International Journal of Computer Vision, 279-302 (1988)

Optical Flow Using Spatiotemporal Filters

DAVID J. HEEGER
Vision Sciences Group, Media Laboratory, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139

Abstract

A model is presented, consonant with current views regarding the neurophysiology and psychophysics of motion perception, that combines the outputs of a set of spatiotemporal motion-energy filters to estimate image velocity. A parallel implementation computes a distributed representation of image velocity. A measure of image-flow uncertainty is formulated; preliminary results indicate that this uncertainty measure may be used to recognize ambiguity due to the aperture problem. The model appears to deal with the aperture problem as well as the human visual system since it extracts the correct velocity for some patterns that have large differences in contrast at different spatial orientations.

1 Introduction

The world we live in is constantly in motion—an observer (either a biological organism or a computer being) who depends on visual perception to gain an understanding of his environment must be able to interpret visual motion. Some of the important functions of motion perception are: (1) to act as an early warning system; (2) to allow an observer to track the location of moving objects and recover their three-dimensional structure; (3) to help an observer determine his own movement (egomotion) through the environment; (4) to help an observer divide the visual field into meaningful segments (e.g., moving vs. stationary or rigid vs. nonrigid).

The perception of visual motion does not depend on prior interpretation or recognition of shape and form. However, it does depend on there being motion information, i.e., changes in intensity over time throughout the visual field. Without texture, a perfectly smooth moving surface yields an image sequence in which most local regions do not change over time. But in a highly textured world (e.g., natural outdoor scenes with trees and grass), there is motion information throughout the visual field.

It is generally believed that the analysis of visual motion proceeds in two stages. The first stage is the extraction of two-dimensional motion information (direction of motion, speed, displacement) from image sequences. The second stage is the interpretation of image motion. Optical flow, a two-dimensional velocity vector for each small region of the visual field, is one representation of image motion. This paper addresses the issue of extracting a velocity vector for each region of the visual field by taking advantage of the abundance of motion information in a highly textured image sequenced.

Most machine vision efforts that try to extract image flow employ just two frames from an image sequence; either matching features from one frame to the next [1] or computing the change in intensity between successive frames along the image gradient direction [2,3]. In a highly textured world neither of these approaches seems appropriate, since there may be too many features for matching to be successful and the image gradient direction may vary randomly from point to point. In fact, an error analysis of gradient-based methods [3] confirms that a major problem with the approach is that large errors are made where the image is highly textured, precisely where there is the greatest amount of motion information!

There have recently been several approaches to motion measurement based on spatiotemporal

filters [4,5,6,7,8,9,10,11] that utilize a large number of frames sampled closely together in time. These papers describe families of motion-sensitive mechanisms each of which is selective for motion in different directions. To be able to use such mechanisms in computing optical flow, one must overcome two obstacles: (1) the aperture problem; (2) the fact that the filter outputs do not depend solely on the velocity of a stimulus, but rather on its spatial frequency, temporal frequency, and contrast.

In the next section, I review the mathematics of motion in the spatiotemporal-frequency domain. A family of motion-sensitive Gabor filters are described in section 3. Section 4 derives a model for extracting image velocity from the outputs of these filters. Section 4.3 reformulates the model as a parallel mechanism that computes a distributed representation of image velocity. In section 5, I formulate a measure of uncertainty in the velocity estimates. Section 6 discusses how the model deals with the aperture problem, comparing its performance to that of the human visual system.

2 Motion in the Frequency Domain

Several authors [4,5,6,7,8,12] have pointed out that some properties of image motion are most evident in the Fourier domain. This section describes one-dimensional motion in terms of spatial and temporal frequencies and observes that the power spectrum of a moving one-dimensional signal occupies a line in the spatiotemporal-frequency domain. Analogously, the power spectrum of a translating two-dimensional texture occupies a tilted plane in the frequency domain.

One-Dimensional Motion. The spatial frequency of a moving sine wave is expressed in cycles per unit of distance (e.g., cycles per pixel), and its temporal frequency is expressed in cycles per unit of time (e.g., cycles per frame). Velocity which is distance over time or pixels per frame, equals the temporal frequency divided by the spatial frequency:

$$v = \omega_t/\omega_x \tag{1}$$

When a signal is sampled evenly in time frequency components greater than the Nyquist frequency (1/2 cycle per frame) become undersampled, or aliased. As a consequence, if a sine wave pattern is shifted more than half its period from frame to frame it will appear to move in the opposite direction. For example, a sine wave with a spatial frequency of 1/2 cycle per pixel can have a maximum velocity of one pixel per frame and a sine wave with spatial frequency 1/4 cycle per pixel can have a maximum velocity of two pixels per frame. In other words, the range of possible velocities of a moving sine wave is limited by its spatial frequency.

Now consider a one-dimensional signal moving with a given velocity v that has many spatial-frequency components. Each such component ω_x has a temporal frequency of $\omega_{t_1} = \omega_x v$, while each spatial-frequency component $2\omega_x$ has twice the temporal frequency $\omega_{t_2} = 2\omega_x v$. In fact, the temporal frequency of this moving signal as a function of its spatial frequency is a straight line passing through the origin where the slope of the line is v.

Two-Dimensional Motion. Analogously, two-dimensional patterns (textures) translating in the image plane occupy a plane in the spatiotemporal-frequency domain:

$$\omega_t = u\omega_x + v\omega_y \tag{2}$$

where $\mathbf{v} = (u,v)$ is the velocity of the pattern [8]. For example, the expected value of the sample power spectrum of a translating random-dot field is a constant within this plane and zero outside of it.

If the motion of a small region of an image may be approximated by translation in the image plane, the velocity of the region may be computed in the Fourier domain by finding the plane in which all the power resides. To extract optical flow we could take small spatiotemporal windows out of the image sequence and fit a plane to each of their power spectra. Below I present a technique for estimating velocity by using motion-sensitive spatiotemporal Gabor-energy filters to efficiently sample these power spectra (as depicted in figure 3).

The Aperture Problem in the Frequency Domain. An oriented pattern, such as a two-dimensional sine grating or an extended step edge, suffers from what has been called the aperture problem

(for example, see Hildreth [13]). For such a pattern there is not enough information in the image sequence to disambiguate the true direction of motion. At best, we may extract only one of the two velocity components as there is one extra degree of freedom. In the spatiotemporal-frequency domain the power spectrum of such an image sequence is restricted to a line and the many planes that contain the line correspond to the possible velocities. Normal flow, defined as the component of motion in the direction of the image gradient, is the slope of that line.

3 Motion-Sensitive Filters

Adelson and Bergen [9] have pointed out that image motion is characterized by orientation in space-time. For example, figure 1(a) depicts a vertical bar moving to the right over time. Imagine that we film a movie of this stimulus and stack the consecutive frames one after the next. We end up with a three-dimensional volume (space-time cube) of luminance data like that shown in figure 1(b). Figure 1(c) shows an $x - t$ slice through this space-time cube; the slope of the edges in the $x - t$ slice equals the horizontal component of the bar's velocity (change in position over time). The figure also depicts a linear filter that is tuned for the motion of this moving bar. Thus, motion is like orientation in space-time and spatiotemporally oriented filters can be used to detect it. Three-dimensional Gabor-energy filters, presented below, are such oriented spatiotemporal filters.

3.1 Gabor-Energy Filters

A one-dimensional sine- (or odd-) phase Gabor filter is simply a sine wave multiplied by a Gaussian window:

$$g(t) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{\frac{-t^2}{2\sigma^2}\right\} \sin(2\pi\omega t) \qquad (3)$$

These filters were originally introduced by Gabor [14]. The power spectrum of a sine wave is a pair of impulses located at ω and $-\omega$ in the frequency domain. The power spectrum of a Gaussian is itself a Gaussian (i.e., it is a lowpass filter). Since multiplication in the space (or time) domain is equivalent to convolution in the frequency domain, the power spectrum of a Gabor filter is the sum of a pair of Gaussians centered at ω and $-\omega$ in the frequency domain, i.e., it is a bandpass filter. Thus, a Gabor function is localized in a Gaussian window in the space (or time) domain

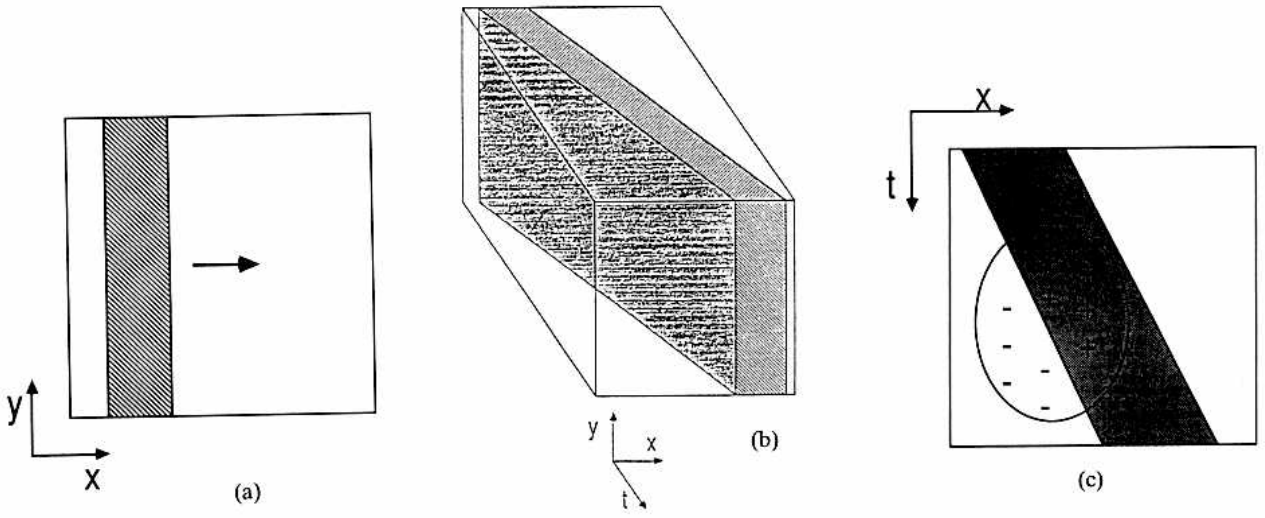


Fig. 1. Spatiotemporal orientation. (a) A vertical bar translating to the right. (b) The space-time cube for a vertical bar moving to the right. (c) An $x - t$ slice through the space-time cube. The orientation of the edges in the $x - t$ slice is the horizontal component of the velocity. Motion is like orientation in space-time and spatiotemporally oriented filters can be used to detect it. (Redrawn from Adelson and Bergen [9].)

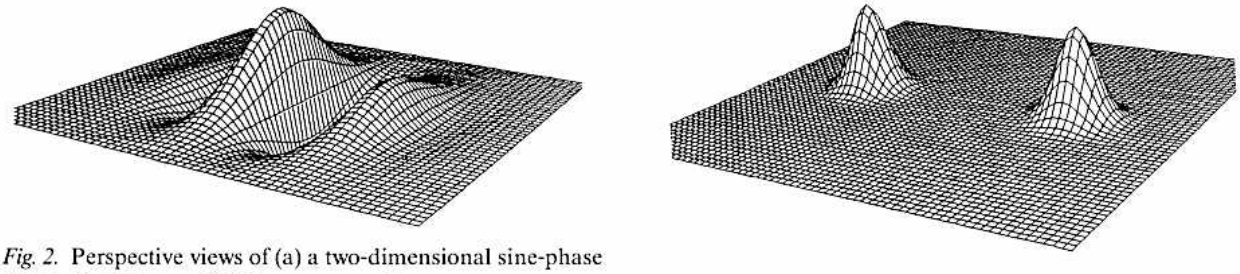


Fig. 2. Perspective views of (a) a two-dimensional sine-phase Gabor function and (b) its power spectrum.

and it is localized in a pair of Gaussian windows in the frequency doamin.

Daugman [15,16] has extended Gabor filters to a family of two-dimensional functions, an example of which is shown along with its power spectrum in figure 2.

An example of a 3D (space-time) Gabor filter is

$$g(x,y,t) = \frac{1}{\sqrt{2}\pi^{3/2}\sigma_x\sigma_y\sigma_t} \times \exp\left\{-\left(\frac{x^2}{2\sigma_x^2} + \frac{y^2}{2\sigma_y^2} + \frac{t^2}{2\sigma_t^2}\right)\right\} \times \sin\left(2\pi\omega_{x_0}x + 2\pi\omega_{y_0}y + 2\pi\omega_{t_0}t\right) \tag{4}$$

where $(\omega_{x_0},\omega_{y_0},\omega_{t_0})$ is the center frequency (the spatial and temporal frequency for which this filter gives its greatest output) and $(\sigma_x,\sigma_y,\sigma_t)$ is the spread of the spatiotemporal Gaussian window. Three-dimensional Gabor functions look something like a stack of plates with small plates on the top and bottom of the stack and the largest plates in the middle of the stack. The stack can be tilted in any orientation in space-time.

It is a simple matter to tune the filter to different frequencies and orientations while trading bandwidth for localization. To change the frequency tuning we independently vary ω_{x_0},ω_{y_0}, and ω_{t_0}. Narrowing the Gaussian window in the space-time domain broadens the bandpass window in the spatiotemporal-frequency domain and vice versa.

Gabor filters have the additional property that they can be built from separable components, thereby greatly increasing the efficiency of the computations. A new technique for computing Gabor filter outputs from separable convolutions is presented in [17]. Let k be the size of the convolution kernel, let m be the number of images in a sequence, and let each image be n pixels in size. By simplifying the complexity of three-dimensional convolution from $O(k^3n^2m)$ to $O(kn^2m)$, separability speeds it up by two orders of magnitude, given a kernel size of 10 pixels.[1]

The model presented in the following sections employs quadrature pairs of filters, odd-phase and even-phase filters of identical orientation and bandwidth. The sum of the squared output of a sine-phase filter, equation (4), plus the squared output of a cosine-phase filter gives a measure of Gabor energy that is invariant to the phase of the signal. The frequency response of such a Gabor-energy filter is the sum of a pair of 3D Gaussians (a one-dimensional version of this equation is derived in [18]):

$$G(\omega_x,\omega_y,\omega_t) = (\tfrac{1}{4})\exp\{-4\pi^2[\sigma_x^2(\omega_x - \omega_{x_0})^2 + \sigma_y^2(\omega_y - \omega_{y_0})^2 + \sigma_t^2(\omega_t - \omega_{t_0})^2]\} (\tfrac{1}{4})\exp\{-4\pi^2[\sigma_x^2(\omega_x + \omega_{x_0})^2 + \sigma_y^2(\omega_y + \omega_{y_0})^2 + \sigma_t^2(\omega_t + \omega_{t_0})^2]\} \tag{5}$$

Equation (5) means that a motion-energy filter with center frequency $(\omega_{x_0},\omega_{y_0},\omega_{t_0})$ will give an output of $G(\omega_x,\omega_y,\omega_t)$ for a moving sine grating with spatial and temporal frequencies $(\omega_x,\omega_y,\omega_t)$. The filter will give a large output for a stimulus that has a lot of power near the filter's center frequency and it will give a smaller output for a

[1]Complexity is defined as the order of magnitude, $O()$, of the number of operations required for a computation.

stimulus that has little power near the filter's center frequency.

In principle, the model could be built using oriented spatiotemporal bandpass filters other than Gabor filters. For example, Mallat [19] and Adelsen and Simoncelli [34] have proposed an orthonormal multiscale representation for two-dimensional images that could be extended to space-time. Also, it may be important for some applications to eliminate delay and use filters with a causal temporal response (Gabor filters are not causal) like those suggested by Adelson and Bergen [9] or Watson and Ahumada [8].

3.2 A Family of Motion-Energy Filters

The model uses a family of Gabor-energy filters all of which are tuned to the same spatial frequency band but to different spatial orientations and temporal frequencies, i.e., $\omega_0 = \sqrt{\omega_{x_0}^2 + \omega_{y_0}^2}$ is constant for all of the filters in one such family.

Eight of the twelve energy filters used in the present implementation have their peak response for patterns moving in a given direction—for example, one of them is most sensitive to rightward motion of vertically oriented patterns, while another is most sensitive to leftward motion. The other four filters have their peak response for stationary patterns, each with a different spatial orientation. The power spectra of the 12 filters are pairs of 3D Gaussians (each pair of Gaussians corresponds to one filter) that are positioned on the surface of a cylinder in the spatiotemporal-frequency domain (figure 3): eight of them around the top of the cylinder, eight of them around the middle, and eight around the bottom.

We can build several such families of filters tuned to different spatiotemporal-frequency bands. For the current implementation I have opted to compute a Gaussian pyramid [20] for each image in the sequence and I convolve with a single family of filters at each level of the pyramid. This is essentially the same as using families of filters with equal bandwidths that are spaced one octave apart in spatial frequency, but are tuned to the same temporal frequencies. Filters higher up in the pyramid achieve their peak response for patterns with lower spatial frequency, but with the same temporal frequency. Thus, the lower-frequency filters have their greatest outputs for patterns moving at greater velocities.

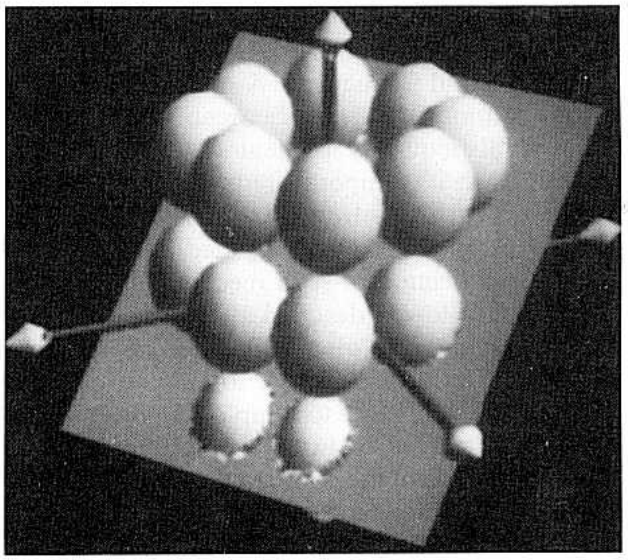


Fig. 3. The power spectra of the 12 motion-sensitive Gabor-energy filters are positioned in pairs on a cylinder in the spatiotemporal-frequency domain. Each symmetrically positioned pair of ellipsoids represents the power spectrum of one filter. The plane represents the power spectrum of a translating texture. A filter will give a large output only for a stimulus that has a lot of power near the centers of its corresponding ellipsoids and it will give a relatively small output for a stimulus that has no power near the centers of its ellipsoids. Each velocity corresponds to a different tilt of the plane, and thus to a different distribution of outputs for the collection of motion-energy mechanisms.

Psychophysical evidence [21,6] suggests that human motion channels exhibit such a relationship between spatial frequency and velocity. This makes sense from a computational viewpoint since patterns containing only high spatial frequencies may move at only low velocities, whereas patterns containing only lower spatial frequencies may move at greater velocities (see the discussion in section 2 on sampling and temporal aliasing).

4 Motion Energy to Extract Image Flow

Spatiotemporal bandpass filters like Gabor-energy filters and those filters discussed in pre-

vious papers [8,9,11] are *not* velocity-selective mechanisms, but rather are tuned to particular spatiotemporal frequencies. A single such mechanism cannot distinguish between variations in the spatial-frequency content of the stimulus, variations in its temporal-frequency content, or variations in its contrast. But, an unambiguous velocity estimate may be computed from the outputs of a collection of such mechanisms.

In what follows I describe a new way of combining the outputs of a collection of motion-energy mechanisms in order to extract velocity. The role of the filters is to sample the power spectrum of the moving texture. The problem is to estimate the slope of the plane in the frequency domain that corresponds to the actual velocity. First, I derive equations for Gabor energy resulting from motion of random textures or random-dot fields. Based on these equations I formulate a least-squares estimate of velocity.

Consider an analogous two-dimensional problem—estimating the slope of a line that passes through the origin by viewing it with a finite number of circular windows. Figure 4 shows a dotted line and two circular windows. We are given a family of such windows, a finite number of them centered at known positions. The only information we have is the number of points from the dotted line that lie within each window (in particular, we do not know the spacing between the dots). The upper window in the figure encloses many points while the lower one encloses significantly fewer. Therefore, the line must pass close to the center of the upper window while staying far from the center of the lower one. Notice that it is impossible to estimate the slope given only one circular window since the number of dots within a particular window depends both on the slope of the line and on the dot density.

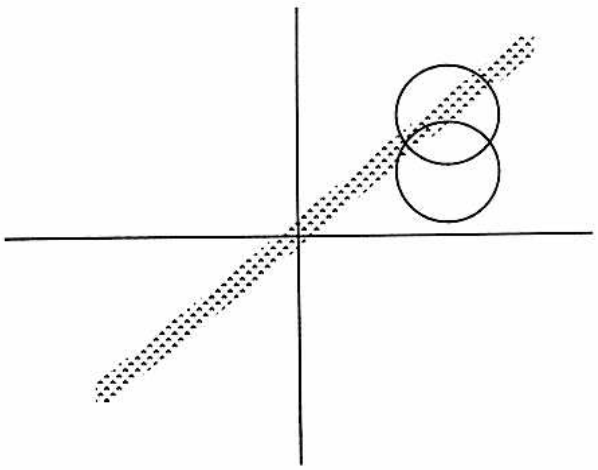


Fig. 4. A problem analogous to that of extracting velocity—estimating the slope of a line that passes through the origin by viewing it with a finite number of circular windows. The upper window encloses many points while the lower one encloses significantly fewer. In other words, the line must pass close to the center of the upper window while staying far from the center of the lower one.

4.1 Extracting Pattern Flow

In order to extract image velocity from the outputs of motion-energy filters we replace, in figure 4, both the dotted line with a plane and the circular windows with 3D Gaussian windows. A circular window simply counts the number of points it encloses. A Gaussian window counts the points and weights each according to its distance from the center of the window. This is formalized by Parseval's theorem that states that the integral of the squared values over the space-time domain is proportional to the integral of the squared Fourier components over the frequency domain:

$$\begin{aligned}&\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}|f(x,y,t)|^2\,dx\,dy\,dt\\&=\frac{1}{8\pi^3}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}|F(\omega_x,\omega_y,\omega_t)|^2\,d\omega_x\,d\omega_y\,d\omega_t\\&=\frac{1}{8\pi^3}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}P(\omega_x,\omega_y,\omega_t)\,d\omega_x\,d\omega_y\,d\omega_t\end{aligned}\qquad(6)$$

where $F(\omega_x,\omega_y,\omega_t)$ is the Fourier transform of $f(x,y,t)$ and $P(\omega_x,\omega_y,\omega_t)$ is the power spectrum. Convolution with a bandpass filter results in a signal that is restricted to a limited range of frequencies. Therefore, the integral of the square of the convolved signal is proportional to the integral of the power of the original signal over this range of frequencies.

Parseval's theorem may be used to derive an equation that predicts the output of a Gabor-energy filter in response to a moving random texture. The expected value of the sample power

spectrum of a translating random-dot field is zero, except within a plane [equation (2)] where it is a constant k. The frequency response of a Gabor-energy filter is the sum of a pair of 3D Gaussians. By Parseval's theorem, Gabor energy in response to a moving random texture is twice the integral of the product of a 3D Gaussian and a plane—by substituting equation (2) for ω_t in equation (5), multiplying by two, and integrating over the frequency domain we get

$$\begin{aligned}&\mathscr{R}(u, v, k; \omega_{x_0}, \omega_{y_0}, \omega_{t_0})\\ &= \frac{k^2}{2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \exp\{-4\pi^2[\sigma_x^2(\omega_x - \omega_{x_0})^2\\ &\quad + \sigma_y^2(\omega_y - \omega_{y_0})^2\\ &\quad + \sigma_t^2(u\omega_x + v\omega_y - \omega_{t_0})^2\}\, d\omega_x d\omega_y \qquad (7)\end{aligned}$$

where $(\omega_{x_0},\omega_{y_0},\omega_{t_0})$ is the center frequency of the motion-energy filter, $(\sigma_x,\sigma_y,\sigma_t)$ is the spread of the filter's spatiotemporal Gaussian window, (u, v) is the velocity of the stimulus, and k is proportional to image contrast. This integral evaluates to

$$\begin{aligned}\mathscr{R}(u, v, k; \omega_{x_0}, \omega_{y_0}, \omega_{t_0}) &= H_4(u, v, k)\\ &\quad \times \exp[-4\pi^2\sigma_x^2\sigma_y^2\sigma_t^2\\ &\quad \times H_1(u, v; \omega_{x_0}, \omega_{y_0,\omega_{t_0}})] \qquad (8)\end{aligned}$$

$$H_1(u, v; \omega_{x_0}, \omega_{y_0}, \omega_{t_0}) = \frac{H_2(u, v)}{H_3(u, v)}$$

$$H_2(u, v; \omega_{x_0}, \omega_{y_0}, \omega_{t_0}) = (u\omega_{x_0} + v\omega_{y_0} + \omega_{t_0})^2$$

$$H_3(u, v) = (u\sigma_x\sigma_t)^2 + (v\sigma_y\sigma_t)^2 + (\sigma_x\sigma_y)^2$$

$$H_4(u, v, k) = \frac{k^2}{8\pi\sqrt{H_3(u, v)}}$$

Equation (8) means that a motion-energy filter with center frequency $(\omega_{x_0}, \omega_{y_0}, \omega_{t_0})$, will give an output of $\mathscr{R}(u, v, k)$ for a random-dot texture moving with speed (u, v). If we multiply the grey levels at each pixel of the image sequence by a constant k, then the filter's output will increase by a factor of k^2.

For a family of Gabor-energy filters, we get a system of equations (one for each filter) in the three unknowns (u, v, k). The factor $H_4(u, v, k)$ which appears in each of these equations does not depend on the center frequency of the filters—it can be eliminated by dividing each equation by the sum or average of them all. This results in a system of equations depending only on u and v that predict the outputs of the family of Gabor-energy filters. These predicted energies are exact for a pattern with a flat power spectrum.

But, what if the power spectrum of the pattern is not flat? In particular, what if the image contrast is different for different spatial orientations? Rather than dividing each filter output by the sum of *all* of the filter outputs, we can group the filters according to their spatial orientation and normalize each spatial orientation separately. Filters that differ only in their temporal-frequency tunings line up in vertical columns in the spatiotemporal-frequency domain (see figure 3). One such column is sensitive only to a small range of spatial frequencies and orientations.

In order to specify a procedure for estimating velocity, we must now introduce some additional notation. Let $m_i (i = 1 - 12)$ be the twelve measured motion energies where each i corresponds to the output of a filter with a different center frequency. For each m_i, let $\mathscr{R}_i(u, v)$ be the corresponding predicted motion energy,

$$\begin{aligned}&\mathscr{R}_i(u, v)\\ &= \exp\{-4\pi^2\sigma_x^2\sigma_y^2\sigma_t^2 H_1(u, v; \omega_{x_i}, \omega_{y_i}, \omega_{t_j})\}\end{aligned} \qquad (9)$$

where $H_1(u, v; \omega_{x_i}, \omega_{y_i}, \omega_{t_j})$ is defined in equation (8). In addition, let $\bar{m}_i$ be the sum of the outputs of those filters that have the same preferred spatial orientation as the ith filter, and let $\bar{\mathscr{R}}_i(u, v)$ be the corresponding sum of the predicted motion energies,

$$\bar{m}_i = \sum_{j \in M_i} m_j \qquad (10)$$

$$\bar{\mathscr{R}}_i = \sum_{j \in M_i} \mathscr{R}_j(u, v)$$

where M_i is the set of motion-energy filters that share the same spatial orientation as the ith filter.

A least-squares estimate for (u, v) minimizes

the difference between the predicted and measured motion energies, i.e., it minimizes

$$I(u, v) = \sum_{i=1}^{12} \left[m_i - \bar{m}_i \frac{\mathscr{R}_i(u, v)}{\bar{\mathscr{R}}_i(u, v)} \right]^2 \qquad (11)$$

There are standard numerical methods for estimating $\mathbf{v} = (u, v)$ to minimize equation (11), e.g., the Gauss-Newton gradient-descent method [22].

Alternatively, the least-squares estimate of $\mathbf{v} = (u, v)$ maximizes

$$f(u, v) = (2\pi)^{-6}\sigma^{-12} \exp\left\{ \frac{1}{2\sigma^2} \sum_{i=1}^{12} \left[m_i - \bar{m}_i \frac{\mathscr{R}_i(u, v)}{\bar{\mathscr{R}}_i(u, v)} \right]^2 \right\} \qquad (12)$$

for some constant, σ.

Equation (12) is a response surface; the location of the peak in this surface coresponds to the velocity extracted by the model. Section 5 describes how equation (12) can be used to compute a distributed representation of image velocity.

4.2 The Algorithm

The main steps in the computations performed by the model are: (1) to convolve the image sequence with 3D Gabor filters; (2) to compute motion energy as the squared sum of the sine- and cosine-phase Gabor-filter outputs; (3) to estimate velocity by either minimizing equation (11) or maximizing equation (12). In this section I explain the additional steps that need to be computed and I summarize the entire algorithm.

Firstly Parseval's theorem, equation (6), relates an integral over the space-time domain to an integral over the frequency domain—since the filters are localized in both domains, convolving with a Gaussian is one way to approximate this integral. We can think of the model as computing the average image velocity within this Gaussian window.

Of course, Gaussian convolution will tend to smooth over motion boundaries and other regions where the velocity changes rapidly from point to point. Some possible solutions to this problem are: (1) to use images of higher resolution; (2) to use a different method for combining information other than Gaussian convolution, e.g., relaxation labeling methods (for references, see Hummel and Zucker [23]) or finite-element regularization methods (for references, see Terzopoulos [24] or Poggio et al. [25]).

There are two situations for which this smoothing problem is particularly bad. First, in regions moving with high speed we must use filters that are higher in the pyrmid, i.e., of lower spatial resolution. Second, where there is a region of low image contrast adjacent to one of high contrast the filter outputs for the high-contrast region (since they are greater on average) will bias the velocity estimates for the low-contrast region. The former situation may be controlled by incorporating eye/camera movements—an initial low-resolution estimate may be used to drive tracking eye movements thereby decreasing the image velocity and allowing for estimates of higher spatial resolution. The latter situation may be avoided by adaptation (automatic gain control)—for example, we may 'equalize' image contrast by computing the zero-crossings [26] of each image and then applying the model to the resulting zero-crossing image sequence.

Finally, a problem with Gabor filters is that all but the sine-phase filters have some dc response. If an image is very bright (large mean luminance) and of low contrast the output of the filter may be dominated by response to the dc rather than to the image-contrast signal. Clearly this is undesirable. This difficulty can be alleviated by first subtracting the local mean luminance, e.g., by convolving with a center-surround filter that has a very sharp positive center and a broad negative surround. The dc problem may also be alleviated by using only sine-phase filters—if the stimulus has uncorrelated random phase, then a phase-independent motion energy can be computed from sine-phase filters alone by averaging their squared outputs within appropriately sized windows.

In summary, an algorithm for extracting image flow proceeds as follows:

1. Compute a Gaussian pyramid for each image in the image sequence.

2. Convolve each of the resulting image sequences with a center-surround filter to remove the dc and lowest spatial frequencies.
3. Compute the sine- and cosine-phase Gabor-filter outputs using the separable convolutions described in [17].
4. Compute motion energy as the squared sum of the sine- and cosine-phase Gabor-filter outputs.
5. Convolve the resulting motion energies with a Gaussian to approximate the integral in Parseval's theorem.
6. Find the "best" choice of u and v given by equation (11) or (12), e.g., by employing the Gauss-Newton gradient-descent method or the parallel technique presented in section 4.3.
7. Compute the uncertainty in the velocity estimate as discussed in section 5.

4.3 Parallel Distributed Processing

Electrophysiological studies of the middle temporal (MT) area in macaque and owl monkeys reveal cells that are velocity tuned. Thus, it is generally believed that one of the functions of MT cells is to encode local image velocity. This section describes how the conditional probability density given by equation (12) can be used to compute a distributed representation of image velocity.

The distributed representation of image velocity is made up of velocity-tuned units analogous to the velocity-tuned cells of area MT. The outputs of each of the velocity-tuned units are computed in parallel by combining the motion-energy measurements (recall that the motion-energy filters are not themselves velocity tuned since they confound spatial frequency, temporal frequency, and image contrast).

The last step in the algorithm in section 4.2 is to find the maximum of a two-parameter function, $f(u, v)$ in equation (12). One way to locate this maximum is to evaluate the function in parallel at a number of points (say, on a fixed-square grid), and pick the largest result. The maximum can be located to any precision by using a finer or coaser grid. The grid need only be of limited extent since bandpass filtering limits the range of possible velocities (as discussed in section 2). In the context of the model each point on the grid corresponds to a velocity. Thus, evaluating the function for a particular point on the grid gives an output that is velocity tuned.

For a fixed velocity the predicted motion energies $\mathscr{R}_i(u, v)$ defined by equation (9) are fixed constants, denote them by w_{in} each where i corresponds to a different motion-energy filter and each n corresponds to a different velocity. We may rewrite equation (12) for a fixed $\mathbf{v}$ as

$$f_n = \exp\left\{\frac{1}{2\sigma^2}\sum_{i=1}^{12}\left[m_i - \bar{m}_i\frac{w_{in}}{\bar{w}_{in}}\right]^2\right\} \qquad (13)$$

where σ is a fixed constant, f_n is the response of a single velocity-tuned unit, and w_{in} and $\bar{w}_{in}$ are constant weights corresponding to the ith filter and the nth velocity. A mechanism that computes a velocity-tuned output from the motion-energy measurements performs the following operations:

1. A linear stage, a weighted summation given by $[m_i - \bar{m}_i(w_{in}/\bar{w}_{in})]$.
2. A nonlinear stage, squaring.
3. A second linear stage, the summation over i.
4. A second nonlinear stage, multiplication by $1/2\sigma^2$, and exponentiation.

The model's computations are simply a series of linear steps (convolutions, weighted sums) alternating with point nonlinearities (squaring, exponentiation). The model is therefore encompassed by the general framework for parallel distributed processing put forth by Rummelhart and McClelland [27].

An example of the resulting distributed representation is shown in figure 5 that displays a map of velocity space with each point corresponding to a particular velocity. The brightness at each point is the velocity-tuned output for that particular velocity, i.e., brightness is proportional to the likelihood of being the true velocity. Therefore, the maximum in the distribution of outputs corresponds to the velocity estimate and the

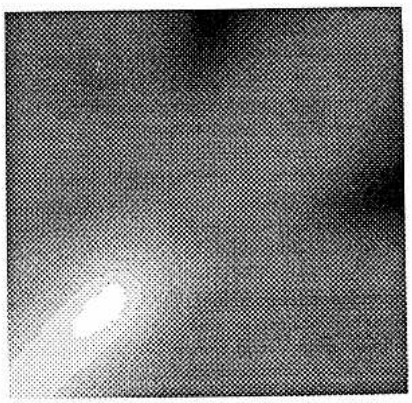


Fig. 5. Distributed representation of image velocity for a random-dot field moving leftward and downward one pixel per frame. Each point in the image corresponds to a different velocity—for example, $\mathbf{v} = (0, 0)$ is at the center of the image; $\mathbf{v} = (2, 2)$ is at the top-right corner. The maximum in the distribution of outputs corresponds to the velocity estimate and the broadness of the peak reflects the uncertainty in the estimate.

broadness of the peak reflects the uncertainty in that estimate.

4.4 Some Results

All of the results presented in this paper were produced with a single choice for each of the model's parameters—the spatial frequency tuning of each Gabor filter is $\sqrt{\omega_{x_0}^2 + \omega_{y_0}^2} = \frac{1}{4}$ cycles per pixel; the temporal frequency tunings are either $\omega_{t_0} = 0$ cycles per frame (stationary filters), or $\omega_{t_0} = \pm\frac{1}{4}$ cycles per frame (right/left, up/down, etc.); the standard deviation of all of the spatial Gaussians is $\sigma_x = \sigma_y = 4$ (the spatial kernel size of the filters is 23 pixels) and that of the temporal Gaussians is $\sigma_t = 1$ (the temporal kernel size is 7 frames). Except for the Yosemite fly-through sequence discussed below, all of the results are computed using only the lowest level of the pyramid.

Each vector in the flow fields depicted below represents a motion in a direction given by the vector's angle at a speed given by the vector's length. Errors in the velocity estimates are expressed in terms of the percentage error in each component of the actual velocity vectors.

Translating Image Sequences. Translating image sequences were generated from a textured image by: (1) blowing the image up to four times its original size; (2) shifting the resulting image by an integral number of pixels i horizontally and j vertically for each consecutive frame; (3) reducing each image in the resulting sequence back to the original resolution. The final result is an image sequence with velocity $(i/4, j/4)$ pixels per frame.

The model gives accurate velocity estimates (within 10% of the actual velocities) for translating image sequences of a wide variety of textured patterns including random-dot patterns (with dot densities ranging from 5 to 50%), images of fractal textures, some sine-grating plaid patterns (discussed in section 6), and natural textures (discussed below).[2]

Noise Sensitivity. Translating random-dot image sequences were used to study the error in the velocity estimates. For image sequences with speeds ranging from 0.25 to 1.75 pixels per frame, the absolute value of the error in the velocity estimates is proportional to the actual speed (see figure 11). The mean percentage error is −2.9% and the standard deviation is 3.6%.

Noise sensitivity was studied by adding spatiotemporal white (Gaussian) noise to translating random-dot sequences. Define the signal-to-noise ratio (S/N) to be the brightness of the image dots divided by the standard deviation of the noise. If $S/N = 10$, then the mean percentage error in the estimates is −4.3% and the standard deviation is 4.1%. This demonstrates that when the standard deviation of the sensor noise is as much as 10% of the sensor's dynamic range most velocity estimates are still within 10% of the actual values.

Images of Natural Textures. Image sequences were generated from each of the 14 natural textures

[2]Brownian fractal functions (see Mandlebrot [28] for definitions and references) are characterized by similarity across scales, and have an expected power spectrum that falls off as $P(\omega) = \omega^{-\beta}$ for some constant β. Fractals may be used to generate natural-looking textures.

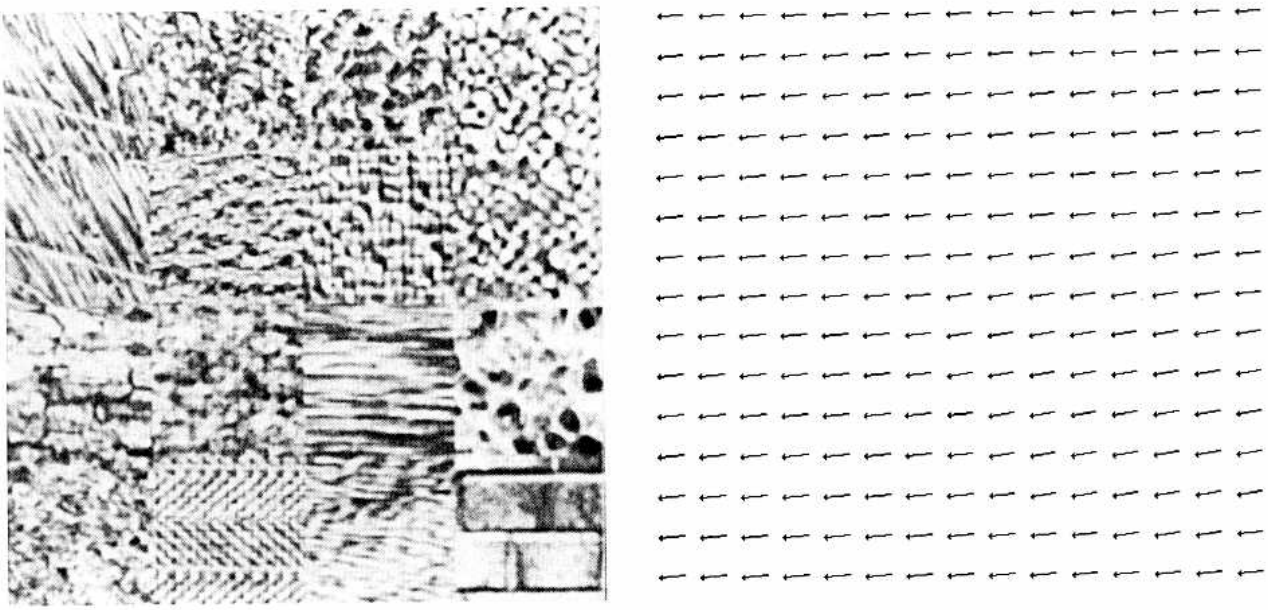


Fig. 6. (a) Fourteen natural textures (the two texture squares in the upper left are the same, and so are the two in the upper right). Each texture square was used to generate motion sequences translating 1/2 pixel per frame in each of eight directions. The velocities extracted by the model are accurate to within 10%. (b) Example flow field extracted from a motion sequence generated from the straw texture in the upper-left corner of (a). The actual motion was $(-0.5, 0.0)$. The mean of the extracted velocities is $(-0.473, -0.04)$ and the standard deviation for both the horizontal and vertical components is 0.01.

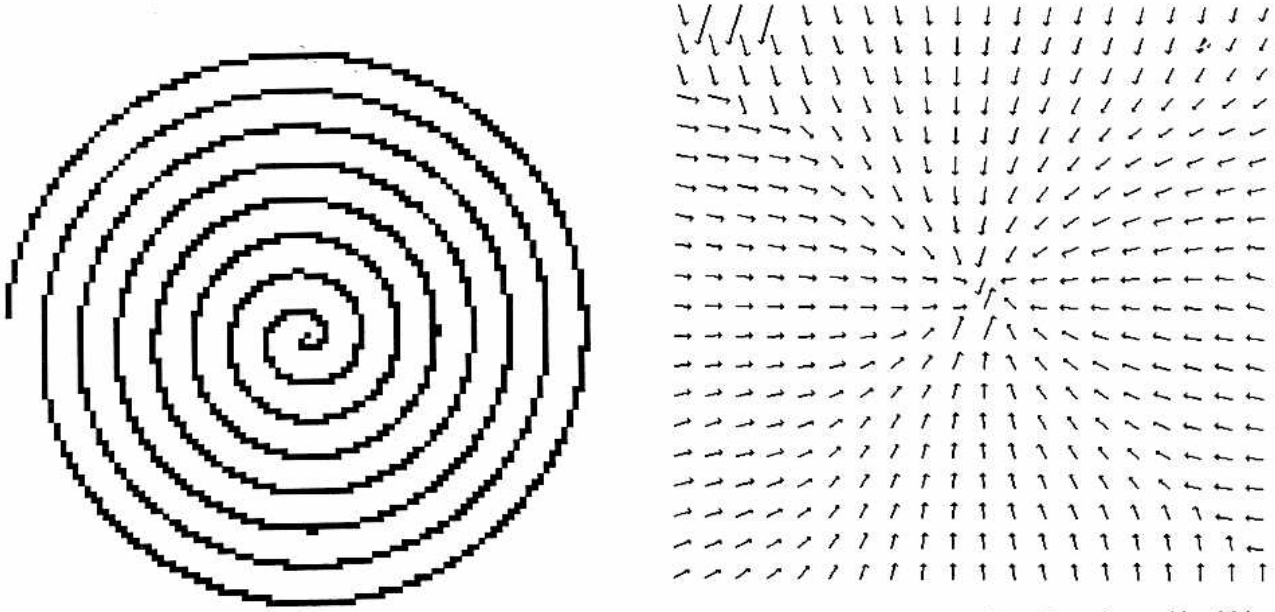


Fig. 7. (a) A frame from a motion sequence of counterclockwise rotating spiral. The perceived direction of motion is toward the center of the image and the actual displacement in that direction is $2\pi/7$ pixels per frame. (b) The extracted flow field. For 72% of the flow vectors the estimated speed is within 10% of the actual value. For 94% of the flow vectors the estimated speed is within 20% of the actual value.

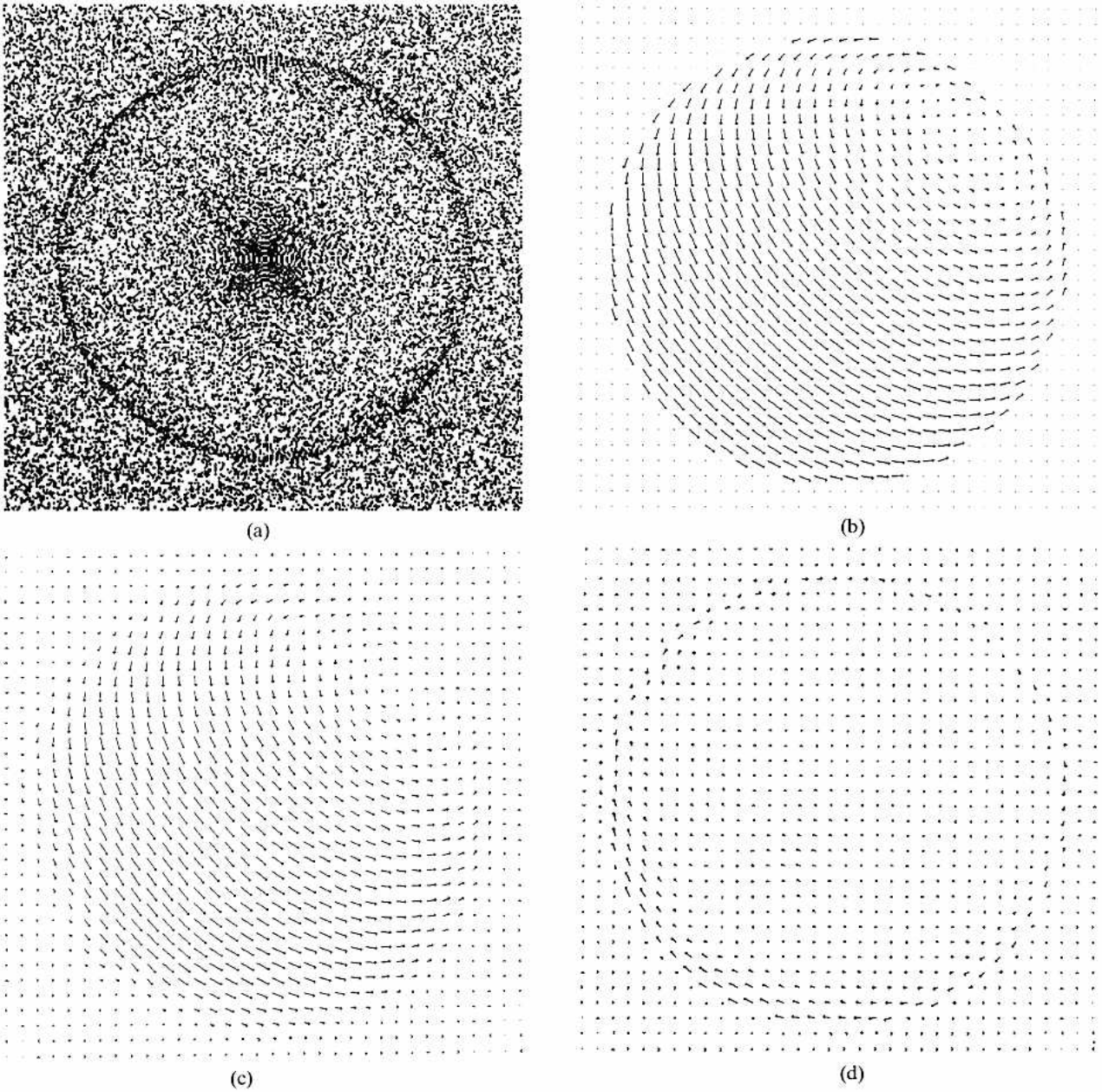


Fig. 8. A rotating random-dot sphere. (a) A frame from the motion sequence. (b) The actual flow field. (c) Flow field extracted by the model. (d) Difference between (b) and (c).

shown in figure 6(a). A sample flow field, shown in 6(b), was extracted from an image sequence of the straw texture in the upper-left corner of 6(a). The model correctly estimates the velocity (to within 10%) for every one of these textures. This is particularly impressive for the straw texture in the upper-left corner, the brick texture in lower-right corner, and the texture second from the lower-right corner of 6(a), because they have such strong spatial orientations. The model is capable of recovering accurate velocity estimates for these textures since it normalizes each spatial orientation separately in equations (11) and (12). Conversely if we were to normalize the filter outputs isotropically (i.e., by dividing each motion energy by the sum of them all), then the estimates for these three textures would be erroneous.

A Rotating Spiral. Figure 7(a) shows one frame of a rotating-spiral image sequence. The spiral, defined in polar coordinates by $r = \theta$, was rotated counterclockwise one full revolution over seven frames. Figure 7(b) shows the extracted flow field. The flow vectors point inward corresponding to what human observers see.

A Rotating Sphere. Figure 8(a) shows one frame of a random-dot image sequence of a sphere rotating in front of a stationary background. Figure 8(b) shows the actual flow field for this image sequence; 8(c) shows the flow field extracted by the model; and 8(d) shows the difference between them. The impact of the Gaussian smoothing is clearly evident as there are errors along the motion boundary.

A Realistic Example. Figure 9(a) shows one frame of a computer-generated image sequence flying through Yosemite valley. Each frame was generated by mapping an aerial photograph onto a digital-terrain map (altitude map). The observer is moving toward the horizon. The clouds in the background were generated with fractals (see Mandlebrot [28] and recent IEEE SIGGRAPH[3] conference proceedings for definitions and references) and move to the right while changing their shape over time.

Since the image velocities in the Yosemite fly-through image sequence are as high as 5 pixels per frame, we must use three levels from the pyramid. In future research, I hope to develop a rule for automatically combining estimates from the different levels. For now, I simply pick the level that is most appropriate for a given image region—the level-zero estimate is chosen if the actual velocity is between 0 and 1.25 pixels per frame, the level-one estimate is chosen if it is between 1.25 and 2.5 pixels per frame, and the level-two estimate is chosen if it is between 2.5 and 5.0 pixels per frame.

In the Yosemite fly-through image sequence, there are regions of low contrast adjacent to high-contrast regions (e.g., the face of El Capitan and the cloud region are of low contrast). This exacerbates the smoothing problem as discussed in section 4.2. For this image sequence, contrast was first equalized by computing the zero-crossings [26] of each image. The model was then applied to the resulting zero-crossing image sequence. Using the zero-crossing image sequence improves the accuracy of the velocity estimates only within the low-contrast regions. If we window the low-contrast regions to remove them from the context of the surrounding high-contrast regions, then there is little difference between the accuracy of the velocity estimates using either the zero-crossing image sequence or the original grey-level image sequence.

Figure 9(b) shows the actual flow field for this image sequence; 9(c) shows the flow field extracted by the model; and 9(d) shows the difference between them. The impact of Gaussian smoothing is evident along the boundary at the horizon. Small errors are also evident on the face of El Capitan (in the lower left) since it is moving with high speed (see the discussion in section 4.2), and in the cloud region since the clouds change shape over time while moving rightward.

5 Image-Flow Uncertainty

Information from perceptual sources is inherently noisy and uncertain. A sensing system can make substantial gains by explicitly representing the uncertainty in sensor data and taking actions to reduce it. In particular, estimates of image motion are noisy due to the stochastic nature of most textures—thus equation (8) is correct only on average. Decisions and computations that rely on motion estimates will be more robust if we keep track of uncertainty.

Section 4.3 describes how a distributed representation of image velocity may encode both a velocity estimate and the uncertainty in that estimate. For many applications, however, it is convenient to have a numerical measure of uncertainty rather than to work with the entire distribution.

This section uses tools from probability and statistical estimation theory to formulate a measure of uncertainty for image flow by characterizing the variability in the model's velocity estimates for translating image sequences of Gaussian white-noise random textures. Since image

[3]SIGGRAPH is the IEEE special interest group in computer graphics.

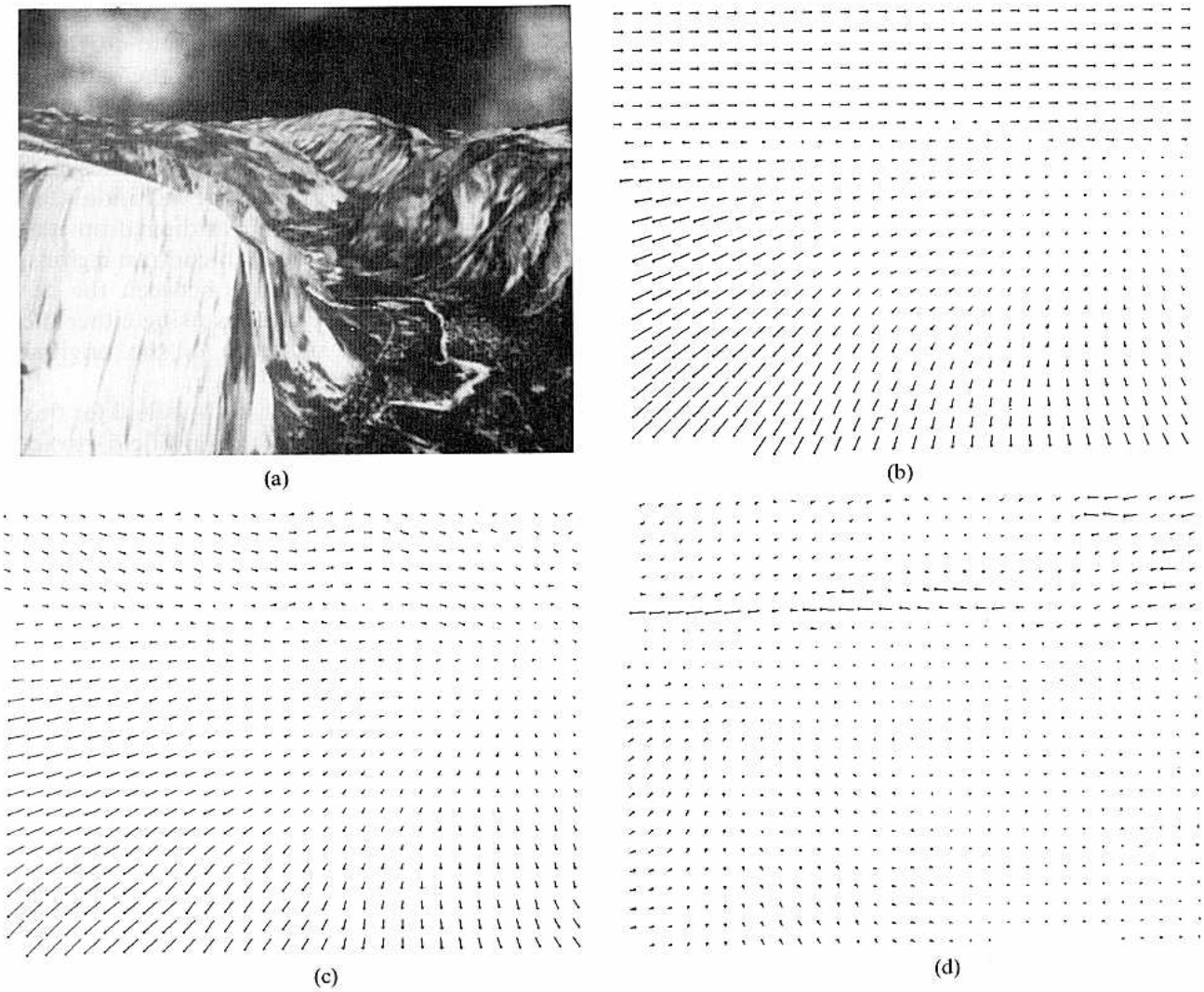


Fig. 9. (a) One frame of an image sequence flying through Yosemite valley. (b) The actual flow field. (c) Flow field extracted by the model. (d) Difference between (b) and (c).

textures are stochastic the predicted motion energies given by equation (8) are correct only on average. For a particular region of a translating image sequence the measured motion energies deviate from the expected value.

Below I posit an additive Gaussian model for the variability in the motion-energy measurements. If a normal distribution is a valid approximation for this variability, then the least squares estimate is optimal in the sense that it is equal to the maximum-likelihood estimate. Normality can be tested empirically by translating a camera a fixed distance in front of a variety of planar textured surfaces. If the camera motion is known, then the actual image translation is easily computed, and we can compare the predicted motion energies given by equation (8) to those measured from the image sequence. However, if normal distributions were not valid approximations for the measurement variability, then least-squares estimation would not be optimal. In addition, the uncertainty measure formulated below would not be accurate.

First, I review some aspects of statistical parameter estimation in the presence of additive noise. I use the notation $\hat{\theta}$ to denote estimates of the parameter θ.

Consider the case in which we take independent measurements, $\mathbf{m} = (m_1, \ldots, m_{12})$, that are nonlinearly related to an unknown parameter, $\mathbf{v} = (u, v)$, in the presence of zero-mean additive Gaussian noise, $\mathbf{n} = (n_1, \ldots, n_{12})$,

$$\mathbf{m} = \mathbf{R}(\mathbf{v}) + \mathbf{n} \tag{14}$$
$$n_i \sim N(0, \sigma_i^2)$$

for some nonlinear vector function, $\mathbf{R}(\mathbf{v}) = [\mathscr{R}_1(\mathbf{v}), \ldots, \mathscr{R}_{12}(\mathbf{v})]$. Equation (14) may be rewritten as

$$[m_i - \mathscr{R}_i(\mathbf{v})] \sim N(0, \sigma_i^2) \tag{15}$$

The error in the measurements may be represented by the Fisher information matrix (see DeGroot [29] for definition). For a jointly normal density the information matrix, denoted by $\mathbf{V}_\mathbf{m}^{-1}$ is the inverse of the variance-covariance matrix, $\mathbf{V}_\mathbf{m}$. Assuming that the measurements are independent, the information matrix is diagonal with entries

$$\mathbf{V}_\mathbf{m}^{-1} = \begin{pmatrix} \frac{1}{\sigma_1^2} & 0 & \cdots & 0 \\ 0 & \frac{1}{\sigma_2^2} & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & \frac{1}{\sigma_{12}^2} \end{pmatrix} \tag{16}$$

and the conditional probability density is given by

$$f(\mathbf{m}|\mathbf{v}) = (2\pi)^{-6}\left(\prod_{i=1}^{12} \sigma_i\right)^{-1} \times \exp\left\{\sum_{i=1}^{12} \frac{1}{2\sigma_i^2}[m_i - \mathscr{R}_i(\mathbf{v})]^2\right\} \tag{17}$$

The posterior density, $f(\mathbf{v}|\mathbf{m})$, is the probability that a certain value of $\mathbf{v}$ is equal to the true value, given the measurements $\mathbf{m}$. In the absence of prior information on the parameter $\mathbf{v}$, the posterior density and the conditional density are one in the same. The maximum-likelihood estimate (MLE), $\hat{\mathbf{v}} = (\hat{u}, \hat{v})$, is that which maximizes the conditional density, thus maximizing the probability that the estimate is equal to the true value.

The uncertainty in the estimate may be represented as an information matrix, $\mathbf{V}_\mathbf{v}^{-1}(\hat{\mathbf{v}})$, computed from $\mathbf{V}_\mathbf{m}^{-1}$ and from $\hat{\mathbf{v}}$ (see Melsa and Cohn [30] for derivation):

$$\mathbf{V}_\mathbf{v}^{-1}(\hat{\mathbf{v}}) = \mathbf{J}^T(\hat{\mathbf{v}})\mathbf{V}_\mathbf{m}^{-1}\mathbf{J}(\hat{\mathbf{v}}) \tag{18}$$

where $\mathbf{J}(\mathbf{v})$ is the Jacobian matrix of $\mathbf{R}(\mathbf{v})$ and $\mathbf{J}^T(\mathbf{v})$ is the transpose of $\mathbf{J}(\mathbf{v})$. The information matrix, $\mathbf{V}_\mathbf{v}^{-1}(\hat{\mathbf{v}})$, is a random variable that depends on the estimate $\hat{\mathbf{v}}$. In general the estimate must be reasonably close to the true value for this uncertainty measure to be accurate.

The eigenvectors and eigenvalues of the information matrix, $\mathbf{V}_\mathbf{v}^{-1}(\hat{\mathbf{v}})$, are the directions and values in parameter space (e.g., in image-velocity space) of minimum and maximum information. The mean-squared-error, given by the trace of the variance-covariance matrix, is an estimate of the actual squared-error of the estimate, that is, $\mathrm{Tr}\,[\mathbf{V}_\mathbf{v}(\hat{\mathbf{v}})]$ is an estimate of $\|(u - \hat{u}, v - \hat{v})\|^2$.

If there is only partial information about $\mathbf{v}$ then the minimum-information eignevalue is zero. For example, in the presence of the aperture problem (as discussed in section 6) there is only partial information about image motion. We represent uncertainty with the information matrix, $\mathbf{V}_\mathbf{v}^{-1}(\hat{\mathbf{v}})$, instead of using the variance-covariance matrix, $\mathbf{V}_\mathbf{v}(\hat{\mathbf{v}})$, because the latter may be undefined when there is only partial information.

Equation (18) may be used to compute the uncertainty of an image-flow estimate. But we must have a statistical model, called a *sensor model,* of the measurement variability (denoted above by $\mathbf{V}_\mathbf{m}^{-1}$).[4] As discussed at the beginning of this section, we posit a Gaussian model for the variability in the motion-energy measurements

$$m_i = K_i \mathscr{R}_i(u, v) + n_i \tag{19}$$

where $\mathscr{R}_i(u, v)$ is defined in equation (9), K_i is an unknown constant that depends on image contrast, and n_i is additive process variability. The procedure presented in section 4 for estimating image velocity picks the estimate, $(\hat{u}, \hat{v})$, to minimize equation (11) rewritten here as

[4]Note that the sensor model is *not* a statistical model of the camera noise, but rather a model of the motion-energy measurement variability.

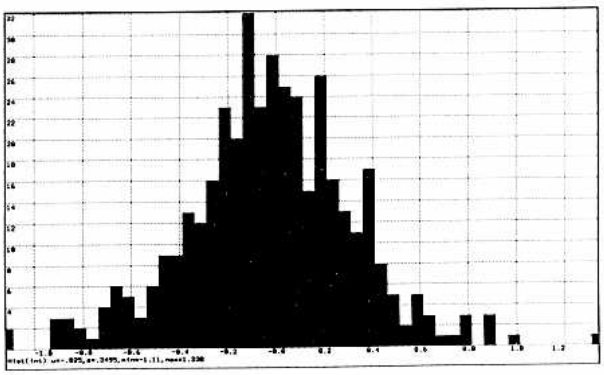


Fig. 10. Four hundred Gaussian white-noise random textures were generated, and each was used to generate a translating image sequence with the same velocity (one pixel per frame upward and rightward). The plot shows a histogram of $[m_i - \hat{K}_i \mathcal{R}_i(u, v)]^2$ for the motion-energy filter that is most sensitive for rightward motion. The data in this histogram pass both the chi-squared and the Kolomogorov–Smirnov tests for Gaussianity. The distribution is zero-mean and its variance is 0.12.

$$l(u, v) = \sum_{i=1}^{12} [m_i - \hat{K}_i \mathcal{R}_i(u, v)]^2 \qquad (20)$$

$$\hat{K}_i = \frac{\bar{m}_i}{\bar{\mathcal{R}}_i(u, v)}$$

where $\hat{K}_i$ is used as an estimate of K_i, with $\bar{m}_i$ and $\bar{\mathcal{R}}_i(u, v)$ as defined in equation (10).

Figure 10 shows a histogram of $[m_i - \hat{K}_i \mathcal{R}_i(u, v)]$ for one motion-energy filter over several hundred trials. It is zero-mean Gaussian:

$$[m_i - \hat{K}_i \mathcal{R}_i(u, v)] \sim N(0, \sigma_i^2) \qquad (21)$$

Although the data in figure 10 pass statistical tests for Gaussianity, some other examples fail these tests. Further experimentation with real image sequences is called for.

The Jacobian matrix is given by

$$\mathbf{J}(\hat{u}, \hat{v}) = \begin{pmatrix} \hat{K}_1 \dfrac{\partial \mathcal{R}_1(\hat{u}, \hat{v})}{\partial u} & \hat{K}_1 \dfrac{\partial \mathcal{R}_1(\hat{u}, \hat{v})}{\partial v} \\ \vdots & \vdots \\ \hat{K}_{12} \dfrac{\partial \mathcal{R}_{12}(\hat{u}, \hat{v})}{\partial u} & \hat{K}_{12} \dfrac{\partial \mathcal{R}_{12}(\hat{u}, \hat{v})}{\partial v} \end{pmatrix} \qquad (22)$$

where $[\partial \mathcal{R}_i(u, v)/\partial u]$ and $[\partial \mathcal{R}_i(u, v)/\partial v]$ are obtained by differentiating equation (9).[5]

The conditional probability density of the motion-energy measurements given the actual image velocity is given by

$$f(\vec{m}|u, v) \approx (2\pi)^{-6} |\mathbf{V}_v|^{-1/2} \times \exp\left\{-\frac{1}{2}[\mathbf{m} - \mathbf{KR}(\mathbf{v})]^T \mathbf{V}_m^{-1} [\mathbf{m} - \mathbf{KR}(\mathbf{v})]\right\} \qquad (23)$$

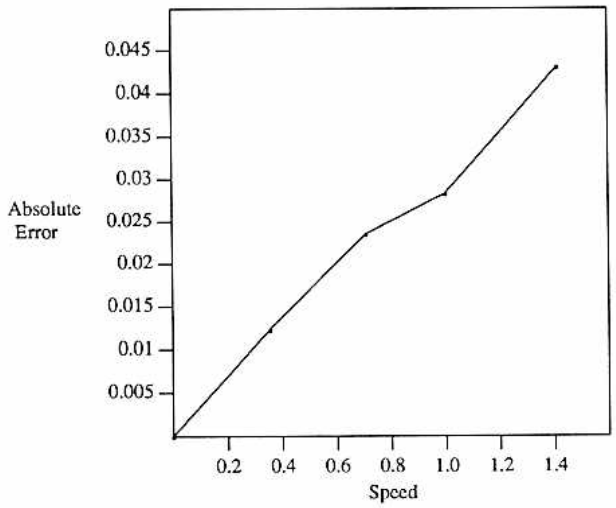
Equation (23) may be used as in section 4.3 to compute a distributed representation of image velocity.

The variances, of $\mathbf{v_m}$ are given by

$$\begin{aligned} &\operatorname{var}\left(m_i - \bar{m}_i \frac{\mathcal{R}_i(u, v)}{\bar{\mathcal{R}}_i(u, v)}\right) \qquad (24) \\ &= \left(\frac{\mathcal{R}_i(u, v)}{\bar{\mathcal{R}}_i(u, v)} - 1\right)^2 \operatorname{var}(m_i) \\ &\quad + \left(\frac{\mathcal{R}_i(u, v)}{\bar{\mathcal{R}}_i(u, v)}\right)^2 [\operatorname{var}(m_2) + \operatorname{var}(m_3)] \\ &\quad + 2\left(\frac{\mathcal{R}_i(u, v)}{\bar{\mathcal{R}}_i(u, v)} - 1\right)\left(\frac{\mathcal{R}_i(u, v)}{\bar{\mathcal{R}}_i(u, v)}\right) \\ &\qquad \times [\operatorname{cov}(m_i, m_2) + \operatorname{cov}(m_i, m_3)] \\ &\quad + 2\left(\frac{\mathcal{R}_i(u, v)}{\bar{\mathcal{R}}_i(u, v)}\right)^2 [\operatorname{cov}(m_2, m_3)] \end{aligned}$$

where m_i is the output of the ith filter, m_1 and m_2 are the outputs of the two filters that share the same orientation the ith filter, and $\mathcal{R}_i(u, v)$, $\mathcal{R}_1(u, v)$ and $\mathcal{R}_2(u, v)$ are the corresponding predicted motion energies given by equation (9). In [18] I derive an equation for the covariances of the motion-energy measurements, cov (m_i, m_j), for image sequences of translating Gaussian white-noise random textures. The covariances of $\mathbf{v_m}$ may be expressed similarly.

[5] To be thorough, we could treat the K_i's as variables and include derivatives $\{\partial[K_i \mathcal{R}_i(\hat{u}, \hat{v})]/\partial K_i\} = \mathcal{R}_i(\hat{u}, \hat{v})$ in the Jacobian matrix. But we are not interested in estimating K_i, so there is no reason to estimate the uncertainty in $\hat{K}_i$.



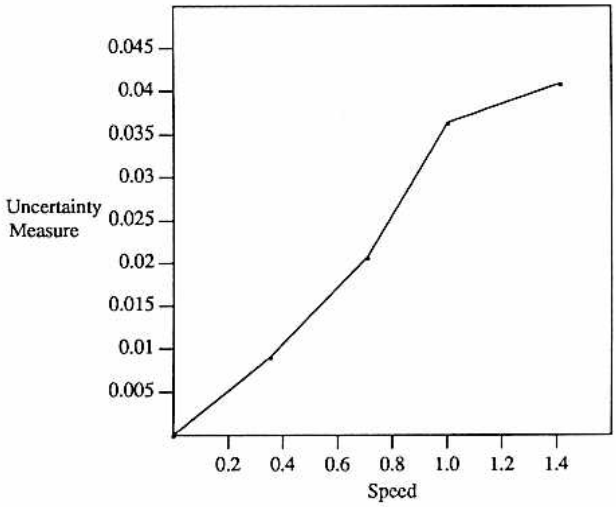


Fig. 11. Two hundred translating Gaussian white-noise random textures were generated with each of four different velocities ranging from 0.0 pixels per frame to $\sqrt{2}$ pixels per frame. (a) The average absolute error in the velocity estimates as a function of speed. The best-fit line through the data points has a slope of 0.029 and a y-intercept of 0.0024. (b) The average square-root of the trace of the estimated variance-covariance matrix as a function of speed plotted on the same scale as in (a). The best-fit line through the data points has a slope of 0.030 and a y-intercept of 0.0010.

The results presented in [18] demonstrate that this is a reasonably good characterization of the measurement variability. However since $\mathbf{v}_m$ now depends on the actual value of (u, v) we must make one further approximation—we approximate the measurement variability using the image velocity estimate, $\mathbf{v}_m(\hat{u}, \hat{v}) \approx \mathbf{v}_m(u, v)$.

In summary, we may estimate image-flow uncertainty by

1. computing the image-velocity estimate, $(\hat{u}, \hat{v})$, as discussed in section 4;
2. computing the measurement variability estimate, $\mathbf{v}_m(\hat{u}, \hat{v})$;
3. computing the information matrix, $\mathbf{V}_v^{-1}(\hat{u}, \hat{v})$, using equations (18) and (22).

A test of the accuracy of the uncertainty measure is to compare the mean-squared-error, Tr $[\mathbf{V}_v(\hat{u}, \hat{v})]$, with the actual squared error-of-velocity estimates, $\|(u - \hat{u}, v - \hat{v})\|^2$. Figure 11 shows that the uncertainty measure reflects the actual error for translating random-dot image sequences.

However, the uncertainty measure significantly underestimates the actual errors for the Yosemite fly-through image sequence (figure 9) because these errors are mainly due to the blurring problem discussed in section 4.2, not due to the motion-energy measurement variability.

6 Dealing with the Aperture Problem

In this section, I use a class of moving stimuli known as sine-grating plaids in order to test the model's capability for solving the aperture problem and I compare the model's performance to that of the human visual system. I also suggest that the uncertainty measure presented in the previous section might be used to recognize when there is an ambiguous velocity estimate resulting from the motion of a strongly oriented pattern.

6.1 Sine-Grating Plaids

A sine-grating plaid is the sum of two moving gratings and may be seen as a single coherent plaid motion. The gratings are not combined as the vector sum or vector average of the two component normal-flow velocities, but rather as the intersection of the perpendiculars to the two velocity vectors. Figure 12(a) depicts a single grating moving behind an aperture—the arrows rep-

resent flow vectors and the diagonal line represents the locus of velocities compatible with the grating's motion. There is an infinite number of such compatible motions any of which will result in exactly the same stimulus. Figure 12(b) shows a plaid composed of two orthogonal gratings moving at the same speed—the intersection of the perpendiculars to the two normal-flow velocities (the intersection of the two constraint lines) is the only shared motion, and corresponds to what is seen. Figure 12(c) shows a plaid composed of two oblique gratings, one moving slowly and the other more rapidly—one grating moves rightward and the other moves downward and rightward, but the pattern moves *upward* and rightward.

The model recovers the correct pattern-flow velocity for a number of such plaids. Examples of flow fields extracted by the model for plaids made up of gratings with equal contrasts and spatial frequencies are shown in figure 13. The combined motion extracted by the model in both 13(a) and 13(b) is accurate to within 5%.

The model does not always recover the correct pattern-flow velocity for sine-grating plaids—for example, the model's estimates are in error (correct direction of motion but wrong speed) when the spatial frequencies of the gratings are not equal to the spatial-frequency tuning of the filters.

6.2 *Sine-Grating Plaids and the Aperture Problem*

Adelson and Movshon [31] studied the phenomenon of coherence by varying the angle between the two gratings, their relative contrasts, and their relative spatial frequencies. They found that for a range of relative angles, contrasts, and spatial frequencies the two gratings are seen as a single coherent plaid motion, and that beyond this range the two gratings look like separate motions moving past one another. The phenomenon of coherence tests the ability of the human visual system to solve the aperture problem; given the ambiguous motion of a single moving grating, how much additional information is needed from the second grating to give an unambiguous coherent percept?

The model is capable of extracting the correct pattern-flow velocity for plaids that have large differences in contrast, e.g., for plaids made up of orthogonal gratings, the velocity estimates are accurate to within 10% for contrast ratios of greater than 32 : 1. This is comparable with human performance [32]. As the contrast difference between the two component gratings gets larger than this, the model begins to tilt the extracted velocity vector toward the higher-contrast grating. Although

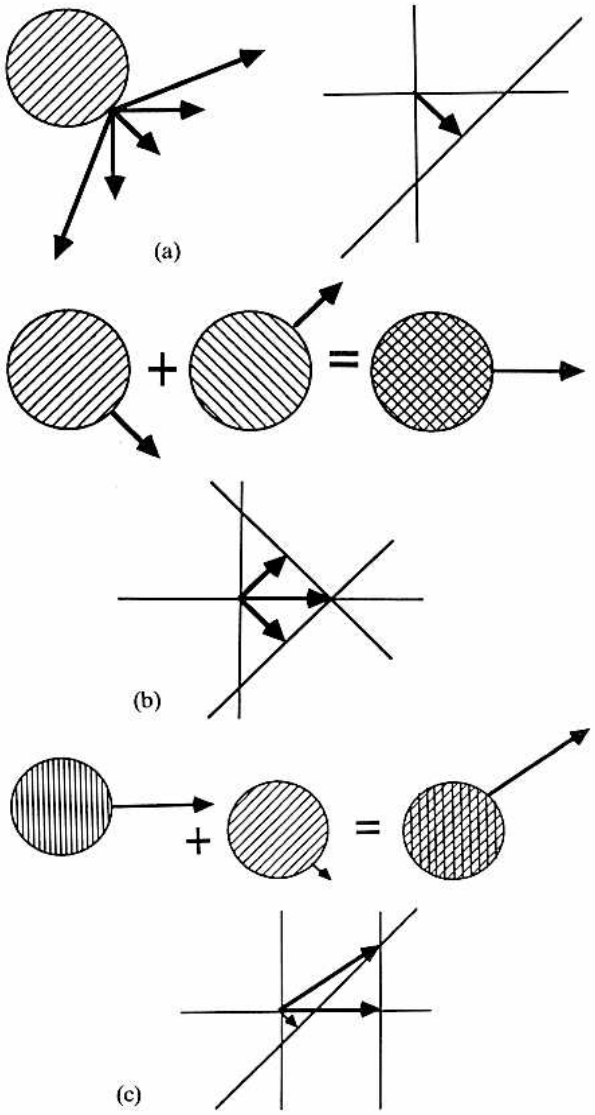


Fig. 12. The perceived motion of two moving gratings is the intersection of the perpendiculars to the two velocity vectors. (a) A single moving grating—the diagonal line indicates the locus of velocities compatible with the motion of the grating. (b) and (c) Plaids composed of two moving gratings. The lines give the possible motions of each grating alone. Their intersection is the only shared motion, and corresponds to what is seen. (Redrawn from Adelson and Movshon [31].)

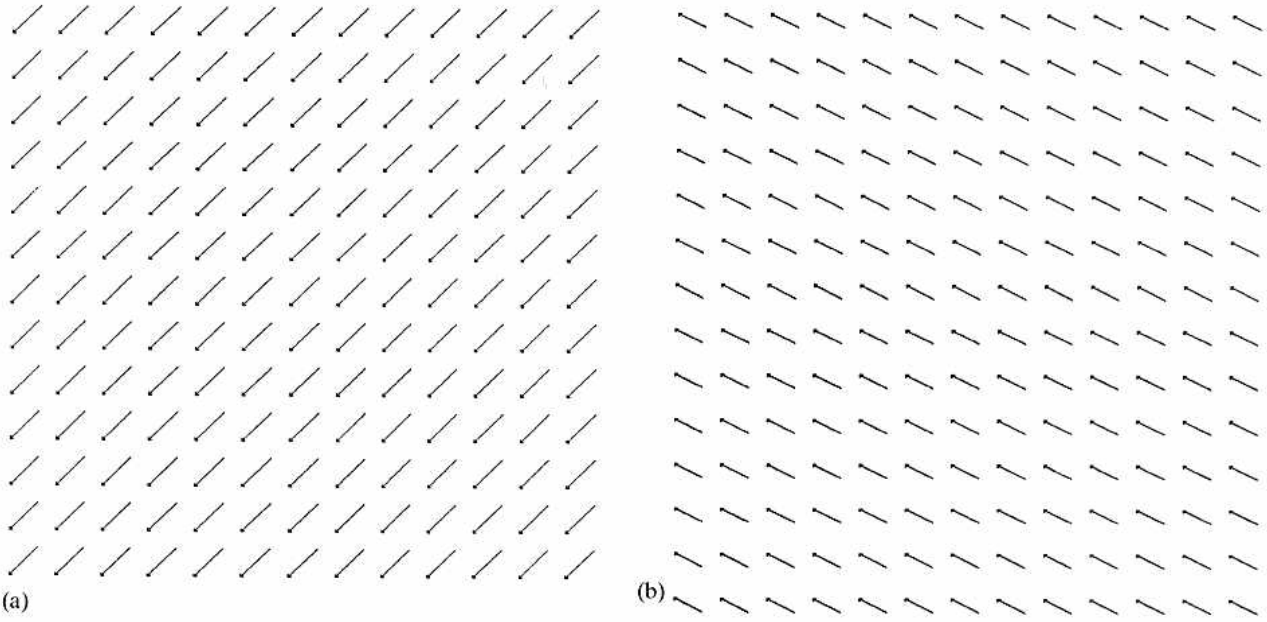


Fig. 13. (a) Flow field extracted by the model for a plaid pattern made up of a sine grafting moving leftward one pixel per frame plus a sine grating moving downward one pixel per frame. The combined motion extracted by the model is one pixel leftward and one pixel downward each frame. (b) Flow field for a plaid pattern made up of a sine grating moving leftward one pixel per frame plus a sine grating moving downward and leftward a quarter pixel each frame. The counterintuitive combined motion is leftward one pixel per frame and *upward* a half pixel per frame as shown in the flow field extracted by the model. The spatial frequency of the gratings for both (a) and (b) was 0.25 cycle pixel^{-1}.

the perceived velocity of plaids has not yet been measured precisely Adelson [32] notes that observers also see the direction of motion tilt toward the higher-contrast grating when the relative contrast difference is large.

To withstand large contrast ratios, it is crucial that the spatial bandwidths of the model's filters be less than their temporal bandwidths—in the frequency domain, this means that the filters are oblong hotdog-shaped (longer in t than in x and y) instead of spherical. As an illustrative example, consider a plaid made up of rightward- and upward-moving gratings. The idea of normalizing the filter outputs separately for each spatial orientation is that the upward- and downward-sensitive filters should give the same responses relative to one another regardless of the contrast ratio between the two gratings. If the filters were spherical in shape, then the response of the downward-sensitive filter would be dominated by the rightward-moving grating (the impulse from the rightward-moving grating is closest to the center-frequency of the downward-sensitive filter). This would be bad because we want the relative responses of the upward- and downward-sensitive filters to be unaffected by varying the contrast of the rightward-moving grating. But, since the filters are oblong in shape the response of the downward-sensitive filter is dominated by the grating moving upward for a wide range of relative contrasts.

6.3 Recognizing Ambiguity

An isotropic texture (e.g., a random-dot field) does not suffer from the aperture problem since there is enough information within a local window to disambiguate the true direction of motion. A strongly oriented pattern (e.g., a sine grating) offers only partial information about image velocity. Between these two extremes there is a continuum of stimuli offering information about image velocity that is more and more ambiguous. The level of ambiguity should be reflected by the level of uncertainty in the velocity estimate.

The distributed representation of image veloc-

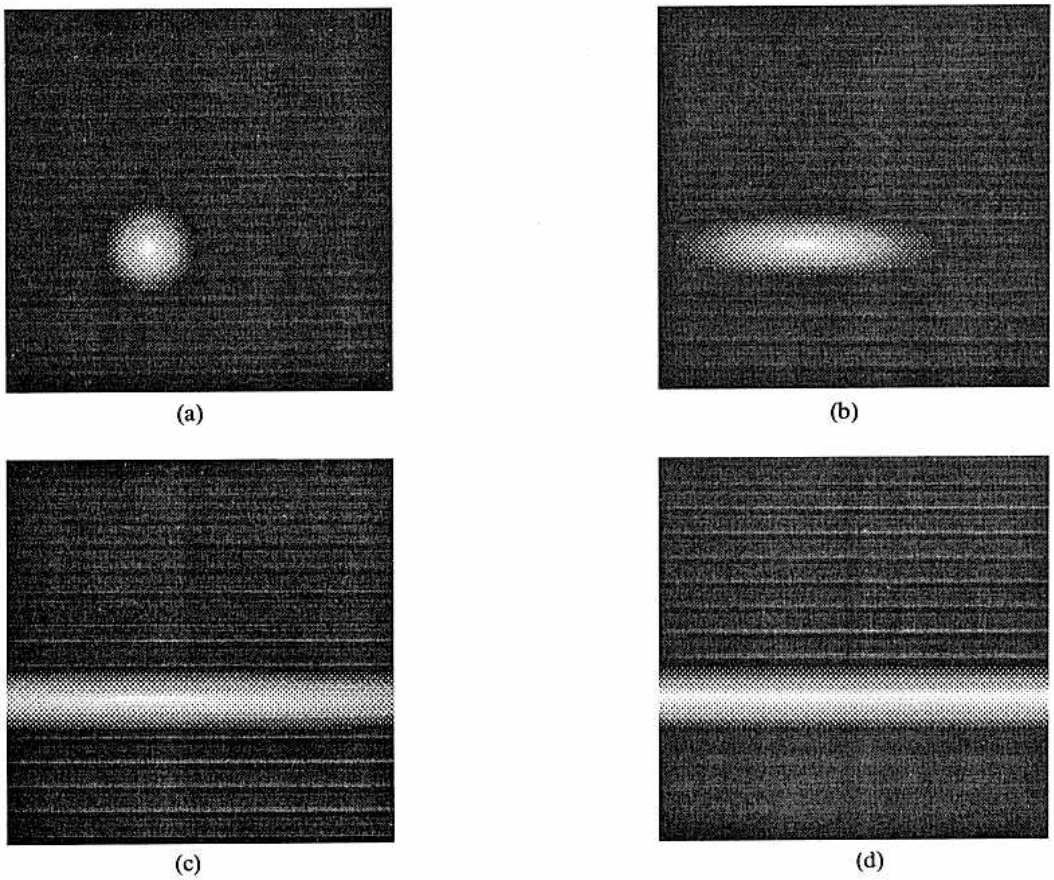


Fig. 14. Distributed representation of image velocity for sine-grating plaids made up of orthogonal gratings. The gratings moved 1 pixel frame^{-1} leftward and downward and their spatial frequency was 0.25 cycle pixel^{-1}. (a) The two component gratings had the same contrast. The location of the maximum in the distribution corresponds to the velocity extracted by the model. (b) One grating had twice the contrast of the other grating. (c) One grating had four times the contrast of the other grating. (d) One grating had zero contrast; the aperture problem is evident, as there is a ridge of maxima. Each velocity-tuned unit along this ridge has the same output (to within 1 part in 100,000).

ity introduced in section 4.3 forms a surface in velocity space; the height of the surface at a particular velocity is the likelihood that it is the true velocity. Some examples will illustrate that ambiguity due to the aperture problem is reflected by the shape of this response surface.

Figure 14 shows the distributed representation of image velocity for some sine-grating plaids. As the relative contrast of one of the component gratings is varied the peak in the surface gets broader in one direction. This is evident by comparing figures 14(a), (b), (c), and (d). In (a), the two component gratings are of equal contrast and the peak is symmetrical. When the contrast ratio is increased as in (b) and (c), the location of the peak does not change, but its shape elongates in one direction. Eventually as shown in (d), the peak turns into a ridge.

Figure 15 shows the distributed representations for image sequences generated from the straw-texture image. There is enough information in these image sequeqnces for the model to disambiguate the true direction of motion as there are clearly defined peaks in the distributions. The shape of each peak matches the orientation of the texture thereby reflecting the image-flow uncertainty.

When there is an unambiguous peak we can extract the correct pattern-flow velocity, but how

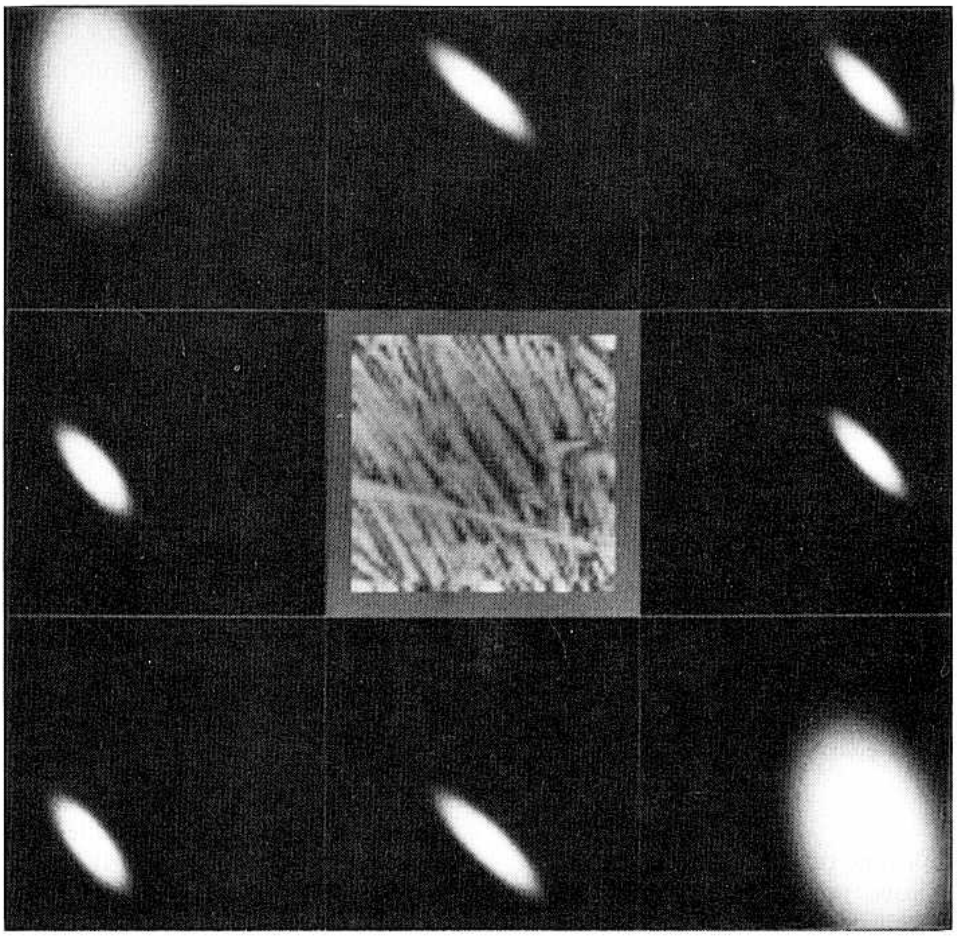


Fig. 15. Translating image sequences were generated from the straw texture shown in the middle. Each pane shows the distributed representation of velocity computed from sequences moving 1/2 pixel frame^{-1} in each of eight directions. The locations of the peaks in these distributions correspond to the velocities extracted by the model. The shape of each peak matches the orientation of the texture thereby reflecting the image-flow uncertainty.

do we know if there is a ridge or a peak? Intuitively, it is a peak if it falls off sharply in all directions and it is a ridge if it stays constant in one direction. We know from differential geometry (for example, see doCarmo [33]) that a surface can be characterized locally by its maximum and minimum curvatures. If the minimum curvature of a surface is small or zero at a point while the maximum curvature is large then the surface looks like a ridge. If both curvatures are large then it looks like a peak.

In [17] I suggest using the minimum curvature of the surface at the peak divided by the height of the peak as a measure of ambiguity due to the aperture problem. We may pick a value to act as a threshold; if the curvature measure is above this value we pick the pattern flow given by the location of the peak, and if it falls below this value we may pick the normal-flow vector or we may choose any other velocity along the ridge (a familiar example of when people see motion other than in the normal-flow direction is the barberpole illusion).

Instead, we may use the information matrix introduced in section 5 to recognize ambiguity. Define the *ambiguity mesure* as the quotient of the minimum eigenvalue of the information matrix divided by its maximum eigenvalue. Figure 16 shows a plot of the ambiguity measure as the relative contrast of a plaid's component gratings is varied. The ambiguity measure decreases monotonically with the contrast of the test grat-

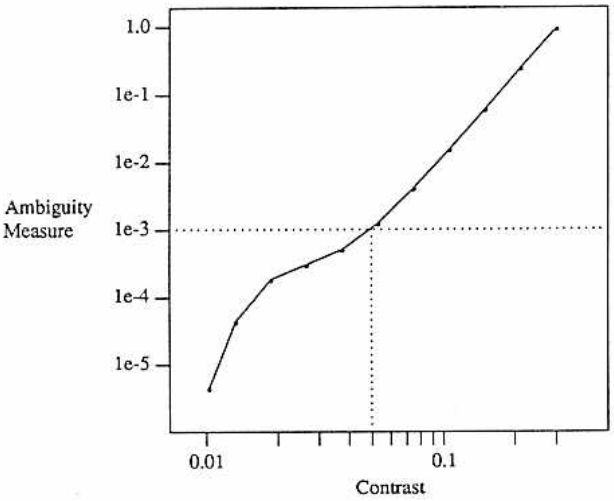


Fig. 16. The influence of contrast on the ambiguity measure. One grating had a fixed contrast of 0.3 while the other was of variable contrast. The two gratings moved at an angle of 120°, both had a spatial frequency of 0.25 cycle pixel^{-1}, and their speeds were chosen so that the coherent plaid moved at a speed of 2/3 pixel frame^{-1}. The plot shows the ambiguity measure as the contrast of the test grating was varied. The dotted lines indicate the test-grating contrast needed to attain threshold (0.001) ambiguity.

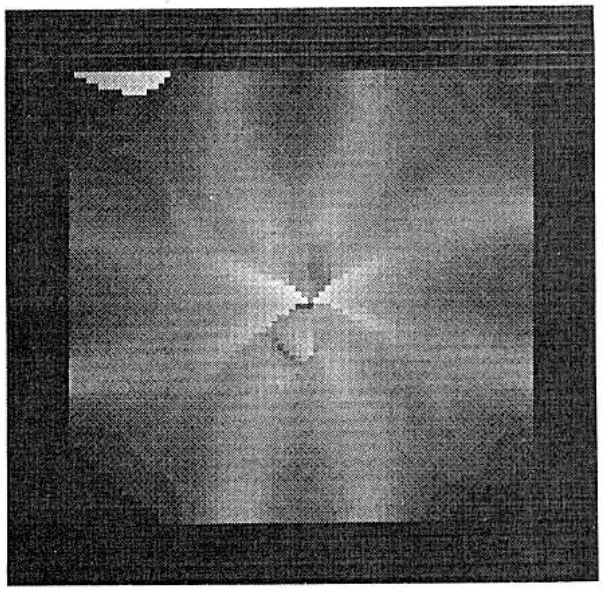


Fig. 17. The brightness at each pixel is proportional to the ambiguity measure for the rotating spiral image sequence (figure 7). The ambiguity measure reflects the ambiguity in the image sequence.

ing for a wide range of relative contrasts.

Figure 17 shows the values of the ambiguity measure for each pixel of the rotating spiral image sequence (figure 7). As we move away from the center of the image there is less and less curvature in the contour of the spiral. The ambiguity measure reflects this variation in the level of velocity ambiguity.

The results in Figures 16 and 17 indicate that the ambiguity measure may lead to a relaible test for ambiguity due to the aperture problem.

7 Summary

This paper presents a model for computing local image velocity consonant with current views regarding the neurophysiology and psychophysics of motion perception. The power spectrum of a moving texture occupies a tilted plane in the spatiotemporal-frequency domain. The model uses 3D (space-time) Gabor filters to sample this power spectrum and by combining the outputs of several such filters the model estimates the slope of the plane (i.e., the velocity of the moving texture). The model gives accurate estimates of two-dimensional velocity for a wide variety of test cases including realistic images, sequences generated from images of natural textures, and some sine-grating plaid patterns.

The error in the velocity estimates for translating image sequences is from two sources. First, since image textures are stochastic, equation (8) is correct only on average. Second, the maximum-likelihood estimate is equal to the least-squares estimate only if the variability in equation (8) is well approximated by a Gaussian process.

The primary source of error for realistic image sequences is that the model assumes image translation, ignoring motion boundaries, accelerations, deformations (rotation, divergence, shear), and motion transparency. Rather, the model computes the average image velocity within a Gaussian-shaped window.

A parallel implementation of the model results in a distributed representation of image velocity. The computations leading to this distributed representation are simply a series of linear steps (convolutions, weighted sums) alternating with

point nonlinearities (squaring, exponentiation). The model is therefore encompassed by the general framework for parallel distributed processing put forth by Rummelhart and McClelland [27].

A measure of image-flow uncertainty is formulated and it is demonstrated that this uncertainty measure reflects the actual error in the velocity estimates for translating image sequences of random textures. It is suggested that the uncertainty measure might be used to test for ambiguity due to the aperture problem.

The model appears to solve the aperture problem as well as the human visual system since it extracts the correct velocity for patterns having large differences in contrast at different spatial orientations ($> 32:1$ contrast ratio for some patterns). As discussed in [18] the model's capability for velocity discrimination is also comparable to that of the human visual system.

This paper demonstrates the promise of computing optical flow using spatiotemporal filters. There are any number of related techniques using different filters, or using different rules for combining the filter outputs.

In [17] and [18] I show that the model may be used to simulate psychophysical data on velocity discrimination and on the coherence of sine-grating plaids. In [17] I compare the computations performed by the model to the stages of the visual motion pathway of the primate brain, and I suggest how the model might be used to simulate electrophysiological data.

For the most part, simulating physiological and psychophysical data merely demonstrates that the model is consistent with some of the experimental results on biological motion perception. The emphasis in future research will be to compare the predictions made by this model to those made by alternative image-flow models and to test those predictions with further experiments. Thus, the model may prove to be an interesting framework for future research in the psychophysics and neurophysiology of motion perception as well as in computer vision.

Acknowledgments

Special thanks to Ted Adelson for motivating this research and for providing the psychophysical data on coherence of sine-grating plaids, to Jack Nachmias and Grahame Smith for their detailed comments on earlier drafts of this paper, to Tony Movshon for his dialogs on physiology, to David Marimont for his mathematical insight, to Mark Turner for introducing me to Gabor filters, and to Lynn Quam for generating the Yosemite fly-through image sequence.

I particularly wish to thank Ruzena Bajcsy for her encouragement, and Sandy Pentland for his invaluable help and advice. As a result of their efforts, I have had the unique opportunity of doing this research both at the GRASP Laboratory at the University of Pennsylvania and at the Artificial Intelligence Center at SRI International.

This research is supported at the University of Pennsylvania by contracts ARO DAA6-29-84-k-0061, AFOSR 82-NM-299, NSF MCS-8219196-CER, NSF MCS 82-07294, AVRO DAABO7-84-K-FO77, and NIH 1-RO1-HL-29985-01; at SRI International by contracts NSF DCR-83-12766, DARPA MDA 903-83-C-0027, DARPA DACA 76-85,C-0004; and by the Systems Development Foundation.

References

1. S.T. Barnard and W.B. Thomson, "Disparity analysis of images," IEEE TRANS. PAMI-2(4), pp. 333–340, 1980.
2. B.K.P. Horn and B.G. Schunk, "Determining optical flow," ARTIFICIAL INTELLIGENCE, vol. 17; pp. 185–203, 1981.
3. J.K. Kearney and W.B. Thompson, "An error analysis of gradient-based methods for optical flow estimation," IEEE TRANS. PAMI-9(2), pp. 229–244, 1987.
4. H. Gafni and Y. Zeevi, "A model for separation of spatial and temporal information in the visual system," BIOLOGICAL CYBERNETICS, vol. 28; pp. 73–82, 1977.
5. H. Gafni and Y. Zeevi, "A model for processing of movement in the visual system," BIOLOGICAL CYBERNETICS, vol. 32; pp. 165–173, 1979.
6. M. Fahle and T. Poggio, "Visual hyperacuity: Spatiotemporal interpolation in human vision," PROC. R. SOC. (LONDON), vol. 213; pp. 451–477, 1981.
7. A.B. Watson and A.J. Ahumada, "A look at motion in the frequency domain," Tech. Rep. 84352, NASA-Ames Research Center, 1983.
8. A.B. Watson and A.J. Ahumada, "Model of human visual-motion sensing," J. OPT. SOC. AMER., vol. A2(2); pp. 322–342, 1985.
9. E.H. Adelson and J.R. Bergen, "Spatiotemporal energy models for the perception of motion," J. OPT. SOC. AMER., vol. A2(2), pp. 284–299, 1985.

10. J.P.H. van Santen and G. Sperling, "Elaborated reichardt detectors," J. OPT. SOC. AMER., vol. A2(2); pp. 300–321, 1985.
11. D.J. Fleet, The early processing of spatio-temporal visual information, Master's thesis, Dept. of Computer Science, Univ. of Toronto, 1984. (available as Tech. Report RBCV-TR-84-7.)
12. D.J. Fleet and A.D. Jepson, "A cascaded filter approach to the construction of velocity selective mechanisms," Tech. Report RBCV-TR-84-6, Dept. of Computer Science, Univ. Toronto, 1984.
13. E.C. Hildreth, "Computations underlying the measurement of visual motion," ARTIFICIAL INTELLIGENCE, vol. 23(3); pp. 309–355, 1984.
14. D. Gabor, "Theory of communication," J. IEE (LONDON), vol. 93; pp. 429–457, 1946.
15. J.G. Daugman, "Two-dimensional analysis of cortical receptive field profiles," VISION RESEARCH, vol. 20; pp. 846–856, 1980.
16. J.G. Daugman, "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," J. OF THE OPT. SOC. OF AMER., vol. A2(7); pp. 1160–1169, 1985.
17. David J. Heeger, "A model for the extraction of image flow," J. OPT. SOC. AMER., vol A4(8); pp. 1455–1471, 1987.
18. David J. Heeger, "Models for motion perception," Ph.D. thesis, CIS Department, Univ. of Pennsylvania, 1987. (Available as technical report MS-CIS-87-91.)
19. S.G. Mallat, "Scale change versus scale space representation," in PROC. FIRST INT. CONF. ON COMPUTER VISION, pp. 592–596, IEEE, London, 1987.
20. P. Burt, "Fast algorithms for estimating local image properties," COMPUTER VISION, GRAPHICS, AND IMAGE PROCESSING, vol. 21; pp. 368–382, 1983.
21. D.C. Burr and J. Ross, "Contrast sensitivity at high velocities," VISION RESEARCH, vol. 22; pp. 479–484, 1982.
22. P.E. Gill, W. Murray, and M.H. Wright, PRACTICAL OPTIMIZATION. Academic Press: New York, 1981.
23. R.A. Hummel and S.W. Zucker, "On the foundations of relaxing labelling processes," IEEE PAMI-5(3); pp. 267–287, 1983.
24. D. Terzopoulos, "Regularization of inverse visual problems involving discontinuities," IEEE PAMI-8(4); pp. 413–424, 1986.
25. T. Poggio, V. Torre, and C. Koch, "Computational vision and regularization theory," Nature, vol. 317(6035); pp. 314–319, 1985.
26. D. Marr and E. Hildreth, "Theory of edge detection," PROC. ROY. SOC. (LONDON), vol. B207; pp. 187–217, 1980.
27. D.E. Rummelhart and J.L. McClelland, eds., PARALLEL DISTRIBUTED PROCESSING: EXPLORATIONS IN THE MICROSTRUCTURE OF COGNITION. MIT Press: Cambridge, Mass., 1986.
28. B.B. Mandlebrot, THE FRACTAL GEOMETRY OF NATURE. W.H. Freeman: New York, 1983.
29. M.H. DeGroot, PROBABILITY AND STATISTICS. Addison-Wesley: Menlo Park, Calif., 1975.
30. J. Melsa and D. Cohn, DECISION AND ESTIMATION THEORY. McGraw-Hill: New York, 1978.
31. E.H. Adelson and J.A. Movshon, "Phenomenal coherence of moving visual patterns," NATURE, vol. 300(5892); pp. 523–525, 1982.
32. E.H. Adelson, Media-Technology Laboratory, MIT, personal communication.
33. M.P. doCarmo, DIFFERENTIAL GEOMETRY OF CURVES AND SURFACES. Prentice-Hall: Englewood Cliffs, N.J., 1976.
34. E.H. Adelson and E. Simonelli, "Orthogonal pyramid transfers for image coding," in PROC. SPIE, VISUAL COMMUN. and IMAGE PROC. II, pp. 50–58, Cambridge, MA, 1987.