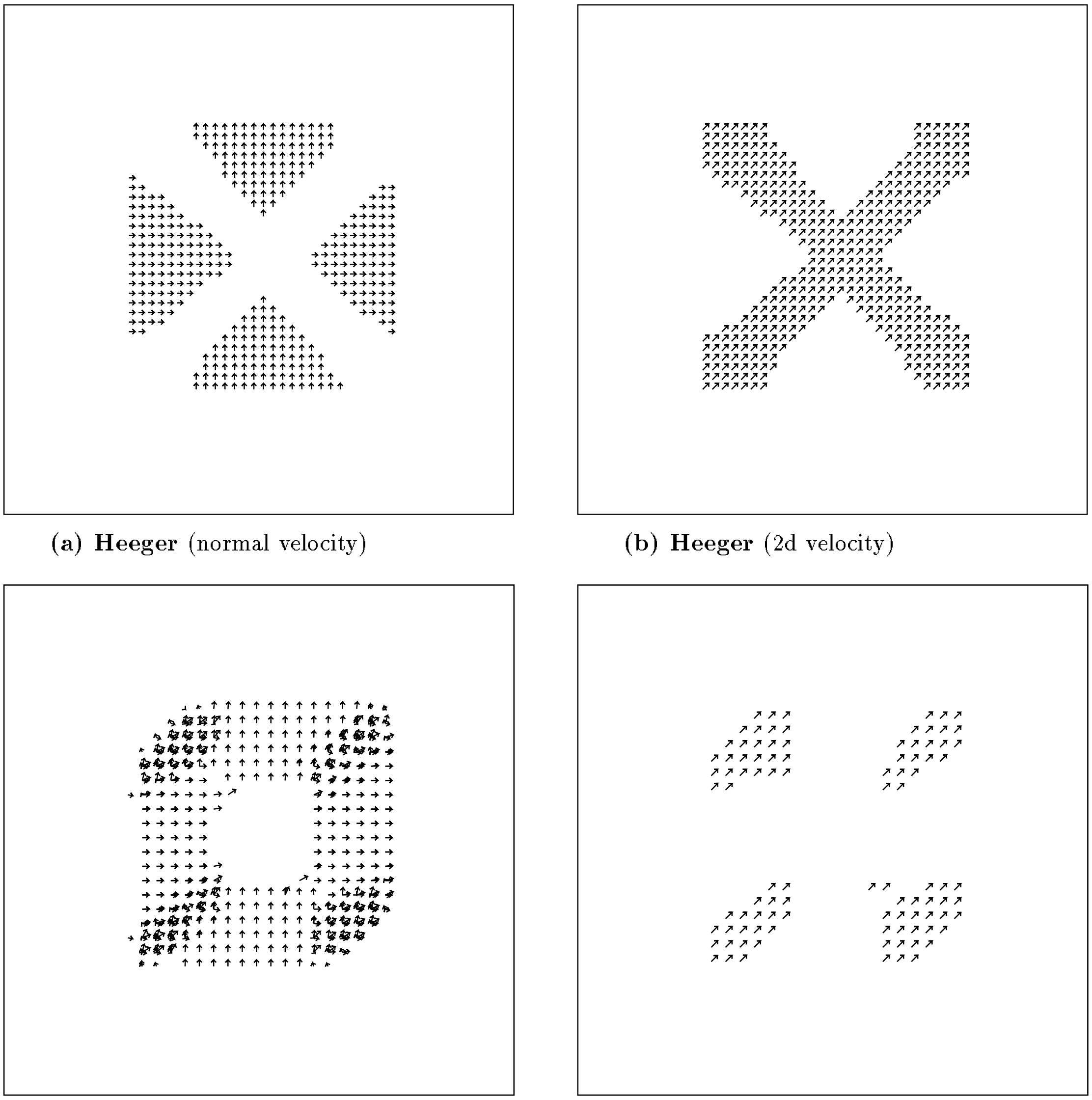
(a) **Anandan** (unthresholded)

(b) **Anandan** ($c_{min} > 1.0$)

(c) **Waxman et al.** (normal velocity)

(d) **Waxman et al.** (2d velocity)

Figure C.5: **Square Input:** Flow fields for **Square 2** input: (top) For **Anandan's** technique with and without thresholding on c_{min}; (bottom) For the method of **Waxman et al.** we show both 2d and normal velocity estimates.



(a) **Heeger** (normal velocity)

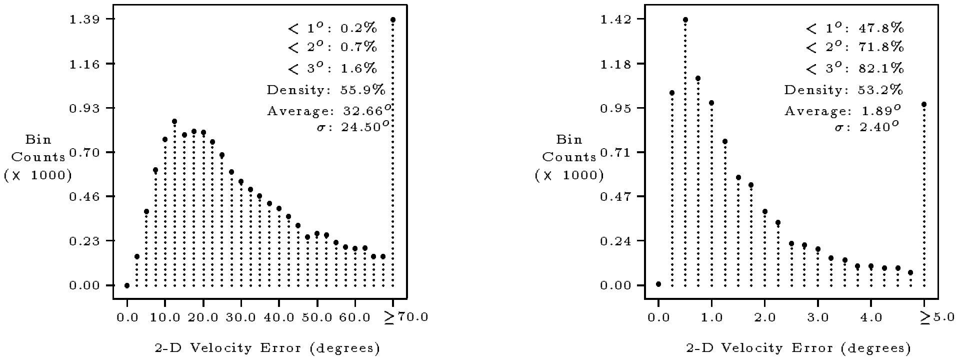
(b) **Heeger** (2d velocity)

(c) **Fleet and Jepson** (normal velocity)

(d) **Fleet and Jepson** (2d velocity)

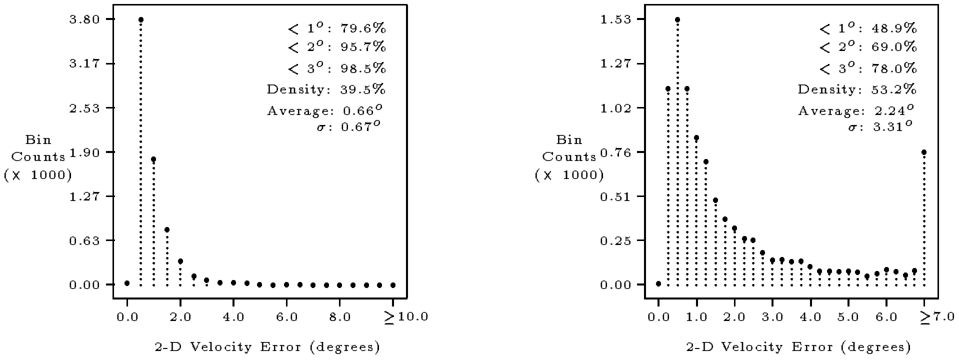
Figure C.6: **Square Input:** 2d and normal velocity estimates produced by the methods of **Heeger** and **Fleet and Jepson** on the **Square 2** input sequence.

D Translating Tree Results



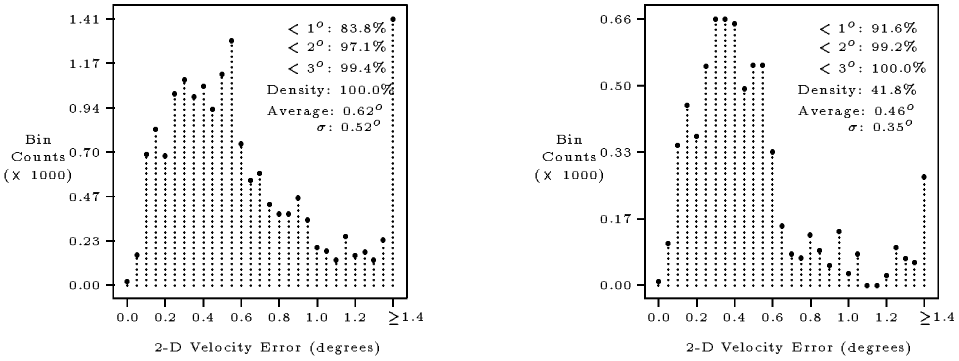
(D.1a) (D.1b)

Histogram D.1: **Translating Tree:** Error Histograms for (a) standard **Horn and Schunck** and (b) modified **Horn and Schunck**, both with thresholding $||\nabla I||_2 \geq 5.0$.



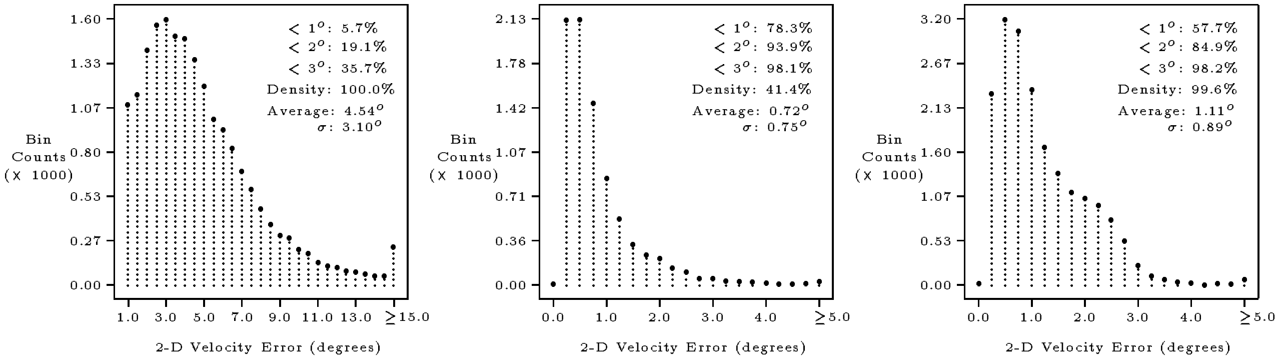
(D.2a) (D.2b)

Histogram D.2: **Translating Tree:** Error Histograms for (a) **Lucas and Kanade** ($\lambda_2 > 1.0$) and (b) **Nagel** ($||\nabla I||_2 > 5.0$).



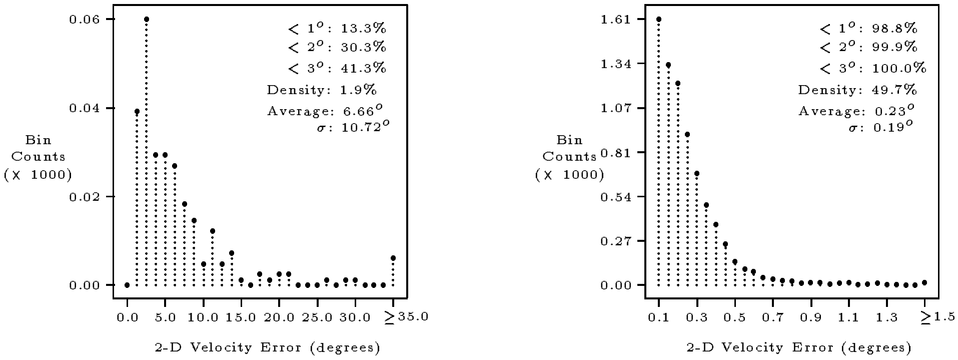
(D.3a) (D.3b)

Histogram D.3: **Translating Tree:** Error Histograms for **Uras et al.**: (a) unthresholded and (b) thresholded ($det(H) \geq 1.0$).



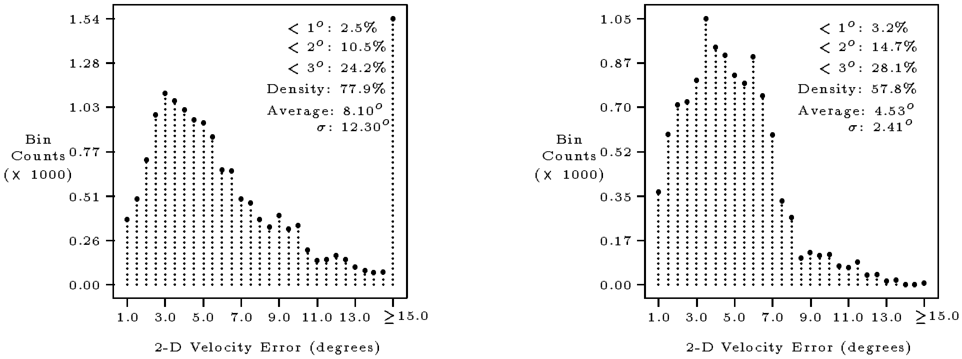
(D.4a) (D.4b) (D.4c)

Histogram D.4: **Translating Tree:** Error Histograms for **Anandan**: (a) unthresholded and for **Singh**: (b) for step 1, with thresholding on $\lambda_1 \geq 5.0$ and (c) for step 2, with thresholding on $\lambda_1 \geq 0.1$.



(D.5a) (D.5b)

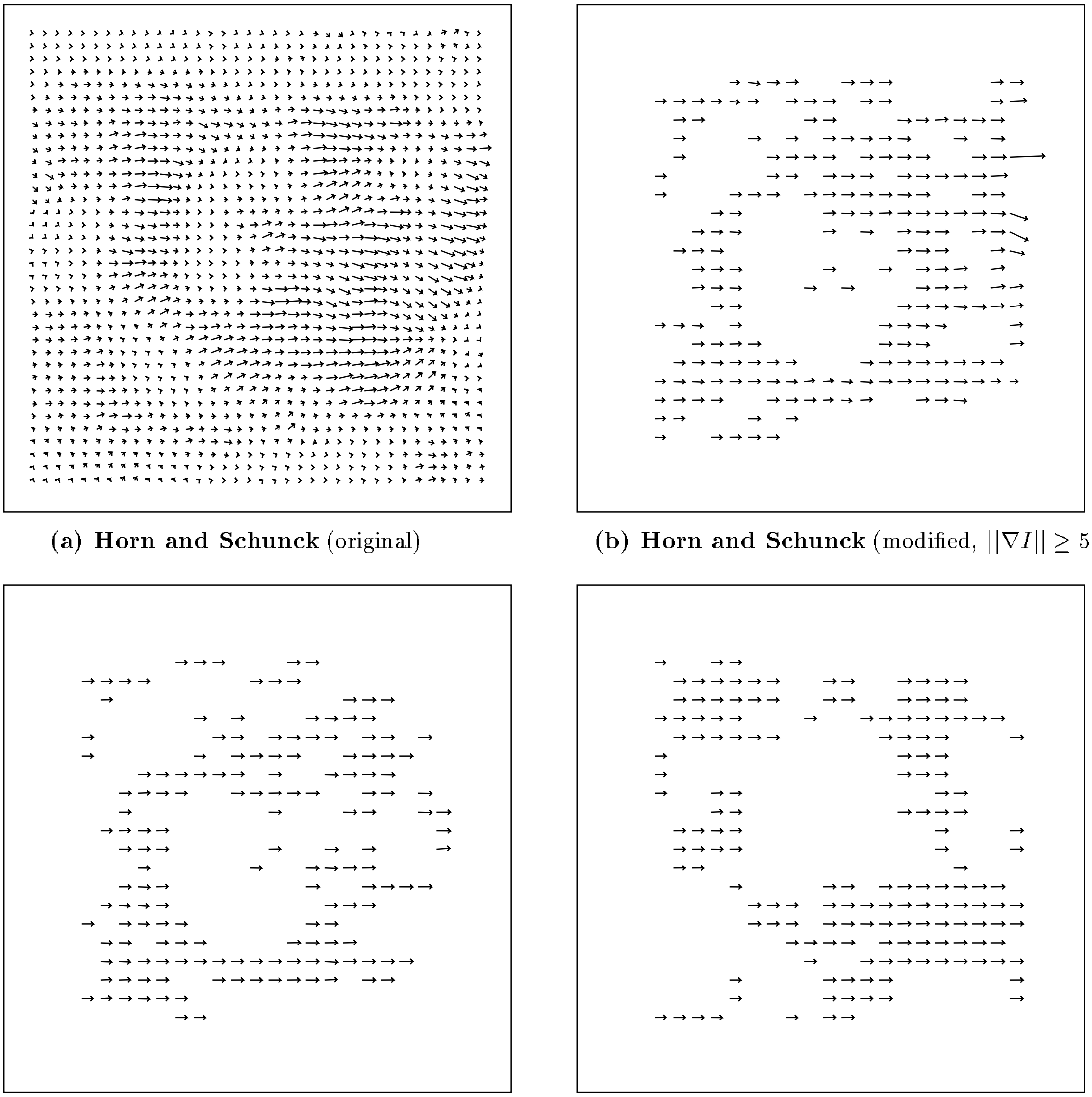
Histogram D.5: **Translating Tree:** Error Histograms for (a) **Waxman et al** and (b) **Fleet and Jepson** ($\tau = 1.25$).



(D.6a)

(D.6a)

Histogram D.6: **Translating Tree:** Error Histograms for **Heeger** for (a) level 0 and (b) level 1.



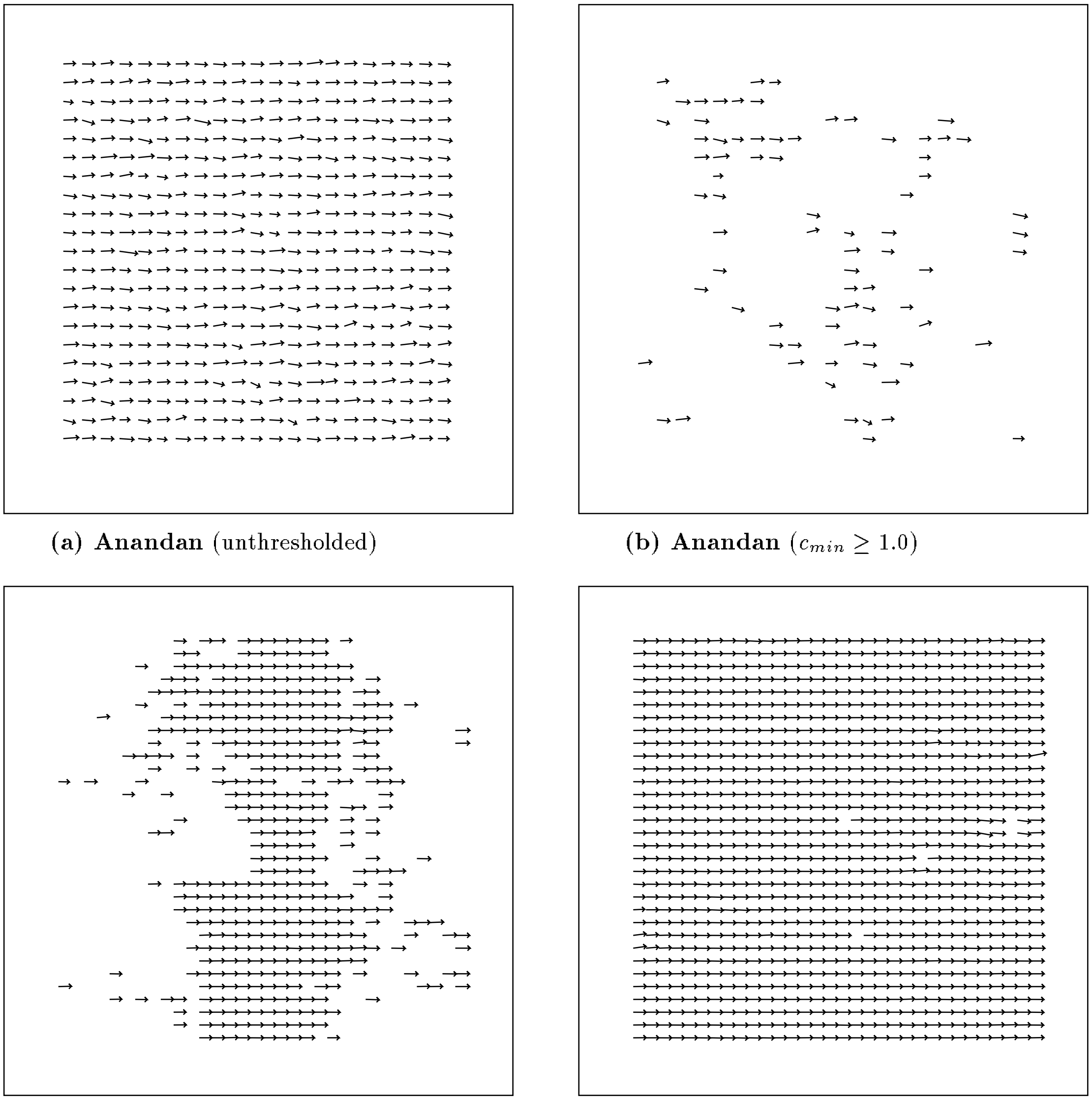
(a) **Horn and Schunck** (original)

(b) **Horn and Schunck** (modified, $||\nabla I|| \geq 5.0$)

(c) **Lucas and Kanade** ($\lambda_2 \geq 1.0$)

(d) **Uras et al.** ($det(H) \geq 1.0$)

Figure D.1: **Translating Tree:** Flow fields produced by differential techniques.



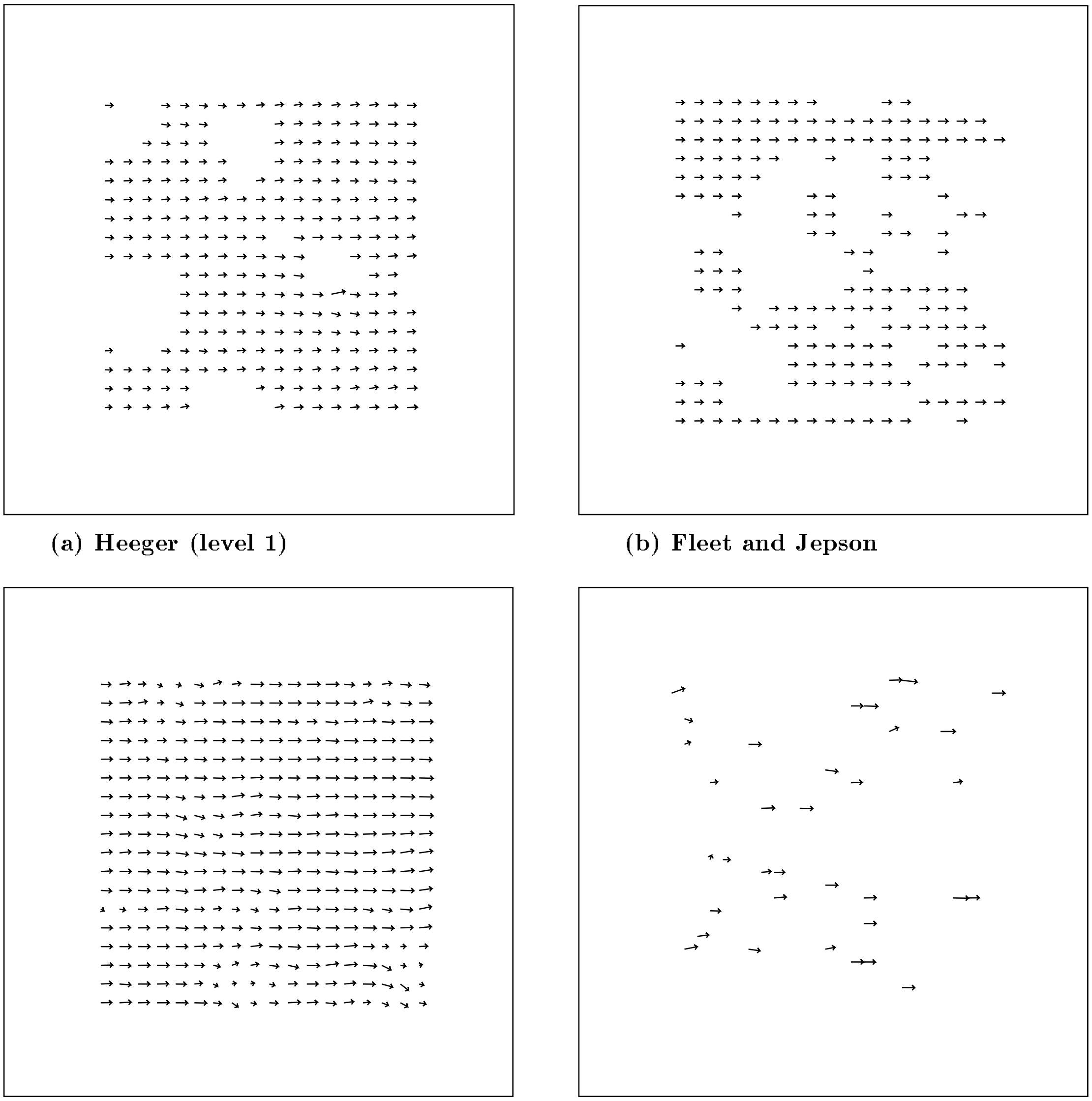
(a) **Anandan** (unthresholded)

(b) **Anandan** ($c_{min} \geq 1.0$)

(c) **Singh** (step 1, $\lambda_1 \geq 5.0$)

(d) **Singh** (step 2, $\lambda_1 \geq 0.1$)

Figure D.2: **Translating Tree:** Flow fields produced by matching techniques. For **Singh's** technique, the parameters used were $n = 2$, $w = 2$, and $N = 4$.



(a) **Heeger (level 1)**

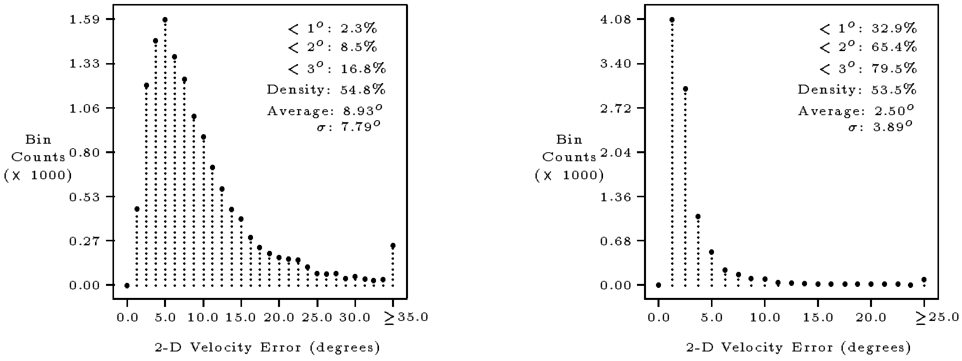
(b) **Fleet and Jepson**

(c) **Nagel** $||\nabla I||_2 \geq 5.0$

(d) **Waxman** $(\sigma_f = 2.0)$

Figure D.3: **Translating Tree:** Flow fields produced by assorted methods.

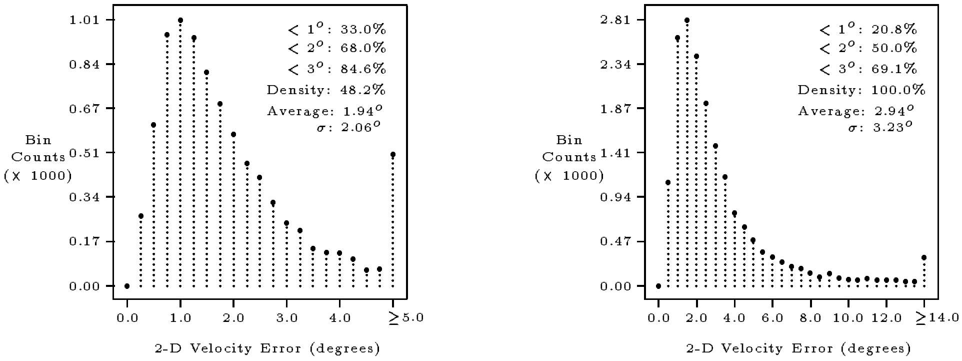
E Diverging Tree Results



(E.7a)

(E.7b)

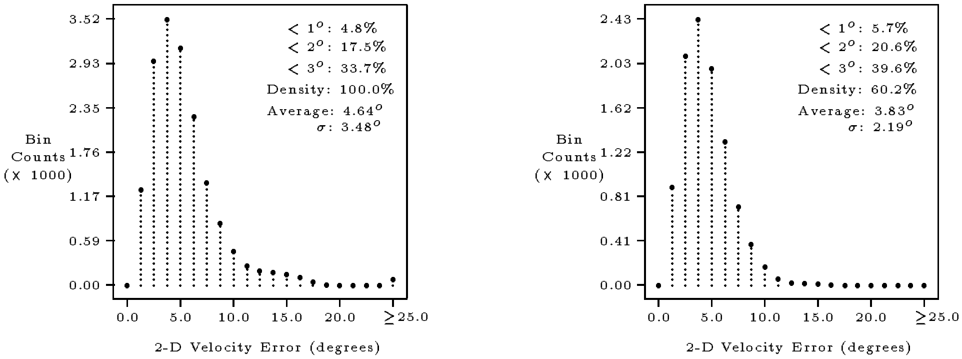
Histogram E.7: **Diverging Tree:** Error Histograms for (a) standard **Horn and Schunck**, (b) modified **Horn and Schunck**, both with thresholding $||\nabla I||_2 \geq 5.0$.



(E.8a)

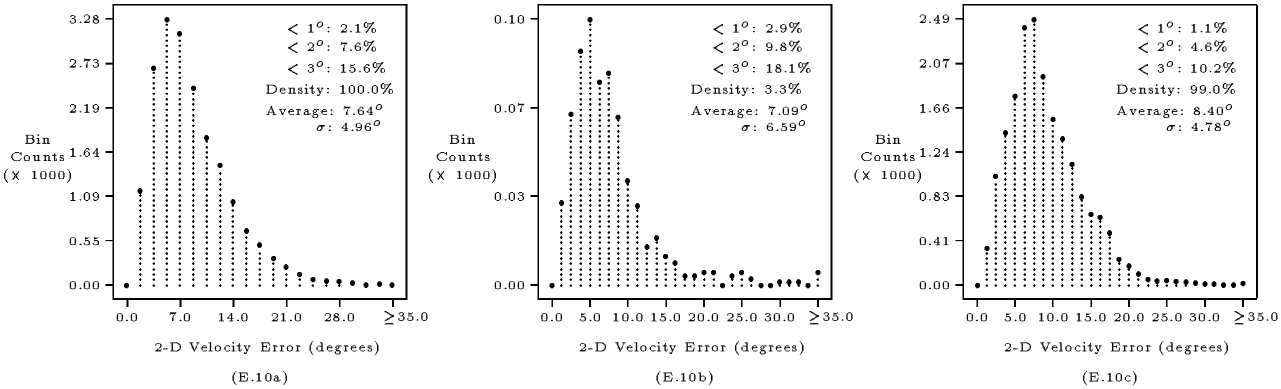
(E.8a)

Histogram E.8: **Diverging Tree:** Error Histograms for: (a) **Lucas and Kanade** with $\lambda_2 \geq 1.0$ thresholding and (b) **Nagel** (no thresholding).

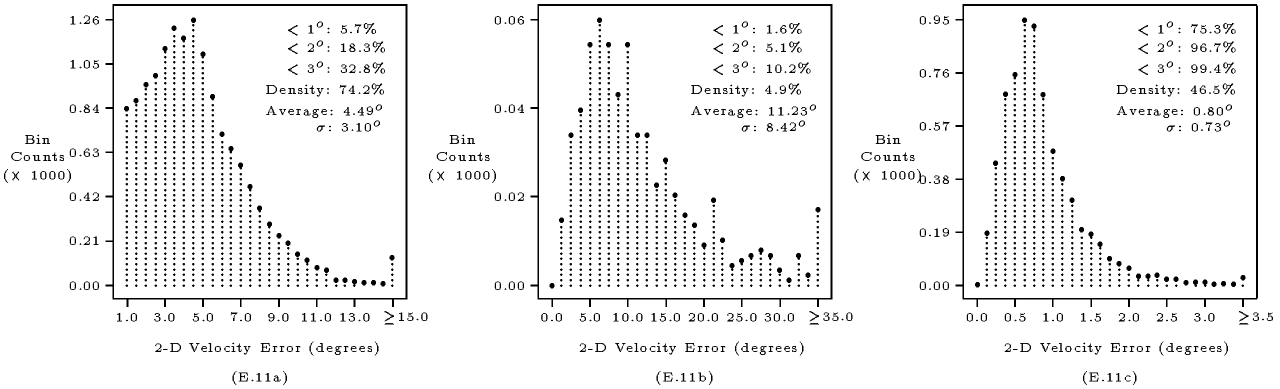


(E.9a) (E.9b)

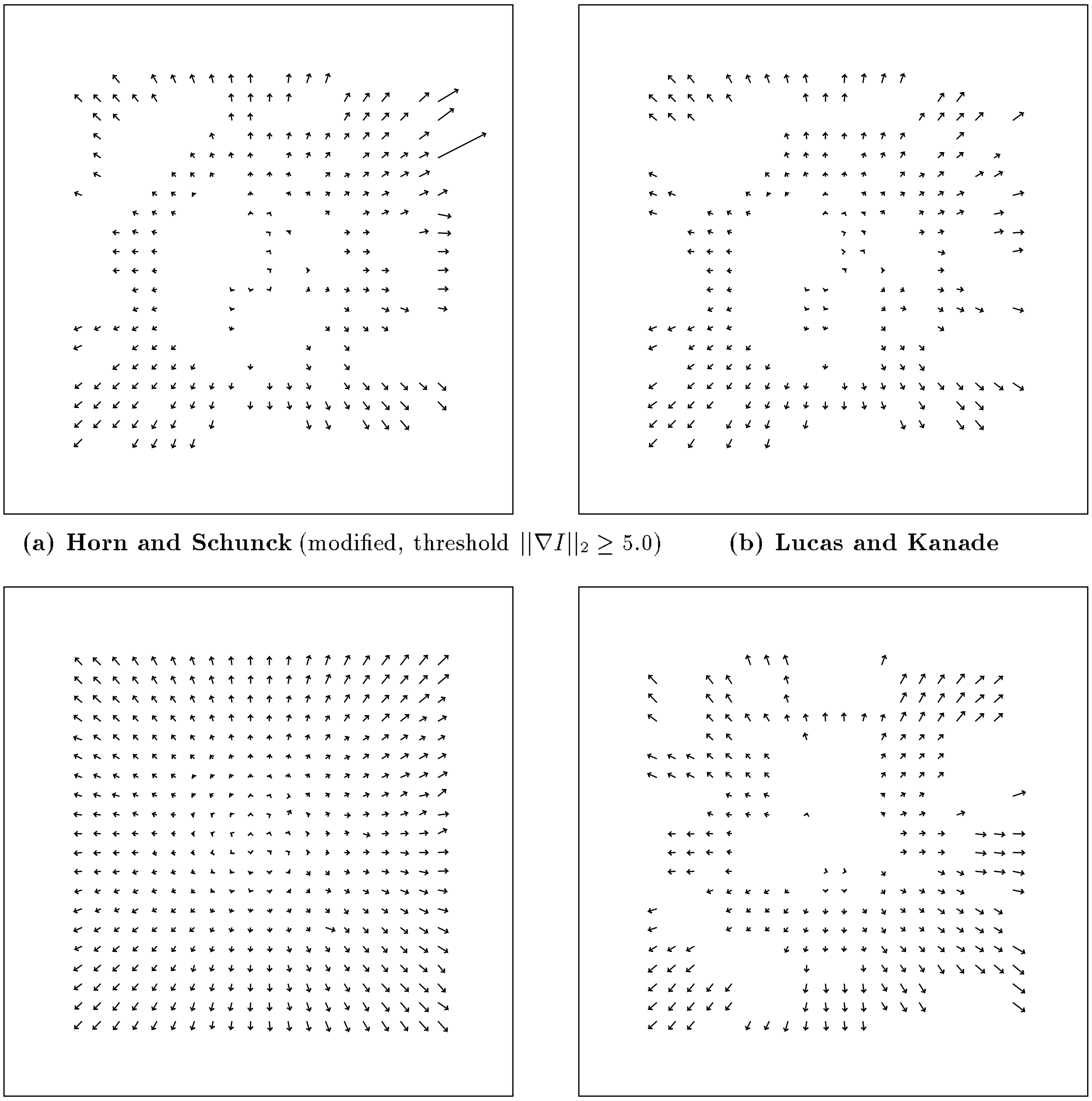
Histogram E.9: **Diverging Tree:** Error Histograms for **Uras et al.**: (a) unthresholded and (b) thresholded $(det(H) \geq 1.0)$.



Histogram E.10: **Diverging Tree:** Error Histograms for (a) **Anandan** with no thresholding and **Singh**: (b) for step 1 with $\lambda_1 \geq 5.0$ thresholding and (c) for step 2 with $\lambda_1 \geq 0.1$ thresholding.



Histogram E.11: **Diverging Tree:** Error Histograms for (a) **Heeger** (level 0), (b) **Waxman et al.** and (c) **Fleet and Jepson** ($\tau = 1.25$).



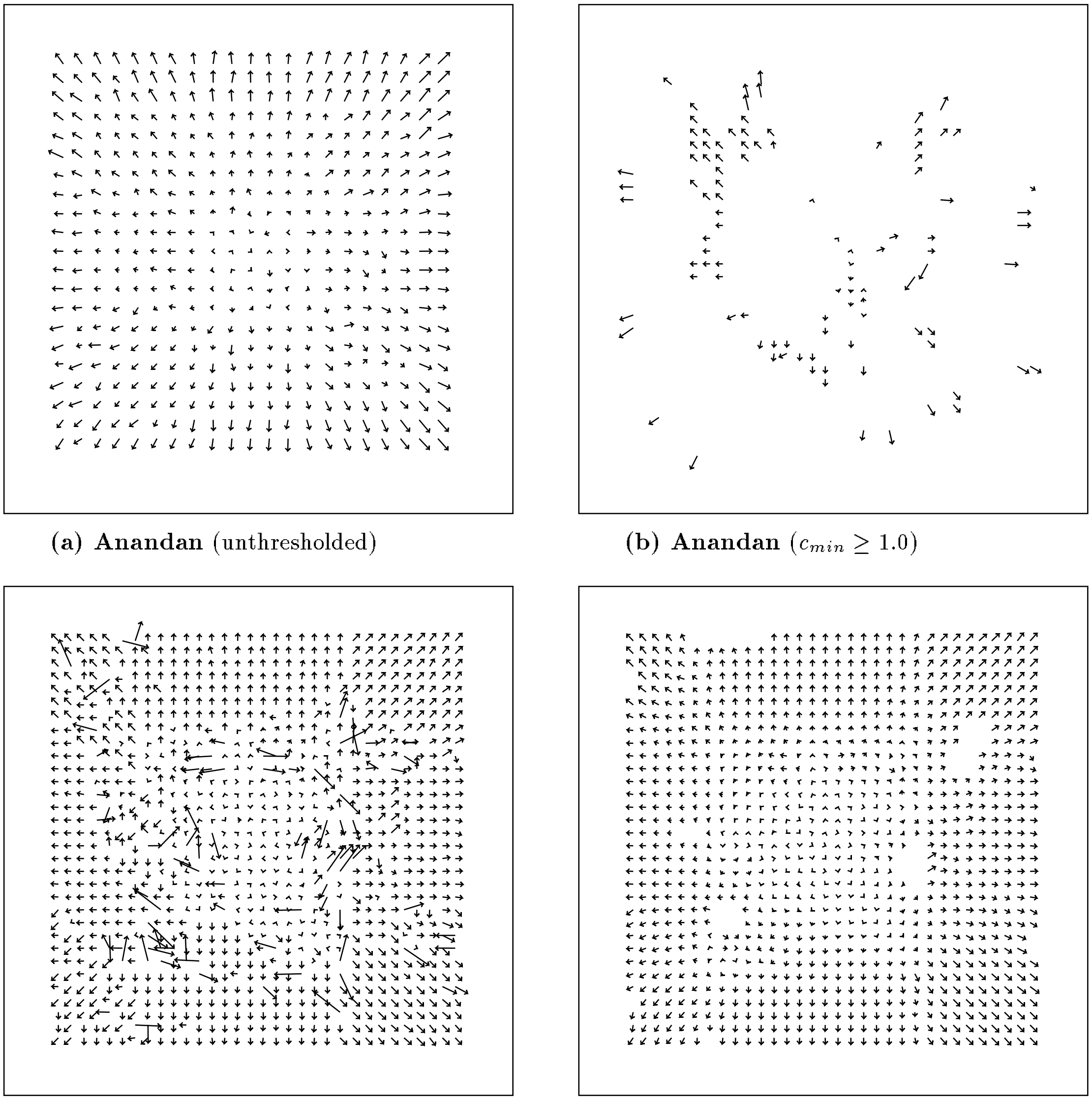
(a) **Horn and Schunck** (modified, threshold $||\nabla I||_2 \geq 5.0$)

(b) **Lucas and Kanade**

(c) **Nagel (unthresholded)**

(d) **Uras et al.** ($det(H) \geq 1.0$)

Figure E.1: **Diverging Tree:** Flow fields for differential techniques.



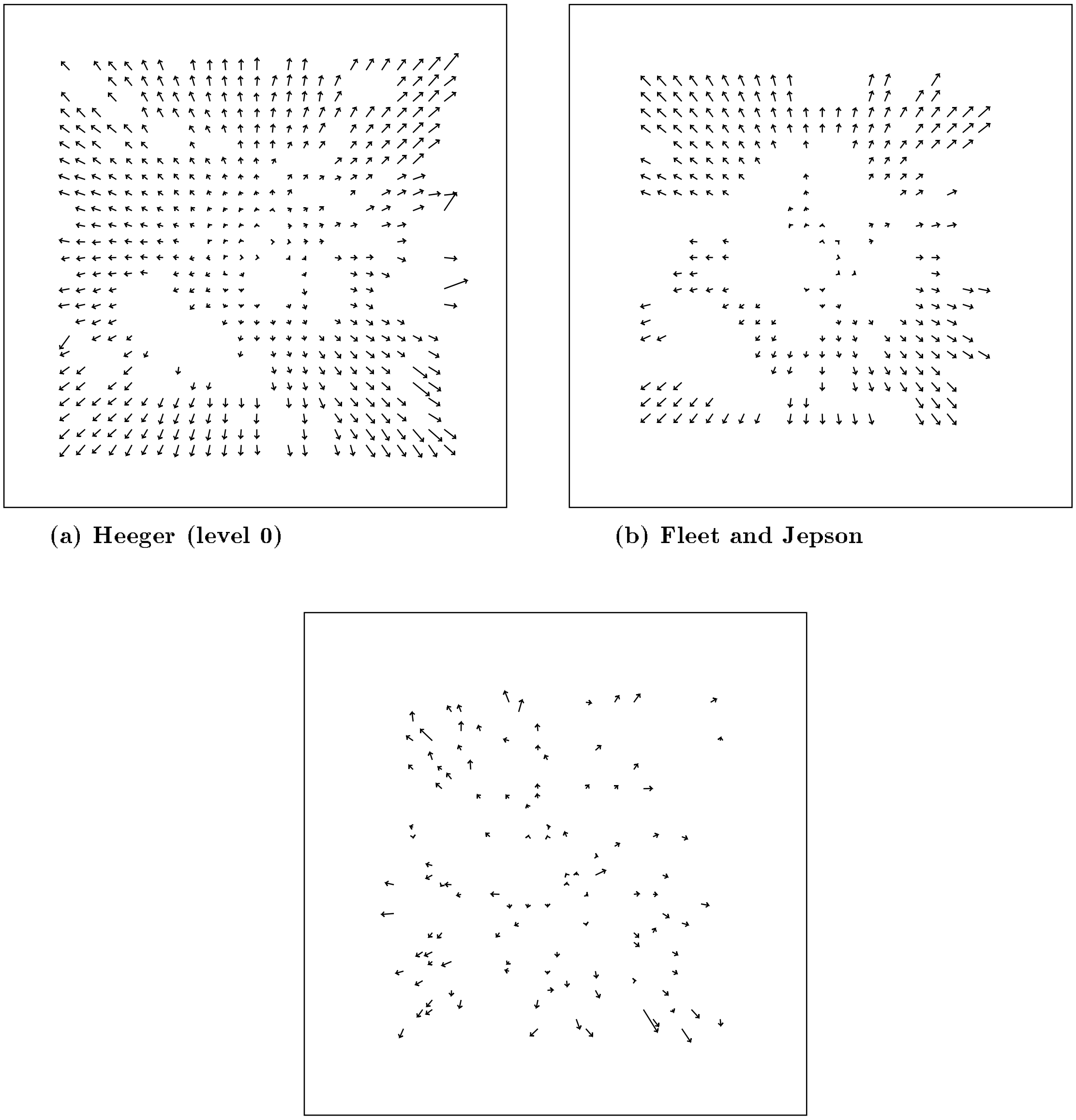
(a) **Anandan** (unthresholded)

(b) **Anandan** ($c_{min} \geq 1.0$)

(c) **Singh** (step 1, unthresholded)

(d) **Singh** (step 2, $\lambda_1 \geq 0.1$)

Figure E.2: **Diverging Tree:** Flow fields produced by matching techniques. For **Singh's** technique, the parameters used were $n = 2$, $w = 2$, and $N = 4$.



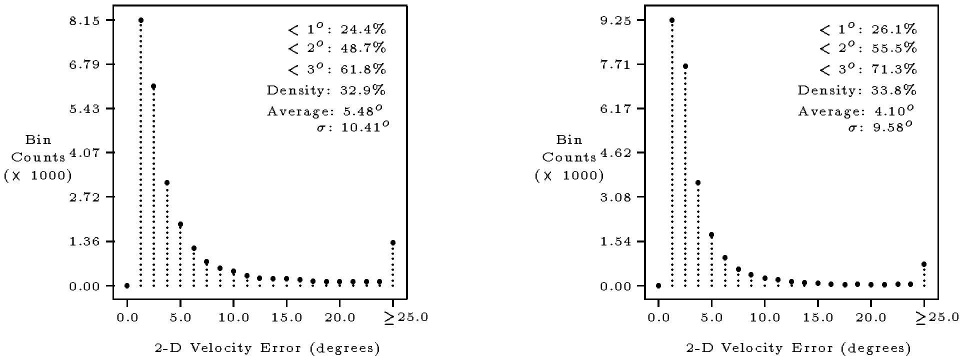
(a) **Heeger (level 0)**

(b) **Fleet and Jepson**

(c) **Waxman** ($\sigma_f = 2.0$)

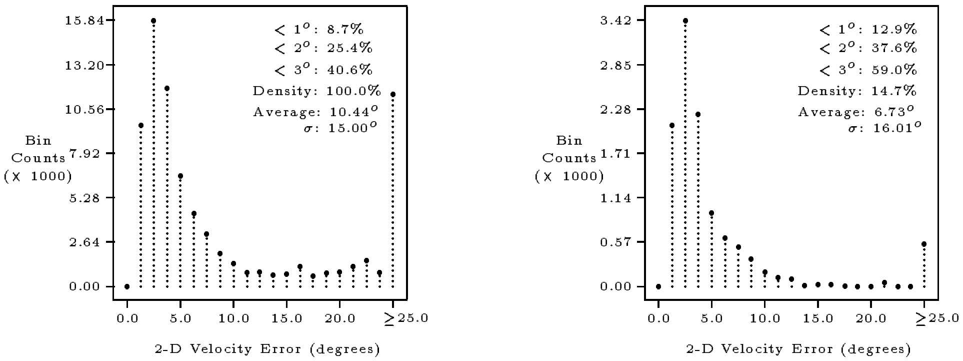
Figure E.3: **Diverging Tree:** Flow fields produced **Heeger**, **Fleet and Jepson** and **Waxman et al.**.

F Yosemite Results



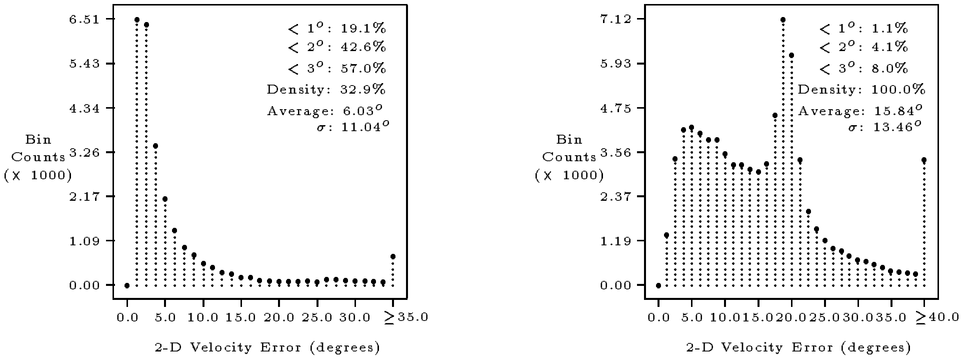
(F.12a) (F.12b)

Histogram F.12: **Yosemite Fly-Though sequence:** Error Histograms for (a) modified **Horn and Schunck** with $||\nabla I||_2 \geq 5.0$ thresholding and (b) **Lucas and Kanade** with $\lambda_2 \geq 1.0$ thresholding.



(F.13a) (F.13b)

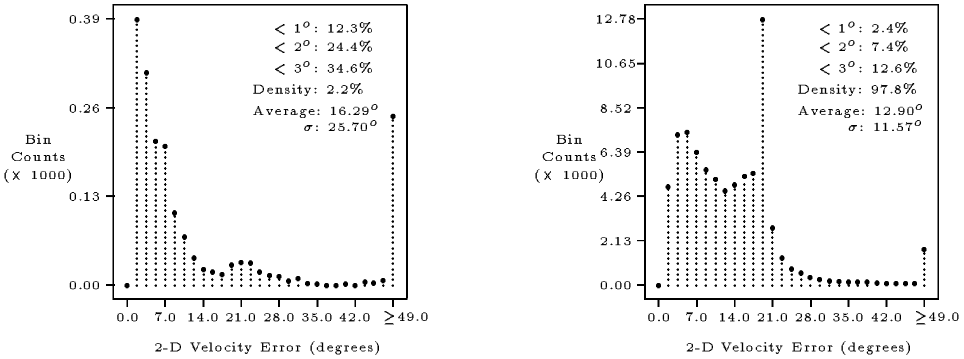
Histogram F.13: **Yosemite Fly-Through sequence:** Error Histograms for **Uras et al.**: (a) unthresholded and (b) thresholded $(det(H) \geq 1.0)$.



(F.14a)

(F.14b)

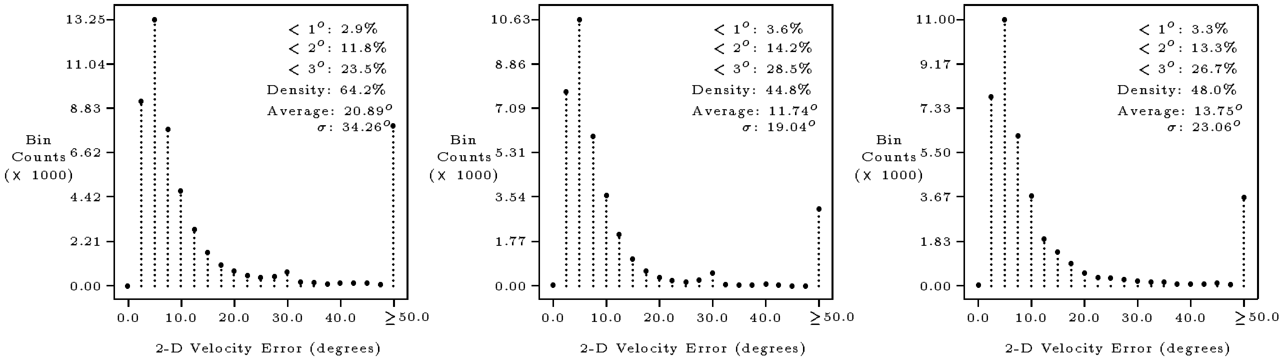
Histogram F.14: **Yosemite Fly-Through sequence:** Error Histograms for: (a) **Nagel** (with $||\nabla I||_2 \geq 5.0$) and (b) **Anandan** (unthresholded).



(E.10a)

(F.15b)

Histogram F.15: **Yosemite Fly Through sequence:** Error Histograms for **Singh**: (a) for step 1, thresholded by $\lambda_1 \geq 5.0$ and (b) for step 2, thresholded by $\lambda_1 \geq 0.1$.

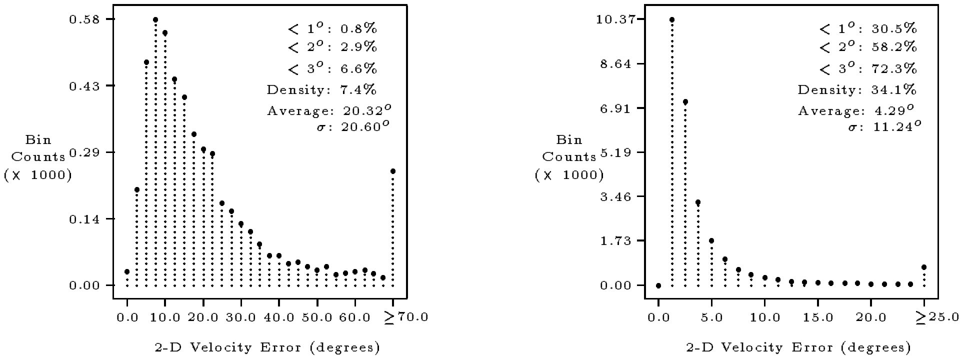


(F.16a)

(F.16b)

(F.16c)

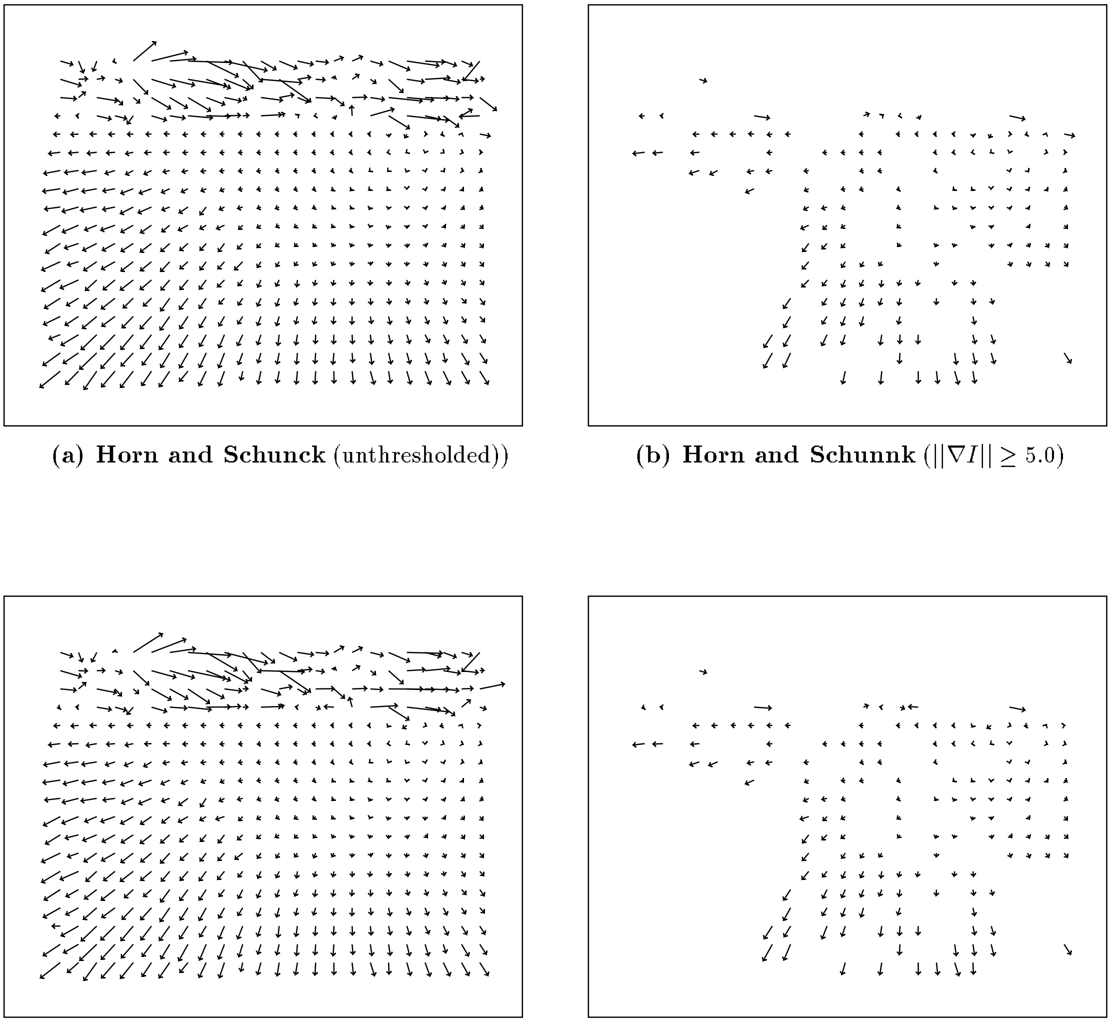
Histogram F.16: **Yosemite Fly-Through Sequence:** Error Histograms for **Heeger** for (a) level 0 of the pyramid, (b) the combined flow using the correct velocities and (c) the combined flow using the velocity from the lowest pyramid level satisfying that level's velocity range.



(F.17a)

(F.17b)

Histogram F.17: **Yosemite Fly-Through sequence:** Error Histograms for (a) **Waxman et al.** and (b) **Fleet and Jepson** ($\tau = 2.5$).



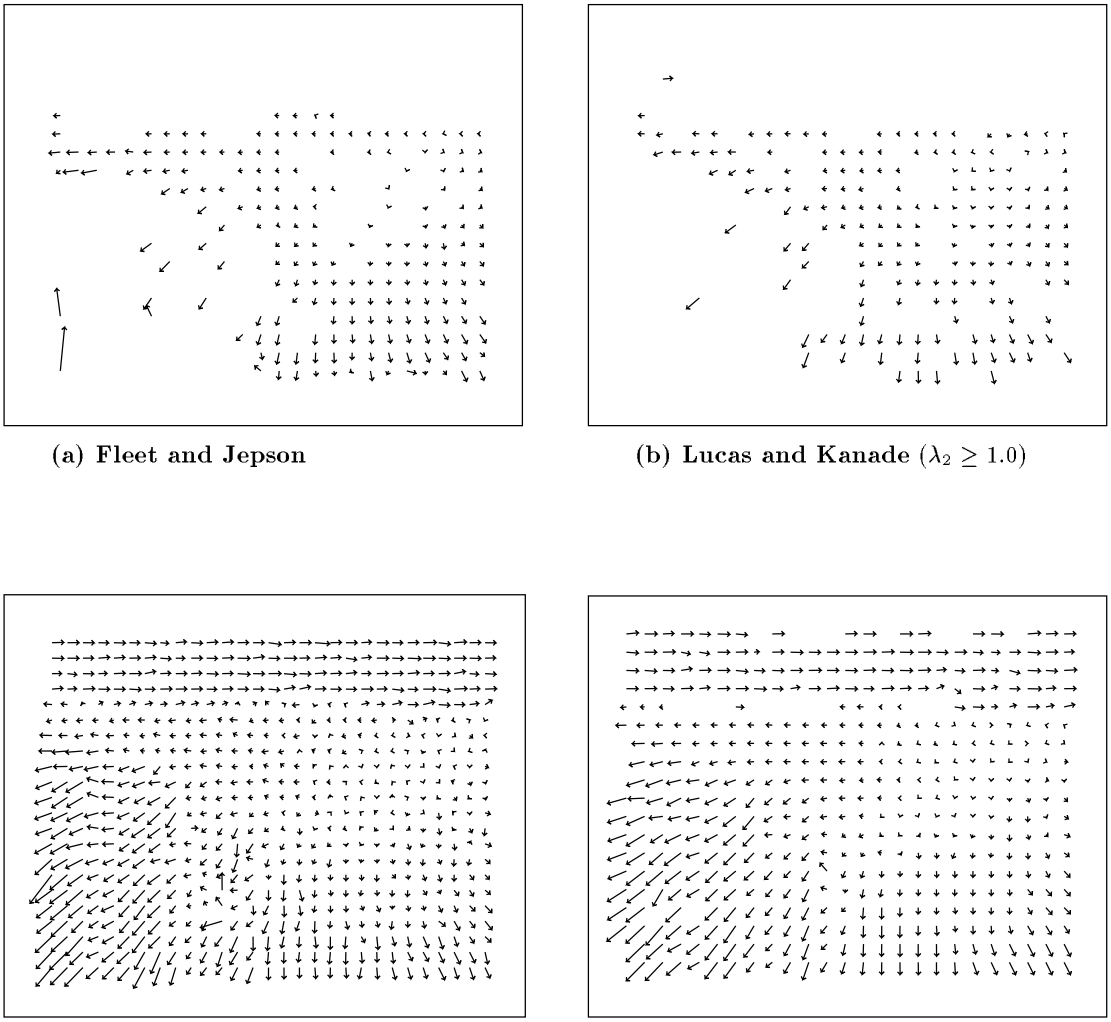
(a) **Horn and Schunck** (unthresholded))

(b) **Horn and Schunnk** ($||\nabla I|| \geq 5.0$)

(c) **Nagel** (unthresholded)

(d) **Nagel** ($det(H) \geq 1.0$)

Figure F.1: **Yosemite:** 2d velocity estimates produced by differential techniques.



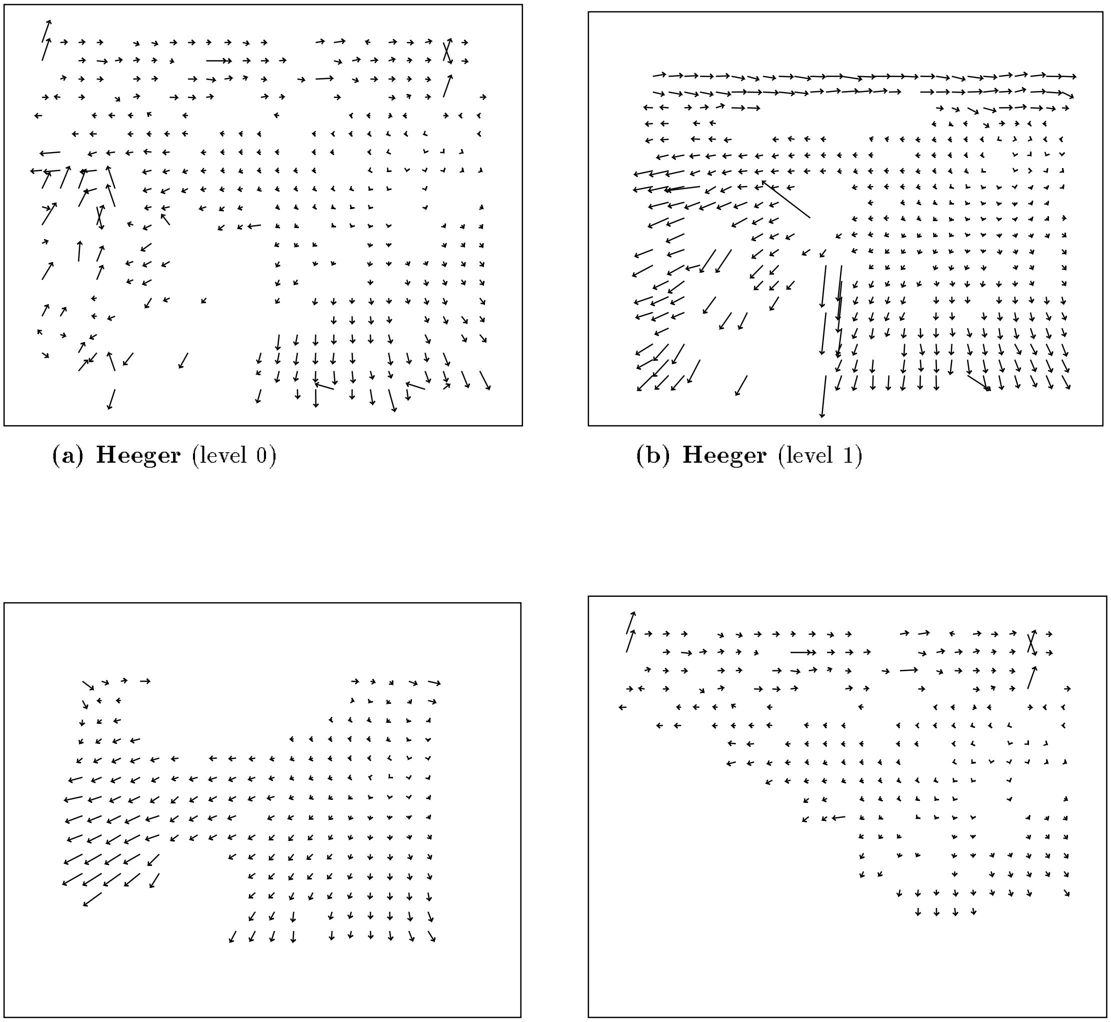
(a) Fleet and Jepson

(b) Lucas and Kanade ($\lambda_2 \geq 1.0$)

(c) Anandan

(d) Singh (step 2, $\lambda_1 < 0.1$)

Figure F.2: **Yosemite:** The flow fields produced by assorted methods.



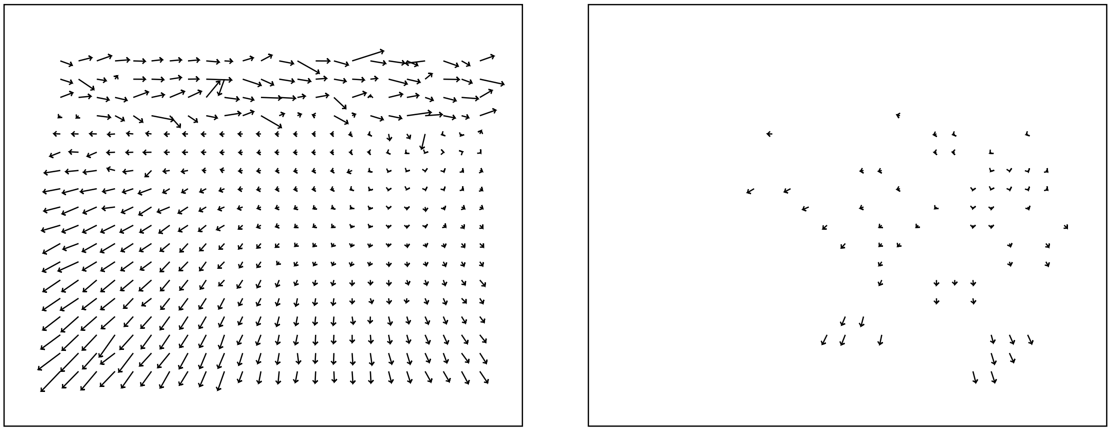
(a) Heeger (level 0)

(b) Heeger (level 1)

(c) Heeger (level 2)

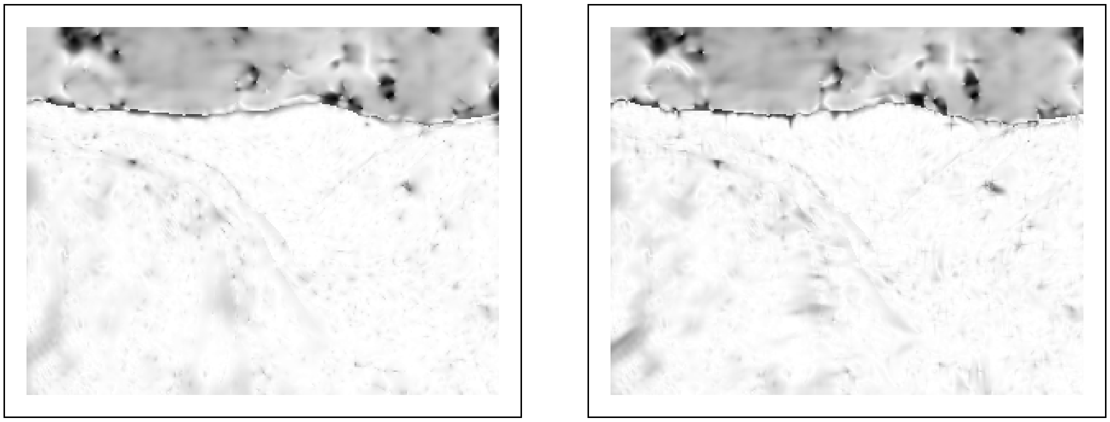
(d) Heeger (combined)

Figure F.3: **Yosemite:** Flow fields produced by **Heeger**'s method for the three levels, and then with all levels combined (choosing estimates from levels with speeds preferences corresponding to the true motion field.



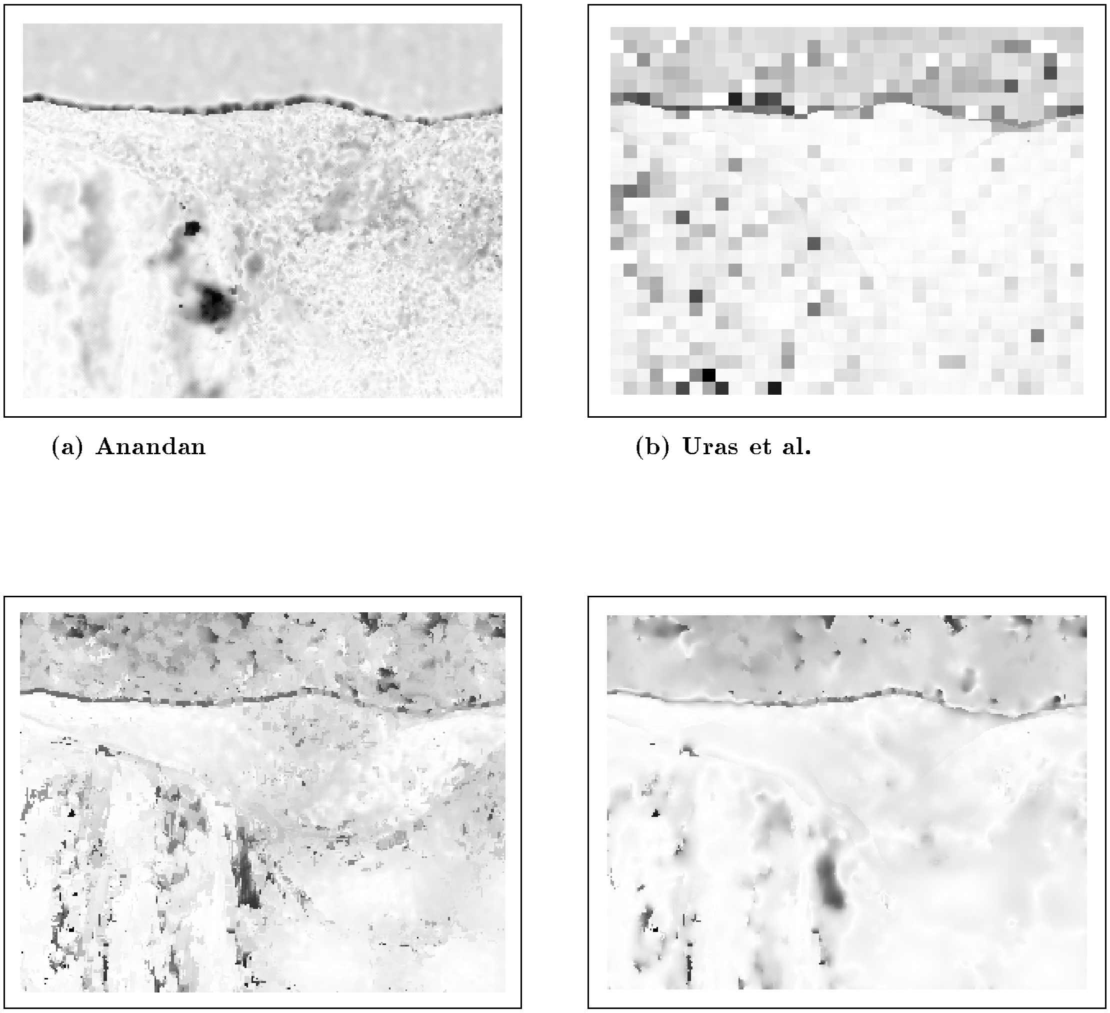
(a) Uras et al. (unthresholded) **(b) Uras et al.** ($det(H) \geq 1.0$)

Figure F.4: **Yosemite:** 2d velocity estimates produced by differential techniques.



(a) Horn and Schunck **(b) Nagel**

Figure F.5: Some error intensity images for the **Yosemite Fly-Through** sequence.



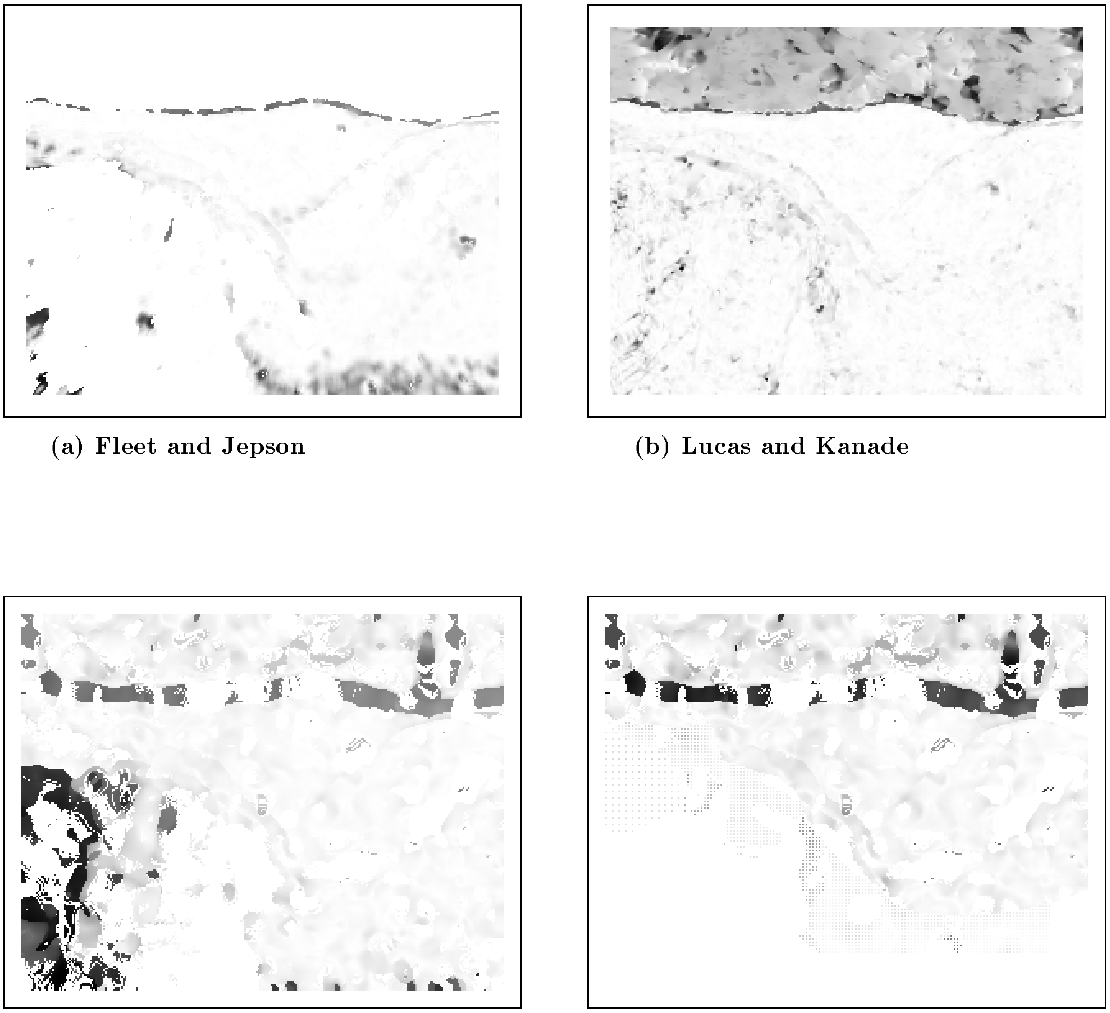
(a) Anandan

(b) Uras et al.

(c) Singh, step 1

(d) Singh, step 2

Figure F.6: Some error intensity images for the **Yosemite Fly-Through** sequence.



(a) Fleet and Jepson

(b) Lucas and Kanade

(c) Heeger, level 0, the original image

(d) Heeger, all 3 levels combined

Figure F.7: Some error intensity images for the **Yosemite Fly-Through** sequence.

G Confidence Measures Results

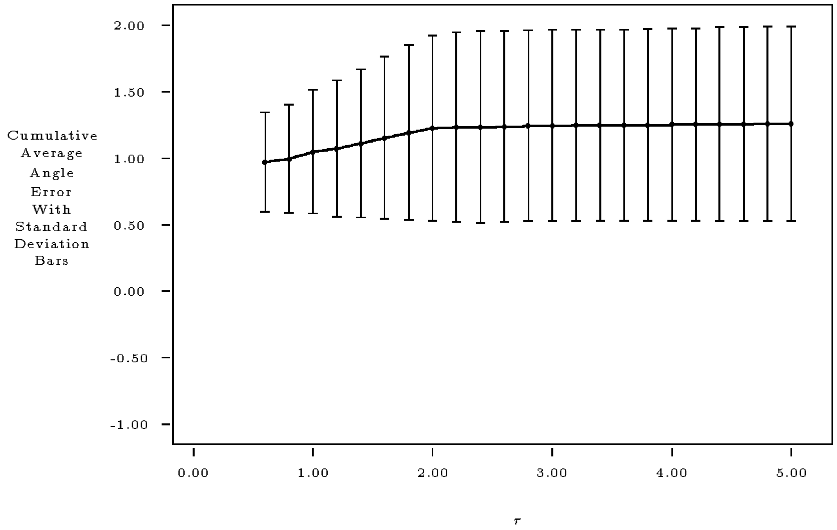


Figure G.1: Cumulative full velocity average angle error versus phase stability threshold τ for the method of **Fleet and Jepson** for the **Diverging Tree** sequence.

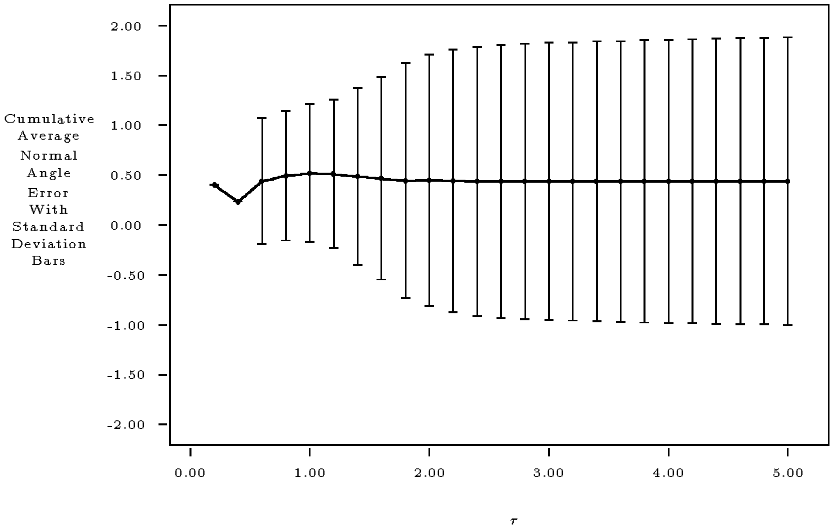


Figure G.2: Cumulative component velocity average angle error versus phase stability threshold τ for the method of **Fleet and Jepson** for the **Diverging Tree** sequence.