Depth and Flow From Motion Energy

David J. Heeger

CIS Department
University of Pennsylvania
Philadelphia, Pa. 19104

AI Center
SRI International
Menlo Park, Ca. 94025

Abstract

This paper presents a model of motion perception that utilizes the output of motion-sensitive spatiotemporal filters. The power spectrum of a moving texture occupies a tilted plane in the spatiotemporal-frequency domain. The model uses 3-D (space-time) Gabor filters to sample this power spectrum. By combining the outputs of several such filters, the model estimates the velocity of the moving texture — without first computing component (or normal) velocity. A parallel implementation of the model encodes velocity as the peak in a distribution of velocity-sensitive units. For a fixed 3-D rigid-body motion, depth values parameterize a line through image-velocity space. The model estimates depth by finding the peak in the distribution of velocity-sensitive units lying along this line. In this way, depth and velocity are simultaneously extracted.

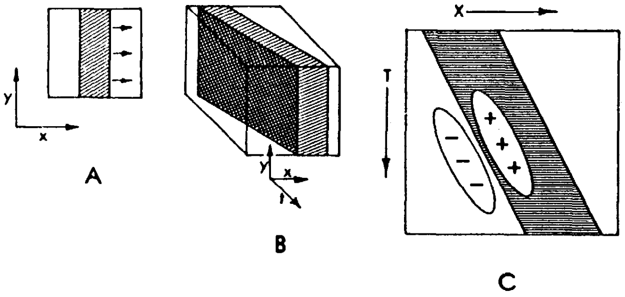

Figure 1: (from Adelson and Bergen [5]) Spatiotemporal Orientation. (a) a vertical bar translating to the right (b) the space-time cube for a vertical bar moving to the right. (c) an $x - t$ slice through the space-time cube.

1 Introduction

Image motion may be used to estimate both the motion of objects in 3-space and 3-D structure/depth. Motion information may also be utilized for percepetual organization, since regions that move in a "coherent" fashion may correspond to meaningful segments of the world around us.

Optical flow, a 2-D velocity vector for each small region of the visual field, is one representation of image motion. To compute a velocity vector locally for each region of an image, there must be motion information, i.e., changes in intensity over time, everywhere in the visual field. Depth may be recovered from image motion given prior knowledge of the 3-D rigid-body motion parameters. A dense depth map is recoverable only if there is motion information throughout the visual field.

Without texture, a perfectly smooth 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. This paper addresses the issues of extracting velocity and depth for each region of the visual field by taking advantage of the abundance of motion information in highly textured image sequences.

Most machine vision efforts that try to extract information from image motion utilize 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]. 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.

There have recently been several approaches to motion measurement based on spatiotemporal filtering [3,4,5,6,7] 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.

In the next section, I describe a family of motion-sensitive Gabor filters. The mathematics of motion in the spatiotemporal-frequency domain, discussed in Section 3, is used in Section 4 to derive a model for extracting image velocity from the outputs of these filters. Section 5 presents a parallel implementation of the model that operates as a collection of velocity-sensitive mechanisms. Section 6 discusses how depth is encoded by these velocity-sensitive mechanisms given prior knowledge of the 3-D rigid-body motion parameters. Section 7 discusses the model's outputs for strongly oriented patterns that suffer from the aperture problem and suggests some future directions for this research.

2 Motion-Sensitive Filters

The concept of orientation in space-time is well explained by Adelson and Bergen [5]. Figure 1 shows the space-time cube for a vertical bar moving to the right. The slope of the edges in an $x - t$ slice through this cube equals the horizontal component of the bar's velocity. The most successful technique for estimating edge orientation has been based on linear systems theory, e.g., as depicted in Figure 1(c) convolution with linear filters.

A 1-D Gabor filter [8] is simply a sine wave multiplied by a

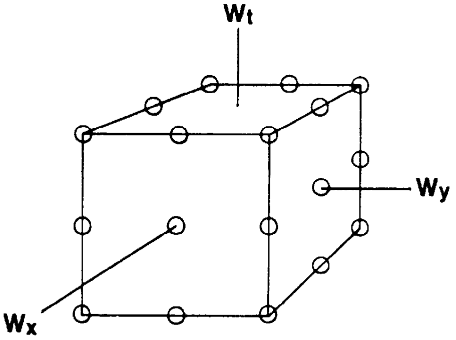

Figure 2: The power spectra of the 12 motion-sensitive, Gabor-energy filters are positioned in pairs on a cube in the spatiotemporal-frequency domain.

Gaussian window:

$$Gabor(t) = G(t)\sin(\omega t + t_0) \quad (1)$$

where $G(t)$ is a Gaussian. The power spectrum of a sine wave, $\sin(\omega t)$, 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 a pair of Gaussians centered at ω and $-\omega$ in the frequency domain, i.e., it is an oriented bandpass filter. Thus, a Gabor function is localized in a Gaussian window in the space (or time) domain, and it is localized in a pair of Gaussian windows in the frequency domain.

Similarly, an example of a 3-D Gabor filter is

$$Gabor_s(x, y, t) = G(x, y, t)\sin(\omega_{x_0}x + \omega_{y_0}y + \omega_{t_0}t) \quad (2)$$

where $G(x, y, t)$ is a 3-D Gaussian. This function looks like a stack of plates with small plates on the top and bottom of the stack and the largest plate in the middle of the stack. The stack can be tilted in any orientation in space-time. The power spectrum of Equation (2) is a pair of 3-D Gaussians.

The model uses a family of Gabor-energy filters, each of which is the squared sum of the response of a sine- and cosine-phase Gabor filter, giving an output that is invariant to the phase of the signal. The presenet implementation uses 12 filters, each tuned to the same range of spatial frequencies but to different spatiotemporal orientations. Their power spectra are positioned in pairs on a cube in the spatiotemporal-frequency domain (Figure 2): four of them are at the eight corners of the cube, two at the centers of the four sides, and six at the midpoints of the twelve edges. For example, the filter that is most sensitive to down-left motion has the following power spectrum:

$$G(\omega_x - \omega_0, \omega_y - \omega_0, \omega_t - \omega_0) + G(\omega_x + \omega_0, \omega_y + \omega_0, \omega_t + \omega_0) \quad (3)$$

where $G(\omega_x, \omega_y, \omega_t)$ is a 3-D Gaussian, ω_x, ω_y, and ω_t are spatial and temporal frequencies, and ω_0 specifies the tuning frequency at which the filter achieves its peak output. Gabor filters can be built from separable components, thereby greatly increasing the efficiency of the computations.

3 Motion in the Frequency Domain

Now let us review some properties of image motion, first presented by Watson and Ahumada [3,4], that are evident in the spatiotemporal-frequency domain. I shall begin by describing 1-D motion in terms of spatial and temporal frequencies, and observe that the power spectrum of a moving 1-D signal occupies a line in the frequency domain. Analogously, the power spectrum of a translating 2-D texture occupies a tilted plane in the frequency domain.

3.1 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:

$$\vec{v} = \omega_t / \omega_x \quad (4)$$

Now consider a 1-D signal, moving with a given velocity $\vec{v}$, that has many spatial-frequency components. Each such component ω_x has a temporal frequency of $\omega_{t_1} = \omega_x\vec{v}$, while each spatial-frequency component $2\omega_x$ has twice the temporal frequency $\omega_{t_2} = 2\omega_x\vec{v}$. 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 $\vec{v}$.

3.2 Two-Dimensional Motion

Analogously, 2-D patterns (textures) translating in the image plane occupy a plane in the spatiotemporal-frequency domain:

$$\omega_t = u\omega_x + v\omega_y \quad (5)$$

where $\vec{v} = (u, v)$ is the velocity of the pattern. For example, a region of a translating random-dot field or a translating field of normally distributed intensity values fills a plane in the frequency domain uniformly, i.e., the power of such an image sequence is a constant within that plane and zero outside of it (a dot or impulse, and a normally distributed random texture have equal power at all spatial frequencies). Because the motion of a small region of an image is approximated by translation in the image plane, the velocity of such a region may be computed in the Fourier domain by finding the plane in which all the power resides. The motion-sensitive spatiotemporal Gabor filters introduced earlier are an efficient way of "sampling" these power spectra (image a plane passing through the center of Figure 2).

4 Motion Energy to Extract Image Flow

Spatiotemporal bandpass filters like Gabor energy filters and those filters discussed in previous papers [4,5,7] are *not* velocity-selective mechanisms, but rather are tuned to particular spatiotemporal frequencies. Consider, for example, two sine gratings with the same velocity but different spatial frequencies (i.e., they have proportionately different temporal frequencies as well). A spatiotemporal bandpass filter will respond differently to these two signals even though they have the same velocity.

Motion-energy filters were first presented by Adelson and Bergen [5]. Watson and Ahumada [3,4] first explained that a moving texture occupies a tilted plane in the frequency domain.

This section combines these two concepts and derives a technique that uses a family of motion-energy filters to extract velocity. The role of the filters is to sample the power spectrum of the moving texture. By combining the outputs of several filters, the model estimates the slope of the plane (i.e., the velocity of the moving texture) directly from the motion energies without first computing component (or normal) velocity.

4.1 Extracting Image Flow

First, I derive equations for Gabor energy resulting from motion of random textures or random-dot fields. Based on these equations, I then formulate a least-squares estimate of velocity.

Parseval's theorem states that the integral of squared values over the spatial domain is proportional to the integral of the squared Fourier components over the frequency domain. 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 within the original signal over this range of frequencies.

The power spectrum of a normally distributed random texture (or random-dot field) fills a plane in the spatiotemporal-frequency domain uniformly (Equation 5). The power spectrum of a Gabor filter is a 3-D Gaussian. By Parseval's theorem Gabor energy, in response to a moving random texture, is proportional to the integral of the product of a Gaussian and a plane. For example, the formula for the response of the Gabor-energy filter most sensitive to down-left motion is derived by substituting Equation (5) for ω_t in Equation (3), and integrating over the frequency domain:

$$2k \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-c[(\omega_x-\omega_0)^2+(\omega_y-\omega_0)+(u\omega_x+v\omega_y-\omega_0)^2]} d\omega_x d\omega_y \tag{6}$$

where k is a scale factor and c depends on the bandwidth of the filter. This integral evaluates to

$$t_1(u,v,k)e^{-t_2(u,v)(u^2+v^2+2uv-2u-2v+1)} \tag{7}$$

where

$$t_1(u,v,k) = \frac{2k\pi}{c\sqrt{u^2+v^2+1}} \tag{8}$$

$$t_2(u,v) = \frac{c\omega_0^2}{u^2+v^2+1} \tag{9}$$

Similar equations can be derived for all twelve Gabor-energy filters, thus yielding a system of twelve equations in the three unknowns (u, v, k). The factor $t_1(u, v, k)$ appears in each of these twelve equations. We can eliminate it by dividing each equation by the sum of all twelve of them resulting in a system of equations that depend only on u and v. These equations predict the output of Gabor energy filters due to local translation. The predicted energies are exact for a pattern with a flat power spectrum (e.g., random-dot or random-noise fields).

Now let us formulate the "best" choice for u and v as a least-squares estimate for this nonlinear system of equations. Let O_i' ($i = 1$ to 12) be the twelve observed Gabor energies. Let

$$O_i = \frac{O_i'}{\sum O_i'} \tag{10}$$

Also, let $\mathcal{R}_i(u,v)$ be the twelve predicted energies as in Equation 7. The least-squares estimate of $\vec{v} = (u, v)$ minimizes

$$F(u,v) = \sum_{i=1}^{12} [O_i - \mathcal{R}_i(u,v)]^2 \tag{11}$$

There are standard numerical methods for estimating $\vec{v} = (u, v)$ to minimize Equation (11), e.g., the Gauss-Newton gradient-descent method. In Section 5, I describe a parallel approach for finding this minimum.

Since the system of equations is overconstrained (12 equations in two unknowns), the residuals $[O_i - \mathcal{R}_i(u,v)]$ may be used to compute a confidence index for the solution. I am investigating the possibility of using, as potential confidence indices, the sum of the squares of the residuals, the variance of the residuals divided by their mean, as well as the sharpness/width/curvature of the minima. Computations that use the flow vectors as inputs, e.g., for estimating 3-D structure and motion, could weight each vector according to its confidence.

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

1. Convolve each image in the image sequence with a center-surround filter to remove the dc and lowest spatial frequencies.
2. Compute motion energy as the squared sum of the sine- and cosine-phase Gabor filters.
3. Smooth the resulting motion energies and divide each by the sum of all twelve.
4. Find the best choice of u and v to minimize Equation (11), e.g., by employing the Gauss-Newton method or the parallel approach presented in Section 5.

4.2 Results

The system of equations discussed above are precisely correct only for images with a flat power spectrum, but the model has been succesfully tested for a variety of computer-generated and natural images.

Figure 4 shows the flow field extracted from a random-dot image sequence of a sphere rotating about an axis through its center, in front of a stationary background.

Figure 6 shows the flow field extracted from a computer-generated image sequence flying through Yosemite valley. Each frame of the sequence was generated by mapping an aerial photograph onto a digital-terrain map (alititude map). The observer is moving toward the rightward horizon. The clouds in the background were generated with fractals (see recent SIGGRAPH conference proceedings) and move to the right while changing their shape over time.

One way to test the accuracy of the flow field is to use it to compensate for the image motion by shifting each local region of each image in the sequence opposite to the extracted flow. This should result in a new image sequence that is motionless, i.e., the intensity at each pixel should not change over time. Figure 6(d) shows the variance in the intensity at each pixel after compensating for the image motion in this way. The variance is very small for most of the landscape region indicating that the extracted flow field is quite accurate. Most of the high variance regions can be attributed to occlusions — more of the landscape comes into view over time thereby changing the grey levels. The extracted flow field is erroneous in two regions: (1) the cloud motion is blurred over the landscape near the horizon; (2) some of the flow vectors near the bottom of the image are incorrect probably due to temporal aliasing and/or the aperture problem.

The clouds change their shape over time while moving rightward. Compensating for the extracted rightward flow yields stationary clouds that still change their shape over time resulting in the high variance at the top of 6(d). The procedure of estimating image flow and then compensating for it has allowed us to separate the cloud region from the rigidly moving landscape. A fractal-based model for the recognition of nonrigid, turbulent flows is presented by Heeger and Pentland [9].

5 A Parallel Implementation

The last step in the above algorithm is to find the minimum of a two-parameter function. One way to locate this minimum is to evaluate the function in parallel at a number of points (say, on a fixed square grid), and pick the smallest result. 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-sensitive. Local image velocity may be encoded as the minimum in the distribution of the outputs of a number of such velocity-sensitive units, each tuned to a different $\vec{v}$. The units tuned to velocities close to the true velocity will have relatively small outputs (small error), while those tuned to velocities that deviate substantially from the true velocity will have large outputs (large error).

For a fixed velocity, the predicted motion energies from the system of equations discussed above (e.g., Equation 7) are fixed constants — denote them by w_i. Thus, we may rewrite Equation (11) for a fixed $\vec{v}$ as

$$F_j = \sum_{i=1}^{12} [O_i - w_{ij}]^2 \tag{12}$$

where F_j is the response of a single velocity-sensitive unit and w_{ij} are constant weights, each corresponding to one of the motion energies for a particular velocity. A mechanism that computes a velocity-tuned output from the motion-energy measurements performs the following simple operations:

1. Divides each motion energy by the sum or average of all twelve motion energies.
2. Subtracts a constant from each of the results of Step (1).
3. Sums the squares of the results of Step (2).

An example of the outputs of these velocity-tuned units is shown in Figure 3(a) that displays a map of velocity space, with each point corresponding to a different velocity — for example, $\vec{v} = (0,0)$ is at the center of each image, $\vec{v} = (2,2)$ at the top-right corner. The brightness at each point is the output of a velocity-sensitive unit tuned to that velocity, therefore, the minimum in the distribution of responses corresponds to the velocity extracted by the model.

6 Motion Energy to Recover Depth

This section presents a technique for recovering a dense depth map from the velocity-sentive units discussed above given prior knowledge of the 3-D rigid-body motion parameters. There are a number of situations in which we have prior estimates of these parameters — for example, they may have been estimated from sparse image motion data or from other sensors, e.g., a mobile robot equipped with an inertial guidance system moving through a static environment.

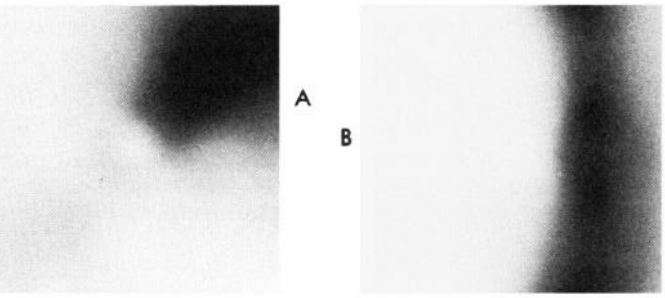

Figure 3: (a) velocity-sensitive units responding to a moving random-dot field. The minimum in the distribution of responses corresponds to the velocity extracted by the model. (b) velocity-sensitive units responding to a single moving sinusoidal grating; the aperture problem is evident as there is a trough of minima.

First, I show that for a fixed 3-D rigid-body motion, depth values parameterize a line through image-velocity space. Each point on the surface of a rigidly moving surface patch has an associated velocity vector, $\vec{V} = (V_x, V_y, V_z)$, given by

$$\vec{V} = \vec{\Omega} \times \vec{R} + \vec{T} \tag{13}$$

where $\vec{\Omega} = (\omega_x, \omega_y, \omega_z)$ are the rotational components of the rigid motion, $\vec{T} = (t_x, t_y, t_z)$ are the translational components of the rigid motion, and $\vec{R} = [x, y, z(x,y)]$ is the 3-D position of each point on the rigidly moving surface [10].

Under orthographic projection, image velocity, $\vec{v} = (u, v) = (V_x, V_y)$. Thus, we may rewrite Equation 13 as

$$\begin{aligned} u &= \omega_y z - \omega_z y + t_x & (14) \\ v &= -\omega_x z + \omega_z x + t_y & (15) \end{aligned}$$

For fixed $\vec{\Omega}$, $\vec{T}$, x, and y this is the parametric form of the equation of a line — changing z corresponds to sliding along this line.

Now, I explain how to recover depth from the collection of velocity-sensitive units presented in the preceding section. Since depth parameterizes a line through velocity space, the local depth estimate corresponds to the minimum in the distribution of the outputs of those velocity-sensitive units that lie along this line. We need only locate the minimum along this line.

Formally, we substitute Equations 14 and 15 for u and v in Equation 11 giving

$$\begin{aligned} F'(z) &= F[u(z), v(z)] \\ &= \sum_{i=1}^{12} [O_i - \mathcal{R}_i(u(z), v(z))]^2 & (16) \end{aligned}$$

and pick z that minimizes $F'(z)$. In this way, depth and velocity are simultaneously extracted from motion energy.

A depth map recovered from the random-dot sphere sequence discussed above is shown in Figure 5. The technique may be extended to perspective projection by approximating with locally orthographic projection.

7 The Aperture Problem

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 [11]): there is not enough information in the image sequence to disambiguate the true motion of the pattern. 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.

Figure 3(b) shows the outputs of velocity-sensitive units for a moving sinusoidal grating. The aperture problem is evident as there is a trough of minima. Preliminary investigation indicates that the velocity extracted by the model defaults to normal flow for such strongly oriented patterns. Depth may be recovered even for local regions that suffer from the aperture problem — though the velocity-sensitive units output a trough of minima, there will be a *single* minimum along a line passing across the trough. Future research will study how the velocity and depth estimates vary for patterns that are more and more strongly oriented.

8 Summary

This paper presents a model for extracting velocity and depth at each location in the visual field by taking advantage of the abundance of motion information in highly textured image sequences. The power spectrum of a moving texture occupies a tilted plane in the spatiotemporal-frequency domain. The model uses 3-D (space-time) Gabor filters to sample this power spectrum and estimate the slope of the plane (i.e., the velocity of the moving texture) without first computing component velocity. A parallel implementation of the model encodes velocity as the peak in a distribution of velocity-sensitive units. For a fixed 3-D rigid-body motion, depth values parameterize a line through image-velocity space. The model estimates depth by finding the peak in the distribution of velocity-sensitive units lying along this line. In this way, depth and velocity are simultaneously extracted from motion energy.

Acknowledgments

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 DACA76-85-C-0004, NSF DCR-8519283, and by the Systems Development Foundation.

Special thanks to Ted Adelson for motivating this research, to Sandy Pentland for his invaluable help and advice, to David Marimont for his mathematical insight, and to Mark Turner for introducing me to Gabor filters. I also want to thank Jack Nachmias, Grahame Smith, and Dorothy Jameson for commenting on earlier drafts of this paper, and especially Lynn Quam for generating the Yosemite fly-through image sequence.

References

[1] Barnard, S.T., and Thomson, W.B. (1980) Disparity analysis of images, *IEEE Pattern Analysis and Machine Intelligence, 2*, No. 4, pp. 333-340.

[2] Horn, B.K.P., and Schunk, B.G. (1981) Determining optical flow, *Artificial Intelligence, 17*, pp. 185-203.

[3] Watson, A.B., and Ahumada, A.J. (1983) A look at motion in the frequency domain, NASA technical memorandum 84352.

[4] Watson, A.B., and Ahumada, A.J. (1985) Model of human visual-motion sensing, *Journal of the Optical Society of America, 2*, No. 2, pp. 322-342.

[5] Adelson, E.A., and Bergen, J.R. (1985) Spatiotemporal energy models for the perception of motion, *Journal of the Optical Society of America, 2*, No. 2, pp. 284-299.

[6] van Santen, J.P.H., and Sperling, G. (1985) Elaborated Reichardt detectors, *Journal of the Optical Society of America, 2*, No. 2, pp. 300-321.

[7] Fleet, D.J. (1984) The early processing of spatio-temporal visual information, MSc. Thesis (available as RBCV-TR-84-7, Department of Computer Science, University of Toronto).

[8] Daugman, J.G. (1985) Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters *Journal of the Optical Society of America, 2*, No. 7, pp. 1160-1169.

[9] Heeger, D.J., and Pentland, A.P. (1986) Seeing Structure Through Chaos, *proceedings of the IEEE Workshop on Motion: Representation and Analysis*, Kiawah Island Resort, Charleston, North Carolina, p. 131-136.

[10] Hoffman, D.D. (1982) Inferring local surface orientation from motion fields, *Journal of the Optical Society of America, 72*, No. 7, pp. 888-892.

[11] Hildreth, E.C. (1984) Computations Underlying the Measurement of Visual Motion, *Artificial Intelligence, 23*, No. 3, pp. 309-355.

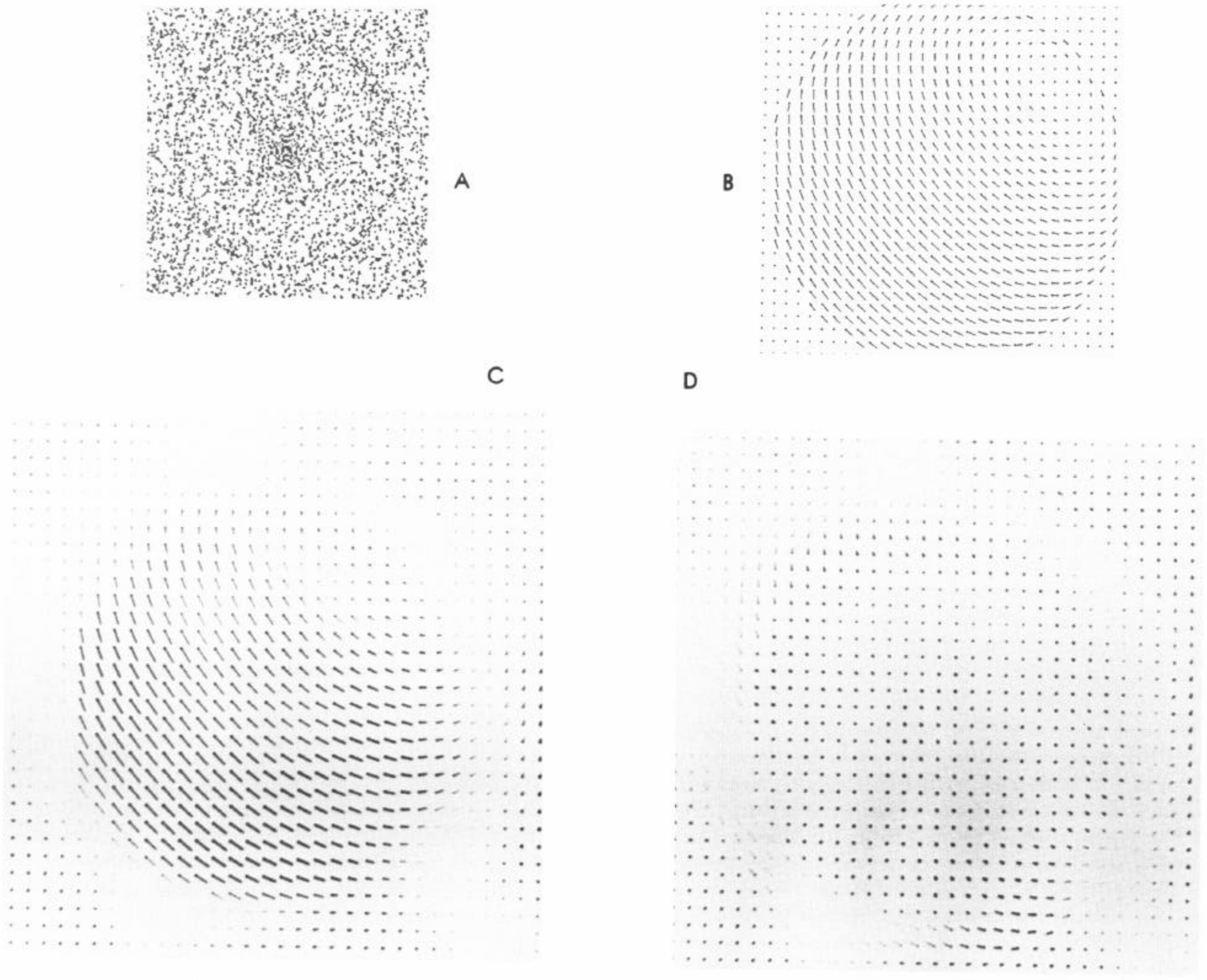

Figure 4: A rotating random-dot sphere. (a) one frame from the image sequence. (b) actual flow field. (c) extracted flow — the brightness of each vector is weighted by a confidence index computed from the residuals of the least squares. (d) difference between (b) and (c).

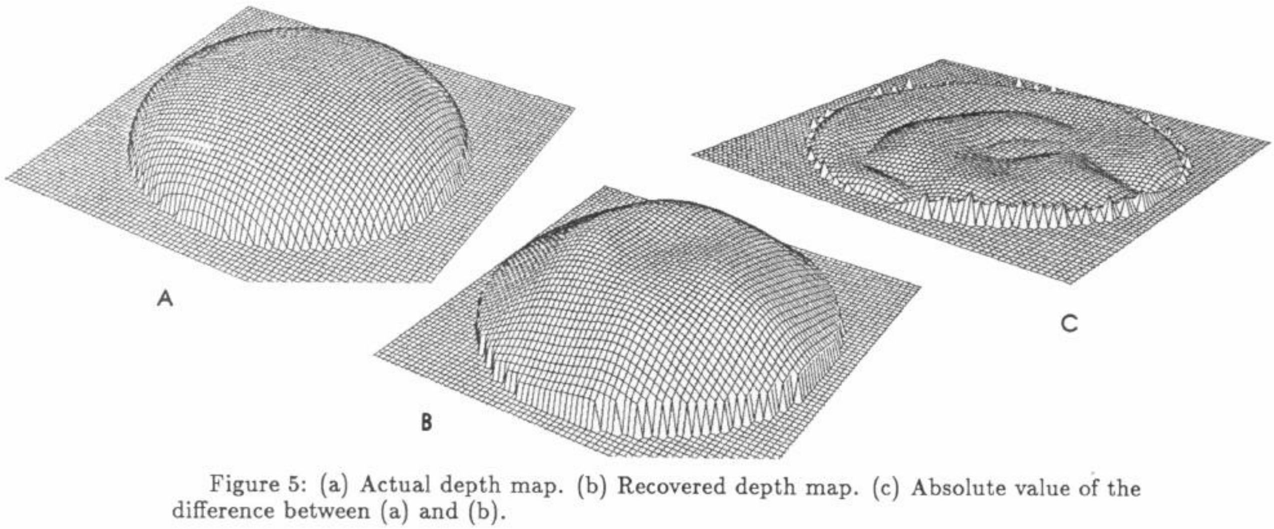

Figure 5: (a) Actual depth map. (b) Recovered depth map. (c) Absolute value of the difference between (a) and (b).

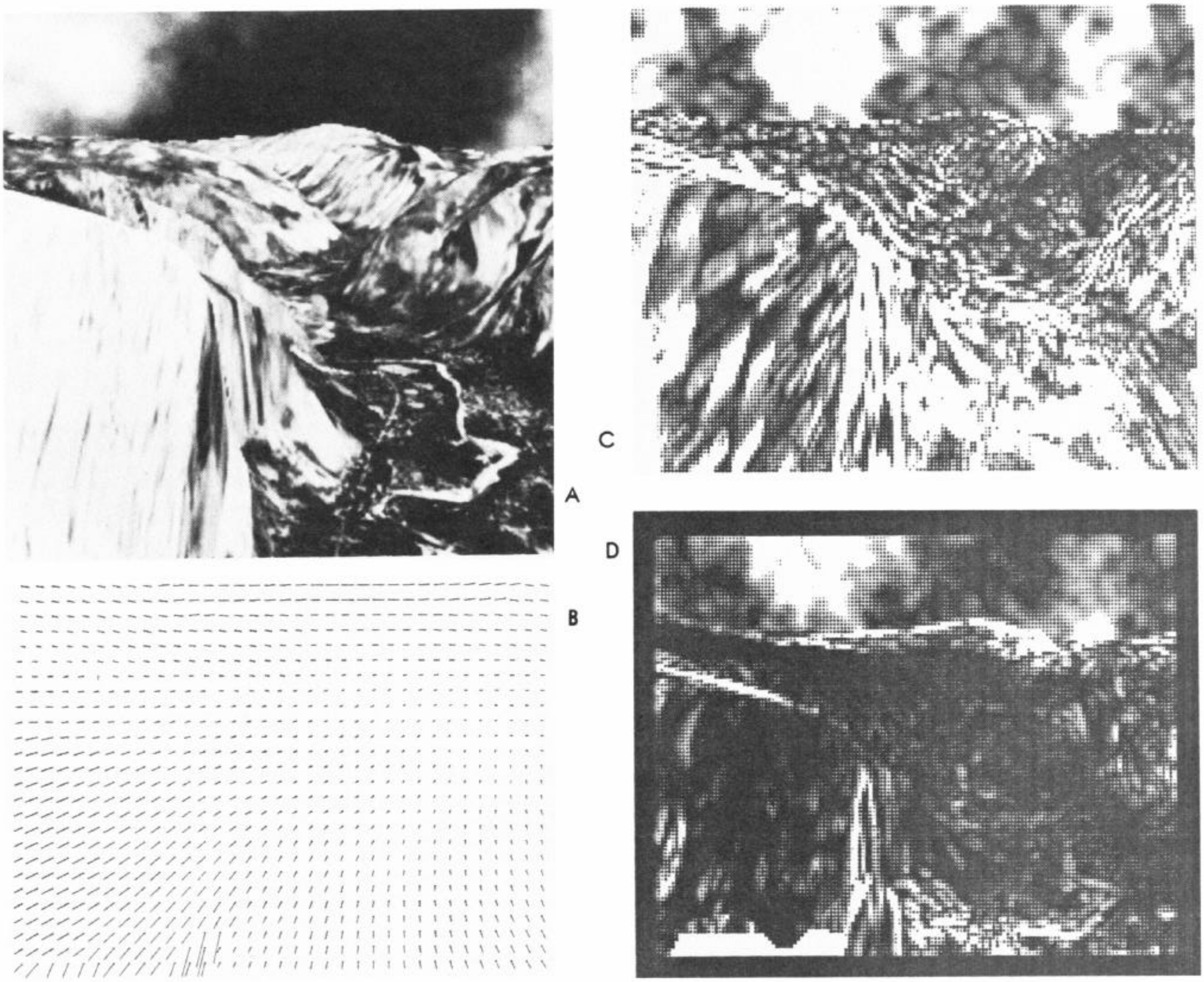

Figure 6: (a) one frame of an image sequence flying through Yosemite valley. (b) extracted flow field. (c) the variance of image intensity over time at each pixel of the original image sequence. (d) variance after compensating for the image motion.