REGION-BASED OPTICAL FLOW ESTIMATION

Chiou-Shann Fuh and *Petros Maragos*

Division of Applied Sciences, Harvard University, Cambridge, MA 02138.

ABSTRACT:[1] This paper presents a correspondence method to determining optical flow where the primitive motion tokens to be matched between consecutive time frames are regions. The computation of optical flow consists of three stages: region extraction, region matching, and optical flow smoothing. For *region extraction*, in each image frame the regions are extracted either from the sign of the $\nabla^2 G_\sigma * I$ bandpass operator, or by thresholding the output of morphological image transformations for peak/valley detection. For *region matching*, a general correspondence approach is applied to region tokens by using an *affinity measure* based on region features. Optical flow is then identified as the spatial vector displacements among *centroids* of corresponding regions. The computation is completed by *smoothing* the initial optical flow, where the sparse velocity data are either smoothed with a vector median filter or interpolated to obtain dense velocity estimates by using a motion-coherence regularization.

The proposed region-based method for optical flow is simple, computationally efficient, and (as our experiments on real images indicate) more robust than iterative gradient methods, especially for medium-range motion.

1 Introduction

Motion analysis is a major task of computer vision systems. It deals with the general problems of recovering the 3-D motion and structure of visible surfaces given a discrete-time sequence $I(x, y, t)$ of 2-D intensity images. An earlier and some recent reviews on this topic include [1,2,3]. When objects are being imaged through a camera (or a human retina) moving relative to the objects, the apparent motion of brightness patterns is called *optical flow*. It is represented by a vector field (v_x, v_y), where v_x, v_y denote velocities in x, y direction. The optical flow is an ambiguous concept and is not generally equal to the true 2-D velocity field except for some special cases. Nevertheless, due to its accessibility and the rich information it contains for recovering the motion of 3-D rigid bodies, the detection of optical flow is very important in motion analysis. For instance, there are many approaches to 3-D motion and structure recovery which assume that 2-D velocity data (sparse or dense) have been obtained in advance; examples include [4,5,6,7].

The major approaches to computing optical flow (or velocity fields) can be classified as either using *gradient* models or *correspondence* of motion tokens. The gradient models are based on some relationships among the image spatial and temporal derivatives. For example, Horn and Schunck [5] used the optical flow constraint $\frac{dI}{dt} = 0 \Longleftrightarrow \frac{\partial I}{\partial x} v_x + \frac{\partial I}{\partial y} v_y = -\frac{\partial I}{\partial t}$, whereas in [8] the constraint $\frac{d(\nabla I)}{dt} = 0$ was proposed. Although gradient models are analytically more tractable, lead to iterative local image operations and can provide spatially dense velocity estimates, they are computationally intensive, apply only to short-range motion, and are highly susceptible to noise. By contrast, the correspondence methods are more immune to noise and can be also applied to medium- or long-range motion. They are based on matching and tracking over time simple tokens (sets of elementary image features) in one frame with their counterparts on the same object in subsequent time frames. Their main difficulty lies in solving the motion correspondence problem. When correspondence is solved the sparse velocity estimates at tokens are equated to the spatial displacement vectors between corresponding tokens. Aggarwal *et al.* [9] discussed several procedures to solve the motion correspondence problem based on both ikonic and structural representations of image parts.

This paper presents a correspondence approach to determining optical flow where the simple tokens are *regions*. [Alternatively, the tokens could be *intensity points*, whose matching could be accomplished via correlation, or *lines* (e.g., edges represented by zero-crossing contours [10]).] We agree with Ullman's conclusions [11] who, based on perceptual experiments, views the gray-level matching as insufficient to solve the correspondence problem. Further, we view the region matching as more robust than edge matching, because noise perturbs the coherence of a region less than its boundaries (edges). This was demonstrated by Nishihara [12] who solved the correspondence problem for binocular stereo by cross-correlating the binary regions (sign areas) bounded from the zero-crossing contours of the band-pass filtered images $\nabla^2 G_\sigma * I$. (∇^2 is the operator $\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$, and $G_\sigma(x, y) = \frac{e^{-(x^2+y^2)/2\sigma^2}}{2\pi\sigma^2}$.) Our proposed estimation of optical flow consists of three stages: 1) *Region extraction*, where the image sequence $I(x, y, t)$ is either convolved with $\nabla^2 G_\sigma$ and regions are identified as the sign representation of these convolutions, or the image sequence un-

[1]This work was supported by the National Science Foundation under Grant MIPS-86-58150 with matching funds from Bellcore, Xerox and SUN, and in part by the ARO under Grant DAALO3-86-K-0171.

CH2752-4/89/0000/0130$01.00 © 1989 IEEE

dergoes morphological transformations [13,14,15] that extract peaks/valleys and regions are identified as binarized versions of these peak/valley signals. 2) *Region matching*, where Ullman's general correspondence theory is applied to region tokens by using an *affinity measure* for matching. Optical flow is then identified as the spatial displacements among *centroids* of corresponding regions. 3) *Optical flow smoothing*, where the sparse velocity data are either smoothed with a vector median filter or interpolated to obtain dense velocity estimates by using a motion-coherence regularization method developed in [16].

2 Region Extraction

The first step in our region-based approach to determining optical flow is to find the region tokens in consecutive image frames. We define as "regions" connected sets of pixels (x, y), which are subsets of the spatial image domain and correspond to subparts of the moving object(s). We have experimented with two different methods to extract regions:

(A) Sign representation of $\nabla^2 G_\sigma * I$: The Marr-Hildreth [17] edge detection operator $\nabla^2 G_\sigma * I$ is applied first. For each image frame, the set of image pixels at which this edge signal has a positive sign identifies the collection of *positive regions*, and its set complement yields the *negative regions*. There is a trade-off in selecting a value for the scale parameter σ. For large σ, the regions are large, and their number per frame is small. To achieve dense optical flow, small values of σ are preferred. On the other hand, to achieve a matching that is more robust and less susceptible to noise, a larger σ is preferred. In our experiments we implemented the $\nabla^2 G_\sigma$ operator as the difference of two Gaussians, one (the excitatory) with $\sigma = 1$ and another (the inhibitory) with $\sigma = 3$; the size of the convolution kernel was 11 pixels in each direction. Figs. 1c and 2c,d show examples of extracted regions. The region extraction process is completed by *labeling connected components*, where at each time t_k, each positive (or negative) region has been identified as a 4-connected component of the binary image representing the positive (or negative) sign of $\nabla^2 G_\sigma * I(x, y, t_k)$.

(B) Binarized Peak/Valley Detection Transformations: If I is the intensity image at some time frame, two transformations [13,14,15] that can extract its peaks and valleys, respectively, are:

$$\begin{aligned} \text{Peak}(I) &= I - (I \circ B) \geq 0 \\ \text{Valley}(I) &= (I \bullet B) - I \geq 0 \end{aligned},$$

where $I \circ B$ and $I \bullet B$ are, respectively, the morphological opening and closing [14,15] of I by a structuring element B (a window of pixels). We produce binary *peak regions* by thresholding at level T, i.e., by setting all pixels (x, y) at which $[\text{Peak}(I)](x, y) \geq T$ equal to 1 and 0 elsewhere. Similarly, the binary *valley regions* result from thresholding the valley signal $\text{Valley}(I)$ at T (an adjustable level). The shape and size of B controls the shape and maximum size of extracted regions.

In general, the regions extracted via morphological peak-valley detectors yield a more faithful representation of the binary shapes of various features in the image, than the regions extracted via the $\nabla^2 G_\sigma * I$ process which tends to blur the regions' boundaries.

3 Region Matching

The matching algorithm proposed herein is guided by Ullman's general correspondence principles and attempts to match correspondence tokens via an affinity measure. The two aspects with respect to which our algorithm is different are the nature of the correspondence tokens (i.e., we use regions as tokens as opposed to line segments) and the large set of features for each token that affect the affinity measure. Specifically, let R_i and R_j be two regions extracted at two consecutive time frames (at $t = t_k, t_{k+1}$), and let $\vec{c}_i$, $\vec{c}_j$ denote their centroids. For each region to be considered as a legitimate correspondence token, its area $A(R_i)$ must be larger than A_{min}. An alternative approach (that we tried with similar performance) is to morphologically open all regions by a structuring element C whose area is A_{min}. This *cleans* the regions by eliminating all regions inside which C does not fit. The affinity measure between R_i and R_j depends on the following features:

1) Distance between centroids: $||\vec{d}_{ij} = \vec{c}_i - \vec{c}_j||$.

2) Signs (positive or negative) of regions if they resulted from the $\nabla^2 G_\sigma * I$ approach, or their peak vs. valley identities if the regions resulted from the morphological feature detection process.

3) Area difference between regions: $AD_{ij} = A(R_i) - A(R_j)$.

4) Difference between average intensities.

Further, we set $||\vec{d}_{ij}|| = +\infty$ if either the x- or y-component of $\vec{d}_{ij}$ exceeds L pixels, and we set $AD_{ij} = +\infty$ if $|AD_{ij}| > P \cdot A(R_i)$, where $0 < P < 1$. Clearly, the fixed numbers A_{min}, L, and P are *control parameters* for the correspondence process. Specifically, A_{min} controls the robustness of a region to be considered as a legitimate correspondence token; i.e., all regions whose area $\leq A_{min}$ are viewed as either noise or unreliable tokens and hence are not matched. L controls the *range* of correspondence. Region R_i may be matched with R_j only if the centroid of R_j lies inside a square window of $(2L+1) \times (2L+1)$ pixels centered at the centroid of R_i, and their $\nabla^2 G_\sigma * I$ signs or their peak/valley identities are the same. P determines the maximum percentage of area difference between two regions above which a match is impossible. In our experiments the control parameters were set to $A_{min} = 9$ (pixels), $L = 15$ (pixels), and $P = 0.3$.

The above criteria may result in a situation where, if there are no candidates, there is no match for a particular region. If there are some matching candidates, however, the region that has the closest average intensity is selected to be the correct match. This rule is conceptually related to the optical flow constraint $\frac{dI}{dt} = 0$ in [5]. Figs. 1e,f and 2e,f show examples of optical flow detected when the mo-

tion was *translation* along the z-axis or *rotation*. In these experiments velocity estimates were obtained up to 10-15 pixels in x and y directions.

Clearly, each successful match of two regions in two consecutive image frames yields a spatial displacement vector (d_x, d_y) among the two region centroids. Estimating the velocity of a region's centroid by bringing it into correspondence with another region's centroid is not an arbitrary choice. The classical mechanics theory dictates that, with respect to an external force or torque, the motion of a rigid body can be represented by the motion of its centroid. Thus, we implicitly assume that each region is a small patch of a rigid body. We do *not* assume, however, that over a whole region the velocity remains constant. We simply estimate it only at the centroid. Finally, the average velocity is equal to $(v_x, v_y) = (d_x, d_y)/(t_{k+1} - t_k)$. Henceforth, we assume a uniform sampling of image frames in time and set $t_{k+1} - t_k = 1$, which amounts to equating pixel displacements with velocities.

4 Smoothing the Optical Flow

4.1 Vector Median Filtering

Although most of the region matches appear to be accurate and robust, there may be a few mismatches. We view the latter as noise on the estimated optical flow. Then a question naturally arises of how to smooth the optical flow. We exclude the smoothing via linear filtering (e.g., local averaging) because linear smoothing filters have the well-known tendency to blur and shift sharp discontinuities in signals. In the case of optical flow these sharp discontinuities may indicate object boundaries and, hence, must be preserved. Therefore, we choose median filtering to smooth optical flow, because, median filters are nonlinear smoothers that can eliminate outliers from the original data while preserving abrupt discontinuities (e.g., edges) in the signal. Median filtering has been applied extensively to scalar real-valued signals. Here we define a *vector median filter* to operate on the vector-valued optical flow. Let $\vec{v}_i = (v_{x,i}, v_{y,i})$, $i = 1, 2, \ldots, n$, be the estimated velocities at various centroids around and including a centroid $\vec{c}$. Due to the relative sparseness of centroids, the estimates are found by searching inside windows centered at $\vec{c}$ and whose size increases (but does not exceed twice the maximum window of Section 3) until n velocity estimates are found. Then the output smoothed velocity at $\vec{c}$ is

$$\text{med}_i\{\vec{v}_i\} = (\text{med}_i\{v_{x,i}\}, \text{med}_i\{v_{y,i}\}) .$$

The scalar medians $\text{med}_i\{v_{x,i}\}$ are computed by sorting the n values $v_{x,i}$ and picking the middle value. (If n is even, the median is set equal to the average of the two middle values.) The parameter n (set equal to 17 in our experiments) controls the degree of smoothing. Figs. 1g,2g show that this vector median filtering is effective in smoothing the optical flow.

4.2 Interpolation by Motion-Coherence

So far, the optical flow is being estimated at most region centroids but not at every pixel. However, there are many approaches to recovery of 3-D motion and 3-D structure that require a dense optical flow field. Our region-based approach can be extended to yield dense velocity estimates by using *interpolation*. As a by-product of the interpolation process, we can also achieve *smoothing*. In [8], many interpolation approaches have been reviewed in the context of regularization of ill-posed problems. In a similar spirit, Yuille and Grzywacz [16] proposed a motion-coherence regularization procedure to smooth motion fields. Their result can also be viewed as an interpolation process. Specifically, let $\vec{V}_i$ be the velocity estimates at centroids $\vec{c}_i$ by using our region-based method. Let also $\vec{r}$ denote any vector on the image plane and let $\vec{v}(\vec{r})$ be the velocity to be provided via the regularization process. Yuille and Grzywacz find the unknown $\vec{v}(\vec{r})$ by minimizing the functional

$$E = \sum_i ||\vec{v}(\vec{c}_i) - \vec{V}_i||^2 + \lambda \int \sum_{m=0}^{\infty} c_m ||D^m \vec{v}||^2 ,$$

where $D^{2m}\vec{v} = \nabla^{2m}\vec{v}$, $D^{2m+1}\vec{v} = \nabla(\nabla^{2m}\vec{v})$, and the index i runs over all centroids in the image whose velocities have been estimated. Their solution, obtained by calculus of variations, has the form:

$$\vec{v}(\vec{r}) = \sum_i \frac{\vec{\beta}_i}{2\pi\sigma^2} exp\left(\frac{-||\vec{r} - \vec{c}_i||^2}{2\sigma^2}\right),$$

where the $\vec{\beta}_i$ are solutions of

$$\sum_j (\lambda\delta_{ij} + G_{ij})\vec{\beta}_j = \vec{V}_i,$$

where

$$G_{ij} = \frac{1}{2\pi\sigma^2} exp\left(\frac{-||\vec{c}_j - \vec{c}_i||^2}{2\sigma^2}\right) .$$

The error functional E contains an approximation-error term and a deviation-from-smoothness term. If we choose $\lambda < 1$, we emphasize more the agreement of the solution $\vec{v}(\vec{r})$ with the given data $\vec{V}_i$ rather than its smoothness. The solution can be seen as an interpolation formula that convolves the given velocities at the centroids with properly scaled Gaussian functions. The parameter σ should vary in proportion with the desired zone of influence of this smoothing/interpolation process. Fig. 1h shows the result of applying the above interpolation to the raw optical flow data of Fig. 1f.

5 Discussion

We have developed a region-based method for determining the optical flow. This method solves first a motion correspondence problem by matching region tokens via an affinity measure that depends on certain region features. For

each image frame, the regions are extracted either from the sign of the $\nabla^2 G_\sigma * I$ bandpass operator, or by thresholding the output of morphological image transformations for peak/valley detection. Optical flow is then identified as the spatial displacement vectors between centroids of corresponding regions. Finally, the relatively sparse optical flow data are either smoothed using a vector median filter or interpolated to produce dense velocity fields by using a motion-coherence regularization method. Our experiments indicate that the median smoothing almost always improves the original optical flow data by eliminating outliers while preserving abrupt discontinuities in the optical flow (that may indicate object boundaries). Smoothing via interpolation gives good dense results only if the *whole* original image is in motion. Otherwise, it forces a nonzero velocity on parts of the image that were originally not moving.

Our experiments on real images further indicate that this region-based method for optical flow is robust, computationally efficient, and more immune to noise than gradient methods, especially for medium-range motion. For example, Fig. 2h shows the optical flow estimated by the approach in [5], and Fig. 2g shows the optical flow estimated by our region-based method and smoothed via median filtering. (Displacements of up to 10 pixels were involved.) Clearly, there is a superiority of the region-based method over the gradient method, due to the robustness of the correspondence approach and the efficacy of regions as tokens to match.

References

[1] T. S. Huang and R. Y. Tsai, "Image Sequence Analysis: Motion Estimation", in *Image Sequence Analysis*, T.S. Huang, Ed., Springer-verlag, 1981.

[2] E. C. Hildreth and C. Koch, "The Analysis of Visual Motion: From Computational Theory to Neuronal Mechanisms", A.I.L. Memo 919, M.I.T., 1986.

[3] J. K. Aggarwal and N. Nandhakumar, "On the Computation of Motion from Sequences of Images-A Review", *Proc. IEEE*, 76, pp.917-935, Aug. 1988.

[4] J. J. Koenderinck and A. J. van Doorn, "Local structure and movement parallax of the plane", *J. Opt. Soc. Amer.*, 66, pp.717-723, July 1976.

[5] B. K. P. Horn, and B. G. Schunck, "Determining Optical Flow," *Artif. Intellig*, 17, pp. 185-203, Aug. 1981.

[6] A. M. Waxman and K. Wohn, "Image Flow Theory: A Framework for 3-D Inference from Time-Varying Imagery", in *Advances in Computer Vision*, Vol. 1, C. Brown, Ed., NJ: Erlbaum Publ., 1988.

[7] K. I. Kanatani, "Transformation of Optical Flow by Camera Rotation", *IEEE Trans. Pattern Anal. Mach. Intell.*, PAMI-10, March 1988.

[8] M. Bertero, T. Poggio and V. Torri, "Ill-Posed Problems in Early Vision", *Proc. IEEE*, 76, pp.869-889, Aug. 1988.

[9] J. K. Aggarwal, L. S. Davis and W. N. Martin, "Correspondence Processes in Dynamic Scene Analysis", *Proc. IEEE*, 69, pp.562-572, May 1981.

[10] D. Marr and S. Ullman, "Directional selectivity and its use in early visual processing", *Proc. Roy. Soc. Lond. B*, 211, pp.151-180, 1981.

[11] S. Ullman, "The Interpretation of Visual Motion", MA: MIT press, 1979.

[12] H. K. Nishihara, "Practical real-time imaging stereo matcher", *Optic. Enginr.*, 23, pp.536-545, 1984.

[13] F. Meyer, "Iterative Image Transformations for an Automatic Screening of Cervical Smears", *J. Histochem. and Cytochem.*, 27, pp.128-135, 1979.

[14] J. Serra, *Image Analysis and Mathematical Morphology*, NY: Acad. Press, 1982.

[15] P. Maragos and R. W. Schafer, "Morphological Filters", *IEEE Trans. Acoust. Speech Signal Process.*, ASSP-35, Aug. 1987.

[16] A. L. Yuille, and N. M. Grzywacz, "A computational theory for the perception of coherent visual motion", *Nature*, 333, pp. 71-74, 5 May 1988.

[17] D. Marr and E. C. Hildreth, "Theory of edge detection", *Proc. Roy. Soc. Lond. B*, 207, pp.187-217, 1980.

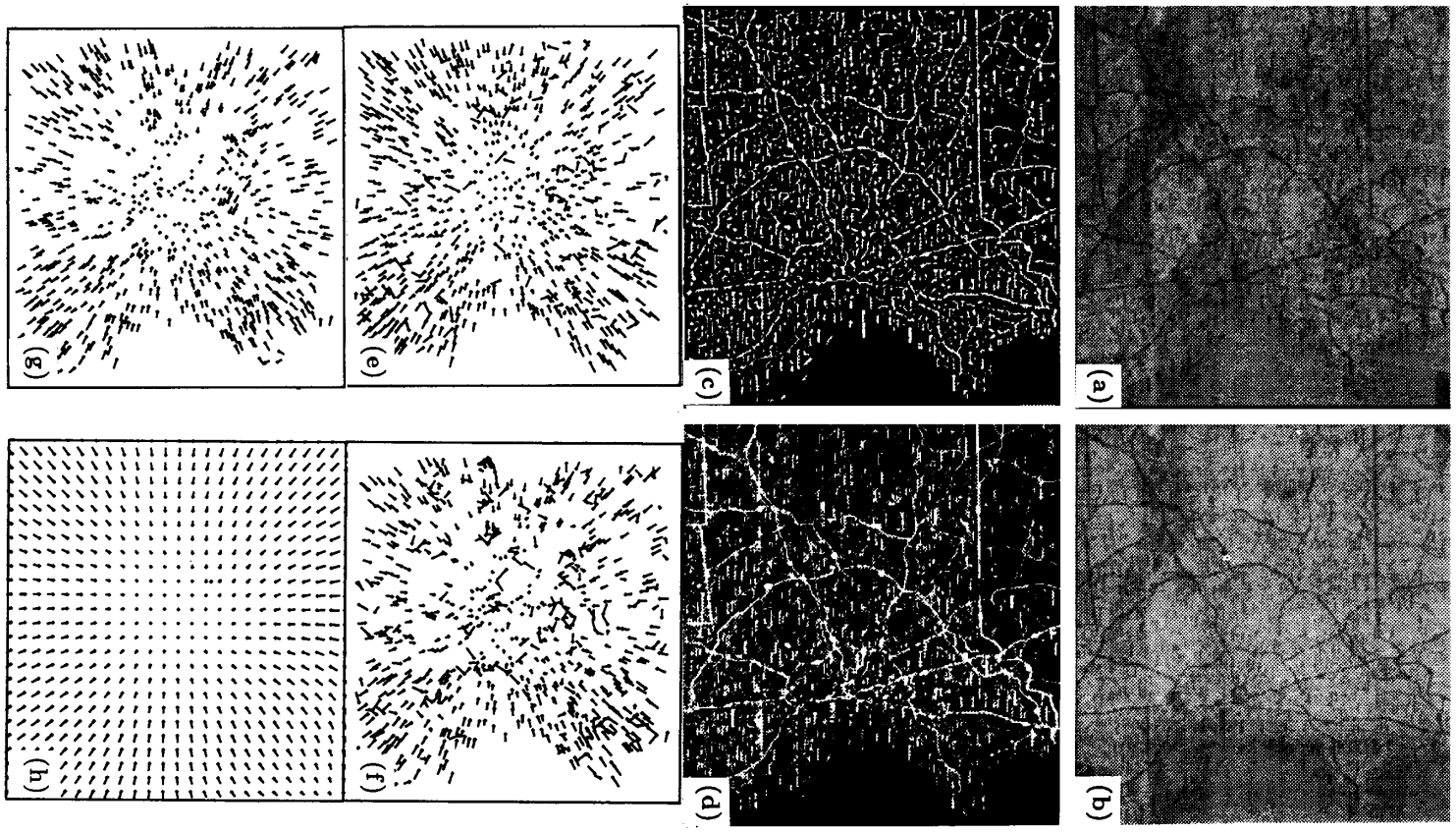

Figure 1. (a) First frame of a "Map" image sequence (485×512 pixels, 8-bit/pixel). (b) Second frame of "Map" after zooming in the scene of Fig. 1a. (Camera translation along the z-axis; camera plane parallel to xy-plane.) (c) Regions (positive in white and negative in black) showing the sign of the $\nabla^2 G_\sigma * I$ operator applied to the first frame. (d) Valley regions (in white) extracted from the first frame by thresholding its valley signal at $T = 25$. (The structuring element B was a 21-pixel octagon dilated four times.) (e) Optical flow between first and second frame from matching the positive/negative regions of the $\nabla^2 G_\sigma * I$ sign. (f) Optical flow from matching the peak/valley regions from the morphological feature detectors. (g) Smoothing the optical flow of Fig. 1f with vector median filtering. (h) Interpolating and smoothing the optical flow of Fig. 1f using the motion-coherence regularization. ($\lambda = \sigma = 0.3$ and the image plane was normalized to having size=1 in each direction.)

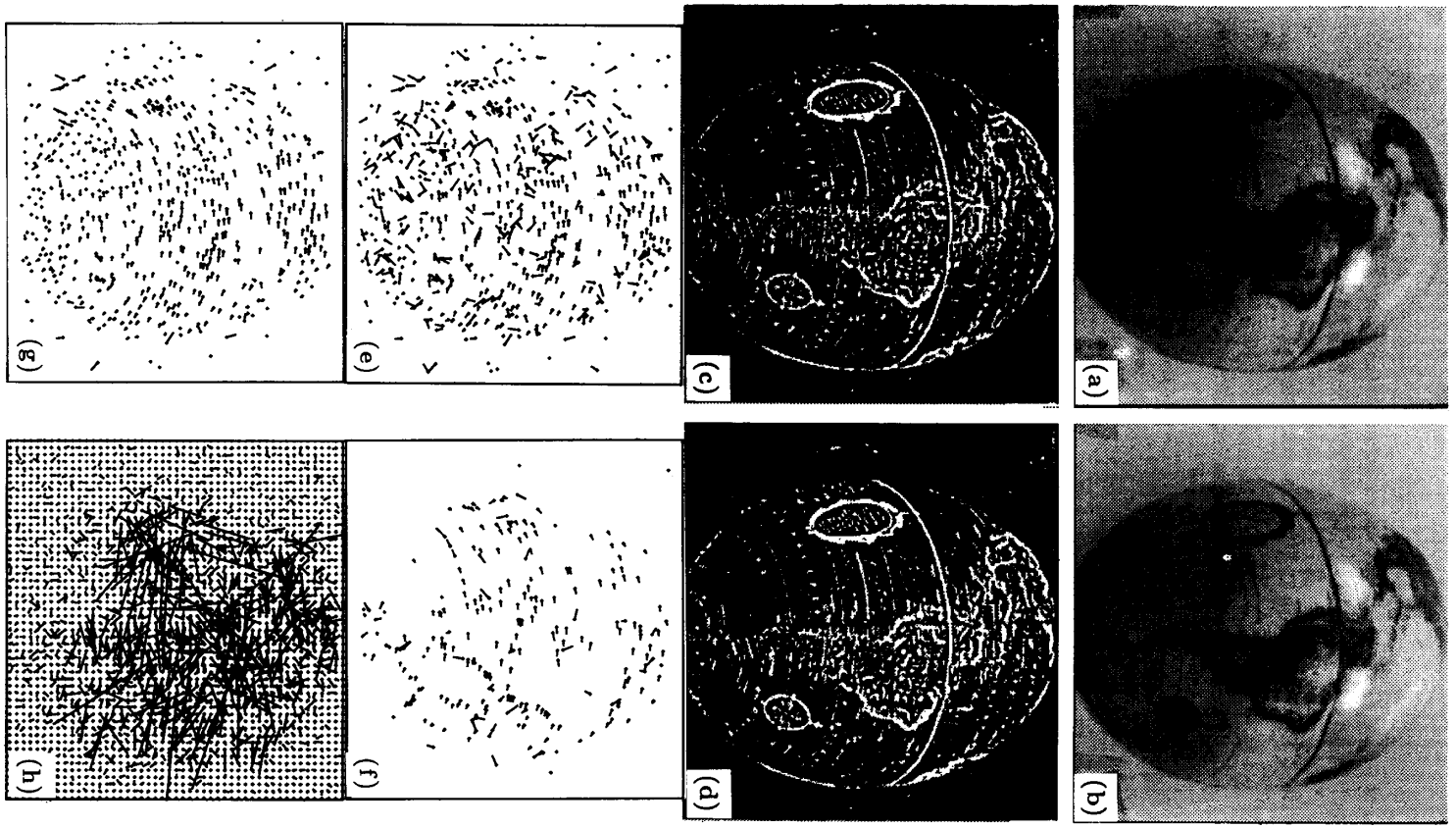

Figure 2. (a) First frame of a "Globe" image sequence (485×512 pixels). (b) Second image frame resulting from rotating the "Globe" in front of the camera. (c) Regions (positive in white and negative in black) showing the sign of the $\nabla^2 G_\sigma * I$ operator applied to the first frame. (d) $\nabla^2 G_\sigma * I$ sign regions of the second frame. (e) Optical flow between first and second frame from matching the positive/negative regions of the $\nabla^2 G_\sigma * I$ sign. (f) Optical flow from matching the peak/valley regions from the morphological feature detectors. (g) Smoothing the optical flow of Fig. 2e with vector median filtering. (h) Optical flow obtained by using Horn & Schunck's [5] gradient method (264 iterations).