

Dynamic Occlusion Analysis in Optical Flow Fields

WILLIAM B. THOMPSON, MEMBER, IEEE, KATHLEEN M. MUTCH, MEMBER, IEEE,
AND VALDIS A. BERZINS, MEMBER, IEEE

Abstract—Optical flow can be used to locate dynamic occlusion boundaries in an image sequence. We derive an edge detection algorithm sensitive to changes in flow fields likely to be associated with occlusion. The algorithm is patterned after the Marr-Hildreth zero-crossing detectors currently used to locate boundaries in scalar fields. Zero-crossing detectors are extended to identify changes in direction and/or magnitude in a vector-valued flow field. As a result, the detector works for flow boundaries generated due to the relative motion of two overlapping surfaces, as well as the simpler case of motion parallax due to a sensor moving through an otherwise stationary environment. We then show how the approach can be extended to identify which side of a dynamic occlusion boundary corresponds to the occluding surface. The fundamental principle involved is that at an occlusion boundary, the image of the surface boundary moves with the image of the occluding surface. Such information is important in interpreting dynamic scenes. Results are demonstrated on optical flow fields automatically computed from real image sequences.

Index Terms—Dynamic occlusion, dynamic scene analysis, edge detection, optical flow, visual motion.

I. INTRODUCTION

AN optical flow field specifies the velocity of the image of points on a sensor plane due to the motion of the sensor and/or visible objects. Optical flow can be used to estimate aspects of sensor and object motion, the position and orientation of visible surfaces relative to the sensor, and the relative position of different objects in the field of view. As a result, the determination and analysis of optical flow is an important part of dynamic image analysis. In this paper, we develop an operator for finding occlusion boundaries in optical flow fields. We deal exclusively with *dynamic occlusions* in which flow properties differ on either side of the boundary. The operator is effective for both motion parallax, when a sensor is moving through an otherwise stationary environment, and for more general motion in which multiple moving objects can be in the field of view. The multiple moving object situation is more difficult because boundaries are marked by almost arbitrary combinations of changes in magnitude and/or direction of flow.

Manuscript received June 1, 1984; revised February 14, 1985. Recommended for acceptance by W. E. L. Grimson. This work was supported by the National Science Foundation under Grant MCS-81-05215, the Air Force Office of Scientific Research under Contract F49620-83-0140, and by Zonta International.

W. B. Thompson and V. A. Berzins are with the Department of Computer Science, University of Minnesota, Minneapolis, MN 55455.

K. M. Mutch was with the Department of Computer Science, University of Minnesota, Minneapolis, MN 55455. She is now with the Department of Computer Science, Arizona State University, Tempe, AZ 85287.

The technique is extended so that a determination may be made about which side of a dynamic occlusion boundary corresponds to the occluding surface. Such a determination is of great importance for interpreting the shape and spatial organization of visible surfaces. Results are demonstrated on real image sequences with flow fields computed using the token matching technique described in [1]. Reliability is obtained by dealing only with methods able to integrate flow field information over relatively large neighborhoods so as to reduce the intrinsic noise in fields determined from real image sequences.

II. BOUNDARY DETECTION

Conventional edge operators detect discontinuities in image luminance. These discontinuities are difficult to interpret, however, because of the large number of factors that can produce luminance changes. Boundaries in optical flow can arise from many fewer causes and, hence, are often more informative. If a sensor is moving through an otherwise static scene, a discontinuity in optical flow occurs only if there is a discontinuity in the distance from the sensor to the visible surfaces on either side of the flow boundary [2]. Discontinuities in flow will occur for all visible discontinuities in depth, except for viewing angles directly toward or away from the direction of sensor motion. If objects are moving with respect to one another in the scene, then all discontinuities in optical flow correspond either to depth discontinuities or surface boundaries, and most depth discontinuities correspond to flow discontinuities.

The use of local operators to detect discontinuities in optical flow has been suggested by others. Nakayama and Loomis [3] propose a "convexity function" to detect discontinuities in image plane velocities generated by a moving observer. Their function is a local operator with a center-surround form. That is, the velocity integrated over a band surrounding the center of the region is subtracted from the velocity integrated over the center. The specifics of the operator are not precisely stated, but a claim is made [3, Fig. 3] that the operator returns a positive value at flow discontinuities. (In fact, most reasonable formulations of their operator would yield a value of 0 at the boundary, with a positive value to one side or the other.) Clocksin [2] develops an analysis of optical flow fields generated when an observer translates in a static environment. He shows that, in such circumstances, discontinuities in the magnitude of flow can be detected with a Laplacian operator. In particular, singularities in the Laplacian occur at discontinuities in the flow. He also showed that, in this restricted environ-

ment, the magnitude of optical flow at a particular image point is inversely proportional to distance, and the distances can be recovered to within a scale factor of observer speed. It is thus trivial to determine which of two surfaces at an edge is occluding, for example, by simply comparing magnitudes of the two surface velocities, even when observer speed is unknown.

For this restricted situation in which a sensor moves through an otherwise static world

$$\text{flow}(\vec{x}) = f_r(\vec{x}) + \frac{f_t(\vec{x})}{r(\vec{x})} \quad (1)$$

where at an image point \vec{x} , $\text{flow}(\vec{x})$ is the optical flow (a two-dimensional vector), f_r is the component of the flow due to the rotation of the scene with respect to the sensor, f_t is dependent on the translational motion of the sensor and the viewing angle relative to the direction of translation, and r is the distance between the sensor and the surface visible at \vec{x} [4]. For a fixed \vec{x} , flow varies inversely with distance. Both f_r and f_t vary slowly (and continuously) with \vec{x} . Discontinuities in flow thus correspond to discontinuities in r . Furthermore, it is sufficient to look only for discontinuities in the magnitude of flow. This relationship holds only for relative motion between the sensor and a single, rigid structure. When multiple moving objects are present, (1) must be modified so that there is a separate $f_r^{(i)}$ and $f_t^{(i)}$ specifying the relative motion between the sensor and each rigid object. Discontinuities associated with object boundaries may now be manifested in the magnitude and/or direction of flow.

Boundary detectors for optical flow fields should satisfy two criteria: 1) sensitivity to rapid spatial change in one or both of the magnitude and direction of flow, and 2) operation over a sufficiently large neighborhood to reduce sensitivity to noise in computed flow fields. It is desirable to achieve the second criterion without an unnecessary loss of spatial resolution in locating the boundary or a need for postprocessing to reduce the width of detected boundaries. The *zero-crossing* detectors of Marr and Hildreth [5] may be extended to optical flow fields in a manner that achieves both objectives [6]. For scalar fields (e.g., intensity images), zero-crossing edge detection proceeds as follows. 1) Smooth the field using a symmetrical Gaussian kernel. 2) Compute the Laplacian of the smoothed function. 3) Look for directional zero crossings of the resulting function (e.g., look for points at which, along some direction, the function changes sign). Under a set of relatively weak assumptions, these zero crossings can be shown to correspond to points of most rapid change in some direction in the original function. The convolution with a Gaussian provides substantial noise reduction and, in addition, allows tuning of the method for edges of a particular scale. Steps 1) and 2) involve evaluating the function $\nabla^2 G * I$, where G is a Gaussian kernel, $*$ is the convolution operation, and I is the original image. The effect of the $\nabla^2 G$ operator can be approximated by blurring the original function with two different Gaussian kernels of appropriate standard deviation, and then taking the difference of the result. This formulation results in computational simplifications [7], [8] and also corresponds nicely to several physiological models that have been proposed for early visual processing.

The effect of this approach is to identify edge points where the intensity of the blurred image is locally steepest. More precisely, an edge can be defined as a peak in the first directional derivative, or as a zero crossing in the second directional derivative. At an edge, the second directional derivative has zero crossings in almost all directions, but the preferred direction is normal to the locus of the zero crossings, which is the same as the direction where the zero crossing is steepest for linearly varying fields [5]. For vector images such as optical flow fields, the directional derivatives are vector valued, and we want the *magnitude* of the first directional derivative to have a peak.

This extension to two-dimensional flow fields is relatively straightforward. The optical flow field is first split into separate scalar components corresponding to motion in the x and y directions. The $\nabla^2 G$ operator is applied to each of these component images, and the results combined into a componentwise Laplacian of the original flow field. (The Laplacian is a vector operator which can be expressed in arbitrary coordinate systems. For convenience, we choose a Cartesian coordinate system.) This componentwise Laplacian operation is implemented by subtracting two componentwise blurred versions of the original. With the proper set of weak assumptions, discontinuities in optical flow correspond to zeros in both of these component Laplacian fields. At least one of the components will have an actual zero crossing. The other will have either a zero crossing or will have a constant zero value in a neighborhood of the discontinuity. If the componentwise Laplacians are treated as a two-dimensional vector field, discontinuities are indicated by directional reversals in the combined field. Because of the discrete spatial sampling and a variety of noise sources, the zeros or zero crossings in the two components of the field may not actually be exactly spatially coincident. Thus, exact reversal is not expected, and a range of direction changes of about 180° is accepted. A threshold on the sum of the vector magnitudes at the location of the flip is used to ensure that the zero crossing is of significant slope. This is analogous to the threshold on zero-crossing slope which is often used in practice when zero-crossing techniques are used on intensity images, and serves to filter out small discontinuities.

The approximations made by the computations described above will be good if the variation of the field parallel to the edge is much more uniform than the variation normal to the edge. For scalar images, exact results will be obtained if the intensity varies at most linearly along the edge contour [5]. For vector images, the field must vary at most linearly in some *neighborhood* of the edge contour, so that the assumptions required are slightly stronger than for scalar images. Appendix I contains the analysis for the case of vector images.

Two examples of this technique applied to real images are shown below. In both examples, the objects are toy animals with flat surfaces, shown moving in front of a textured background. In Fig. 1(a), the tiger translates parallel to the image plane from right to left between frames 1 and 2. The elephant rises off its front legs between frames 1 and 2, effectively rotating about an axis at its hind feet oriented perpendicularly to the image plane. The elephant also translates slightly to the left parallel to the image plane. The optical flow vectors, shown

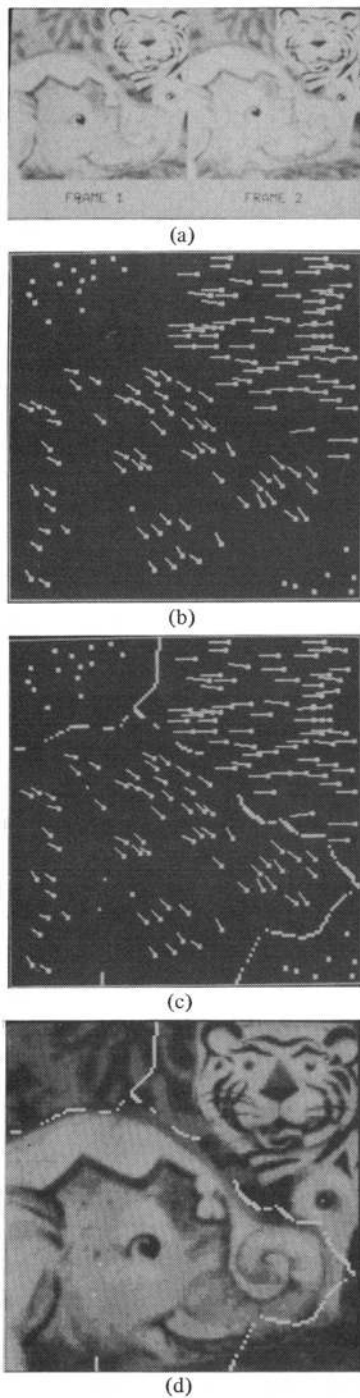


Fig. 1. (a) Image pair. (b) Optical flow. (c) Detected edge overlaid onto flow field. (d) Detected edge overlaid onto first frame of sequence.

in Fig. 1(b), were obtained by relaxation labeling token matching, as described in [1]. Notice that the flow vectors on the elephant and tiger have approximately the same magnitude but differ in direction. Each component of this flow field was convolved with approximated Gaussians of standard deviations 3.65 and 5.77. The ratio of these standard deviations is 1 : 1.6. The two convolved flow fields were subtracted, and the resulting vector field was searched for reversals in vector direction. A boundary strength threshold was chosen to eliminate noise points due to small, local variations in estimated flow. In Fig. 1(c), the points where reversals were found are shown

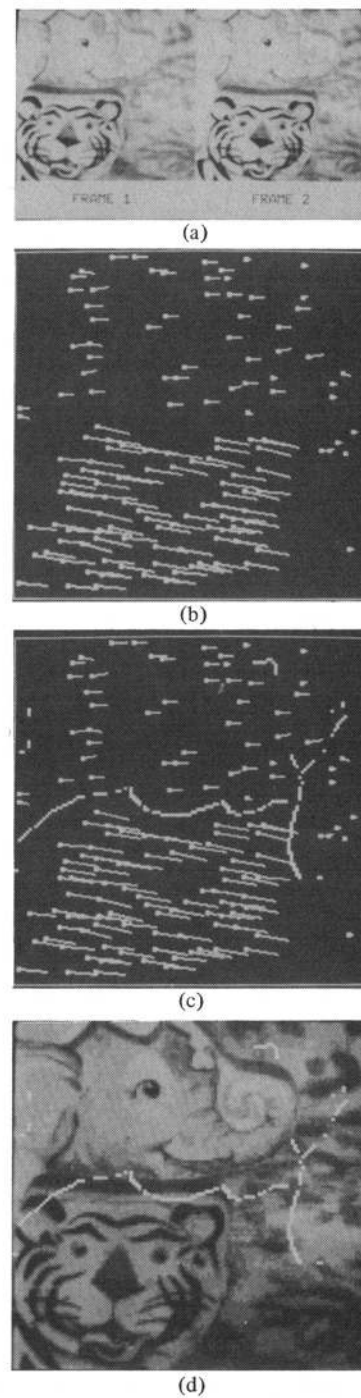


Fig. 2. (a) Image pair. (b) Optical flow. (c) Detected edge overlaid onto flow field. (d) Detected edge overlaid onto first frame of sequence.

overlaid on the original flow field, and in Fig. 1(d) the points are overlaid in white on the first image of the pair. The edge points form a good boundary between the discontinuous optical flow vector fields [Fig. 1(c)]; but because these fields are so sparse, the edge points match only the approximate locations of the true edges [Fig. 1(d)].

In Fig. 2(a), both the tiger and elephant are translating to the right, parallel to the image plane between frames 1 and 2. The flow field shown in Fig. 2(b) was obtained in the same manner as in Fig. 1(b). The direction of the flow vectors on both animals is approximately the same, but there is a dis-

continuity in magnitude. Two Gaussian filtered versions of the flow fields were obtained with standard deviations of 3.16 and 5.16—a ratio of 1:1.6. The locations of vector reversals resulting from differencing the two filtered fields are shown in Fig. 2(c) and (d).

The width of the Gaussian kernel used in the $\nabla^2 G$ operator, the density of the computed optical flow field, and the spatial variability of flow all interact to affect the performance of the boundary detection. As with the use of zero-crossing detectors for scalar fields, it may be desirable to use a range of kernel sizes and then combine the results to obtain a more robust indicator for the presence of a boundary. While zero-crossing contours are, in principle, connected, the use of a threshold on the slope at the zero crossing results in some portions of the boundary being missed. In practice, zero-crossing boundary detection for both scalar and vector fields often requires such thresholds to avoid significant problems with false boundary indications in slowly varying regions of the fields. Work still needs to be done on better techniques for selecting zero crossings that correspond to true boundaries.

III. IDENTIFYING OCCLUDING SURFACES

When analyzing edges between dissimilar image regions that arise due to occlusion boundaries, it is important to determine which side of the edge corresponds to the occluding surface. Occlusion boundaries arise due to geometric properties of the occluding surface, not the occluded surface. Thus, while the shape of the edge provides significant information on the structure of the occluding surface, it says little or nothing about the structure of the surface being occluded. In situations where a sensor is translating through an otherwise static scene, any significant local decrease in r in (1) increases the magnitude of flow. Thus, at a flow boundary, the side having the larger magnitude of flow will be closer, and thus will be occluding the farther surface. Sensor rotation complicates the analysis, while if objects in the field of view move with respect to each other, there is no direct relationship between magnitude of flow and r . Surfaces corresponding to regions on opposite sides of a boundary may move in arbitrary and unrelated ways. However, by considering the flow values on either side of the boundary and the manner in which the boundary itself changes over time, it is usually possible to find which side of the boundary corresponds to the occluding surface, although the depth to the surfaces on either side cannot be determined.

The principle underlying the approach is that the image of the occluding contour moves with the image of the occluding surface. Fig. 3 illustrates the effect for simple translational motion. Shown on the figure are the optical flow of points on each surface and the flow of points on the image of the boundary. In Fig. 3(a), the left surface is in front and occluding the surface to the right. In Fig. 3(b), although the flow values associated with each surface are the same, the left surface is now behind and being occluded by the surface to the right. The occluding surface cannot be determined using only the flow in the immediate vicinity of the boundary. The two cases can be distinguished because, in Fig. 3(a), the flow boundary determined by the next pair of images will be displaced to the left, while in Fig. 3(b) it will be displaced to the right.

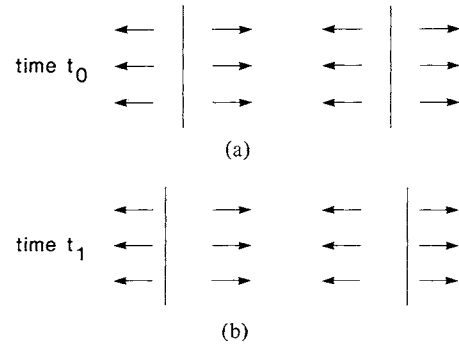


Fig. 3. Optical flow at a boundary at two instants in time. (a) Surface to the left is in front. (b) Surface to the right is in front.

To formalize the analysis, we need to distinguish the optical flow of the boundary itself from the optical flow of surface points. The flow of the boundary is the image plane motion of the boundary, which need not have any direct relationship to the optical flow of regions adjacent to the boundary. The magnitude of the optical flow of boundary points parallel to the direction of the boundary typically cannot be determined, particularly for linear sections of boundary. Thus, we will limit the analysis in this section to the component of optical flow perpendicular to the direction of the image of occlusion boundaries. As a result, if the flow on both sides of the boundary is parallel to the boundary, the boundary will still be detectable, but the method given here will provide no useful information about which surface is occluding.

We can now state the basic principle more precisely. Choose a coordinate system in the image plane with the origin at a particular boundary point and the x axis oriented normal to the boundary contour, with $x > 0$ for the occluding surface. The camera points in the z direction, and the image plane is at $z = 0$. Let $f_x(x, y)$ be the x component of optical flow at the point (x, y) . Let f_b be the x component of the flow of the boundary itself at the origin (i.e., f_b is the image plane velocity of the boundary in a direction perpendicular to the boundary). Then, for rigid objects,

$$f_b = \lim_{x \rightarrow 0^+} f_x(x, 0) = f_x(0, 0). \quad (2)$$

We will show that this relationship is true for arbitrary rigid body motion under an orthographic projection. For a single smooth surface, perspective projections are locally essentially equivalent to a rotation plus a scale change, although the analysis is more complex. Equation (2) specifies a purely local constraint and, as the limit is taken from only one side of the boundary, is dependent on flow values on a single surface. Thus, the limit result will hold as well for perspective projections. Algorithms which utilize the result in (2) may suffer, however, if properties of more than a truly local area of the field are utilized. The instantaneous motion of a rigid object relative to a fixed coordinate system can be described with respect to a six-dimensional, orthogonal basis set. Three values specify translational velocity, the other three specify angular velocity. These six coordinates of motion can be conveniently classified into four types: translation at constant depth, translation in depth, rotation at constant depth, and rotation in depth. Translation at constant depth is translation in a direction parallel to the image plane. Translation in depth is translation per-

pendicular to the image plane. Rotation at constant depth is rotation around an axis perpendicular to the image plane. Rotation in depth is rotation around an axis parallel to the image plane. Any instantaneous motion can be described as a combination of these four types. For orthographic projections, translation in depth has no effect on the image. Thus, we need to show that the above relationship relating boundary and surface flow holds for the three remaining motion types.

A point on the surface of an object in the scene that projects into a boundary point in the image will be referred to as a *generating point* of the occlusion boundary. The family of generating points defines a *generating contour*, which lies along the extremal boundary of the object with respect to the sensor. For both translation and rotation at constant depth, the generating contour remains fixed to the occluding surface over time. Thus, the boundary and adjacent points move with exactly the same motion. As a result, the projection of the surface flow in the direction normal to a particular boundary point is identical to the projection of the boundary flow in the same direction. (The result is strictly true only for instantaneous flow. Over discrete time steps, boundary curvature will affect the projected displacement of the boundary.)

The analysis of rotation in depth is complicated by a need to distinguish between sharp and smooth occlusion boundaries, based on the curvature of the occluding surface. The intersection of the surface of the object and a plane passing through the line of sight to the generating point and the surface normal at the generating point defines a *cross section contour*. The cross section contour and the generating contour cross at right angles at the generating point. *Sharp* boundaries occur when the curvature of the cross section contour at a generating point is infinite. *Smooth* boundaries occur when the curvature is finite.

Sharp generating contours will usually remain fixed on the object surface over time. (Exceptions occur only in the infrequent situations in which, due to changes in the line of sight with respect to the object, either sharp boundary becomes smooth or a flat face on one side of the generating point lines up with the line of sight.) Smooth generating contours will move along the surface of the object any time the surface orientation at a point fixed to the surface near the extremal boundary is changing with respect to the line of sight. Fig. 4 shows examples of both possibilities. The figure shows a view from above, with the sensor looking in the plane of the page and the objects rotating around an axis perpendicular to the line of sight. In Fig. 4(a), an object with a square cross section is being rotated. Fig. 4(b) shows an object with a circular cross section.

For sharp boundaries, a surface point close to a generating point in three-space projects onto the image at a location close to the image of the generating point. The surface point and the generating point move as a rigid body. For rigid body motion, differences in flow between the image of two points go to zero as the points become coincident in three-space. As a result, surface points arbitrarily close to the generating point project to the same flow values as the generating point itself.

For smooth boundaries, the situation is more complex. The surface points corresponding to the boundary may change over time, so that points on the surface near the generating point

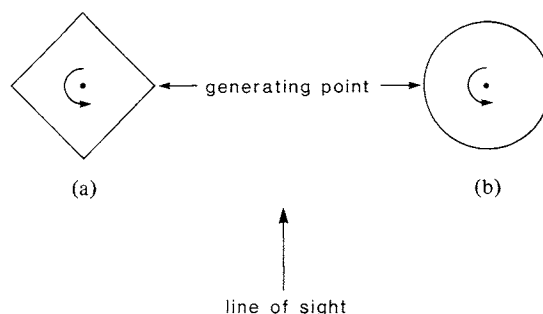


Fig. 4. (a) Generating contour at a sharp boundary remains fixed to the object surface. (b) Generating contour at a smooth boundary moves relative to the object surface.

and the generating point itself may not maintain a fixed relationship in three-space. The property described in (2) still holds for rotation in depth, however. The formal proof of this assertion is relatively complex and is given in Appendix B. (The Appendix actually shows that the limit of surface flow is equal to boundary flow for rotation of smooth objects around an arbitrarily oriented axis.) Informally, the result holds because the surface is tangent to the line of sight at the generating point, so that any motion of the generating point with respect to a point fixed to the surface is along the line of sight. The *difference* between the motion of the surface near the generating point and the motion of the generating point itself is a vector parallel to the line of sight and, hence, does not appear in the projected flow. This means that the motion of the boundary in the x direction will be the same as that of a point fixed to the surface at the instantaneous location of the generating point. The limit property holds because the surface flow varies continuously with x in the vicinity of the generating point, as long as we restrict our attention to points that are part of the same object.

To develop an algorithm for actually identifying the occluding surface at a detected boundary, we will start by assuming only translational motion is occurring. (Violations of this assumption are discussed below.) According to (2), we need only look at the flow at the edge point and immediately to either side to determine which side corresponds to the occluding surface. In practice, however, this is inadequate. Edges will be located imprecisely in each frame due to a variety of effects. This imprecision is compounded when the location of edge points is compared across frames to determine the flow of the edge. By considering the pattern of change in the Laplacian of the optical flow field, however, a simple binary decision test can be constructed to determine which surface velocity most closely matches that of the edge. As before, we will use a coordinate system with its origin at the location of some particular boundary point at a time t_0 , the x axis oriented normal to the orientation of the boundary, and consider only $flow_x$, the projection of flow onto the x axis. In this new coordinate system, positive velocity values will correspond to motion to the right. We will assume that the flow field in the vicinity of the edge can be approximated by a step function. The algorithm developed here is unaffected by constants added to the flow field or by applying positive multiples to the magnitude of flow. Therefore, to simplify analysis, normalize the flow field by subtracting a constant value f_a such that the pro-

jected velocities of the two surfaces have equal magnitudes and opposite signs, and then multiply by a positive scale factor f_s such that the magnitudes will be normalized to 1 and -1 [i.e., $flow'_x = f_s(flow_x - f_a)$]. The resulting step edges can have one of two possible shapes, depending upon whether the surface to the left is, after scaling and normalizing, moving to the left or to the right (see Fig. 5).

When the two possible velocity functions are convolved with a Gaussian blurring kernel, the resulting functions are shown in Fig. 5(a) and (b). The Laplacian of these functions in the direction perpendicular to the edge is equal to the second derivative, and is shown in Fig. 5(c) and (d). These two cases may be described analytically as follows.

Case 1: Given the step function

$$s(x) = \begin{cases} 1, & x < 0 \\ -1, & x > 0 \end{cases} \quad (3)$$

convolve $s(x)$ with a Gaussian blurring function $g(x)$.

$$h(x) = g * s. \quad (4)$$

Let $s(x) = -2u(x) + 1$ where

$$u(x) = \begin{cases} 0, & x < 0 \\ 1, & x > 0. \end{cases}$$

Then

$$h(x) = 1 - 2 \int_{-\infty}^x \frac{1}{\sigma \sqrt{2\pi}} e^{-\lambda^2/2\sigma^2} d\lambda \quad (5)$$

$$h''(x) = \frac{2x}{\sigma^3 \sqrt{2\pi}} e^{-x^2/2\sigma^2}. \quad (6)$$

Therefore,

$$\begin{aligned} h''(x) &< 0 \quad \text{when } x < 0 \\ h''(x) &> 0 \quad \text{when } x > 0. \end{aligned} \quad (7)$$

Case 2: The step function for case 2 is $-s(x)$, where $s(x)$ and $u(x)$ are defined above

$$h''(x) = \frac{-2x}{\sigma^3 \sqrt{2\pi}} e^{-x^2/2\sigma^2}. \quad (9)$$

Therefore,

$$\begin{aligned} h''(x) &> 0 \quad \text{when } x < 0 \\ h''(x) &< 0 \quad \text{when } x > 0. \end{aligned} \quad (10)$$

At some later time t_1 , the entire second derivative curve $h''(x)$ will have shifted right or left, depending upon whether the edge moves with the surface moving to the right or left. Based upon the analysis above, in case 1, if the left surface is occluding, the second derivative curve will be moving to the right and the sign at the origin will become negative, while if the right surface is occluding, the curve will be moving left and the sign at the origin will be positive. In case 2, if the left surface is occluding, the curve will be moving to the left and the sign at the origin will be negative; while if the right surface is occluding, the curve will be moving to the right and the sign

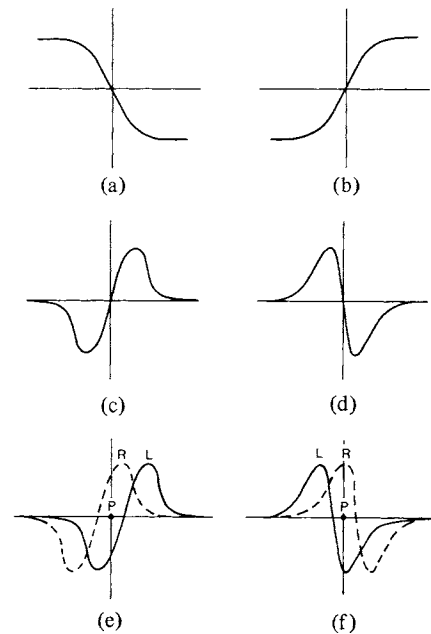


Fig. 5. Smoothed magnitude of flow for (a) case 1 and (b) case 2. (c) and (d) Laplacian of the functions in (a) and (b). (e) and (f) Two possible locations of the Laplacian curves after an interval of time. The dashed curve indicates the location of the curve if the edge moves with the surface to the right. The solid curve indicates the location of the curve if the edge moves with the surface to the left.

at the origin will be positive. Note that in both cases, when the left surface is the occluding surface, the sign at the origin will become negative, and when the right surface is occluding, the sign at the origin will become positive. This is illustrated in Fig. 5(e) and (f). In the original, unrotated coordinate system, this is equivalent to stating that at time t_1 the direction normal to the edge for which the second directional derivative of optical flow is positive, evaluated at the location of the edge at t_0 , points toward the occluding surface. (The approach is similar to that used in [9] to determine the direction of motion of an intensity contour.) This analysis may be extended to the general case where the original step function has not been normalized. The direction of the second derivative at t_1 must now, however, be evaluated at the point $(x_0, y_0) + (t_1 - t_0)f_a$. (As f_a is the average flow of the surfaces on either side of the boundary, this point may be thought of as lying half-way between the two possible image locations of the boundary at time t_1 .)

In practice, difficulties may arise for very large differential flows between the two surfaces. The second derivative function $h''(x)$ approaches zero away from the zero crossing. Noise sensitivity of the classification technique is likely to increase when the value is small. It is useful to determine a guideline for the size of the Gaussian blurring kernel to ensure that the curve will be observed near its extrema, where the sign is more likely to be correct. The form of the function $h''(x)$ may be simplified by substitution for analysis purposes. Let

$$b = \frac{x}{\sigma \sqrt{2}} \quad \text{and} \quad c = \frac{2}{\sigma^2 \sqrt{\pi}}. \quad (12)$$

Then, in case 1,

$$h''(x) = f(b) = cbe^{-b^2} \quad (13)$$

$$f'(b) = ce^{-b^2} (1 - 2b^2). \quad (14)$$

The extrema of $f(b)$ will occur at $b = \pm 1/\sqrt{2}$, and the extrema of $h''(x)$ occur at $x = \pm\sigma$. The ratio

$$\frac{h''(2.7\sigma)}{h''(\sigma)} = 0.12 \quad (15)$$

indicates that at $\pm 2.7\sigma$ the magnitude of $h''(x)$ is 12 percent of its magnitude at the extrema, and thus is relatively close to zero. If the noise is such that the sign will be accurate when the expected Laplacian value is at least 10 percent of the extrema value, then a Gaussian blurring kernel should be used of standard deviation at least $1/2.7$ of the maximum expected magnitude of flow of the edge. For cases where the noise presents more of a problem, a Gaussian of larger standard deviation should be used. The analysis for case 2 can be performed similarly with the same result.

The algorithm is implemented as follows. Optical flow fields are obtained for two temporally adjacent image pairs. Approximation to the Laplacians of Gaussian blurred versions of these flow fields are calculated by computing the difference of the flow fields convolved with two different Gaussian kernels. (Again, the componentwise Laplacian is used.) As before, edge points are found in the first flow field by searching for vector reversals in the Laplacian of the field. At such points, the value of the smoothed flow field obtained from the larger of the Gaussian kernels is considered to approximate the average flow of the two surface regions on either side of the edge. This average flow is used to find the appropriate offset to add to the edge location to find P , a point midway between the two possible edge locations in the second Laplacian field. Next, the direction perpendicular to the edge point is estimated by finding the direction of greatest change in the Laplacian of the first flow field. The Laplacian of the second flow field at the point P is then examined. The Laplacian component in the second field perpendicular to the edge orientation points toward the occluding surface.

An example of this technique applied to an image sequence is shown in Fig. 6. The leopard translates from left to right approximately equally between frames 1, 2, and 3 in Fig. 6(a). The edge points shown in Fig. 6(b) are obtained as described in Section II. At each edge point, an offset based on the flow vector from the smoother version of the field at that point is added to the location of the edge point. The resulting location is examined in the Laplacian of the second flow field. The component of this Laplacian perpendicular to the edge will point toward the occluding surface. Shown in Fig. 6(c) are the edge points, each of which has an associated line segment. The line segment projects in the direction of the occluding surface, as determined by the algorithm. The correct classification is made for all except a few points at the bottom of the edge. In this region, several nearby tokens were matched in one frame pair but not the other, significantly affecting the smoothed flow fields in the neighborhood of the boundary.

IV. ROTATIONAL MOTION

Rotation in depth introduces several complexities for the analysis of optical flow at occlusion boundaries. The first is an

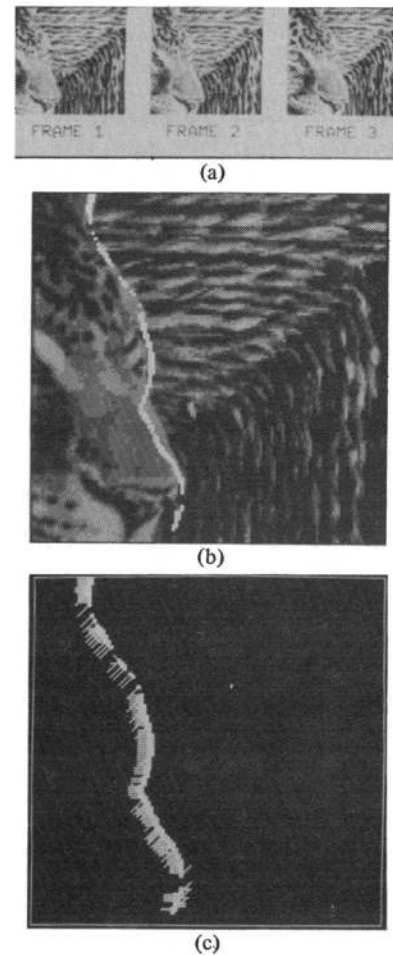


Fig. 6. (a) Image sequence. (b) Detected boundary overlaid onto first frame of sequence. (c) Identification of occluding surface. Each edge point has a line segment projecting from it toward the occluding surface.

unexpected corollary of (2): in certain situations, there is *no* discontinuity in flow at occlusion boundaries. This occurs for pure rotation in depth of objects that are circularly symmetric, rotating about their axis of symmetry, and otherwise stationary with respect to the background. In such cases, the image of the boundary over time maintains a fixed position with respect to the background. As a consequence of (2), the projected surface flows on either side of the boundary are identical and are the same as the boundary flow itself. Fortunately, the zero-crossing-based boundary detection method is still usually applicable, although the detected location of the boundary may be displaced.

The second complication involves the determination of occluding surfaces. Rotations in depth produce a dynamic *self*-occlusion—the rotating object occludes sections of itself over time. In the situation described in the previous paragraph, self-occlusion is the only dynamic occlusion occurring. In these circumstances, the relationship in (2) is of no direct value in identifying the occluding surface. No information is available on which side of the boundary corresponds to a true occluding surface. (The situation is truly ambiguous in that two very different classes of spatial organizations can produce the same flow pattern.) If the rotating object is also translating relative to the background, if the object is not rotationally symmetric, or if it

is not rotating around an axis of symmetry, then (2) will, in principle, correctly identify the occluding surface. Difficulties arise in practice, however, because the algorithm given above depends on surface flow in the neighborhood of the boundary, not just at the edge. In the presence of rotation in depth, misclassifications are possible, particularly if no translation relative to the background is occurring and/or the rotating object is small, leading to rapidly changing flow values near the extremal boundary.

Rotation also complicates inferences about relative depth based on the analysis of occlusion boundaries. For translational motion, the occluding surface on one side of a boundary is necessarily in front of the occluded surface. For rotation in depth, the occluded and occluding surfaces are on the same side of the boundary, and no definitive information is available about the surface on the other side of the boundary. (Reference [10] shows an example in which a nonrotating surface on one side of a boundary is in front of a rotating surface on the other side of the boundary.) One approach to determining the actual relative depth involves first determining whether or not rotation in depth is actually occurring. Such an analysis is beyond the scope of this paper (see [11]). As an alternative, an analysis of surface regions that are appearing or disappearing due to dynamic occlusion gives information about the occluded surfaces at a boundary [10]. The method described here gives information about the occluding surface. By combining the two approaches, self-occlusion is recognized by noting a boundary where one side is marked as both occluding and occluded.

V. CONCLUSION

Motion-based boundary detection is sensitive only to depth discontinuities and/or object boundaries. Thus, unlike intensity-based edge detection, all detected edge points are of direct significance to the interpretation of object shape. On the other hand, significant edges will not be detected unless there is perceived motion between the surfaces on either side. Motion-based analysis offers another significant advantage. In most cases, the side of a boundary corresponding to the occluding surface can be identified. As we have shown, this is possible for general motion, not just for a sensor moving through an otherwise static environment. This determination is quite difficult using only static information, and has received only little attention (e.g., [12]).

APPENDIX A

The following is an analysis of the appropriateness of using zero crossings in the componentwise Laplacian of a flow field to detect contours of maximal rate of change in the flow field.

Theorem: Let V be a twice continuously differentiable vector field, let N be an open neighborhood containing the origin such that $\partial V/\partial y$ is constant on N , let L be the intersection of N and the y axis, and let u be a unit vector. Then $|\nabla V \cdot u|^2$ has an extremum in the x direction on L if and only if $u_x(u \cdot \nabla V) \cdot \nabla^2 V$ has a zero crossing on L .

Justification: The magnitude of the directional derivative in the u direction is

$$|\nabla V \cdot u|^2 = (\nabla V_x \cdot u)^2 + (\nabla V_y \cdot u)^2 \quad (16)$$

$$= \left[u_x \frac{\partial V_x}{\partial x} + u_y \frac{\partial V_x}{\partial y} \right]^2 + \left[u_x \frac{\partial V_y}{\partial x} + u_y \frac{\partial V_y}{\partial y} \right]^2 \quad (17)$$

$$= u_x^2 \left[\left[\frac{\partial V_x}{\partial x} \right]^2 + \left[\frac{\partial V_y}{\partial x} \right]^2 \right] + 2u_x u_y \left[\frac{\partial V_x}{\partial x} \frac{\partial V_x}{\partial y} + \frac{\partial V_y}{\partial x} \frac{\partial V_y}{\partial y} \right] + u_y^2 \left[\left[\frac{\partial V_x}{\partial y} \right]^2 + \left[\frac{\partial V_y}{\partial y} \right]^2 \right] \quad (18)$$

$$= u_x^2 \left| \frac{\partial V}{\partial x} \right|^2 + 2u_x u_y \frac{\partial V}{\partial x} \cdot \frac{\partial V}{\partial y} + u_y^2 \left| \frac{\partial V}{\partial y} \right|^2. \quad (19)$$

The partial derivative of this quantity can be simplified as follows:

$$\frac{\partial}{\partial x} |\nabla V \cdot u|^2 = 2u_x^2 \left[\frac{\partial V}{\partial x} \cdot \frac{\partial^2 V}{\partial x^2} \right] + 2u_x u_y \left[\frac{\partial V}{\partial y} \cdot \frac{\partial^2 V}{\partial x^2} \right] \quad (20)$$

$$= 2u_x \left[u_x \frac{\partial V}{\partial x} + u_y \frac{\partial V}{\partial y} \right] \cdot \frac{\partial^2 V}{\partial x^2} \quad (21)$$

$$= 2u_x (u \cdot \nabla V) \cdot \frac{\partial^2 V}{\partial x^2} \quad (22)$$

since $\partial V/\partial y$ is constant on N . For the same reason, $\partial^2 V/\partial y^2 = 0$ and $\partial^2 V/\partial x^2 = \nabla^2 V$. Therefore, $\partial/\partial x |\nabla V \cdot u|^2$ has a zero crossing whenever $u_x(u \cdot \nabla V) \cdot \nabla^2 V$ does. But $|\nabla V \cdot u|^2$ has an extremum in the x direction whenever $\partial/\partial x |\nabla V \cdot u|^2$ has a zero crossing. ■

Whenever the Laplacian $\nabla^2 V$ has a zero crossing, so must $u_x(u \cdot \nabla V) \cdot \nabla^2 V$, except when $u_x(u \cdot \nabla V) = 0$, which is unlikely because real edges are places with steep gradients. Zero crossings in the Laplacian will therefore almost always correspond to extrema in the magnitude of the directional derivative, with respect to almost all directions. It is possible for the magnitude of the directional derivative to have an extremum without a zero in the Laplacian because the component at right angles to the preferred direction defined by $u \cdot \nabla V$ need not be small. If there is no variation of the field parallel to the edge, then the steepest directional derivative occurs in the direction normal to the edge; and if the variation parallel to the edge is much less than that normal to the edge, as we expect for most images, then the steepest directional derivative occurs in a direction nearly normal to the edge. If we choose u in the x direction, then $u \cdot \nabla V$ will be parallel to $\partial V/\partial x$, so that the above theorem states the component of the Laplacian in the direction parallel to the difference in the flow on both sides of the boundary will have a zero crossing. The Laplacian can fail to have a Laplacian reversal at an edge only if the component of the Laplacian at right angles to the flow difference is not small, which occurs when the normal component of the flow gradient at an edge is changing in direction more rapidly

than it is changing in magnitude. Such situations do not appear to be common in real optical flows, and can occur only when the unfiltered flow is changing appreciably in a neighborhood of the edge for at least one of the two surfaces. For the case of a boundary between two surfaces with distinct uniform flows on each surface, the smoothed Laplacian has a directional zero crossing in all directions except along the boundary. In that direction, the value of the smoothed Laplacian is zero.

The extremum can be either a maximum or a minimum. A maximum is of course desired, and minima are discarded in practice by requiring the slope of the zero crossing to be sufficiently steep. While this is not a guaranteed test, it works in almost all cases because of the Gaussian filtering applied to the images before the Laplacian is calculated. Minima in the gradient usually correspond to areas where the field is uniform, and due to the tails on a Gaussian curve, gradients near the minima tend to be small, with small values for derivatives of all orders.

APPENDIX B

This Appendix contains the analysis showing that the limit of surface flow is equal to boundary flow for the rotation of smooth objects for orthographic projections. Any motion of a rigid body can be described by giving the trajectory of an arbitrary point attached to the body and the instantaneous rotation about some axis passing through that point. Define a set of Cartesian axes (X, Y, Z) with the origin at the distinguished point on the body and with the Z axis along the axis of rotation, and let (r, θ, ϕ) be spherical coordinates with respect to these axes. Let the orientations of the axes (X, Y, Z) be fixed with respect to the axes (x, y, z) of the image plane coordinates, so that the angular velocity of an arbitrary rotation is the same in both coordinate systems. Let the surface of the body be described by

$$r = R(\theta - \psi(t), \phi) \quad (23)$$

where $\psi(0) = 0$, so that $r = R(\theta, \phi)$ at time $t = 0$. The parameter $\alpha = \theta - \psi(t)$ is the longitudinal angle of a point fixed to the surface at $t = 0$, and points with constant values of α and ϕ rotate along with the surface. Since $\theta = \alpha + \psi(t)$, $\omega = d\psi/dt$ gives the angular velocity of the object about the Z axis.

At some particular instant of time, let G be a generating point (r_g, θ_g, ϕ_g) and \mathbf{n} be the unit surface normal at G . Since G is a generating point and orthographic projection is involved, \mathbf{n} will be parallel to the image plane. The normal component of the flow for an arbitrary point $\mathbf{p} = (r, \theta, \phi)$ fixed to the surface is as follows:

$$\begin{aligned} f_x(\mathbf{p}) &= (\boldsymbol{\Omega} \times \mathbf{p}) \cdot \mathbf{n} \\ &= \omega R(\theta - \psi, \phi) \sin \phi [-\mathbf{n}_X \sin \theta + \mathbf{n}_Y \cos \theta] \end{aligned} \quad (24)$$

where $\boldsymbol{\Omega}$ is the vector angular velocity of magnitude ω and oriented along the Z axis. The orientation of $\boldsymbol{\Omega}$ and \mathbf{n} may be changing, but the analysis below is based on the instantaneous values of both quantities at some particular point in time.

The x axis in the image plane is oriented parallel to the constant unit vector \mathbf{n} . Since we are working with an orthographic projection, the x coordinate of the point \mathbf{p} is as follows:

$$x = \mathbf{p} \cdot \mathbf{n} = R(\theta - \psi, \phi) [p(\theta, \phi) \cdot \mathbf{n}] \quad (25)$$

$$[p(\theta, \phi) \cdot \mathbf{n}] = \mathbf{n}_X \sin \phi \cos \theta + \mathbf{n}_Y \sin \phi \sin \theta + \mathbf{n}_Z \cos \phi \quad (26)$$

where p is the unit vector parallel to \mathbf{p} . Since the generating point is on the extremal boundary of the object, x must have an extremum at the generating point for variations in both θ and ϕ . This leads to

$$\begin{aligned} \frac{\partial x}{\partial \theta} = 0 &= \frac{\partial R(\theta - \psi, \phi)}{\partial \theta} [p(\theta, \phi) \cdot \mathbf{n}] \\ &+ R(\theta - \psi, \phi) \frac{\partial}{\partial \theta} [p(\theta, \phi) \cdot \mathbf{n}] \end{aligned} \quad (27)$$

$$\begin{aligned} \frac{\partial x}{\partial \phi} = 0 &= \frac{\partial R(\theta - \psi, \phi)}{\partial \phi} [p(\theta, \phi) \cdot \mathbf{n}] \\ &+ R(\theta - \psi, \phi) \frac{\partial}{\partial \phi} [p(\theta, \phi) \cdot \mathbf{n}] \end{aligned} \quad (28)$$

for $\theta = \theta_g, \phi = \phi_g$. Let x_g denote the x coordinate of the generating point. From (25), the flow of the boundary is as follows:

$$f_b = \frac{dx_g}{dt} = \left[\frac{d}{dt} R(\theta - \psi, \phi) [p(\theta, \phi) \cdot \mathbf{n}] \right]_{\theta=\theta_g, \phi=\phi_g} \quad (29)$$

$$\begin{aligned} &= \frac{\partial R(\theta - \psi, \phi)}{\partial \theta} [p(\theta, \phi) \cdot \mathbf{n}] \left[\frac{d\theta_g}{dt} - \frac{d\psi}{dt} \right] \\ &+ R \frac{\partial}{\partial \theta} [p(\theta, \phi) \cdot \mathbf{n}] \frac{d\theta_g}{dt} \\ &+ \frac{\partial R(\theta - \psi, \phi)}{\partial \phi} [p(\theta, \phi) \cdot \mathbf{n}] \frac{d\phi_g}{dt} \\ &+ R \frac{\partial}{\partial \phi} [p(\theta, \phi) \cdot \mathbf{n}] \frac{d\phi_g}{dt} \end{aligned} \quad (30)$$

evaluated at $\theta = \theta_g, \phi = \phi_g$. From (27), (28), and (26) we get

$$\begin{aligned} f_b &= - \frac{d\psi}{dt} \frac{\partial R(\theta - \psi, \phi)}{\partial \theta} [p(\theta, \phi) \cdot \mathbf{n}] \\ &= \frac{d\psi}{dt} R \frac{\partial}{\partial \theta} [p(\theta, \phi) \cdot \mathbf{n}] \end{aligned} \quad (31)$$

$$= \omega R(\theta_g - \psi, \phi_g) \sin \phi [-\mathbf{n}_X \sin \theta_g + \mathbf{n}_Y \cos \theta_g] \quad (32)$$

$$= f_x(0, 0) \quad (33)$$

using (24) and $d\psi/dt = \omega$. This establishes (2) for arbitrary orientations of the axis of rotation with respect to the image plane, assuming an orthographic projection.

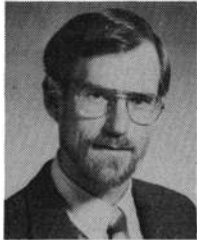
ACKNOWLEDGMENT

M. Kaveh, D. Boley, and L. Qi provided much useful assistance in portions of the formal analysis.

REFERENCES

- [1] S. T. Barnard and W. B. Thompson, "Disparity analysis of images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-2, pp. 333-340, July 1980.

- [2] W. F. Clocksin, "Perception of surface slant and edge labels from optical flow: A computational approach," *Perception*, vol. 9, pp. 253-269, 1980.
- [3] K. Nakayama and J. M. Loomis, "Optical velocity patterns, velocity sensitive neurons, and space perception: A hypothesis," *Perception*, vol. 3, pp. 63-80, 1974.
- [4] H. C. Longuet-Higgins and K. Prazdny, "The interpretation of a moving retinal image," in *Proc. Roy. Soc. London*, vol. B-208, 1980, pp. 385-397.
- [5] D. Marr and E. Hildreth, "Theory of edge detection," in *Proc. Roy. Soc. London*, vol. B-207, 1980, pp. 187-217.
- [6] W. B. Thompson, K. M. Mutch, and V. A. Berzins, "Edge detection in optical flow fields," in *Proc. 2nd Nat. Conf. Artif. Intell.*, Aug. 1982.
- [7] J. L. Crowley and R. M. Stern, "Fast computation of the difference of low-pass transform," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, pp. 212-222, Mar. 1984.
- [8] P. J. Burt, "Fast filter transforms for image processing," *Comput. Graph. Image Processing*, vol. 16, pp. 20-51, 1981.
- [9] D. Marr and S. Ullman, "Directional selectivity and its use in early visual processing," in *Proc. Roy. Soc. London*, vol. B-211, pp. 151-180, 1981.
- [10] K. M. Mutch and W. B. Thompson, "Analysis of accretion and deletion at boundaries in dynamic scenes," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-7, pp. 133-138, 1985.
- [11] W. B. Thompson, K. M. Mutch, and V. A. Berzins, "Analyzing object motion based on optical flow," in *Proc. 7th Int. Conf. Pattern Recog.*, July 1984.
- [12] A. P. Witkin, "Intensity-based edge classification," in *Proc. 2nd Nat. Conf. Artif. Intell.*, Aug. 1982.



William B. Thompson (S'72-M'75) received the Sc.B. degree in physics from Brown University, Providence, RI, in 1970, and the M.S. and Ph.D. degrees in computer science from the University of Southern California, Los Angeles, in 1972 and 1975, respectively.

He is currently an Associate Professor in the Department of Computer Science at the University of Minnesota, Minneapolis. Previously, he was a member of the Image Processing Institute at the University of Southern California. His

primary research interest is in the area of computer vision, with an emphasis on the development of techniques for perceiving spatial organization. In addition, he is a principal in the expert problem solving research group at the University of Minnesota.

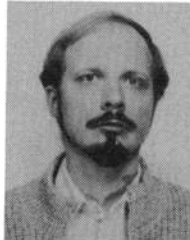
Dr. Thompson is a member of the American Association for Artificial Intelligence and the Association for Computing Machinery.



Kathleen M. Mutch (S'80-M'83) received the M.S. and Ph.D. degree in computer science from the University of Minnesota, Minneapolis, in 1981 and 1983, respectively.

She is currently an Assistant Professor in the Department of Computer Science at Arizona State University, Tempe. Her research interests include time-varying image analysis and applications of computer vision.

Dr. Mutch is a member of the Association for Computing Machinery, SIGART, SIGCAPH, and the American Association for Artificial Intelligence.



Valdis A. Berzins (S'76-M'78) received the S.B. degree in physics and the S.M. and E.E. degrees in electrical engineering in 1975, and the Ph.D. degree in computer science in 1979, all from the Massachusetts Institute of Technology, Cambridge.

He is presently Assistant Professor of Computer Science at the University of Minnesota. His research interests include database support for computer aided design, software engineering, and image analysis.