International Journal of Computer Vision, 5:1, 77–104 (1990)

Computation of Component Image Velocity from Local Phase Information

DAVID J. FLEET AND ALLAN D. JEPSON
Department of Computer Science, University of Toronto, 10 King's College Rd., Toronto, Ontario, Canada M5S 1A4

Abstract
We present a technique for the computation of 2D component velocity from image sequences. Initially, the image sequence is represented by a family of spatiotemporal velocity-tuned linear filters. Component velocity, computed from spatiotemporal responses of identically tuned filters, is expressed in terms of the local first-order behavior of surfaces of constant phase. Justification for this definition is discussed from the perspectives of both 2D image translation and deviations from translation that are typical in perspective projections of 3D scenes. The resulting technique is predominantly linear, efficient, and suitable for parallel processing. Moreover, it is local in space-time, robust with respect to noise, and permits multiple estimates within a single neighborhood. Promising quantitative results are reported from experiments with realistic image sequences, including cases with sizeable perspective deformation.

1 Introduction

This article addresses the quantitative measurement of velocity in image sequences. The important issues are (1) the accuracy with which velocity can be computed; (2) robustness with respect to smooth contrast variations and affine deformation (i.e., deviations from 2D image translation that are typical in perspective projections of 3D scenes); (3) localization in space-time; (4) noise robustness; and (5) the ability to discern different velocities within a single neighborhood. Our approach is based on the phase information in a local-frequency representation of the image sequence that is produced by a family of velocity-tuned linear filters. The velocity measurements are limited to *component velocity*: the projected components of 2D velocity onto directions normal to oriented structure in the image (a definition is given in section 3). The combination of these measurements to derive the *full* 2D velocity is briefly discussed.

Our reasons for concentrating on component velocity (also referred to as normal velocity) stem from a desire for local measurements, and the well-known aperture problem (Marr and Ullman 1981). Local measurements allow smoothly varying velocity fields to be estimated based on translational image velocity as opposed to more complicated descriptions of the velocity field over larger image patches. However, in narrow spatiotemporal apertures the intensity structure is often roughly one-dimensional so that only one component of the image velocity can be accurately determined. To obtain *full* 2D velocity fields, larger space-time support is therefore required. In our view, the common *assumptions* of smoothness, uniqueness, and the coherence of neighboring measurements that are involved in combining local measurements to determine 2D velocity, to fill in regions without measurements, and to reduce the effects of noise, should be viewed as aspects of *interpretation*, and as such, are distinct issues. In considering just normal components of velocity we hope to obtain more accurate estimates of motion within smaller apertures, which leads to better spatial resolution of velocity fields. As a result, the effects of image rotation and perspective distortions such as shear and dilation, as well as measurements near occlusion boundaries, may be handled more reliably.

Before we discuss the use of phase information, it is instructive to contrast local frequency-based approaches with other common approaches. The difference lies mainly in the use of a different representation of the raw image sequence, which in frequency-based approaches is provided by a collection of velocity-tuned, scale-specific linear filters. Most other techniques, including gradient-based techniques (e.g., Enkelmann 1986; Glazer 1987; Horn and Schunck 1981; Nagel 1983), correlation-based techniques (e.g., Anandan 1989; Burt et al. 1983; van Santan and Sperling 1985), and contour-based approaches (e.g., Buxton and Buxton 1984; Marr and Ullman 1981; Waxman et al. 1988), take as input an image sequence, often with some form of spatial preprocessing. The preprocessing typically takes the form of spatial Gaussian smoothing, or scale-specific, bandpass filtering. The main objectives of such filtering have been to lessen the effects of noise, and to isolate image structure of interest (e.g., zero-crossing contours, or different scales for coarse-fine analysis). Recently it was shown that the initial filters themselves could be tuned to ranges of component image velocity (Adelson and Bergen 1985, 1986; Fleet and Jepson 1984, 1989; Watson and Ahumada 1985). As a result, noise robustness is enhanced because the filters can be designed to attenuate noise through time as well as space. Furthermore, the different filters simplify some occlusion relationships (such as those produced by waving your fingers in front of your eyes) by separating image structure on the basis of its motion. In a given spatiotemporal window there may be several signal structures with different component velocities, possibly resulting from texture, partial occlusion, and transparency. The separation of image structure provided by the filters permits independent measurements of velocity at different orientations and scales within a single neighborhood. Finally, although a large number of filters may be involved, prefiltering can lead to bit compression.

One particular limitation on this initial representation concerns *velocity resolution*, that is, the number of discernible velocities within a local neighborhood. The uncertainty relation places an upper bound on the simultaneous attainment of spatiotemporal resolution and velocity resolution for a given spatiotemporal frequency range (see (Daugman 1985) for the related case in 2D). As a consequence, if the spatiotemporal support is kept small, we can expect only a limited separation of velocities. Fortunately, in most vision applications a crude degree of separation is sufficient in that the occurrence of more than two or three velocities in a small neighborhood is unlikely. This also means, however, that a subsequent stage of processing is required because the accuracy required for tasks such as the determination of ego-motion and surface parameters is greater than the tuning width of single filters (Barron 1988). Previous frequency-based approaches toward this end have been *amplitude-based*, and have sacrificed velocity resolution as a consequence of using the relative amplitudes of differently tuned filters (Adelson and Bergen 1986; Heeger 1987, 1988). Because of this, two different component velocities could be confused with a single component velocity; the sum of two different component velocities in a single neighborhood, as can occur with textured, semi-transparent, or partially occluding objects, can have the same distribution of output amplitudes as a single component velocity. This can occur even if the two component velocities lie in the tuning regions of different filters. Heeger solves for a unique 2D velocity from the amplitudes of all velocity channels in a given spatial patch (Heeger 1988; Horn and Schunck 1981). While this method appears to yield reasonable accuracy, it does not reliably resolve different velocities on the same spatial patch.

The three main advantages of *phase-based* methods are:

1. *Velocity Resolution*—Measurements can be computed from the neighboring responses of filters having identical velocity tuning, thereby preserving velocity resolution.
2. *Subpixel Accuracy*—The measurement accuracy significantly exceeds the tuning width of single filters, and is obtained without explicit subpixel reconstruction or feature localization. In our experimental work the accuracy is roughly an order of magnitude higher than that of the filter tuning.
3. *Robustness*—Phase information is robust with respect to smooth contrast changes and (near-identity) affine deformations. In particular, phase is more robust than amplitude with changes in contrast, scale, orientation, and speed. Such variations are often caused by the perspective projection of moving textured surfaces in 3D, and are deviations from the model of 2D image translation upon which most techniques are based.

Section 2 outlines the initial image representation. Section 3 gives the definition of component velocity in terms of local phase behavior and outlines a method for its measurement. This has been implemented, and several demonstrations of the accuracy and robustness of the technique are given in sections 4, 5, and 6. These experiments involve real and synthetic image sequences with sizeable time-varying perspective distortions.

2 Image Representation

The initial image representation is provided by a set of linear shift-invariant filters, each of which is tuned to a narrow range of orientation, speed, and scale, and has only local spatiotemporal support. Collectively, they span frequency space providing a complete and efficient representation. The basic constraints for velocity-tuned filters are discussed in the literature (Adelson and Bergen 1985, 1986; Fleet and Jepson 1984, 1989; Heeger 1987, 1988; Watson and Ahumada 1985).

2.1 Velocity-Tuned Filters

The utility of linear filters follows from the simple properties of 2D image translation when viewed in the frequency domain. To see this, consider a 1D intensity profile $I_0(x)$ with orientation θ, translating with velocity $\mathbf{v}_n = v\mathbf{n}$, where $\mathbf{n} = (\sin\theta, -\cos\theta)$; that is,

$$I(\mathbf{x}, t) = I_0(\mathbf{x}\cdot\mathbf{n} - tv) \qquad (1)$$

Here, $\mathbf{x} \equiv (x_1, x_2)$ and t denote space-time variables, while $\mathbf{k} \equiv (k_1, k_2)$ and ω denote their respective frequency domain variables. Also, $\mathbf{x}\cdot\mathbf{n}$ denotes the usual dot product. It can be shown that the Fourier transform of (1), $\hat{I}(\mathbf{k}, \omega) \equiv \mathfrak{F}[I(\mathbf{x}, t)]$, is

$$\hat{I}(\mathbf{k}, \omega) = \hat{I}_0(\mathbf{k}\cdot\mathbf{n})\ \delta(\mathbf{k}\cdot\mathbf{n}^{\perp})\ \delta(\mathbf{k}\cdot\mathbf{v}_n + \omega) \qquad (2)$$

where $\delta(x)$ is a Dirac delta function, and $\mathbf{n}^{\perp} = (\cos\theta, \sin\theta)$ is perpendicular to $\mathbf{n}$. From (2) note that all nonzero frequency components associated with the moving profile must lie on a line through the origin in the frequency domain. The speed v determines the angle between this line and the spatial frequency plane $\omega = 0$. The direction of motion $\mathbf{n}$ determines the orientation of the line about the ω-axis. Thus, a velocity-tuned linear filter should have its amplitude spectrum concentrated about the appropriate line in frequency space. In addition, it is important that the filter support be local in space-time; otherwise, several image properties may be merged accidentally into a single measurement.

2.2 Gabor Filters

Unfortunately, the simultaneous localization of a window in space-time and the frequency domain is restricted by the uncertainty relation (Bracewell 1978). If the radius of the window is defined as one standard deviation, then a Gaussian envelope minimizes this joint localization (Gabor 1946; Slepian 1983). This leads to the class of 3D Gabor kernels:

$$\text{Gabor}(\mathbf{x}, t;\ \mathbf{k}_0, \omega_0, C) = e^{i(\mathbf{x},t)\cdot(\mathbf{k}_0,\omega_0)}G(\mathbf{x}, t;\ C) \qquad (3)$$

where $e^{i\mathbf{x}\cdot\mathbf{k}}$ is a complex exponential, and $G(\mathbf{x}, t;\ C)$ denotes a 3D Gaussian envelope with covariance matrix C. The Fourier transform of (3) is simply a Gaussian centered at $(\mathbf{k}_0, \omega_0)$, that is,

$$\mathfrak{F}[\text{Gabor}(\mathbf{k}, \omega;\ \mathbf{k}_0, \omega_0, C)] = \hat{G}(\mathbf{k} - \mathbf{k}_0, \omega - \omega_0;\ C) \qquad (4)$$

where $\hat{G}(\mathbf{k}, \omega;\ C)$ is also a Gaussian but with covariance matrix C^{-1}. The Gaussian in (3) determines the profile of the amplitude spectrum, and the complex modulation determines its placement in the frequency domain. Here we concentrate on the use of spheroidal envelopes for which $C = \sigma I$, so that the Gaussian envelope is separable in space-time with standard deviation σ. Using this separability, along with the phase symmetries that exist among those kernels that comprise an entire family of filters, very efficient implementations are feasible. The phase-based technique described below requires bandpass, constant-phase filters, but is not restricted to separable amplitude spectra.

To be an efficient representation, the directional and scale tunings of the various filters should not overlap significantly. As well, the output of each filter should be sampled at a rate that avoids unnecessary correlation in the resulting representation. In principle, the dimension of the representation should not be larger than the original image, and in fact, substantial bit compression may be possible, such as an order of magnitude or more (Burt and Adelson 1983).

2.3 Directional Tuning

Let the extent of the amplitude spectra be measured at one standard deviation. The required number of directionally tuned filters can be viewed as a function of bandwidth. In 2D, for frequency bandwidth β (in octaves) and a central frequency f_0 about which the filters should be tuned, the standard deviation of the Gaussian envelope in frequency space is easily shown to be

$$\sigma_k = \frac{f_0(2^{\beta} - 1)}{(2^{\beta} + 1)} \qquad (5)$$

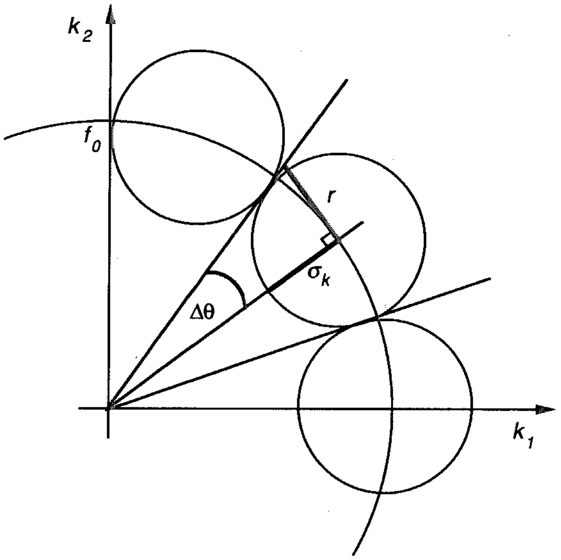

Fig. 1. Directionally tuned filters with neighboring amplitude spectra should not overlap significantly. The orientation tuning $\Delta\theta$, measured at one standard deviation σ_k, is determined by the bandwidth β. Here, $\Delta\theta = \tan\,(r/f_0)$. For small angles, with $r = \sigma_k$, it follows that $\Delta\theta \approx \sigma_k/f_0$.

Then, as shown in figure 1, for reasonably narrow directional tuning, the range of orientations for which a single filter is responsible may be expressed as $2\Delta\theta = 2\sigma_k/f_0$. The appropriate number of differently tuned filter types can then be written as

$$N(\beta, \alpha) = \left\lceil \frac{\pi}{2\Delta\theta} \right\rceil = \left\lceil \frac{\pi\,(2^\beta + 1)}{2\,(2^\theta - 1)} \right\rceil \tag{6}$$

For example, with a bandwidth of 0.8 octaves this yields 6 filter types, each tuned to a range of $2\Delta\theta = 30$ degrees. The ceiling will yield a small amount of overlap when the bandwidth is such that the filters do not tile the bandpass region cleanly.

In essence this analysis amounts to dividing the bandpass region by the area required by the amplitude spectra of the directionally tuned filters, the sizes of which are inversely proportional to the extent of their respective kernels in space. This generalizes in a straightforward way to space-time where a scale-specific bandpass region is the volume between two spheres. For a fixed volume of spatiotemporal support, the number of differently tuned velocity filters increases quadratically as a function of the peak tuning frequency (the octave number). This yields a scale-invariant distribution of filters.

2.4 Subsampling and Interpolation

We now consider the discrete sampling (representation) of the output of a single filter type. Let $R(\mathbf{x}, t)$ be the result of convolving the image $I(\mathbf{x}, t)$ with a complex-valued Gabor kernel (3). For general images, $R(\mathbf{x}, t)$ will be bandpass with its amplitude spectrum concentrated near $(\mathbf{k}_0, \omega_0)$. The minimal sampling rate for $R(\mathbf{x}, t)$ is defined according to the extent of filter's amplitude spectrum. Given a tiling of frequency space as discussed in the previous section, it is sufficient to represent frequencies in the cube centered on $(\mathbf{k}_0, \omega_0)$, having sides of length $2\sigma_k$. The appropriate sampling rate is the Nyquist rate for frequency σ_k (as can be seen by demodulating the response $R(\mathbf{x}, t)$ by $e^{-i(\mathbf{x},t)\cdot(\mathbf{k}_0,\omega_0)}$, and considering the sampling rate for the cube now centered at the origin (Bracewell 1978)). Therefore, the minimal sampling rate corresponds to a sampling distance of $\Delta s = \pi/\sigma_k = \pi\sigma$, where σ is the standard deviation of the Gaussian support in space-time. If the tiles do not overlap then the total number of samples, at the minimal rate, will be equal to the number of pixels in the space-time region that are collectively encoded; that is, there are no redundant degrees of freedom in the representation.

The computation of component velocity described below implicitly requires an interpolant for $R(\mathbf{x}, t)$. Toward this end we sacrifice efficient image encoding in favor of redundancy so that suitable interpolants are easier to compute. In particular, in the experiments reported below, we retain one (complex) sample every σ. This allows reasonably accurate interpolants to be obtained from local samples of a single filter type. Although the issue of accurate interpolation from a minimal sampling rate of $\Delta s = \pi\sigma$ is important, it is beyond the scope of this paper (cf. (Jepson 1989)).

3 Component Image Velocity

Because Gabor$(\mathbf{x}, t)$ is complex valued, so is its response $R(\mathbf{x}, t)$. Therefore, it can be expressed as

$$R(\mathbf{x}, t) = \rho(\mathbf{x}, t)e^{i\phi(\mathbf{x},t)} \tag{7}$$

where $\rho(\mathbf{x}, t)$ and $\phi(\mathbf{x}, t)$ denote its amplitude and phase components:

$$\begin{aligned} \rho(\mathbf{x}, t) &= |R(\mathbf{x}, t)| \\ &\equiv \sqrt{\mathrm{Re}[R(\mathbf{x}, t)]^2 + \mathrm{Im}[R(\mathbf{x}, t)]^2} \end{aligned} \tag{8}$$

$$\phi(\mathbf{x}, t) = \arg[R(\mathbf{x}, t)] \equiv \text{Im}[\log_e R(\mathbf{x}, t)] \in (-\pi, \pi] \qquad (9)$$

To find an appropriate definition of component image velocity, a fundamental problem is to determine which properties of the response $R(\mathbf{x}, t)$ evolve in time according to the projected motion field. We argue that the temporal evolution of (spatial) contours of constant phase provides a better approximation to the motion field than do contours of constant amplitude, and hence level contours of $R(\mathbf{x}, t)$.

3.1 Phase Robustness

If the temporal variation of image intensity was due solely to image translation, as in $I(\mathbf{x}, t) = I_0(\mathbf{x} - \mathbf{v}t)$, then it is easy to show that the filter outputs would also translate, as in $R(\mathbf{x}, t) = R_0(\mathbf{x} - \mathbf{v}t)$, for some $R_0(\mathbf{x})$. Accordingly, various standard methods (e.g., (Anandan 1989; Horn and Schunck 1981; Nagel 1983)) could then be used to measure the velocity $\mathbf{v}$. Unfortunately, image translation is a crude approximation to the typical time-varying behavior of image intensity. A more realistic model includes contrast variation and affine deformation (caused by perspective projection); and it is from this perspective that we propose the use of phase information. In particular, we argue that the evolution of phase contours provides a much better approximation to the projected motion field than the filter response $R(\mathbf{x}, t)$ in that the amplitude of response $\rho(\mathbf{x}, t)$ is generally very sensitive to changes in contrast and local variations in input scale, speed, and orientation.

Below we demonstrate the robustness of phase as compared to amplitude with two 1D examples which serve to approximate the dilation of an image as a camera approaches a planar surface. In particular, given a 1D signal $I_0(x)$ we consider the time-varying image

$$I(x, t) = I_0(x(1 - \alpha t)) \qquad (10)$$

for some $\alpha > 0$. The pattern $I_0(x)$ is simply stretched at t increases. The velocity field for this deformation is given by the motion of fixed points, say ξ, in the pattern $I_0(x)$. In image coordinates these points appear on paths generated by $x(1 - \alpha t) = \xi$. These paths are clearly visible from the inputs in Figures 2 and 3 (top left).

Figure 2 (top) shows the time-varying intensity pattern generated by equation (10) for $I_0(x) = \sin(2\pi f_0 x)$, and the time-varying response of the real part of a Gabor filter tuned to spatial frequency $2\pi f_0$ and to zero velocity. The amplitude and phase components of $R(x, t)$ are shown in figure 2 (middle). Figure 2 (bottom) shows the level contours of constant amplitude and constant phase superimposed upon the input. While the phase contours provide a good approximation to the motion field, the amplitude contours do not. In this simple example, $\rho(x, t)$ simply reflects the tuning of the filter. Because the amplitude spectrum is Gaussian-shaped, $\rho(x, t)$ depends on the local scale and speed of the input. Other things being equal, $\rho(x, t)$ increases for inputs closer to the principal frequency to which the filter is tuned. With two spatial dimensions the amplitude will depend on local orientation as well as speed and scale, all of which vary locally in typical projections of 3D scenes.

Figure 3 depicts a similar camera motion except that $I_0(x)$ is taken to be a sample from white Gaussian noise. This is a more realistic example in that $I_0(x)$ now has structure at all scales, so that different image structure will be emphasized by the filter at different times. Figure 3 (top) shows the input, and the real part of the Gabor response. The time-varying amplitude and phase components of response are given in figure 3 (middle). Figure 3 (bottom) shows their level contours superimposed upon the input. Again, note that the amplitude contours are very sensitive to small scale perturbations, and do not evolve according the motion field. On the other hand, except near a few spatiotemporal locations, the phase contours do provide a good approximation to the motion field.

Similar simulations show that phase behavior is relatively insensitive to photometric deformations that result from changes in viewing direction, surface normal, and lighting conditions. In particular, as long as the photometric effects (e.g., shadows, highlights, etc.) do not introduce power over a wide frequency band relative to the surface texture, the phase behavior of most filter outputs will be largely unaffected. For example, in the case of smooth shading gradients, the main spatiotemporal photometric effects are relatively smooth contrast variations. Although the amplitude response of filters tuned to high spatial frequencies will be affected, the phase behavior will remain stable. Conversely, in local neighborhoods where the image is dominated by a steep shading gradient (i.e., shadow boundaries with little surface texture), the spatiotemporal phase structure will reflect the motion of the shading edge. In general, we expect that for textured surfaces, the phase behavior

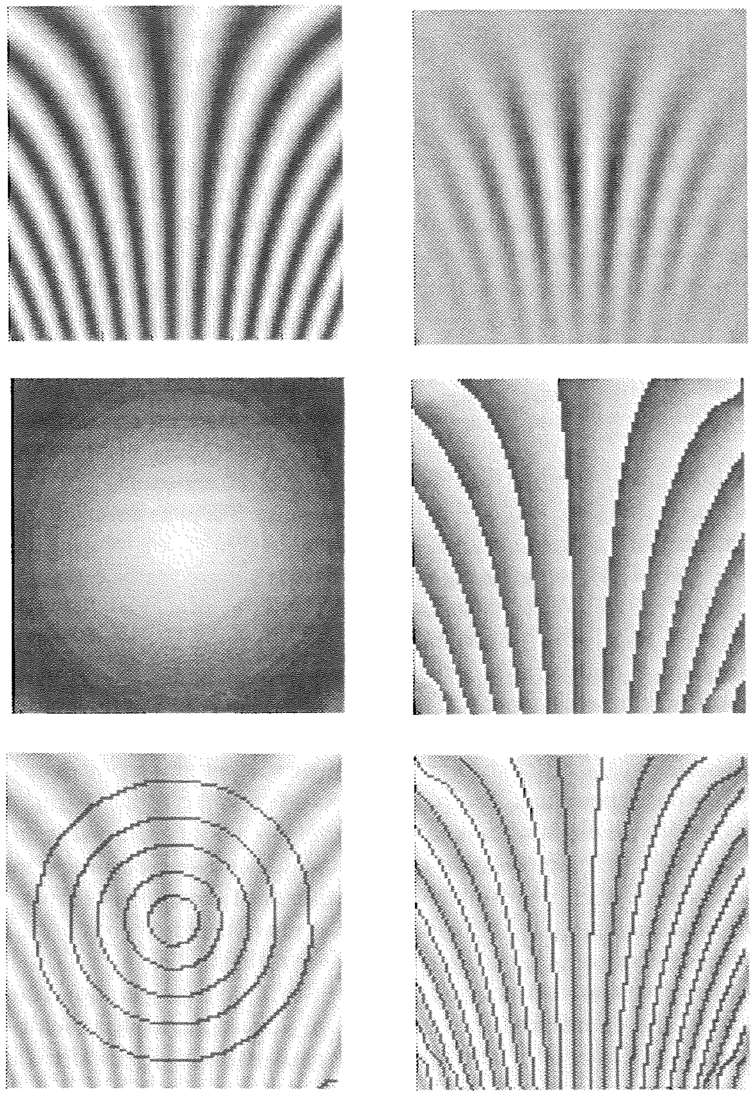

Fig. 2. (*top-left*) $I(x, t) = \sin(2\pi x f_0(1 - \alpha t))$ where $f_0 = 0.08$ pixels, $\alpha = 0.0125$, and $x = t = 0$ in the center (time on the vertical axis). The filter was tuned to speeds about 0, frequencies about f_0, and an octave bandwidth of 0.8. (*top-right*) Real part of Gabor output. (*middle*) Amplitude and phase outputs. (*bottom*) Level contours of constant amplitude and phase superimposed on the input.

will be predominantly influenced by the projected velocity field.

The basic ideas behind our approach can now be summarized. First we use the temporal evolution of constant phase contours to define image velocity. Second, a threshold technique is used to detect and remove velocity measurements in regions for which phase contours are not likely to provide reliable information about the motion field. The performance of the resulting technique is evaluated through extensive experimentation.

3.2 Component Velocity from Phase Contours

As motivated above, we consider space-time surfaces of constant phase, that is, solutions to

$$\phi(\mathbf{x}, t) = c, \qquad c \in \mathsf{R} \tag{11}$$

Assuming that constant phase surfaces evolve according to the motion field, a point $\mathbf{x}_0(t)$ moving with the motion field satisfies $\phi(\mathbf{x}_0(t), t) = c$. Differentiating with respect to t, we find that

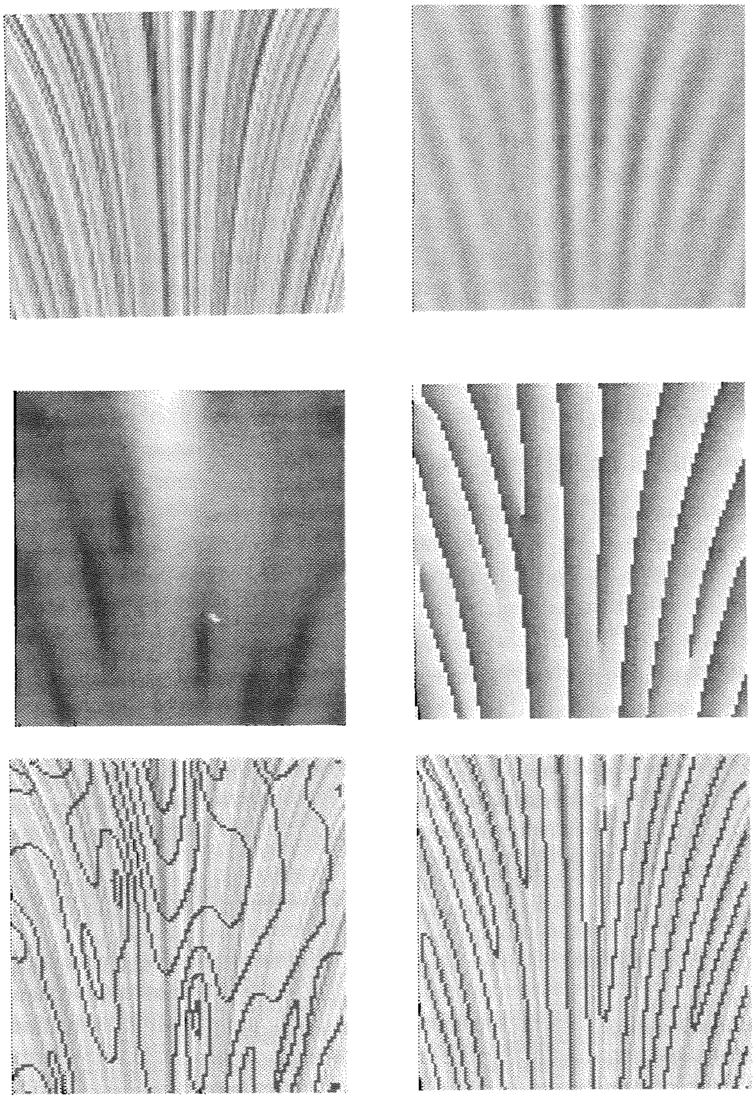

Fig. 3. The camera approaches a surface covered by white Gaussian noise. The filter tuning was identical to that in figure 2. (*top-right*) Real part of Gabor output. (*middle*) Amplitude and phase outputs. (*bottom*) Level contours of constant amplitude and phase superimposed on the input.

$$\nabla\phi(\mathbf{x}, t) \cdot (\mathbf{v}, 1) = 0 \tag{12}$$

where $\nabla\phi = (\phi_x, \phi_y, \phi_t)$, and $\mathbf{v} = (dx_0/dt, dy_0/dt)$. The aperture problem is apparent from (12) since the component of the velocity $\mathbf{v}$ in the direction perpendicular to the spatial gradient $\phi_{\mathbf{x}} = (\phi_x, \phi_y)$ is not determined. Therefore we only consider the component of $\mathbf{v}$ in the direction

$$\mathbf{n}(\mathbf{x}, t) = \frac{\phi_{\mathbf{x}}(\mathbf{x}, t)}{\|\phi_{\mathbf{x}}(\mathbf{x}, t)\|} \tag{13}$$

where $\|\cdot\|$ is the 2-norm. The combination of (12) with (13) provides our definition for component image velocity $\mathbf{v}_n$, at a point $(\mathbf{x}, t)$, as the solution of the following two equations:

$$\nabla\phi(\mathbf{x}, t) \cdot (\mathbf{v}_n, 1) = 0 \tag{14}$$

$$\mathbf{v}_n = \alpha\mathbf{n}(\mathbf{x}, t), \qquad \alpha \in \mathsf{R} \tag{15}$$

After outlining the computation of $\nabla\phi(\mathbf{x}, t)$ in the following section, we discuss this definition in relation to previously used definitions, to frequency analysis, and to phase-difference methods.

3.3 Measuring Local Phase Behavior

Rather than compute $\nabla\phi(\mathbf{x}, t)$ from the subsampled phase signal directly, we use the identity

$$\nabla\phi(\mathbf{x}, t) = \frac{\mathrm{Im}[R^*(\mathbf{x}, t)\ \nabla R(\mathbf{x}, t)]}{\rho^2(\mathbf{x}, t)} \tag{16}$$

where R^* denotes complex conjugate of R, $\mathrm{Im}[z]$ denotes the imaginary part of z, and $\mathrm{Im}[\mathbf{z}] \equiv (\mathrm{Im}[z_1], \mathrm{Im}[z_2], \mathrm{Im}[z_3])$. In terms of the real and imaginary parts of $R(\mathbf{x}, t)$ and $\nabla R(\mathbf{x}, t)$, (16) becomes

$$\begin{aligned}\nabla\phi(\mathbf{x}, t) = \{&\mathrm{Im}[\nabla R(\mathbf{x}, t)]\ \mathrm{Re}[R(\mathbf{x}, t)] \\ &- \mathrm{Re}[\nabla R(\mathbf{x}, t)]\ \mathrm{Im}[r(\mathbf{x}, t)]\} \\ &\div \{\mathrm{Re}[R(\mathbf{x}, t)]^2 + \mathrm{Im}[R(\mathbf{x}, t)]^2\}\end{aligned} \tag{17}$$

In the complex plane, equation (17) corresponds to a projection of each component of the $\nabla R(\mathbf{x}, t)$ onto the unit vector orthogonal to $R(\mathbf{x}, t)$. This formulation eliminates the need for an explicit trigonometric function to compute the phase signal from $R(\mathbf{x}, t)$. It also avoids problems arising from phase unwrapping and discontinuities.

From the subsampled representation of the filter output, it is necessary to numerically estimate $R(\mathbf{x}, t)$ and $\nabla R(\mathbf{x}, t)$. The numerical interpolation and differentiation of the filter output can be accomplished by convolution with a discrete kernel, as explained in the Appendix.

3.4 Relation to Previous Approaches

Our definition of component image velocity has much in common with standard gradient-based approaches that assume an initial stage of prefiltering in order to reduce the effects of noise, or to isolate intensity structure of interest (e.g., (Enkelmann 1986; Glazer 1987)). One difference is that the initial representation of the image sequence is provided by velocity-tuned filters. The other major difference is that, in general, the use of phase behavior provides a closer approximation to the projected motion field than does the motion of level contours of constant filter output which depend significantly on amplitude. Moreover, note that although the arguments in section 3.1 were made in terms of velocity-tuned filters, they apply equally well to lowpass and bandpass filters.

It is also of interest to compare the use phase information with zero-crossing approaches (e.g., (Buxton and Buxton 1984; Duncan and Chou 1988; Waxman et al. 1988)). To begin, note that spatiotemporal zero-crossing surfaces are surfaces of constant phase. For example, zero-crossings of the sine-Gabor ($\mathrm{Im}[R(\mathbf{x}, t)]$) output are given by (11) when $c = n\pi$, $n \in \mathbb{Z}$. Although zero-crossings of Gabor responses are not identical to zero-crossings of Laplacian of Gaussian output, they share the same relevant properties (Fleet 1990). Thus, the tracking of zero-crossing contours is in accordance with phase-based approaches. However, Daugman (1987) has argued that zero-crossings are insufficiently rich because there exist signals with discernible structure that produce no zero-crossings after bandpass filtering. Mayhew and Frisby (1981) argue that peaks, in addition to zero-crossings, are required to explain binocular stereopsis. Interestingly, crests (peaks) are also surfaces of constant phase. But both crests and zeros are special cases to which we are not restricted. With arbitrary values of phase, better use is made of the entire signal. Furthermore, subpixel detection and localization of zero-crossings is unnecessary. As a consequence, the density of velocity measurements when based on phase information will be higher than when restricted to zero-crossings. For those who match zero-crossing contours over relatively large distances between frames, note that analogous methods exist for phase information (Jenkin and Jepson 1988; Sanger 1988).

3.5 Local Frequency Analysis

We now examine the relationship between our use of the phase gradient and frequency analysis. First, consider a simplified 2D situation in which the input is a *uniform* sinusoidal waveform: $I(x) = \cos(\mathbf{k}_1 \cdot \mathbf{x})$. It is easily shown that the output $R(\mathbf{x})$ of a Gabor filter tuned to $\mathbf{k}_0$ is a complex waveform with frequency $\mathbf{k}_1$ and amplitude $\hat{G}(\mathbf{k}_0 - \mathbf{k}_1; C)$. The output phase $\phi(\mathbf{x}) = \mathbf{k}_1 \cdot \mathbf{x}$, like the phase of the input, is a linear function of $\mathbf{x}$. The spatial phase gradient is equal to the frequency $\mathbf{k}_1$, and therefore specifies the directional information. Similarly, for a sinusoidal plane-wave in space-time $\rho e^{i\phi(\mathbf{x},t)}$, where $\phi(\mathbf{x}, t) = (\mathbf{x}, t) \cdot (\mathbf{k}, \omega)$, the phase-gradient yields the spatial and temporal frequencies—i.e., $\nabla\phi(\mathbf{x}, t) = (\mathbf{k}, \omega)$.

The present situation is somewhat more involved. The Gabor output is a *nonuniform* waveform because its amplitude and frequency are not typically constant functions of space and time. However, note from (7) that $R(\mathbf{x}, t)$ may be rewritten as

$$R(\mathbf{x}, t) = \rho(\mathbf{x}, t)e^{i[\phi_M(\mathbf{x},t) + (\mathbf{x},t)\cdot(\mathbf{k}_0,\omega_0)]} \qquad (18)$$

where $M(\mathbf{x}, t) \equiv \rho(\mathbf{x}, t)e^{i\phi_M(\mathbf{x},t)}$ is a lowpass signal. Thus, $R(\mathbf{x}, t)$ is essentially a slowly varying modulation, namely $M(\mathbf{x}, t)$, of the base signal $e^{i(\mathbf{x},t)\cdot(\mathbf{k}_0,\omega_0)}$ to which the filter was tuned. The phase behavior of $M(\mathbf{x}, t)$ can be viewed as a local correction to the linear phase behavior of the base signal. Following Whitham (1974), the local instantaneous frequency can be defined as the phase gradient:

$$(\mathbf{k}(\mathbf{x}, t), \omega(\mathbf{x}, t)) \equiv \nabla\phi(\mathbf{x}, t) \qquad (19)$$

If the phase of $M(\mathbf{x}, t)$ is linear in space-time, such as $\phi_M(x) = \mathbf{k}_1 \cdot \mathbf{x} + \omega_1 t$, then $R(\mathbf{x}, t)$ is just an amplitude-modulated sinusoid with constant frequency $(\mathbf{k}_1 + \mathbf{k}_0, \omega_1 + \omega_0)$. Otherwise, the phase gradient $\nabla\phi(\mathbf{x}, t) = \nabla\phi_M(\mathbf{x}, t) + (\mathbf{k}_0, \omega_0)$ yields a local, amplitude-modulated constant-frequency approximation to $R(\mathbf{x}, t)$.

In terms of spatiotemporal frequency, component velocity may then be expressed in the usual way. At a particular location $\mathbf{x}_0$ and time t_0, the local spatial frequency $\mathbf{k}(\mathbf{x}_0, t_0)$ (which is normal to level curves of constant phase at $\mathbf{x}_0$ in the plane $t = t_0$) gives the normal direction,

$$\tilde{\mathbf{n}}(\mathbf{x}_0, t_0) = \frac{\mathbf{k}(\mathbf{x}_0, t_0)}{\|\mathbf{k}(\mathbf{x}_0, t_0)\|} \qquad (20)$$

From (19) and (20), note that $\tilde{\mathbf{n}}(\mathbf{x}_0, t_0)$ is equivalent to $\mathbf{n}(\mathbf{x}_0, t_0)$ in (13). The corresponding local orientation estimate is then $\tilde{\theta}(\mathbf{x}_0, t_0) = \arg[\mathbf{k}^{\perp}(\mathbf{x}_0, t_0)]$. Similarly, the 2D normal speed is given by

$$\tilde{v}_n(\mathbf{x}_0, t_0) = \frac{-\omega(\mathbf{x}_0, t_0)}{\|\mathbf{k}(\mathbf{x}_0, t_0)\|} \qquad (21)$$

Again, note that $\tilde{v}_n$ in (21) is equivalent to the speed α in (15). From (20) and (21), the local phase velocity of $R(\mathbf{x}, t)$ is given by

$$\tilde{\mathbf{v}}_n(\mathbf{x}, t) = \tilde{v}_n(\mathbf{x}, t)\tilde{\mathbf{n}}(\mathbf{x}, t) = \frac{-\mathbf{k}(\mathbf{x}, t)\omega(\mathbf{x}, t)}{\|\mathbf{k}(\mathbf{x}, t)\|^2} \qquad (22)$$

which is a standard expression of velocity in frequency space (e.g., see (Adelson and Bergen 1985; Fleet and Jepson 1984; Watson and Ahumada 1985)). It is also equivalent to $\mathbf{v}_n$ provided by (14) and (15). Thus, we have shown that the expression of component velocity in terms of level surfaces of constant phase is consistent with that in terms of spatial and temporal frequencies.

3.6 Phase-Difference Techniques

Interestingly, there exist somewhat similar phase-based techniques for the measurement of binocular disparity in which disparity between the left and right views is expressed in terms of phase differences between bandpass versions of the left and right images (Jenkin and Jepson 1988; Sanger 1988). Relative to the local wavelength of the filter output, the phase difference provides a measure of the shift required to match the phase of one view with that of the other. Because the bandwidths of the filters were relatively narrow (near one octave), the local wavelength was assumed to be equal to the principal wavelength to which the filters were tuned. In comparison to our use of $\nabla\phi$, the phase difference between the left and right filter outputs can be viewed as an approximation to one component of the phase gradient based on linear interpolation. The other component of the phase gradient is given implicitly in the assumption that the local wavelength of the filter output is determined by the filter tuning. Thus, we expect errors in phase-difference techniques to arise because of the difference between the local wavelength of response and peak tuning wavelength of the filter. Errors will also arise because of the implicit form of interpolation. That is, linear interpolation yields substantial error in signal reconstruction unless the sampling rate is prohibitively high (Gardenhire 1964).

In our earlier motion experiments, we used linear interpolation to measure all components of the gradient thereby removing errors due to discrepancies between the response wavelength and the filter tuning. It was found that, although many of the errors due to poor interpolation are detectable because they give estimates far from the frequencies to which the filters were tuned, there is a large decrease in the density of accurate measurements. With respect to *any* phase-based technique, errors in velocity measurements can be expected because of (1) input noise, (2) deviation of the input behavior from image translation, and (3) quantization and noise introduced through signal encoding and the form of (implicit) reconstruction. For the technique presented in this paper, component velocity was defined in terms of surfaces of constant phase, and hence the phase gradient. This is a considerble improvement over the previous phase-difference techniques both theoretically and in practice since more accurate forms of interpolation and measurement follow naturally.

4 Experimental Results

The technique described above has been implemented and applied to a variety of image sequences. To obtain controlled yet realistic image sequences a 3D graphics package was used to generate a simple geometric environment in which real images were used to create surface texture. Rendering the scene from a sequence of camera positions under perspective projection produced image sequences complete with perspective distortions such as shear, dilation/contraction, and rotation. Image speeds ranged between 0 and 4 pixels/frame. To allow for comparison of results from different motions (image deformations) we concentrate here on the *tree* image shown in figure 4 (which shows three frames from the image sequence used in Experiment 4). Results from other images are reported in (Fleet 1990). In addition to these sequences we report results from sequences with additive Gaussian noise, transparency, the *Yosemite* sequence used by Heeger (1988), and the *Hamburg Taxi-Cab* sequence used by Nagel and Enkelmann (Enkelmann 1986; Nagel 1983; Nagel and Enkelmann 1986).

At present, we use only those Gabor filters that cover a single spatiotemporal bandpass region of 0.8 octaves. The small bandwidth is important because it reduces sensitivity to mean illumination and low frequencies. Natural images have significant amounts of power at low frequencies (Netravali and Limb 1980), which, if passed by the filters, will cause unwanted aliasing after subsampling, and therefore distortion of local phase. The dc amplitude sensitivity for a Gabor with octave bandwidth β is $e^{-b^2/2}$ where $b = (2^\beta + 1)/(2^\beta - 1)$; for $\beta = 0.8$ this is roughly 10^{-3}. In order to remove this residual dc sensitivity, a low-pass version of the input (scaled by $e^{-b^2/2}$ can be subtracted from the real (cosine) part of each Gabor output. This produces an altered cosine-Gabor kernel of the form $[\cos(\mathbf{x} \cdot \mathbf{k}_f + t\omega_f) - e^{-b^2/2}]\, G(\mathbf{x}, t; \sigma I)$, but does not significantly alter the quadrature relationship with the corresponding sine-Gabor kernel. (With this modified kernel we found that errors were reduced by approximately 5%.) In total there were 23 complex kernels: 6 tuned to speeds about 0 with preferred directions at multiples of 30 degrees; 10 tuned to speeds of $1/\sqrt{3}$ with directions at every 36 degrees; 6 tuned to speeds of $\sqrt{3}$ with directions every 60 degrees; and a *flicker* channel tuned to nonzero temporal frequencies and zero spatial frequency. The 46 real 3D convolutions can be implemented as 75 1D stages (Fleet 1990). The organization is scale invariant so that other frequency bands would have a similar arrangement of filter tunings. Although the number of filters may appear large, the subsampling (one complex sample every $\lfloor \sigma \rfloor$ in space and time) and quantization (to 8 bits) ensures that the representation remains reasonably efficient.

Because the filter bank is scale invariant, the velocity resolution available at each scale is constant. But, as the spatiotemporal filter support increases the spatiotemporal resolution deteriorates. In the experiments reported below we used high spatiotemporal frequencies, thereby emphasizing spatiotemporal resolution with a small support width. Unless stated otherwise, the filters were tuned to a spatiotemporal wavelength of 4 pixels (frames). The support radius at one standard deviation was 2.4 pixels (2.4 frames) in space (time, respectively); the total operator width, out to 3σ, was 15. By comparison, in the human fovea the cones are roughly 20 arc seconds apart and have a temporal

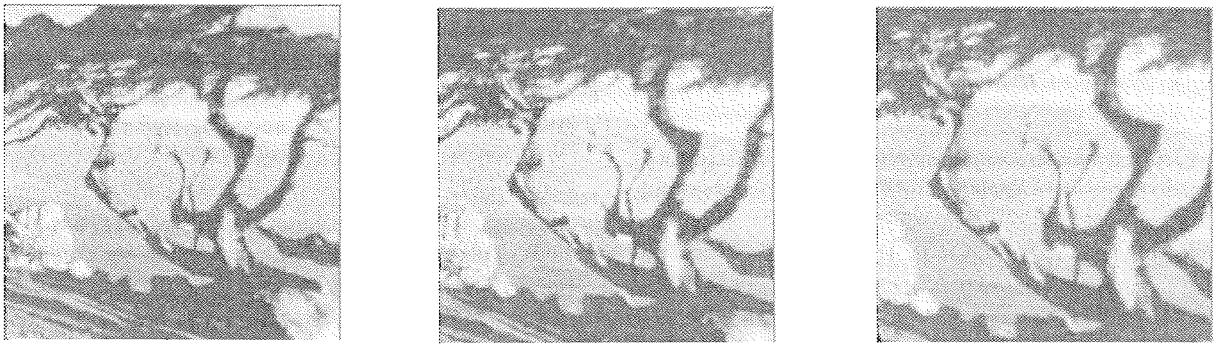

Fig. 4. Frames 10, 20, and 30, from experiment 4 with camera motion along the line of sight.

integration time of roughly 20 msec. In these terms, the spatiotemporal extent of our filters would be roughly 2 arc minutes and 0.2 seconds. Given this small spatial extent, and the accuracy of the method as demonstrated below, the trade-off between spatiotemporal and velocity resolution does not seem to present a significant limit for practical applications.

4.1 Error Measure

In principle, all component velocities $v_n\mathbf{n}$ that are generated by a given 2D velocity $\mathbf{v}$ satisfy

$$(\mathbf{v},\ 1) \cdot (\mathbf{n},\ -v_n) = 0 \tag{23}$$

where $(\mathbf{v},\ 1)$ is the direction vector (in space-time) tangent to particle paths for which the instantaneous velocity is $\mathbf{v}$. Equation (23) implies that all component velocities consistent with $\mathbf{v}$, represented as $(\mathbf{n},\ -v_n)$, must lie in the plane normal to $(\mathbf{v},\ 1)$. Conversely, each component velocity estimate $\tilde{v}_n\tilde{\mathbf{n}}$ constrains the local 2D velocity to the plane normal to $(\tilde{\mathbf{n}},\ -\tilde{v}_n)$. Therefore, as discussed in section 5, (23) provides the basis for the estimation of 2D velocity $\tilde{\mathbf{v}}$ from estimates of component velocity. Accordingly, an appropriate measure of component velocity error, given a 2D velocity $\mathbf{v}$ and an estimate of component velocity $\tilde{v}_n\tilde{\mathbf{n}}$, is the angle ψ_ϵ between the estimate and the constraint plane normal to $(\mathbf{v},\ 1)$; i.e.,

$$\psi_\epsilon = \arcsin\left[\frac{(\mathbf{v},\ 1) \cdot (\tilde{\mathbf{n}},\ -\tilde{v}_n)}{\sqrt{1 + \|\mathbf{v}\|^2}\sqrt{1 + \tilde{v}_n^2}}\right] \tag{24}$$

The appropriateness of this error measure follows from the use of (23) as the basis for the computation of 2D velocity from the component estimates, and is discussed in section 5.

Velocity estimates obtained from all 22 filters (excluding the flicker channel) are presented collectively. Two main constraints are used to discard those velocity estimates that are deemed *unreliable*:

1. *Frequency Constraint*—The computed local frequency $(\tilde{\mathbf{k}},\ \tilde{\omega})$ must satisfy

$$\|(\mathbf{k}_0,\ \omega_0) - (\tilde{\mathbf{k}},\ \tilde{\omega})\| < 1.2\sigma_k \tag{25}$$

where $(\mathbf{k}_0,\ \omega_0)$ is the peak tuning frequency, and σ_k denotes the standard deviation of the filter's amplitude spectrum. That is, local frequencies up to 20% outside the nominal tuning range of the filters are accepted. This is based on current work concerning the pathological behavior of phase signals, the results of which are forthcoming (Fleet 1990).

2. *Amplitude Constraints*—The local signal amplitude must be as large as the average local amplitude, and at least 5% of the largest response amplitude (across all filters at that frame). Local amplitude was computed as a Gaussian weighted average about the pixel in question, averaged over all filters. The standard deviation of the Gaussian was identical to that of the initial Gabor filters. The amplitude constraints detect situations in which there was no significant power at frequencies near $(\mathbf{k}_0,\ \omega_0)$, for in such cases the local response may be dominated by noise and quantization error which makes the measurement of the phase gradient very sensitive.

4.2 Translational Camera Motion

In the first set of experiments we used a class of image velocity fields similar to those considered in (Koenderink and van Doorn 1976) in which the camera undergoes translational motion with respect to a textured, planar surface.

4.2.1 Camera and Scene Geometry. Let the world coordinate system be camera-centered, with the instantaneous line of sight defined as the Z-axis, a focal length of 1, and a relatively wide field of view subtending 53 degrees (75 degrees diagonally). The scene consists of a single planar surface $P(X,\ Y)$, the gradient of which is expressed as $(\tan\beta,\ \tan\gamma)$. The camera's motion is contained in the XZ-plane, and is expressed as an angle α, measured relative to the line of sight. The camera velocity is given by

$$\mathbf{v}_c = v_c(\sin\alpha,\ 0,\ \cos\alpha) \tag{26}$$

where v_c is the camera speed expressed in world coordinates (focal length units) per frame. Finally, the distance to the surface along the line of sight is $d(t) = d_0 + v_c t(\sin\alpha\tan\beta - \cos\alpha)$, where d_0 is the distance at time $t = 0$.

Points on the surface with coordinates (X, Y, Z) project onto the image plane such that $\mathbf{x} = (X/Z,\ Y/Z)$. Surface depth, as a function of image location, is given by $Z = d(t)/(1 + x\tan\beta + y\tan\gamma)$. Following Longuet-Higgins and Pradzny (1980), it can be shown that the 2D image velocity *induced* at location x at time t is

$$\mathrm{v}(\mathrm{x};\, t) = \left(\frac{v_c(1 + x \tan \beta + y \tan \gamma)}{d(t)}\right) \times \quad (x \cos \alpha - \sin \alpha,\, y \cos \alpha) \quad (27)$$

From the partial derivatives of (27), the magnitudes of divergence (div v), curl (curl v), and deformation (def v) can be determined. These quantities are of interest below in determining the extent to which the projected image velocity deviates from a model of local translation. Note that for $\alpha \neq 90$ the velocity field is quadratic. Image speeds can vary significantly throughout the image, as does the direction of motion near the focus of expansion. Also note that these quantities change nonlinearly through time as the distance to the surface changes.

Two types of translational motion are reported in detail: (1) with the camera moving perpendicular to the line of sight, as if one were looking at the ground while moving, or out the window of a train ($\alpha = 90$); and (2) with the camera moving along the line of sight ($\alpha = 0$). The camera and scene parameters, insofar as they change with each image sequence, are given below.

4.2.2 Side-View Motion. Experiment 1 ($\alpha = 90$; $\beta = \gamma = 0$; $d_0 = 15$; $v_c = 0.075$): The first sequence most closely resembles image translation as the surface is perpendicular to the line of sight and image velocity is constant. The image velocity was 0.75 pixels/frame. Figure 5 (left) shows the histogram of the component velocity errors (24). The inset gives the proportions of the accepted estimates that had errors (in absolute value) less than 1, 2, and 3 degrees. Figure 5 (right) shows mean error and standard deviation bars as functions of the distance between the estimated local frequencies and the principal filter tunings of the respective channels from which they were obtained (as in (25)). Notice the increase in error as the distance from the filter tuning increases. Although the frequency cutoff used to select estimates to compute the histogram in figure 5 (left) was $1.2\sigma_k$, it is clear from figure 5 (right) that the errors are still well-behaved beyond this boundary. Up to the cut-off most errors are less than 1 degree.

Figure 6 (left) shows the component velocity error behavior as a function of the estimated orientation. Notice the relatively even distribution of errors as a function of orientation. This is an important property for any scheme used to infer 2D velocity from the component estimates. Figure 6 (right) shows the distribution of errors across the image. Intensity is proportional to the average (absolute) error per estimate; entirely black regions denote regions containing no estimates that satisfied the frequency and amplitude constraints. Over 70% of the pixels had at least one component velocity estimate (c.f. figure 4).

Experiment 2 ($\alpha = 90$; $\beta = 15$; $\gamma = 0$; $d_0 = 13$; $v_c = 0.173$): The second case involved faster speeds and a nonzero surface gradient. This produces a speed gradient in the direction of image velocity, and differs from Experiment 1 in that div v and def v are nonzero. As $|\beta|$ increases so does the speed gradient. Here, image speeds ranged from 1.73 pixels/frame on the left side of the image to 2.63 on the right. Despite the faster, nonuniform image velocities, the results are similar to those shown in figures 5 and 6. In particular, the proportions of estimates with errors below 1, 2, and 3 degrees were 90.2%, 98.6%, and 99.7%.

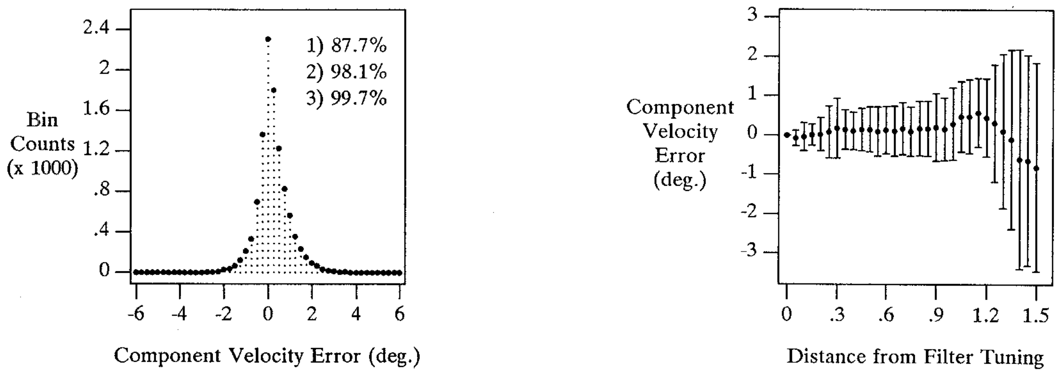

Fig. 5. Experiment 1: *side-view motion*. (*left*) Histogram of errors. The inset shows the proportions of estimates with errors less than 1, 2, and 3 degrees. (*right*) Mean error and standard deviation bars as a function of distance between the estimated local frequencies and the filter tunings.

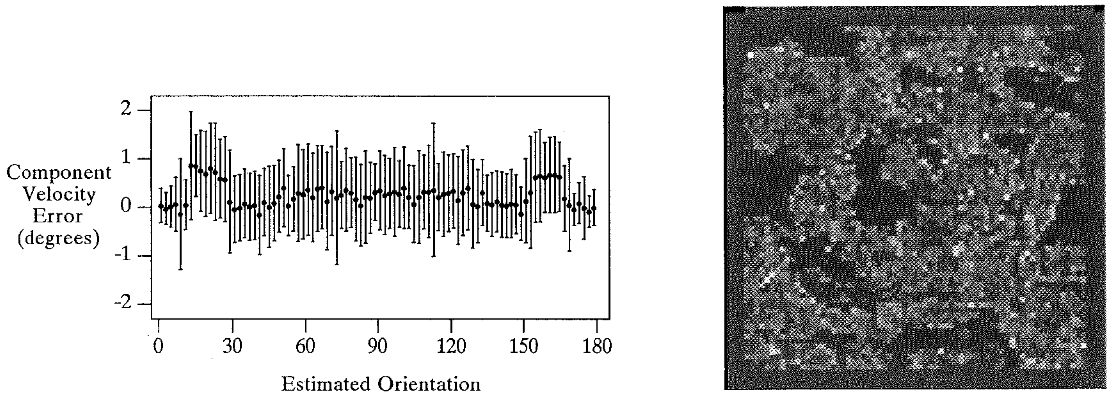

Fig. 6. (*left*) Mean component velocity error and standard deviation bars are shown as a function of the estimated orientation. (*right*) Average absolute error per estimate is shown (as intensity) as a function of image location. Black regions denote no estimates.

EXPERIMENT 3 ($\alpha = 90$; $\beta = 0$; $\gamma = 20$; $d_0 = 13.5$; $v_c = 0.135$): The final side-view sequence had a vertical surface gradient. This produces a speed gradient that is perpendicular to the direction of image velocity. As a consequence, curl **v** and def **v** were nonzero while div $\mathbf{v} = 0$. The perceived effect is motion parallax, the magnitude of which depends on $|\gamma|$; in this case it was quite visible. The image speeds ranged from 1.2 pixels/frame at the top of the image to 1.8 at the bottom. The results are again similar to those in figures 5 and 6, with 86.7%, 97.6%, and 99.3% of the estimates having errors less than 1, 2, and 3 degrees. Other cases with different images and more substantial curl and shear also yield similar results (Fleet 1990).

4.2.3 Front-View Motion. In the next experiment the camera moved along the line of sight. Image velocities point radially out from the center of the image (the focus of expansion), with speeds increasing toward the image boundaries; div **v** is nonzero and varies as a function of time. As a result of the dilation there are local variations in scale, speed, and the direction of image velocity, especially near the focus of expansion. Relative to the accuracy with which we hope to measure velocity, these local changes in the direction of motion, speed, and scale constitute significant deviations from a model of image translation. Also note that there did exist significant structure at spatiotemporal frequencies higher than those to which the filters were tuned. Although relatively high, the frequency range to which the filters were tuned was not at the Nyquist limit. Furthermore, as time progressed and the camera moved closer to the surface, new structure appeared because the initial frames were (effectively) down-sampled versions of the original. Subsequent frames were rendered by reprojecting the surface (and the texture) at each frame, and not by simply interpolating the first frame.

EXPERIMENT 4 ($\alpha = 0$; $\beta = 20$; $\gamma = 0$; $d_0 = 13$; $v_c = 0.2$): The time to collision was 65 frames and the induced image speeds ranged from 0 in the center of the image to 1.4 pixels/frame on the left, and 2 on the right. With 2D velocity expressed as a direction vector in space-time, and speed expressed in degrees (i.e., arctan $\|\mathbf{v}\|$), this local speed variation amounts to speed differences of close to 1.0 degree between neighboring pixels (about 15 degrees over the entire operator width). In addition, over the width of temporal support, the distance to the surface $d(t)$ decreased by about 20%. As a consequence, div **v** changes significantly. Figure 7 shows the histogram of component velocity errors (left) as well as the error behavior as a function of the distance between local frequency and the peak tuning frequencies of the filters. As above, the errors are still well behaved. Although the estimates are not accurate to within 1 degree of the true velocity, the proportions of estimates with errors less than 2 and 3 degrees are high (similar to experiments 1-3). This accuracy is good considering the speed, direction, and scale changes within the spatiotemporal support width of the filters. The distribution of errors over the image is again similar to that shown in figure 6. Similar performance was observed in other tests with contraction and even stronger divergence. With a slower approach to the surface, and therefore less dilation per frame, the results improve (Fleet 1990).

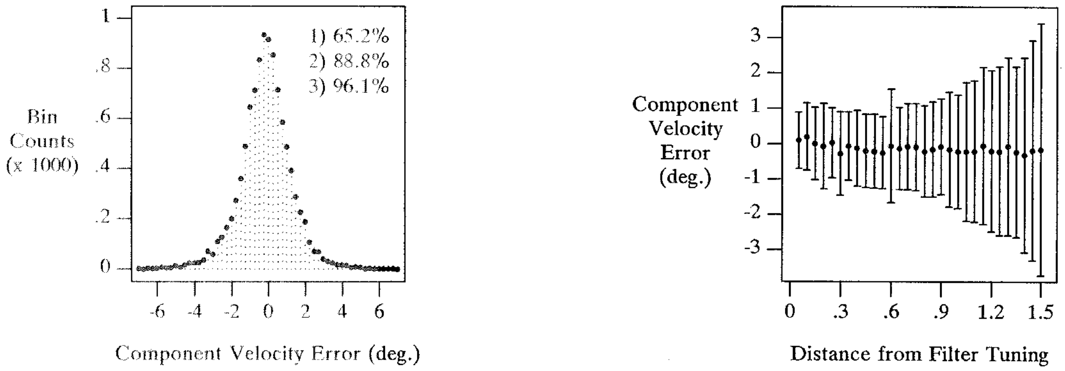

Fig. 7. Experiment 4: *front-view motion*. Image velocity is 0 at the center, 1.4 pixels/frame on the left, and 2 on the right. (*left*) Histogram of component velocity errors. (*right*) Error behavior as a function of distance from filter tuning.

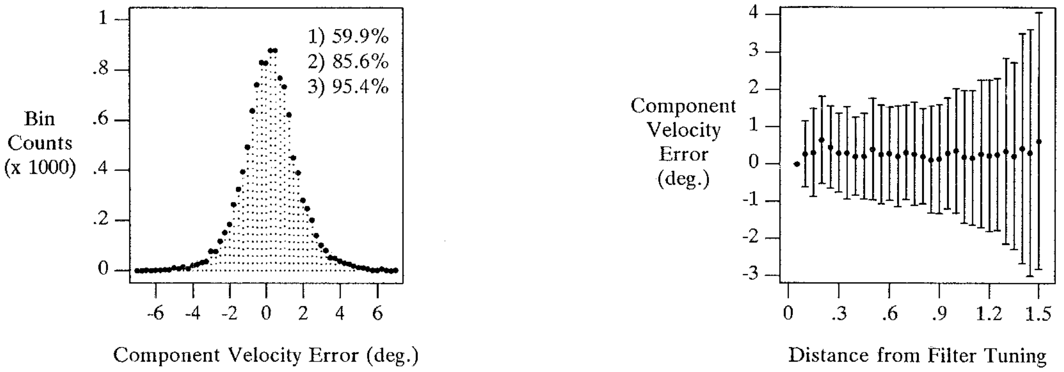

Fig. 8. Experiment 5: *Image rotation*. Image speeds ranged from 0 at the center to 1.31 at the edges of the image (1.85 in the corners). (*left*) Histogram of component velocity errors. (*right*) Error behavior as a function of distance from filter tuning.

4.3 Image Rotation

The image velocity fields considered in the first four experiments were dominated by translation and dilation (despite the nonzero curl in experiment 3). This experiment deals explicitly with image rotation.

Experiment 5 (counter-clockwise rotation, 1 degree/frame): The image velocity fields that result from instantaneous camera rotation (with no translational component) do not depend on the depth of scene points (Longuet-Higgins and Prazdny 1980). Therefore, we test only the simplest case in which the camera rotates while the planar surface remains normal to the line of sight. With rotation of 1 degree per frame, and an image size of 150×150, the image speeds ranged from 0 in the center of the image to 1.31 pixels/frame at the edges (1.85 in the corners). The fixed-point of rotation is a flow singularity. The results, shown in figure 8, are similar to the dilation sequence above.

4.4 Additive Noise

We now consider the robustness of phase information when significant amounts of noise degrade the input. Spatiotemporal white Gaussian noise was added to several image sequences to demonstrate the error in component velocity estimates as a function of the noise level. Here we report results from two sequences—experiments 2 and 4. The noise had a mean of zero with standard deviations σ_n up to 50. Relative to an 8-bit image this is a significant amount of noise (cf. figure 9).

Figure 10 shows the decrease in the proportions of errors falling within 1, 2, and 3 degrees of the correct velocity as a function of σ_n. As expected, the accuracy deteriorates with increased noise levels. However, note that the total number of estimates that survived the thresholds remained roughly constant. Furthermore, the deterioration occurred smoothly and relatively slowly. The high proportion of estimates within 3 degrees of

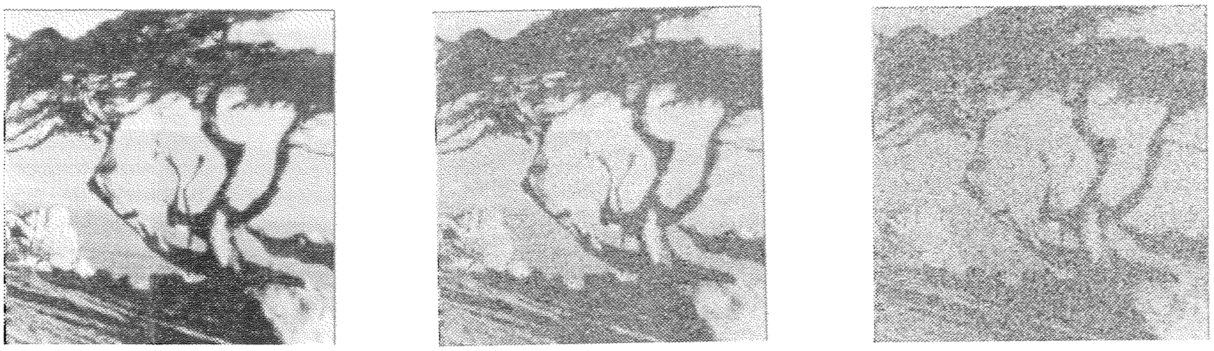

Fig. 9. The *tree* image (*left*) is shown with additive mean-zero white Gaussian noise with standard deviations σ_n = 15 (*middle*) and σ_n = 40 (*right*).

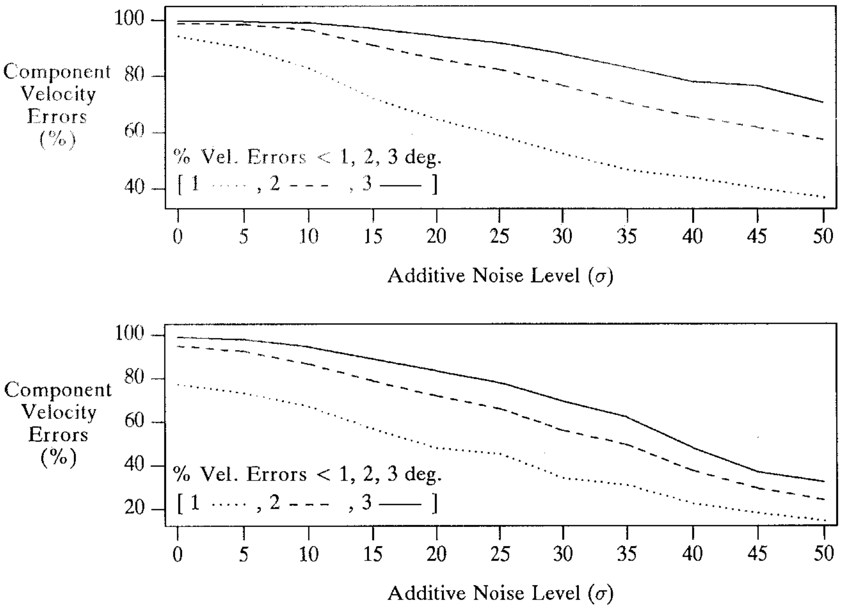

Fig. 10. Proportions of estimates with errors below 1, 2, and 3 degrees as a function of σ_n for single filters from the family. (*top*) Image sequence of experiment 2, and a filter tuned to normal velocities about 1.732 pixels/frame down to the right. (*bottom*) Image sequence of experiment 5, and a filter tuned to normal velocities about 0.577 pixels/frame up toward the top.

the true velocity is especially encouraging. The sharper increase in errors in figure 10 (bottom) arises from the generally poorer performance in experiment 4, in conjunction with the tuning of the filter to the low-contrast horizontal image structure near the center of the image. As is clear in figure 9, these regions are easily degraded by small amounts of noise.

4.5 Rotating Sphere and Yosemite Sequence

The next two experiments involved synthetic image sequences depicting more complex scene structure. The first sequence contained a rotating, textured sphere (figure 11). The second was the Yosemite image sequence used by Heeger (1988) (figure 12).

For the rotating sphere the image size was 200×200 with an angular field of view of 40 degrees (54 degrees diagonally). The distance between the centroid of the sphere and the focal point of the camera, was 4 times the radius of the sphere. The rotation was 1.5 degrees/frame about its centroid, with the axis of rotation given by (45, 35) (degrees) in standard spherical coordinates. This induces image speeds of up to 2.6 pixels/frame along the equator, and 0 at the fixed point (see figure 19).

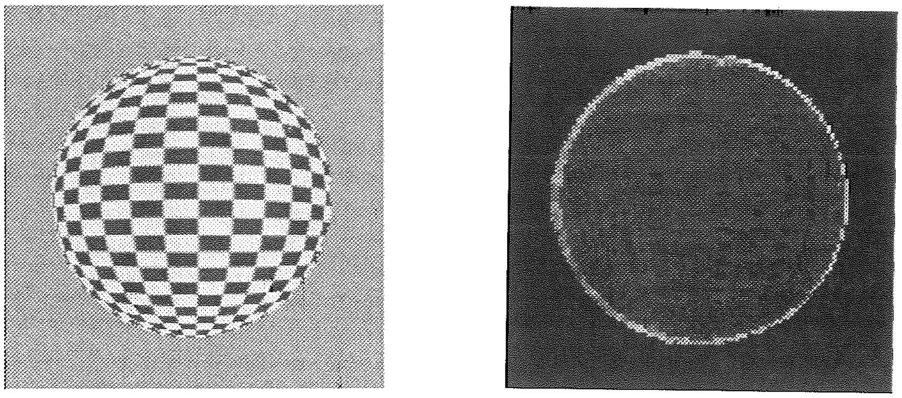

Fig. 11. Rotating sphere. 2-D image velocities were up to 2.6 pixel/frame along the equator and zero at the fixed point. (*left*) One frame of the image sequence. (*right*) Average component velocity error as a function of image location.

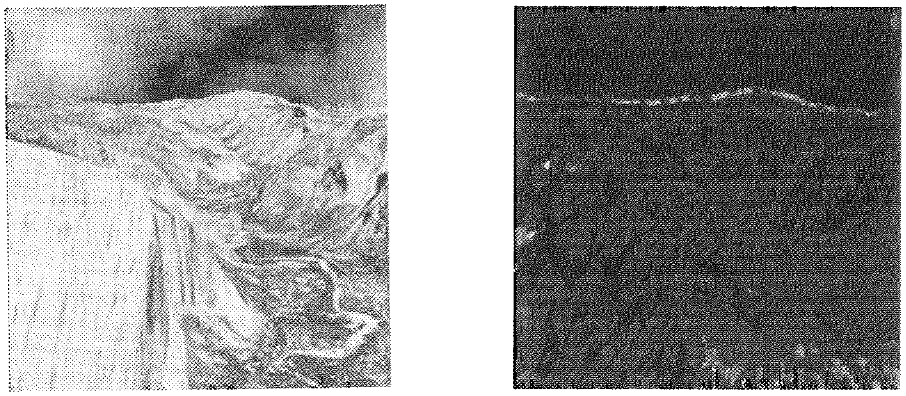

Fig. 12. Yosemite sequence. Velocities in the Yosemite sequence were predominately toward the left with speeds ranging from 0 to 4. The clouds moved (nonrigidly) to the right at 1 pixel/frame. (*left*) One frame from the sequence. (*right*) Average component velocity error as a function of image location.

Along the boundary of the sphere there was a large amount of noise caused by the texture-mapping algorithm. In addition, because of the loss of resolution caused by the filter support, about 9% of the estimates lay outside the projected boundary of the sphere. As a result the extreme errors occur in this region. However, within the boundary of the sphere the accuracy is similar to that in experiments 4 and 5, with 58%, 80%, and 89% of the estimates having errors less than 1, 2, and 3 degrees. Outside the boundary of the sphere, the estimates are still consistent with the motion of the sphere as can be seen from the estimated 2D velocity in section 5. When real images were texture-mapped onto the sphere, instead of the checker-board pattern, the results were similar.

With the Yosemite sequence, only 15 frames were available. Therefore a smaller spatiotemporal window of support (7 pixels-frames) was used with a corresponding decrease in the spatiotemporal wavelength to which the filters were tuned. All other parameter settings and thresholds were identical to those used above. The results are shown in figure 12. Performance with the Yosemite sequence was similar to experiments 4 and 5. Like Heeger's results, and those with the rotating sphere, most of the extreme errors lie on the occlusion boundaries. A few others are due to the cloud movement for which we had no exact velocity information. However, note that most of the sky region was dominated by relatively low spatiotemporal frequencies to which the filters were relatively insensitive. In a more complete implementation (with more than one scale) the cloud motion would be detected. Errors were also due to aliasing and numerical error, as the filters were tuned to the highest end of the frequency spectrum.

Excluding the sky region, about 85% of the image had at least one component velocity estimate, with 60%, 79%, and 87% of the estimates having errors less than 1, 2, and 3 degrees.

4.6 Transparency

Our final experiment in this section addresses the issue of velocity resolution and the motion of transparent surfaces. Two samples of white Gaussian noise (mean zero with $\sigma = 250$) were combined additively. The first covered the entire image and was stationary. The second, masked by the characteristic function shown in figure 13 (top-left), moved with speed 1.5 pixels/frame and direction 31 degrees. This roughly simulates the motion of a textured object viewed through a window on which there is the reflection of a stationary textured surface. Our goal is simply to show that reliable estimates of component velocity can be obtained within a spatial region in which more than one motion exists. Conversely, note that any technique based on purely spatial image properties (e.g., zero-crossings of $\nabla^2 G$) will yield incorrect results.

In order to demonstrate the velocity resolution, the component velocity estimates were divided into three groups according to whether they were consistent with the stationary window, the moving object, or neither. Consistency was defined in terms of the error between the component velocity estimates and the constraint planes of the 2D velocities (24). In particular, each estimate was associated with the constraint plane to which it was closest, unless it was not within 10 degrees of either, in which case it was deemed inconsistent. Estimates with orientations within 5 degrees of the direction of motion were ignored as they would be consistent with both motions.

Figure 13 shows the average (absolute) error ($|\psi_\epsilon|$ in (24)) as a function of image location for the three groups of estimates. As above, where there are no estimates the pixel is black. In comparing the three images, note first that the estimates consistent with the window cover the entire image, while those consistent with the moving object coincide essentially with the characteristic

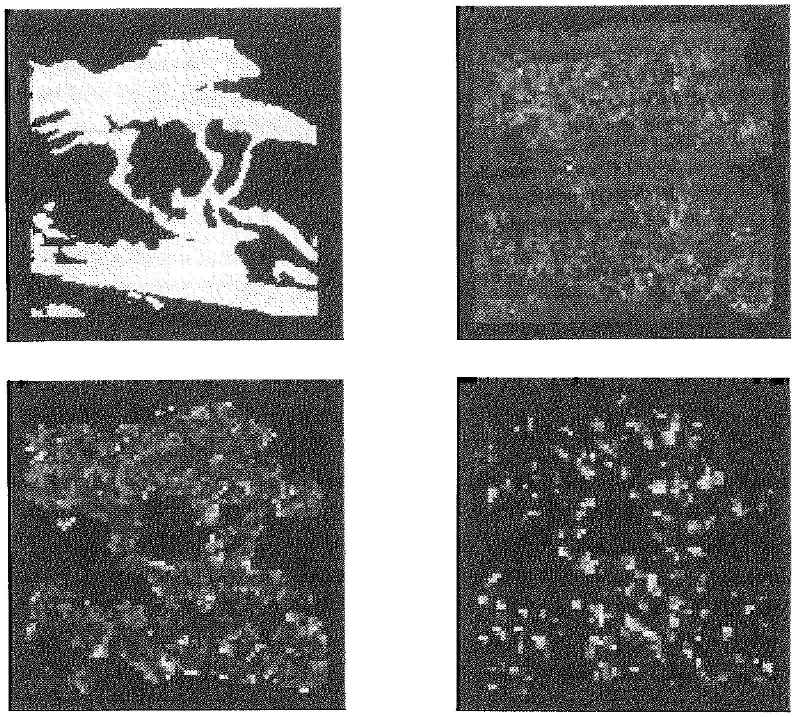

Fig. 13. Transparency. The characteristic function for the moving object (*top-left*). The final three images show average (absolute) component velocity error as a function of image location for the three groups of estimates that are consistent with the stationary window (*top-right*), consistent with the moving object (*bottom-left*), and consistent with neither (*bottom-right*).

function. The mean (absolute) errors per pixel (averaged over pixels with at least one estimate) for those estimates consistent with the window and the object were 0.75 and 1.9 degrees respectively, with standard deviations 0.7 and 1.6. Errors were generally worse in the background where the window and object overlapped. Pixels containing estimates consistent with neither surface are sparsely distributed. The mean number of estimates per pixel (averaged over pixels with at least one estimate) for the three cases were 3.9 ($\sigma = 1.1$) for the stationary window; 2.8 ($\sigma = 1.4$) for the moving tree; and 1.1 ($\sigma = 0.3$) for the inconsistent estimates. Thus, in addition to being sparsely distributed, rarely does more than one inconsistent estimate appear at any one pixel.

5 Computing 2D Velocity

As a further demonstration of the accuracy of the component velocity estimates it was decided to compute estimates of 2D velocity in local patches with a least-squares approach (cf. (Waxman and Wohn 1985)). The derivation of the approach follows from (23), which gives the linear constraint that each component velocity estimate imposes on the local 2D velocity. In addition, we assume that the local 2D velocity field reflects the relative motion of a smooth surface, and may be approximated by

$$\tilde{\mathbf{v}}(\mathbf{x}, t) = (\alpha_0 + \alpha_1 x + \alpha_2 y,\ \beta_0 + \beta_1 x + \beta_2 y) \quad (28)$$

Each collection of local estimates therefore yields a system of linear equations $\mathbf{Ra} = \mathbf{s}$ in the six unknowns $\mathbf{a}^t = (\alpha_0, \alpha_1, \alpha_2, \beta_0, \beta_1, \beta_2)$, where each component velocity estimate $\tilde{v}_n\tilde{\mathbf{n}}$ provides one linear constraint:

$$(\tilde{n}_1,\ \tilde{n}_1 x,\ \tilde{n}_1 y,\ \tilde{n}_2,\ \tilde{n}_2 x,\ \tilde{n}_2 y)\ \mathbf{a} = \tilde{v}_n \quad (29)$$

A least-squares solution (assuming at least 6 local constraints) minimizes $\| \mathbf{Ra} - \mathbf{s} \|^2$, where $\mathbf{s}$ is the vector of normal speeds, $\tilde{v}_n$.

The estimates of component velocity that satisfied the constraints in section 4.1 were collected about each pixel (on the subsampled grid) within a radius of 2 pixels. A singular-value decomposition (SVD) was then used to determine the conditioning of the resultant system (the condition number κ is the ratio of the largest to smallest singular-values of $\mathbf{R}$). Condition numbers greater than 10 were taken to reflect an insufficient amount of local structure from which the 2D velocity could be computed. This restricts the sensitivity of the least-squares solution which is proportional to κ. The restriction on κ may also be viewed in terms of the minimal distribution of component velocities required to compute the 2D velocity estimate. In particular, in just two dimensions, $\kappa < 10$ means that the input must span at least 5 degrees of the constraint plane. We found that with $\kappa \leq 10$ a dense set of 2D velocity estimates was usually obtained; condition numbers between 5 and 10 were common. When there was sufficient information, the least-squares system was solved using the pseudo-inverse provided by the SVD. Finally, we also discarded estimates for which the residual error $\|\mathbf{R}\tilde{\mathbf{a}} - \mathbf{s}\|/\|\mathbf{s}\|$was greater than 0.5.

The estimated 2D velocity $\tilde{\mathbf{v}}$ was then taken to be (α_0, β_0), the constant parameters in (28). The error in the estimated 2D velocities was taken to be the angle between the space-time direction vectors $(\mathbf{v}, 1)$ and $(\tilde{\mathbf{v}}, 1)$:

$$\psi_\epsilon = \arccos\left[\frac{(\mathbf{v},\ 1) \cdot (\tilde{\mathbf{v}},\ 1)}{\sqrt{1 + \|\mathbf{v}\|^2}\ \sqrt{1 + \|\tilde{v}\|^2}}\right] \quad (30)$$

This measure of error is consistent with that used for component velocities, and complements the notion of velocity in terms of space-time orientation. In particular, because the component velocity errors are typically concentrated about the constraint plane (23), and are uniformly distributed over orientation (cf. figure 6), we expect that for reasonably well-conditioned systems, the estimated 2D velocities will be concentrated within a cone about the true velocity $(\mathbf{v}, 1)$ in space-time (see figure 18). The opening angle of the cone depends on the magnitude of errors in component velocity estimates, and the distribution of estimates of the constraint plane. Note that the absolute velocity error ($\|\mathbf{v} - \tilde{\mathbf{v}}\|$) and the relative error ($\| \mathbf{v} - \tilde{\mathbf{v}} \|/\| \mathbf{v} \|$) corresponding to a given angular error (30) depend on speed $\| \mathbf{v} \|$. These relationships are depicted in figure 14.

Figure 15 shows histograms of 2D velocity errors from the 2D velocities computed from experiments 1, 4, and 5. The first three experiments, with predominantly translational velocity fields, produced the most accurate estimates of component velocity, and hence the most accurate 2D velocity estimates. Figure 15 (left) is also characteristic of experiments 2 and 3. The cases of dilation and rotation were not handled quite as well. Despite this, the errors are almost all less than two degrees. Figure 16 shows the estimated 2D velocity fields for experiments 4 and 5. In both cases, the estimated 2D velocities are sufficiently accurate that the vector differences between the true and estimated velocities are not resolvable at this scale. Therefore the true

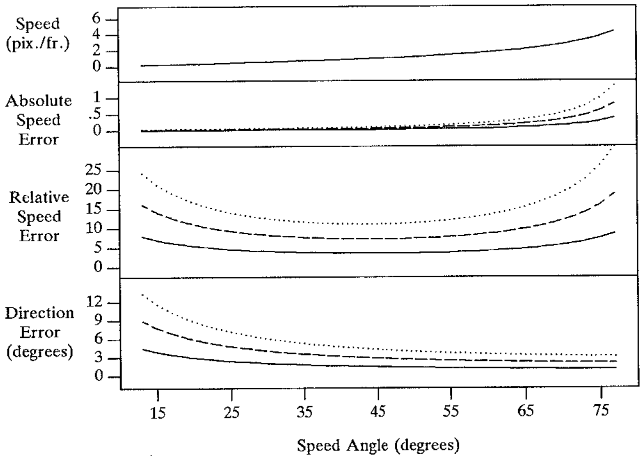

Fig. 14. For fixed angular 2D velocity errors (31) of 1 (solid), 2 (dashed), and 3 (dotted) degrees, this shows the dependence of speed (in pixels/frame), relative and absolute speed errors, and direction errors on angular speed. (In space-time spherical coordinates (ϕ, θ), angular speed and direction correspond to ϕ and θ.) (*top*) Speed in pixels/frame: tan (ϕ). (*middle*) Maximum absolute speed errors (in pixels/frame): tan (ϕ) − tan (ϕ + ψ_ϵ) and relative speed errors 100.0(tan (ϕ) − tan (ϕ + ψ_ϵ))/tan (ϕ), for ψ_ϵ = 1, 2, and 3 degrees. (*bottom*) Maximum error in spatial direction (in degrees): ψ_ϵ/sin (ϕ).

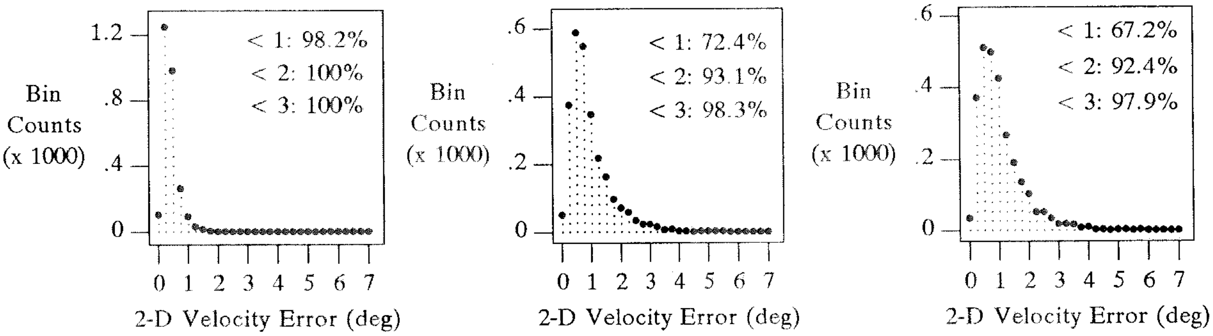

Fig. 15. Histograms of 2D velocity errors for experiments 1, 4, and 5. The corresponding components velocity errors are shown in figures 5–8.

velocities and vector differences are not shown (cf. figure 19). Figure 17 shows the 2D velocity error (in degrees) as a function of spatial location. Note that the errors are concentrated along the boundaries of regions without measurements where the component velocities are not as accurate, and where the least-squares systems were less well conditioned. In particular, note that the errors near the flow singularities are not significantly worse than in other areas. Finally, figure 18 shows histograms of angular differences between the true and estimated velocities in spherical coordinates for experiments 4 and 5. This helps to show that the errors are distributed evenly about the true velocities.

2D velocity fields were also computed for the rotating sphere and the Yosemite sequences. Figures 19 and 20 show the estimated 2D velocities, the true velocity fields, the vector differences between them, and their respective 2D velocity errors as a function of spatial location. As shown in figures 11 and 12, both of these sequences produce estimates that spread across occlusions boundaries. This is also evident in figures 19 and 20. However, it is also clear that these estimates are generally consistent with the corresponding surface motion.

Taking into account only those estimates within the boundary of the sphere, the proportions of 2D estimates with errors less than 1, 2, and 3 degrees were 72%, 88%, and 93%. Over this region the errors are generally uniform. In particular, the flow singularity is handled well. Also note the vertical regions containing no 2D velocity estimates, where, from figure 11 (right),

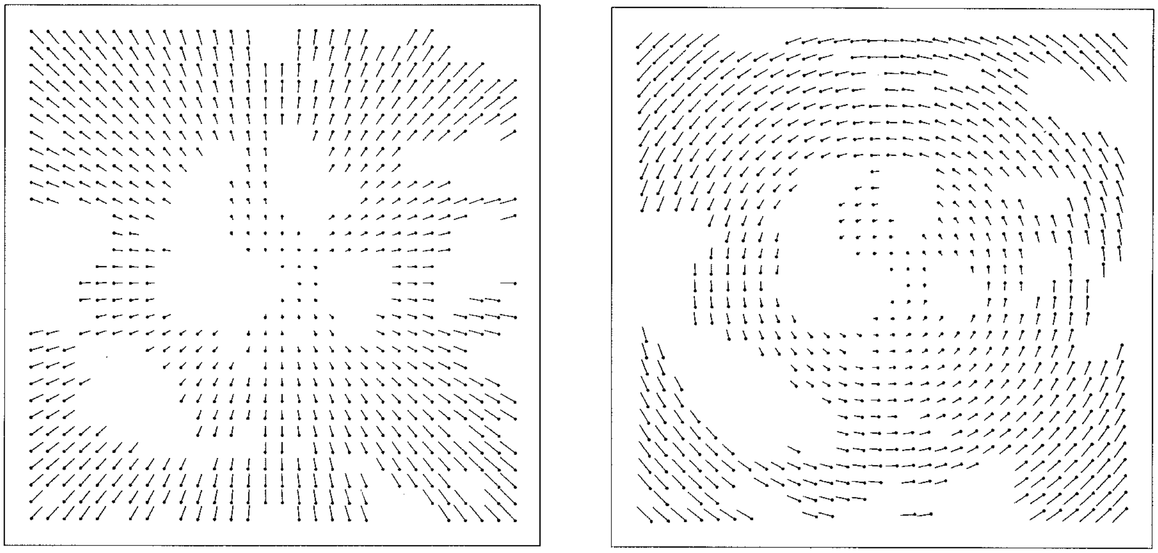

Fig. 16. Computed 2D velocity fields from experiments 4 (*left*) and 5 (*right*).

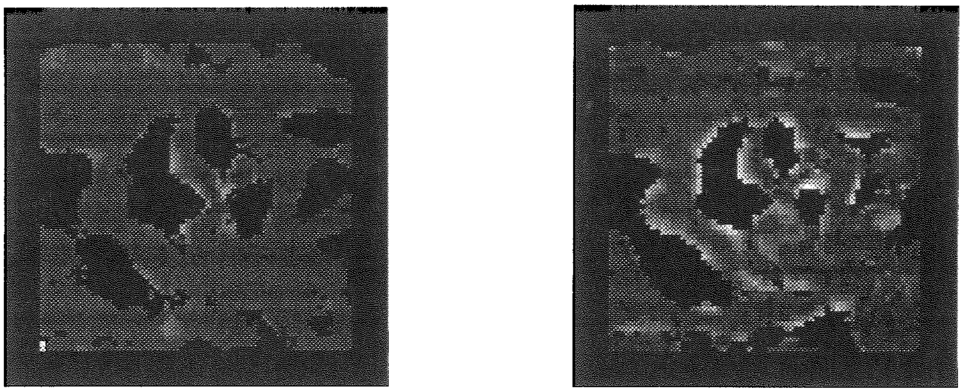

Fig. 17. 2D velocity errors for experiments 4 (*left*) and 5 (*right*) are shown as a function of image location.

it is clear that there were component estimates. Interestingly, these regions fall precisely down the centers of the wide rectangular checkers (cf. figure 11 (left)). Relative to the tuning of the filters to high spatiotemporal frequencies, and the small radius within which the component velocity estimates were combined to estimate 2D velocity, the structure in these regions is essentially one dimensional. About the equator, however, the checkers are somewhat foreshortened and the image speeds are faster so that lower spatial frequencies will stimulate those filters tuned to faster speeds. Hence there is a greater density of significant filter activity and therefore sufficient structure for the estimation of 2D velocity. The regions currently without 2D estimates would be filled in by filters tuned to lower spatiotemporal frequencies in a more complete implementation (with more than one scale).

For the Yosemite sequence, if we neglect errors just above the horizon in the sky region, then the proportions of 2D velocity estimates with errors less than 1, 2, and 3 degrees are 45%, 71%, and 82%. Although these results are not as good as those above, most of the poor estimates of 2D velocity coincide with a poorly conditioned system or a high residual error. For example, when we discarded all estimates with condition numbers greater than 5 (instead of 10), or residual errors greater than 0.1 (instead of 0.5), the proportions of errors below 1, 2, and 3 degrees increased to 63%,

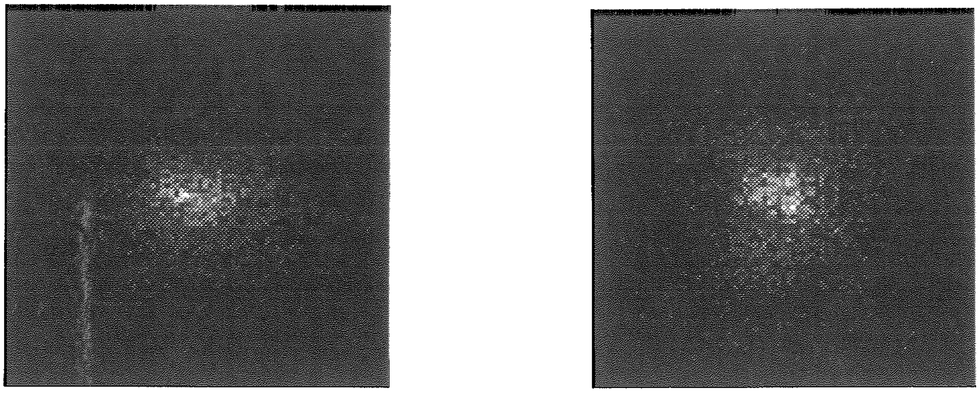

Fig. 18. Histograms of 2D velocity errors in spherical coordinates for experiments 4 and 5. The horizontal and vertical axes correspond to error in orientation and speed. If the true and estimated 2D velocities are given by (θ, ϕ), and $(\tilde{\theta}, \tilde{\phi})$, then we increment location $((\theta - \tilde{\theta}) \sin \phi, \phi - \tilde{\phi})$. Both histograms have widths of 4 degrees.

89%, and 95% while the total number of estimates dropped by about one third. This is important as it means that the errors in the 2D velocity estimates were due mainly to the conditioning of the least-squares system, and not inaccuracy in the component velocity estimates (which is our principal concern). Finally, note that these results compare favorably with those obtained with Heeger's model, for which the histogram of errors was relatively flat with about 90% of the estimates having errors less than 25 degrees. The proportions of estimates with errors less than 5, 10, and 15 degrees were only 30%, 60%, and 80%. However, note that Heeger used different spatiotemporal scales and larger spatial support which leads to different results. In particular, it gives results in areas (e.g., the sky and the lower right) where the present method does not.

6 Hamburg Taxi Sequence

Finally, we report results obtained from the *Hamburg Taxi Sequence*, which has been used extensively by Nagel and Enkelmann (Enkelmann 1986; Nagel 1983; Nagel and Enkelmann 1986). Unfortunately, the actual motion field is unknown so that the results can only be evaluated qualitatively. We used the same filters and parameters as in all but the Yosemite sequence above. There are 46 frames of the sequence. Velocity was computed at frame 21, which is shown in figure 21 (left). There are four moving objects in the scene: the taxi, which has speeds of just under one pixel/frame; the Golf in the lower left, which has speeds of about 3.75 pixels/frame; the van in the lower right, which is partially occluded and exhibits speeds similar to the Golf; and a pedestrian in the upper left, which moves down to the left at about 0.3 pixels/frame. The branches of the two trees are also moving slowly.

Figure 21 (right) shows where (in the image) estimates of component velocities were obtained. Black areas denote regions within which no estimates occur. The darker grey areas denote regions in which all estimates had normal speeds between 0 and 0.015 pixels/frame. The brighter areas show regions in which there existed estimates with normal speeds greater than 0.15 pixels/frame. Estimates arising from the four main moving objects are clear.

Figure 22 shows the 2D velocities that were computed from the component estimates. In particular, figure 22 (top) shows the estimated speed (shown as intensity) as a function of image location. The vector fields corresponding to the four boxed areas are then shown below (blown up so that the individual vectors are resolvable). The black dots not joined to vectors represent speeds close to 0. Note that not all regions with component velocity measurements yielded 2D estimates due to the local nature of the computation. This is particularly evident along the rear windows of the taxi cab. Also, from figures 21 and 22, note the large number of estimates in low-contrast regions (e.g., the street marking to the right of the taxi). The robustness of local phase behavior as compared to amplitude is especially clear in areas of low contrast.

Finally, note that no smoothing has been applied to these measurements. This is important in comparing the results to other techniques that impose smoothness constraints to the raw measurements (Nagel and Enkelmann

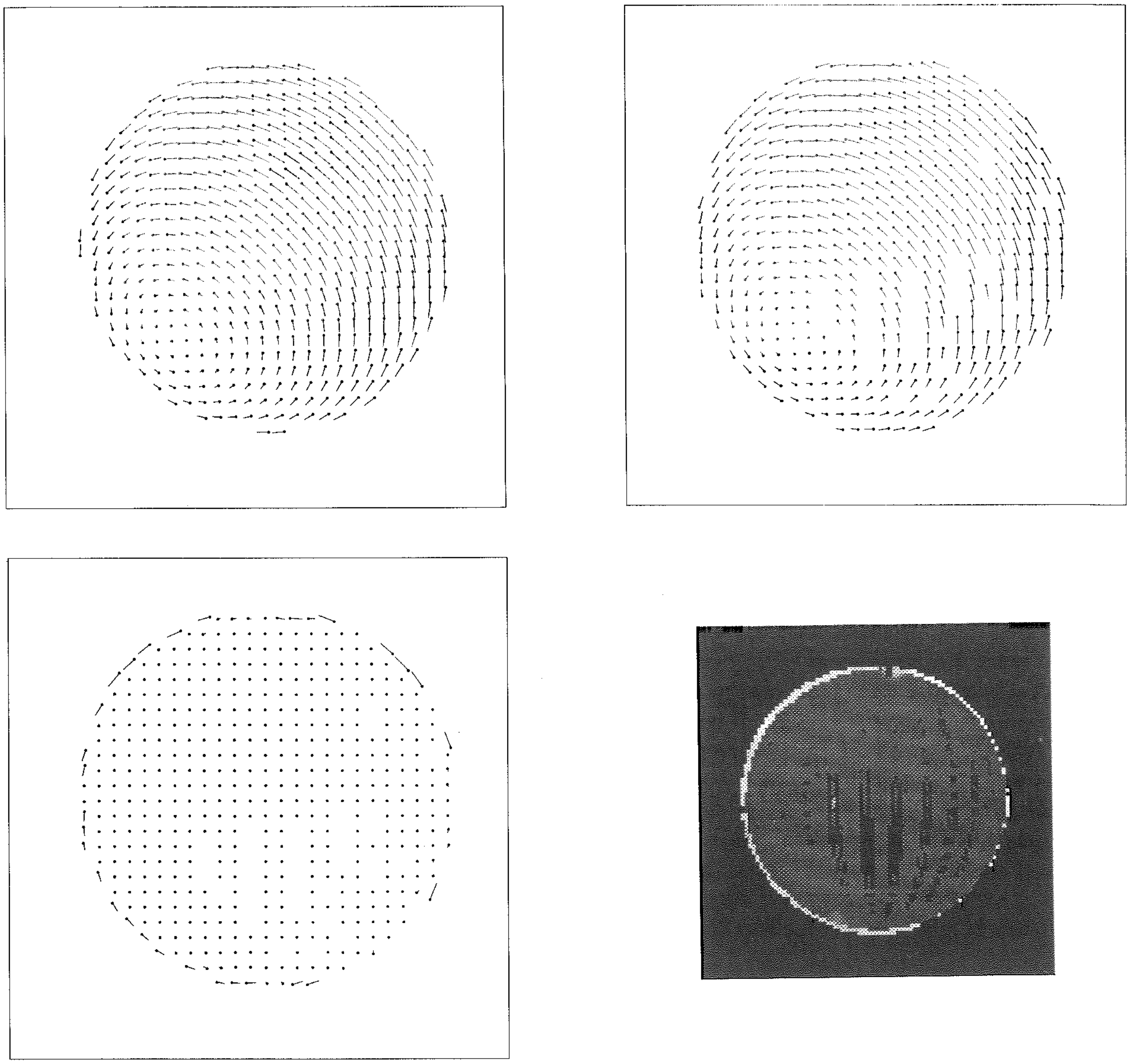

Fig. 19. (*top-left*) True (induced) 2D velocity field from the rotating sphere. (*top-right*) Estimated 2D velocities. (*bottom-left*) The vector difference between the two fields. (*bottom-right*) Velocity error as a function of image location.

1986). Similarly, it is important to remember that our objective in computing 2D velocity from the component velocity estimates was to illustrate the accuracy of the component velocity estimates. However, as mentioned in the introduction, the integration of local measurements implicitly assumes that they arise from the same physical object. Here, a unique 2D velocity arising from a single object is assumed within each local neighborhood. The velocity estimates near the front of the van in figure 22 (bottom-right) show that this is, in general, inappropriate. In this case, measurements from a tree branch and the van are combined, and yield a good fit to 2D velocity (with low residual error). Subsequent smoothing of the velocity estimates would aggravate the problem. This result supports our approach of considering the inference of 2D velocity from component velocities as an interpretation issue that is distinct from measurement. An appropriate framework for performing this

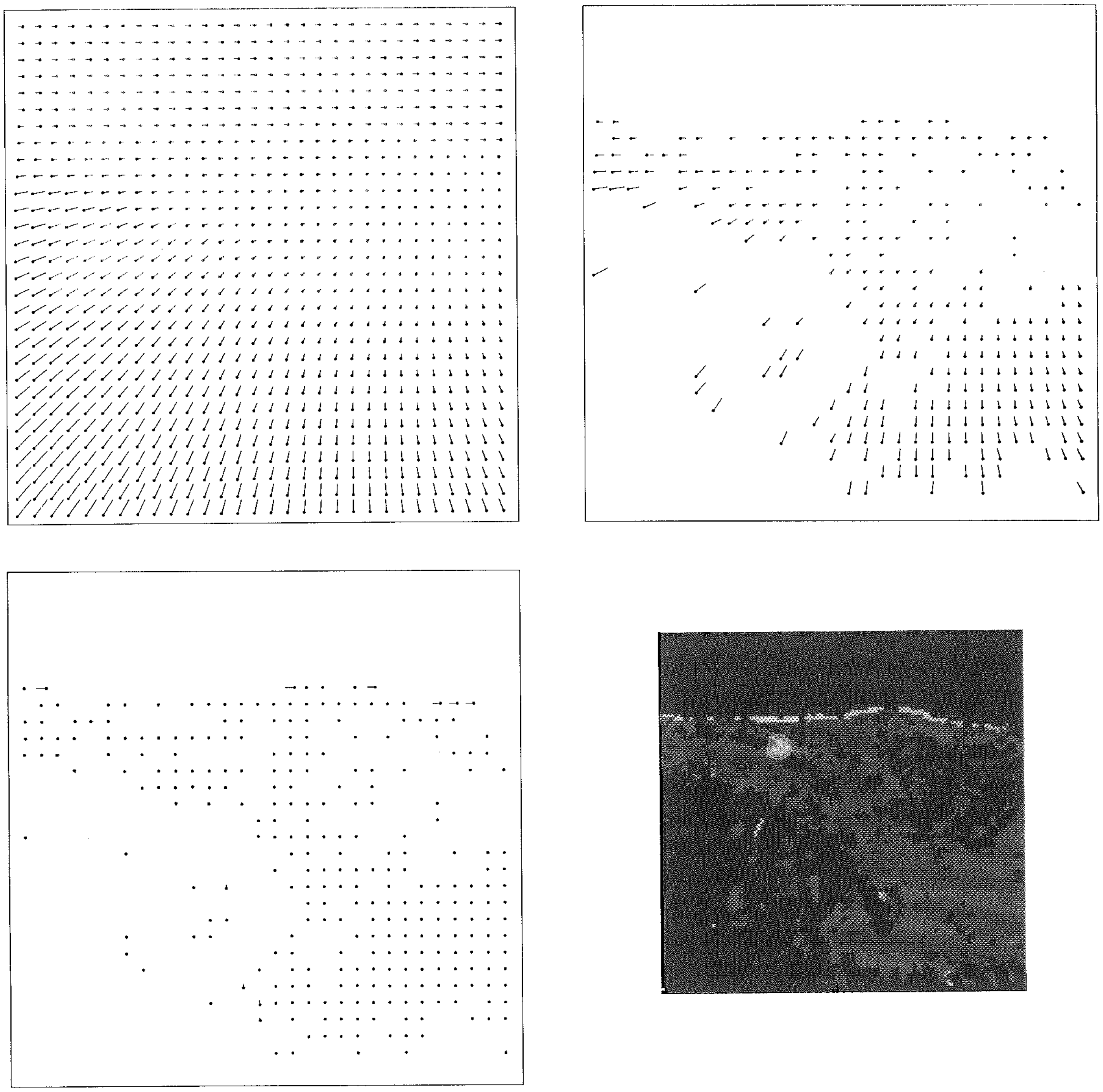

Fig. 20. (*top-left*) True (induced) 2D velocity field from the Yosemite sequence. (*top-right*) Estimated 2D velocities. (*bottom-left*) The vector difference between the two fields. (*bottom-right*) Velocity error as a function of image location.

interpretation in the face of multiple moving objects is an important area for future study.

7 Summary and Discussion

Component velocity was defined in terms of the gradient of the phase output of individual velocity-tuned linear filters. The approach involves two main stages of processing:

- The time-varying image is first represented with a family of constant-phase velocity-tuned filters.
- The local phase gradient is then measured from the output of the individual filter types to obtain estimates of component velocity.

It was argued that this use of phase information yields accurate and robust estimates of component velocity. In particular, local phase information was shown to be more robust than amplitude under variations in lighting

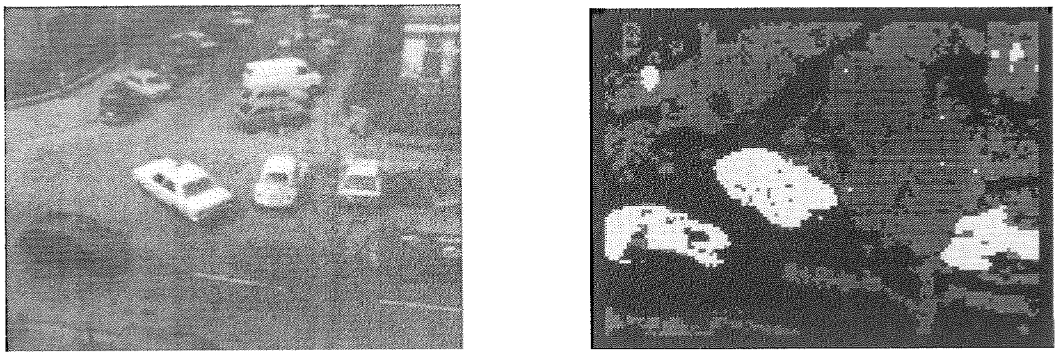

Fig. 21. Hamburg taxi sequence. (*left*) Frame 21. (*right*) Regions with no component velocity estimates are black. Grey and white regions correspond to regions containing component velocity estimates with normal speeds between 0 and 0.15 pixels/frame and greater than 0.15 pixels/frame respectively.

conditions, relative surface orientation, as well as under changes in local orientation, wavelength, and speed caused by geometric deformation in space-time. As a consequence, the assumptions of constant amplitude and the translation of filter outputs are relaxed. It was also argued that the use of phase behavior may be viewed as a generalization of the use of zero-crossing contours, and leads to a denser set of velocity estimates. Finally, it was shown that the expression of component velocity in terms of local phase behavior is consistent with that in terms of spatiotemporal frequency.

The technique's accuracy, robustness, and localization in space-time were demonstrated through a series of experiments involving image sequences of textured surfaces moving in 3D under perspective projection. Many of the cases considered involved sizable time-varying perspective deformation. The width of support (in all but the Yosemite sequence) was limited to 5 pixels (frames) in space (time, respectively) at one standard deviation. In most experiments with reasonably large amounts of dilation, rotation, and shear, we find that approximately 65–80% of the errors are less than 1 degree, 80–90% are within 2 degrees, and the proportion within 3 degrees is generally greater than 90%. In cases dominated by image translation (section 4.2.2) the estimates are even more accurate, often with 90% of all estimates having errors less than 1 degree. As shown in section 5 this accuracy yields local estimates of 2D velocity that are also very accurate, with most estimates within 2 degrees of the correct 2D velocity. As shown in figure 14, a speed error of 2 degrees amounts to relative errors of 6–10%. This subpixel accuracy compares favorably with the approaches of Heeger (1988) (as discussed in section 5); Duncan and Chou (1988), who report relative errors of 20% on realistic images; Little et al. (1988), whose technique is limited to integer pixel velocities per frame; and Waxman et al. (1988), who claim relative errors of 10%. Finally, the results reported in section 6 with the Hamburg Taxi Sequence are accurate compared with those previously reported (Enkelmann 1986; Nagel 1983; Nagel and Enkelmann 1986).

Other important properties of the approach are as follows: First, it is image-independent in that specific features or tokens are not a prerequisite. As a consequence, problems associated with their detection, localization, descriptive richness, and matching are avoided. For example, with zero-crossings there remain questions concerning the robustness of detection and localization (Jenkin and Jepson 1988). Moreover, as noted by Waxman et al. (1988) and Duncan and Chou (1988), there are also problems in areas of high edge density. Second, the present approach also differs from most others in that, because of the initial representation, the image structure is separated to some degree based on velocity and scale, so that multiple velocity estimates are allowed in local neighborhoods. This may be useful in the case of transparency, or partial occlusion. Third, the resultant computational scheme is efficient, and suitable to parallel processing. Each velocity-tuned channel may be handled independently, and the various stages of processing are predominantly local, linear, shift-invariant, and separable in space-time. Finally, the questions of smoothing and filling-in of regions with no measurements are postponed as they are considered distinct issues of interpretation.

With respect to the initial filters we note the following properties: (1) The technique is not strictly limited to

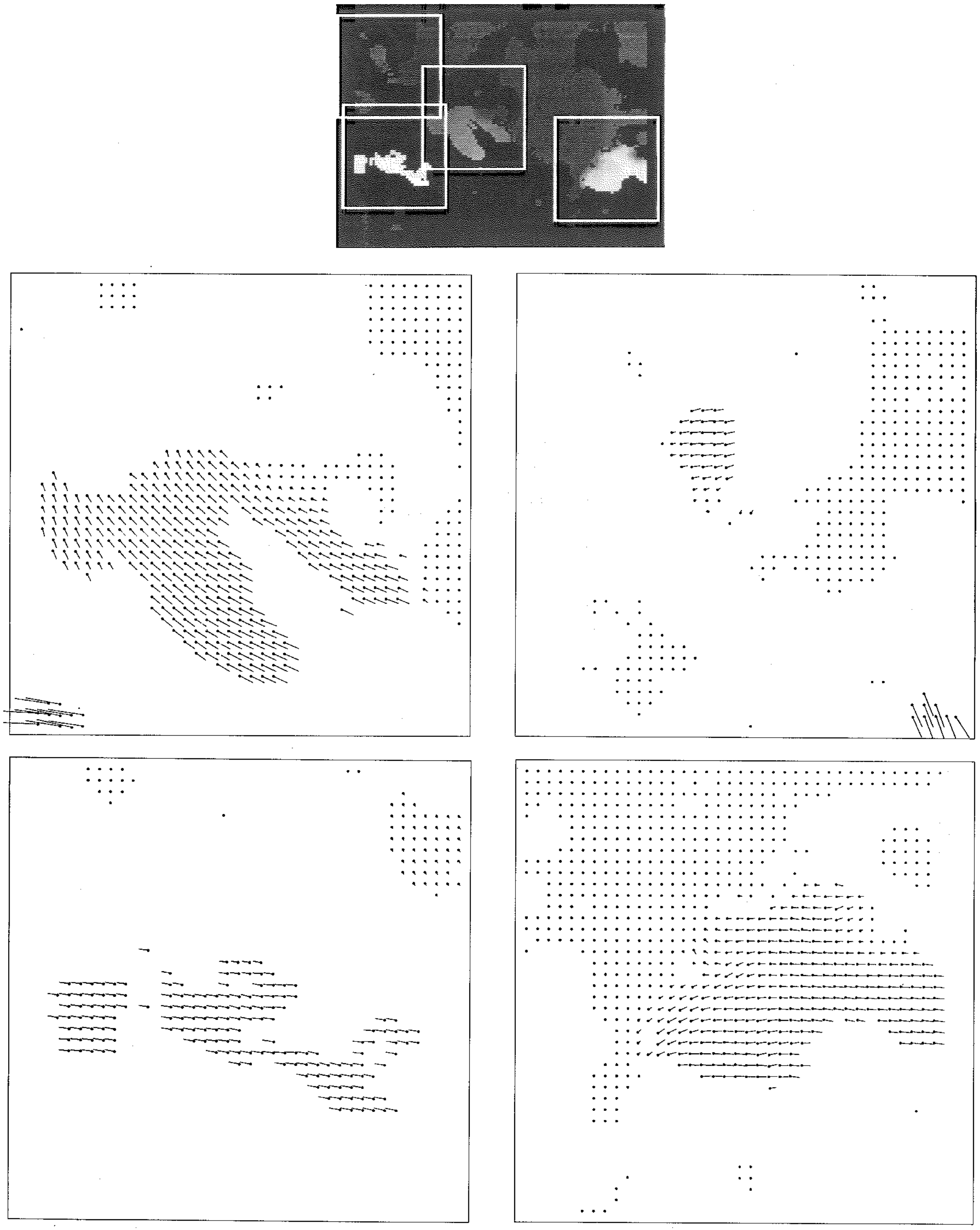

Fig. 22. Hamburg taxi sequence. (*top*) 2D speed (as intensity) is shown as a function of image location. Regions without estimates of 2D velocity are black. The four white rectangular boxes delineate the regions within which 2D velocities are shown in the vector fields below. Each region has been blown up with the speeds scaled to avoid too much overlap of the individual vectors (for better visibility). (*middle*) Velocity estimates about the taxi and the pedestrian with speeds scaled to 0–1.0 pixels/frame (taxi), and 0–0.3 pixels/frame (pedestrian). (*bottom*) Velocity estimates about the Golf and the van with speeds scaled to 0–3.5 pixels/frame in both cases.

the use of Gabor kernels. Other filters can be used, such as inseparable kernels with nonunit aspect ratios, as long as they occur in quadrature pairs and exhibit constant phase properties. (2) A relatively small bandwidth is important because it reduces sensitivity to mean illumination and low frequencies. (3) The initial representation based on the filter outputs is efficient because it is subsampled at a reasonable rate and quantized. It is hoped that with better forms of interpolation even lower sampling rates can be tolerated with similar or better accuracy.

Acknowledgments

Comments from David Heeger, Evangelos Milios. Michael Mohnhaupt, and Whitman Richards have helped to improve the clarity of the paper. We also thank David Heeger for providing his results and the *Yosemite* image sequence (created by Lynn Quam at SRI International). Part of this work was completed while the first author was visiting the Cognitive Systems Group at the University of Hamburg. Financial support from NSERC, ITRC, the University of Toronto, and the University of Hamburg is appreciated.

References

E.H. Adelson and J.R. Bergen, "Spatiotemporal energy models for the perception of motion," *J. Opt. Soc. Amer. A* 2: 284–299, 1985.

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

P. Anandan, "A computational framework and an algorithm for the measurement of visual motion," *Intern. J. Comput. Vision* 2: 283–310, 1989.

J.L. Barron, "Computing motion and structure from time-varying image velocity information." Ph.D. thesis, Computer Science Dept., University of Toronto; available as TR: RBCV-TR-88-24), 1988.

R.N. Bracewell, *The Fourier Transform and Its Applications*. McGraw-Hill: New York, 1978.

P.J. Burt and E.H. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Trans. Commun.* 31: 532–540, 1983.

P.J. Burt, C. Yen, and X. Xu, "Multiresolution flow-through motion analysis," *Proc. IEEE Conf. Comput. Vision Pattern Recog.* Washington, pp. 246–252, 1983.

B.F. Buxton and H. Buxton, "Computation of optic flow from the motion of edge features in image sequences," *Image Vision Comput.* 2: 59–75, 1984.

J.G. Daugman, "Pattern and motion vision without Laplacian zero crossings," *J. Opt. Soc. Amer. A* 5: 1142–1148, 1987.

J.G. Daugman, "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," *J. Opt. Soc. Amer. A* 2: 1160–1169, 1985.

D.E. Dudgeon and R.M. Mersereau, *Multidimensional Digital Signal Processing*. Prentice-Hall: Englewood Cliffs, NJ, 1984.

J.H. Duncan and T.C. Chou, "Temporal edges: The detection of motion and the computation of optical flow," *Proc. 2nd Intern. Conf. Comput. Vision*, pp. 374–382, Tampa, 1988.

W. Enkelmann, "Investigations of multigrid algorithms for the estimation of optical flow fields in image sequences," *Proc. IEEE Workshop on Motion*, pp. 81–87, Charleston, 1986.

D.J. Fleet, "Measurement of image velocity," Ph.D. dissertation, Dept. of Computer Science, University of Toronto, 1990.

D.J. Fleet and A.D. Jepson, "A cascaded filter approach to the construction of velocity selective mechanisms," Technical Report: RBCV-TR-84-6, Dept. Computer Science, Univ. of Toronto, 1984.

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

D. Gabor, "Theory of communication," *J. IEEE* 93: 429–457, 1946.

L.W. Gardenhire, "Selecting sampling rates," *Proc. 19th Instrument Soc. Amer. Conf.*, July 1964.

F. Glazer, "Hierarchical gradient-based motion detection," *Proc. DARPA Image Understanding Workshop*, pp. 733–748, Los Angeles, 1987.

D.J. Heeger, "A model for the extraction of image flow," *J. Opt. Soc. Amer. A* 4: 1455–1471, 1987.

D.J. Heeger, "Optical flow using spatiotemporal filters," *Intern. J. Comput. Vision* 1: 279–302, 1988.

B.K.P. Horn and B.G. Schunck, "Determining optic flow," *Artificial Intelligence* 17: 185–204, 1981.

H.S. Hou and H.C. Andrews, "Cubic splines for image interpolation and digital filtering," *IEEE Trans. Acoustics, Speech, and Signal Process.* 26: 508–517, 1978.

M. Jenkin and A.D. Jepson, "The measurement of binocular disparity." In Z. Pylyshyn (ed.), *Computational Processes in Human Vision*. Ablex Publishing: Norwood, NJ, 1988.

A.D. Jepson, "Discrete scale-space, multi-scale image representation, and interpolation," in preparation, 1989.

J.J. Koenderink and A.J. van Doorn, "Local structure of movement parallax of the plane," *J. Opt. Soc. Amer.* 66: 717–723, 1976.

J.J. Little, H.H. Bulthoff, and T. Poggio, "Parallel optical flow using local voting," *Proc. 2nd Intern. Conf. Comput. Vision*, pp. 454–459, Tampa, 1988.

H.C. Longuet-Higgins and K. Prazdny, "The interpretation of a moving retinal image," *Proc. Roy. Soc. London* B 208: 385–397, 1980.

D. Marr and S. Ullman, "Directional selectivity and its use in early visual processing," *Proc. Roy. Soc. London* B 211: 151–180, 1981.

J. Mayhew and J. Frisby, "Computational studies toward a theory of human stereopsis," *Artificial Intelligence* 17: 340–385, 1981.

H.H. Nagel, "Displacement vectors derived from second-order intensity variations in image sequences," *Comput. Vision Graph. Image Process.* 21: 85–117, 1983.

H.H. Nagel and W. Enkelmann, "An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences," *IEEE Trans. PAMI* 8: 565–593, 1986.

A.N. Netravali and J.O. Limb, "Picture coding: A review," *Proc. IEEE* 68: 366–406, 1980.

A.V. Oppenheim and R.W. Shafer, *Digital Signal Processing*. Prentice-Hall: Englewood Cliffs, NJ, 1975.

T. Sanger, "Stereo disparity computation using Gabor filters," *Biol. Cybern.* 59: 405–418, 1988.

J.P.H. van Santen and G. Sperling, "Elaborated Reichardt detectors," *J. Opt. Soc. Amer. A* 2: 300–321, 1985.

R.W. Schafer and L.R. Rabiner, "A digital signal approach to interpolation," *Proc. IEEE* 61: 692–702, 1973.

D. Slepian, "Some comments on Fourier analysis, uncertainty and modelling," *Siam Review* 25: 379–393, 1983.

K.M. Ty and A.N. Venetsanopoulos, "Sampling non-bandlimited signals," *Proc. Telecon '84*, Halkidiki, Greece, 1984.

A.B. Watson and A.J. Ahumada, "Model of human visual-motion sensing," *J. Opt. Soc. Amer. A* 2: 322–342, 1985.

A.M. Waxman and K. Wohn, "Contour evolution, neighbourhood deformation and global image flow: Planar surfaces in motion," *Intern. J. Robotics Res.* 4: 95–108, 1985.

A.M. Waxman, J. Wu, and F. Bergholm, "Convected activation profiles: Receptive fields for real-time measurement of short-range visual motion," *Proc. IEEE Conf. Comput. Vision Pattern Recog.* pp. 717–723, Ann Arbor, 1988.

G.B. Whitham, *Linear and Nonlinear Waves*. Wiley: NY, 1974.

Appendix: Computation of Phase Gradient

From equation (17), the phase gradient is computed straight-forwardly in terms of the filter output $R(\mathbf{x}, t)$ and its gradient $\nabla R(\mathbf{x}, t)$. Here we assume that the filter output has been subsampled using space-time sampling distances $\mathbf{S} = (S_1, S_2, S_3)$. Let the nodes of the sampling lattice (i.e., the sampling locations) be given by $S[\mathbf{m}] = \Sigma_{j=1}^{3} m_j S_j e_j$ where $\mathbf{m} \in Z^3$, and the vectors $\mathbf{e}_j$ are standard basis vectors for R^3 (columns of the identity matrix). Finally, for notational convenience, throughout the appendix let the space-time variables $(\mathbf{x}, t)$ be denoted by $\mathbf{x} = (x_1, x_2, x_3)$, and their respective Fourier variables $(\mathbf{k}, \omega)$ by $\mathbf{k} = (k_1, k_2, k_3)$. Our goal is to derive estimates of $R(\mathbf{x}, t)$ and $\nabla R(\mathbf{x}, t)$ from the sub-sampled filter output.

It is convenient to view the bandpass filter response as

$$R(\mathbf{x}) = M(\mathbf{x})C(\mathbf{x}) \quad \text{where} \quad C(\mathbf{x}) = e^{i\mathbf{x}\cdot\mathbf{k}_0}, \qquad (31)$$

and $\mathbf{k}_0$ is the peak tuning frequency of the filter in question. From (31), $\nabla R(\mathbf{x})$ has the form

$$\begin{aligned} \nabla R(\mathbf{x}) &= \nabla M(\mathbf{x})C(\mathbf{x}) + M(\mathbf{x})\nabla C(\mathbf{x}) \\ &= \nabla M(\mathbf{x})C(\mathbf{x}) + i\mathbf{k}_0 R(\mathbf{x}) \qquad (32) \end{aligned}$$

Because $M(\mathbf{x})$ is a lowpass signal it may be interpolated and differentiated from a subsampled representation using standard methods (Dudgeon and Mersereau 1984). This is treated below as a precursor to the explicit interpolation and differentiation of $R(\mathbf{x})$.

A subsampled encoding of $M(\mathbf{x})$, that is, $M(S[\mathbf{m}]) = R(S[\mathbf{m}])C(-S[\mathbf{m}])$, can be interpolated as

$$\tilde{M}(\mathbf{x}) = \sum_{m_1} \sum_{m_2} \sum_{m_3} M(S[\mathbf{m}])\, Q(\mathbf{x} - S[\mathbf{m}]) \qquad (33)$$

where $Q(\mathbf{x})$ is an appropriate interpolation kernel. The derivatives of $\tilde{M}(\mathbf{x})$ have the same form:

$$\tilde{M}_{x_j}(\mathbf{x}) = \sum_{m_1} \sum_{m_2} \sum_{m_3} M(S[\mathbf{m}])\, Q_{x_j}(\mathbf{x} - S[\mathbf{m}]) \qquad (34)$$

If $M(\mathbf{x})$ was strictly lowpass, then the appropriate interpolant $Q(\mathbf{x})$ would be a (separable) product of three sinc functions (Dudgeon and Mersereau 1984). Both (33) and (34) can therefore be viewed as cascades of three 1D convolutions. If $\nabla\phi(\mathbf{x})$ is computed only at nodes of the sampling lattice, then $\tilde{M}(\mathbf{x})$ is given directly by the subsampled representation (without interpolation), and (34) reduces to a single 1D convolution with the differentiated interpolant.

In order to obtain expressions for $\tilde{R}(\mathbf{x})$ and $\nabla\tilde{R}(\mathbf{x})$ in terms of explicit interpolation and differentiation of the subsampled filter output $R(S[\mathbf{m}])$, we can replace $M(S[\mathbf{m}])$ in (33) and (34) with $R(S[\mathbf{m}])C(-S[\mathbf{m}])$. For instance, (34) becomes

$$\begin{aligned} \tilde{M}_{x_j}(\mathbf{x}) = C(-\mathbf{x}) \sum_{m_1} \sum_{m_2} \sum_{m_3} \{&R(S[\mathbf{m}]) \\ &\times C(\mathbf{x} - S[\mathbf{m}])\, Q_{x_j}(\mathbf{x} - S[\mathbf{m}])\} \qquad (35) \end{aligned}$$

Therefore, instead of demodulating the filter output before subsampling, we simply modulate the appropriate interpolation/differentiation kernel. After substitution into (31) and (32) we now have expressions for $R(\mathbf{x}, t)$ and $\nabla R(\mathbf{x}, t)$ in terms of the subsampled filter output.

In the experiments reported in section 4, $\nabla\phi(\mathbf{x})$ was computed only at the nodes of the sampling lattice (i.e., with $\mathbf{x} = S[\mathbf{m}]$ for $\mathbf{m} \in Z^3$). In this case, $R(\mathbf{x})$ is given explicitly at the sampling points, and its derivative, $\tilde{R}_{x_j}(\mathbf{x}) = \tilde{M}_{x_j}(\mathbf{x})C(\mathbf{x}) + i(\mathbf{e}_j \cdot \mathbf{k}_0)R(\mathbf{x})$, reduces to

$$\begin{aligned} \tilde{R}_{x_j}(\mathbf{x}) &= C(\mathbf{x}) \sum_{n} M(\mathbf{x} - n\mathbf{e}_j)h(n) \\ &\quad + i(\mathbf{e}_j \cdot \mathbf{k}_0)R(\mathbf{x}) \qquad (36) \\ &= C(\mathbf{x})C(-\mathbf{x}) \sum_{n} R(\mathbf{x} - n\mathbf{e}_j)e^{in(\mathbf{e}_j\cdot\mathbf{k}_0)}h(n) \\ &\quad + i(\mathbf{e}_j \cdot \mathbf{k}_0)R(\mathbf{x}) \qquad (37) \end{aligned}$$

$$= \sum_{n} R(\mathbf{x} - n\mathbf{e}_j)H(n) + i(\mathbf{e}_j \cdot \mathbf{k}_0)R(\mathbf{x}) \qquad (38)$$

where $h(n)$ is an appropriate kernel for numerical differentiation of a low-pass signal, and $H(x) = h(x)c(x)$ is the new kernel that is to be applied directly to $R(S[\mathbf{m}])$.

For appropriately bandlimited signals, the interpolation error decreases as the spatiotemporal extent of the interpolating kernel increases. Toward efficiency and localization in space-time, it is desirable to limit their extent. Rather than use a truncated sinc function, more accurate interpolants can be found either in low-order polynomials, in splines, or through optimization techniques (e.g., (Hou and Andrews 1978; Oppenheim and Schafer 1975; Shafer and Rabiner 1973)). The choice of interpolant is important because it affects the choice of subsampling rate for the Gabor outputs. In general, the appropriate subsampling rate depends on several factors including (1) the input spectral density, (2) the form of interpolation, and (3) a tolerance bound on reconstruction error. For 1D signals with Gaussian power spectra, interpolation with local polynomial interpolants (e.g., 4 or 5 points) is generally possible to within 5% RMS error if the sampling rate is one complex sample every σ (following Gardenhire (1964) and Ty and Vanetsanopoulos (1984)). As mentioned in section 2.4 we adopted this rate. However, because we also consider the measurement of derivatives, this is not an overly generous rate. On the other hand, because a higher sampling rate means a significant increase in computational expense, it is clear that the relationship between sampling rates and appropriate form of interpolation/differentiation deserves further attention (although this is beyond the current paper).

With respect to section 4, we base the numerical differentiation on a standard 4-pt central-difference formula with coefficients $h(n) = (1/12s)\,(-1, 8, 0, -8, 1)$, where s is the sampling distance. The corresponding kernel that was applied to $R(S[\mathbf{m}])$ to find $\tilde{R}_{x_j}(\mathbf{x})$ is therefore given by $H(n) = (1/12s)\,(-e^{-i2sk_j}, 8e^{-isk_j}, 0, -8e^{isk_j}, e^{i2sk_j})$, where $k_j = \mathbf{e}_j \cdot \mathbf{k}_0$.